MOL.19980728.0005

CRWMS/M&O

Calculation Cover Sheet

Complete only applicable items.

1. QA: L
Page: 1 Of: 63

2. Calculation Title CRC Reactivity Calculations for Sequoyah Unit 2			
3. Document Identifier (including Revision Number) B00000000-01717-0210-00006 REV 00			4. Total Pages 63
5. Total Attachments 6	6. Attachment Numbers - Number of pages in each I-656, II-1, III-1, IV-1, V-1, VI-1 (See box 10)		
	Print Name	Signature	Date
7. Originator	John M. Scaglione	John M. Scaglione	5/13/98
8. Checker	Ken D. Wright	Kenneth D. Wright	6/4/98
9. Lead Design Engineer	Dan A. Thomas	D A Thomas	06/04/98

10. Remarks

Attachments II through VI are contained on an attachment tape which has been moved to Reference 7.13. The #'s shown in box 6 refer to the # of pages in the hard-copy listing of each attachment's file content on the tape. Attachments V and VI were added after the release of the official check copy (REV 00A) and contains revised data. The information listed in these attachments supersede the previously documented data. RG compliance review by WEWallin 5/13/98

The information in Attachments I and II have been checked. KDW 6/4/98

The electronic attachments have been checked. KDW 6/4/98

Revision History

11. Revision No.	12. Date Approved	13. Description of Revision
REV 00		Initial Issuance

Table of Contents

1. Purpose

The purpose of this calculation is to document the Sequoyah Unit 2 pressurized water reactor (PWR) reactivity calculations performed as part of the commercial reactor critical (CRC) evaluation program. CRC evaluation reactivity calculations are performed at a number of statepoints, representing reactor start-up critical conditions at either beginning of life (BOL), beginning of cycle (BOC), or mid-cycle when the reactor resumed operation after a shutdown. The CRC evaluations support the development and validation of the neutronics models used for criticality analyses involving commercial spent nuclear fuel in a geologic repository.

2. Method

The calculational method used to perform the Sequoyah Unit 2 core reactivity calculations consisted of using the MCNP code (Ref. 7.1) to calculate the effective neutron multiplication factor (k_{eff}) for the various critical core configurations. Each of the critical core configurations were modeled in detail using measured critical conditions. The various fuel assemblies were modeled explicitly in the critical core configurations. The SAS2H code of the SCALE 4.3 modular code system (Ref. 7.2) was used to deplete the various fuel assemblies as necessary to obtain the burned fuel isotopics for use in the reactivity calculations documented herein. These fuel assembly depletion calculations are documented in Reference 7.3. The Sequoyah Unit 2 CRC configurations are actual PWR cores which contained fuel loadings that varied from all fresh fuel (BOL) to a mixture of fresh and burned fuel (BOC) to a mixture of all burned fuel (mid-cycle restart).

3. Assumptions

Not Used

4. Use of Computer Software

4.1. Software Approved for QA Work

4.1.1. MCNP

The MCNP code was used to calculate the k_{eff} of the Sequoyah Unit 2 critical reactor configurations. The software specifications are as follow:

- Program Name: MCNP
- Version/Revision Number: Version 4B2
- CSCI Number: 30033 V4BLV
- Computer Type: HP 9000 Series Workstations

The input and output files for the various MCNP calculations are documented in the attachments to this calculation file as described in Sections 5 and 8 (the attachment tape has been moved to Reference 7.13), such that an independent repetition of the software use may be performed. The MCNP software used was: (a) appropriate for the application of commercial reactor k_{eff} calculations, (b) used only within the range of validation as documented throughout References 7.1 and 7.4, (c) obtained from the Software Configuration Manager in accordance with appropriate procedures.

4.2. Software Routines

4.2.1. MACE

- Title: MCNP Accessory for CRC Evaluations (MACE)
- Version/Revision Number: Version 3

The MACE code automates the production of MCNP input decks to calculate the k_{eff} of the critical reactor configurations in the CRC evaluations. The input and output for the various MACE calculations are documented in Sections 5 and 8, such that an independent repetition of the software routine use may be performed. The description of the MACE software routine is provided in Attachment I of this document. This description documentation contains the following information:

- Descriptions and equations of mathematical algorithms
- Description of software routine including execution environment
- Range of input parameter values for which results were verified
- Identification of any limitations on software routine applications or validity
- Reference list of all documentation relevant to the qualification
- Directory listing of executable and data files
- Computer listing of source code

The MCNP input decks that were produced for the Sequoyah Unit 2 CRC evaluations and presented in this calculation file serve as the test cases for MACE. These input decks were thoroughly reviewed to verify that MACE was performing correctly.

4.2.2. Excel

- Title: Excel
- Version/Revision Number: Microsoft® Excel 97

The Excel spreadsheet program was used for simple numeric calculations as documented in Section 5 of this calculation file. The user-defined formulas, inputs, and results were documented in sufficient detail in Section 5 to allow an independent repetition of the various computations.

5. Calculation

The Sequoyah Unit 2 CRC reactivity calculations are detailed calculations of the neutron multiplication factor for actual critical reactor configurations. This analysis provides the geometry, material, core loading, and calculational control descriptions for each CRC reactivity calculation performed with MCNP. The MCNP input decks for each CRC reactivity calculation documented in this analysis were created with the MACE software routine. Complete documentation of the MACE software routine and MACE input deck preparation instructions are provided in Attachment I. The MACE input decks used to create each of the MCNP input decks are presented in Attachment II. The MACE generated MCNP input decks are presented in Attachment III. The MCNP output decks are presented in Attachment IV (the attachment tape has been moved to Reference 7.13). The k_{eff} results for each CRC reactivity calculation are presented in Section 6.

5.1. Sequoyah Unit 2 CRC Reactivity Calculations

The Sequoyah Unit 2 CRC reactivity calculations represent three critical statepoints at which either BOL, BOC, or mid-cycle reactor start-ups were performed. Table 5.1-1 presents a listing of these three statepoints by reactor cycle and effective full-power day (EFPD) time.

Table 5.1-1. Sequoyah Unit 2 CRC Reactivity Calculations

Cycle	Critical Statepoint EFPD Time
1	0.0
3	0.0
3	210.9

5.2. Sequoyah Unit 2 MCNP Geometrical Descriptions

The MCNP models for the Sequoyah Unit 2 PWR incorporated detailed and explicit representations of the fuel assemblies and reactor core components. Extensive fuel assembly and core modeling was incorporated for regions beyond the extent of the active fuel in the axial direction to ensure that neutron leakage was correctly simulated. Actual core loading patterns were utilized in all of the critical configuration models. Core symmetry was used wherever possible to minimize the number of unique fuel assembly descriptions that were required. The use of core symmetry also served to expedite the k_{eff} calculations. The depleted fuel in each assembly was composed of sixteen unique, axially delineated, fuel compositions. These depleted fuel compositions were calculated with SAS2H as documented throughout Reference 7.3. Burnable poison rod assemblies (BPRAs) and rod cluster control assemblies (RCCAs) were modeled explicitly in the core locations corresponding to the measured critical conditions at the various statepoints. The average system temperature and soluble boron concentration that was measured at each critical statepoint was utilized in the MCNP models. Sections 5.2.1 through 5.2.7 discuss the MCNP geometric modeling details for the various components of the Sequoyah Unit 2 CRC configurations.

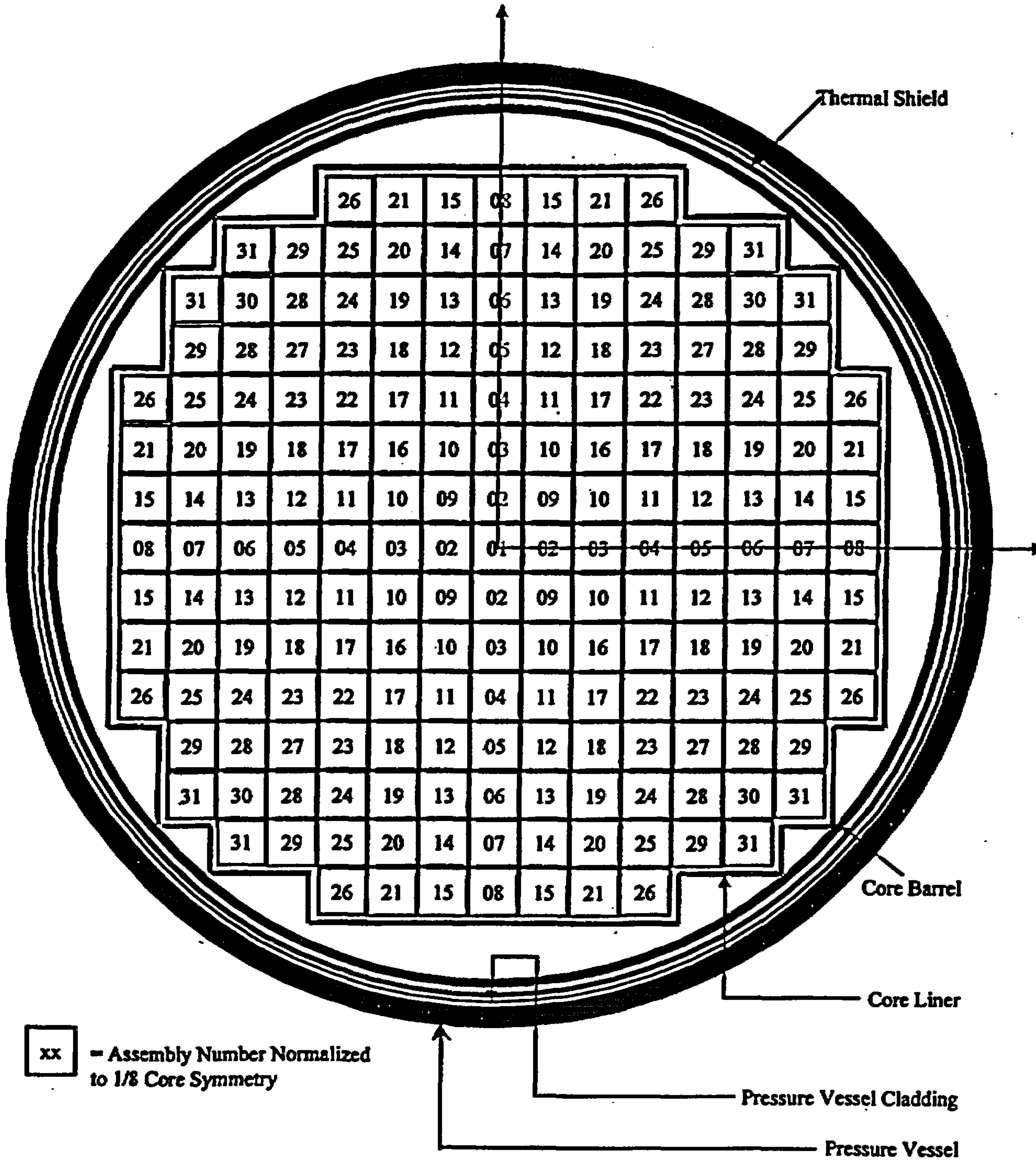
5.2.1. Sequoyah Unit 2 Reactor Core Geometric Description

The Sequoyah Unit 2 PWR is a Westinghouse reactor core design consisting of 193, 17x17 cell lattice, fuel assemblies (p. 6, Ref. 7.11). A core liner surrounds the periphery fuel assemblies in the core. The periphery of the reactor consists of the core barrel, the thermal shield, the pressure vessel cladding, and the pressure vessel. These peripheral components are separated by regions of moderator (borated water). A radial view of the reactor internals is shown in Figure 5.2.1-1. The height of the active fuel region in the core is 365.76 cm (p. 6, Ref. 7.11). The assembly pitch in the core is 21.50364 cm (p. 6, Ref. 7.11). Table 5.2.1-1 presents the dimensions from the center of the core to the outside surface of the pressure vessel. An axial view of the reactor core internals is shown in Figure 5.2.1-2. Due to their geometric complexity and low neutronic importance, the components in the reactor regions above and below the upper and lower end-fittings of the fuel assemblies are homogenized into one material composition for each region. These reactor regions above and below the fuel assembly end-fittings are modeled as uniform geometric cells, each containing the appropriately homogenized material composition. The homogenization of these components will allow MCNP to simulate the average axial leakage from the system.

Table 5.2.1-1. Sequoyah Unit 2 Reactor Radial Dimensions[1] (p. 3, Ref. 7.11)

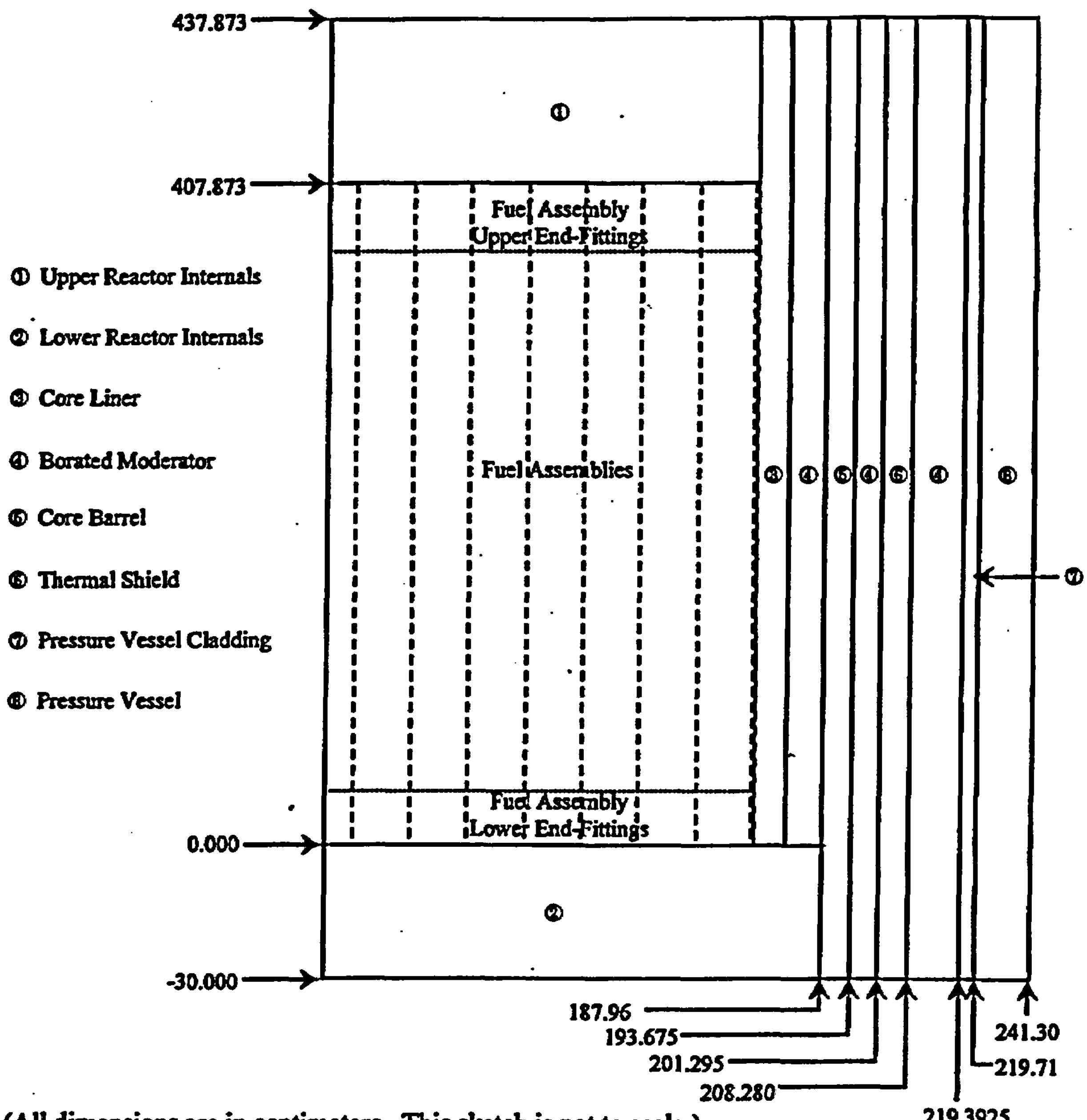
Description	Thickness (cm)	Outer Radius (cm)
Core center	—	0
Half of FA-1	10.70102	10.70102
Water	0.10160	10.80262
FA-2	21.40204	32.20466
Water	0.10160	32.30626
FA-3	21.40204	53.70830
Water	0.10160	53.80990
FA-4	21.40204	75.21194
Water	0.10160	75.31354
FA-5	21.40204	96.71558
Water	0.10160	96.81718
FA-6	21.40204	118.21922
Water	0.10160	118.32082
FA-7	21.40204	139.72286
Water	0.10160	139.82446
FA-8	21.40204	161.22650
Water	0.19350	161.42000
Core liner	2.85750	164.27750
Water	23.68250	187.96000
Core barrel	5.71500	193.67500
Water	7.62000	201.29500
Thermal shield	6.98500	208.28000
Water	11.11250	219.39250
Vessel cladding	0.31750	219.71000
Pressure vessel	21.59000	241.30000

[1] Dimensions are for a standard (STD) Westinghouse 17x17 fuel assembly core loading.



(This sketch is not to scale.)

Figure 5.2.1-1. Radial View of the Sequoyah Unit 2 Reactor Internals as Modeled in MCNP (p. 4, Ref. 7.11)



(All dimensions are in centimeters. This sketch is not to scale.)

Figure 5.2.1-2. Axial View of the Sequoyah Unit 2 Reactor Internals as Modeled in MCNP
(Radial Dimensions: p. 4, Ref. 7.11)
(Axial Dimensions: p. 8, Ref. 7.11)

5.2.2. Sequoyah Unit 2 Fuel Assembly Geometric Descriptions

The Sequoyah Unit 2 CRC configurations contained fuel assemblies from six different fuel batches. Fuel assemblies from the various fuel batches were inserted into the reactor core in different combinations for each cycle. Only one fuel assembly design was represented in the various fuel batches-- Westinghouse standard fuel assembly design (STD). The fuel assembly design utilizes 17x17 pin cell lattices with a pin cell lattice pitch of 1.25984 cm (p. 6, Ref. 7.11). The specifications for each fuel batch are summarized in Table 5.2.2-1.

Table 5.2.2-1. Fuel Assembly Specification Summary (p. 22, Ref. 7.11)

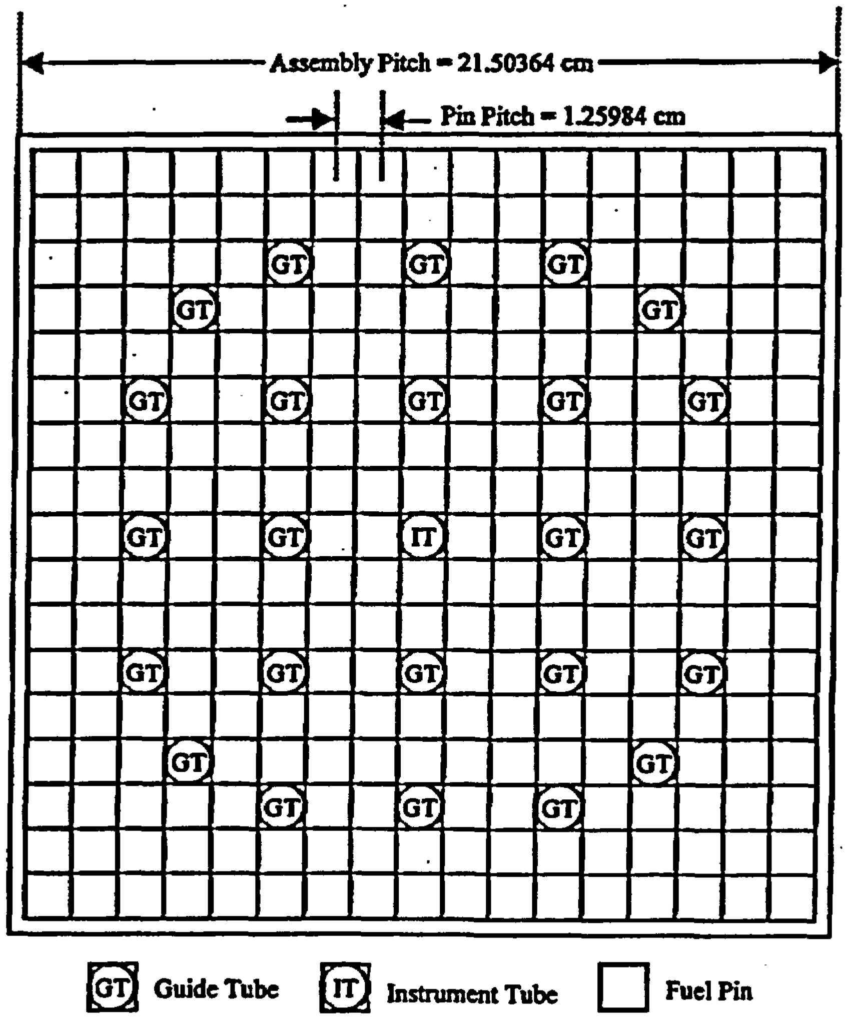
Fresh Batch Cycle	Fuel Batch	FA[1] Type	Wt%[5] U-235	kg U per FA	FP[2] Pellet OD[3] (cm)	FP Clad OD (cm)	FP Clad ID[4] (cm)	FA Grid Material
1	1	STD	2.10	458.88	0.819150	0.94996	0.83566	Inconel
1	2	STD	2.60	458.88	0.819150	0.94996	0.83566	Inconel
1	3	STD	3.10	458.88	0.819150	0.94996	0.83566	Inconel
2	4	STD	3.50	458.97	0.819150	0.94996	0.83566	Inconel
3	5A	STD	3.80	461.50	0.819150	0.94996	0.83566	Inconel
3	5B	STD	3.60	460.71	0.819150	0.94996	0.83566	Inconel

[1] FA = Fuel Assembly, [2] FP = Fuel Pin, [3] OD = Outer Diameter, [4] ID = Inner Diameter, [5] Wt% = Weight Percent

Each fuel assembly contains one instrument tube and twenty-four guide tubes (p. 6, Ref. 7.11). The guide tubes consist of two axial sections each having different dimensions (p. 22, Ref. 7.11). Table 5.2.2-2 summarizes the instrument tube and guide tube specifications. The fuel pin, guide tube, and instrument tube positions are shown in Figure 5.2.2-1.

Table 5.2.2-2. Instrument and Guide Tube Specification Summary (p. 22, Ref. 7.11)

Description	Assembly Design	Material	OD (cm)	ID (cm)
Instrument Tube	STD	Zircaloy-4	1.22428	1.14300
Guide Tube Upper Region	STD	Zircaloy-4	1.22428	1.14300
Guide Tube Lower Region	STD	Zircaloy-4	1.08966	1.00838



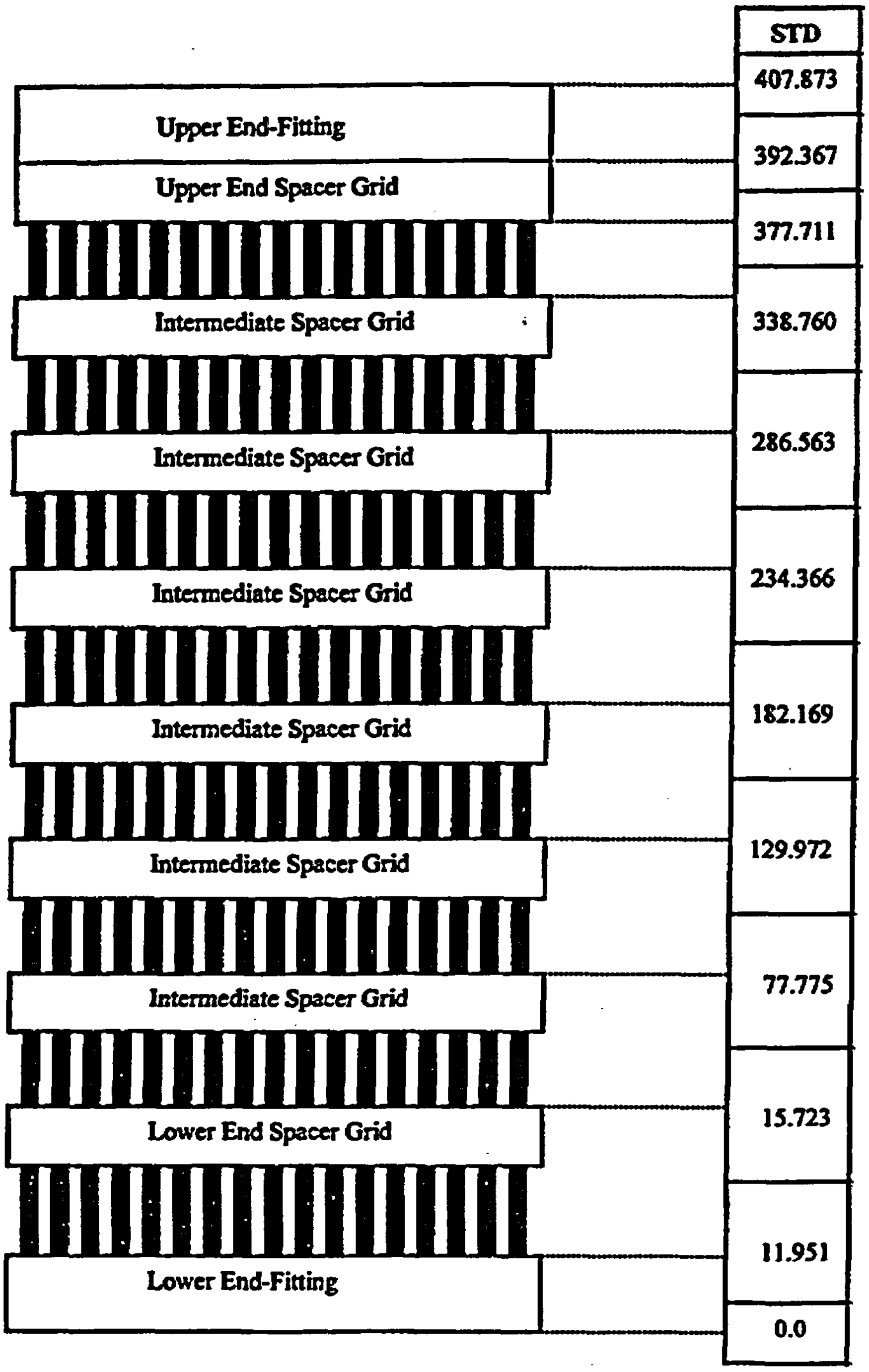
(This sketch is not to scale.)

Figure 5.2.2-1. Fuel Pin, Guide Tube, and Instrument Tube Locations in Fuel Assembly (p. 7, Ref. 7.11)

Each fuel assembly has six intermediate spacer grids and one lower end spacer grid (p. 8, Ref. 7.11). The intermediate spacer grids are made of Inconel in the STD fuel assembly design (p. 22, Ref. 7.11). The intermediate spacer grid height and volume is summarized in Table 5.2.2-3. The spacer grid material volume was homogenized with the corresponding borated moderator volume and placed uniformly between the assembly rods and within the assembly pitch boundaries in each spacer grid location. The axial locations of the spacer grids are shown in Figure 5.2.2-2. The lower end-fitting of each fuel assembly design is modeled as a homogenized region, 11.951 cm in height (p. 8, Ref. 7.11), distributed uniformly between and below the fuel rods, guide tubes, and instrument tubes. The upper end-fitting of each fuel assembly design is modeled as a homogenized region, 15.506 cm in height (p. 8, Ref. 7.11), distributed uniformly between and above the fuel rods, guide tubes, instrument tubes, burnable poison rods (BPRs), and RCCAs.

Table 5.2.2-3. Intermediate and Lower End Spacer Grid Height and Volume (p. 6, Ref. 7.11)

Description	STD Assembly
Material	Inconel
Height (cm)	3.35788
Volume (cm^3)	95.234

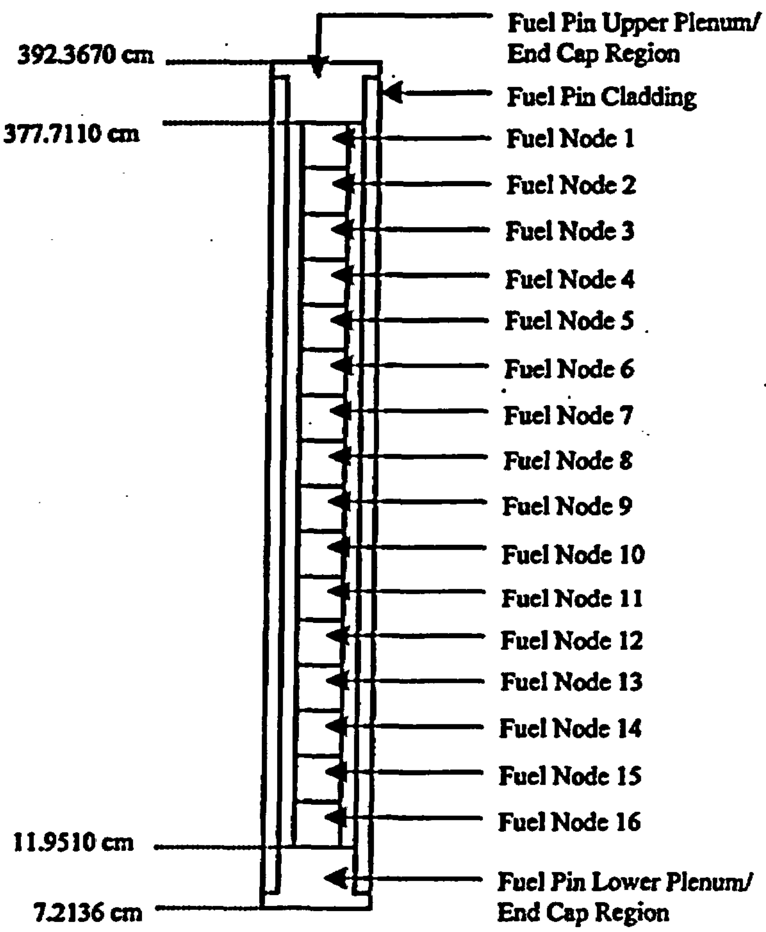


(All dimensions are in centimeters. This sketch is not to scale.)

Figure 5.2.2-2. Axial View of STD Fuel Assembly (p. 8, Ref. 7.11)

5.2.3. Fuel Pin Geometric Description

The cross-sectional view along the length of a fuel pin is shown in Figure 5.2.3-1, to present the modeled axial dimensions. The radial dimensions of the fuel pins for each fuel batch are presented in Table 5.2.2-1. The fuel pins in each assembly are modeled with sixteen axial fuel nodes, each representing a unique fuel composition corresponding to the fuel node depletion. The height of each fuel node is 22.86 cm (p. 36, Ref. 7.11). The fuel pin upper end cap and upper plenum materials are homogenized and distributed uniformly throughout the plenum and end cap region. The fuel pin lower end cap and lower plenum materials are also homogenized and distributed uniformly throughout the plenum and end cap region.



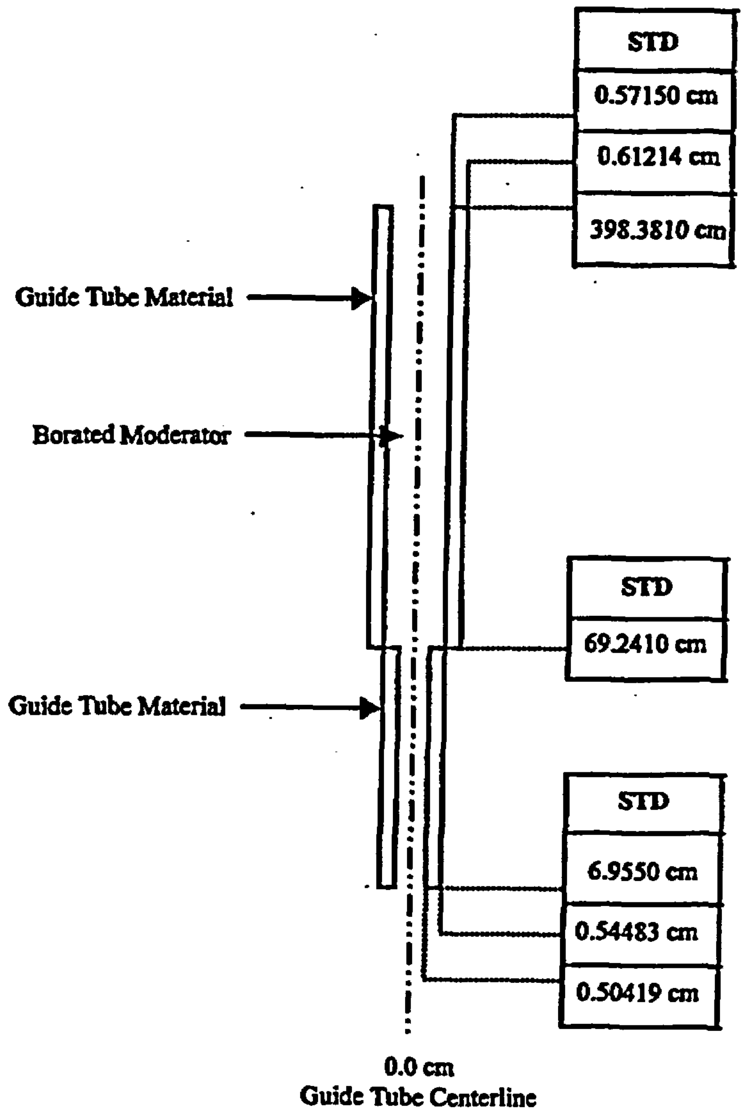
(This sketch is not to scale.)

Note: The 0.0 cm reference point for the axial dimensions is located at the bottom of the lower end-fitting.

Figure 5.2.3-1. Fuel Pin Geometry Model in MCNP (p. 12, Ref. 7.11)

5.2.4. Guide Tube Geometric Description

The cross-sectional view along the length of a guide tube is presented in Figure 5.2.4-1. The MCNP model dimensions are shown in Figure 5.2.4-1. The guide tubes are modeled explicitly into the upper and lower end-fittings of the fuel assembly.



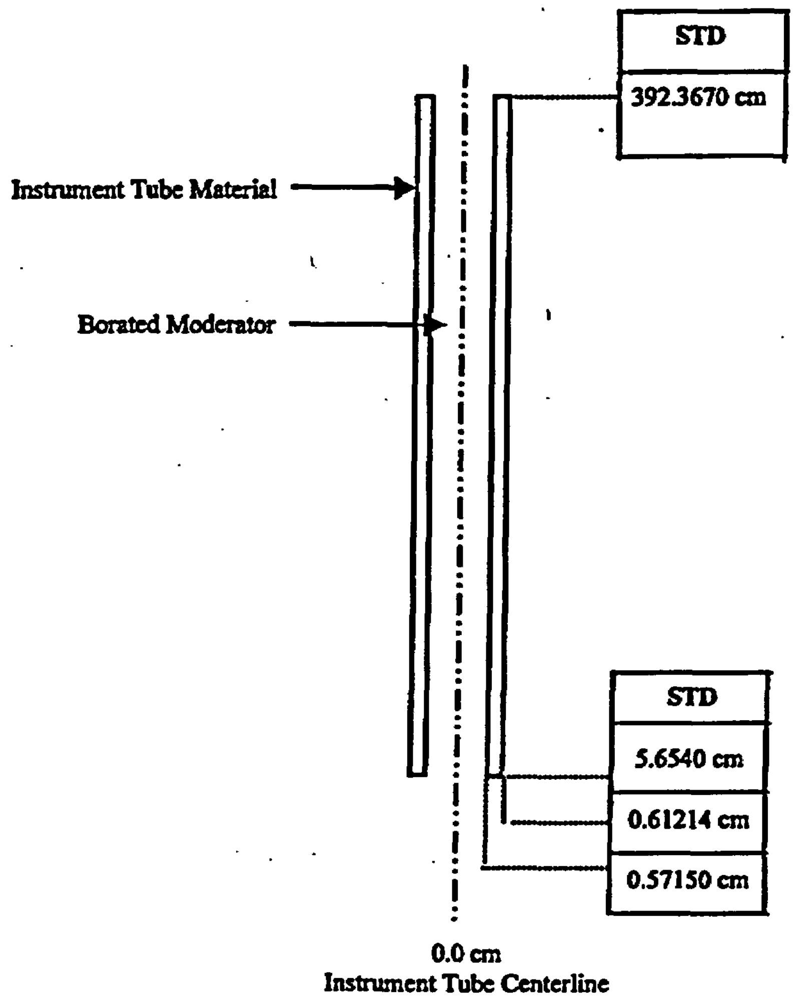
(This sketch is not to scale.)

Note: The 0.0 cm reference point for the axial dimensions is located at the bottom of the lower end-fitting.

Figure 5.2.4-1. Guide Tube Geometry Model in MCNP
(Radial Dimensions: p. 22, Ref. 7.11)
(Axial Dimensions: p. 10, Ref. 7.11)

5.2.5. Instrument Tube Geometric Description

The cross-sectional view along the length of an instrument tube is presented in Figure 5.2.5-1. The MCNP model dimensions are shown in Figure 5.2.5-1. The instrument tubes are modeled explicitly up to the bottom of the upper end-fitting and into the lower end-fitting of the fuel assembly. Truncating the instrument tube at the bottom of the upper end-fitting of the assembly has a negligible effect on the reactor core k_{eff}.



(This sketch is not to scale.)

Note: The 0.0 cm reference point for the axial dimensions is located at the bottom of the lower end-fitting.

Figure 5.2.5-1. Instrument Tube Geometry Model in MCNP
(Radial Dimensions: p. 22, Ref. 7.11)
(Axial Dimensions: p. 11, Ref. 7.11)

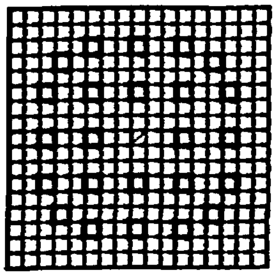
5.2.6. BPRA Geometric Description

The BPRAs utilized in Sequoyah Unit 2 are composed of either of two types of BPRs: Pyrex or WABA (Wet Annular Burnable Absorber). The Pyrex BPRs use B_2O_3-SiO_2 as the absorber material (p. 22, Ref. 7.11). The WABA BPRs use B_4C-Al_2O_3 as the absorber material (p. 22, Ref. 7.11). Both the Pyrex and WABA BPRs are annular in design. However, the Pyrex BPR has a dry annular gap, and the WABA BPR has a wet annular gap. The specifications for both the Pyrex and WABA BPRs are summarized in Table 5.2.6-1. The various BPRAs utilized in Sequoyah Unit 2 were composed of a number of either Pyrex or WABA BPRs arranged in specific geometric patterns. Figure 5.2.6-1 shows the different geometric patterns available for BPRAs in Sequoyah Unit 2. The burnable poison (BP) in each BPRA is depleted during reactor operation. This BP depletion was modeled in the fuel depletion calculations (p. 9, Ref. 7.3). The depleted BP material was retained for use in the MCNP models. For the depletion calculations, the BP was delineated into axial regions corresponding to the axial fuel node delineation (p. 16, Ref. 7.3). The cross-sectional view along the length of a modeled Pyrex and a modeled WABA BPR is shown in Figure 5.2.6-2. The upper cap and upper stem of the Pyrex and WABA BPRs were neglected in the MCNP models. The upper cap and upper stem components have negligible reactivity worth. In the actual WABA BPRs, the water annulus extends through the upper and lower plenum regions. In the MCNP models, the water in the annulus of the WABA BPRs is homogenized with the other materials in the upper and lower plenum regions. This approximation has no effect on reactivity.

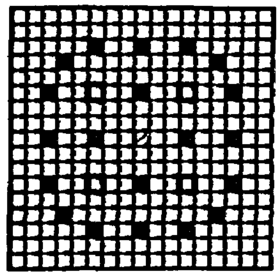
Table 5.2.6-1. Pyrex and WABA BPR Specification Summary (p. 22, Ref. 7.11)

Description	Pyrex	WABA
BP Material	B_2O_3-SiO_2	B_4C-Al_2O_3
Boron Loading	12.5 wt% B_2O_3 0.00624 g B-10/cm	14.0 wt% Al_2O_3 0.006165 g B-10/cm
BP Density (g/cm^3)	2.299[a]	2.593[a]
BP Outer Diameter (OD) (cm)	0.85344	0.8077
BP Inner Diameter (ID) (cm)	0.48260	0.7061
BPR Clad Material	Stainless Steel (Type 304)	Zircaloy-4
BPR Outer Clad OD (cm)	0.96774	0.96774
BPR Outer Clad ID (cm)	0.87376	0.83570
BPR Inner Clad OD (cm)	0.46101	0.67820
BPR Inner Clad ID (cm)	0.42799	0.57150

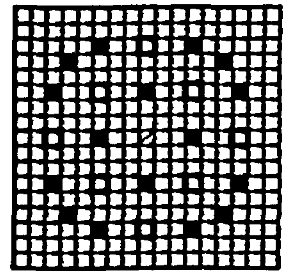
[a] The calculations performed to obtain these densities, using the information provided on page 22 of Reference 7.11, are documented on pages 10 and 11 of Reference 7.3.



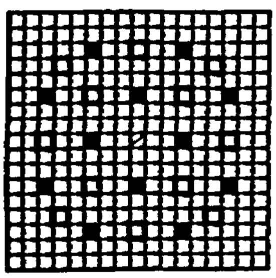
24 Guide Tubes and 1 Instrument Tube



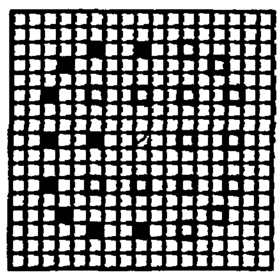
20 Burnable Poison Rods



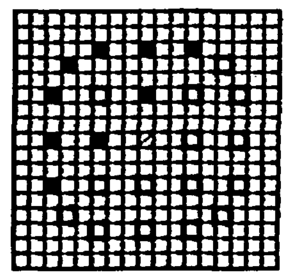
16 Burnable Poison Rods



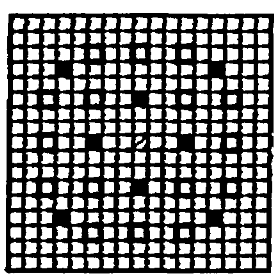
12 Burnable Poison Rods



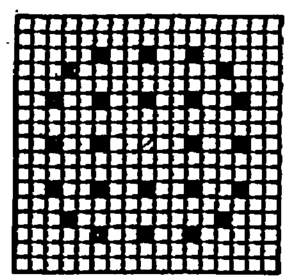
10 Burnable Poison Rods (BPRs toward core center)



9 Burnable Poison Rods



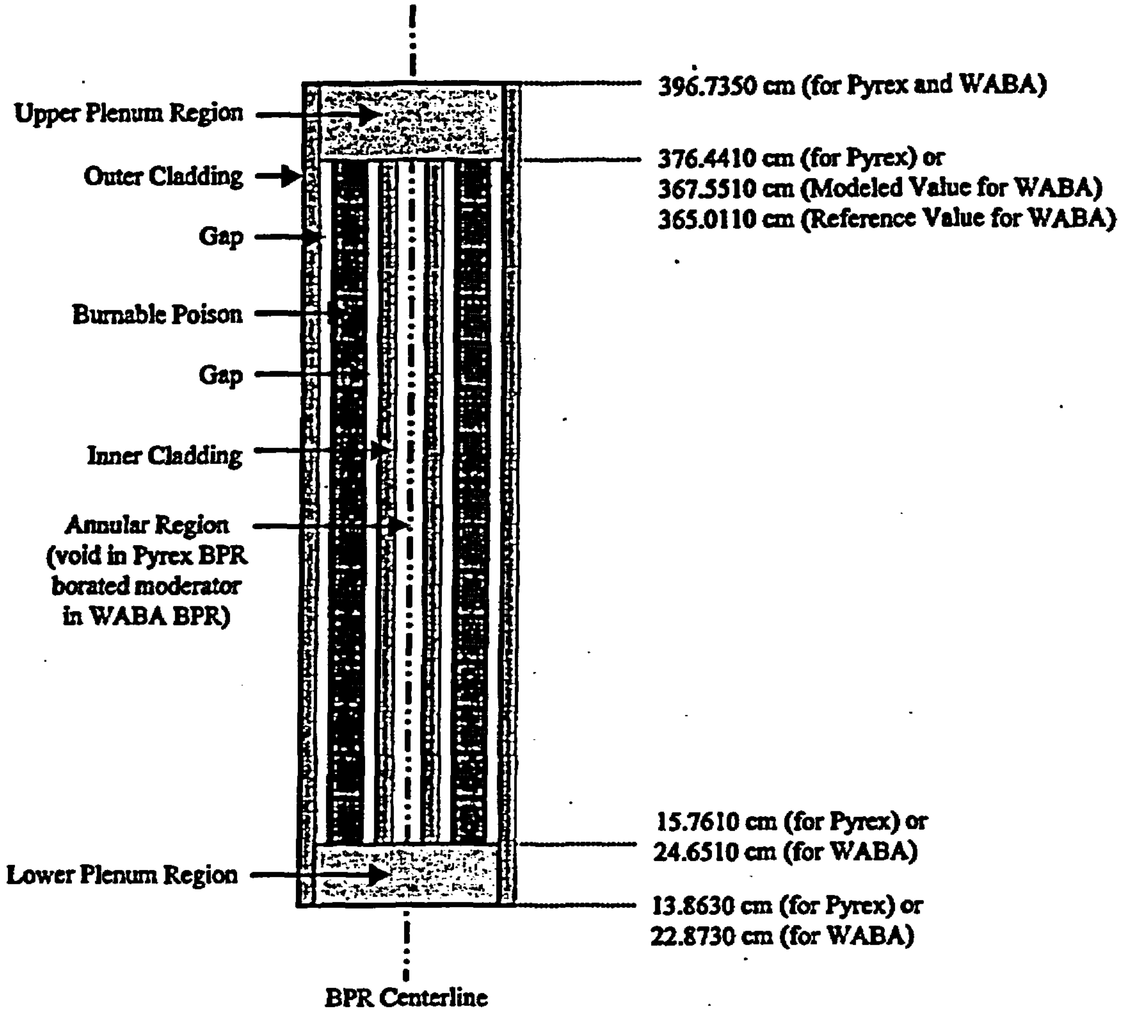
8 Burnable Poison Rods



24 Burnable Poison Rods

Instrument Tube   Guide Tube   Burnable Poison Rod   Fuel Pin

Figure 5.2.6-1. BPR Loading Patterns for Sequoyah Unit 2 BPRAs (p. 28, Ref. 7.11)



(This sketch is not to scale.)

Note: The 0.0 cm reference point for the axial dimensions is located at the bottom of the lower end-fitting.

Figure 5.2.6-2. Cross-Sectional View Along Length of Pyrex or WABA BPR
(Pyrex Dimensions: p. 17, Ref. 7.11)
(WABA Dimensions: p. 18, Ref. 7.11)

5.2.7. RCCA Geometric Description

A RCCA is composed of twenty-four control rods (CRs) distributed such that each guide tube has an inserted CR and all CRs are at the same height in the assembly. The CR specifications are summarized in Table 5.2.7-1. The Sequoyah Unit 2 reactor contains four RCCA banks that may be inserted into the core during startup and operation. Each RCCA in a given bank is moved up or down simultaneously. Each of the four RCCA banks is at a specified axial location in each CRC statepoint reactivity calculation. Table 5.2.7-2 shows the RCCA bank positions in the core for each of the CRC statepoint reactivity calculations. The absorber material of the CRs was modeled with a maximum height of 360.68 cm depending on the depth of the RCCA bank insertion (p. 14, Ref. 7.11). The CRs were always explicitly modeled to the top of the fuel assembly upper end-fitting. The truncation of the RCCA at the top of the assembly upper end-fittings is acceptable due to the decreasing reactivity worth of regions extending beyond the length of the active fuel. If the RCCA bank was partially inserted, the absorber material in the CRs was modeled explicitly from the top of the upper end-fitting to the depth of insertion. The CR lower end-plug was modeled inside the CR cladding directly below the absorber material. A cross-sectional view along the length of the CR is shown in Figure 5.2.7-1.

Table 5.2.7-1. RCCA Control Rod Geometric Specification Summary (p. 22, Ref. 7.11)

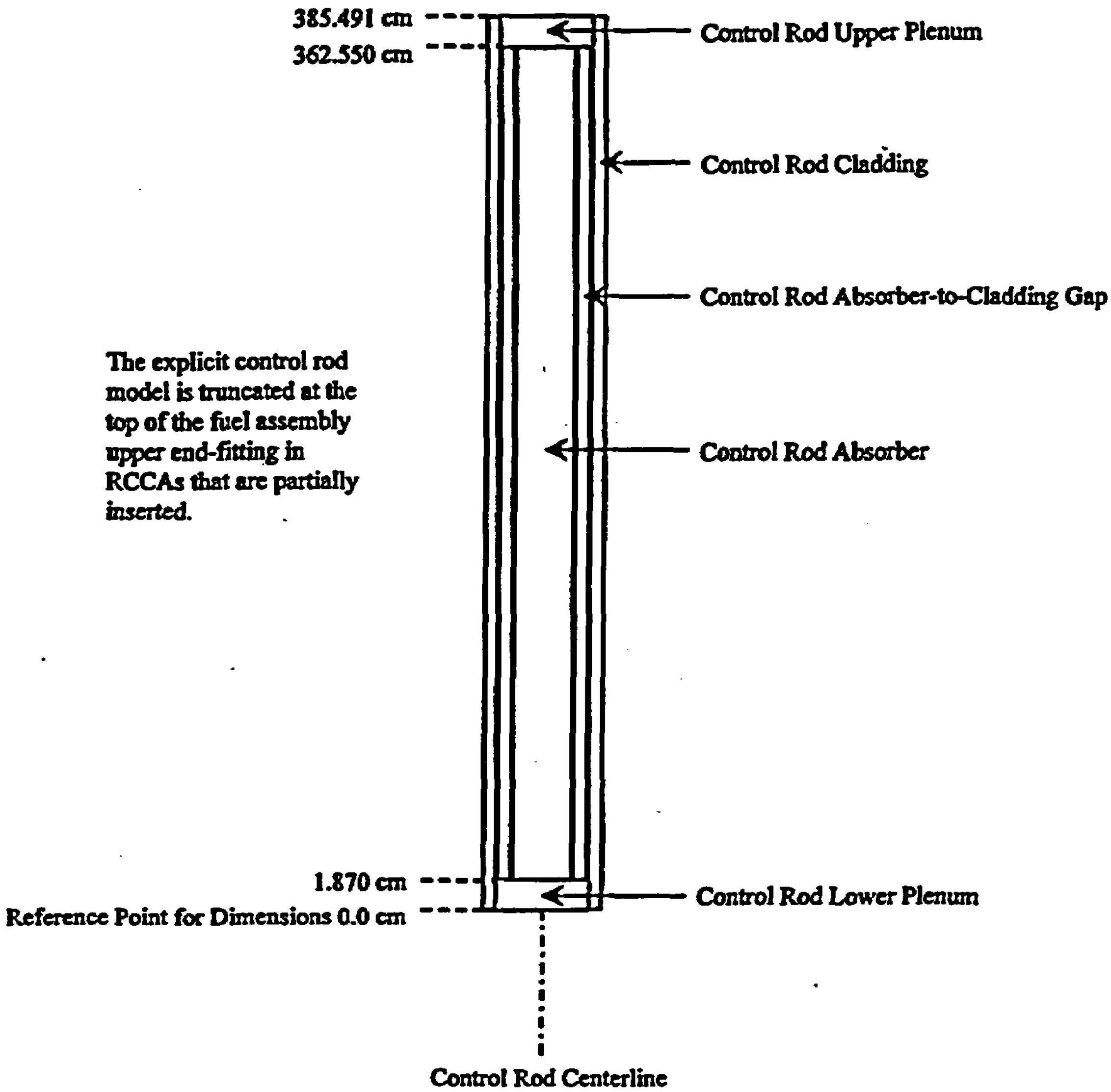
Pellet Material	Ag-In-Cd
Fraction of Pellet Materials	Ag (80 wt%), In (15 wt%), Cd (5 wt%)
Pellet Density	10.16 g/cm^3
Pellet Outer Diameter	0.86614 cm
Clad Material	Stainless Steel (Type 304)
Clad Outer Diameter	0.96774 cm
Clad Inner Diameter	0.87376 cm

Table 5.2.7-2. RCCA Bank Insertion Heights for the Sequoyah Unit 2 CRC Statepoints[1] (p. 63, Ref. 7.11)

Cycle	Statepoint EFPD	Bank CA	Bank CB	Bank CC	Bank CD
1	0.0	WD[2]	WD	WD	267.0
3	0.0	WD	WD	WD	328.0
3	210.9	WD	WD	WD	285.0

[1] The RCCA bank insertion heights are presented as the distance in centimeters between the bottom of the CR absorber material and the bottom of the active fuel.

[2] WD means that the RCCA bank is 100% withdrawn. This corresponds to a height of 378.0 cm in the table. The exact RCCA insertion height at the 100% withdrawn position was not specified in Reference 7.11. The value of 378.0 cm was an acceptable approximation since the RCCA absorber material was placed outside of the active fuel region.



(This sketch is not to scale.)

Note: Due to the axial position of the RCCA banks in the CRC configurations, modeling of the CR upper plenum was not required in any of the MCNP calculations for Sequoyah Unit 2. The 0.0 cm reference point for the axial dimensions is located at the bottom of the lower end-fitting.

Figure 5.2.7-1. Cross-Sectional View Along the Length of a Control Rod (p. 14, Ref. 7.11)

5.3. Sequoyah Unit 2 MCNP Material Descriptions

The material descriptions used in the MCNP CRC reactivity calculations correspond to the actual reactor component materials. Components with detailed geometric features were homogenized where appropriate. The homogenization of these materials preserves the average neutron interaction rate such that the reactivity worth of these materials in the system is approximated. All homogenizations are based on the explicit volumes of the various component materials in the regions of interest. The depleted fuel and depleted burnable poison materials utilized in the MCNP reactivity calculations are obtained from depletion calculations performed using the SAS2H code in the SCALE 4.3 Modular Code System (Ref. 7.2). Detailed descriptions of the fuel and burnable poison depletion calculations are documented throughout Reference 7.3.

5.3.1. MCNP Cross Section Libraries

The MCNP cross section libraries utilized in the reactivity calculations are one of the primary components of the calculation that determines whether or not the neutronic behavior of the system is simulated correctly. Table 5.3.1-1 lists all of the MCNP cross section library identifiers (ZAID's) utilized in the CRC reactivity calculations documented in this calculation file. The MCNP ZAID's are used to identify the cross section libraries. The ZAID consists of a 5 integer element and isotope identifier followed by a cross section library designation suffix. The first one or two integers in the ZAID refer to the atomic number of the corresponding element. The three integers preceding the decimal always refer to the isotopic mass number. The ZAID suffixes presented in Table 5.3.1-1 correspond to libraries compiled from either ENDF/B-V, ENDF/B-VI, LANL/T-2, or LLNL evaluated cross section data sets. The atom percent in nature of the various isotopes presented in Table 5.3.1-1 were obtained from Reference 7.5. The atomic weight ratios, temperatures, library names, and data sources were obtained from Attachment I of Reference 7.12.

The cross section libraries used for the various isotopes and elements do not correspond to the temperature at which these isotopes and elements exist in the critical configurations. The U-235 and U-238 cross section libraries were processed at 587.0 K. The effects of temperature on the U-238 cross sections dominate with respect to the effects of temperature on the other isotopic and elemental cross sections. The majority of the other cross section libraries utilized in the MCNP calculations were processed at 294.0 K. Some less significant isotopic and elemental cross section libraries were processed at 0 K.

The isotopes used in the fuel of the MCNP calculations represent the majority of the isotopes present in the actual material. However, cross section libraries for some of the less significant isotopes were not available in the standard cross section package that accompanies the MCNP software distribution. The isotopes not modeled in fuel of the MCNP calculations have a relatively low reactivity worth due to a combination of their cross sections and low abundance.

Table 5.3.1-1. MCNP Cross Section Libraries Used in the CRC Reactivity Calculations

Element / Isotope	MCNP ZAID	Atom % in Nature	Atomic Wt. Ratio [1]	Temp. (K)	Library Name	Data Source
H-1	1001.50c	99.985	0.999167	294.0	rmccs	ENDF/B-V.0

Table 5.3.1-1. MCNP Cross Section Libraries Used in the CRC Reactivity Calculations

Element / Isotope	MCNP ZAID	Atom % in Nature	Atomic Wt. Ratio [1]	Temp. (K)	Library Name	Data Source
H-3	1003.50c	0.0	2.990140	294.0	rmccs	ENDF/B-V.0
He-4	2004.50c	99.999	3.968219	294.0	rmccs	ENDF/B-V.0
Li-6	3006.50c	7.5	5.963450	294.0	rmccs	ENDF/B-V.0
Li-7	3007.55c	92.5	6.955733	294.0	rmccs	ENDF/B-V.2
Be-9	4009.50c	100.0	8.934763	294.0	rmccs	ENDF/B-V.0
B-10	5010.50c	19.400 [2]	9.926922	294.0	rmccs	ENDF/B-V.0
B-11	5011.56c	80.600 [2]	10.914730	294.0	newxs	LANL/T-2
C-nat	6000.50c	100.0	11.907856	294.0	rmccs	ENDF/B-V.0
N-14	7014.50c	99.630	13.882780	294.0	rmccs	ENDF/B-V.0
O-16	8016.50c	99.760	15.857510	294.0	rmccs	ENDF/B-V.0
Al-27	13027.50c	100.0	26.749756	294.0	rmccs	ENDF/B-V.0
Si-nat	14000.50c	100.0	27.844241	294.0	endf5p	ENDF/B-V.0
P-31	15031.50c	100.0	30.707682	294.0	endf5u	ENDF/B-V.0
S-32	16032.50c	95.02	31.788939 [3]	294.0	endf5u	ENDF/B-V.0
Ti-nat	22000.50c	100.0	47.467124	294.0	endf5u	ENDF/B-V.0
Cr-50	24050.60c	4.345	49.516983	294.0	endf60	ENDF/B-VI.1
Cr-52	24052.60c	83.790	51.494313	294.0	endf60	ENDF/B-VI.1
Cr-53	24053.60c	9.500	52.485863	294.0	endf60	ENDF/B-VI.1
Cr-54	24054.60c	2.365	53.475519	294.0	endf60	ENDF/B-VI.1
Mn-55	25055.50c	100.0	54.466099	294.0	endf5u	ENDF/B-V.0
Fe-54	26054.60c	5.900	53.476242	294.0	endf60	ENDF/B-VI.1
Fe-56	26056.60c	91.720	55.454429	294.0	endf60	ENDF/B-VI.1
Fe-57	26057.60c	2.100	56.446290	294.0	endf60	ENDF/B-VI.1
Fe-58	26058.60c	0.280	57.435600	294.0	endf60	ENDF/B-VI.1
Co-59	27059.50c	100.0	58.426930	294.0	endf5u	ENDF/B-V.0
Ni-58	28058.60c	68.270	57.437652	294.0	endf60	ENDF/B-VI.1
Ni-60	28060.60c	26.100	59.415952	294.0	endf60	ENDF/B-VI.1
Ni-61	28061.60c	1.130	60.407628	294.0	endf60	ENDF/B-VI.1
Ni-62	28062.60c	3.590	61.396349	294.0	endf60	ENDF/B-VI.1
Ni-64	28064.60c	0.910	63.378793	294.0	endf60	ENDF/B-VI.1
Cu-63	29063.60c	69.170	62.389001	294.0	endf60	ENDF/B-VI.2

Table 5.3.1-1. MCNP Cross Section Libraries Used in the CRC Reactivity Calculations

Element / Isotope	MCNP ZAID	Atom % in Nature	Atomic Wt. Ratio [1]	Temp. (K)	Library Name	Data Source
Cu-65	29065.60c	30.830	64.370028	294.0	endf60	ENDF/B-VI.2
As-75	33075.35c	100.0	74.277979	0.0	rmccsa	ENDF/B-V.0
Kr-80	36080.50c	2.25	79.229851	294.0	rmccsa	ENDF/B-V.0
Kr-82	36082.50c	11.6	81.209803	294.0	rmccsa	ENDF/B-V.0
Kr-83	36083.50c	11.5	82.201858	294.0	rmccsa	ENDF/B-V.0
Kr-84	36084.50c	57.0	83.190662	294.0	rmccsa	ENDF/B-V.0
Kr-86	36086.50c	17.3	85.172596	294.0	rmccsa	ENDF/B-V.0
Y-89	39089.50c	100.0	88.142108	294.0	endf5u	ENDF/B-V.0
Zr-nat	40000.60c	100.0	90.439990	294.0	endf60	ENDF/B-VI.1
Zr-93	40093.50c	0.0	92.108361	294.0	kidman	ENDF/B-V.0
Nb-93	41093.50c	100.0	92.108263	294.0	endf5p	ENDF/B-V.0
Mo-nat	42000.50c	100.0	95.107188	294.0	endf5u	ENDF/B-V.0
Mo-95	42095.50c	15.92	94.090546	294.0	kidman	ENDF/B-V.0
Tc-99	43099.50c	0.0	98.056595	294.0	kidman	ENDF/B-V.0
Ru-101	44101.50c	17.1	100.038748	294.0	kidman	ENDF/B-V.0
Ru-103	44103.50c	0.0	102.022	294.0	kidman	ENDF/B-V.0
Rh-103	45103.50c	100.0	102.021490	294.0	rmccsa	ENDF/B-V.0
Rh-105	45105.50c	0.0	104.005	294.0	kidman	ENDF/B-V.0
Pd-105	46105.50c	22.33	104.003885	294.0	kidman	ENDF/B-V.0
Pd-108	46108.50c	26.46	106.976942	294.0	kidman	ENDF/B-V.0
Ag-107	47107.60c	51.839	105.986724	294.0	endf60	ENDF/B-VI.0
Ag-109	47109.60c	48.161	107.969204	294.0	endf60	ENDF/B-VI.0
Cd-nat	48000.50c	100.0	111.445880	294.0	endf5u	ENDF/B-V.0
In-nat	49000.60c	100.0	113.831536	294.0	endf60	ENDF/B-VI.0
Sn-nat	50000.35c	100.0	117.690428	0.0	endl85	LLNL
Xe-131	54131.50c	21.2	129.780532	294.0	kidman	ENDF/B-V.0
Xe-134	54134.35c	10.4	132.755077	0.0	endl85	LLNL
Xe-135	54135.53c	0.0	133.748208	587.0	eprixs	ENDF/B-V
Cs-133	55133.50c	100.0	131.763705	294.0	kidman	ENDF/B-V.0
Cs-135	55135.50c	0.0	133.746975	294.0	kidman	ENDF/B-V.0
Ba-138	56138.50c	71.70	136.720557	294.0	rmccs	ENDF/B-V.0

Table 5.3.1-1. MCNP Cross Section Libraries Used in the CRC Reactivity Calculations

Element / Isotope	MCNP ZAID	Atom % in Nature	Atomic Wt. Ratio [1]	Temp. (K)	Library Name	Data Source
Pr-141	59141.50c	100.0	139.697185	294.0	kidman	ENDF/B-V.0
Nd-143	60143.50c	12.18	141.682152	294.0	kidman	ENDF/B-V.0
Nd-145	60145.50c	8.30	143.667706	294.0	kidman	ENDF/B-V.0
Nd-147	60147.50c	0.0	145.654	294.0	kidman	ENDF/B-V.0
Nd-148	60148.50c	5.76	146.646216	294.0	kidman	ENDF/B-V.0
Pm-147	61147.50c	0.0	145.653	294.0	kidman	ENDF/B-V.0
Pm-148	61148.50c	0.0	146.647	294.0	kidman	ENDF/B-V.0
Pm-149	61149.50c	0.0	147.639	294.0	kidman	ENDF/B-V.0
Sm-147	62147.50c	15.0	145.652830	294.0	kidman	ENDF/B-V.0
Sm-149	62149.50c	13.8	147.637915	294.0	endf5u	ENDF/B-V.0
Sm-150	62150.50c	7.4	148.629416	294.0	kidman	ENDF/B-V.0
Sm-151	62151.50c	0.0	149.623	294.0	kidman	ENDF/B-V.0
Sm-152	62152.50c	26.7	150.614670	294.0	kidman	ENDF/B-V.0
Eu-151	63151.55c	47.8	149.623378	294.0	newxs	LANL/T-2
Eu-152	63152.50c	0.0	150.616668	294.0	endf5u	ENDF/B-V.0
Eu-153	63153.55c	52.2	151.607568	294.0	newxs	LANL/T-2
Eu-154	63154.50c	0.0	152.600719	294.0	endf5u	ENDF/B-V.0
Eu-155	63155.50c	0.0	153.592	294.0	kidman	ENDF/B-V.0
Gd-152	64152.50c	0.20	150.614731	294.0	endf5u	ENDF/B-V.0
Gd-154	64154.50c	2.18	152.598614	294.0	endf5u	ENDF/B-V.0
Gd-155	64155.50c	14.80	153.591761	294.0	endf5u	ENDF/B-V.0
Gd-156	64156.50c	20.47	154.582676	294.0	endf5u	ENDF/B-V.0
Gd-157	64157.50c	15.65	155.575907	294.0	endf5u	ENDF/B-V.0
Gd-158	64158.50c	24.84	156.567459	294.0	endf5u	ENDF/B-V.0
Gd-160	64160.50c	21.86	158.553203	294.0	endf5u	ENDF/B-V.0
Ho-165	67165.55c	100.0	163.513493	294.0	newxs	LANL/T-2
Ta-181	73181.50c	99.988	179.393575	294.0	endf5u	ENDF/B-V.0
Th-232	90232.50c	100.0	230.044724	294.0	endf5u	ENDF/B-V.0
Pa-233	91233.50c	0.0	231.038304	294.0	endf5u	ENDF/B-V.0
U-233	92233.50c	0.0	231.037695	294.0	rmccs	ENDF/B-V.0
U-234	92234.50c	0.0055	232.030412	294.0	endf5p	ENDF/B-V.0

Table 5.3.1-1. MCNP Cross Section Libraries Used in the CRC Reactivity Calculations

Element / Isotope	MCNP ZAID	Atom % in Nature	Atomic Wt. Ratio [1]	Temp. (K)	Library Name	Data Source
U-235	92235.53c	0.7200	233.024773	587.0	eprixs	ENDF/B-V.0
U-236	92236.50c	0.0	234.017806	294.0	endf5p	ENDF/B-V.0
U-237	92237.50c	0.0	235.012352	294.0	endf5p	ENDF/B-V.0
U-238	92238.53c	99.2745	236.005803	587.0	eprixs	ENDF/B-V.0
Np-235	93235.35c	0.0	233.024904	0.0	endl85	LLNL
Np-236	93236.35c	0.0	234.018854	0.0	endl85	LLNL
Np-237	93237.50c	0.0	235.011799	294.0	endf5p	ENDF/B-V.0
Np-238	93238.35c	0.0	236.005958	0.0	endl85	LLNL
Pu-237	94237.35c	0.0	235.012031	0.0	endl85	LLNL
Pu-238	94238.50c	0.0	236.004583	294.0	endf5p	ENDF/B-V.0
Pu-239	94239.55c	0.0	236.998573	294.0	rmccs	ENDF/B-V.2
Pu-240	94240.50c	0.0	237.991619	294.0	rmccs	ENDF/B-V.0
Pu-241	94241.50c	0.0	238.986041	294.0	endf5p	ENDF/B-V.0
Pu-242	94242.50c	0.0	239.979326	294.0	endf5p	ENDF/B-V.0
Am-241	95241.50c	0.0	238.986019	294.0	endf5u	ENDF/B-V.0
Am-242m	95242.50c	0.0	239.980121	294.0	endf5u	ENDF/B-V.0
Am-243	95243.50c	0.0	240.973348	294.0	endf5u	ENDF/B-V.0
Cm-242	96242.50c	0.0	239.979418	294.0	endf5u	ENDF/B-V.0
Cm-243	96243.35c	0.0	240.973356	0.0	endl85	LLNL
Cm-244	96244.50c	0.0	241.966119	294.0	endf5u	ENDF/B-V.0
Cm-245	96245.35c	0.0	242.960245	0.0	endl85	LLNL
Cm-246	96246.35c	0.0	243.953373	0.0	endl85	LLNL
Cm-247	96247.35c	0.0	244.947884	0.0	endl85	LLNL
Cm-248	96248.35c	0.0	245.941272	0.0	endl85	LLNL

[1] The atomic weight ratio presented for each isotope/element is the ratio of the isotope/element mass to the mass of a neutron. The mass of a neutron is 1.008664904 amu (p. 57, Ref. 7.5). The atomic weight ratio values are obtained from the "xsdir" file for MCNP as identified on page III-2 of Reference 7.4.

[2] The atom percent in nature of B-10 and B-11 varies significantly between different geographical regions of the world. The atom percents in nature that are listed in Table 5.3.1-1 for B-10 and B-11 were obtained from page 232 of Reference 7.6.

[3] The atomic weight ratio for natural sulfur is utilized in conjunction with the S-32 cross section library in the determination of the sulfur content in the various materials modeled in the MCNP calculations documented herein.

5.3.2. Reactor Materials

The tables presenting calculated material compositions in this section show excessive significant figures. The number of significant figures in the composition values are a result of the composition calculation and should not be interpreted as reflecting an excessively high level of accuracy.

The reactor components modeled in the MCNP CRC reactivity calculations include the following: core liner, core barrel, thermal shield, pressure vessel cladding, pressure vessel, borated moderator, upper reactor internals region, and lower reactor internals region. The material compositions are described in terms of elemental or isotopic weight percents with an overall material density.

The core liner, core barrel, thermal shield, and pressure vessel cladding are composed of Stainless Steel 304 (SS304) (p. 3, Ref. 7.11). The SS304 composition is shown in Table 5.3.2-1. The pressure vessel is composed of carbon steel (p. 3, Ref. 7.11). The carbon steel composition is shown in Table 5.3.2-2.

The borated moderator is composed of a homogeneous mixture of boron and water. The boron concentration in water is provided in terms of parts-per-million (ppm) by mass. Since the moderator in each CRC statepoint configuration has a different boron concentration and temperature, the overall borated moderator composition and density is different in each configuration.

The composition of the borated moderator and the borated moderator constituents in the homogenized spacer grid compositions as defined in the MCNP input decks are calculated by MACE. MACE uses linear interpolation in a steam table to obtain the borated moderator density value as described in Attachment I. Other materials in the MCNP input deck that contain borated moderator as a constituent are not calculated by MACE. These other material compositions are calculated in an EXCEL spreadsheet and are provided to MACE as input to be placed in the MCNP input decks. The density of the borated moderator that is used in the spreadsheet calculation of the material compositions is obtained using linear interpolation in the steam tables utilized by MACE. The MACE calculated moderator density and the spreadsheet moderator density are identical in value. Table 5.3.2-3 presents the borated moderator composition, temperature, and density for each CRC statepoint reactivity calculation. The borated moderator is used throughout the core configuration and between the various reactor components.

The following set of equations are used to calculate the borated moderator compositions shown in Table 5.3.2-3. The atomic weight ratio values for hydrogen, oxygen, boron-10, and boron-11 are obtained from Table 5.3.1-1. The atomic weight ratio for natural boron is 10.718156 (Ref. 7.12).

Equation 5.3.2-1. Boron Weight Percent in Borated Moderator

$$Boron\ wt\,\% = \frac{(Boron\ ppm)(1.0E-4)}{1+(Boron\ ppm)(1.0E-6)}$$

Equation 5.3.2-2. Boron-10 (B-10) Weight Percent in Borated Moderator

$$B-10\ wt\% = \frac{(B-10\ atom\%\ in\ B)(B-10\ Atomic\ Wt.\ Ratio)}{(B\ Atomic\ Wt.\ Ratio)(100.0)}(B\ wt\%)$$

where B is natural boron.

Equation 5.3.2-3. Boron-11 (B-11) Weight Percent in Borated Moderator

$$B-11\ wt\% = \frac{(B-11\ atom\%\ in\ B)(B-11\ Atomic\ Wt.\ Ratio)}{(B\ Atomic\ Wt.\ Ratio)(100.0)}(B\ wt\%)$$

Equation 5.3.2-4. Hydrogen Weight Percent in Borated Moderator

$$Hydrogen\ wt\% = \frac{(H\ Atomic\ Wt.\ Ratio)(2)(100.0-B\ wt\%)}{[(H\ Atomic\ Wt.\ Ratio)(2)+(O\ Atomic\ Wt.\ Ratio)]}$$

where H is hydrogen, B is natural boron, and O is oxygen.

Equation 5.3.2-5. Oxygen Weight Percent in Borated Moderator

$$Oxygen\ wt\% = \frac{(O\ Atomic\ Wt.\ Ratio)(100.0-B\ wt\%)}{[(H\ Atomic\ Wt.\ Ratio)(2)+(O\ Atomic\ Wt.\ Ratio)]}$$

where H is hydrogen, B is natural boron, and O is oxygen.

A large number of homogenized material compositions are provided to MACE as input. These homogenized material compositions are made up of various base components such as SS304, Inconel, Zircaloy-4, and borated moderator that are present in certain volume fractions. The homogenization of the base components into single homogenized material compositions is performed using Equations 5.3.2-6 through 5.3.2-8. Once the calculations in Equations 5.3.2-6 through 5.3.2-8 are performed, the homogenized material composition is provided as input to MACE in terms of the homogenized material composition density and base component material constituent weight percents.

Equation 5.3.2-6. Homogenized Material Density Calculation

$$Homogenized\ Material\ Density = \sum_{m}^{M}\left[(\rho)_m(Volume\ Fraction\ in\ Homogenized\ Material)_m\right]$$

where, m= a single base component material of the homogenized material, M= the total number of base component materials in the homogenized material, and ρ= the mass density of the base component material.

Equation 5.3.2-7. Calculation of Mass Fraction of Base Component Material in Homogenized Material

$$\begin{pmatrix} \textit{Mass Fraction of Base Component} \\ \textit{Material in Homogenized Material} \end{pmatrix} = \left[\frac{(\rho)_m (\textit{Volume Fraction in Homogenized Material})_m}{\textit{Homogenized Material Density}} \right]$$

Equation 5.3.2-8. Calculation of Weight Percent of Base Component Material Constituent in Homogenized Material

$$\begin{pmatrix} \textit{Weight Percent of Base} \\ \textit{Component Material} \\ \textit{Constituent in} \\ \textit{Homogenized Material} \end{pmatrix} = \begin{pmatrix} \textit{Mass Fraction of Base} \\ \textit{Component Material in} \\ \textit{Homogenized Material} \end{pmatrix} \begin{pmatrix} \textit{Weight Percent of Base} \\ \textit{Component Material Constituent} \\ \textit{in Base Component Material} \end{pmatrix}$$

The upper reactor internals region contains borated moderator and hardware composed of SS304 and Inconel (p. 9, Ref. 7.11). This region is modeled with a homogenized material composition in the MCNP CRC reactivity calculations. The upper reactor internals region is modeled as a number of rectangular sub-regions each placed directly above a fuel assembly. The material volume fractions in each of the rectangular upper reactor internals sub-regions depend on whether or not the fuel assembly below the sub-region is empty or has either a BPRA or RCCA inserted at the critical statepoint. Table 5.3.2-4 contains the material volume fractions for the upper reactor internals sub-region positioned above a fuel assembly containing no insertion assembly, a BPRA, and a RCCA. The SS304 material composition is presented in Table 5.3.2-1. The Inconel and borated moderator compositions are presented in Tables 5.3.2-5 and 5.3.2-3, respectively. The component material compositions are used in conjunction with their volume fractions in each of the upper reactor internals sub-regions to obtain a homogenized material composition and density that can be specified in the MCNP input decks. The calculated homogenized material compositions for the upper reactor internals sub-regions positioned above a fuel assembly containing no insertion assembly, a BPRA, and a RCCA are presented in Tables 5.3.2-6 through 5.3.2-8, respectively. The homogenized material compositions for each of the upper reactor internals sub-regions are different between CRC statepoints, as shown in Tables 5.3.2-6 through 5.3.2-8, due to the difference in moderator specifications between the statepoints.

The lower reactor internals region contains SS304 hardware and borated moderator. The volume fractions of SS304 and borated moderator in the lower reactor internals region is presented in Table 5.3.2-9. The SS304 and borated moderator compositions are presented in Tables 5.3.2-1 and 5.3.2-3, respectively. The calculated homogenized material compositions for the lower reactor internals region are presented in Table 5.3.2-10. The homogenized material composition for the lower reactor internals region is different between CRC statepoints, as shown in Table 5.3.2-10, due to the difference in moderator specifications between the statepoints.

The homogenizations of the upper and lower reactor internals regions are expected to have a minimal effect on the core reactivity due to their limited reactivity worth and proximity to the active fuel. The primary objective in modeling the upper and lower reactor internals regions is to obtain a reasonable approximation of the axial leakage from the reactor core.

Table 5.3.2-1. Type 304 Stainless Steel Composition (p. 12, Ref. 7.7)

Ele./Iso.	MCNP ZAID	Wt%	Ele./Iso.	MCNP ZAID	Wt%
C-nat	6000.50c	0.080	Fe-54	26054.60c	3.918
N-14	7014.50c	0.100	Fe-56	26056.60c	63.156
Si-nat	14000.50c	0.750	Fe-57	26057.60c	1.472
P-31	15031.50c	0.045	Fe-58	26058.60c	0.200
S-nat	16032.50c	0.030	Ni-58	28058.60c	6.234
Cr-50	24050.60c	0.793	Ni-60	28060.60c	2.465
Cr-52	24052.60c	15.903	Ni-61	28061.60c	0.109
Cr-53	24053.60c	1.838	Ni-62	28062.60c	0.350
Cr-54	24054.60c	0.466	Ni-64	28064.60c	0.092
Mn-55	25055.50c	2.000	Density = 7.9 g/cm^3		

Table 5.3.2-2. Grade 55 A 516 Carbon Steel Composition (p. 5, Ref. 7.7)[1]

Ele./Iso.	MCNP ZAID	Wt%	Ele./Iso.	MCNP ZAID	Wt%
C-nat	6000.50c	0.220	Fe-54	26054.60c	5.615
Si-nat	14000.50c	0.275	Fe-56	26056.60c	90.524
P-31	15031.50c	0.035	Fe-57	26057.60c	2.110
S-nat	16032.50c	0.035	Fe-58	26058.60c	0.286
Mn-55	25055.50c	0.900	Density = 7.832 g/cm^3		

[1] The pressure vessel was actually made of CS508 carbon steel (p. 3, Ref. 7.11). Grade 55 A 516 was substituted for CS508. The pressure vessel has no neutronic importance with respect to the k_{eff} of the reactor core. Therefore, this substitution is acceptable.

Table 5.3.2-3. Borated Moderator Composition for Each CRC Statepoint Calculation

Cycle / EFPD	Temp. (F)	Boron (ppm)	Density (g/cm^3)	H wt%	O wt%	B-10 wt%	B-11 wt%
1 / 0.0	547.0	1296	0.75395	11.17700	88.69357	0.02325	0.10619
3 / 0.0	549.4	1661	0.75155	11.17293	88.66126	0.02978	0.13604
3 / 210.9	545.0	922	0.75595	11.18118	88.72671	0.01654	0.07557

Table 5.3.2-4. Upper Reactor Internals Sub-Region Material Volume Fractions

Insertion Assembly	Material Volume Fractions		
	SS304	Inconel	Borated Water
None (p. 9, Ref. 7.11)	0.1770	---	0.8230
BPRA[1]	0.1907	0.0035	0.8058
RCCA (p. 13, Ref. 7.11)	0.1907	0.0035	0.8058

[1] The material volume fractions for the reactor internals sub-region above a fuel assembly containing a BPRA were not provided in the reference document (Ref. 7.11), so the material volume fractions for the reactor internals sub-region above a fuel assembly containing a RCCA were used. This is acceptable given the insignificant reactivity worth of this region.

Table 5.3.2-5. Inconel 718 Composition (Ref. 7.8)

Ele./Iso.	MCNP ZAID	Wt%	Ele./Iso.	MCNP ZAID	Wt%
C-nat	6000.50c	0.080	Ni-60	28060.60c	13.993
Si-nat	14000.50c	0.350	Ni-61	28061.60c	0.616
P-31	15031.50c	0.015	Ni-62	28062.60c	1.989
S-nat	16032.50c	0.015	Ni-64	28064.60c	0.520
Cr-50	24050.60c	0.793	B-10	5010.50c	1.078E-03
Cr-52	24052.60c	15.903	B-11	5011.56c	4.925E-03
Cr-53	24053.60c	1.838	Ti-nat	22000.50c	0.900
Cr-54	24054.60c	0.466	Al-27	13027.50c	0.500
Mn-55	25055.50c	0.350	Co-59	27059.50c	1.000
Fe-54	26054.60c	0.958	Cu-63	29063.60c	0.205
Fe-56	26056.60c	15.442	Cu-65	29065.60c	0.095
Fe-57	26057.60c	0.360	Nb-93	41093.50c	2.563
Fe-58	26058.60c	0.049	Mo-nat	42000.50c	3.050
Ni-58	28058.60c	35.382	Ta-181	73181.50c	2.563
Density = 8.19 g/cm^3					

Table 5.3.2-6. Homogenized Composition for Upper Reactor Internals Sub-Region Above a Fuel Assembly Containing No Insertion Assembly

MCNP ZAID	Wt% of Element/ Isotope in Material Composition		
	Cycle 1 0.0 EFPD	Cycle 3 0.0 EFPD	Cycle 3 210.9 EFPD
6000.50c	0.055	0.055	0.055
7014.50c	0.069	0.069	0.069
14000.50c	0.519	0.520	0.519
15031.50c	0.031	0.031	0.031
16032.50c	0.021	0.021	0.021
24050.60c	0.549	0.550	0.549
24052.60c	11.015	11.026	11.006
24053.60c	1.273	1.274	1.272
24054.60c	0.323	0.323	0.323
25055.50c	1.385	1.387	1.384
26054.60c	2.713	2.716	2.711
26056.60c	43.744	43.787	43.709

Table 5.3.2-6. Homogenized Composition for Upper Reactor Internals Sub-Region Above a Fuel Assembly Containing No Insertion Assembly

MCNP ZAID	Wt% of Element/ Isotope in Material Composition		
	Cycle 1 0.0 EFPD	Cycle 3 0.0 EFPD	Cycle 3 210.9 EFPD
26057.60c	1.019	1.020	1.019
26058.60c	0.138	0.138	0.138
28058.60c	4.318	4.322	4.314
28060.60c	1.708	1.709	1.706
28061.60c	0.075	0.075	0.075
28062.60c	0.243	0.243	0.243
28064.60c	0.064	0.064	0.063
1001.50c	3.435	3.427	3.443
5010.50c	7.157E-03	9.153E-03	5.101E-03
5011.56c	3.270E-02	4.181E-02	2.330E-02
8016.50c	27.261	27.191	27.321
Density (g/cm^3)	2.0188	2.0168	2.0204

Table 5.3.2-7: Homogenized Composition for Upper Reactor Internals Sub-Region Above a Fuel Assembly Containing a BPRA

MCNP ZAID	Wt% of Element/ Isotope in Material Composition		
	Cycle 1 0.0 EFPD	Cycle 3 0.0 EFPD	Cycle 3 210.9 EFPD
6000.50c	0.057	0.057	0.057
7014.50c	0.070	0.070	0.070
14000.50c	0.532	0.532	0.532
15031.50c	0.032	0.032	0.032
16032.50c	0.021	0.021	0.021
24050.60c	0.568	0.569	0.568
24052.60c	11.394	11.404	11.386
24053.60c	1.317	1.318	1.316
24054.60c	0.334	0.334	0.334
25055.50c	1.411	1.412	1.410
26054.60c	2.767	2.770	2.765
26056.60c	44.611	44.651	44.577
26057.60c	1.040	1.041	1.039
26058.60c	0.141	0.141	0.141
28058.60c	4.856	4.861	4.853
28060.60c	1.921	1.922	1.919

Table 5.3.2-7. Homogenized Composition for Upper Reactor Internals Sub-Region Above a Fuel Assembly Containing a BPRA

MCNP ZAID	Wt% of Element/ Isotope in Material Composition		
	Cycle 1 0.0 EFPD	Cycle 3 0.0 EFPD	Cycle 3 210.9 EFPD
28061.60c	0.085	0.085	0.084
28062.60c	0.273	0.273	0.273
28064.60c	0.071	0.071	0.071
1001.50c	3.169	3.161	3.176
5010.50c	6.617E-03	8.457E-03	4.720E-03
5011.56c	3.023E-02	3.863E-02	2.156E-02
8016.50c	25.147	25.081	25.205
13027.50c	0.007	0.007	0.007
22000.50c	0.012	0.012	0.012
27059.50c	0.013	0.013	0.013
29063.60c	0.003	0.003	0.003
29065.60c	0.001	0.001	0.001
41093.50c	0.034	0.034	0.034
42000.50c	0.041	0.041	0.041
73181.50c	0.034	0.034	0.034
Density (g/cm^3)	2.1427	2:1408	2.1443

Table 5.3.2-8. Homogenized Composition for Upper Reactor Internals Sub-Region Above a Fuel Assembly Containing a RCCA

MCNP ZAID	Wt% of Element/ Isotope in Material Composition		
	Cycle 1 0.0 EFPD	Cycle 3 0.0 EFPD	Cycle 3 210.9 EFPD
6000.50c	0.057	0.057	0.057
7014.50c	0.070	0.070	0.070
14000.50c	0.532	0.532	0.532
15031.50c	0.032	0.032	0.032
16032.50c	0.021	0.021	0.021
24050.60c	0.568	0.569	0.568
24052.60c	11.394	11.404	11.386
24053.60c	1.317	1.318	1.316
24054.60c	0.334	0.334	0.334
25055.50c	1.411	1.412	1.410
26054.60c	2.767	2.770	2.765
26056.60c	44.611	44.651	44.577

Table 5.3.2-8. Homogenized Composition for Upper Reactor Internals Sub-Region Above a Fuel Assembly Containing a RCCA

MCNP ZAID	Wt% of Element/ Isotope in Material Composition		
	Cycle 1 0.0 EFPD	Cycle 3 0.0 EFPD	Cycle 3 210.9 EFPD
26057.60c	1.040	1.041	1.039
26058.60c	0.141	0.141	0.141
28058.60c	4.856	4.861	4.853
28060.60c	1.921	1.922	1.919
28061.60c	0.085	0.085	0.084
28062.60c	0.273	0.273	0.273
28064.60c	0.071	0.071	0.071
1001.50c	3.169	3.161	3.176
5010.50c	6.617E-03	8.457E-03	4.720E-03
5011.56c	3.023E-02	3.863E-02	2.156E-02
8016.50c	25.147	25.081	25.205
13027.50c	0.007	0.007	0.007
22000.50c	0.012	0.012	0.012
27059.50c	0.013	0.013	0.013
29063.60c	0.003	0.003	0.003
29065.60c	0.001	0.001	0.001
41093.50c	0.034	0.034	0.034
42000.50c	0.041	0.041	0.041
73181.50c	0.034	0.034	0.034
Density (g/cm^3)	2.1427	2.1408	2.1443

Table 5.3.2-9. Lower Reactor Internals Sub-Region Material Volume Fractions (p. 9, Ref. 7.11)

SS304	Borated Water
0.1720	0.8280

Table 5.3.2-10. Homogenized Composition for Lower Reactor Internals Region

MCNP ZAID	Wt% of Element/ Isotope in Material Composition		
	Cycle 1 0.0 EFPD	Cycle 3 0.0 EFPD	Cycle 3 210.9 EFPD
6000.50c	0.055	0.055	0.055
7014.50c	0.069	0.069	0.068
14000.50c	0.514	0.514	0.513
15031.50c	0.031	0.031	0.031
16032.50c	0.021	0.021	0.021
24050.60c	0.543	0.544	0.543
24052.60c	10.897	10.908	10.888
24053.60c	1.259	1.261	1.258
24054.60c	0.319	0.320	0.319
25055.50c	1.370	1.372	1.369
26054.60c	2.684	2.687	2.682
26056.60c	43.274	43.318	43.238
26057.60c	1.008	1.010	1.008
26058.60c	0.137	0.137	0.137
28058.60c	4.272	4.276	4.268
28060.60c	1.689	1.691	1.688
28061.60c	0.074	0.074	0.074
28062.60c	0.240	0.240	0.240
28064.60c	0.063	0.063	0.063
1001.50c	3.519	3.510	3.526
5010.50c	7.331E-03	9.375E-03	5.225E-03
5011.56c	3.349E-02	4.282E-02	2.387E-02
8016.50c	27.921	27.850	27.982
Density (g/cm^3)	1.9831	1.9811	1.9847

5.3.3. Fuel Assembly Materials

The fuel assembly materials listed in this section refer to the upper and lower end-fitting materials and the spacer grid materials. The upper end-fitting material compositions vary within a given fuel assembly design depending upon whether the assembly contains no insertion assembly, a BPRA, or a RCCA at the critical statepoint. Both the upper and lower end-fitting homogenized material compositions vary between critical statepoint configurations due to the different moderator conditions. The primary material components in the upper and lower end-fitting regions are SS304, Inconel, and borated moderator. Both the upper and lower end-fitting regions are modeled with material compositions that represent the homogenization of all of the components in the regions. Table 5.3.2-1 presents the material composition of SS304. Table 5.3.2-3 presents the material compositions for the borated moderator in CRC statepoint configuration. Table 5.3.2-5 presents the material composition of Inconel. Table 5.3.3-1 presents the component material volume fractions for the upper end-fitting region for the STD assembly design. Table 5.3.3-2 presents the component material volume fractions for the lower end-fitting region for the STD assembly design. Table 5.3.3-3 presents the upper end-fitting homogenized material compositions for each CRC statepoint configuration for the STD assembly design with no insertion assembly. Table 5.3.3-4 presents the upper end-fitting homogenized material compositions for each CRC statepoint configuration for the STD assembly design with a BPRA inserted. Table 5.3.3-5 presents the upper end-fitting homogenized material compositions for each CRC statepoint configuration for the STD assembly design with a RCCA inserted. Table 5.3.3-6 presents the lower end-fitting homogenized material compositions for each CRC statepoint configuration for the STD assembly design. The homogenized material compositions presented in this section were calculated using the method described in Section 5.3.2.

The upper end spacer grid region is composed of SS304, Inconel, and borated moderator (pp. 9, 13, 16, Ref. 7.11). The upper end spacer grid region is located directly below the upper end-fitting, and covers a height of 14.656 cm along the length of the fuel assembly (p. 8, Ref. 7.11). The materials of the upper end spacer grid are homogenized and modeled in the region between the fuel rods, guide tubes, and instrument tube. Table 5.3.3-7 presents the referenced upper spacer grid material volume fractions for the STD fuel assembly design. The homogenized material composition for each upper spacer grid will be different between the CRC statepoint configurations due to the different moderator conditions.

The six spacer grids below the upper end spacer grid are called the intermediate spacer grids. These intermediate spacer grids are composed of Inconel in the STD design (p. 6, Ref. 7.11). The intermediate spacer grid height is 3.35788 cm for the STD assembly design (p. 6, Ref. 7.11). The individual intermediate spacer grid volume is 95.234 cm^3 for the STD design (p. 6, Ref. 7.11). The volume between the fuel rods, guide tubes, and instrument tube that is occupied by an explicit intermediate spacer grid and borated moderator is 825.579 cm^3 for the STD design (p. 6, Ref. 7.11). Table 5.3.3-9 shows the referenced volume fractions of intermediate spacer grid material and borated moderator in the region between the fuel rods, guide tubes, and instrument tube for the STD assembly design. The intermediate spacer grid materials and borated moderator are homogenized and modeled in the region between the fuel rods, guide tubes, and instrument tube over the explicit height of each spacer grid. The homogenized material composition for the intermediate spacer grid of each fuel assembly will vary between the CRC statepoint configurations due to the different moderator conditions.

The lower end spacer grid is composed of Inconel in the STD assembly design (p. 6, Ref. 7.11). The lower end spacer grid height is 3.35788 cm for the STD design (p. 6, Ref. 7.11). The lower end spacer grid volume is 95.234 cm^3 for the STD design (p. 6, Ref. 7.11). The volume between the fuel rods, guide tubes, and instrument tube that is occupied by an explicit lower end spacer grid and borated moderator is 845.296 cm^3 for the STD design (p. 6, Ref. 7.11). Table 5.3.3-11 shows the referenced volume fractions of lower end spacer grid material and borated moderator in the region between the fuel rods, guide tubes, and instrument tube for the STD assembly design. The lower end spacer grid materials and borated moderator are homogenized and modeled in the region between the fuel rods, guide tubes, and instrument tube over the explicit height of each spacer grid. The homogenized material composition for the lower end spacer grid of each fuel assembly will vary between the CRC statepoint configurations due to the different moderator conditions.

The volume fractions of the component materials in the homogenized spacer grid compositions that were specified in the MCNP input decks are shown in Tables 5.3.3-7, 5.3.3-9, and 5.3.3-11, for the upper, intermediate, and lower spacer grids, respectively. Table 5.3.3-8 presents the modeled homogenized material compositions for the upper end spacer grid of the STD design for each CRC statepoint configuration. Table 5.3.3-10 presents the modeled homogenized material compositions for the intermediate spacer grid of the STD design for each CRC statepoint configuration. Table 5.3.3-12 presents the modeled homogenized material compositions for the lower end spacer grid of the STD design for each CRC statepoint configuration.

Table 5.3.3-1. Upper End-Fitting Component Material Volume Fractions

Assembly Design [Insertion Specification]	Volume Fractions in Upper End-Fitting Region		
	SS304	Inconel	Borated Moderator
STD [No Insertion Assembly] (p. 9, Ref. 7.11)	0.1243	0.0168	0.8589
STD [BPRA Inserted] (p. 16, Ref. 7.11)	0.1649	0.0228	0.8123
STD [RCCA Inserted] (p. 13, Ref. 7.11)	0.1444	0.0218	0.8338

Table 5.3.3-2. Lower End-Fitting Component Material Volume Fractions

Assembly Design	Volume Fractions in Lower End-Fitting Region	
	SS304	Borated Moderator
STD (p. 9, Ref. 7.11)	0.1625	0.8375

Table 5.3.3-3. Upper End-Fitting Homogenized Material Composition for the STD Assembly Design with No Insertion Assembly

MCNP ZAID	Wt% of Element/Isotope in Material Composition		
	Cycle 1 0.0 EFPD	Cycle 3 0.0 EFPD	Cycle 3 210.9 EFPD
6000.50c	0.051	0.051	0.051
7014.50c	0.056	0.056	0.056
14000.50c	0.444	0.445	0.444
15031.50c	0.026	0.026	0.026
16032.50c	0.018	0.018	0.018
24050.60c	0.502	0.503	0.502
24052.60c	10.075	10.087	10.066
24053.60c	1.164	1.166	1.163

Table 5.3.3-3. Upper End-Fitting Homogenized Material Composition for the STD Assembly Design with No Insertion Assembly

MCNP ZAID	Wt% of Element/Isotope in Material Composition		
	Cycle 1 0.0 EFPD	Cycle 3 0.0 EFPD	Cycle 3 210.9 EFPD
24054.60c	0.295	0.296	0.295
25055.50c	1.139	1.140	1.138
26054.60c	2.252	2.254	2.249
26056.60c	36.297	36.340	36.262
26057.60c	0.846	0.847	0.845
26058.60c	0.115	0.115	0.115
28058.60c	6.219	6.226	6.213
28060.60c	2.459	2.462	2.457
28061.60c	0.108	0.108	0.108
28062.60c	0.350	0.350	0.349
28064.60c	0.091	0.092	0.091
1001.50c	4.096	4.086	4.104
5010.50c	8.617E-03	1.100E-02	6.165E-03
5011.56c	3.936E-02	5.024E-02	2.816E-02
8016.50c	32.502	32.424	32.569
13027.50c	0.039	0.039	0.039
22000.50c	0.070	0.070	0.070
27059.50c	0.078	0.078	0.078
29063.60c	0.016	0.016	0.016
29065.60c	0.007	0.007	0.007
41093.50c	0.200	0.200	0.199
42000.50c	0.237	0.238	0.237
73181.50c	0.200	0.200	0.199
Density (g/cm^3)	1.7671	1.7651	1.7688

Table 5.3.3-4. Upper End-Fitting Homogenized Material Composition for the STD Assembly Design with a BPRA Inserted

MCNP ZAID	Wt% of Element/Isotope in Material Composition		
	Cycle 1 0.0 EFPD	Cycle 3 0.0 EFPD	Cycle 3 210.9 EFPD
6000.50c	0.057	0.057	0.057
7014.50c	0.062	0.062	0.062
14000.50c	0.496	0.496	0.496
15031.50c	0.029	0.029	0.029
16032.50c	0.020	0.020	0.020

Table 5.3.3-4. Upper End-Fitting Homogenized Material Composition for the STD Assembly Design with a BPRA Inserted

MCNP ZAID	Wt% of Element/Isotope in Material Composition		
	Cycle 1 0.0 EFPD	Cycle 3 0.0 EFPD	Cycle 3 210.9 EFPD
24050.60c	0.562	0.562	0.562
24052.60c	11.269	11.280	11.261
24053.60c	1.302	1.304	1.301
24054.60c	0.330	0.331	0.330
25055.50c	1.271	1.272	1.270
26054.60c	2.513	2.515	2.511
26056.60c	40.515	40.553	40.484
26057.60c	0.944	0.945	0.943
26058.60c	0.128	0.128	0.128
28058.60c	7.007	7.014	7.002
28060.60c	2.771	2.774	2.769
28061.60c	0.122	0.122	0.122
28062.60c	0.394	0.394	0.394
28064.60c	0.103	0.103	0.103
1001.50c	3.257	3.248	3.264
5010.50c	6.881E-03	8.772E-03	4.932E-03
5011.56c	3.143E-02	4.007E-02	2.253E-02
8016.50c	25.843	25.775	25.901
13027.50c	0.044	0.044	0.044
22000.50c	0.080	0.080	0.080
27059.50c	0.089	0.089	0.089
29063.60c	0.018	0.018	0.018
29065.60c	0.008	0.008	0.008
41093.50c	0.228	0.228	0.227
42000.50c	0.271	0.271	0.271
73181.50c	0.228	0.228	0.227
Density (g/cm^3)	2.1019	2.0999	2.1035

Table 5.3.3-5. Upper End-Fitting Homogenized Material Composition for the STD Assembly Design with a RCCA Inserted

MCNP ZAID	Wt% of Element/Isotope in Material Composition		
	Cycle 1 0.0 EFPD	Cycle 3 0.0 EFPD	Cycle 3 210.9 EFPD
6000.50c	0.054	0.054	0.054
7014.50c	0.059	0.059	0.059

Table 5.3.3-5. Upper End-Fitting Homogenized Material Composition for the STD Assembly Design with a RCCA Inserted

MCNP ZAID	Wt% of Element/Isotope in Material Composition		
	Cycle 1 0.0 EFPD	Cycle 3 0.0 EFPD	Cycle 3 210.9 EFPD
14000.50c	0.471	0.472	0.471
15031.50c	0.028	0.028	0.028
16032.50c	0.019	0.019	0.019
24050.60c	0.537	0.538	0.537
24052.60c	10.771	10.782	10.762
24053.60c	1.245	1.246	1.244
24054.60c	0.316	0.316	0.315
25055.50c	1.203	1.205	1.202
26054.60c	2.382	2.384	2.380
26056.60c	38.401	38.440	38.368
26057.60c	0.895	0.896	0.894
26058.60c	0.121	0.122	0.121
28058.60c	6.894	6.901	6.888
28060.60c	2.726	2.729	2.724
28061.60c	0.120	0.120	0.120
28062.60c	0.387	0.388	0.387
28064.60c	0.101	0.101	0.101
1001.50c	3.607	3.598	3.615
5010.50c	7.614E-03	9.710E-03	5.455E-03
5011.56c	3.478E-02	4.435E-02	2.492E-02
8016.50c	28.623	28.551	28.685
13027.50c	0.046	0.046	0.046
22000.50c	0.082	0.083	0.082
27059.50c	0.092	0.092	0.092
29063.60c	0.019	0.019	0.019
29065.60c	0.009	0.009	0.009
41093.50c	0.235	0.235	0.235
42000.50c	0.280	0.280	0.279
73181.50c	0.235	0.235	0.235
Density (g/cm^3)	1.9479	1.9459	1.9496

Table 5.3.3-6. Lower End-Fitting Homogenized Material Composition for the STD Assembly Design

MCNP ZAID	Wt% of Element/Isotope in Material Composition		
	Cycle 1 0.0 EFPD	Cycle 3 0.0 EFPD	Cycle 3 210.9 EFPD
6000.50c	0.054	0.054	0.054
7014.50c	0.067	0.067	0.067
14000.50c	0.503	0.503	0.502
15031.50c	0.030	0.030	0.030
16032.50c	0.020	0.020	0.020
24050.60c	0.532	0.532	0.531
24052.60c	10.660	10.671	10.651
24053.60c	1.232	1.233	1.231
24054.60c	0.312	0.313	0.312
25055.50c	1.341	1.342	1.339
26054.60c	2.626	2.629	2.624
26056.60c	42.334	42.378	42.297
26057.60c	0.987	0.988	0.986
26058.60c	0.134	0.134	0.134
28058.60c	4.179	4.183	4.175
28060.60c	1.653	1.654	1.651
28061.60c	0.073	0.073	0.073
28062.60c	0.235	0.235	0.235
28064.60c	0.061	0.062	0.061
1001.50c	3.685	3.676	3.693
5010.50c	7.677E-03	9.819E-03	5.472E-03
5011.56c	3.507E-02	4.485E-02	2.499E-02
8016.50c	29.242	29.169	29.305
Density (g/cm^3)	1.9152	1.9132	1.9169

Table 5.3.3-7. Upper Spacer Grid Material Volume Fractions

Assembly Design	Volume Fraction in Upper End Spacer Grid Region		
	SS304	Inconel	Borated Moderator
STD (p. 9, Ref. 7.11)	0.0031	0.0264	0.9705

Table 5.3.3-8. Upper Spacer Grid Homogenized Material Composition for the STD Assembly Design

MCNP ZAID	Wt% of Element/Isotope in Material Composition		
	Cycle 1 0.0 EFPD	Cycle 3 0.0 EFPD	Cycle 3 210.9 EFPD
1001.50c	8.410318	8.400615	8.418975
8016.50c	66.738945	66.661934	66.807632
5010.50c	0.017731	0.022632	0.012696
5011.56c	0.080996	0.103384	0.057995
6000.50c	0.019803	0.019850	0.019763
7014.50c	0.002519	0.002525	0.002514
14000.50c	0.096712	0.096944	0.096519
15031.50c	0.004469	0.004479	0.004460
16032.50c	0.004091	0.004101	0.004083
24050.60c	0.196294	0.196765	0.195903
24052.60c	3.936524	3.945976	3.928683
24053.60c	0.454966	0.456059	0.454060
24054.60c	0.115351	0.115628	0.115121
25055.50c	0.128197	0.128505	0.127942
26054.60c	0.311694	0.312443	0.311073
26056.60c	5.024261	5.036324	5.014252
26057.60c	0.117104	0.117385	0.116871
26058.60c	0.015911	0.015949	0.015879
28058.60c	8.024076	8.043342	8.008092
28060.60c	3.172818	3.180436	3.166498
28061.60c	0.140299	0.140636	0.140020
28062.60c	0.450502	0.451583	0.449604
28064.60c	0.118417	0.118702	0.118182
13027.50c	0.111173	0.111440	0.110951
22000.50c	0.200111	0.200591	0.199712
27059.50c	0.222345	0.222879	0.221902
29063.60c	0.045559	0.045668	0.045468
29065.60c	0.021145	0.021196	0.021103
41093.50c	0.569759	0.571127	0.568624
42000.50c	0.678153	0.679781	0.676802
73181.50c	0.569759	0.571127	0.568624
Density (g/cm^3)	0.972413	0.970084	0.974354

Table 5.3.3-9. Volume Fractions for Intermediate Spacer Grid Homogenization

Assembly Design	Volume Fraction of Material in Homogenized Intermediate Spacer Grid	
	Inconel	Borated Moderator
STD	0.115	0.885

Table 5.3.3-10. Intermediate Spacer Grid Homogenized Material Composition for the STD Assembly Design

MCNP ZAID	Wt% of Element/Isotope in Material Composition		
	Cycle 1 0.0 EFPD	Cycle 3 0.0 EFPD	Cycle 3 210.9 EFPD
1001.50c	4.625369	4.615045	4.634285
8016.50c	36.703991	36.622063	36.774734
5010.50c	0.010251	0.012934	0.007488
5011.56c	0.046829	0.059083	0.034205
6000.50c	0.046894	0.046956	0.046842
14000.50c	0.205160	0.205430	0.204935
15031.50c	0.008793	0.008804	0.008783
16032.50c	0.008793	0.008804	0.008783
24050.60c	0.464833	0.465447	0.464324
24052.60c	9.321876	9.334171	9.311653
24053.60c	1.077382	1.078803	1.076201
24054.60c	0.273156	0.273516	0.272856
25055.50c	0.205160	0.205430	0.204935
26054.60c	0.561551	0.562292	0.560936
26056.60c	9.051754	9.063693	9.041827
26057.60c	0.210976	0.211254	0.210745
26058.60c	0.028665	0.028703	0.028634
28058.60c	20.739992	20.767349	20.717251
28060.60c	8.200848	8.211664	8.191855
28061.60c	0.362634	0.363112	0.362236
28062.60c	1.164421	1.165957	1.163144
28064.60c	0.306076	0.306480	0.305740
13027.50c	0.293085	0.293472	0.292764
22000.50c	0.527554	0.528250	0.526975
27059.50c	0.586171	0.586944	0.585528
29063.60c	0.120106	0.120265	0.119975
29065.60c	0.055745	0.055818	0.055684
41093.50c	1.502063	1.504044	1.500416
42000.50c	1.787821	1.790179	1.785861

Table 5.3.3-10. Intermediate Spacer Grid Homogenized Material Composition for the STD Assembly Design

MCNP ZAID	Wt% of Element/Isotope in Material Composition		
	Cycle 1 0.0 EFPD	Cycle 3 0.0 EFPD	Cycle 3 210.9 EFPD
73181.50c	1.502063	1.504044	1.500416
Density (g/cm^3)	1.611726	1.609603	1.613495

Table 5.3.3-11. Volume Fractions for Lower End Spacer Grid Homogenization

Assembly Design	Volume Fraction of Material in Homogenized Intermediate Spacer Grid	
	Inconel	Borated Moderator
STD	0.113	0.887

Table 5.3.3-12. Lower End Spacer Grid Homogenized Material Composition for the STD Assembly Design

MCNP ZAID	Wt% of Element/Isotope in Material Composition		
	Cycle 1 0.0 EFPD	Cycle 3 0.0 EFPD	Cycle 3 210.9 EFPD
1001.50c	4.697753	4.687364	4.706727
8016.50c	37.278385	37.195942	37.349594
5010.50c	0.010395	0.013120	0.007588
5011.56c	0.047484	0.059931	0.034662
6000.50c	0.046376	0.046438	0.046324
14000.50c	0.202893	0.203165	0.202667
15031.50c	0.008695	0.008707	0.008686
16032.50c	0.008695	0.008707	0.008686
24050.60c	0.459698	0.460314	0.459186
24052.60c	9.218885	9.231236	9.208617
24053.60c	1.065479	1.066906	1.064292
24054.60c	0.270138	0.270500	0.269837
25055.50c	0.202893	0.203165	0.202667
26054.60c	0.555347	0.556091	0.554729
26056.60c	8.951747	8.963741	8.941777
26057.60c	0.208645	0.208925	0.208413
26058.60c	0.028348	0.028386	0.028317
28058.60c	20.510849	20.538332	20.488008
28060.60c	8.110241	8.121108	8.101210
28061.60c	0.358627	0.359108	0.358228
28062.60c	1.151556	1.153099	1.150273

Table 5.3.3-12. Lower End Spacer Grid Homogenized Material Composition for the STD Assembly Design

MCNP ZAID	Wt% of Element/Isotope in Material Composition		
	Cycle 1 0.0 EFPD	Cycle 3 0.0 EFPD	Cycle 3 210.9 EFPD
28064.60c	0.302694	0.303100	0.302357
13027.50c	0.289847	0.290236	0.289525
22000.50c	0.521725	0.522424	0.521144
27059.50c	0.579695	0.580471	0.579049
29063.60c	0.118779	0.118939	0.118647
29065.60c	0.055129	0.055203	0.055068
41093.50c	1.485467	1.487458	1.483813
42000.50c	1.768069	1.770438	1.766100
73181.50c	1.485467	1.487458	1.483813
Density (g/cm^3)	1.591718	1.589588	1.593493

5.3.4. Fuel Rod Materials

The fuel rod components include the fuel rod cladding, the upper and lower fuel rod plenums (including end-caps), and the fuel. The fuel rod cladding is modeled as Zircaloy-4 as presented in Table 5.3.4-2. The upper and lower fuel rod plenum regions contain SS304 springs. The Zircaloy-4 end-caps are also homogenized in the upper and lower fuel rod plenum. Fission gases present in the upper and lower fuel rod plenum region are modeled as void in the homogenization. Table 5.3.4-1 contains the component material volume fractions for the fuel rod plenum regions (with end-caps included). Since the cladding is modeled explicitly, the values in Table 5.3.4-1 were calculated by renormalizing the volume fractions of the SS304, Zircaloy-4, and gas to the sum of the reference volume fractions (p. 9, Ref. 7.11) for the SS304, Zircaloy-4, and gas. Table 5.3.4-2 contains the Zircaloy-4 material compositions. Table 5.3.4-3 contains the homogenized material compositions for the upper and lower fuel rod plenum regions. The helium-filled gap between the fuel rod cladding and the fuel is modeled as void. The fresh fuel composition is uniform along the axial length of the fuel rod. The weight percent (wt%) enrichment of U-235 in the uranium of the fabricated UO_2 is presented in Table 5.3.4-4 for each fuel batch. The mass loading of uranium in the entire fuel assembly is also presented in Table 5.3.4-4. The compositions of the fresh fuel are presented in Table 5.3.4-5. The isotopic weight percentages in the fresh fuel composition are calculated using the following equations.

Equation 5.3.4-1. Uranium Isotope Weight Percents in Fabricated UO_2 (p. 20, Ref. 7.10)

$$U^{234}\ wt\% = (0.007731) * (U^{235}\ wt\%)^{1.0837}$$

$$U^{236}\ wt\% = (0.0046) * (U^{235}\ wt\%)$$

$$U^{238}\ wt\% = 100 - U^{234}\ wt\% - U^{235}\ wt\% - U^{236}\ wt\%$$

Equation 5.3.4-2. Uranium Mass per mol of UO_2

$$\frac{U\ Mass}{mol\ UO_2} = (1.008664904)\begin{bmatrix}(232.030)(U^{234}wt\%) + (233.025)(U^{235}wt\%) + \\ (234.018)(U^{236}wt\%) + (236.006)(U^{238}wt\%)\end{bmatrix}(0.01)$$

where the weight percentages of the uranium isotopes (U^{234}, U^{235}, U^{236}, and U^{238}) in uranium are calculated using Equation 5.3.4-1.

Equation 5.3.4-3. Oxygen Mass per mol of UO_2

$$\frac{O\ Mass}{mol\ UO_2} = (2)(1.008664904)(15.858)$$

Equation 5.3.4-4. Oxygen Mass in UO_2

$$O\ Mass\ in\ UO_2 = \left(\frac{O\ Mass / mol\ UO_2}{U\ Mass / mol\ UO_2}\right)(U\ Mass\ in\ UO_2)$$

The wt% of each uranium isotope in the fresh UO_2 composition is determined by multiplying the wt% of each uranium isotope in the enriched uranium by the weight fraction of uranium in the UO_2. The wt% of oxygen in the UO_2 is the weight fraction of oxygen in UO_2 multiplied by 100.

The burned fuel is delineated into sixteen axial regions each having a unique material composition. The height of each axial node is 22.86 cm. These nodal heights correspond directly to the nodal heights utilized in the fuel depletion calculations (p. 16, Ref. 7.3). Each nodal depleted fuel composition is obtained from SAS2H depletion calculations documented throughout Reference 7.3. The depleted fuel compositions for the best-estimate reactivity calculations may contain up to 85 isotopes from the list presented in Table 5.3.4-6. The depleted fuel compositions for the principal isotope reactivity calculations may contain up to 30 isotopes from the list presented in Table 5.3.4-7. The depleted fuel compositions for the principal actinide reactivity calculations may contain up to 15 isotopes from the list presented in Table 5.3.4-8. The depleted fuel compositions for the principal actinide reactivity calculations may contain up to 11 isotopes from the list presented in Table 5.3.4-9. Each depleted fuel composition is modeled in terms of isotopic weight percents and an overall nodal fuel density. The weight percent of each isotope in the nodal depleted fuel composition is calculated based on the total mass of all isotopes in the nodal composition. The mass of oxygen in each nodal depleted fuel composition is calculated based on the fresh fuel characteristics as described in Equations 5.3.4-1 through 5.3.4-4. This mass of oxygen is combined with the total isotopic fuel mass obtained from the depletion calculations to determine an overall total depleted fuel mass upon which the various isotopic weight percents are based. The MCNP output files for each CRC reactivity calculation are contained in Attachment IV (moved to Reference 7.13). These output files contain an echo of the MCNP input decks for each CRC statepoint reactivity calculation. The nodal fuel isotopic compositions are listed in the input decks in terms of ZAID's, weight percents, and density (g/cm^3). Each nodal fuel composition is identified by assembly and node in the material specification section of the input decks. The nodal fuel densities are shown on the geometric cell specifications for each fuel node. The nodal fuel densities are based on the fuel mass and fuel volume in each nodal region. The fuel volume is calculated using the number of fuel rods, nodal height, and pellet diameter. Therefore, dishing and chamfering of the fresh fuel pellets are accounted for on a mass basis by a slightly adjusted fuel density. However, the geometrical features of the fresh fuel pellet dishing and chamfering are not captured in the MCNP models. The purpose of the pellet dishing and chamfering is to enhance fuel performance. These geometrical features have no significant impact on system reactivity. The most important concern in determining system reactivity is to assure that fuel mass preservation is maintained. The fuel densities used in the MCNP models ensure preservation of mass.

Table 5.3.4-1. Fuel Rod Plenum Material Volume Fractions[1]

Plenum Location	Type 304 Stainless Steel	Gas (modeled as void)	Zircaloy-4
STD Upper	0.0976	0.8369	0.0655
STD Lower	0.1532	0.6388	0.2080

[1] The upper and lower fuel rod plenum volume fractions presented in Reference 7.11 were renormalized to account for the fact that the fuel rod cladding is not included in the homogenized compositions. The fuel rod cladding is modeled explicitly.

Table 5.3.4-2. Zircaloy-4 Composition (p. 21, Ref. 7.7)

Ele./Iso.	MCNP ZAID	Wt%	Ele./Iso.	MCNP ZAID	Wt%
Cr-50	24050.60c	0.004	Fe-57	26057.60c	0.004
Cr-52	24052.60c	0.084	Fe-58	26058.60c	0.001
Cr-53	24053.60c	0.010	O-16	8016.50c	0.120
Cr-54	24054.60c	0.002	Zr-nat	40000.60c	98.180
Fe-54	26054.60c	0.011	Sn-nat	50000.35c	1.400
Fe-56	26056.60c	0.184	Density = 6.56 g/cm^3		

Table 5.3.4-3. Fuel Rod Plenum Homogenized Material Compositions for STD Assembly

MCNP ZAID	Wt% of Element/Isotope in Material Composition	
	Upper Fuel Rod Plenum	Lower Fuel Rod Plenum
6000.50c	0.051	0.038
7014.50c	0.064	0.047
14000.50c	0.482	0.353
15031.50c	0.029	0.021
16032.50c	0.019	0.014
24050.60c	0.511	0.375
24052.60c	10.242	7.520

Table 5.3.4-3. Fuel Rod Plenum Homogenized Material Compositions for STD Assembly

MCNP ZAID	Wt% of Element/Isotope in Material Composition	
	Upper Fuel Rod Plenum	Lower Fuel Rod Plenum
24053.60c	1.184	0.869
24054.60c	0.300	0.220
25055.50c	1.284	0.940
26054.60c	2.520	1.848
26056.60c	40.621	29.784
26057.60c	0.947	0.694
26058.60c	0.128	0.094
28058.60c	4.003	2.930
28060.60c	1.583	1.159
28061.60c	0.070	0.051
28062.60c	0.225	0.165
28064.60c	0.059	0.043
8016.50c	0.043	0.064
40000.60c	35.134	52.030
50000.35c	0.501	0.742
Density (g/cm^3)	1.2007	2.5748

Table 5.3.4-4. Fuel Batch Enrichment and Uranium Mass Loading (p. 22, Ref. 7.11)

Fuel Batch Identifier	U-235 wt% in Uranium	Mass of Uranium per Fuel Assembly (kg)
1	2.10	458.88
2	2.60	458.88
3	3.10	458.88
4	3.50	458.97
5A	3.80	461.50
5B	3.60	460.71

Table 5.3.4-5. Fresh Fuel Material Composition for Each Fuel Batch

Fuel Batch Identifier	Wt% of Element/Isotope in Material Composition					
	U-234	U-235	U-236	U-238	Oxygen	Density (g/cm^3)[1]
1	0.015228	1.851162	0.008515	86.275673	11.849425	10.2296
2	0.019194	2.291897	0.010543	85.828262	11.850099	10.2296
3	0.023224	2.732626	0.012570	85.380814	11.850767	10.2297
4	Not Used[2]	Not Used	Not Used	Not Used	Not Used	Not Used
5A	0.028957	3.349635	0.015408	84.754288	11.851705	10.2882

Table 5.3.4-5. Fresh Fuel Material Composition for Each Fuel Batch

Fuel Batch Identifier	Wt% of Element/Isotope in Material Composition					
	U-234	U-235	U-236	U-238	Oxygen	Density (g/cm^3)[1]
5B	0.027310	3.173348	0.014597	84.933304	11.851438	10.2706

[1] This density is the fresh fuel density based on preservation of mass using the mass loading of uranium in the assembly, the initial enrichment, and the pellet stack height dimensions.

[2] The fresh fuel compositions for fuel batches 4 did not have to be specified in any of the MCNP input decks for the Sequoyah Unit 2 CRC evaluations. However, depleted fuel compositions were specified for this fuel batch.

Table 5.3.4-6. Isotope Set from which Best-Estimate MCNP Depleted Fuel Compositions are Developed

Isotope	MCNP ZAID	Isotope	MCNP ZAID	Isotope	MCNP ZAID
H-3	1003.50c	Cs-135	55135.50c	Pa-233	91233.50c
He-4	2004.50c	Ba-138	56138.50c	U-233	92233.50c
Li-6	3006.50c	Pr-141	59141.50c	U-234	92234.50c
Li-7	3007.55c	Nd-143	60143.50c	U-235	92235.53c
Be-9	4009.50c	Nd-145	60145.50c	U-236	92236.50c
O-16	8016.50c	Nd-147	60147.50c	U-237	92237.50c
As-75	33075.35c	Nd-148	60148.50c	U-238	92238.53c
Kr-80	36080.50c	Pm-147	61147.50c	Np-235	93235.35c
Kr-82	36082.50c	Pm-148	61148.50c	Np-236	93236.35c
Kr-83	36083.50c	Pm-149	61149.50c	Np-237	93237.50c
Kr-84	36084.50c	Sm-147	62147.50c	Np-238	93238.35c
Kr-86	36086.50c	Sm-149	62149.50c	Pu-237	94237.35c
Y-89	39089.50c	Sm-150	62150.50c	Pu-238	94238.50c
Zr-93	40093.50c	Sm-151	62151.50c	Pu-239	94239.55c
Nb-93	41093.50c	Sm-152	62152.50c	Pu-240	94240.50c
Mo-95	42095.50c	Eu-151	63151.55c	Pu-241	94241.50c
Tc-99	43099.50c	Eu-152	63152.50c	Pu-242	94242.50c
Ru-101	44101.50c	Eu-153	63153.55c	Am-241	95241.50c
Ru-103	44103.50c	Eu-154	63154.50c	Am-242	95242.50c
Rh-103	45103.50c	Eu-155	63155.50c	Am-243	95243.50c
Rh-105	45105.50c	Gd-152	64152.50c	Cm-242	96242.50c

Table 5.3.4-6. Isotope Set from which Best-Estimate MCNP Depleted Fuel Compositions are Developed

Isotope	MCNP ZAID	Isotope	MCNP ZAID	Isotope	MCNP ZAID
Pd-105	46105.50c	Gd-154	64154.50c	Cm-243	96243.35c
Pd-108	46108.50c	Gd-155	64155.50c	Cm-244	96244.50c
Ag-107	47107.50c	Gd-156	64156.50c	Cm-245	96245.35c
Ag-109	47109.50c	Gd-157	64157.50c	Cm-246	96246.35c
Xe-131	54131.50c	Gd-158	64158.50c	Cm-247	96247.35c
Xe-134	54134.35c	Gd-160	64160.50c	Cm-248	96248.35c
Xe-135	54135.53c	Ho-165	67165.55c		
Cs-133	55133.50c	Th-232	90232.50c		

Table 5.3.4-7. Isotope Set from which Principal Isotope MCNP Depleted Fuel Compositions are Developed

Isotope	MCNP ZAID	Isotope	MCNP ZAID	Isotope	MCNP ZAID
O-16	8016.50c	Sm-150	62150.50c	U-238	92238.53c
Mo-95	42095.50c	Sm-151	62151.50c	Np-237	93237.50c
Tc-99	43099.50c	Sm-152	62152.50c	Pu-238	94238.50c
Ru-101	44101.50c	Eu-151	63151.55c	Pu-239	94239.55c
Ru-103	44103.50c	Eu-153	63153.55c	Pu-240	94240.50c
Ag-109	47109.50c	Gd-155	64155.50c	Pu-241	94241.50c
Nd-143	60143.50c	U-233	92233.50c	Pu-242	94242.50c
Nd-145	60145.50c	U-234	92234.50c	Am-241	95241.50c
Sm-147	62147.50c	U-235	92235.53c	Am-242	95242.50c
Sm-149	62149.50c	U-236	92236.50c	Am-243	95243.50c

Table 5.3.4-8. Isotope Set from which Principal Actinide MCNP Depleted Fuel Compositions are Developed

Isotope	MCNP ZAID	Isotope	MCNP ZAID	Isotope	MCNP ZAID
O-16	8016.50c	U-238	92238.53c	Pu-241	94241.50c
U-233	92233.50c	Np-237	93237.50c	Pu-242	94242.50c
U-234	92234.50c	Pu-238	94238.50c	Am-241	95241.50c
U-235	92235.53c	Pu-239	94239.55c	Am-242	95242.50c
U-236	92236.50c	Pu-240	94240.50c	Am-243	95243.50c

Table 5.3.4-9. Isotope Set from which Actinide-Only MCNP Depleted Fuel Compositions are Developed

Isotope	MCNP ZAID	Isotope	MCNP ZAID	Isotope	MCNP ZAID
O-16	8016.50c	U-238	92238.53c	Pu-241	94241.50c
U-234	92234.50c	Pu-238	94238.50c	Pu-242	94242.50c
U-235	92235.53c	Pu-239	94239.55c	Am-241	95241.50c
U-236	92236.50c	Pu-240	94240.50c		

5.3.5. Guide Tube and Instrument Tube Materials

The guide tubes and instrument tubes are composed of Zircaloy-4 (p. 22, Ref. 7.11). The Zircaloy-4 material composition is presented in Table 5.3.4-2. The guide tubes and instrument tubes contain borated moderator as presented in Table 5.3.2-3.

5.3.6. BPRA Materials

Each BPRA may contain a different number of BPRs in a specific geometric arrangement. The BPR components include inner and outer cladding, upper plenum, and lower end-plug, and burnable poison (BP). The cladding of the Pyrex BPRs is SS304 as presented in Table 5.3.2-1 (p. 22, Ref. 7.11). The cladding of the WABA BPRs is Zircaloy-4 as presented in Table 5.3.4-2 (p. 22, Ref. 7.11). For the Pyrex BPRs, the upper plenum region is modeled as a homogenization of helium and SS304 inside the outer cladding and the lower end-plug region is modeled as SS304 inside the outer cladding (p. 15, Ref. 7.11). The upper plenum region is modeled from the bottom of the upper stem to the top of the absorber inside the cladding. The volume fraction of SS304 in the upper plenum region was calculated to be 0.1236 and the void volume fraction 0.8764. For the WABA BPRs, both the upper and lower plenum regions are modeled as a homogenization of Zircaloy-4 and borated water. It was assumed that He gas occupied the region between the inner and outer Zircaloy-4 clad above the absorber and below the upper cap. The volume fractions of Zircaloy-4, gas, and water in the upper plenum region were calculated to be 0.2383, 0.3214, and 0.4403, respectively. The volume fraction of Zircaloy-4 in the lower plenum region was calculated as 0.9076. Table 5.3.6-1 shows the volumetric breakdown of the upper plenum regions for the BPRs, and Table 5.3.6-2 shows the volumetric breakdown of the WABA BPR lower plenum. The WABA BPR upper and lower plenum region material compositions are presented in Tables 5.3.6-3 and 5.3.6-4, respectively. The upper plenum region material compositions for the Pyrex BPR are presented in Table 5.3.6-5.

The fresh BP is uniform along the axial length of the BPR. For the Pyrex BPRs, the BP material is B_2O_3-SiO_2 with an initial density of 2.299 g/cm^3 (p. 10, Ref. 7.3). The Pyrex BP contains 12.5 wt% of B_2O_3 resulting in 0.00624 grams of B-10 per cm (p. 22, Ref. 7.11). For the WABA BPRs, the BP material is Al_2O_3-B_4C with an initial density of 2.593 g/cm^3 (p. 10, Ref. 7.3). The WABA BP contains 14.0 wt% of B_4C resulting in 0.006165 grams of B-10 per cm (p. 22, Ref. 7.11). Table 5.3.6-6 presents the fresh BP compositions for both the Pyrex and WABA BPRs. The placement of the various BPRAs in the reactor core, including the number of BPRs in each, in the CRC statepoint configurations is presented in Section 5.4.

During the depletion calculations, the BP material is delineated axially along with the burned fuel, as described in Section 5.3.4, with the exception that the bottom axial fuel node does not have a corresponding BP node. In the MCNP calculations, the BP is positioned axially as shown in Figure 5.2.6-2. The B-10 and B-11 isotopic concentrations in the depleted BP are obtained from the SAS2H depletion calculations documented throughout Reference 7.3. The masses of aluminum, oxygen, silicon, and carbon in the depleted BP are modeled with the same masses as in the fresh BP. The SAS2H calculated B-10 and B-11 nodal BP masses are added to the aluminum, oxygen, silicon, and carbon masses to obtain a total mass for the depleted nodal BP composition. The weight percents of each element and isotope are calculated based on the total mass loading of depleted BP in a given node. The depleted BP density is calculated based on the mass of depleted BP in a given nodal volume. The MCNP output files in Attachment IV (moved to Reference 7.13) contain an echo of the input decks for each CRC statepoint reactivity calculation. The depleted BP isotopic compositions for each node are listed in the input decks in terms of ZAID's, weight percents, and density (g/cm^3). Each nodal BP composition is identified by assembly and node in the material specification section of the input decks. The nodal BP densities are shown on the geometric cell specifications for each BP node.

Table 5.3.6-1. BPR Upper Plenum Volumetric Breakdown

PYREX BPR			WABA BPR		
Description	Volume (cm^3)	Volume Fraction	Description	Volume (cm^3)	Volume Fraction
Inner Clad Tube (SS304)	0.46788	--	Inner Clad in Plenum (Zircaloy-4)	3.32237	--
Upper Cap (SS304)	1.18305	--	Upper Cap (Zircaloy-4)	1.08223	--
Total SS304	1.65093	0.1236	Total Zircaloy-4	4.40460	0.2383
Total Plenum	13.35168	--	Water Annulus	8.13786	0.4403
			Total Plenum	18.48340	--
Void	11.70075	0.8764	Void	5.94094	0.3214

Table 5.3.6-2. WABA BPR Lower Plenum Volumetric Breakdown

Description	Volume (cm^3)	Volume Fraction
Lower Cap (Zircaloy-4)	0.88517	0.9076
Water Annulus	0.09009	0.0924
Total Lower Plenum	0.97527	--

Table 5.3.6-3. WABA BPR Upper Plenum Homogenized Material Composition

MCNP ZAID	Wt% of Element/Isotope in Material Composition		
	Cycle 1 0.0 EFPD	Cycle 3 0.0 EFPD	Cycle 3 210.9 EFPD
24050.60c	0.003	0.003	0.003
24052.60c	0.069	0.069	0.069
24053.60c	0.008	0.008	0.008
24054.60c	0.002	0.002	0.002

Table 5.3.6-3. WABA BPR Upper Plenum Homogenized Material Composition

MCNP ZAID	Wt% of Element/Isotope in Material Composition		
	Cycle 1 0.0 EFPD	Cycle 3 0.0 EFPD	Cycle 3 210.9 EFPD
26054.60c	0.009	0.009	0.009
26056.60c	0.152	0.152	0.151
26057.60c	0.004	0.004	0.004
1001.50c	1.958	1.952	1.963
5010.50c	4.079E-03	5.214E-03	2.908E-03
5011.56c	1.863E-02	2.382E-02	1.328E-02
8016.50c	15.634	15.588	15.674
40000.60c	80.983	81.028	80.945
50000.35c	1.155	1.155	1.154
Density (g/cm^3)	1.8952	1.8942	1.8961

Table 5.3.6-4. WABA BPR Lower Plenum Homogenized Material Composition

MCNP ZAID	Wt% of Element/Isotope in Material Composition		
	Cycle 1 0.0 EFPD	Cycle 3 0.0 EFPD	Cycle 3 210.9 EFPD
24050.60c	0.004	0.004	0.004
24052.60c	0.083	0.083	0.083
24053.60c	0.010	0.010	0.010
24054.60c	0.002	0.002	0.002
26054.60c	0.011	0.011	0.011
26056.60c	0.182	0.182	0.182
26057.60c	0.004	0.004	0.004
26058.60c	0.001	0.001	0.001
1001.50c	0.129	0.129	0.130
5010.50c	2.693E-04	3.441E-04	1.921E-04
5011.56c	1.230E-03	1.572E-03	8.775E-04
8016.50c	1.144	1.141	1.147
40000.60c	97.045	97.048	97.042
50000.35c	1.384	1.384	1.384
Density (g/cm^3)	6.0235	6.0233	6.0237

Table 5.3.6-5. Pyrex BPR Upper Plenum Homogenized Material Compositions

MCNP ZAID	Wt% of Element/Isotope in Material Composition
6000.50c	0.080
7014.50c	0.100
14000.50c	0.750

Table 5.3.6-5. Pyrex BPR Upper Plenum Homogenized Material Compositions

MCNP ZAID	Wt% of Element/Isotope in Material Composition
15031.50c	0.045
16032.50c	0.030
24050.60c	0.793
24052.60c	15.903
24053.60c	1.838
24054.60c	0.466
25055.50c	2.000
26054.60c	3.918
26056.60c	63.156
26057.60c	1.472
26058.60c	0.200
28058.60c	6.234
28060.60c	2.465
28061.60c	0.109
28062.60c	0.350
28064.60c	0.092

Table 5.3.6-6. Fresh Burnable Poison Material Composition

MCNP ZAID	Wt% of Element/Isotope in Material Composition	
	Pyrex BP (B_2O_3-SiO_2)	WABA BP (B_4C-Al_2O_3)
5010.50c	0.6976	1.9684
5011.56c	3.1866	8.9917
6000.50c	—	3.0400
8016.50c	55.2092	40.4789
13027.50c	—	45.5211
14000.50c	40.9067	—

5.3.7. RCCA Materials

Each RCCA contains 24 identical control rods (CRs). The CR components include cladding, upper plenum, lower end-plug, and absorber material. The CR cladding is modeled as SS304 as presented in Table 5.3.2-1 (p. 22, Ref. 7.11). The CR upper plenum is not modeled in any of the CRC statepoint configurations due to the partial insertion of the RCCAs. The CR lower end-plug is modeled as SS304 as presented in Table 5.3.2-1 (p. 13, Ref. 7.11). The CR absorber material is Ag-In-Cd with a density of 10.16 g/cm^3 (p. 22, Ref. 7.11). Table 5.3.7-1 presents the Ag-In-Cd material composition.

Table 5.3.7-1. Ag-In-Cd Material Composition

Element / Isotope	MCNP ZAID	Wt%
Ag-107	47107.60c	41.101
Ag-109	47109.60c	38.899
Cd	48000.50c	5.000

Table 5.3.7-1. Ag-In-Cd Material Composition

Element / Isotope	MCNP ZAID	Wt%
In	49000.60c	15.000

5.4. Core Loading Descriptions

The core loading description for each CRC statepoint reactivity calculation includes the specification of the various fuel assembly locations, RCCA locations, and BPRA locations. A core loading description is provided for a particular cycle. All CRC statepoint reactivity calculations in the same reactor cycle use the same core loading description. Figures 5.4-1 and 5.4-2 present the core loading descriptions for cycles 1 and 3 of Sequoyah Unit 2, respectively. Each fuel assembly has a unique identifier corresponding to the identifiers used in the SAS2H depletion analyses. The fuel assembly placements in each core loading description are presented in Figures 5.4-3 and 5.4-4. The fuel assembly identifiers shown in Figures 5.4-3 and 5.4-4 refer to the assembly identifiers used in the depletion analyses documented throughout Reference 7.3.

	H	G	F	E	D	C	B	A
8	F(1) 1	F(1) 1	F(1) 1	F(1) 2	F(1) 1	F(1) 2	F(1) 1	F(1) 3
9		F(1) 1	F(1) 2	F(1) 1	F(1) 2	F(1) 1	F(1) 2	F(1) 3
10			F(1) 1	F(1) 2	F(1) 1	F(1) 2	F(1) 1	F(1) 3
11				F(1) 1	F(1) 2	F(1) 1	F(1) 2	F(1) 3
12					F(1) 1	F(1) 2	F(1) 3	
13						F(1) 3	F(1) 3	

CR	= Previous FA position Column/Row (C/R) - 1/8th Core
F	= Cycle FA was Fresh (F)
B	= Fuel Batch (B)

Cycle	Batch	Wt % U-235
1	1	2.10
	2	2.60
	3	3.10

BPRA Loading	
Fuel Assembly Location	Number BP Rods/ Assembly
B11, B13	9
A8, A10	10
G8, B9	12
D9, C10, D11, C12	16
E8, C8, F9, E10	20

Control Rod Bank	Core Location
CA	F8
CB	B10
CC	B8, F10
CD	H8,D8, D12

Figure 5.4-1. Core Loading Description for Cycle 1 of Sequoyah Unit 2 (p. 25, Ref. 7.11)

	H	G	F	E	D	C	B	A
8	E10 F(1) 3	F(3) 6B	C13 F(2) 4	F(3) 6B	A8 F(1) 2	F(3) 6B	C8 F(1) 3	F(3) 6B
9		D12 F(1) 3	B11 F(2) 4	B9 F(2) 4	B12 F(2) 4	E10* F(1) 3	F(3) 6A	F(3) 6A
10			E8 F(1) 3	F(3) 6B	F9 F(1) 3	F(3) 6A	D11 F(1) 3	F(3) 6A
11				E9 F(1) 3	A9 F(2) 4	C10 F(2) 4	F(3) 6A	B13 F(2) 4
12					C12 F(1) 2	A10 F(2) 4	A11 F(2) 4	
13		G8* F(1) 3				F(3) 6B	C12^ F(1) 2	
14						E9^ F(1) 3		

* Not one-eighth core symmetric
^ Not one-eighth core symmetric

CR	= Previous FA position Column/Row (C/R) - 1/8th Core
F	= Cycle FA was Fresh (F)
B	= Fuel Batch (B)

Cycle	Batch	Wt % U-235
3	2	2.60
	3	3.10
	4	3.50
	6A	3.80
	6B	3.60

BPRA Loading	
Fuel Assembly Location	Number BP Rods/ Assembly
B11	8
C8, E10	12
B9, G8	16
C10	20
E8	24

Control Rod Bank	Core Location
CA	F8
CB	B10
CC	B8, F10
CD	H8, D8 D12

Figure 5.4-2. Core Loading Description for Cycle 3 of Sequoyah Unit 2 (p. 27, Ref. 7.11)

	H	G	F	E	D	C	B	A
8	A1	A2	A3	A4	A5	A6	A7	A8
9		A9	A10	A11	A12	A13	A14	A15
10			A16	A17	A18	A19	A20	A21
11				A22	A23	A24	A25	A26
12					A27	A28	A29	
13						A30	A31	

XXX = Fuel Assembly Identifier

Figure 5.4-3. Fuel Assembly Placement in Cycle 1 of Sequoyah Unit 2 (p. 29, Ref. 7.11)

	H	G	F	E	D	C	B	A
8	A26	C2	B30	C4	A6	C6	A21	C8
9		A30	B25	B14	B29	A26a*	C14	C15
10			A8	C17	A15	C19	A31	C21
11				A29	B15	B19	C25	B31
12					A25	B21	B26	
13		A21a*				C30	A25a^	
14						A29a^		

* Not one-eighth core symmetric
^ Not one-eighth core symmetric

XXX = Fuel Assembly Identifier

Figure 5.4-4. Fuel Assembly Placements in Cycle 3 of Sequoyah Unit 2 (p. 31, Ref. 7.11)

6. Results

This calculation file documents the CRC reactivity evaluations that were performed for three statepoints from Sequoyah Unit 2. Four reactivity calculations were performed for each of the statepoints other than the beginning-of-life of the reactor (Cycle 1, 0.0 EFPD). Each of these four calculations for each statepoint used a different depleted fuel composition. The four sets of depleted fuel isotopes shown in Tables 5.3.4-6 through 5.3.4-9 were used for the "Best-Estimate", "Principal Isotope", "Principal Actinide", and "Actinide-Only" calculations. Table 6-1 presents the k_{eff} results for each of the Sequoyah Unit 2 CRC evaluations. The k_{eff} results represent the average combined collision, absorption, and track-length estimator from the MCNP calculations. The standard deviation represents the standard deviation of k_{eff} about the average combined collision, absorption, and track-length estimate due to the Monte Carlo calculation statistics.

Table 6-1. k_{eff} Results for the Sequoyah Unit 2 CRC Evaluations

Fuel Isotope Set	Sequoyah Unit 2 CRC Statepoint (k_{eff} / standard deviation)		
	Cycle 1, 0.0 EFPD	Cycle 3, 0.0 EFPD	Cycle 3, 210.9 EFPD
Best-Estimate	0.99631 / 0.00043	0.99158 / 0.00044	0.99180 / 0.00050
Principal Isotope	Not Applicable	1.00109/ 0.00047	1.00679/ 0.00046
Principal Actinide	Not Applicable	1.03341 / 0.00047	1.06657 / 0.00044
Actinide-Only	Not Applicable	1.03468 / 0.00047	1.07018 / 0.00042

The corresponding MCNP input and output filenames for the cases shown in Table 6-1 are presented in Table 6-2. The MACE input decks used to generate the MCNP input decks are presented in Attachment II (moved to Reference 7.13). The MACE generated MCNP input decks are presented in Attachment III (moved to Reference 7.13). The MCNP output files are presented in Attachment IV (moved to Reference 7.13). An error was found in the Cycle 3, 210.9 EFPD calculations and they were redone. The error showed that the fresh fuel from BOC-3 was not switched to burned fuel in the Cycle 3, 210.9 EFPD cases. Attachment V contains the revised MACE input decks, MACE generated MCNP input decks, and MCNP output decks for Cycle 3, 210.9 EFPD calculations. The data in Attachment V (moved to Reference 7.13) supersedes the previously documented data for the Cycle 3, 210.9 EFPD calculations. An additional error was found in the MACE code in which the ZAID for Ru-103 was used instead of the ZAID for Rh-103 in the Principle Isotope fuel materials cases. The MCNP input decks were corrected by hand for these cases by replacing ZAID 44103.50c with 45103.50c. The corrected MCNP input decks for the Principle Isotope cases and their corresponding outputs are listed in Attachment VI (Reference 7.13). The data listed in attachment VI supersede the previous data for the Principle Isotope cases.

Table 6-2. MCNP Input and Output Filenames for the Sequoyah Unit 2 CRC Evaluations

Fuel Isotope Set	Sequoyah Unit 2 CRC Statepoint (input filename / output filename)		
	Cycle 1, 0.0 EFPD	Cycle 3, 0.0 EFPD	Cycle 3, 210.9 EFPD
Best-Estimate	seqi1a / seqi1a.O	seqi2a / seqi2a.O	seqi3a / seqi3a.O

Table 6-2. MCNP Input and Output Filenames for the Sequoyah Unit 2 CRC Evaluations

Fuel Isotope Set	Sequoyah Unit 2 CRC Statepoint (input filename / output filename)		
	Cycle 1, 0.0 EFPD	Cycle 3, 0.0 EFPD	Cycle 3, 210.9 EFPD
Principal Isotope	Not Applicable	seqi2b / seqi2b.O	seqi3b / seqi3b.O
Principal Actinide	Not Applicable	seqi2c / seqi2c.O	seqi3c / seqi3c.O
Actinide-Only	Not Applicable	seqi2d / seqi2d.O	seqi3d / seqi3d.O

7. References

7.1 *MCNP 4B: Monte Carlo N-Particle Transport Code System.* User manual. Los Alamos National Laboratory, Los Alamos, NM. Document Number: LA-12625-M.

7.2 *SCALE 4.3: Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation.* User Manual Volumes 0 through 3, Oak Ridge National Laboratory, Document Number: CCC-545.

7.3 *CRC Depletion Calculations for Sequoyah Unit 2.* Document Identifier Number (DI#): B00000000-01717-0210-00005 REV 00, Civilian Radioactive Waste Management System (CRWMS) Management and Operating Contractor (M&O).

7.4 *Software Qualification Report for MCNP Version 4B2, A General Monte Carlo N-Particle Transport Code.* DI#: 30033-2003 REV 01, CRWMS M&O.

7.5 *Nuclide and Isotopes, Chart of the Nuclides, Fourteenth Edition.* General Electric Company, 1989.

7.6 *Radiological Health Handbook, January 1970 Revision.* Bureau of Radiological Health; U. S. Department of Health, Education, and Welfare; Public Health Service; Food and Drug Administration.

7.7 *Material Compositions and Number Densities for Neutronics Calculations.* DI#: BBA000000-01717-0200-00002 REV 00, CRWMS M&O.

7.8 *Huntington Alloys: Inconel Alloy 718,* Third Edition, 1978.

7.9 This reference is intentionally left blank.

7.10 *Scale-4 Analysis of Pressurized Water Reactor Critical Configurations: Volume 2-Sequoyah Unit 2 Cycle 3.* Document Number: ORNL/TM-12294/V2. Oak Ridge National Laboratory, March 1995.

7.11 *Summary Report of Commercial Reactor Criticality Data for Sequoyah Unit 2.* DI#: B00000000-01717-5705-00064 REV 01, CRWMS M&O.

7.12 *Addendum to Software Qualification Report for MCNP4A Covering Addition of ENDF/B-VI Cross Sections.* DI#: 30006-2005 REV 00, CRWMS M&O.

7.13 *Sequoyah Unit 2 CRC Reactivity Calculations. (DI#: B00000000-01717-0210-00006, CRWMS M&O) Attachments II through VI – 1 Data Cartridge.* Batch Number: MOY-980428-18.

8. Attachments

Table 8-1 presents the attachment specifications for this calculation file.

Table 8-1. Attachment Listing

Attachment #	# of Pages	Creation Date	Description
I	656	03/18/98	MACE, version 3, user information
II	1 (Hard-copy listing of tape content)	03/18/98	MACE input decks for the Sequoyah Unit 2 reactivity calculations (attachment tape moved to Reference 7.13)
III	1 (Hard-copy listing of tape content)	03/18/98	MACE generated MCNP input decks for the Sequoyah Unit 2 reactivity calculations (attachment tape moved to Reference 7.13)
IV	1 (Hard-copy listing of tape content)	03/18/98	MCNP output files for the Sequoyah Unit 2 reactivity calculations (attachment tape moved to Reference 7.13)
V	1 (Hard-copy listing of tape content)	4/15/98	Revised MACE input decks, MACE generated MCNP input decks, and MCNP output files for Cycle3, 210.9 EFPD of Sequoyah Unit 2 reactivity calculations. (attachment tape moved to Reference 7.13)
VI	1 (Hard-copy listing of tape content)	5/10/98	Revised MCNP input decks and MCNP output files for Principle Isotope fuel materials selection of Sequoyah Unit 2 reactivity calculations (attachment tape moved to Reference 7.13)

MACE Version 3
MCNP Accessory for CRC Evaluations

Developed by Kenneth D. Wright
Framatome Cogema Fuels
High-Level Waste Division

under contract with the

Management and Operating Contractor for the
Yucca Mountain High-Level Radioactive Waste Repository Project

Table of Contents

Item — Page

1. Introduction

The MACE (MCNP Accessory for CRC Evaluations) software routine creates MCNP input decks to evaluate commercial reactor critical (CRC) configurations. The MACE software routine utilizes three forms of input: a user-defined input deck that describes the CRC configuration, depleted fuel isotopic results from Commercial Reactor Assembly Follow Taskmaster (CRAFT) (Ref. 1) generated "*.cut" files, and depleted burnable poison isotopic results from CRAFT generated "*.cut" files. The MCNP input decks generated by MACE for CRC evaluations represent actual commercial nuclear reactors modeled in 1/8, 1/4, or full core symmetry. These input decks are typically very complex and usually contain huge amounts of data. MACE provides a consistent method for developing these input decks and performing perturbations on the input decks as part of sensitivity studies. The listing of the Fortran source code for MACE Version 3 is presented in Section 9.

2. MACE Methodology

The objective of the MACE methodology was to develop a mechanism for easy generation of the complex MCNP input decks required for CRC evaluations in a timely manner. The resulting MACE software routine is an easy-to-use program that builds a complete MCNP input deck in a modular fashion to evaluate the critical multiplication factor of CRC configurations. The MACE software routine is developed with a straight-forward programming structure that lends itself to future additions or modifications.

The MACE software routine consists of eleven subroutines: Main Program Block, INPUTDATA, WESTONE, MODDEN, INTROSECTION, FUEL, GEOSECTION, WESTCRA, WESTBPR, SURFSECTION, and CONTROL. Information is exchanged between the various subroutines through the main program block. Descriptions of the processes and calculations performed in these subroutines are presented in Sections 3.1 through 3.8. Figure 2-1 shows the flow diagram for the MACE software routine.

The MACE software routine first reads input data into a number of arrays and variables from a user-defined input deck. The input data provided to MACE contains a complete description of the CRC configuration that is required to produce the MCNP model. Using the user-defined moderator temperature and pressure, MACE then calculates the moderator density by using linear interpolation in a temperature versus pressure versus density table that is directly encoded into its programming. MACE then begins the MCNP input deck development by writing an introduction section for the input deck. Prior to continuing with the input deck generation, MACE takes a detour to retrieve and calculate the material compositions (including density) for both the fresh and depleted fuel and burnable poison that will need to be specified in the model. The source of the isotopic data for the depleted fuel and burnable poison are CRAFT generated results files, commonly called "*.cut" files. The fresh fuel and burnable poison compositions are calculated using data from the user-defined input. Once all of the fresh and depleted fuel and burnable poison materials have been defined, MACE continues with the MCNP input deck generation by creating the geometrical specifications of the model. The geometrical specifications of the CRC configurations include the following, in general:

- Non-Fuel-Related Reactor Component Specifications
- Core Assembly Lattice Layout Specification
- Fuel Assembly Lattice Layout Specification
- Fuel Assembly Spacer Grid Specifications
- Fuel Rod Universe Specifications
- Guide Tube Universe Specifications
- Instrument Tube Universe Specifications
- Control Rod Universe Specifications
- Axial Power Shaping Rod Universe Specifications
- Burnable Poison Rod Universe Specifications.

As the geometrical specifications for the model are defined, both the surfaces and materials required to define the geometrical cells are stored for later specification in their appropriate sections. Once the geometry specifications for the model are completed, MACE creates the surface specification section of the input deck. The surfaces, as created during the geometrical specification, are listed in the appropriate MCNP format by specifying the surface type and placement. The non-fuel and non-burnable poison material specifications are complete when the geometry specification is complete. The fuel and burnable poison specifications were previously made. The control section specifications follow the materials specifications. In this section, MACE specifies the parameters required for the KCODE option in the MCNP criticality calculation. These parameters are provided in the user-defined input. The final section of the MCNP input deck is the initial neutron source specification section. MACE defines a number of initial neutron source points throughout the fuel in the reactor core configuration. Once each of the sections of the MCNP input deck have been generated, it is the user responsibility to concatenate the files containing the various sections in the correct order to produce the completed MCNP input deck that is ready for execution.

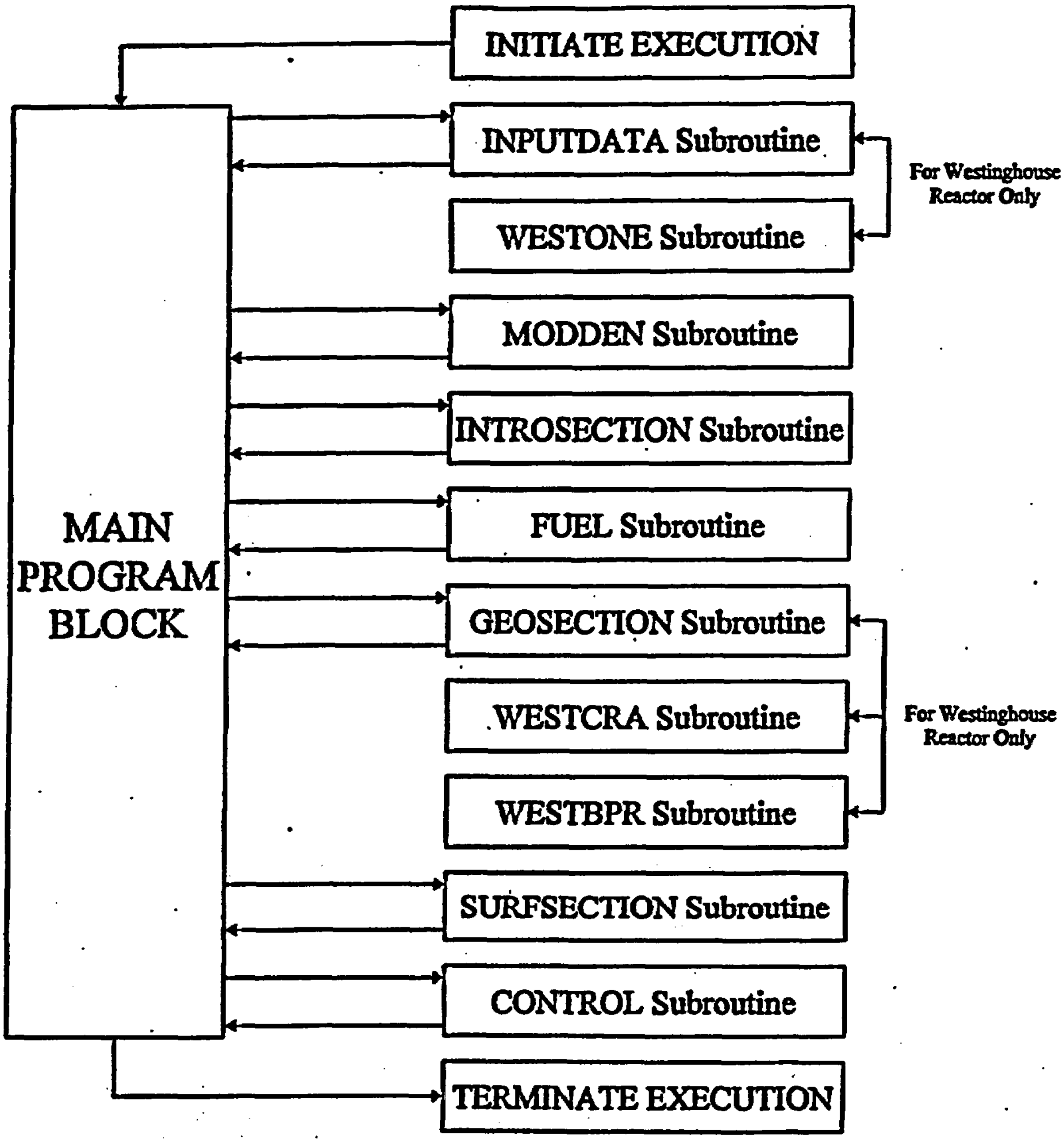


Figure 2-1 MACE Version 3 Flow Diagram

3. MACE Subroutine Descriptions

The MACE software routine is composed of eleven subroutines that exchange information through a main program block as previously shown in Figure 2-1. The following sections describe the processes and calculations performed by the subroutines.

3.1. Main Program Block

The main program block directly calls eight of the eleven subroutines in sequential order. All necessary data is conveyed between the various subroutines through the main program block. The main program block does not perform any calculations or produce any output.

3.2. INPUTDATA Subroutine

The INPUTDATA subroutine reads all of the required user-defined input from a file that is always named "inputdata". The data provided in this input file must follow a specific format. Section 4 contains a complete description of the "inputdata" file required by the MACE software routine. The INPUTDATA subroutine calls the WESTONE subroutine to read input data that is specific to Westinghouse reactor core configurations.

3.3. MODDEN Subroutine

The MODDEN subroutine calculates the moderator (borated water) density in units of grams per cubic centimeter for use in the MCNP input deck. The average moderator temperature and system pressure are provided to the MODDEN subroutine. These temperature and pressure values are used along with a linear interpolation algorithm to determine the density from a temperature versus pressure versus density table for water. The temperature versus pressure versus density table for water that is used by the MODDEN subroutine is presented in Table 3.3-1. This table is taken from the SCALE 4.3 user documentation (Ref. 2, p. S2.5.12).

The MACE software routine uses a standard linear interpolation scheme to determine the moderator density value once the temperature and pressure are known. Linear interpolation is performed using the following equation:

$$\frac{\textit{Target Value} - x_1}{\textit{Reference Value} - y_1} = \frac{x_2 - x_1}{y_2 - y_1}$$

where,

Target Value = the value for which the interpolation is being performed to obtain;
Reference Value = the known value which has a one-to-one correspondence to the Target Value;

x_1 = the target parameter value displayed in the table which corresponds to y_1;
x_2 = the target parameter value displayed in the table which corresponds to y_2;
y_1 = the reference parameter value displayed in the table which is the largest value less than the Reference Value;
y_2 = the reference parameter value displayed in the table which is the smallest value greater than the Reference Value.

The MODDEN subroutine utilizes the following procedure to perform the linear interpolation.

1) Determine which two adjacent columns of densities in the table correspond to pressures which bound the user-defined system pressure.
2) Linearly interpolate between each of the columns defined in step 1 for each row of the table to create a new density column which corresponds to the system pressure.
3) Determine which two adjacent rows in the temperature column bound the average moderator temperature in the system.
4) Linearly interpolate between the density values in the density column generated in step 2, that correspond to the two bounding temperature rows determined in step 3, to calculate the moderator density which corresponds to the moderator pressure and temperature.

Table 3.3-1. Density (g/cm³) of Subcooled Water at Various Temperatures and Pressures

Temp.	Pressure, psia								
(°F)	3000	2500	2000	1500	1000	800	600	400	200
50	1.0084	1.0069	1.0055	1.0040	1.0025	1.0019	1.0013	1.0007	1.0000
100	1.0018	1.0004	0.9989	0.9975	0.9960	0.9954	0.9948	0.9942	0.9936
150	0.9893	0.9878	0.9864	0.9849	0.9834	0.9828	0.9822	0.9815	0.9809
200	0.9725	0.9709	0.9694	0.9679	0.9663	0.9656	0.9650	0.9644	0.9637
250	0.9522	0.9505	0.9489	0.9472	0.9455	0.9449	0.9442	0.9435	0.9428
300	0.9289	0.9271	0.9252	0.9234	0.9215	0.9208	0.9200	0.9192	0.9185
350	0.9026	0.9006	0.8985	0.8964	0.8943	0.8934	0.8925	0.8916	—
400	0.8733	0.8709	0.8685	0.8660	0.8634	0.8624	0.8613	0.8603	—
450	0.8405	0.8375	0.8345	0.8314	0.8281	0.8268	0.8255	—	—
500	0.8029	0.7992	0.7952	0.7911	0.7869	0.7851	—	—	—
510	0.7947	0.7907	0.7866	0.7822	0.7776	—	—	—	—
520	0.7862	0.7820	0.7776	0.7729	0.7680	—	—	—	—
530	0.7775	0.7729	0.7682	0.7632	0.7579	—	—	—	—
540	0.7683	0.7635	0.7584	0.7530	0.7472	—	—	—	—
550	0.7589	0.7537	0.7482	0.7423	—	—	—	—	—
560	0.7490	0.7434	0.7374	0.7310	—	—	—	—	—
570	0.7386	0.7326	0.7261	0.7190	—	—	—	—	—
580	0.7278	0.7212	0.7141	0.7062	—	—	—	—	—

Table 3.3-1. Density (g/cm³) of Subcooled Water at Various Temperatures and Pressures

Temp. (°F)	Pressure, psia								
	3000	2500	2000	1500	1000	800	600	400	200
590	0.7164	0.7092	0.7012	0.6923	—	—	—	—	—
600	0.7043	0.6963	0.6874	—	—	—	—	—	—
610	0.6915	0.6825	0.6724	—	—	—	—	—	—
620	0.6777	0.6676	0.6558	—	—	—	—	—	—
630	0.6629	0.6512	0.6370	—	—	—	—	—	—
640	0.6467	0.6329	—	—	—	—	—	—	—
650	0.6288	0.6119	—	—	—	—	—	—	—
660	0.6086	0.5866	—	—	—	—	—	—	—
670	0.5850	—	—	—	—	—	—	—	—
680	0.5559	—	—	—	—	—	—	—	—

3.4. INTROSECTION Subroutine

The INTROSECTION subroutine writes the introduction section of the MCNP input deck for the CRC statepoint criticality calculation to a file called "___C__T___.intro". The first three blanks in this filename contain the three character reactor prefix specified in the MACE input deck. The two blanks following the "C" contain the two character identifier for the reactor cycle containing the CRC statepoint. The three blanks following the "T" contain the effective full-power day (EFPD) value of the CRC statepoint, rounded to the nearest whole number.

The introduction section of the MCNP input deck contains a title and a general problem description. The title of the MCNP input deck contains the reactor name, the reactor cycle identifier containing the CRC statepoint, and the EFPD value of the CRC statepoint. The general problem description provides the following information:

- The reactor design (B&W, Westinghouse, CE). The B&W and Westinghouse reactor design options are available for use in the MACE Version 3 software routine.
- The source of the isotopic data for depleted fuel and depleted burnable poison is provided. For CRC statepoint calculations, the depleted fuel and burnable poison isotopics are always obtained from SAS2H.
- The number of axial nodes used to delineate the fuel and burnable poison in both the SAS2H calculations previously performed and the MCNP model being developed by MACE. MACE Version 3 requires that the number and description of the axial nodes being modeled in the MCNP input deck be the same as those previously used in the SAS2H isotopic calculations which feed the MCNP input deck. The option will be available in future revisions of MACE to use different MCNP nodal descriptions by averaging isotopic compositions obtained from SAS2H calculations performed with more axial node detail.

3.5. FUEL Subroutine

The FUEL subroutine retrieves the depleted fuel and depleted burnable poison isotopics from CRAFT generated "*.cut" files for the appropriate assembly axial nodes. The retrieved fuel isotopics are used to make the depleted fuel material compositions for the MCNP input deck. The retrieved burnable poison isotopics are used to make the depleted burnable poison material compositions for the MCNP input deck. The material composition identifier numbers in the MCNP input deck for the fuel always begin at the number 6000. The material composition identifier numbers in the MCNP input deck for the burnable poison always begin at the number 3000.

3.5.1. Output Files Generated by the FUEL Subroutine

This section lists the files generated by the MACE software routine when executing the FUEL subroutine. These files are created in the directory in which MACE is being executed.

- The "fuel.out" file contains all of the fresh and depleted fuel material specifications ready for direct implementation in the MCNP input deck.
- The "fuelden.out" file contains a listing of all the fuel material composition densities that are utilized in the geometry specification section of the MCNP input deck.
- The "fuelch.out" file contains a listing of the isotopic masses retrieved from the CRAFT generated "*.cut" files and their corresponding MCNP cross section library identifiers to facilitate checking of the various fuel compositions.
- The "bp.out" file contains all of the fresh and depleted burnable poison material specifications ready for direct implementation in the MCNP input deck.

3.5.2. Available Fuel Isotope Sets and Cross Sections

In this section, the isotopic cross section libraries utilized in the MCNP input decks for the CRC evaluations are presented in Table 3.5.2-1. Additionally, the various fuel isotopic composition sets are listed in Tables 3.5.2-2 through 3.5.2-5. The MCNP cross section libraries utilized in the reactivity calculations are one of the primary components of the calculation that determines whether or not the neutronic behavior of the system is simulated correctly. Table 3.5.2-1 lists all of the MCNP cross section library identifiers (ZAIDs) utilized in the CRC input decks created by MACE. The MCNP ZAIDs are used to identify the cross section libraries. The ZAID consists of a 5 integer element and isotope identifier followed by a cross section library designation suffix. The first one or two integers in the ZAID refer to the atomic number of the corresponding element. The three integers preceding the decimal always refer to the isotopic mass number. The ZAID suffixes presented in Table 3.5.2-1, correspond to libraries compiled from either ENDF/B-V, ENDF/B-VI, LANL/T-2, or LLNL evaluated cross section data sets. The atom percent in nature of the various isotopes presented in Table 3.5.2-1 are obtained from Reference 3. The atomic weight ratios, temperatures, library names, and data sources are obtained from Attachment IV of Reference 4.

Table 3.5.2-1. MCNP Cross section Libraries Used in the CRC Reactivity Calculations

Element / Isotope	MCNP ZAID	Atom % in Nature	Atomic Wt. Ratio [1]	Temp. (K)	Library Name	Data Source
H-1	1001.50c	99.985	0.999167	294.0	rmccs	ENDF/B-V.0
H-3	1003.50c	0.0	2.990140	294.0	rmccs	ENDF/B-V.0
He-4	2004.50c	99.999	3.968219	294.0	rmccs	ENDF/B-V.0
Li-6	3006.50c	7.5	5.963450	294.0	rmccs	ENDF/B-V.0
Li-7	3007.55c	92.5	6.955733	294.0	rmccs	ENDF/B-V.2
Be-9	4009.50c	100.0	8.934763	294.0	rmccs	ENDF/B-V.0
B-10	5010.50c	19.400 [2]	9.926922	294.0	rmccs	ENDF/B-V.0
B-11	5011.56c	80.600 [2]	10.914730	294.0	newxs	LANL/T-2
C-nat	6000.50c	100.0	11.896914 [4]	294.0	rmccs	ENDF/B-V.0
N-14	7014.50c	99.630	13.882780	294.0	rmccs	ENDF/B-V.0
O-16	8016.50c	99.760	15.857510	294.0	rmccs	ENDF/B-V.0
Al-27	13027.50c	100.0	26.749756	294.0	rmccs	ENDF/B-V.0
Si-nat	14000.50c	100.0	27.844241	294.0	endf5p	ENDF/B-V.0
P-31	15031.50c	100.0	30.707682	294.0	endf5u	ENDF/B-V.0
S-32	16032.50c	95.02	31.788939 [3]	294.0	endf5u	ENDF/B-V.0
Ti-nat	22000.50c	100.0	47.467124	294.0	endf5u	ENDF/B-V.0
Cr-50	24050.60c	4.345	49.516983	294.0	endf60	ENDF/B-VI.1
Cr-52	24052.60c	83.790	51.494313	294.0	endf60	ENDF/B-VI.1
Cr-53	24053.60c	9.500	52.485863	294.0	endf60	ENDF/B-VI.1
Cr-54	24054.60c	2.365	53.475519	294.0	endf60	ENDF/B-VI.1
Mn-55	25055.50c	100.0	54.466099	294.0	endf5u	ENDF/B-V.0
Fe-54	26054.60c	5.900	53.476242	294.0	endf60	ENDF/B-VI.1
Fe-56	26056.60c	91.720	55.454429	294.0	endf60	ENDF/B-VI.1
Fe-57	26057.60c	2.100	56.446290	294.0	endf60	ENDF/B-VI.1
Fe-58	26058.60c	0.280	57.435600	294.0	endf60	ENDF/B-VI.1
Co-59	27059.50c	100.0	58.426930	294.0	endf5u	ENDF/B-V.0
Ni-58	28058.60c	68.270	57.437652	294.0	endf60	ENDF/B-VI.1
Ni-60	28060.60c	26.100	59.415952	294.0	endf60	ENDF/B-VI.1
Ni-61	28061.60c	1.130	60.407628	294.0	endf60	ENDF/B-VI.1
Ni-62	28062.60c	3.590	61.396349	294.0	endf60	ENDF/B-VI.1
Ni-64	28064.60c	0.910	63.378793	294.0	endf60	ENDF/B-VI.1
Cu-63	29063.60c	69.170	62.389001	294.0	endf60	ENDF/B-VI.2
Cu-65	29065.60c	30.830	64.370028	294.0	endf60	ENDF/B-VI.2
As-75	33075.35c	100.0	74.277979	0.0	rmccsa	ENDF/B-V.0
Kr-80	36080.50c	2.25	79.229851	294.0	rmccsa	ENDF/B-V.0

Table 3.5.2-1. MCNP Cross section Libraries Used in the CRC Reactivity Calculations

Element / Isotope	MCNP ZAID	Atom % in Nature	Atomic Wt. Ratio [1]	Temp. (K)	Library Name	Data Source
Kr-82	36082.50c	11.6	81.209803	294.0	rmccsa	ENDF/B-V.0
Kr-83	36083.50c	11.5	82.201858	294.0	rmccsa	ENDF/B-V.0
Kr-84	36084.50c	57.0	83.190662	294.0	rmccsa	ENDF/B-V.0
Kr-86	36086.50c	17.3	85.172596	294.0	rmccsa	ENDF/B-V.0
Y-89	39089.50c	100.0	88.142108	294.0	endf5u	ENDF/B-V.0
Zr-nat	40000.60c	100.0	90.439990	294.0	endf60	ENDF/B-VI.1
Zr-93	40093.50c	0.0	92.108361	294.0	kidman	ENDF/B-V.0
Nb-93	41093.50c	100.0	92.108263	294.0	endf5p	ENDF/B-V.0
Mo-nat	42000.50c	100.0	95.107188	294.0	endf5u	ENDF/B-V.0
Mo-95	42095.50c	15.92	94.090546	294.0	kidman	ENDF/B-V.0
Tc-99	43099.50c	0.0	98.056595	294.0	kidman	ENDF/B-V.0
Ru-101	44101.50c	17.1	100.038748	294.0	kidman	ENDF/B-V.0
Ru-103	44103.50c	0.0	102.022	294.0	kidman	ENDF/B-V.0
Rh-103	45103.50c	100.0	102.021490	294.0	rmccsa	ENDF/B-V.0
Rh-105	45105.50c	0.0	104.005	294.0	kidman	ENDF/B-V.0
Pd-105	46105.50c	22.33	104.003885	294.0	kidman	ENDF/B-V.0
Pd-108	46108.50c	26.46	106.976942	294.0	kidman	ENDF/B-V.0
Ag-107	47107.60c	51.839	105.986724	294.0	endf60	ENDF/B-VI.0
Ag-109	47109.60c	48.161	107.969204	294.0	endf60	ENDF/B-VI.0
Cd-nat	48000.50c	100.0	111.445880	294.0	endf5u	ENDF/B-V.0
In-nat	49000.60c	100.0	113.831536	294.0	endf60	ENDF/B-VI.0
Sn-nat	50000.35c	100.0	117.690428	0.0	endl85	LLNL
Xe-131	54131.50c	21.2	129.780532	294.0	kidman	ENDF/B-V.0
Xe-134	54134.35c	10.4	132.755077	0.0	endl85	LLNL
Xe-135	54135.53c	0.0	133.748208	587.0	eprixs	ENDF/B-V
Cs-133	55133.50c	100.0	131.763705	294.0	kidman	ENDF/B-V.0
Cs-135	55135.50c	0.0	133.746975	294.0	kidman	ENDF/B-V.0
Ba-138	56138.50c	71.70	136.720557	294.0	rmccs	ENDF/B-V.0
Pr-141	59141.50c	100.0	139.697185	294.0	kidman	ENDF/B-V.0
Nd-143	60143.50c	12.18	141.682152	294.0	kidman	ENDF/B-V.0
Nd-145	60145.50c	8.30	143.667706	294.0	kidman	ENDF/B-V.0
Nd-147	60147.50c	0.0	145.654	294.0	kidman	ENDF/B-V.0
Nd-148	60148.50c	5.76	146.646216	294.0	kidman	ENDF/B-V.0
Pm-147	61147.50c	0.0	145.653	294.0	kidman	ENDF/B-V.0
Pm-148	61148.50c	0.0	146.647	294.0	kidman	ENDF/B-V.0

Table 3.5.2-1. MCNP Cross section Libraries Used in the CRC Reactivity Calculations

Element / Isotope	MCNP ZAID	Atom % in Nature	Atomic Wt. Ratio [1]	Temp. (K)	Library Name	Data Source
Pm-149	61149.50c	0.0	147.639	294.0	kidman	ENDF/B-V.0
Sm-147	62147.50c	15.0	145.652830	294.0	kidman	ENDF/B-V.0
Sm-149	62149.50c	13.8	147.637915	294.0	endf5u	ENDF/B-V.0
Sm-150	62150.50c	7.4	148.629416	294.0	kidman	ENDF/B-V.0
Sm-151	62151.50c	0.0	149.623	294.0	kidman	ENDF/B-V.0
Sm-152	62152.50c	26.7	150.614670	294.0	kidman	ENDF/B-V.0
Eu-151	63151.55c	47.8	149.623378	294.0	newxs	LANL/T-2
Eu-152	63152.50c	0.0	150.616668	294.0	endf5u	ENDF/B-V.0
Eu-153	63153.55c	52.2	151.607568	294.0	newxs	LANL/T-2
Eu-154	63154.50c	0.0	152.600719	294.0	endf5u	ENDF/B-V.0
Eu-155	63155.50c	0.0	153.592	294.0	kidman	ENDF/B-V.0
Gd-152	64152.50c	0.20	150.614731	294.0	endf5u	ENDF/B-V.0
Gd-154	64154.50c	2.18	152.598614	294.0	endf5u	ENDF/B-V.0
Gd-155	64155.50c	14.80	153.591761	294.0	endf5u	ENDF/B-V.0
Gd-156	64156.50c	20.47	154.582676	294.0	endf5u	ENDF/B-V.0
Gd-157	64157.50c	15.65	155.575907	294.0	endf5u	ENDF/B-V.0
Gd-158	64158.50c	24.84	156.567459	294.0	endf5u	ENDF/B-V.0
Gd-160	64160.50c	21.86	158.553203	294.0	endf5u	ENDF/B-V.0
Ho-165	67165.55c	100.0	163.513493	294.0	newxs	LANL/T-2
Ta-181	73181.50c	99.988	179.393575	294.0	endf5u	ENDF/B-V.0
Th-232	90232.50c	100.0	230.044724	294.0	endf5u	ENDF/B-V.0
Pa-233	91233.50c	0.0	231.038304	294.0	endf5u	ENDF/B-V.0
U-233	92233.50c	0.0	231.037695	294.0	rmccs	ENDF/B-V.0
U-234	92234.50c	0.0055	232.030412	294.0	endf5p	ENDF/B-V.0
U-235	92235.53c	0.7200	233.024773	587.0	eprixs	ENDF/B-V.0
U-236	92236.50c	0.0	234.017806	294.0	endf5p	ENDF/B-V.0
U-237	92237.50c	0.0	235.012352	294.0	endf5p	ENDF/B-V.0
U-238	92238.53c	99.2745	236.005803	587.0	eprixs	ENDF/B-V.0
Np-235	93235.35c	0.0	233.024904	0.0	endl85	LLNL
Np-236	93236.35c	0.0	234.018854	0.0	endl85	LLNL
Np-237	93237.50c	0.0	235.011799	294.0	endf5p	ENDF/B-V.0
Np-238	93238.35c	0.0	236.005958	0.0	endl85	LLNL
Pu-237	94237.35c	0.0	235.012031	0.0	endl85	LLNL
Pu-238	94238.50c	0.0	236.004583	294.0	endf5p	ENDF/B-V.0
Pu-239	94239.55c	0.0	236.998573	294.0	rmccs	ENDF/B-V.2

Table 3.5.2-1. MCNP Cross section Libraries Used in the CRC Reactivity Calculations

Element / Isotope	MCNP ZAID	Atom % in Nature	Atomic Wt. Ratio [1]	Temp. (K)	Library Name	Data Source
Pu-240	94240.50c	0.0	237.991619	294.0	rmccs	ENDF/B-V.0
Pu-241	94241.50c	0.0	238.986041	294.0	endf5p	ENDF/B-V.0
Pu-242	94242.50c	0.0	239.979326	294.0	endf5p	ENDF/B-V.0
Am-241	95241.50c	0.0	238.986019	294.0	endf5u	ENDF/B-V.0
Am-242m	95242.50c	0.0	239.980121	294.0	endf5u	ENDF/B-V.0
Am-243	95243.50c	0.0	240.973348	294.0	endf5u	ENDF/B-V.0
Cm-242	96242.50c	0.0	239.979418	294.0	endf5u	ENDF/B-V.0
Cm-243	96243.35c	0.0	240.973356	0.0	endl85	LLNL
Cm-244	96244.50c	0.0	241.966119	294.0	endf5u	ENDF/B-V.0
Cm-245	96245.35c	0.0	242.960245	0.0	endl85	LLNL
Cm-246	96246.35c	0.0	243.953373	0.0	endl85	LLNL
Cm-247	96247.35c	0.0	244.947884	0.0	endl85	LLNL
Cm-248	96248.35c	0.0	245.941272	0.0	endl85	LLNL

[1] The atomic weight ratio presented for each isotope/element is the ratio of the isotope/element mass to the mass of a neutron. The mass of a neutron is 1.008664904 amu (Ref. 3). The atomic weight ratio values are obtained from the "xsdir" file for MCNP as described in Reference 4.

[2] The atom percent in nature of B-10 and B-11 varies significantly between different geographical regions of the world. The atom percents in nature that are listed in Table 3.5.2-1 for B-10 and B-11 were obtained from page 232 of Reference 5.

[3] The atomic weight ratio for natural sulfur is used with the S-32 cross section library.

[4] The atomic weight ratio for carbon-12 is used with the natural carbon cross-section library.

Table 3.5.2-2 Best-Estimate Isotope Set from which MCNP Spent Fuel Compositions May Be Developed

Isotope	MCNP ZAID	Isotope	MCNP ZAID	Isotope	MCNP ZAID
H-3	1003.50c	Cs-135	55135.50c	Pa-233	91233.50c
He-4	2004.50c	Ba-138	56138.50c	U-233	92233.50c
Li-6	3006.50c	Pr-141	59141.50c	U-234	92234.50c
Li-7	3007.55c	Nd-143	60143.50c	U-235	92235.53c
Be-9	4009.50c	Nd-145	60145.50c	U-236	92236.50c
O-16	8016.50c	Nd-147	60147.50c	U-237	92237.50c

Table 3.5.2-2 Best-Estimate Isotope Set from which MCNP Spent Fuel Compositions May Be Developed

Isotope	MCNP ZAID	Isotope	MCNP ZAID	Isotope	MCNP ZAID
As-75	33075.35c	Nd-148	60148.50c	U-238	92238.53c
Kr-80	36080.50c	Pm-147	61147.50c	Np-235	93235.35c
Kr-82	36082.50c	Pm-148	61148.50c	Np-236	93236.35c
Kr-83	36083.50c	Pm-149	61149.50c	Np-237	93237.50c
Kr-84	36084.50c	Sm-147	62147.50c	Np-238	93238.35c
Kr-86	36086.50c	Sm-149	62149.50c	Pu-237	94237.35c
Y-89	39089.50c	Sm-150	62150.50c	Pu-238	94238.50c
Zr-93	40093.50c	Sm-151	62151.50c	Pu-239	94239.55c
Nb-93	41093.50c	Sm-152	62152.50c	Pu-240	94240.50c
Mo-95	42095.50c	Eu-151	63151.55c	Pu-241	94241.50c
Tc-99	43099.50c	Eu-152	63152.50c	Pu-242	94242.50c
Ru-101	44101.50c	Eu-153	63153.55c	Am-241	95241.50c
Ru-103	44103.50c	Eu-154	63154.50c	Am-242	95242.50c
Rh-103	45103.50c	Eu-155	63155.50c	Am-243	95243.50c
Rh-105	45105.50c	Gd-152	64152.50c	Cm-242	96242.50c
Pd-105	46105.50c	Gd-154	64154.50c	Cm-243	96243.35c
Pd-108	46108.50c	Gd-155	64155.50c	Cm-244	96244.50c
Ag-107	47107.50c	Gd-156	64156.50c	Cm-245	96245.35c
Ag-109	47109.50c	Gd-157	64157.50c	Cm-246	96246.35c
Xe-131	54131.50c	Gd-158	64158.50c	Cm-247	96247.35c
Xe-134	54134.35c	Gd-160	64160.50c	Cm-248	96248.35c
Xe-135	54135.53c	Ho-165	67165.55c		
Cs-133	55133.50c	Th-232	90232.50c		

Table 3.5.2-3 Principal Isotope Set from which MCNP Spent Fuel Compositions May Be Developed

Isotope	MCNP ZAID	Isotope	MCNP ZAID	Isotope	MCNP ZAID
O-16	8016.50c	Sm-150	62150.50c	U-238	92238.53c
Mo-95	42095.50c	Sm-151	62151.50c	Np-237	93237.50c

Table 3.5.2-3 Principal Isotope Set from which MCNP Spent Fuel Compositions May Be Developed

Isotope	MCNP ZAID	Isotope	MCNP ZAID	Isotope	MCNP ZAID
Tc-99	43099.50c	Sm-152	62152.50c	Pu-238	94238.50c
Ru-101	44101.50c	Eu-151	63151.55c	Pu-239	94239.55c
Rh-103	45103.50c	Eu-153	63153.55c	Pu-240	94240.50c
Ag-109	47109.50c	Gd-155	64155.50c	Pu-241	94241.50c
Nd-143	60143.50c	U-233	92233.50c	Pu-242	94242.50c
Nd-145	60145.50c	U-234	92234.50c	Am-241	95241.50c
Sm-147	62147.50c	U-235	92235.53c	Am-242m	95242.50c
Sm-149	62149.50c	U-236	92236.50c	Am-243	95243.50c

Table 3.5.2-4 Principal Actinide Set from which MCNP Spent Fuel Compositions May Be Developed

Isotope	MCNP ZAID	Isotope	MCNP ZAID	Isotope	MCNP ZAID
O-16	8016.50c	U-238	92238.53c	Pu-241	94241.50c
U-233	92233.50c	Np-237	93237.50c	Pu-242	94242.50c
U-234	92234.50c	Pu-238	94238.50c	Am-241	95241.50c
U-235	92235.53c	Pu-239	94239.55c	Am-242m	95242.50c
U-236	92236.50c	Pu-240	94240.50c	Am-243	95243.50c

Table 3.5.2-5 Actinide-Only Set from which MCNP Spent Fuel Compositions May Be Developed

Isotope	MCNP ZAID	Isotope	MCNP ZAID	Isotope	MCNP ZAID
O-16	8016.50c	U-238	92238.53c	Pu-241	94241.50c
U-234	92234.50c	Pu-238	94238.50c	Pu-242	94242.50c
U-235	92235.53c	Pu-239	94239.55c	Am-241	95241.50c
U-236	92236.50c	Pu-240	94240.50c	—	—

3.5.3. Fresh Fuel Material Composition Calculations

The fresh fuel (UO_2) material composition is calculated using the initial weight percent (wt. %) U-235 enrichment in the uranium of the UO_2 and the mass loading of uranium in the assembly. The following equations are used to calculate the fresh fuel composition. The wt. % of each uranium isotope in the following equations refers to the wt. % of that isotope in the uranium of the UO_2.

Equation 3.5.3-1 (Ref. 6).

$$Wt.\%U^{234} = 0.007731\,(Wt.\%U^{235})^{1.0837}$$

Equation 3.5.3-2 (Ref. 6).

$$Wt.\%U^{236} = 0.0046\,(Wt.\%U^{235})$$

Equation 3.5.3-3.

$$Wt.\%U^{238} = 100.0 - (Wt.\%U^{234}) - (Wt.\%U^{235}) - (Wt.\%U^{236})$$

Equation 3.5.3-4.

$$\frac{grams\ U}{mol\ UO_2} = \begin{array}{l} 1.008664904[(232.030)(Wt.\%U^{234}) + (233.025)(Wt.\%U^{235}) + \\ (234.018)(Wt.\%U^{236}) + (236.006)(Wt.\%U^{238})](\frac{1}{100}) \end{array}$$

Equation 3.5.3-5.

$$\frac{grams\ O}{mol\ UO_2} = (2)(1.008664904)(15.858)$$

Equation 3.5.3-6.

$$\frac{grams\ O}{fuel\ assembly} = \frac{(\frac{grams\ O}{mol\ UO_2})}{(\frac{grams\ U}{mol\ UO_2})}(\frac{grams\ U}{fuel\ assembly})$$

Equation 3.5.3-7.

$$Total\ Grams\ UO_2\ in\ Fuel\ Assembly = \begin{array}{l} Grams\ O\ in\ Fuel\ Assembly + \\ Grams\ U\ in\ Fuel\ Assembly \end{array}$$

Equation 3.5.3-8.

$$Wt.\%\,O\,in\,UO_2 = \frac{grams\,O\,in\,assembly}{total\,grams\,UO_2\,in\,assembly}(100)$$

Equation 3.5.3-9.

$$Wt.\%\,U\,in\,UO_2 = 100 - (Wt.\%\,O\,in\,UO_2)$$

Equation 3.5.3-10.

$$Wt.\%\,U^{234}\,in\,UO_2 = \frac{(Wt.\%\,U^{234}\,in\,U)}{100}(Wt.\%\,U\,in\,UO_2)$$

Equation 3.5.3-11.

$$Wt.\%\,U^{235}\,in\,UO_2 = \frac{(Wt.\%\,U^{235}\,in\,U)}{100}(Wt.\%\,U\,in\,UO_2)$$

Equation 3.5.3-12.

$$Wt.\%\,U^{236}\,in\,UO_2 = \frac{(Wt.\%\,U^{236}\,in\,U)}{100}(Wt.\%\,U\,in\,UO_2)$$

Equation 3.5.3-13.

$$Wt.\%\,U^{238}\,in\,UO_2 = \frac{(Wt.\%\,U^{238}\,in\,U)}{100}(Wt.\%\,U\,in\,UO_2)$$

If the fuel density option in the MACE input deck is specified as either "T" or "C" then the fresh fuel density is modeled as 10.41 g/cm^3 which corresponds to a nominal pressed fuel density. If the fuel density option in the MACE input deck is something other than "T" or "C" then the following equation is used to calculate the fresh fuel density.

Equation 3.5.3-14.

$$Fuel\,Density(\frac{g}{cm^3}) = \frac{(grams\,of\,U\,in\,assembly + grams\,of\,O\,in\,assembly)}{\left[\begin{array}{c}(\pi)(fuel\,pellet\,radius(cm))^2 * (number\,of\,fuel\,rods\,in\,assembly) * \\ (total\,active\,fuel\,height\,(cm))\end{array}\right]}$$

3.5.4. Depleted Fuel Material Composition Calculations

The depleted fuel (UO_2 + actinides + fission products) material composition is calculated by retrieving actinide and fission product isotopics from appropriate CRAFT generated "*.cut" files. Reference 1 has a complete description of the "*.cut" files and their naming convention. The necessary "*.cut" files must be accessible to MACE in a predetermined arrangement. A directory bearing the identifier of each fuel assembly must exist in the directory from which MACE is executed. Each assembly identifier directory must contain the "*.cut" files that correspond to the CRC statepoint being evaluated for each axial node of that assembly. During the MACE execution, each "*.cut" file is copied to a file called "N__.temp" in the directory in which MACE is being executed. The two blanks after the "N" contain the node number corresponding to that file. Each "*.cut" file is copied in this manner to facilitate isotopic retrieval. The "*.cut" files are copied for one assembly, processed, and then removed so that the next assembly can be processed.

The grams-per-assembly values obtained from the ORIGEN-S output in the various "*.cut" files will correspond to grams-per-assembly-node (g/node) if the CRAFT calculations were performed by preserving the actual fuel mass per axial node. All CRC CRAFT calculations should be performed such that the actual fuel mass per axial node is preserved. The actinide and fission product isotopic masses (g/node) are retrieved for each axial node and stored in arrays. MACE allows the use of one of four different isotopic sets to model the depleted fuel compositions. These four isotopic sets are identified as follows:

- Best-Estimate Isotopic Set (See Table 3.5.2-2 for a listing of isotopics.);
- Principal Isotope Set (See Table 3.5.2-3 for a listing of isotopics.);
- Principal Actinide Set (See Table 3.5.2-4 for a listing of isotopics.);
- Actinide-Only Set (See Table 3.5.2-5 for a listing of isotopics.).

The total mass of fuel isotopics retrieved for the isotopic set of choice in a given node is summed. The mass of oxygen in the fresh assembly (Equation 3.5.3-6) is multiplied by the ratio of the node height to the total active fuel height to determine the mass of oxygen in the node of interest. This value is added to the total mass of retrieved fuel isotopics to determine the total depleted fuel mass in the assembly axial node. The mass of each retrieved isotope and oxygen in the node is divided by the total depleted fuel mass in the node and multiplied by 100 to obtain the weight percent of the isotope in the fuel material composition. The MCNP cross section identifier and weight percent for each isotope are then written to the file called "fuel.out" to be used in the MCNP input deck.

If the fuel density option is specified as "T" in the MACE input deck, the depleted fuel material density is calculated with the following equations.

Equation 3.5.4-1.

$$\text{Stack Fraction} = \frac{\left[\begin{array}{c}(10.41\,\frac{grams}{cm^3})(\text{node height (cm)})(\pi)^* \\ (\text{pellet radius (cm)})^2(\text{number of fuel rods})\end{array}\right]}{\left[\left(\begin{array}{c}\text{grams U in assembly} + \\ \text{grams O in assembly}\end{array}\right)\left(\frac{\text{node height (cm)}}{\text{active fuel height (cm)}}\right)\right]}$$

Equation 3.5.4-2.

$$\begin{array}{c}\text{Depleted} \\ \text{Fuel} \\ \text{Density}\end{array}\left(\frac{grams}{cm^3}\right) = \frac{\left(\begin{array}{c}\text{total grams of} \\ \text{depleted fuel in node}\end{array}\right)}{\left[\begin{array}{c}(\text{node height (cm)})(\pi)^* \\ (\text{pellet radius (cm)})^2 * \\ (\text{number of fuel rods})\end{array}\right]}(\text{Stack Fraction})$$

If the fuel density option is not specified as "T" in the MACE input deck, the depleted fuel material density is calculated with the following equation.

Equation 3.5.4-3.

$$\begin{array}{c}\text{Depleted} \\ \text{Fuel} \\ \text{Density}\end{array}\left(\frac{grams}{cm^3}\right) = \frac{\left(\begin{array}{c}\text{total grams of} \\ \text{depleted fuel in node}\end{array}\right)}{\left[\begin{array}{c}(\text{node height (cm)})(\pi)^* (\text{pellet radius (cm)})^2 * \\ (\text{number of fuel rods})\end{array}\right]}$$

3.5.5. Fresh Burnable Poison Material Composition Calculations

The fresh burnable poison compositions are calculated with the following equations. The fresh burnable poison density is entered by the user in the MACE input deck and applied directly in the MCNP input deck.

Equation 3.5.5-1.

$$\frac{grams\ C}{mol\ B_4C} = (1.008664904)(11.8969)$$

Equation 3.5.5-2.

$$\frac{grams\ B}{mol\ B_4C} = (4)(1.008664904)[(9.9269)(0.194) + (10.9147)(0.806)]$$

Equation 3.5.5-3.

$$\frac{grams\ Al}{mol\ Al_2O_3} = (2)(1.008664904)(26.75)$$

Equation 3.5.5-4.

$$\frac{grams\ O}{mol\ Al_2O_3} = (3)(1.008664904)(15.858)$$

Equation 3.5.5-5.

$$B^{10}\ Wt.\%\ in\ B_4C = \frac{(9.9269)(0.194)(100)\left(\frac{grams\ B}{mol\ B_4C}\right)}{\left\{\begin{array}{c}[(9.9269)(0.194)+(10.9147)(0.806)]^{*}\\ \left(\frac{grams\ B}{mol\ B_4C} + \frac{grams\ C}{mol\ B_4C}\right)\end{array}\right\}}$$

Equation 3.5.5-6.

$$B^{11}\ Wt.\%\ in\ B_4C = \frac{(10.9147)(0.806)(100)\left(\frac{grams\ B}{mol\ B_4C}\right)}{\left\{\begin{array}{c}[(9.9269)(0.194)+(10.9147)(0.806)]^{*}\\ \left(\frac{grams\ B}{mol\ B_4C} + \frac{grams\ C}{mol\ B_4C}\right)\end{array}\right\}}$$

Equation 3.5.5-7.

$$C\ Wt.\%\ in\ B_4C = \frac{\left(\frac{grams\ C}{mol\ B_4C}\right)(100)}{\left(\frac{grams\ B}{mol\ B_4C} + \frac{grams\ C}{mol\ B_4C}\right)}$$

Equation 3.5.5-8.

$$Al\,Wt.\%\,in\,Al_2O_3 = \frac{\left(\frac{grams\,Al}{mol\,Al_2O_3}\right)(100)}{\left(\frac{grams\,Al}{mol\,Al_2O_3} + \frac{grams\,O}{mol\,Al_2O_3}\right)}$$

Equation 3.5.5-9.

$$O\,Wt.\%\,in\,Al_2O_3 = \frac{\left(\frac{grams\,O}{mol\,Al_2O_3}\right)(100)}{\left(\frac{grams\,Al}{mol\,Al_2O_3} + \frac{grams\,O}{mol\,Al_2O_3}\right)}$$

Equation 3.5.5-10.

$$B^{10}\,Wt.\%\,in\,Al_2O_3\text{-}B_4C = \left(\frac{B^{10}\,Wt.\%\,in\,B_4C}{100}\right) * (B_4C\,Wt.\%\,in\,Al_2O_3\text{-}B_4C)$$

Equation 3.5.5-11.

$$B^{11}\,Wt.\%\,in\,Al_2O_3\text{-}B_4C = \left(\frac{B^{11}\,Wt.\%\,in\,B_4C}{100}\right) * (B_4C\,Wt.\%\,in\,Al_2O_3\text{-}B_4C)$$

Equation 3.5.5-12.

$$C\,Wt.\%\,in\,Al_2O_3\text{-}B_4C = \left(\frac{C\,Wt.\%\,in\,B_4C}{100}\right) * (B_4C\,Wt.\%\,in\,Al_2O_3\text{-}B_4C)$$

Equation 3.5.5-13.

$$Al\,Wt.\%\,in\,Al_2O_3\text{-}B_4C = \left(\frac{Al\,Wt.\%\,in\,Al_2O_3}{100}\right) * (100 - B_4C\,Wt.\%\,in\,Al_2O_3\text{-}B_4C)$$

Equation 3.5.5-14.

$$O\,Wt.\%\,in\,Al_2O_3-B_4C=\left(\frac{O\,Wt.\%\,in\,Al_2O_3}{100}\right)*(100-B_4C\,Wt.\%\,in\,Al_2O_3-B_4C)$$

3.5.6. Depleted Burnable Poison Material Composition Calculations

The first step in calculating the depleted burnable poison composition is to retrieve the B-10 and B-11 isotopic masses from the "*.cut" files in the same manner as previously described for the fuel. The carbon, aluminum, and oxygen weight percent values are calculated for the depleted burnable poison material in the same manner as the fresh burnable poison material. The volume of the burnable poison material in the fuel assembly axial node is calculated with the following equation.

Equation 3.5.6-1.

$$\begin{matrix}Burnable\ Poison\ (BP)\\ Volume\ in\ Node\end{matrix}=\left[\begin{matrix}(\pi)(BP\ Pellet\ Radius(cm))^2*\\ (Node\ Height\ (cm))(Number\ of\ BP\ Rods)\end{matrix}\right]$$

Sections 3.5.6.1 and 3.5.6.2 present the calculations performed to determine the depleted burnable poison compositions for Al_2O_3-B_4C and burnable poison other than Al_2O_3-B_4C, respectively.

3.5.6.1. Calculation of Depleted Al_2O_3-B_4C Burnable Poison Compositions

The composition of depleted Al_2O_3-B_4C in the fuel assembly axial node is calculated with the following equations.

Equation 3.5.6.1-1

$$\begin{matrix}Total\ Grams\ of\\ Depleted\ Al_2O_3-B_4C\end{matrix}=\left[\begin{matrix}(Al\,Wt.\%\,in\,Al_2O_3-B_4C+O\,Wt.\%\,in\,Al_2O_3-B_4C+\\ C\,Wt.\%\,in\,Al_2O_3-B_4C)*(Fresh\ Burnable\ Poison\ (BP)\ Density\ (\frac{g}{cm^3}))*\\ (BP\ Volume\ in\ Node(cm^3))(\frac{1}{100})+(Retrieved\ B^{10}\ Grams\ in\ Node)+\\ (Retrieved\ B^{11}\ Grams\ in\ Node)\end{matrix}\right]$$

Equation 3.5.6.1-2.

$$\frac{B^{10}\ Wt.\%\ in}{Depleted\ Al_2O_3 - B_4C} = \frac{(Retrieved\ B^{10}\ Grams\ in\ Node)(100)}{(Total\ Depleted\ Al_2O_3 - B_4C\ Grams\ in\ Node)}$$

Equation 3.5.6.1-3.

$$\frac{B^{11}\ Wt.\%\ in}{Depleted\ Al_2O_3 - B_4C} = \frac{(Retrieved\ B^{11}\ Grams\ in\ Node)(100)}{(Total\ Depleted\ Al_2O_3 - B_4C\ Grams\ in\ Node)}$$

Equation 3.5.6.1-4.

$$\begin{pmatrix} Al\ Wt.\%\ in \\ Depleted\ Al_2O_3 - B_4C \end{pmatrix} = \frac{\begin{pmatrix} Al\ Wt.\%\ in \\ Fresh\ Al_2O_3 - B_4C \end{pmatrix}\begin{pmatrix} Density\ of \\ Fresh\ Al_2O_3 - B_4C \end{pmatrix}\begin{pmatrix} BP\ Volume \\ in\ Node \end{pmatrix}}{Total\ Mass\ of\ Depleted\ Al_2O_3 - B_4C}$$

Equation 3.5.6.1-5.

$$\begin{pmatrix} O\ Wt.\%\ in \\ Depleted\ Al_2O_3 - B_4C \end{pmatrix} = \frac{\begin{pmatrix} O\ Wt.\%\ in \\ Fresh\ Al_2O_3 - B_4C \end{pmatrix}\begin{pmatrix} Density\ of \\ Fresh\ Al_2O_3 - B_4C \end{pmatrix}\begin{pmatrix} BP\ Volume \\ in\ Node \end{pmatrix}}{Total\ Mass\ of\ Depleted\ Al_2O_3 - B_4C}$$

Equation 3.5.6.1-6.

$$\begin{pmatrix} C\ Wt.\%\ in \\ Depleted\ Al_2O_3 - B_4C \end{pmatrix} = \frac{\begin{pmatrix} C\ Wt.\%\ in \\ Fresh\ Al_2O_3 - B_4C \end{pmatrix}\begin{pmatrix} Density\ of \\ Fresh\ Al_2O_3 - B_4C \end{pmatrix}\begin{pmatrix} BP\ Volume \\ in\ Node \end{pmatrix}}{Total\ Mass\ of\ Depleted\ Al_2O_3 - B_4C}$$

Equation 3.5.6.1-7.

$$\begin{matrix} Depleted\ Burnable \\ Poison\ Density \end{matrix}\left(\frac{g}{cm^3}\right) = \frac{Total\ Depleted\ Grams\ Al_2O_3 - Bsub4C\ in\ Node}{Burnable\ Poison\ Volume\ (cm^3)\ in\ Node}$$

3.5.6.2. Calculation of Depleted Burnable Poison Compositions Other Than Al_2O_3-B_4C

The total mass of depleted burnable poison other than Al_2O_3-B_4C in the fuel assembly axial node is calculated with the following equation:

$$\textit{Total Mass of Depleted BP} = \left[\sum_i \begin{pmatrix} \textit{Original Mass of Isotope/Element} \\ \textit{Other Than } B^{10} \textit{ and } B^{11} \textit{ in Fresh BP} \end{pmatrix}_i + \left(\textit{Mass of Depleted } B^{10}\right) + \left(\textit{Mass of Depleted } B^{10}\right) \right]$$

where i = isotope or element.

The weight percentages of the various isotopes or elements in the depleted burnable poison composition are calculated in a manner similar to that presented in Equations 3.5.6.1-2 through 3.5.6.1-6.

3.5.7. Non-Absorbing Burnable Poison Material Composition Calculations

The MACE input specification allows the user to define burnable poison material nodes that contain a non-absorbing burnable poison material (this facilitates B&W burnable poison rod assembly design modeling). The non-absorbing burnable poison material utilized in the MCNP input decks for B&W reactor CRC evaluations is Al_2O_3. The material composition for the Al_2O_3 is calculated in the same manner as previously described for the regular burnable poison in Equations 3.5.5-8 and 3.5.5-9. The non-absorbing burnable poison density is provided in the MACE input deck and implemented directly in the MCNP input deck. After the necessary fuel and burnable poison compositions are calculated (both fresh and depleted) and written to the appropriate files, the MACE program returns to the main program block to call additional subroutines.

3.6. GEOSECTION Subroutine

The GEOSECTION subroutine creates and writes the entire geometry section of the input deck. In the process of creating the geometrical specifications, this subroutine creates all surface specifications and material compositions other than fuel and burnable poison materials. Two output files are created by the GEOSECTION subroutine that contain the geometry specification and material specifications, respectively.

3.6.1. Output files generated by the GEOSECTION Subroutine

The GEOSECTION subroutine creates two output files. The first of these output files contains the entire geometry specification section that may be incorporated directly into the MCNP input deck. The second of these output files contains the entire material specification section other than fuel and burnable poison materials that may be incorporated directly into the MCNP input deck. The naming conventions for these two output files are "___C__T___.geo" or "___C__T___.mat". The three spaces before the "C" contain the reactor identification prefix as specified in the MACE input deck. The two spaces following the "C" contain the CRC reactor cycle identifier. The three spaces following the "T" contain the CRC statepoint effective full-power day value.

3.6.2. Simultaneous Geometry, Surface, and Material Specification Handling

Surfaces must be defined by the GEOSECTION subroutine to facilitate modeling of the various geometric cells required to develop the CRC MCNP model. As the geometrical features are defined, the created surface definitions are stored in two arrays. One array contains the surface type specification, and the other array contains the corresponding surface positions. As the geometry specification continues, if a surface is required to bound a geometric cell in the MCNP model and that surface has already been defined, the GEOSECTION subroutine recognizes the previous surface definition and utilizes it rather than creating a duplicate surface definition. The two surface definition arrays are exported from the GEOSECTION subroutine for use in the SURFSECTION subroutine.

Like the surface specifications, the GEOSECTION subroutine must define material composition specifications to fill geometric cells that are created. The GEOSECTION subroutine has access to the fuel and burnable poison material identification numbers as defined in the FUEL subroutine. However, all other material compositions are created and given a material identification number by the GEOSECTION subroutine as required during the geometry development.

3.6.3. Non-Fuel-Related Reactor Component Specifications and Core Symmetry Options

Several non-fuel-related reactor components must be specified in the MCNP input decks for the CRC evaluations. The definitions of these reactor components, along with many subsequent component definitions, depends on the core symmetry option being utilized in the calculation. The MACE input deck contains an option for utilizing either one-eighth, one-quarter, or full-core symmetry in the MCNP reactor model. Figures 3.6.3-1, 3.6.3-2, and 3.6.3-3 show a radial view of the one-eighth, quarter, and full-core models, respectively, for a B&W reactor. Figures 3.6.3-4, 3.6.3-5, and 3.6.3-6 show a radial view of the one-eighth, quarter, and full-core models, respectively, for a Westinghouse reactor.

Figures 3.6.3-1 through 3.6.3-6 show radial views of the non-fuel related reactor components that are defined by MACE. Figure 3.6.3-7 shows a typical axial view of the non-fuel related reactor components for a B&W reactor. Figure 3.6.3-8 shows a typical axial view of the non-fuel related reactor components for a Westinghouse reactor. The axial view of the non-fuel related reactor components for a Westinghouse reactor is similar. The number of upper and lower core regions is determined by the user in the MACE input deck. The non-fuel related reactor components that are defined by MACE in the MCNP models include the following:

- Pressure vessel;
- Pressure vessel cladding;
- Moderator between the pressure vessel cladding and the thermal shield or neutron pad;
- Thermal shield or neutron pad;
- Moderator between the thermal shield or neutron pad and the core barrel;
- Core barrel;
- Core lattice window defined inside the core barrel, above the bottom of the lower end-fittings, and below the top of the reactor;
- Lower core regions below the core lattice window;
- A zero importance outside world beyond the pressure vessel, above the reactor, and below the lowest core region.

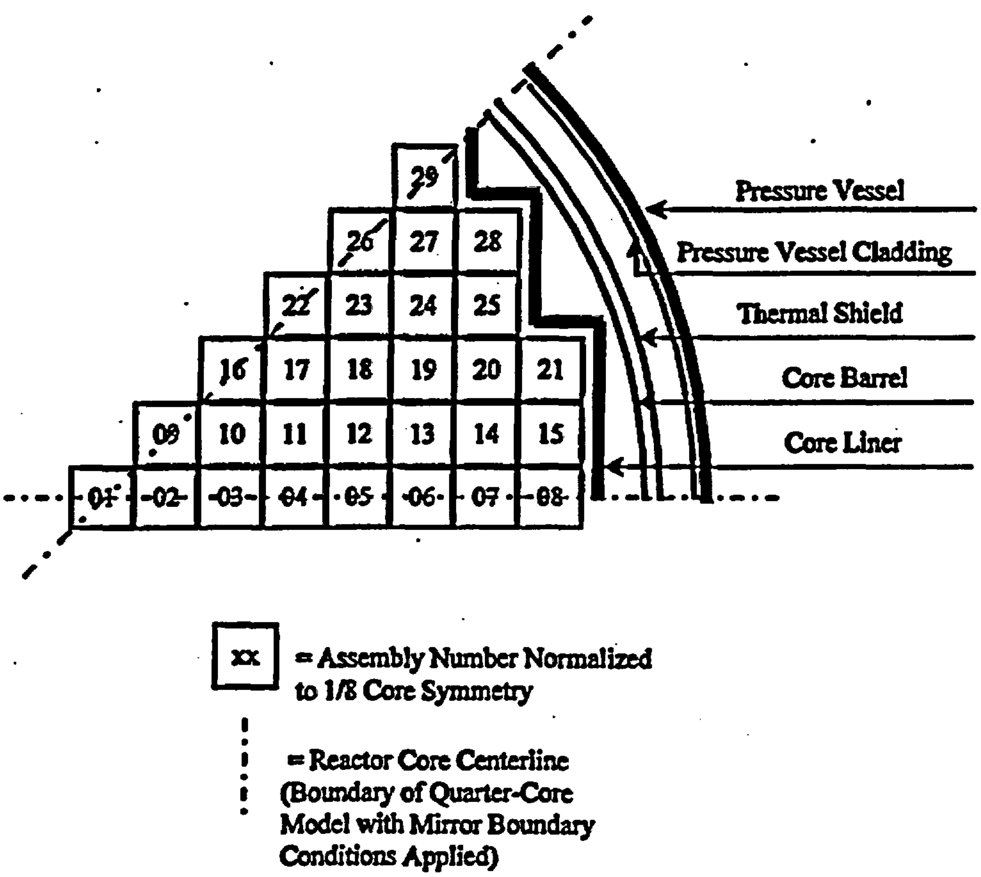


Figure 3.6.3-1. One-Eighth-Core Radial View of the MCNP Model for a B&W Reactor (This sketch is not to scale.)

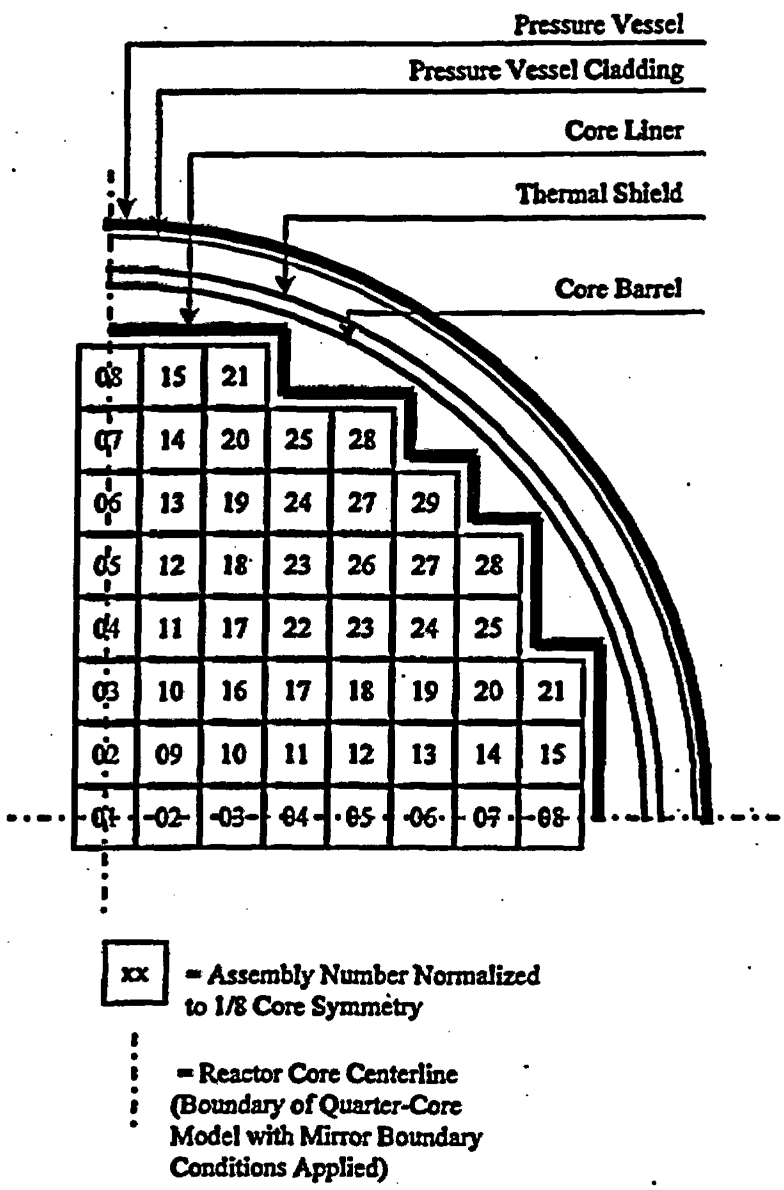


Figure 3.6.3-2. One-Quarter-Core Radial View of the MCNP Model for a B&W Reactor (This sketch is not to scale.)

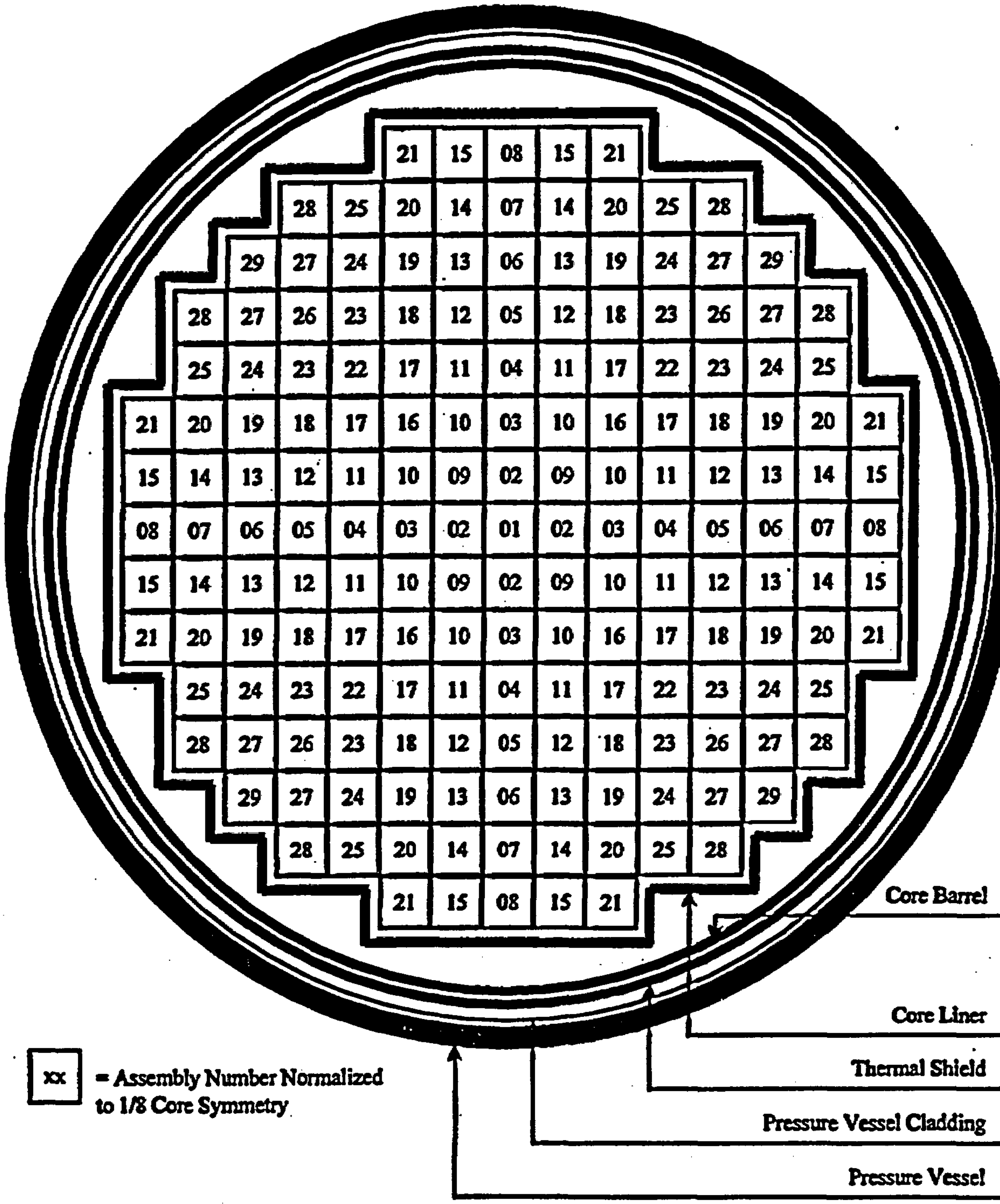


Figure 3.6.3-3. Full-Core Radial View of the MCNP Model for a B&W Reactor
(This sketch is not to scale.)

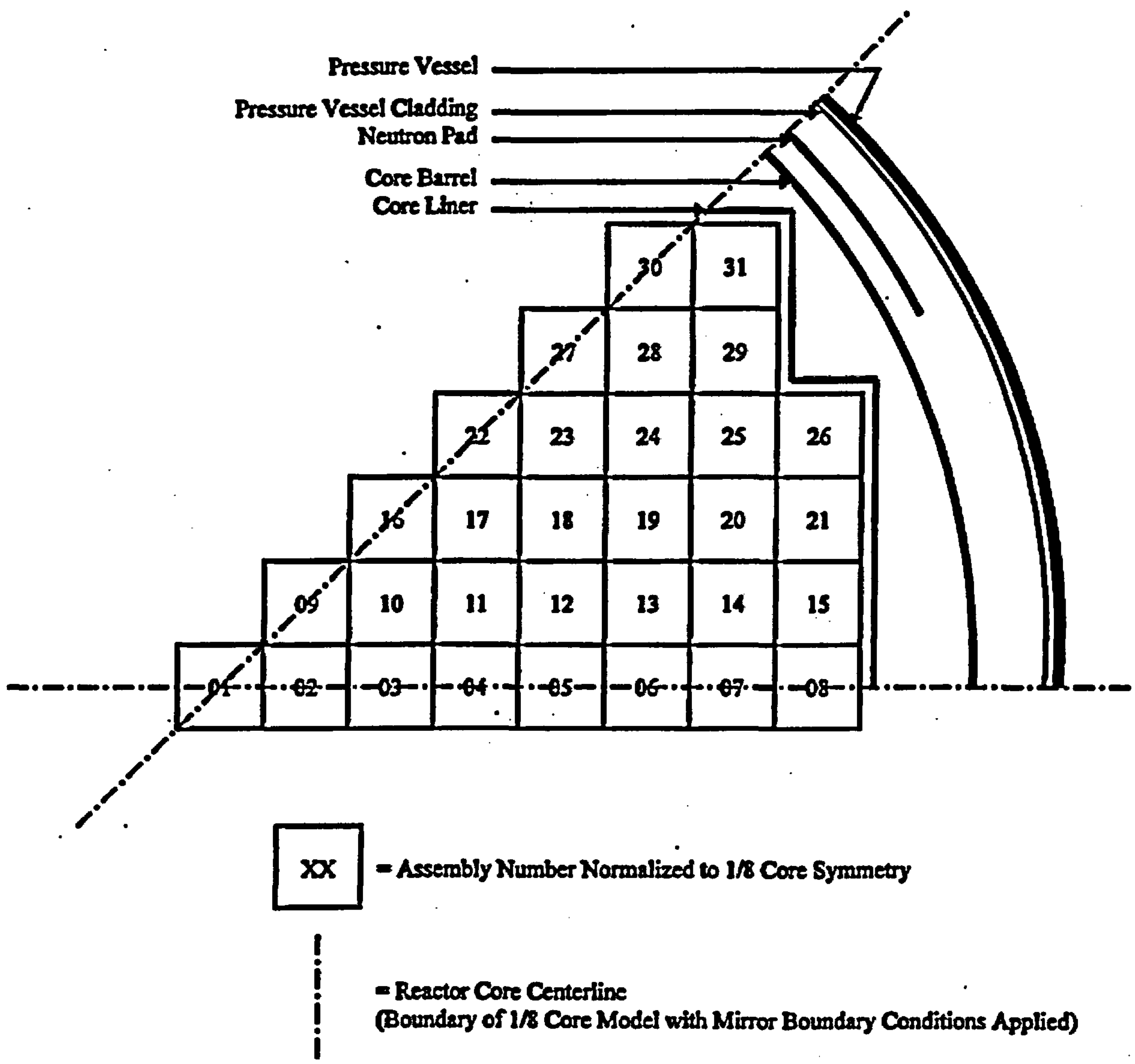


Figure 3.6.3-4. One-Eighth-Core Radial View of the MCNP Model for a Westinghouse Reactor (This sketch is not to scale.)

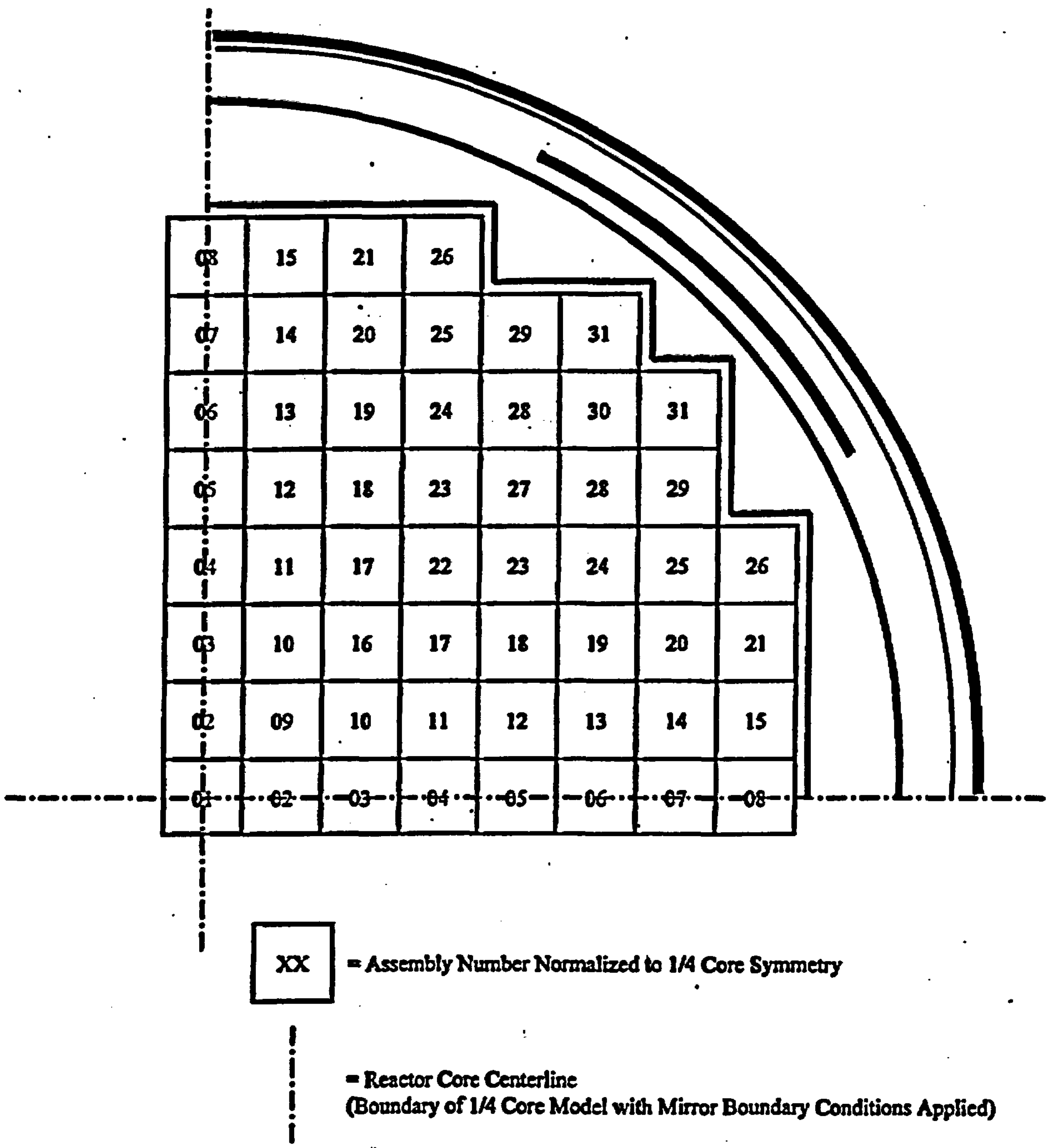


Figure 3.6.3-5. One-Quarter-Core Radial View of the MCNP Model for a Westinghouse Reactor (This sketch is not to scale.)

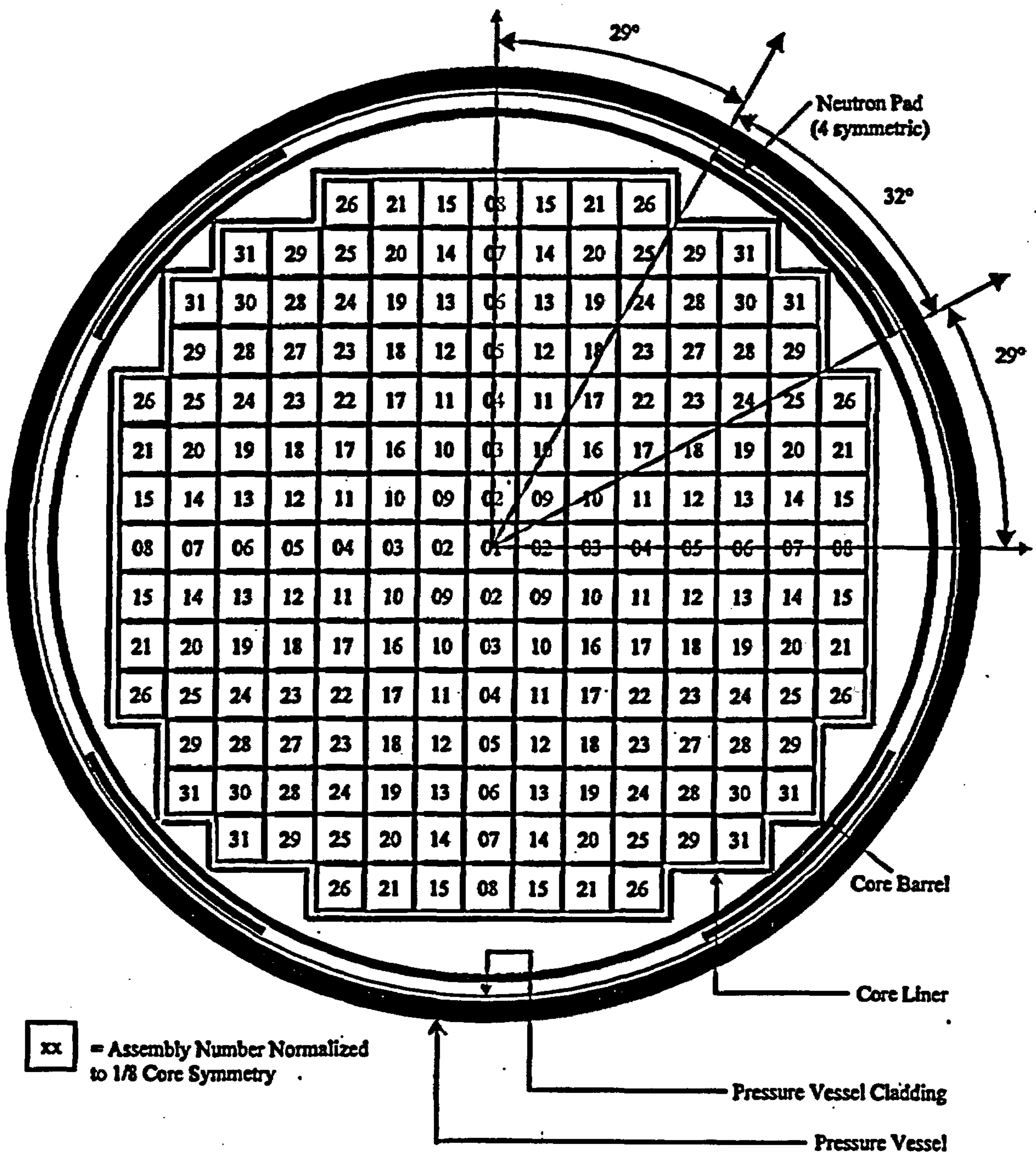


Figure 3.6.3-6. Full-Core Radial View of the MCNP Model for a Westinghouse Reactor (This sketch is not to scale.)

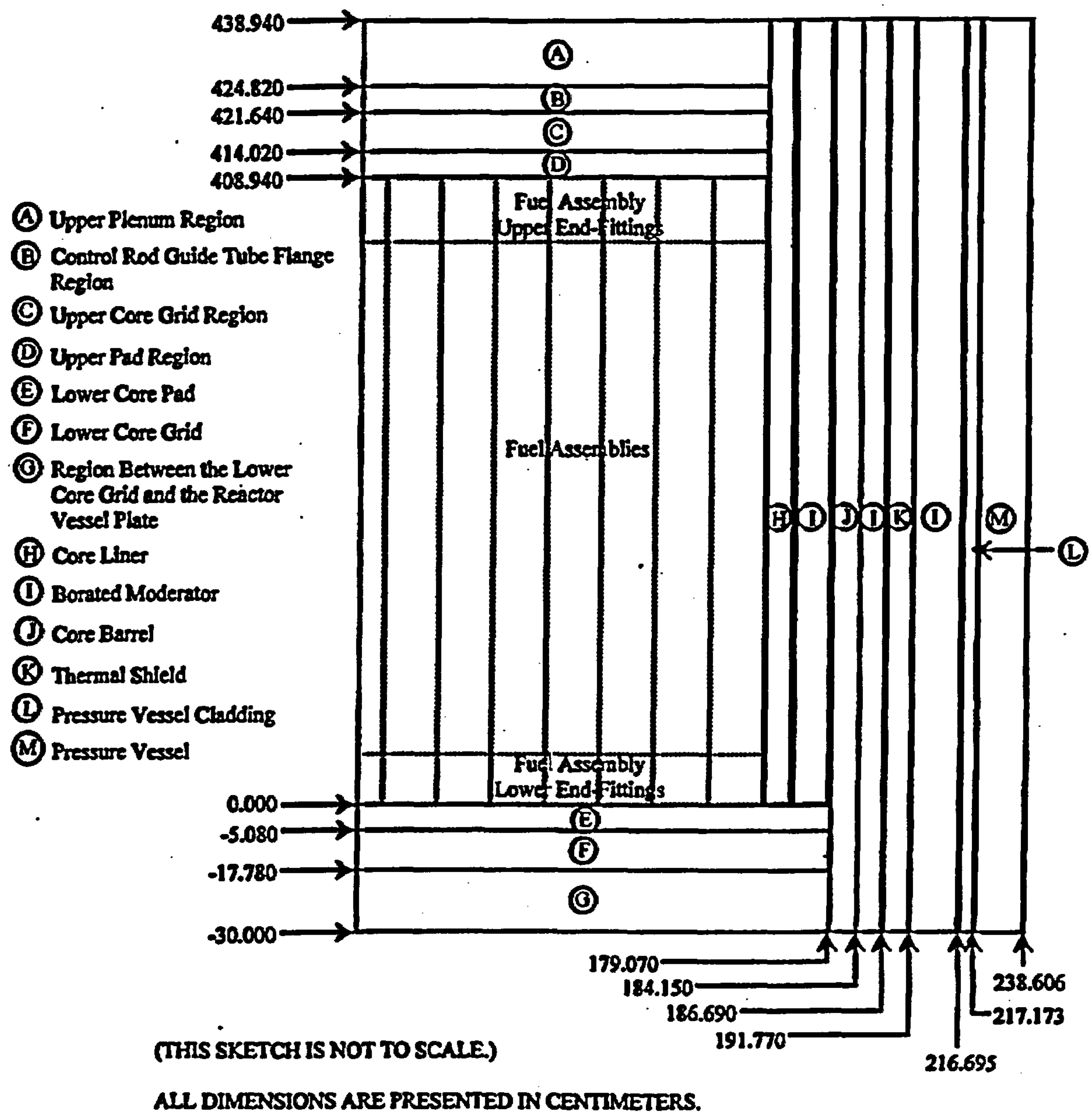
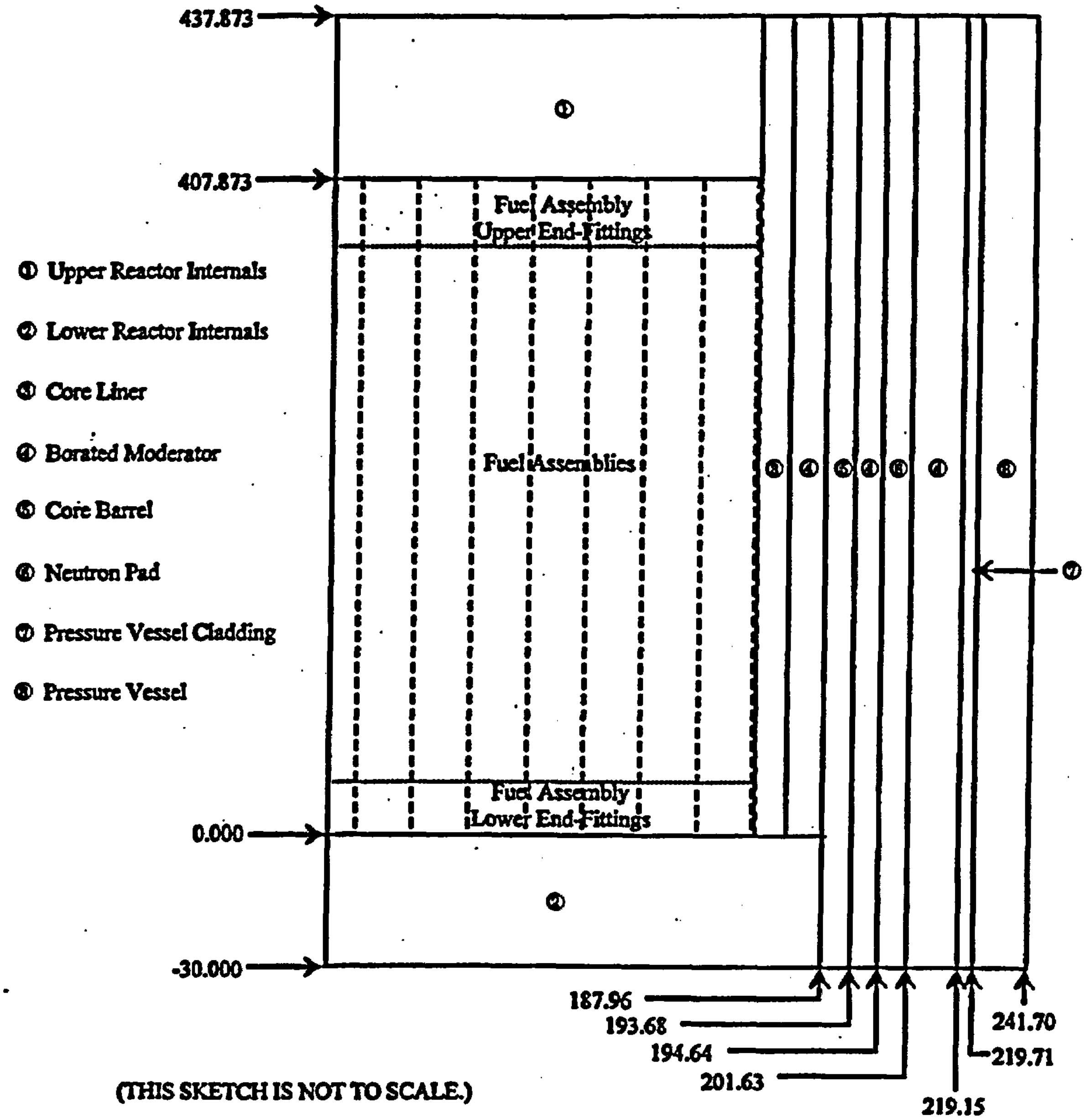


Figure 3.6.3-7. Axial View Along the Core Flat of the MCNP Model for a B&W Reactor (The dimensions presented in this figure are nominal dimensions for a B&W reactor core.)



(THIS SKETCH IS NOT TO SCALE.)

ALL DIMENSIONS ARE PRESENTED IN CENTIMETERS.

Figure 3.6.3-8. Axial View Along the Core Flat of the MCNP Model for a Westinghouse Reactor (The dimensions presented in this figure are nominal dimensions for a Westinghouse reactor core.)

The material specifications for all of the non-fuel-related reactor components must be defined in the MACE input deck by the user. The following set of equations are used to calculate the borated moderator composition. The atomic weight ratio values for hydrogen, oxygen, boron-10, and boron-11 are obtained from Table 3.5.2-1. The atomic weight ratio for natural boron is 10.718156 (Ref. 4).

Equation 3.6.3-1.

$$Boron\ wt\% = \frac{(Boron\ ppm)(1.0E-4)}{1+\left[(Boron\ ppm)(1.0E-6)\right]}$$

where, "ppm" means parts per million by mass of moderator.

Equation 3.6.3-2.

$$B-10\ wt\% = \frac{(B-10\ atom\%\ in\ B)(B-10\ Atomic\ Wt.\ Ratio)}{(B\ Atomic\ Wt.\ Ratio)(100.0)}(B\ wt\%)$$

where, "B" refers to natural boron.

Equation 3.6.3-3.

$$B-11\ wt\% = \frac{(B-11\ atom\%\ in\ B)(B-11\ Atomic\ Wt.\ Ratio)}{(B\ Atomic\ Wt.\ Ratio)(100.0)}(B\ wt\%)$$

where, "B" refers to natural boron.

Equation 3.6.3-4.

$$Hydrogen\ wt\% = \frac{(H\ Atomic\ Wt.\ Ratio)(2)(100.0-B\ wt\%)}{[(H\ Atomic\ Wt.\ Ratio)(2)+(O\ Atomic\ Wt.\ Ratio)]}$$

where, "H" refers to hydrogen, "B" refers to natural boron, and "O" refers to oxygen.

Equation 3.6.3-5.

$$Oxygen\ wt\% = \frac{(O\ Atomic\ Wt.\ Ratio)(100.0-B\ wt\%)}{[(H\ Atomic\ Wt.\ Ratio)(2)+(O\ Atomic\ Wt.\ Ratio)]}$$

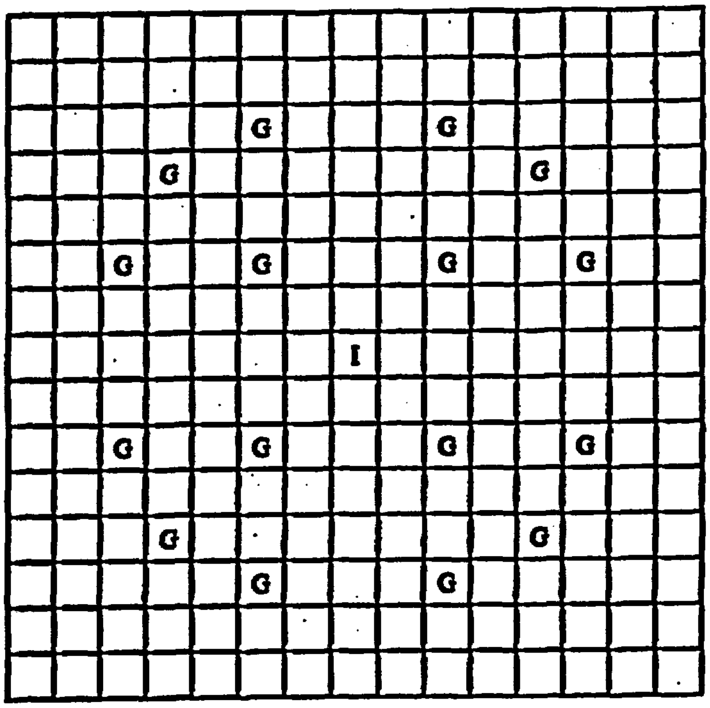
where, "H" refers to hydrogen, "B" refers to natural boron, and "O" refers to oxygen.

3.6.4. Core Assembly Lattice Layout Specification

MACE defines a fuel assembly lattice layout for the core, based on information provided in the MACE input deck. This core lattice layout is a two-dimensional array containing fuel assembly universe identifiers, core liner plate segment universe identifiers, and lattice cells filled with water. A unique universe identifier is assigned to each unique fuel assembly. A unique fuel assembly is delineated by the combination of a new fuel assembly identifier (one that has not been previously defined) and its insertion assembly (RCCA, BPRA, or APSRA) identifier. RCCA and APSRA insertion assemblies that are positioned at a height that has not been previously defined for that assembly type must be treated as a new insertion assembly when creating the MACE input deck. Each BPRA and fuel assembly combination must be given a unique universe identifier. The core liner segments are defined as MCNP universes that are positioned in the core lattice layout in a jigsaw puzzle fashion to define the entire core liner surrounding the outer fuel assemblies. The core lattice cells beyond the cells containing either fuel assembly or core liner universe identifiers are given the universe designation for the core lattice layout itself. Since the background material specification in the core lattice layout is defined as borated water, the cells containing the universe designation for the core lattice layout will contain borated water.

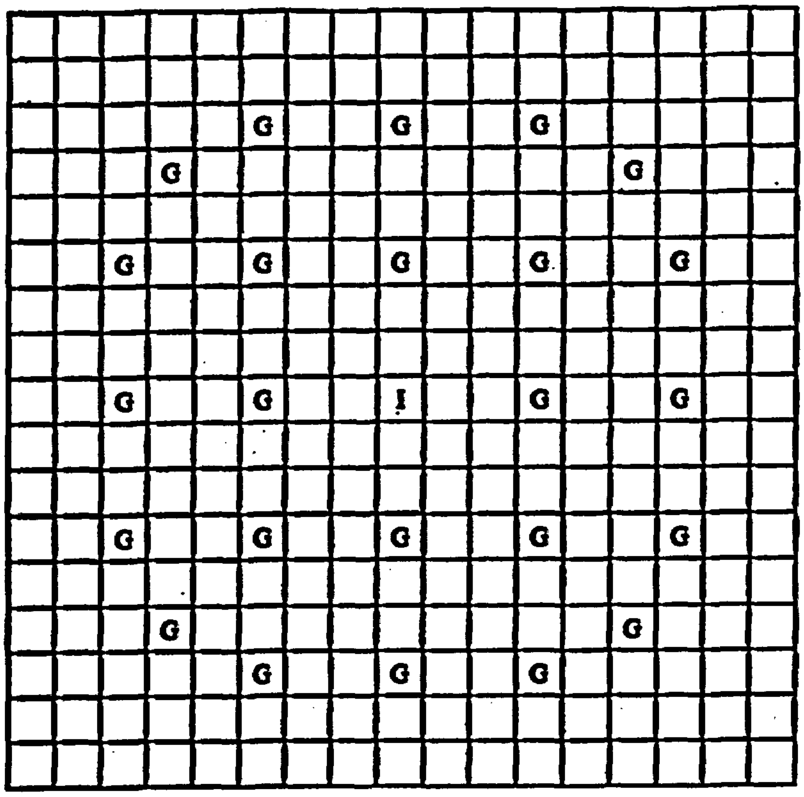
3.6.5. Fuel Assembly Lattice Layout Specifications

A fuel assembly lattice is defined by MACE for each fuel assembly with a unique identifier as provided in the MACE input deck. The fuel assembly lattices are specified as lattices of rod universes. The rod universes used to define the fuel assembly lattices may be fuel rods, water-filled guide tubes, guide tubes containing a control rod, guide tubes containing a burnable poison rod, guide tubes containing an axial power shaping rod, or water-filled instrument tubes. The MACE Version 3 software routine has the capability to model both 15x15 and 17x17 lattice fuel assemblies. In the MCNP model, each fuel assembly lattice specification is surrounded by two rows of lattice cells that are filled with water. This is an MCNP modeling technique that helps to prevent undersized filling universes when filling a window. The general 15x15 fuel assembly arrangement is shown in Figure 3.6.5-1. The general 17x17 fuel assembly arrangement is shown in Figure 3.6.5-2. The guide tubes may be filled with rods from an insertion (RCCA, BPRA, APSRA) assembly as previously described. The assembly dimensions must be provided by the user in the MACE input deck.



G = Guide Tube Location I = Instrument Tube Location = Fuel Rod Location

Figure 3.6.5-1. General 15x15 Lattice Fuel Assembly Arrangement



G = Guide Tube Location I = Instrument Tube Location [ ] = Fuel Rod Location

Figure 3.6.5-2. General 17x17 Lattice Fuel Assembly Arrangement

3.6.6. Fuel Assembly Spacer Grid Specifications

MACE allows the specification of a user-defined number of spacer grids in a fuel assembly. The top spacer grid location is fixed below the fuel assembly upper end-fitting, but its height may be specified by the user. For the other spacer grids, the location and height of each grid must be provided by the user in the MACE input deck. Figure 3.6.6-1 shows an axial view of a typical B&W 15x15 fuel assembly as modeled by MACE. The spacer grid material is homogenized with the borated moderator within each defined spacer grid region. The user must specify either Zircaloy-4, Stainless Steel 304, Inconel, Stainless Steel 304/Inconel mixture, or Stainless Steel 304/Zircaloy-4 mixture as the material for the spacer grids. The volume of spacer grid material must be provided for each spacer grid homogenzied region. If either the Stainless Steel 304/Inconel mixture or Stainless Steel 304/Zircaloy-4 mixture in chosen, the volume fractions of Stainless Steel 304 and either Inconel or Zircaloy-4 in the spacer grid material must be provided. The following set of equations is used to define the homogenized spacer grid material compositions in the MCNP input deck.

Equation 3.6.6-1.

$$VAL1 = (Assembly\ Pitch\ (cm))^2 (Spacer\ Height\ (cm))$$

Equation 3.6.6-2.

$$VAL2 = (Number\ of\ Fuel\ Rods)(\frac{\pi}{4})(Fuel\ Rod\ Clad\ OD\ (cm))^2 * (Spacer\ Height\ (cm))$$

where, "OD" refers to outer diameter.

Equation 3.6.6-3.

$$VAL3 = (Number\ of\ GuideTubes)(Guide\ Tube\ OD\ (cm))^2 (\frac{\pi}{4})(Spacer\ Height\ (cm))$$

Equation 3.6.6-4.

$$VAL4 = (Instrument\ Tube\ OD\ (cm))^2 (\frac{\pi}{4})(Spacer\ Height\ (cm))$$

Equation 3.6.6-5.

$$\begin{matrix}\textit{Total Volume of Homogenized} \\ \textit{Spacer/Moderator Material for a} \\ \textit{Single Spacer Grid Region}\ (cm^3)(TVOL)\end{matrix} = VAL1 - VAL2 - VAL3 - VAL4$$

Equation 3.6.6-6.

$$\begin{matrix}\textit{Moderator Volume in Homogenized} \\ \textit{Spacer Region}\ (cm^3)\ (MODVOL)\end{matrix} = TVOL - \textit{Spacer Volume}\ (cm^3)$$

where, "Spacer Volume" is provide by the user in the MACE input deck.

Equation 3.6.6-7.

$$\begin{matrix}\textit{Homogenized} \\ \textit{Spacer} \\ \textit{Region} \\ \textit{Density}\end{matrix}\left(\frac{g}{cm^3}\right) = \frac{\left[\begin{matrix}(\textit{Spacer Volume}(cm^3))* \\ (\textit{Spacer Material Density}(\frac{g}{cm^3})) + \\ (MODVOL)(\textit{Moderator Density}(\frac{g}{cm^3}))\end{matrix}\right]}{TVOL}$$

where, "Spacer Material Density" is 6.56 g/cm^3 for Zircaloy-4 (Ref. 7), 7.90 g/cm^3 for Stainless Steel 304 (Ref. 7), 8.19 g/cm^3 for Inconel (Ref. 8), ((7.90 g/cm^3)*(SS304 volume fraction in grid))+((8.19 g/cm^3)*(Inconel volume fraction in grid)) for the SS304/Inconel mixture, and ((7.90 g/cm^3)*(SS304 volume fraction in grid))+((6.56 g/cm^3)*(Zircaloy-4 volume fraction in grid)) for the SS304/Zircaloy-4 mixture.

Equation 3.6.6-8.

$$\begin{matrix}\textit{Grams of Spacer in Homogenized} \\ \textit{Spacer Region}\ (SPACMASS)\end{matrix} = (\textit{Spacer Volume}\ (cm^3))(\textit{Spacer Density}\ (\frac{g}{cm^3}))$$

where, "Spacer Density" is 6.56 g/cm^3 for Zircaloy-4 (Ref. 7), 7.90 g/cm^3 for Stainless Steel 304 (Ref. 7), or 8.19 g/cm^3 for Inconel (Ref. 8).

Equation 3.6.6-9.

$$\begin{array}{l}\textit{Grams of Moderator}\\ \textit{in Homogenized Spacer Region (MODMASS)}\end{array} = (MODVOL\,(cm^3))(Moderator\ Density\,(\frac{g}{cm^3}))$$

where, the moderator density is calculated by the MODDEN subroutine.

Equation 3.6.6-10.

$$\begin{array}{c}\textit{Spacer Material Mass Fraction}\\ \textit{in Homogenized Spacer Region}\ (SPACFRAC)\end{array} = \frac{SPACMASS}{(SPACMASS + MODMASS)}$$

Equation 3.6.6-11.

$$\begin{array}{c}\textit{Moderator Mass Fraction}\\ \textit{in Homogenized Spacer Region}\ (MODFRAC)\end{array} = \frac{MODMASS}{(SPACMASS + MODMASS)}$$

Equation 3.6.6-12. (for Stainless Steel 304 and either Inconel or Zircaloy-4 mixtures)

$$\left(\begin{array}{c}\textit{Stainless Steel Mass Fraction}\\ \textit{in Homogenzied Spacer Region}\ (SSFRAC)\end{array}\right) = \left(\frac{\textit{Stainless Steel Mass in Grid}}{(\textit{Total Grid Material Mass})}\right)$$

Equation 3.6.6-13. (for Stainless Steel 304 and Inconel mixture)

$$\left(\begin{array}{c}\textit{Inconel Mass Fraction}\\ \textit{in Homogenzied Spacer Region}\ (INCFRAC)\end{array}\right) = \left(\frac{\textit{Inconel Mass in Grid}}{(\textit{Total Grid Material Mass})}\right)$$

Equation 3.6.6-14. (for Stainless Steel 304 and Zircaloy-4 mixture)

$$\left(\begin{array}{c}\textit{Zircaloy Mass Fraction}\\ \textit{in Homogenzied Spacer Region}\ (ZRFRAC)\end{array}\right) = \left(\frac{\textit{Zircaloy Mass in Grid}}{(\textit{Total Grid Material Mass})}\right)$$

Equation 3.6.6-15.

$$\textit{Natural Boron Wt.\% in Borated Moderator (BINMOD)} = \frac{(ppmb)(1E-6)(100)}{1+(ppmb)(1E-6)}$$

where, "ppmb" refers to parts per million of natural boron by mass of moderator.

Equation 3.6.6-16.

$$\begin{matrix}\textit{Hydrogen Wt.\%} \\ \textit{in Borated Moderator}\end{matrix} = \frac{(0.999167)(2)(100 - BINMOD)}{(0.999167)(2) + (15.857510)}$$

Equation 3.6.6-17.

$$\begin{matrix}\textit{Oxygen Wt.\%} \\ \textit{in Borated Moderator}\end{matrix} = \frac{(15.857510)(100 - BINMOD)}{(0.999167)(2) + (15.857510)}$$

Equation 3.6.6-18.

$$\begin{matrix}B^{10}\textit{ Wt.\% in} \\ \textit{Borated Moderator}\end{matrix} = \frac{(9.926922)(0.194)(BINMOD)}{(9.926922)(0.194) + (10.914730)(0.806)}$$

Equation 3.6.6-19.

$$\begin{matrix}B^{11}\textit{ Wt.\% in} \\ \textit{Borated Moderator}\end{matrix} = \frac{(10.914730)(0.806)(BINMOD)}{(9.926922)(0.194) + (10.914730)(0.806)}$$

Equation 3.6.6-20.

This equation is actually a set of equations used to define a homogenized spacer/moderator material specification for a Zircaloy-4 grid.

$$O\ Wt.\ \%\ in\ Mixture = \frac{(O\ Wt.\ \%\ in\ Moderator)(MODFRAC)\ +}{(0.120)(SPACFRAC)}$$

$$H\ Wt.\ \%\ in\ Mixture = (H\ Wt.\ \%\ in\ Moderator)(MODFRAC)$$

$$B^{10}\ Wt.\ \%\ in\ Mixture = (B^{10}\ Wt.\ \%\ in\ Moderator)(MODFRAC)$$

$$B^{11}\ Wt.\ \%\ in\ Mixture = (B^{11}\ Wt.\ \%\ in\ Moderator)(MODFRAC)$$

$$Cr^{50}\ Wt.\ \%\ in\ Mixture = (0.100)(SPACFRAC)(0.04173684)$$

$$Cr^{52}\ Wt.\ \%\ in\ Mixture = (0.100)(SPACFRAC)(0.83700000)$$

$$Cr^{53}\ Wt.\ \%\ in\ Mixture = (0.100)(SPACFRAC)(0.09673684)$$

$$Cr^{54}\ Wt.\ \%\ in\ Mixture = (0.100)(SPACFRAC)(0.02452632)$$

$$Fe^{54}\ Wt.\ \%\ in\ Mixture = (0.200)(SPACFRAC)(0.05699324)$$

$$Fe^{56}\ Wt.\ \%\ in\ Mixture = (0.200)(SPACFRAC)(0.91868499)$$

$$Fe^{57}\ Wt.\ \%\ in\ Mixture = (0.200)(SPACFRAC)(0.02141247)$$

$$Fe^{58}\ Wt.\ \%\ in\ Mixture = (0.200)(SPACFRAC)(0.00290930)$$

$$Zr\ Wt.\ \%\ in\ Mixture = (98.180)(SPACFRAC)$$

$$Sn\ Wt.\ \%\ in\ Mixture = (1.400)(SPACFRAC)$$

where, O=oxygen, H=hydrogen, B=boron, Cr=chromium, Fe=iron, Zr=zirconium, and Sn=tin. In Equation 3.6.6-17 and subsequent similar equations, the values appearing before the SPACFRAC value in the wt% calculation for isotopes like Cr-50 represent the weight percent of the element in the material composition. The value after the SPACFRAC value represents the mass fraction of the isotope in its elemental composition. The weight percent values for Zircaloy-4 are obtained from Reference 7. The mass fraction values are obtained from data in Table 3.5.2-1.

Equation 3.6.6-21.

This equation is actually a set of equations used to define a homogenized spacer/moderator material specification for a Stainless Steel 304 grid.

$$O\ Wt.\ \%\ in\ Mixture = (O\ Wt.\ \%\ in\ Moderator)(MODFRAC)$$
$$H\ Wt.\ \%\ in\ Mixture = (H\ Wt.\ \%\ in\ Moderator)(MODFRAC)$$
$$B^{10}\ Wt.\ \%\ in\ Mixture = (B^{10}\ Wt.\ \%\ in\ Moderator)(MODFRAC)$$
$$B^{11}\ Wt.\ \%\ in\ Mixture = (B^{11}\ Wt.\ \%\ in\ Moderator)(MODFRAC)$$
$$Cr^{50}\ Wt.\ \%\ in\ Mixture = (19.0)(SPACFRAC)(0.04173684)$$
$$Cr^{52}\ Wt.\ \%\ in\ Mixture = (19.0)(SPACFRAC)(0.83700000)$$
$$Cr^{53}\ Wt.\ \%\ in\ Mixture = (19.0)(SPACFRAC)(0.09673684)$$
$$Cr^{54}\ Wt.\ \%\ in\ Mixture = (19.0)(SPACFRAC)(0.02452632)$$
$$N\ Wt.\ \%\ in\ Mixture = (0.10)(SPACFRAC)$$
$$Si\ Wt.\ \%\ in\ Mixture = (0.75)(SPACFRAC)$$
$$P\ Wt.\ \%\ in\ Mixture = (0.045)(SPACFRAC)$$
$$S\ Wt.\ \%\ in\ Mixture = (0.03)(SPACFRAC)$$
$$C\ Wt.\ \%\ in\ Mixture = (0.08)(SPACFRAC)$$
$$Mn\ Wt.\ \%\ in\ Mixture = (2.0)(SPACFRAC)$$
$$Fe^{54}\ Wt.\ \%\ in\ Mixture = (68.745)(SPACFRAC)(0.05699324)$$
$$Fe^{56}\ Wt.\ \%\ in\ Mixture = (68.745)(SPACFRAC)(0.91868499)$$
$$Fe^{57}\ Wt.\ \%\ in\ Mixture = (68.745)(SPACFRAC)(0.02141247)$$
$$Fe^{58}\ Wt.\ \%\ in\ Mixture = (68.745)(SPACFRAC)(0.00290930)$$
$$Ni^{58}\ Wt.\ \%\ in\ Mixture = (9.250)(SPACFRAC)(0.67394595)$$
$$Ni^{60}\ Wt.\ \%\ in\ Mixture = (9.250)(SPACFRAC)(0.26648649)$$
$$Ni^{61}\ Wt.\ \%\ in\ Mixture = (9.250)(SPACFRAC)(0.01178378)$$
$$Ni^{62}\ Wt.\ \%\ in\ Mixture = (9.250)(SPACFRAC)(0.03783784)$$
$$Ni^{64}\ Wt.\ \%\ in\ Mixture = (9.250)(SPACFRAC)(0.00994594)$$

where, O=oxygen, H=hydrogen, B=boron, Cr=chromium, N=nitrogen, Si=silicon, P=phosphorous, S=sulfur, C=carbon, Mn=manganese, Fe=iron, and Ni=nickel.

The weight percent values for Stainless Steel 304 are obtained from Reference 7. The mass fraction values are obtained from data in Table 3.5.2-1.

Equation 3.6.6-22.

This equation is actually a set of equations used to define a homogenized spacer/moderator material specification for an Inconel grid.

$$O\ Wt.\ \%\ in\ Mixture = (O\ Wt.\ \%\ in\ Moderator)(MODFRAC)$$

$$H\ Wt.\ \%\ in\ Mixture = (H\ Wt.\ \%\ in\ Moderator)(MODFRAC)$$

$$B^{10}\ Wt.\ \%\ in\ Mixture = \begin{matrix}(B^{10}\ Wt.\ \%\ in\ Moderator)(MODFRAC) + \\ (B^{10}\ Wt.\ \%\ in\ B)(0.00006)(SPACFRAC)\end{matrix}$$

$$B^{11}\ Wt.\ \%\ in\ Mixture = \begin{matrix}(B^{11}\ Wt.\ \%\ in\ Moderator)(MODFRAC) + \\ (B^{11}\ Wt.\ \%\ in\ B)(0.00006)(SPACFRAC)\end{matrix}$$

$$Cr^{50}\ Wt.\ \%\ in\ Mixture = (19.0)(SPACFRAC)(0.04173684)$$

$$Cr^{52}\ Wt.\ \%\ in\ Mixture = (19.0)(SPACFRAC)(0.83700000)$$

$$Cr^{53}\ Wt.\ \%\ in\ Mixture = (19.0)(SPACFRAC)(0.09673684)$$

$$Cr^{54}\ Wt.\ \%\ in\ Mixture = (19.0)(SPACFRAC)(0.02452632)$$

$$Si\ Wt.\ \%\ in\ Mixture = (035)(SPACFRAC)$$

$$P\ Wt.\ \%\ in\ Mixture = (0.015)(SPACFRAC)$$

$$S\ Wt.\ \%\ in\ Mixture = (0.015)(SPACFRAC)$$

$$C\ Wt.\ \%\ in\ Mixture = (0.08)(SPACFRAC)$$

$$Mn\ Wt.\ \%\ in\ Mixture = (0.35)(SPACFRAC)$$

$$Fe^{54}\ Wt.\ \%\ in\ Mixture = (16.809)(SPACFRAC)(0.05699324)$$

$$Fe^{56}\ Wt.\ \%\ in\ Mixture = (16.809)(SPACFRAC)(0.91868499)$$

$$Fe^{57}\ Wt.\ \%\ in\ Mixture = (16.809)(SPACFRAC)(0.02141247)$$

$$Fe^{58}\ Wt.\ \%\ in\ Mixture = (16.809)(SPACFRAC)(0.00290930)$$

$$Ni^{58}\ Wt.\ \%\ in\ Mixture = (52.5)(SPACFRAC)(0.67394595)$$

$$Ni^{60}\ Wt.\ \%\ in\ Mixture = (52.5)(SPACFRAC)(0.26648649)$$

$$Ni^{61}\ Wt.\ \%\ in\ Mixture = (52.5)(SPACFRAC)(0.01178378)$$

$$Ni^{62}\ Wt.\ \%\ in\ Mixture = (52.5)(SPACFRAC)(0.03783784)$$

$$Ni^{64}\ Wt.\ \%\ in\ Mixture = (52.5)(SPACFRAC)(0.00994594)$$

$$Al\ Wt.\ \%\ in\ Mixture = (0.50)(SPACFRAC)$$

$$Ti\ Wt.\ \%\ in\ Mixture = (0.90)(SPACFRAC)$$

$$Co\ Wt.\ \%\ in\ Mixture = (1.00)(SPACFRAC)$$

$$Cu^{63}\ Wt.\ \%\ in\ Mixture = (0.30)(SPACFRAC)(0.683)$$

$$Cu^{65}\ Wt.\ \%\ in\ Mixture = (0.30)(SPACFRAC)(0.317)$$

$$Nb\ Wt.\ \%\ in\ Mixture = (2.5625)(SPACFRAC)$$

$$Mo\ Wt.\ \%\ in\ Mixture = (3.050)(SPACFRAC)$$

$$Ta\ Wt.\ \%\ in\ Mixture = (2.5625)(SPACFRAC)$$

where, O=oxygen, H=hydrogen, B=boron, Cr=chromium, Si=silicon, P=phosphorous, S=sulfur, C=carbon, Mn=manganese, Fe=iron, Ni=nickel, Al=aluminum, Ti=titanium, Co=cobalt, Cu=copper, Nb=niobium, Mo=molybdenum, and Ta=tantalum.

The weight percent values for Inconel are obtained from Reference 8. The mass fraction values are obtained from data in Table 3.5.2-1.

Equation 3.6.6-23.

This equation is actually a set of equations used to define a homogenized spacer/moderator material specification for a Stainless Steel 304 and Inconel mixture grid.

$$O\ Wt.\%\ in\ Mixture = (O\ Wt.\%\ in\ Moderator)(MODFRAC)$$
$$H\ Wt.\%\ in\ Mixture = (H\ Wt.\%\ in\ Moderator)(MODFRAC)$$
$$B^{10}\ Wt.\%\ in\ Mixture = (B^{10}\ Wt.\%\ in\ Moderator)(MODFRAC) + (B^{10}\ Wt.\%\ in\ Boron)(0.00006)(INCFRAC)$$
$$B^{11}\ Wt.\%\ in\ Mixture = (B^{11}\ Wt.\%\ in\ Moderator)(MODFRAC) + (B^{11}\ Wt.\%\ in\ Boron)(0.00006)(INCFRAC)$$
$$C\ Wt.\%\ in\ Mixture = (0.080)(INCFRAC) + (0.080)(SSFRAC)$$
$$N\ Wt.\%\ in\ Mixture = (0.100)(SSFRAC)$$
$$Si\ Wt.\%\ in\ Mixture = (0.350)(INCFRAC) + (0.750)(SSFRAC)$$
$$P\ Wt.\%\ in\ Mixture = (0.015)(INCFRAC) + (0.045)(SSFRAC)$$
$$S\ Wt.\%\ in\ Mixture = (0.015)(INCFRAC) + (0.030)(SSFRAC)$$
$$Cr^{50}\ Wt.\%\ in\ Mixture = (0.04173684)[(19.0)(INCFRAC) + (19.0)(SSFRAC)]$$
$$Cr^{52}\ Wt.\%\ in\ Mixture = (0.837)[(19.0)(INCFRAC) + (19.0)(SSFRAC)]$$
$$Cr^{53}\ Wt.\%\ in\ Mixture = (0.09673684)[(19.0)(INCFRAC) + (19.0)(SSFRAC)]$$
$$Cr^{54}\ Wt.\%\ in\ Mixture = (0.02452632)[(19.0)(INCFRAC) + (19.0)(SSFRAC)]$$
$$Mn\ Wt.\%\ in\ Mixture = (0.350)(INCFRAC) + (2.0)(SSFRAC)$$
$$Fe^{54}\ Wt.\%\ in\ Mixture = (0.05699324)[(16.809)(INCFRAC) + (68.745)(SSFRAC)]$$
$$Fe^{56}\ Wt.\%\ in\ Mixture = (0.91868499)[(16.809)(INCFRAC) + (68.745)(SSFRAC)]$$
$$Fe^{57}\ Wt.\%\ in\ Mixture = (0.02141247)[(16.809)(INCFRAC) + (68.745)(SSFRAC)]$$
$$Fe^{58}\ Wt.\%\ in\ Mixture = (0.00290930)[(16.809)(INCFRAC) + (68.745)(SSFRAC)]$$
$$Ni^{58}\ Wt.\%\ in\ Mixture = (0.67394595)[(52.50)(INCFRAC) + (9.25)(SSFRAC)]$$
$$Ni^{60}\ Wt.\%\ in\ Mixture = (0.26648649)[(52.50)(INCFRAC) + (9.25)(SSFRAC)]$$
$$Ni^{61}\ Wt.\%\ in\ Mixture = (0.01178378)[(52.50)(INCFRAC) + (9.25)(SSFRAC)]$$
$$Ni^{62}\ Wt.\%\ in\ Mixture = (0.03783784)[(52.50)(INCFRAC) + (9.25)(SSFRAC)]$$
$$Ni^{64}\ Wt.\%\ in\ Mixture = (0.00994594)[(52.50)(INCFRAC) + (9.25)(SSFRAC)]$$

$$Al\ Wt.\%\ in\ Mixture = (0.500)(INCFRAC)$$
$$Ti\ Wt.\%\ in\ Mixture = (0.900)(INCFRAC)$$
$$Co\ Wt.\%\ in\ Mixture = (1.000)(INCFRAC)$$
$$Cu^{63}\ Wt.\%\ in\ Mixture = (0.683)(0.300)(INCFRAC)$$
$$Cu^{65}\ Wt.\%\ in\ Mixture = (0.317)(0.300)(INCFRAC)$$
$$Nb\ Wt.\%\ in\ Mixture = (2.5625)(INCFRAC)$$
$$Mo\ Wt.\%\ in\ Mixture = (3.050)(INCFRAC)$$
$$Ta\ Wt.\%\ in\ Mixture = (2.5625)(INCFRAC)$$

where, O=oxygen, H=hydrogen, B=boron, N=nitrogen, Cr=chromium, Si=silicon, P=phosphorous, S=sulfur, C=carbon, Mn=manganese, Fe=iron, Ni=nickel, Al=aluminum, Ti=titanium, Co=cobalt, Cu=copper, Nb=niobium, Mo=molybdenum, and Ta=tantalum.

The weight percent values for Stainless Steel 304 are obtained from Reference 7. The weight percent values for Inconel are obtained from Reference 8. The mass fraction values are obtained from data in Table 3.5.2-1.

Equation 3.6.6-24.

This equation is actually a set of equations used to define a homogenized spacer/moderator material specification for a Stainless Steel 304 and Zircaloy-4 mixture grid.

$$O\ Wt.\%\ in\ Mixture = (O\ Wt.\%\ in\ Moderator)(MODFRAC) + (0.120)(ZRFRAC)$$
$$H\ Wt.\%\ in\ Mixture = (H\ Wt.\%\ in\ Moderator)(MODFRAC)$$
$$B^{10}\ Wt.\%\ in\ Mixture = (B^{10}\ Wt.\%\ in\ Moderator)(MODFRAC)$$
$$B^{11}\ Wt.\%\ in\ Mixture = (B^{11}\ Wt.\%\ in\ Moderator)(MODFRAC)$$
$$C\ Wt.\%\ in\ Mixture = (0.080)(SSFRAC)$$
$$N\ Wt.\%\ in\ Mixture = (0.100)(SSFRAC)$$
$$Si\ Wt.\%\ in\ Mixture = (0.750)(SSFRAC)$$
$$P\ Wt.\%\ in\ Mixture = (0.045)(SSFRAC)$$
$$S\ Wt.\%\ in\ Mixture = (0.030)(SSFRAC)$$
$$Cr^{50}\ Wt.\%\ in\ Mixture = (0.04173684)[(0.1)(ZRFRAC) + (19.0)(SSFRAC)]$$
$$Cr^{52}\ Wt.\%\ in\ Mixture = (0.837)[(0.1)(ZRFRAC) + (19.0)(SSFRAC)]$$
$$Cr^{53}\ Wt.\%\ in\ Mixture = (0.09673684)[(0.1)(ZRFRAC) + (19.0)(SSFRAC)]$$
$$Cr^{54}\ Wt.\%\ in\ Mixture = (0.02452632)[(0.1)(ZRFRAC) + (19.0)(SSFRAC)]$$
$$Mn\ Wt.\%\ in\ Mixture = (2.0)(SSFRAC)$$
$$Fe^{54}\ Wt.\%\ in\ Mixture = (0.05699324)[(0.2)(ZRFRAC) + (68.745)(SSFRAC)]$$
$$Fe^{56}\ Wt.\%\ in\ Mixture = (0.91868499)[(0.2)(ZRFRAC) + (68.745)(SSFRAC)]$$

$$Fe^{57}\ Wt.\%\ in\ Mixture = (0.02141247)[(0.2)(ZRFRAC) + (68.745)(SSFRAC)]$$
$$Fe^{58}\ Wt.\%\ in\ Mixture = (0.00290930)[(0.2)(ZRFRAC) + (68.745)(SSFRAC)]$$
$$Ni^{58}\ Wt.\%\ in\ Mixture = (0.67394595)[(9.25)(SSFRAC)]$$
$$Ni^{60}\ Wt.\%\ in\ Mixture = (0.26648649)[(9.25)(SSFRAC)]$$
$$Ni^{61}\ Wt.\%\ in\ Mixture = (0.01178378)[(9.25)(SSFRAC)]$$
$$Ni^{62}\ Wt.\%\ in\ Mixture = (0.03783784)[(9.25)(SSFRAC)]$$
$$Ni^{64}\ Wt.\%\ in\ Mixture = (0.00994594)[(9.25)(SSFRAC)]$$
$$Zr\ Wt.\%\ in\ Mixture = (98.180)(ZRFRAC)$$
$$Sn\ Wt.\%\ in\ Mixture = (1.40)(ZRFRAC)$$

where, O=oxygen, H=hydrogen, B=boron, N=nitrogen, Cr=chromium, Si=silicon, P=phosphorous, S=sulfur, C=carbon, Mn=manganese, Fe=iron, Ni=nickel, Zr=zirconium, Sn=tin.

The weight percent values for Stainless Steel 304 are obtained from Reference 7. The weight percent values for Zircaloy-4 are obtained from Reference 7. The mass fraction values are obtained from data in Table 3.5.2-1.

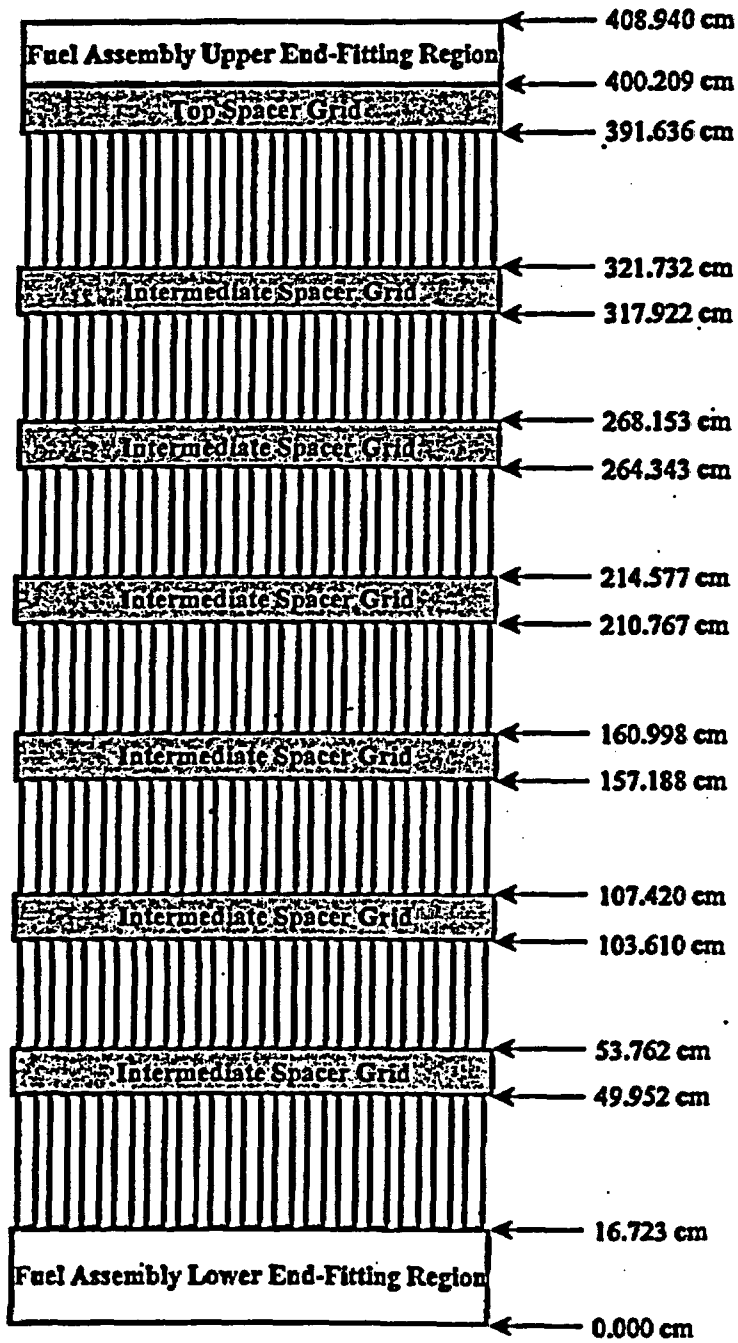


Figure 3.6.6-1. Axial View of a Typical B&W 15x15 Fuel Assembly as Modeled by MACE (This sketch is not to scale.)

3.6.7. Fuel Rod Universe Specifications

MACE creates a different fuel rod universe for each assembly bearing a unique identifier as specified in the MACE input deck. A unique fuel assembly identifier refers to a fuel assembly that contains a unique fuel composition. The fuel rod universe contains the definition of a complete fuel assembly lattice cell containing a fuel rod. The fuel rod, spacer grids, and fuel assembly end-fittings are all modeled in this universe. Additionally, the regions of the core above the upper fuel assembly end-fitting are modeled. The material composition of these upper core regions depends on the type of insertion assembly hardware present in the fuel assembly location. The fuel rod universe extends to infinity in the radial direction. The fuel assembly lower end-fitting and top upper core region both extend to infinity in the axial direction. The fuel cladding material may be specified as either Zircaloy-4, Stainless Steel 304, or Inconel. The spacer grid homogenized region materials are defined as previously described in Section 3.6.6. The fuel material is defined as described in Section 3.5.

3.6.8. Guide Tube Universe Specifications

MACE creates a different guide tube universe for each unique fuel assembly design that contains empty guide tubes in the CRC configuration. The guide tube universes contain the definition of a complete fuel assembly lattice cell containing an empty guide tube. The guide tube, spacer grids, and fuel assembly end-fittings are all modeled in this universe. Additionally, the regions of the core above the upper fuel assembly end-fitting are modeled. The material composition of these upper core regions should be defined by the user to correspond to the upper core regions containing no hardware constituents from an insertion assembly. In Westinghouse reactor configurations, the guide tubes may be specified as containing more than one axial section with different radial dimensions. The guide tube universe extends to infinity in the radial direction. The fuel assembly lower end-fitting and top upper core region both extend to infinity in the axial direction. The guide tube material may be specified as either Zircaloy-4, Stainless Steel 304, or Inconel. The spacer grid homogenized region materials are defined as previously described in section 3.6.6.

3.6.9. Instrument Tube Universe Specifications

MACE creates a different instrument tube universe for each fuel assembly design in the CRC configuration. The instrument tube universes contain the definition of a complete fuel assembly lattice cell containing a water-filled instrument tube. The instrument tube, spacer grids, and fuel assembly end-fittings are all modeled in this universe. Additionally, the regions of the core above the upper fuel assembly end-fitting are also modeled. The material composition of these upper core regions should be defined by the user to correspond to the upper core regions containing no hardware constituents from an insertion assembly. In Westinghouse reactor configurations, the instrument tubes may be specified as containing more than one axial section with different radial dimensions. The instrument tube universe extends to infinity in the radial direction. The fuel assembly lower end-fitting and top upper core region both extend to infinity in the axial direction. The instrument tube material may be specified as Zircaloy-4, Stainless Steel 304, or Inconel. The spacer grid homogenized region materials are defined as previously described in section 3.6.6.

3.6.10. Control Rod Universe Specifications

MACE creates a different control rod universe for each unique fuel assembly and control rod assembly design combination that exists in the core. A unique control rod assembly design is delineated by any control rod assembly that is inserted at a height not previously defined for that fuel assembly design. The control rod universes contain the definition of a complete fuel assembly lattice cell containing a guide tube with a control rod inserted to its specified height. The control rod, guide tube, spacer grids, and fuel assembly end-fittings are all modeled in this universe. Additionally, the regions of the core above the upper fuel assembly end-fitting are also modeled. The material composition of these upper core regions should be defined by the user to correspond to the upper core regions containing hardware constituents from a control rod assembly. The control rod universe extends to infinity in the radial direction. The fuel assembly lower end-fitting and top upper core region both extend to infinity in the axial direction. The guide tube and control rod cladding material may be specified as Zircaloy-4, Stainless Steel 304, or Inconel. The spacer grid homogenized region materials are defined as previously described in section 3.6.6. The control rod absorber material is defined by the user in the MACE input deck. The control rod has an upper and lower plenum between the absorber material and the ends of the rod. The material contained in these plenum regions must be specified by the user in the MACE input deck.

3.6.11. Axial Power Shaping Rod Universe Specifications

MACE creates a different axial power shaping rod (APSR) universe for each unique fuel assembly and axial power shaping rod assembly (APSRA) design combination that exist in the core. A unique APSRA design is delineated by any APSRA that is inserted at a height not previously defined for that fuel assembly design. The APSR universes contain the definition of a complete fuel assembly lattice cell containing a guide tube with an APSR inserted to its specified height. The APSR, guide tube, spacer grids, and fuel assembly end-fittings are all modeled in this universe. Additionally, the regions of the core above the upper fuel assembly end-fitting are also modeled. The material composition of these upper core regions should be defined by the user to correspond to the upper core regions containing hardware constituents from an axial power shaping rod assembly. The APSR universe extends to infinity in the radial direction. The fuel assembly lower end-fitting and top upper core region both extend to infinity in the axial direction. The guide tube and APSR cladding material may be specified as Zircaloy-4, Stainless Steel 304, or Inconel. The spacer grid homogenized region materials are defined as previously described in section 3.6.6. The APSR absorber material is defined by the user in the MACE input deck. The APSR may have a plenum region between the absorber material and either the intermediate plug or the lower end-cap of the APSR. The material contained in these plenum regions must be specified by the user in the MACE input deck. The intermediate plug material is specified as Zircaloy-4, Stainless Steel 304, or Inconel. A volume of intermediate plug material is also provided in the MACE input deck. The intermediate plug is then homogenized with moderator in a given region of the APSR in the same manner previously described for the homogenized spacer grid materials.

3.6.12. Burnable Poison Rod Universe Specifications

MACE creates a unique burnable poison rod (BPR) universe definition for each burnable poison rod assembly (BPRA) inserted in the core within each unique fuel assembly. The BPR universe represents a complete fuel assembly lattice cell containing a guide tube with the BPR inserted to its specified height. The BPR universe contains a number of burnable poison nodes with a user defined nodal delineation. The nodal delineation used in the MACE input deck should correspond to that used in the CRAFT depletion calculations from which the depleted burnable poison isotopics will be retrieved. Any burnable poison node may be specified as non-absorbing in the MACE input deck. The burnable poison rod cladding and guide tube material may be specified as Zircaloy-4, Stainless Steel 304, or Inconel. A plenum may exist between the burnable poison material and either the upper or lower BPR end-caps. The plenum materials are specified by the user in the MACE input deck. The BPR is modeled inside a guide tube to the specified height. BPRs may only be specified as non-annular in the B&W reactor configurations. BPRs may be specified as either non-annular or annular in the Westinghouse reactor configurations. Annular BPRs may have either a void gap or a moderator-filled gap. Several BPR loading formats are available for use with the 17x17 lattice fuel assembly arrangement. These BPR loading formats are shown in Figure 3.6.12-1. If a BPR loading format other than those provided in MACE is required, the user can specify one of the similar formats available in MACE and then easily modify the fuel assembly lattice specification in the MCNP input deck to obtain the desired BPR loading format. The spacer grids, fuel assembly upper and lower end-fittings, and upper core regions are also included in the BPR universe definition. The BPR universe extends to infinity in the radial direction. The fuel assembly lower end-fitting and top upper core region both extend to infinity in the axial direction. The burnable poison material specifications are obtained from the FUEL subroutine.

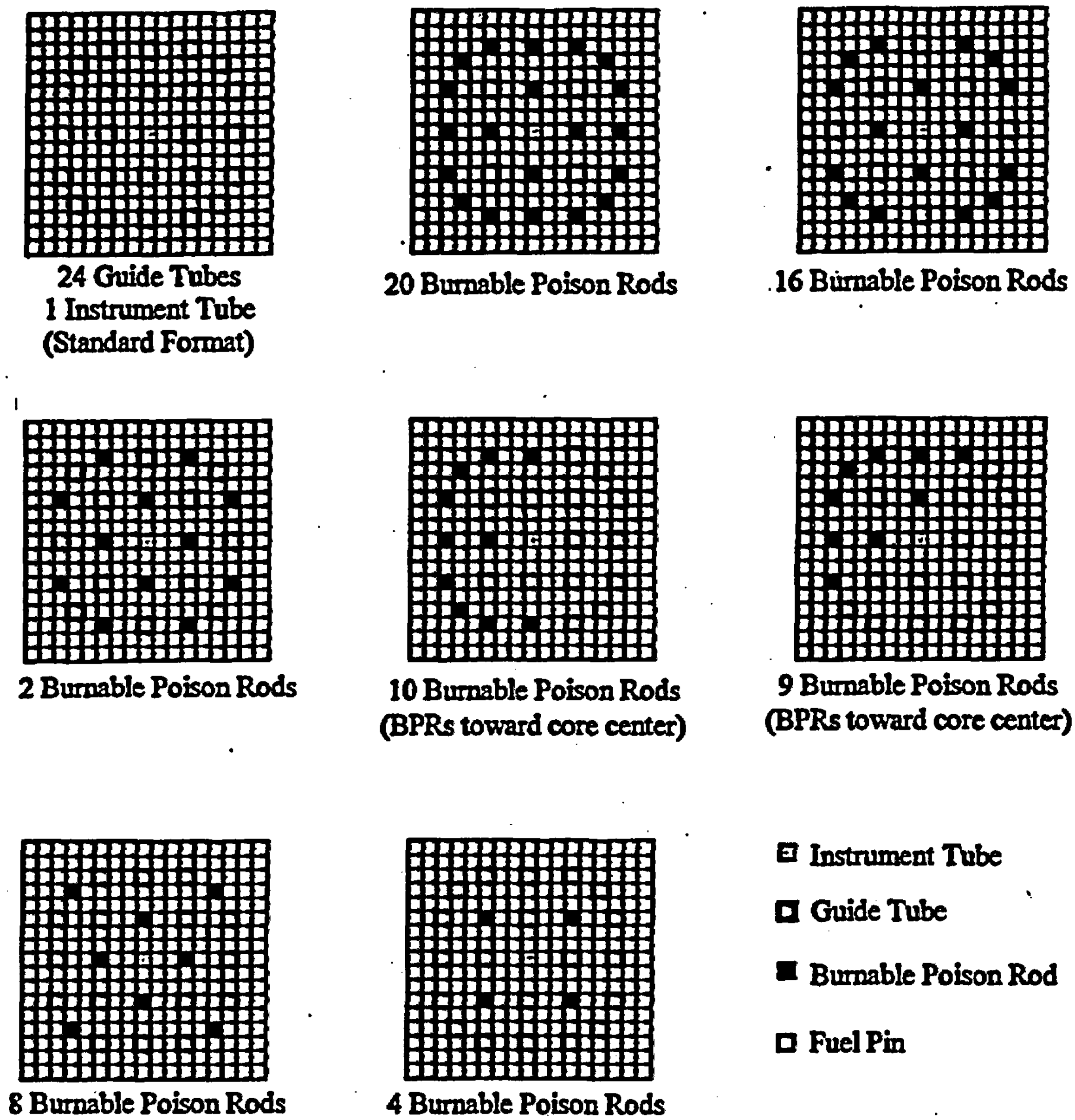


Figure 3.6.12-1. BPR Loading Formats Available in MACE Version 3

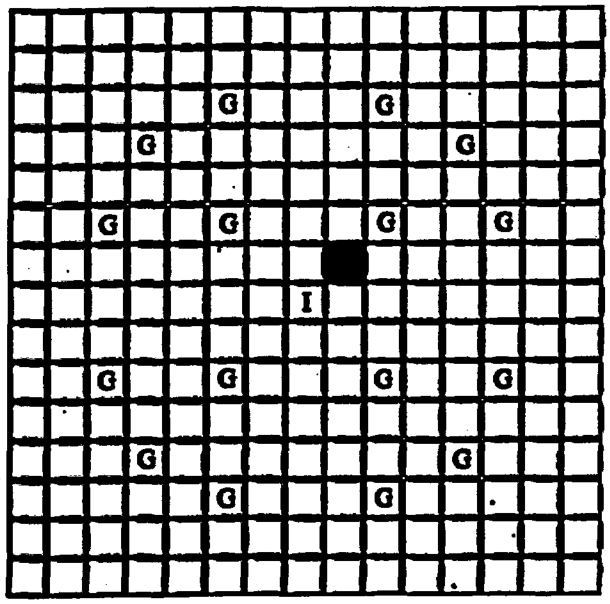
3.7. SURFSECTION Subroutine

This subroutine writes all of the surface specifications to a file called "___C__T___.surf" for direct implementation in the MCNP input deck. The three spaces before the "C" contain the reactor identification prefix. The two spaces after the "C" contain the cycle identifier. The three spaces after the "T" contain the effective full-power day value for the CRC statepoint evaluation. Each surface specification consists of the surface type designator and the surface location designators.

3.8. CONTROL Subroutine

This subroutine writes the control and initial source specifications for the MCNP calculation to a file called "___C__T___.cont" for direct implementation in the MCNP input deck. The spaces in the filename contain the same information as previously described in the SURFSECTION subroutine description. The "KCODE" control card specifications include the number of neutron histories per cycle, the number of converging cycles, the total number of cycles, and the initial guess for k_{eff}. In the MACE generated MCNP input decks, the initial guess for k_{eff} is always set as 1. The other control inputs are provided by the user in the MACE input deck.

The initial source specification for the MCNP calculation is defined by placing one starting neutron source point in the center of a central fuel rod in each fuel node of each assembly in the core. This is shown graphically in Figure 3.8-1.



I = Instrument Tube Location

G = Guide Tube Location

□ = Fuel Rod Location

■ = Location of Fuel Rod Containing Initial Neutron Source Point

THIS SKETCH IS NOT TO SCALE.

Note: Each delineated fuel node in each fuel assembly in the core contains an initial neutron source point. The initial neutron source point in each axial
fuel node is located at the mid-point of each node.

Figure 3.8-1. Initial Neutron Source Distribution

4. MACE Input Description

This section presents the MACE input deck development instructions. The MACE input deck should be developed in accordance with the structure presented in Table 4-1.

Table 4-1. MACE Input Deck Development Instructions

Card Number	Repeat Signals		Card Format	Card Description
1			Integer	Type of Reactor: 1 = B&W, 2 = Westinghouse, 3 = Combustion Engineering (MACE Version 3 only allows the B&W and Westinghouse reactor options.)
2			Integer	Type of core symmetry: 1 = 1/8, 2 = 1/4, 3 = full
3			21 Character Maximum Starting in Column 1	Reactor Identification
4			3 Character Maximum Starting in Column 1	Reactor Identification Prefix
5			2 Character Maximum Starting in Column 1	Cycle Identifier for the CRC Evaluation Statepoint
6			Real	Effective Full-Power Day (EFPD) Value of the CRC Evaluation Statepoint
7			Integer	Fuel Isotopic Composition Request: 1 = Best-Estimate Set (Table 3.5.2-2), 2 = Principal Isotopes (Table 3.5.2-3), 3 = Principal Actinide Set (Table 3.5.2-4), 4 = Actinide-Only Set (Table 3.5.2-5)
8			1 Character in Column 1	Fuel Density Option: "T" = Apply Pressed Density of 10.41 g/cc to all fresh and depleted fuel, "C" = Apply Pressed Density of 10.41 g/cc to all fresh fuel and use preservation of mass to determine the depleted fuel density, "{Any Other Character}" = Use preservation of mass to determine both the fresh and depleted fuel density
9			Integer, Integer, Integer	Number of Neutron Histories per Cycle, Number of Convergence Cycles, Total Number of Cycles

Table 4-1. MACE Input Deck Development Instructions

Card Number	Repeat Signals		Card Format	Card Description
10			Integer	Number of Fuel Nodes Used in CRAFT Calculations that Support the CRC Evaluation
11			Integer {space} Real	CRAFT Calculation Fuel Node Number, Corresponding Fuel Node Height (cm)
12			Real	Assembly Pitch in Core (cm)
13			Real {space} Real {space} Real	Moderator Temperature (°F), System Pressure (psia), Soluble Boron Concentration (ppmb)
14			See Figures 4-1, 4-2, 4-3 for B&W Design See Figures 4-4, 4-5, 4-6 for Westinghouse Design	Assembly Identifier Provided in Core Layout Format
15			Integer	Number of Fuel Assembly Designs in CRC Evaluation
16			See Figures 4-1, 4-2, 4-3 for B&W Design See Figures 4-4, 4-5, 4-6 for Westinghouse Design	Assembly Design Designations Provided in Core Layout Format
17			Integer	Number of Insertion Rod Assembly Banks (Including BPRAs)
18			Integer {space} 5 Characters {space} Real (Field width of 10)	Insertion Rod Assembly Bank Identifier, Bank Description, Distance Between Bottom of Active Fuel and Bottom of Absorber Material in Insertion Assembly (cm)

Table 4-1. MACE Input Deck Development Instructions

Card Number	Repeat Signals		Card Format	Card Description
19			See Figures 4-1, 4-2, 4-3 for B&W Design See Figures 4-4, 4-5, 4-6 for Westinghouse Design	Insertion Assembly Bank Locations in Core
20			See Figures 4-1, 4-2, 4-3 for B&W Design See Figures 4-4, 4-5, 4-6 for Westinghouse Design	Initial Fresh Fuel U-235 Enrichments for Each Assembly in Core (Both Fresh and Depleted)
21			See Figures 4-1, 4-2, 4-3 for B&W Design See Figures 4-4, 4-5, 4-6 for Westinghouse Design	Fuel Status ("F" for Fresh or "B" for Burned) for Each Assembly in Core
22			Integer	Number of Lower Core Regions (Below the Fuel Assembly Lower End-Fittings) to be Modeled (A maximum of ten lower region may be defined.)
23			Real {space} Real {space} Integer	Height of Lower Core Region (cm), Density of Material in Lower Core Region (g/cc), Number of Isotopes in Lower Core Region
24			9 Characters {space} Real	For each isotope in the lower region previously defined: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
25			Real	Distance between the core liner and the outer edge of the fuel assembly outer unit cell boundary (cm)
26			Real {space} Real {space} Integer	Core liner thickness (cm), Core liner density (g/cc), Number of isotopes in the core liner material (maximum of 35 isotopes)

Table 4-1. MACE Input Deck Development Instructions

Card Number	Repeat Signals		Card Format	Card Description
27			9 Characters {space} Real	For each isotope in the core liner region previously defined: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
28			Real	Core barrel inner radius (cm)
29			Real {space} Real {space} Integer	Core barrel thickness (cm), Core barrel density (g/cc), Number of isotopes in core barrel material (maximum of 35 isotopes)
30			9 Characters {space} Real	For each isotope in the core barrel region previously defined: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
31			Real	{FOR B&W ONLY} Thermal shield inner radius (cm)
32			Real {space} Real {space} Integer	{FOR B&W ONLY} Thermal shield thickness (cm), Thermal shield density (g/cc), Number of isotopes in thermal shield material (maximum of 35 isotopes)
33			9 Characters {space} Real	{FOR B&W ONLY} For each isotope in the thermal shield region previously defined: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
34			Real	{FOR Westinghouse ONLY} Neutron pad inner radius (cm)
35			Real {space} Real {space} Integer	{FOR Westinghouse ONLY} Neutron pad thickness (cm), Neutron pad density (g/cc), Number of isotopes in neutron pad material (maximum of 35 isotopes)
36			9 Characters {space} Real	{FOR Westinghouse ONLY} For each isotope in the neutron pad region previously defined: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
37			Real {space} Real	{FOR Westinghouse ONLY} Lowest and highest neutron pad bounding angles in degrees for the neutron pad in the northeast quadrant of the reactor core where the y-axis in 0 degrees

Table 4-1. MACE Input Deck Development Instructions

Card Number	Repeat Signals		Card Format	Card Description
38			Real {space}	Pressure vessel cladding inner radius from the center of the core (cm)
39			Real {space} Real {space} Integer	Pressure vessel cladding thickness (cm), Pressure vessel cladding density (g/cc), Number of isotopes in pressure vessel cladding material (maximum of 35 isotopes)
40			9 Characters {space} Real	For each isotope in the pressure vessel cladding region previously defined: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
41			Real {space} Real {space} Integer	Pressure vessel thickness (cm), Pressure vessel density (g/cc), Number of isotopes in pressure vessel material (maximum of 35 isotopes)
42			9 Characters {space} Real	For each isotope in the pressure vessel region previously defined: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
If BPRAs are present in the core, input cards 43 through 45. Otherwise, go to card 46.				
43			Integer	Number of regions above the upper end-fitting of a fuel assembly containing a BPRA
44			Real {space} Real {space} Integer	For each region above the upper end-fitting of a fuel assembly containing a BPRA (from the top-most region to just above the upper end-fitting): Region height (cm), Region density (g/cc), Number of isotopes in region
45			9 Characters {space} Real	For each isotope in the upper core region previously defined: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
If CRAs are present in the core, input cards 46 through 48. Otherwise, go to card 49.				
46			Integer	Number of regions above the upper end-fitting of a fuel assembly containing a CRA
47			Real {space} Real {space} Integer	For each region above the upper end-fitting of a fuel assembly containing a CRA (from the top-most region to just above the upper end-fitting): Region height (cm), Region density (g/cc), Number of isotopes in region

Table 4-1. MACE Input Deck Development Instructions

Card Number	Repeat Signals		Card Format	Card Description
48			9 Characters {space} Real	For each isotope in the upper core region previously defined: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
If it is a B&W reactor design and APSRAs are present in the core, input cards 49 through 51. Otherwise, go to card 52.				
49			Integer	Number of regions above the upper end-fitting of a fuel assembly containing an APSRA
50			Real {space} Real {space} Integer	For each region above the upper end-fitting of a fuel assembly containing an APSRA (from the top-most region to just above the upper end-fitting): Region height (cm), Region density (g/cc), Number of isotopes in region
51			9 Characters {space} Real	For each isotope in the upper core region previously defined: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
52			Integer	Number of regions above the upper end-fitting of a fuel assembly containing no insertion assembly
53			Real {space} Real {space} Integer	For each region above the upper end-fitting of a fuel assembly containing no insertion assembly (from the top-most region to just above the upper end-fitting): Region height (cm), Region density (g/cc), Number of isotopes in region
54			9 Characters {space} Real	For each isotope in the upper core region previously defined: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
Cards 55 through 87 must be entered for each fuel assembly design. A fuel assembly design is delineated by any unique fuel assembly design/insertion assembly combination. An insertion assembly of the same type that is inserted to a different height must be considered a separate insertion assembly design.				
55			Integer	Number of fuel nodes to be modeled in MCNP input deck.
56			Integer {space} Real	For each fuel node: Node number, Node height (cm)
57			Integer	Number of fuel rods in assembly
58			Real	Pin pitch in assembly (cm)
59			Real	Mass of uranium in assembly (grams)

Table 4-1. MACE Input Deck Development Instructions

Card Number	Repeat Signals		Card Format	Card Description
60			Real {space} Real {space} Real	Fuel pellet radius (cm), Fuel cladding inner radius (cm), Fuel cladding outer radius (cm)
61			Real {space} Real	Upper fuel rod plenum height (should include end-cap height) (cm), Lower fuel rod plenum height (should include end-cap height) (cm)
62			Real {space} Real	Fuel rod upper end-cap height (cm), Fuel rod lower end-cap height (cm)
63			Real {space} Real	Fuel assembly upper end-fitting height (cm), Fuel assembly lower end-fitting height (cm)
64			Integer	Number of spacer grids in assembly (includes the top and intermediate grids) (maximum of 15)
65			Integer {space} Real {space} Real {space} Real {space} Integer {space} Optional: Real {space} Real	For each spacer grid in the assembly as previously identified: Spacer grid number, Height of spacer grid (cm), Distance between the top of the spacer grid and the top of the lower pad (cm) (Except for the top spacer grid whose value should be the distance between the top of the spacer grid and the bottom of the active fuel.), Volume of spacer grid material (cc), Spacer grid material selection (1 = Zircaloy-4, 2 = Stainless Steel 304, 3 = Inconel, 4 = Stainless Steel 304 & Inconel mixture, 5 = Stainless Steel 304 & Zircaloy-4 mixture), Stainless Steel 304 volume fraction in spacer material, Either Inconel or Zircaloy-4 volume fraction in spacer material (The last two entries on this card should only be specified if the spacer grid material selection is either 4 or 5.)
66			Integer	Fuel rod cladding material specification: 1 = Zircaloy-4, 2 = Stainless Steel 304, 3 = Inconel
67			Real {space} Integer	Fuel rod upper plenum density (g/cc), Number of isotopes in the fuel rod upper plenum (maximum of 35 isotopes)
68			9 Characters {space} Real	For each isotope in the fuel rod upper plenum region previously defined: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)

Table 4-1. MACE Input Deck Development Instructions

Card Number	Repeat Signals		Card Format	Card Description
69			Real {space} Integer	Fuel rod lower plenum density (g/cc), Number of isotopes in the fuel rod lower plenum (maximum of 35 isotopes)
70			9 Characters {space} Real	For each isotope in the fuel rod lower plenum region previously defined: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
71			Real {space} Integer	Fuel assembly upper end-fitting density (g/cc), Number of isotopes in the fuel assembly upper end-fitting (maximum of 35 isotopes)
72			9 Characters {space} Real	For each isotope in the fuel assembly upper end-fitting region previously defined: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
73			Real {space} Integer	Fuel assembly lower end-fitting density (g/cc), Number of isotopes in the fuel assembly lower end-fitting (maximum of 35 isotopes)
74			9 Characters {space} Real	For each isotope in the fuel assembly lower end-fitting region previously defined: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
Input card 75 is no longer active and should not be specified.				
Input cards 76 through 79 if the reactor design is B&W.				
76			Real {space} Real {space} Real {space} Real	Guide tube inner radius (cm), Guide tube outer radius (cm), Distance between guide tube top and top of lower core pad (cm), Distance between guide tube bottom and top of lower core pad (cm)
77			Integer	Guide tube material specification: 1 = Zircaloy-4, 2 = Stainless Steel 304, 3 = Inconel
78			Real {space} Real {space} Real {space} Real	Instrument tube inner radius (cm), Instrument tube outer radius (cm), Distance between the top of the instrument tube and the top of the lower core pad (cm), Distance between the bottom of the instrument tube and the top of the lower core pad (cm)
79			Integer	Instrument tube material specification: 1 = Zircaloy-4, 2 = Stainless Steel 304, 3 = Inconel
Input cards 80 through 87 if the reactor design is Westinghouse.				

Table 4-1. MACE Input Deck Development Instructions

Card Number	Repeat Signals		Card Format	Card Description
80			Integer	If any guide tubes in this assembly design have multiple axial regions input a value of 1. Otherwise, input any number.
If the value of input card 80 is 1, input cards 81 through 83.				
81			Integer	Number of guide tube axial regions in this assembly design.
82			Real {space} Real {space} Real {space} Real	Guide tube axial section inner radius (cm), Guide tube axial section outer radius (cm), Distance between guide tube axial section top and top of lower core pad (cm), Distance between guide tube axial section bottom and top of lower core pad (cm)
83			Integer	Guide tube axial section material selection: 1=Zircaloy-4, 2=Stainless Steel 304, 3=Inconel
If the value of input card 80 is not 1, input cards 76 and 77 here.				
84			Integer	If any instrument tubes in this assembly design have multiple axial regions input a value of 1. Otherwise, input any number.
If the value of input card 84 is 1, input cards 85 and 87.				
85			Integer	Number of instrument tube axial regions in this assembly design.
86			Real {space} Real {space} Real {space} Real	Instrument tube axial section inner radius (cm), Instrument tube axial section outer radius (cm), Distance between instrument tube axial section top and top of lower core pad (cm), Distance between instrument tube axial section bottom and top of lower core pad (cm)
87			Integer	Instrument tube axial section material selection: 1=Zircaloy-4, 2=Stainless Steel 304, 3=Inconel
If the value of input card 84 is not 1, input cards 78 and 79 here.				
Cards 88 through 112 must be entered if any BPRAs are present in the CRC configuration.				
88			Integer	Number of different BPRA banks present in the core. A BPRA inserted in a different assembly such that it has a unique depletion history must be considered a part of a unique BPRA bank.
Cards 89 through 112 must be entered for each different BPRA bank as previously identified.				
89			Integer	BPRA bank identifier

Table 4-1. MACE Input Deck Development Instructions

Card Number	Repeat Signals		Card Format	Card Description
90			Integer	Number of burnable poison axial nodes to be used in MCNP model for this BPRA bank (MACE Version 3 requires this value to correspond to the number of burnable poison nodes modeled in the CRAFT calculations from which the depleted burnable poison isotopics will be retrieved.)
91			Integer {space} Real	For each axial burnable poison node in this BPRA bank: Node number, Node height (cm)
92			Real	Distance between the bottom of the fuel assembly lower end-fitting and the bottom of the bottom burnable poison node (cm)
93			1 Character in Column 1	Identify the presence of any non-absorbing burnable poison nodes in this BPRA bank: "Y" = non-absorbing nodes exist, "{any other character}" = no non-absorbing nodes exist
94			Integer {space} 1 Character in Column 4	If non-absorbing burnable poison nodes exist in this BPRA bank then enter the information for this card. Otherwise, skip it. Burnable poison node number, "Y" if node is non-absorbing, any other character indicated otherwise (Enter the information for all burnable poison nodes in sequential order from top node to bottom node.)
95			Integer	{Input this card for Westinghouse ONLY} Type of BPRA: 1= 4 BPRs, 2 = 8 BPRs, 3 = 9 BPRs, 4 = 10 BPRs, 5 = 12 BPRs, 6 = 16 BPRs, 7 = 20 BPRs (See Figure 3.6.12-1)
96			Integer	{Input this card for Westinghouse ONLY} Type of BPR: 1= solid, 2 = gap annular, 3 = water annular
97			Real {space} Real {space} Real	{Input this card for either B&W reactors or Westinghouse reactors with input card 96 equal to 1.} Burnable absorber pellet radius (cm), Burnable poison rod (BPR) inner radius (cm), BPR outer radius (cm)

Table 4-1. MACE Input Deck Development Instructions

Card Number	Repeat Signals		Card Format	Card Description
98			Real {pace} Real {space} Real {space} Real {space} Real {space} Real	{Input this card for Westinghouse reactors with input card 96 equal to either 2 or 3.} BPR inner cladding inner radius (cm), BPR inner cladding outer radius (cm), BP absorber inner radius (cm), BP absorber outer radius (cm), BPR outer cladding inner radius (cm), BPR outer cladding outer radius (cm)
99			Real {space} Real	BPR upper plenum height (cm), BPR lower plenum height (cm)
100			Integer	BPR absorber material specification: 1 = Al_2O_3-B_4C, Any other number delineated otherwise
If the BPR absorber material is not Al_2O_3-B_4C, enter cards 101 and 102.				
101			Real {space} Integer	Burnable poison density (g/cc), Number of isotopes in burnable poison material
102			9 Characters {space} Real	For each isotope in the burnable poison material: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
103			Real {space} Real	If the BPR absorber material is Al_2O_3-B_4C: Al_2O_3-B_4C density (g/cc), wt % of B_4C in Al_2O_3-B_4C
If non-absorbing burnable poison material exists in this BPR bank, enter cards 104 through 107.				
104			Integer	Non-absorbing material specification: 1 = Al_2O_3, Any other number indicates otherwise
105			Real {space} Integer	If the non-absorbing burnable poison material is not Al_2O_3: Non-absorbing burnable poison density (g/cc), Number of isotopes in non-absorbing material (maximum of 35 isotopes)
106			9 Characters {space} Real	If the non-absorbing burnable poison material is not Al_2O_3: For each isotope in the non-absorbing burnable poison material: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
107			Real	If the non-absorbing burnable poison material is Al_2O_3: Al_2O_3 density (g/cc)
108			Integer	BPR cladding material specification: 1 = Zircaloy-4, 2 = Stainless Steel 304, 3 = Inconel

Table 4-1. MACE Input Deck Development Instructions

Card Number	Repeat Signals		Card Format	Card Description
109			Real {space} Integer	BPR upper plenum material density (g/cc), Number of isotopes in the BPR upper plenum material (maximum of 35 isotopes)
110			9 Characters {space} Real	For each isotope in the BPR upper plenum material: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
111			Real {space} Integer	BPR lower plenum material density (g/cc), Number of isotopes in the BPR lower plenum material (maximum of 35 isotopes)
112			9 Characters {space} Real	For each isotope in the BPR lower plenum material: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
Cards 113 through 122 must be entered for B&W reactor designs if any CRAs are present in the CRC configuration.				
113			Integer	Number of different CRA banks. A CRA bank positioned at a different height than previously defined for that CRA bank must be delineated as a different CRA bank.
Cards 114 through 122 must be entered for each CRA bank previously identified.				
114			Integer	CRA bank identifier.
115			Real {space} Real {space} Real {space} Real {space} Real {space} Real {space} Real	Control rod (CR) absorber radius (cm), CR cladding inner radius (cm), CR cladding outer radius (cm), Distance between the CR absorber material bottom and the bottom of the active fuel (cm), CR absorber material height (cm), CR lower plenum height (including end-cap) (cm), CR upper plenum height (including end-cap) (cm)
116			Real {space} Integer	Density of CR absorber material (g/cc), Number of isotopes in CR absorber material
117			9 Characters {space} Real	For each isotope in the CR absorber material: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
118			Integer	CR cladding material specification: 1 = Zircaloy-4, 2 = Stainless Steel 304, 3 = Inconel

Table 4-1. MACE Input Deck Development Instructions

Card Number	Repeat Signals		Card Format	Card Description
119			Real {space} Integer	CR upper plenum material density (g/cc), Number of isotopes in CR upper plenum material
120			9 Characters {space} Real	For each isotope in the CR upper plenum material: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
121			Real {space} Integer	CR lower plenum material density (g/cc), Number of isotopes in CR lower plenum material
122			9 Characters {space} Real	For each isotope in the CR lower plenum material: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
Cards 123 through 135 must be entered for Westinghouse reactor designs if any CRAs are present in the CRC configuration.				
123			Integer	Enter a value of 1 if any of the CRAs contain HYBRID CRs. Otherwise, enter any other value.
If the value of input card 123 is not 1, enter the information requested in input cards 113 through 122 for this Westinghouse reactor. Otherwise, enter input cards 124 through 135.				
124			Integer	Number of different CRA banks. A CRA bank positioned at a different height than previously defined for that CRA bank must be delineated as a different CRA bank.
Input cards 125 through 135 must be entered for each Westinghouse CRA bank.				
125			Integer {space} Integer	CRA bank identifier, Flag to indicate if this CRA contains HYBRID CRs: 1=HYBRID, any other # is not a HYBRID
126			Integer	Number of CR axial sections
Input cards 127 through 130 must be entered for each CR axial section in this CRA.				
127			Real {space} Real {space} Real {space} Real {space} Real	Control rod (CR) absorber radius (cm), CR cladding inner radius (cm), CR cladding outer radius (cm), Distance between the CR absorber material bottom and the bottom of the active fuel (cm), CR axial section height (cm)
128			Real {space} Integer	Density of CR absorber material in this CR axial section (g/cc), Number of isotopes in this CR absorber material

Table 4-1. MACE Input Deck Development Instructions

Card Number	Repeat Signals		Card Format	Card Description
129			9 Characters {space} Real	For each isotope in this CR axial section absorber material: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
130			Integer	CR cladding material specification in this CR axial section: 1 = Zircaloy-4, 2 = Stainless Steel 304, 3 = Inconel
131			Real {space} Real {space} Real	Control rod end-cap height (cm), CR lower plenum height (cm), CR upper plenum height (cm)
132			Real {space} Integer	CR upper plenum material density (g/cc), Number of isotopes in CR upper plenum material
133			9 Characters {space} Real	For each isotope in the CR upper plenum material: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
134			Real {space} Integer	CR lower plenum material density (g/cc), Number of isotopes in CR lower plenum material
135			9 Characters {space} Real	For each isotope in the CR lower plenum material: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
Cards 136 through 148 must be inserted if any APSRAs are present in the CRC configuration (this will only apply to B&W reactor designs).				
136			Integer	Number of different APSRA banks. An APSRA bank positioned at a different height than previously defined for that APSRA bank must be delineated as a different APSRA bank.
Cards 108 through 119 must be entered for each APSRA bank previously identified.				
137			Integer	APSRA bank identifier.
138			Real {space} Real {space} Real {space} Real {space} Real	Axial power shaping rod (APSR) absorber radius (cm), APSR cladding inner radius (cm), APSR cladding outer radius (cm), APSR follow rod inner radius (cm), APSR follow rod outer radius (cm) (The APSR follow rod dimensions must be the same as the APSR cladding dimensions in MACE Version 3.)
139			Real	Volume fraction of the intermediate spacer in the APSR

Table 4-1. MACE Input Deck Development Instructions

Card Number	Repeat Signals		Card Format	Card Description
140			Real {space} Real {space} Real {space} Real {space} Real {space} Real	Distance between the APSR absorber material bottom and the bottom of the active fuel (cm), APSR absorber region height (cm), APSR intermediate plug height (cm), APSR lower plug height (cm), APSR upper plenum height (cm), APSR lower plenum height (cm)
141			Real {space} Integer	Density of APSR absorber material (g/cc), Number of isotopes in APSR absorber material
142	▓		9 Characters {space} Real	For each isotope in the APSR absorber material: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
143			Integer	APSR cladding material specification: 1 = Zircaloy-4, 2 = Stainless Steel 304, 3 = Inconel
144			Integer	APSR follow rod material specification: 1 = Zircaloy-4, 2 = Stainless Steel 304, 3 = Inconel (The APSR follow rod material must be the same as the APSR cladding material in MACE Version 3.)
145			Real {space} Integer	APSR upper plenum material density (g/cc), Number of isotopes in APSR upper plenum material
146	▓		9 Characters {space} Real	For each isotope in the APSR upper plenum material: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)
147			Real {space} Integer	APSR lower plenum material density (g/cc), Number of isotopes in APSR lower plenum material
148	▓		9 Characters {space} Real	For each isotope in the APSR lower plenum material: MCNP ZAID of isotope, wt % of isotope (maximum of 35 isotopes)

▓ : The shaded "Repeat Signals" indicate that the corresponding input cards or sets of input cards must be repeated a number of times. Usually, the number of repetitions required is indicated by the input card directly before the repeated card.

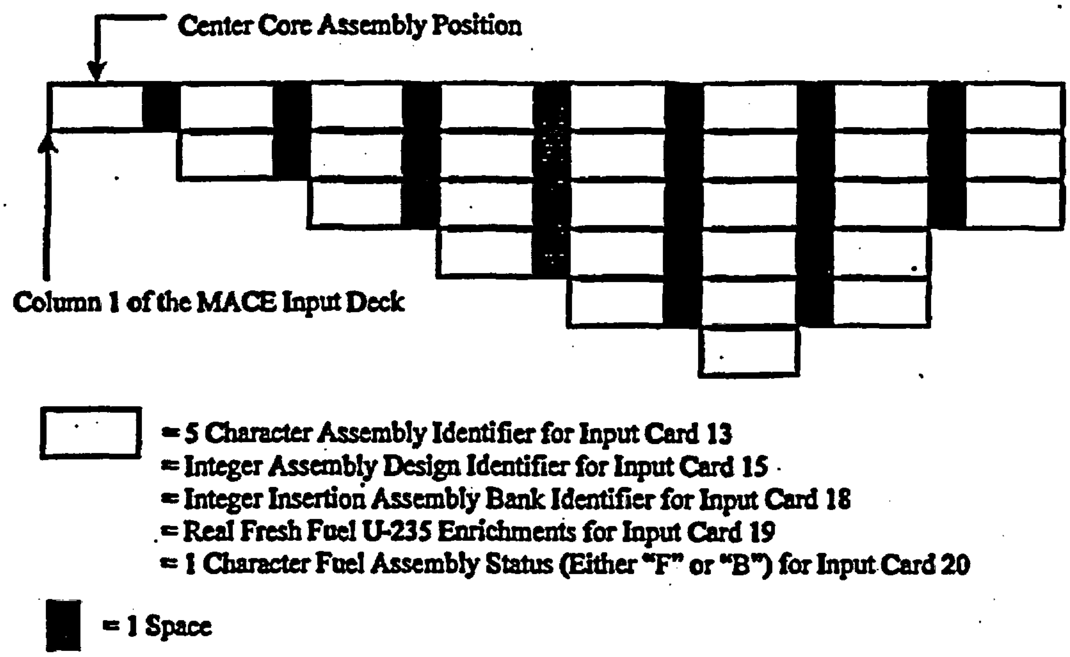
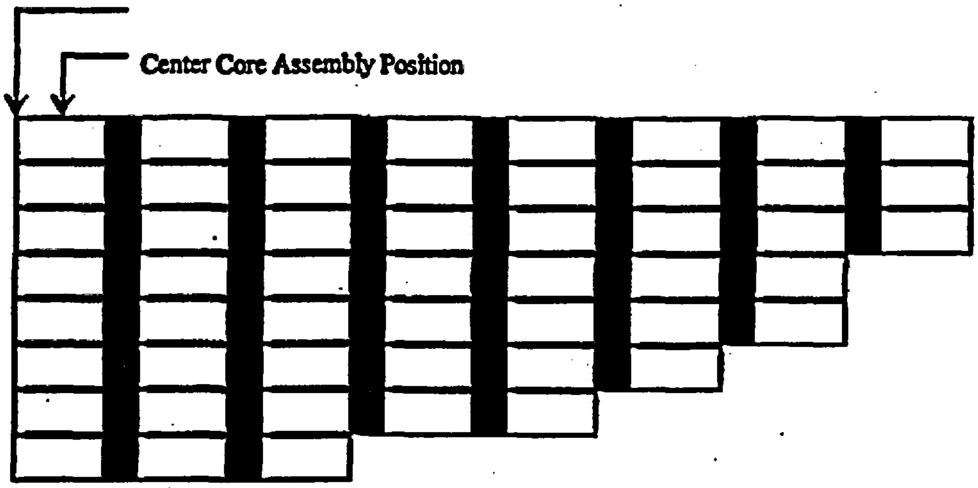


Figure 4-1. 1/8 Core Symmetric Input Specification Format for B&W Reactors



= 5 Character Assembly Identifier for Input Card 13
= Integer Assembly Design Identifier for Input Card 15
= Integer Insertion Assembly Bank Identifier for Input Card 18
= Real Fresh Fuel U-235 Enrichments for Input Card 19
= 1 Character Fuel Assembly Status (Either "F" or "B") for Input Card 20

= 1 Space

Figure 4-2. 1/4 Core Symmetric Input Specification Format for B&W Reactors

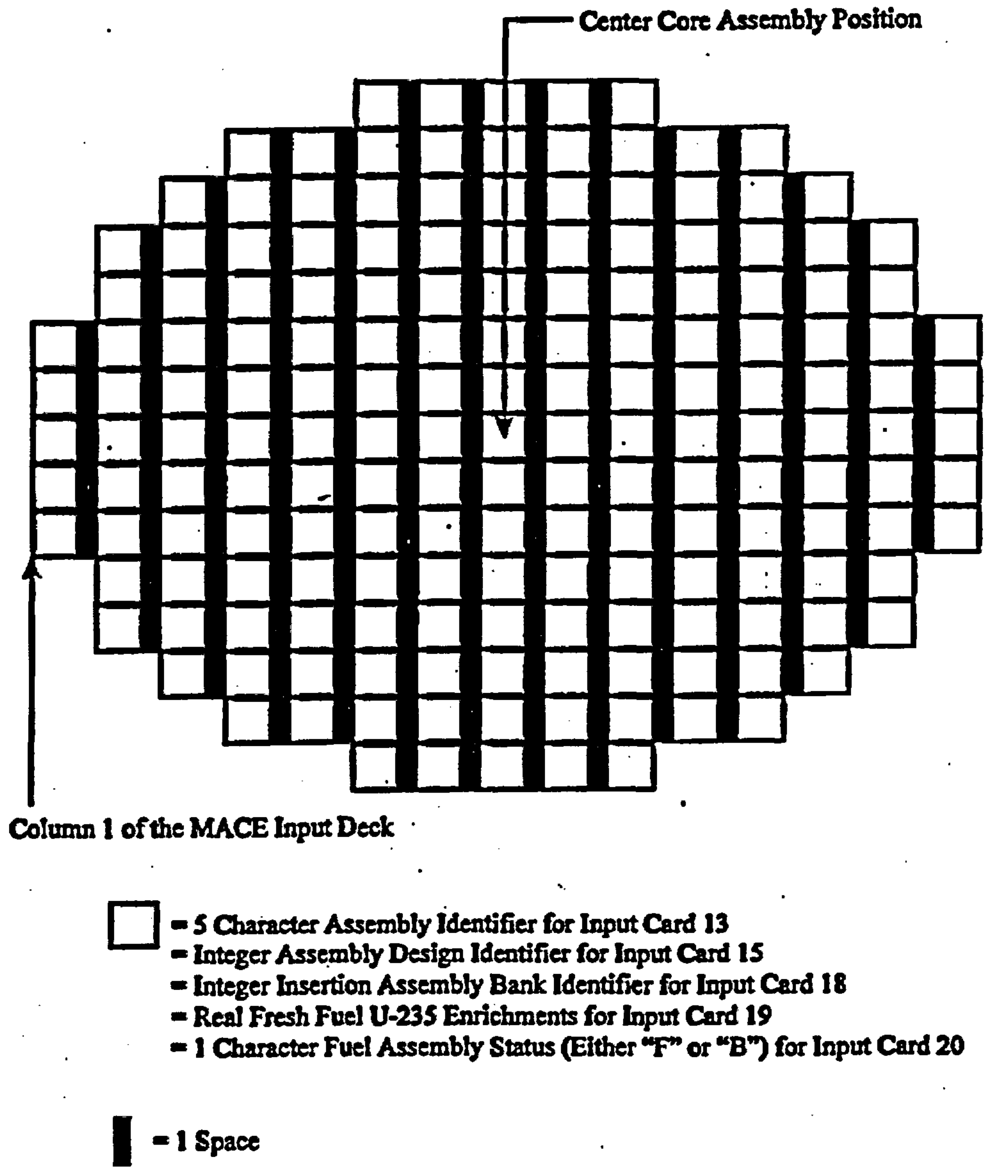


Figure 4-3. Full Core Symmetric Input Specification Format for B&W Reactors

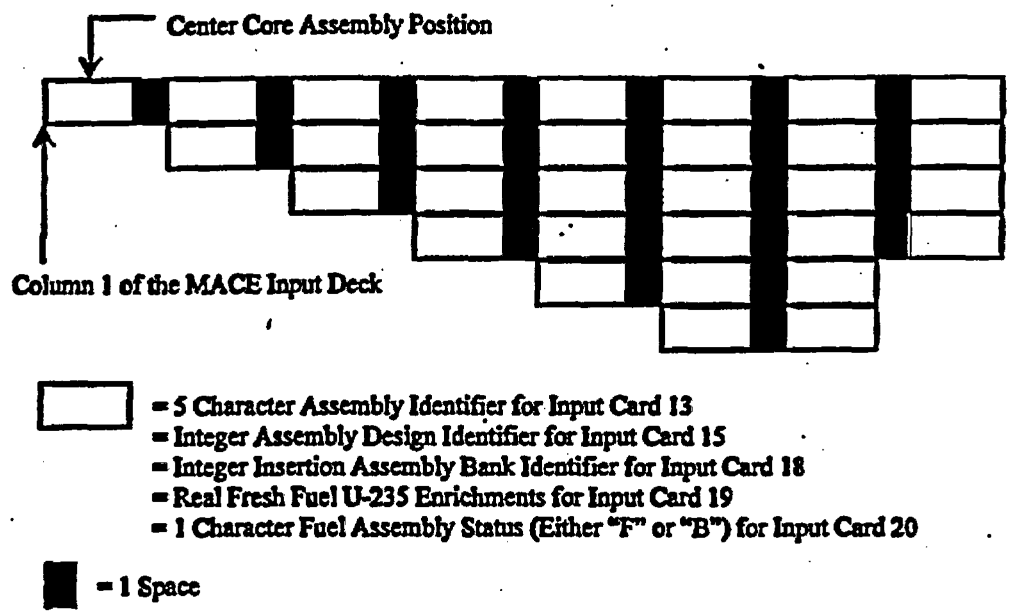


Figure 4-4. 1/8 Core Symmetric Input Specification Format for Westinghouse Reactors

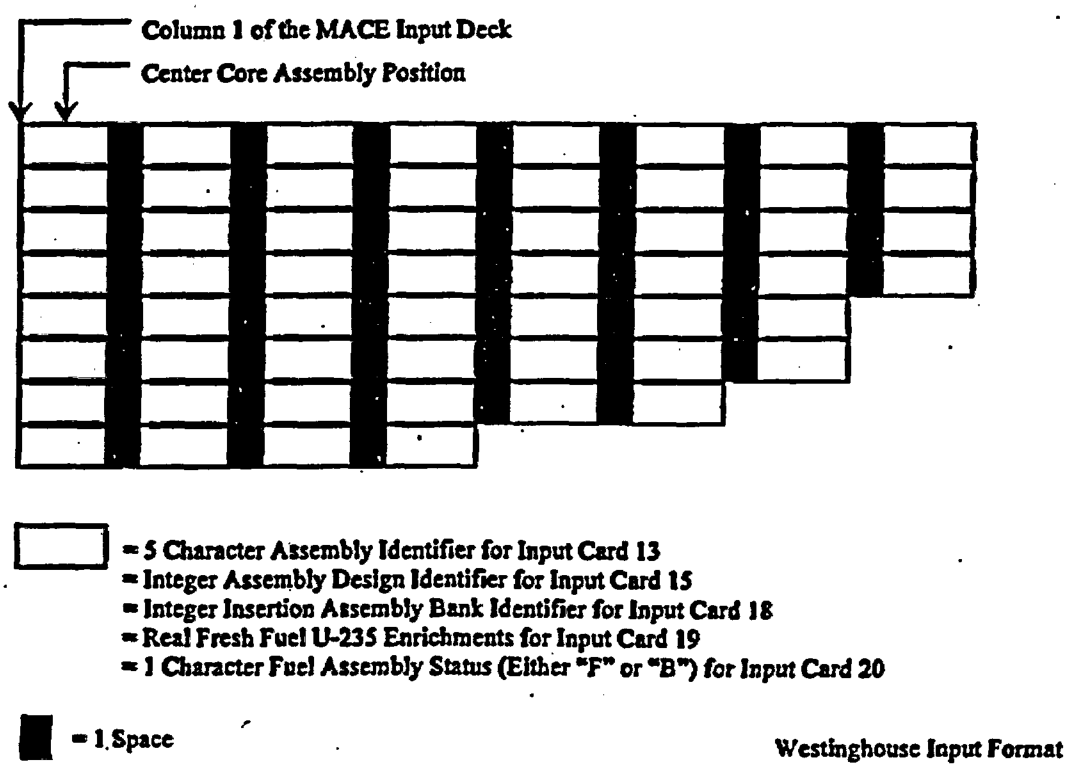


Figure 4-5. 1/4 Core Symmetric Input Specification Format for Westinghouse Reactors

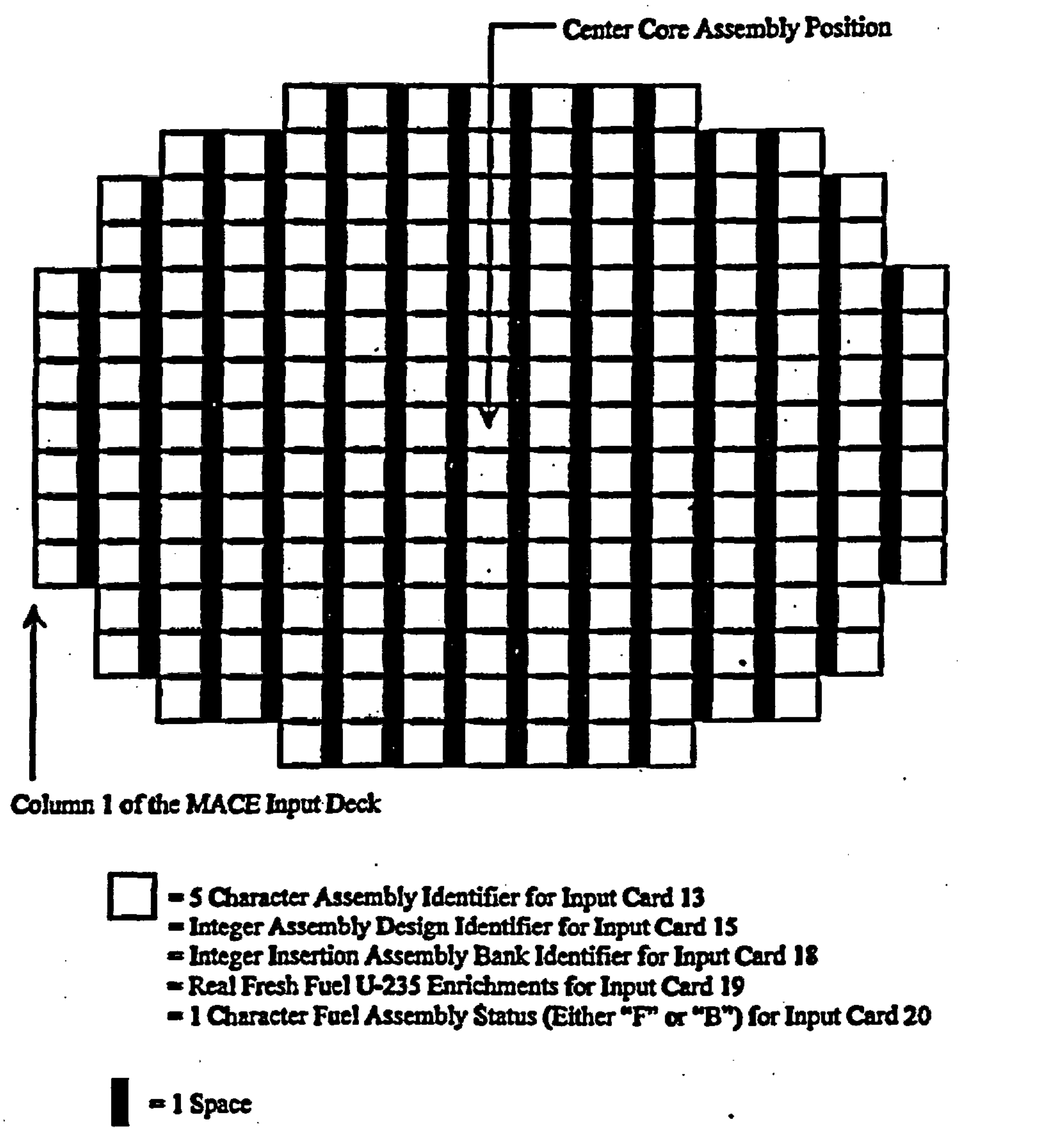


Figure 4-6. Full Core Symmetric Input Specification Format for Westinghouse Reactors

5. MACE Output Description

MACE generates a number of text files containing various sections of a complete MCNP input deck for the CRC statepoint evaluation. The following files are created in the MACE execution directory:

- "___C__T___.intro" : This file contains the introduction section of the input deck.
- "___C__T___.geo" : This file contains the geometry section of the input deck.
- "___C__T___.surf" : This file contains the surface section of the input deck.
- "___C__T___.mat" : This file contains the materials section of the input deck for all materials other than fuel and burnable poison.
- "bp.out" : This file contains the burnable poison materials section of the input deck.
- "fuel.out" : This file contains the fuel materials section of the input deck.
- "___C__T___.cont" : This file contains the calculations control and initial neutron source sections of the input deck.

The first three blanks in these filenames contain the three character reactor prefix specified in the MACE input deck. The two blanks following the "C" contain the two character identifier for the reactor cycle containing the CRC statepoint. The three blanks following the "T" contain the effective full-power day (EFPD) value of the CRC statepoint rounded to the nearest whole number.

To generate the complete MCNP input deck for the CRC statepoint evaluation, the user must concatenate the previously listed files together in the order in which they are listed. This MCNP input deck may then be plotted or executed at the user's leisure.

6. MACE Software Routine Environment and Execution

The MACE Version 3 software routine has some environmental requirements for execution. Each fuel assembly, from which depleted fuel and burnable poison isotopics will be retrieved, must have a sub-directory in the MACE execution directory. These assembly subdirectories must have the various assembly identifiers, as used in the MACE input deck as directory names. These assembly directories must contain all pertinent "*.cut" files as generated by the CRAFT software routine. The MACE input deck must be called "inputdata", and must exist in the MACE execution directory. Once these things are set up properly, MACE may be executed by simply entering the MACE executable file name.

7. MACE Test Cases

The ten CRC statepoint calculations documented in Reference 9 serve as the MACE test cases for the B&W reactor configurations. The nine statepoint calculations documented in this calculation file serve as the MACE test cases for the Westinghouse reactor configurations. Each of the MCNP input decks created by MACE to evaluate these CRC statepoint points were thoroughly checked by visual inspection and hand calculations where required. The MACE software routine works properly for the range of input parameters required for B&W and Westinghouse reactors with 15x15 and 17x17 fuel designs, respectively, as demonstrated in Reference 9 and in this analysis. The MACE software routine will need to be evaluated with additional cases when additional features are implemented to handle other reactor and/or assembly designs.

Note: Spacer grid homogenization calculations performed by MACE Version 3 are only applicable to 15x15 lattice fuel assembly designs that incorporate non-sectioned guide tubes and non-sectioned instrument tubes. The effects due to the spacer grid homogenization approximations that result from using 17x17 lattice fuel assembly designs or sectioned guide tubes or instrument tubes should be evaluated by the user.

8. References

1. *Attachment I of CRC Depletion Calculations for the Rodded Assemblies in Batches 1, 2, 3, and 1X of Crystal River Unit 3.* Document Identifier number (DI#): BBA000000-01717-0200-00040 REV 00, Civilian Radioactive Waste Management System (CRWMS) Maintenance and Operations (M&O).

2. *SCALE 4.3: Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation.* User's Manual Volumes 0 through 3, Oak Ridge National Laboratory, Document Number: CCC-545.

3. *Nuclide and Isotopes, Chart of the Nuclides, Fourteenth Edition.* General Electric Company, 1989.

4. *Software Qualification Report for MCNP, A General Monte Carlo N-Particle Transport Code.* DI#: 30006-2003 REV 02, CRWMS M&O.

5. *Radiological Health Handbook, January 1970 Revision.* Bureau of Radiological Health; U. S. Department of Health, Education, and Welfare; Public Health Service; Food and Drug Administration.

6. S. M. Bowman, O. W. Hermann, and M. C. Brady. *Scale-4 Analysis of Pressurized Water Reactor Critical Configurations: Volume 2-Sequoyah Unit 2 Cycle 3*, Oak Ridge National Laboratory, Document Number: ORNL/TM-12294/V2.

7. *Material Compositions and Number Densities for Neutronics Calculations.* DI#: B00000000-01717-0200-00002 REV 00, CRWMS M&O.

8. *Huntington Alloys: Inconel Alloy 718*, Third Edition, 1978.

9. *CRC Statepoint Reactivity Calculations for Cycles 1A, 1B, 2, 3, and 4, of Crystal River Unit 3.* DI#: B00000000-01717-0200-00046 REV 00, CRWMS M&O.

9. MACE Version 3 Fortran Source Code Listing

```
      PROGRAM MACE
*
      INTEGER DESNUM(50,50), NUMOFFADESIGNS,
     c NUMOFBANKS, BANKID(20), BANKNUM(50,50),
     c NUMOFNODES, NUMOFLOWREG,
     c BAFFLEISONUM, BARRELISONUM, SHIELDISONUM, PVCLADISONUM,
     c PVISONUM, NUMREGABOVEBPRA, NUMREGABOVECRA, NUMREGABOVEAPSRA,
     c NUMREGABOVE, NUMOFMCNPFUELNODES(20),
     c NUMOFSPACERS(20), CLADMATERIAL(20),
     c SPACERMAT(20,10), GTMAT(20), ITMAT(20), NUMDIFFBPRABANKS,
     c NUMOFBPRANODES(20), BPABSMAT(20),
     c BPNONABSMAT(20), BPRCLADMAT(20),
     c NUMDIFFCRABANKS, CRCLADMAT(20), NUMDIFFAPSRABANKS,
     c APSRCLADMAT(20), APSRFOLLOWMAT(20), SYSTEMWEST,
     c PVOUTERSURF, SYSTEMSOUTH,
     c SYSTEMTOP, SYSTEMBOTTOM, SN, ISOREQUEST,
     c RODNUM(20), MN, NUMOFGTAXS(20), GTAXMAT(20,5), NUMOFITAXS(20),
     c ITAXMAT(20,5), WBPRATYPE(20), WBPRA(20),
     c HYBFLAG(20), HYBRID,
     c NUMCRAXS(20), CRAXCLADMAT(20,20), GTSPLIT, NPERCYC, TOTCYCS,
     c GARBCYCS
*
      REAL ASSYPITCH, BANKHEIGHT(20), EFPD, NODEHEIGHT(50),
     c ENRICHMENT(50,50), LOWERREGION(10,3), LOWERREGIONWTS(10,35),
     c BAFFLEDENSITY, BAFFLEWTS(35), BAFFLETHICKNESS,
     c BAFFLESEPARATION, BARRELIR, BARRELTHICKNESS, BARRELDENSITY,
     c BARRELWTS(35), SHIELDIR, SHIELDTHICKNESS, SHIELDDENSITY,
     c SHIELDWTS(35), PVCLADIR, PVCLADTHICKNESS, PVCLADDENSITY,
     c PVCLADWTS(35), PVTHICKNESS, PVDENSITY, PVWTS(35),
     c REGABOVEBPRA(20,3), ABOVEBPRAWTS(20,35), REGABOVECRA(20,3),
     c REGABOVEAPSRA(20,3), ABOVECRAWTS(20,35), ABOVEAPSRAWTS(20,35),
     c REGABOVE(20,3), ABOVEWTS(20,35), MCNPFUELHEIGHT(20,50),
     c FUELRADIUS(20), CLADRADIUS(20,2), ASSYPLENUM(20,2),
     c ENDCAPHEIGHT(20,2), ENDFITHEIGHT(20,2), SPACERHEIGHT(20,10),
     c SPACERDIST(20,10), FRUPLENMAT(20,2), FRUPLENWTS(20,35),
     c FRLPLENMAT(20,2), FRLPLENWTS(20,35), UEFMAT(20,2),
     c UEFWTS(20,35), LEFMAT(20,2), LEFWTS(20,35), SPACERVOL(20,10),
     c GTDATA(20,4), ITDATA(20,4), MCNPBPRAHEIGHT(20,50),
     c BPRDIM(20,3), BPRPLEN(20,2), BPMATDATA(20,2),
     c BPMATWTS(20,35), B4CWTPCT(20),
     c NONBPMATDATA(20,2), NONBPMATWTS(20,35), AL2O3DENSITY(20),
     c BPRUPLENMAT(20,2), BPRUPLENWTS(20,35), BPRLPLENMAT(20,2),
     c BPRLPLENWTS(20,35), CRADIM(20,7), CRABSMAT(20,2),
     c CRABSWTS(20,35), CRLPLENMAT(20,2), CRLPLENWTS(20,35),
     c APSRADIM(20,11), APSRABSMAT(20,2), APSRABSWTS(20,35),
     c APSRUPLENMAT(20,2), APSRUPLENWTS(20,35), APSRLPLENMAT(20,2),
     c APSRLPLENWTS(20,35), FUELNODEDEN(50,50,50),
     c SURFVALUESPEC(200), BPDENTOGO(50,50,50),
     c BOTBPNODEHEIGHT(20), PINPITCH(20), PITCH,
     c MODTEMP, PRESSURE, CRUPLENMAT(20,2), APSRIPLUGDEN(20),
     c APSRIPLUGFRAC(20), CRUPLENWTS(20,35), MODDENSITY,
```

```
     c GRAMS(20), PPMB, NPLOWDEG, NPHIGHDEG, GTAXDATA(20,4,5),
     c ITAXDATA(20,4,5), BPRAXDIM(20,6),
     c CRAXDIM(20,5,20), CRABSAXMAT(20,2,20),
     c CRABSAXWTS(20,35,20), AL2O3B4CDENSITY(20),
     c SPM4SS(20,10), SPM4INC(20,10), SPM4ZR(20,10)
*
      CHARACTER ASSYID(50,50)*5, BANKDES(20)*5, REACTOR*21, PREFIX*3,
     c CYCLE*2, LOWERREGIONZAIDS(10,35)*9, BAFFLEZAIDS(35)*9,
     c BARRELZAIDS(35)*9, SHIELDZAIDS(35)*9, PVCLADZAIDS(35)*9,
     c PVZAIDS(35)*9, ABOVEBPRAZAIDS(20,35)*9,
     c ABOVECRAZAIDS(20,35)*9, ABOVEAPSRAZAIDS(20,35)*9,
     c ABOVEZAIDS(20,35)*9, FRUPLENZAIDS(20,35)*9,
     c FRLPLENZAIDS(20,35)*9, UEFZAIDS(20,35)*9, LEFZAIDS(20,35)*9,
     c NONABSBP(20)*1, BPRABSNODE(20,50)*1, BPMATZAIDS(20,35)*9,
     c NONBPMATZAIDS(20,35)*9, BPRUPLENZAIDS(20,35)*9,
     c BPRLPLENZAIDS(20,35)*9, CRABSZAIDS(20,35)*9,
     c CRLPLENZAIDS(20,35)*9, APSRABSZAIDS(20,35)*9,
     c APSRUPLENZAIDS(20,35)*9, APSRLPLENZAIDS(20,35)*9,
     c INTROFILE*16, GEOFILE*14, SURFTYPESPEC(200)*2,
     c CRUPZS(20,35)*9, SURFFILE*15,
     c STAT(50,50)*1, MATFILE*14, CONTFILE*15,
     c FDENPREF*1, CRABSAXZAIDS(20,35,20)
*
      LOGICAL BANDW, WESTINGHOUSE, CE, EIGHTH, QUARTER, FULL,
     c BOC, EIGHTYFOURSET, PRINCIPALSET, PRINCACTSET,
     c ACTONLYSET
*
*     DATA/
*
*     Call the INPUTDATA subroutine to retrieve necessary input
*     from the input deck with the filename "orders".
      CALL INPUTDATA (BANDW, WESTINGHOUSE, CE, EIGHTH,
     c QUARTER, FULL, BOC, EIGHTYFOURSET, PRINCIPALSET,
     c PRINCACTSET, ACTONLYSET, DESNUM, NUMOFFADESIGNS,
     c NUMOFBANKS, BANKID, BANKNUM, NUMOFNODES,
     c NUMOFLOWREG, BAFFLEISONUM, BARRELISONUM, SHIELDISONUM,
     c PVCLADISONUM, PVISONUM, NUMREGABOVEBPRA, NUMREGABOVECRA,
     c NUMREGABOVEAPSRA, NUMREGABOVE,
     c NUMOFMCNPFUELNODES, NUMOFSPACERS, CLADMATERIAL,
     c SPACERMAT, GTMAT, ITMAT, NUMDIFFBPRABANKS,
     c NUMOFBPRANODES, BPABSMAT, BPNONABSMAT,
     c BPRCLADMAT, NUMDIFFCRABANKS, CRCLADMAT,
     c NUMDIFFAPSRABANKS, APSRCLADMAT, APSRFOLLOWMAT,
     c ASSYPITCH, BANKHEIGHT, EFPD, NODEHEIGHT,
     c ENRICHMENT, LOWERREGION, LOWERREGIONWTS,
     c BAFFLEDENSITY, BAFFLEWTS, BAFFLETHICKNESS,
     c BAFFLESEPARATION, BARRELIR, BARRELTHICKNESS, BARRELDENSITY,
     c BARRELWTS, SHIELDIR, SHIELDTHICKNESS, SHIELDDENSITY,
     c SHIELDWTS, PVCLADIR, PVCLADTHICKNESS, PVCLADDENSITY,
     c PVCLADWTS, PVTHICKNESS, PVDENSITY, PVWTS,
     c REGABOVEBPRA, ABOVEBPRAWTS, REGABOVECRA,
     c REGABOVEAPSRA, ABOVECRAWTS, ABOVEAPSRAWTS,
     c REGABOVE, ABOVEWTS, MCNPFUELHEIGHT,
     c FUELRADIUS, CLADRADIUS, ASSYPLENUM,
```

```
     c ENDCAPHEIGHT, ENDFITHEIGHT, SPACERHEIGHT,
     c SPACERDIST, FRUPLENMAT, FRUPLENWTS,
     c FRLPLENMAT, FRLPLENWTS, UEFMAT,
     c UEFWTS, LEFMAT, LEFWTS, SPACERVOL,
     c GTDATA, ITDATA, MCNPBPRAHEIGHT,
     c BPRDIM, BPRPLEN, BPMATDATA,
     c BPMATWTS, AL2O3B4CDENSITY, B4CWTPCT,
     c NONBPMATDATA, NONBPMATWTS, AL2O3DENSITY,
     c BPRUPLENMAT, BPRUPLENWTS, BPRLPLENMAT,
     c BPRLPLENWTS, CRADIM, CRABSMAT,
     c CRABSWTS, CRLPLENMAT, CRUPLENMAT, CRLPLENWTS,
     c CRUPLENWTS, APSRADIM, APSRABSMAT, APSRABSWTS,
     c APSRUPLENMAT, APSRUPLENWTS, APSRLPLENMAT,
     c APSRLPLENWTS, ASSYID, BANKDES,
     c REACTOR, PREFIX, CYCLE, LOWERREGIONZAIDS,
     c BAFFLEZAIDS, BARRELZAIDS, SHIELDZAIDS,
     c PVCLADZAIDS, PVZAIDS, ABOVEBPRAZAIDS,
     c ABOVECRAZAIDS, ABOVEAPSRAZAIDS,
     c ABOVEZAIDS, FRUPLENZAIDS,
     c FRLPLENZAIDS, UEFZAIDS, LEFZAIDS,
     c NONABSBP, BPRABSNODE, BPMATZAIDS,
     c NONBPMATZAIDS, BPRUPLENZAIDS,
     c BPRLPLENZAIDS, CRABSZAIDS,
     c CRLPLENZAIDS, CRUPZS, APSRABSZAIDS,
     c APSRUPLENZAIDS, APSRLPLENZAIDS,
     c BOTBPNODEHEIGHT, PINPITCH, MODTEMP, PRESSURE,
     c APSRIPLUGDEN, APSRIPLUGFRAC, ISOREQUEST,
     c GRAMS, STAT, RODNUM, PPMB, NPERCYC, TOTCYCS,
     c GARBCYCS, FDENPREF, NPLOWDEG, NPHIGHDEG,
     c GTAXDATA, ITAXDATA, BPRAXDIM,
     c CRAXDIM, CRABSAXMAT, CRABSAXWTS, NUMOFGTAXS,
     c GTAXMAT, NUMOFITAXS, ITAXMAT, WBPRATYPE,
     c WBPRA, HYBRID, HYBFLAG, NUMCRAXS, CRAXCLADMAT,
     c CRABSAXZAIDS, GTSPLIT, SPM4SS, SPM4INC, SPM4ZR)
*     Call the MODDEN subroutine to calculate the moderator
*     density (g/cc) from the fuel temperature (F) and the
*     system pressure (psi).
      CALL MODDEN (MODTEMP, PRESSURE, MODDENSITY)
*     Call the INTROSECTION subroutine to write the introduction
*     section of the MCNP input decks.
      CALL INTROSECTION(BANDW, WESTINGHOUSE, CE, EIGHTH,
     c QUARTER, FULL, PREFIX, REACTOR, CYCLE, EFPD, NUMOFNODES,
     c NUMOFMCNPFUELNODES, INTROFILE)
*     Call the FUEL subroutine to retrieve and write the fuel
*     and burnable poison compositions needed in the MCNP input deck.
      CALL FUEL (ASSYID, ISOREQUEST, CYCLE, EFPD,
     c NUMOFNODES, NODEHEIGHT, FUELRADIUS, DESNUM, GRAMS,
     c ENRICHMENT, RODNUM, STAT, BANKDES, BANKNUM,
     c BPRABSNODE, B4CWTPCT, AL2O3B4CDENSITY,
     c BPRDIM, FUELNODEDEN, BPDENTOGO, FDENPREF,
     c BPABSMAT, BPMATDATA, BPMATZAIDS, BPMATWTS,
     c BPNONABSMAT, NONBPMATDATA, NONBPMATZAIDS,
     c NONBPMATWTS, NONABSBP, BANDW, WESTINGHOUSE,
     c WBPRA, WBPRATYPE, BPRAXDIM, NUMOFBPRANODES)
```

```
*     Call the GEOSECTION subroutine to write the geometry
*     specification section of the MCNP input decks.
      CALL GEOSECTION (NUMREGABOVE, NUMOFLOWREG,
    c SYSTEMWEST, BANKNUM,
    c DESNUM, NUMOFMCNPFUELNODES,
    c CLADMATERIAL, NUMREGABOVEBPRA, NUMOFBANKS,
    c BANKID, NUMREGABOVECRA, NUMREGABOVEAPSRA,
    c NUMOFSPACERS,
    c EFPD, SURFVALUESPEC, PVCLADIR, PVCLADTHICKNESS,
    c PVTHICKNESS, REGABOVE, SPACERDIST,
    c ENDFITHEIGHT, LOWERREGION, PVDENSITY,
    c PVCLADDENSITY, SHIELDIR, SHIELDTHICKNESS,
    c MODDENSITY, SHIELDDENSITY, BARRELIR, BARRELTHICKNESS,
    c BARRELDENSITY, PINPITCH, MCNPFUELHEIGHT,
    c FUELRADIUS, CLADRADIUS, ASSYPLENUM,
    c ENDCAPHEIGHT, FRUPLENMAT, FRLPLENMAT,
    c LEFMAT, UEFMAT, REGABOVEBPRA,
    c REGABOVECRA, REGABOVEAPSRA,
    c SPACERHEIGHT,
    c PREFIX, CYCLE, GEOFILE, SURFTYPESPEC,
    c ASSYID, BANKDES,
    c EIGHTH, QUARTER, FULL, BANDW, WESTINGHOUSE, CE,
    c BOTBPNODEHEIGHT, MCNPBPRAHEIGHT,
    c BPRDIM, BPRPLEN, BPRCLADMAT,
    c GTDATA, GTMAT, NUMOFBPRANODES,
    c BPRUPLENMAT, BPRLPLENMAT, ITDATA,
    c ITMAT, CRADIM, CRABSMAT,
    c CRUPLENMAT, CRLPLENMAT, CRCLADMAT, APSRADIM,
    c APSRABSMAT, APSRCLADMAT, APSRUPLENMAT, APSRLPLENMAT,
    c APSRIPLUGDEN, APSRIPLUGFRAC,
    c BAFFLEDENSITY, PVOUTERSURF, SYSTEMSOUTH, SYSTEMTOP,
    c SYSTEMBOTTOM, SN, ASSYPITCH, BAFFLESEPARATION,
    c BAFFLETHICKNESS, NUMOFNODES, ISOREQUEST,
    c NODEHEIGHT,
    c GRAMS, ENRICHMENT, STAT, RODNUM, MN,
    c LOWERREGIONZAIDS, SHIELDWTS, LOWERREGIONWTS,
    c UEFWTS, BAFFLEZAIDS, PVZAIDS, LEFZAIDS, FRLPLENZAIDS,
    c SHIELDZAIDS, PVCLADZAIDS, ABOVEWTS, FRUPLENZAIDS,
    c PVCLADWTS, FRUPLENWTS, LEFWTS, PVWTS, BARRELWTS,
    c ABOVEZAIDS, BARRELZAIDS, BAFFLEWTS, FRLPLENWTS,
    c UEFZAIDS, SPACERMAT, SPACERVOL, PPMB, PVISONUM,
    c PVCLADISONUM, SHIELDISONUM, BARRELISONUM, BAFFLEISONUM,
    c CRABSZAIDS, CRABSWTS, CRUPZS, CRUPLENWTS,
    c CRLPLENZAIDS, CRLPLENWTS, APSRUPLENWTS, APSRLPLENZAIDS,
    c APSRUPLENZAIDS, APSRLPLENWTS, APSRABSWTS, APSRABSZAIDS,
    c BPRLPLENZAIDS, BPRUPLENZAIDS, BPRLPLENWTS, BPRUPLENWTS,
    c ABOVEBPRAZAIDS, ABOVEBPRAWTS, ABOVECRAZAIDS, ABOVECRAWTS,
    c ABOVEAPSRAZAIDS, ABOVEAPSRAWTS, FUELNODEDEN, BPDENTOGO,
    c BPRABSNODE, BPNONABSMAT, NONBPMATDATA, AL2O3DENSITY,
    c PITCH, MATFILE, NUMOFFADESIGNS, GTAXDATA, GTSPLIT,
    c NUMOFGTAXS, GTAXMAT, ITAXDATA, ITSPLIT,
    c NUMOFITAXS, ITAXMAT, CRAXDIM, CRABSAXMAT,
    c CRABSAXWTS, HYBRID, HYBFLAG, NUMCRAXS, CRAXCLADMAT,
    c CRABSAXZAIDS, BPRAXDIM, WBPRATYPE, WBPRA,
```

```
     c AL2O3B4CDENSITY, SPM4SS, SPM4INC, SPM4ZR)
*     Call the SURFSECTION subroutine to write the surface
*     specification section of the MCNP input decks.
      CALL SURFSECTION (SN, EFPD, SURFVALUESPEC, SURFTYPESPEC,
     c PREFIX, CYCLE, SURFFILE, PVOUTERSURF, SYSTEMSOUTH,
     c SYSTEMWEST, SYSTEMTOP, SYSTEMBOTTOM, NPLOWDEG, NPHIGHDEG)
*     Call the CONTROL subroutine to write the control and
*     initial source specification for this MCNP calculation.
      CALL CONTROL (NPERCYC, TOTCYCS, GARBCYCS,
     c PITCH, ASSYPITCH, NUMOFNODES, NODEHEIGHT,
     c BANDW, WESTINGHOUSE, CE, EIGHTH, QUARTER, FULL,
     c PREFIX, CYCLE, EFPD, CONTFILE)

      END

*********************************************************************
*     SUBROUTINE INPUTDATA                                          *
*     This subroutine retrieves necessary input data from a well-   *
*     defined input file called "inputdata".                        *
*********************************************************************
      SUBROUTINE INPUTDATA (BANDW, WESTINGHOUSE, CE, EIGHTH,
     c QUARTER, FULL, BOC, EIGHTYFOURSET, PRINCIPALSET,
     c PRINCACTSET, ACTONLYSET, DESNUM, NUMOFFADESIGNS,
     c NUMOFBANKS, BANKID, BANKNUM, NUMOFNODES,
     c NUMOFLOWREG, BAFFLEISONUM, BARRELISONUM, SHIELDISONUM,
     c PVCLADISONUM, PVISONUM, NUMREGABOVEBPRA, NUMREGABOVECRA,
     c NUMREGABOVEAPSRA, NUMREGABOVE,
     c NUMOFMCNPFUELNODES, NUMOFSPACERS, CLADMATERIAL,
     c SPACERMAT, GTMAT, ITMAT, NUMDIFFBPRABANKS,
     c NUMOFBPRANODES, BPABSMAT, BPNONABSMAT,
     c BPRCLADMAT, NUMDIFFCRABANKS, CRCLADMAT,
     c NUMDIFFAPSRABANKS, APSRCLADMAT, APSRFOLLOWMAT,
     c ASSYPITCH, BANKHEIGHT, EFPD, NODEHEIGHT,
     c ENRICHMENT, LOWERREGION, LOWERREGIONWTS,
     c BAFFLEDENSITY, BAFFLEWTS, BAFFLETHICKNESS,
     c BAFFLESEPARATION, BARRELIR, BARRELTHICKNESS, BARRELDENSITY,
     c BARRELWTS, SHIELDIR, SHIELDTHICKNESS, SHIELDDENSITY,
     c SHIELDWTS, PVCLADIR, PVCLADTHICKNESS, PVCLADDENSITY,
     c PVCLADWTS, PVTHICKNESS, PVDENSITY, PVWTS,
     c REGABOVEBPRA, ABOVEBPRAWTS, REGABOVECRA,
     c REGABOVEAPSRA, ABOVECRAWTS, ABOVEAPSRAWTS,
     c REGABOVE, ABOVEWTS, MCNPFUELHEIGHT,
     c FUELRADIUS, CLADRADIUS, ASSYPLENUM,
     c ENDCAPHEIGHT, ENDFITHEIGHT, SPACERHEIGHT,
     c SPACERDIST, FRUPLENMAT, FRUPLENWTS,
     c FRLPLENMAT, FRLPLENWTS, UEFMAT,
     c UEFWTS, LEFMAT, LEFWTS, SPACERVOL,
     c GTDATA, ITDATA, MCNPBPRAHEIGHT,
     c BPRDIM, BPRPLEN, BPMATDATA,
     c BPMATWTS, AL2O3B4CDENSITY, B4CWTPCT,
     c NONBPMATDATA, NONBPMATWTS, AL2O3DENSITY,
     c BPRUPLENMAT, BPRUPLENWTS, BPRLPLENMAT,
     c BPRLPLENWTS, CRADIM, CRABSMAT,
     c CRABSWTS, CRLPLENMAT, CRUPLENMAT, CRLPLENWTS,
```

```
     c CRUPLENWTS, APSRADIM, APSRABSMAT, APSRABSWTS,
     c APSRUPLENMAT, APSRUPLENWTS, APSRLPLENMAT,
     c APSRLPLENWTS, ASSYID, BANKDES,
     c REACTOR, PREFIX, CYCLE, LOWERREGIONZAIDS,
     c BAFFLEZAIDS, BARRELZAIDS, SHIELDZAIDS,
     c PVCLADZAIDS, PVZAIDS, ABOVEBPRAZAIDS,
     c ABOVECRAZAIDS, ABOVEAPSRAZAIDS,
     c ABOVEZAIDS, FRUPLENZAIDS,
     c FRLPLENZAIDS, UEFZAIDS, LEFZAIDS,
     c NONABSBP, BPRABSNODE, BPMATZAIDS,
     c NONBPMATZAIDS, BPRUPLENZAIDS,
     c BPRLPLENZAIDS, CRABSZAIDS,
     c CRLPLENZAIDS, CRUPZS, APSRABSZAIDS,
     c APSRUPLENZAIDS, APSRLPLENZAIDS,
     c BOTBPNODEHEIGHT, PINPITCH, MODTEMP, PRESSURE,
     c APSRIPLUGDEN, APSRIPLUGFRAC, ISOREQUEST, GRAMS, STAT,
     c RODNUM, PPMB, NPERCYC, TOTCYCS, GARBCYCS, FDENPREF,
     c NPLOWDEG, NPHIGHDEG,
     c GTAXDATA, ITAXDATA, BPRAXDIM,
     c CRAXDIM, CRABSAXMAT, CRABSAXWTS, NUMOFGTAXS,
     c GTAXMAT, NUMOFITAXS, ITAXMAT, WBPRATYPE,
     c WBPRA, HYBRID, HYBFLAG, NUMCRAXS, CRAXCLADMAT,
     c CRABSAXZAIDS, GTSPLIT, SPM4SS, SPM4INC, SPM4ZR)
*
      INTEGER COREDESIGN, SYMMETRY, DESNUM(50,50), NUMOFFADESIGNS,
     c C, R, NUMOFBANKS, BANKID(20), BANKNUM(50,50), BANK,
     c NUMOFNODES, NODE, NODENUMBER, NUMOFLOWREG, REGION, ISOTOPE,
     c BAFFLEISONUM, BARRELISONUM, SHIELDISONUM, PVCLADISONUM,
     c PVISONUM, NUMREGABOVEBPRA, NUMREGABOVECRA, NUMREGABOVEAPSRA,
     c NUMREGABOVE, DESIGN, NUMOFMCNPFUELNODES(20),
     c NUMOFSPACERS(20), SPACERNUMBER, CLADMATERIAL(20),
     c SPACERMAT(20,10), GTMAT(20), ITMAT(20), NUMDIFFBPRABANKS,
     c CURRENTBANK, NUMOFBPRANODES(20), BPABSMAT(20),
     c BPNONABSMAT(20), BPRCLADMAT(20), BANKIDS,
     c NUMDIFFCRABANKS, CRCLADMAT(20), NUMDIFFAPSRABANKS,
     c APSRCLADMAT(20), APSRFOLLOWMAT(20), ISOREQUEST, RODNUM(20),
     c GTSPLIT, NUMOFGTAXS(20), GTAX,
     c GTAXMAT(20,5), ITSPLIT, NUMOFITAXS(20), ITAX, ITAXMAT(20,5),
     c WBPRATYPE(20), WBPRA(20), HYBRID, HYBFLAG(20),
     c NUMCRAXS(20), CRAXCLADMAT(20,20), NPERCYC, TOTCYCS, GARBCYCS,
     c I
*
      REAL ASSYPITCH, BANKHEIGHT(20), EFPD, NODEHEIGHT(50),
     c ENRICHMENT(50,50), LOWERREGION(10,3), LOWERREGIONWTS(10,35),
     c BAFFLEDENSITY, BAFFLEWTS(35), BAFFLETHICKNESS,
     c BAFFLESEPARATION, BARRELIR, BARRELTHICKNESS, BARRELDENSITY,
     c BARRELWTS(35), SHIELDIR, SHIELDTHICKNESS, SHIELDDENSITY,
     c SHIELDWTS(35), PVCLADIR, PVCLADTHICKNESS, PVCLADDENSITY,
     c PVCLADWTS(35), PVTHICKNESS, PVDENSITY, PVWTS(35),
     c REGABOVEBPRA(20,3), ABOVEBPRAWTS(20,35), REGABOVECRA(20,3),
     c REGABOVEAPSRA(20,3), ABOVECRAWTS(20,35), ABOVEAPSRAWTS(20,35),
     c REGABOVE(20,3), ABOVEWTS(20,35), MCNPFUELHEIGHT(20,50),
     c FUELRADIUS(20), CLADRADIUS(20,2), ASSYPLENUM(20,2),
     c ENDCAPHEIGHT(20,2), ENDFITHEIGHT(20,2), SPACERHEIGHT(20,10),
```

```
     c SPACERDIST(20,10), FRUPLENMAT(20,2), FRUPLENWTS(20,35),
     c FRLPLENMAT(20,2), FRLPLENWTS(20,35), UEFMAT(20,2),
     c UEFWTS(20,35), LEFMAT(20,2), LEFWTS(20,35), SPACERVOL(20,10),
     c GTDATA(20,4), ITDATA(20,4), MCNPBPRAHEIGHT(20,50),
     c BPRDIM(20,3), BPRPLEN(20,2), BPMATDATA(20,2),
     c BPMATWTS(20,35), AL2O3B4CDENSITY(20), B4CWTPCT(20),
     c NONBPMATDATA(20,2), NONBPMATWTS(20,35), AL2O3DENSITY(20),
     c BPRUPLENMAT(20,2), BPRUPLENWTS(20,35), BPRLPLENMAT(20,2),
     c BPRLPLENWTS(20,35), CRADIM(20,7), CRABSMAT(20,2),
     c CRABSWTS(20,35), CRLPLENMAT(20,2), CRLPLENWTS(20,35),
     c CRUPLENMAT(20,2), CRUPLENWTS(20,35),
     c APSRADIM(20,11), APSRABSMAT(20,2), APSRABSWTS(20,35),
     c APSRUPLENMAT(20,2), APSRUPLENWTS(20,35), APSRLPLENMAT(20,2),
     c APSRLPLENWTS(20,35), BOTBPNODEHEIGHT(20), PINPITCH(20),
     c MODTEMP, PRESSURE, APSRIPLUGDEN(20), APSRIPLUGFRAC(20),
     c GRAMS(20), PPMB, NPLOWDEG, NPHIGHDEG, GTAXDATA(20,4,5),
     c ITAXDATA(20,4,5), BPRAXDIM(20,6),
     c CRAXDIM(20,5,20), CRABSAXMAT(20,2,20),
     c CRABSAXWTS(20,35,20), SPM4SS(20,10), SPM4INC(20,10),
     c SPM4ZR(20,10), X
*
      CHARACTER ASSYID(50,50)*5, BANKDES(20)*5, REACTOR*21, PREFIX*3,
     c CYCLE*2, LOWERREGIONZAIDS(10,35)*9, BAFFLEZAIDS(35)*9,
     c BARRELZAIDS(35)*9, SHIELDZAIDS(35)*9, PVCLADZAIDS(35)*9,
     c PVZAIDS(35)*9, ABOVEBPRAZAIDS(20,35)*9,
     c ABOVECRAZAIDS(20,35)*9, ABOVEAPSRAZAIDS(20,35)*9,
     c ABOVEZAIDS(20,35)*9, FRUPLENZAIDS(20,35)*9,
     c FRLPLENZAIDS(20,35)*9, UEFZAIDS(20,35)*9, LEFZAIDS(20,35)*9,
     c NONABSBP(20)*1, BPRABSNODE(20,50)*1, BPMATZAIDS(20,35)*9,
     c NONBPMATZAIDS(20,35)*9, BPRUPLENZAIDS(20,35)*9,
     c BPRLPLENZAIDS(20,35)*9, CRABSZAIDS(20,35)*9,
     c CRLPLENZAIDS(20,35)*9, APSRABSZAIDS(20,35)*9,
     c APSRUPLENZAIDS(20,35)*9, APSRLPLENZAIDS(20,35)*9,
     c CRUPZS(20,35)*9, STAT(50,50)*1, FDENPREF*1,
     c CRABSAXZAIDS(20,35,20)
*
      LOGICAL BANDW, WESTINGHOUSE, CE, EIGHTH, QUARTER, FULL, BOC,
     c EIGHTYFOURSET, PRINCIPALSET, PRINCACTSET, ACTONLYSET
*
*     Open the "inputdata" file
      OPEN(UNIT=15,FILE='inputdata',STATUS='OLD')
      REWIND (UNIT=15)
*     Retrieve core layout description information.
      READ(15,*) COREDESIGN
      BANDW=.FALSE.
      WESTINGHOUSE=.FALSE.
      CE=.FALSE.
      IF (COREDESIGN.EQ.1) THEN
         BANDW=.TRUE.
      ELSEIF (COREDESIGN.EQ.2) THEN
         WESTINGHOUSE=.TRUE.
      ELSEIF (COREDESIGN.EQ.3) THEN
         CE=.TRUE.
      ELSE
```

```
         WRITE (*,*) 'THE CORE DESIGN MUST BE EITHER',
     c   'B&W, WESTINGHOUSE, OR CE.'
         STOP
      ENDIF
      READ(15,*) SYMMETRY
      EIGHTH=.FALSE.
      QUARTER=.FALSE.
      FULL=.FALSE.
      IF (SYMMETRY.EQ.1) THEN
         EIGHTH=.TRUE.
      ELSEIF (SYMMETRY.EQ.2) THEN
         QUARTER=.TRUE.
      ELSEIF (SYMMETRY.EQ.3) THEN
         FULL=.TRUE.
      ELSE
         WRITE (*,*) 'THE CORE SYMMETRY MUST BE EITHER',
     c   'EIGHTH, QUARTER, OR FULL.'
         STOP
      ENDIF
*     Read in the reactor name (maximum of 21 characters)
      READ (15,1) REACTOR
    1 FORMAT(T1,A21)
*     Read in the reactor prefix (maximum of 3 characters)
      READ (15,2) PREFIX
    2 FORMAT(T1,A3)
*     Read in cycle identifier for CRC statepoint calculation.
      READ(15,3) CYCLE
    3 FORMAT(T1,A2)
*     Read in the statepoint EFPD value for the CRC calculation.
      READ(15,*) EFPD
      BOC=.FALSE.
      IF (EFPD.EQ.(0.0)) THEN
         BOC=.TRUE.
      ENDIF
*     Read in the fuel isotopic composition request.
      EIGHTYFOURSET=.FALSE.
      PRINCIPALSET=.FALSE.
      PRINCACTSET=.FALSE.
      ACTONLYSET=.FALSE.
      READ(15,*) ISOREQUEST
      IF (ISOREQUEST.EQ.1) THEN
         EIGHTYFOURSET=.TRUE.
      ELSEIF (ISOREQUEST.EQ.2) THEN
         PRINCIPALSET=.TRUE.
      ELSEIF (ISOREQUEST.EQ.3) THEN
         PRINCACTSET=.TRUE.
      ELSEIF (ISOREQUEST.EQ.4) THEN
         ACTONLYSET=.TRUE.
      ENDIF
*     Read in the fuel density preference
      READ(15,8) FDENPREF
    8 FORMAT(T1,A1)
*     Read in the control specifcations
      READ(15,*) NPERCYC, GARBCYCS, TOTCYCS
```

```
*     Read in the number of axial fuel nodes in the CRC calculation.
      READ(15,*) NUMOFNODES
*     Read in the axial node format from the top node to the bottom node.
      DO 4 NODE=1,NUMOFNODES     ! Node number 1 is the top node.
         READ(15,*) NODENUMBER, NODEHEIGHT(NODE)
    4 CONTINUE
      READ(15,*) ASSYPITCH
      READ(15,*) MODTEMP, PRESSURE, PPMB
      DO 6 C=1,50
         DO 5 R=1,50
            ASSYID(C,R)='     '
            BANKNUM(C,R)=0
            DESNUM(C,R)=0
    5    CONTINUE
    6 CONTINUE
      DO 7 C=1,20
         BANKDES(C)='     '
    7 CONTINUE
      IF (BANDW.EQ..TRUE.) THEN
         IF (EIGHTH.EQ..TRUE.) THEN
*     Read in the fuel assembly archive identifiers for retrieval of
isotopics.
            READ(15,10) ASSYID(1,1), ASSYID(2,1), ASSYID(3,1),
     c      ASSYID(4,1), ASSYID(5,1), ASSYID(6,1), ASSYID(7,1),
     c      ASSYID(8,1)
   10       FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c      A5,1X,A5)
            READ(15,20) ASSYID(2,2), ASSYID(3,2), ASSYID(4,2),
     c      ASSYID(5,2), ASSYID(6,2), ASSYID(7,2), ASSYID(8,2)
   20       FORMAT(T1,5X,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c      A5,1X,A5)
            READ(15,30) ASSYID(3,3), ASSYID(4,3), ASSYID(5,3),
     c      ASSYID(6,3), ASSYID(7,3), ASSYID(8,3)
   30       FORMAT(T1,5X,1X,5X,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c      A5,1X,A5)
            READ(15,40) ASSYID(4,4), ASSYID(5,4), ASSYID(6,4),
     c      ASSYID(7,4)
   40       FORMAT(T1,5X,1X,5X,1X,5X,1X,A5,1X,A5,1X,A5,1X,A5)
            READ(15,50) ASSYID(5,5), ASSYID(6,5), ASSYID(7,5)
   50       FORMAT(T1,5X,1X,5X,1X,5X,1X,5X,1X,A5,1X,A5,1X,A5)
            READ(15,60) ASSYID(6,6)
   60       FORMAT(T1,5X,1X,5X,1X,5X,1X,5X,1X,5X,1X,A5)
*     Read in the number of different fuel assembly designs to be included
*     in the MCNP calculation.
            READ(15,*) NUMOFFADESIGNS
*     Read in the fuel assembly relative design designations.
            READ(15,*) DESNUM(1,1), DESNUM(2,1), DESNUM(3,1),
     c      DESNUM(4,1), DESNUM(5,1), DESNUM(6,1), DESNUM(7,1),
     c      DESNUM(8,1)
            READ(15,*) DESNUM(2,2), DESNUM(3,2), DESNUM(4,2),
     c      DESNUM(5,2), DESNUM(6,2), DESNUM(7,2), DESNUM(8,2)
            READ(15,*) DESNUM(3,3), DESNUM(4,3), DESNUM(5,3),
     c      DESNUM(6,3), DESNUM(7,3), DESNUM(8,3)
            READ(15,*) DESNUM(4,4), DESNUM(5,4), DESNUM(6,4),
```

```
     c        DESNUM(7,4)
              READ(15,*) DESNUM(5,5), DESNUM(6,5), DESNUM(7,5)
              READ(15,*) DESNUM(6,6)
              DO 80 C=1,50
                 DO 70 R=1,50
                    IF (DESNUM(C,R).GT.NUMOFFADESIGNS) THEN
                       WRITE(*,*) 'THE FUEL ASSEMBLY DESIGN NUMBER ',
     c                 'SPECIFIED FOR THE ASSEMBLY IN RELATIVE POSITION ',
     c                 C,',',R,' IS LARGER THAN THE NUMBER OF FUEL ',
     c                 'ASSEMBLY DESIGNS SPECIFIED.'
                       STOP
                    ENDIF
   70            CONTINUE
   80         CONTINUE
*     Read in the number of different insertion rod assembly bank
designations and
*     bank insertion heights for the statepoint calculation.  The insertion
height
*     values should be the distances (cm) between the bottom of the absorber
material
*     in the insertion rods and the bottom of the active fuel region.
              READ(15,*) NUMOFBANKS
              DO 100 BANK=1,NUMOFBANKS
                 READ(15,90) BANKID(BANK), BANKDES(BANK),
     c           BANKHEIGHT(BANK)
   90            FORMAT(T1,I2,1X,A5,1X,F7.3)
  100         CONTINUE
*     Read in the insertion rod assembly core layout.
              READ(15,*) BANKNUM(1,1), BANKNUM(2,1), BANKNUM(3,1),
     c        BANKNUM(4,1), BANKNUM(5,1), BANKNUM(6,1), BANKNUM(7,1),
     c        BANKNUM(8,1)
              READ(15,*) BANKNUM(2,2), BANKNUM(3,2), BANKNUM(4,2),
     c        BANKNUM(5,2), BANKNUM(6,2), BANKNUM(7,2), BANKNUM(8,2)
              READ(15,*) BANKNUM(3,3), BANKNUM(4,3), BANKNUM(5,3),
     c        BANKNUM(6,3), BANKNUM(7,3), BANKNUM(8,3)
              READ(15,*) BANKNUM(4,4), BANKNUM(5,4), BANKNUM(6,4),
     c        BANKNUM(7,4)
              READ(15,*) BANKNUM(5,5), BANKNUM(6,5), BANKNUM(7,5)
              READ(15,*) BANKNUM(6,6)
*     Read in initial enrichments.
              READ(15,*) ENRICHMENT(1,1), ENRICHMENT(2,1),
     c        ENRICHMENT(3,1), ENRICHMENT(4,1), ENRICHMENT(5,1),
     c        ENRICHMENT(6,1), ENRICHMENT(7,1), ENRICHMENT(8,1)
              READ(15,*) ENRICHMENT(2,2), ENRICHMENT(3,2),
     c        ENRICHMENT(4,2), ENRICHMENT(5,2), ENRICHMENT(6,2),
     c        ENRICHMENT(7,2), ENRICHMENT(8,2)
              READ(15,*) ENRICHMENT(3,3), ENRICHMENT(4,3),
     c        ENRICHMENT(5,3), ENRICHMENT(6,3), ENRICHMENT(7,3),
     c        ENRICHMENT(8,3)
              READ(15,*) ENRICHMENT(4,4), ENRICHMENT(5,4),
     c        ENRICHMENT(6,4), ENRICHMENT(7,4)
              READ(15,*) ENRICHMENT(5,5), ENRICHMENT(6,5),
     c        ENRICHMENT(7,5)
              READ(15,*) ENRICHMENT(6,6)
```

```
*     Read in fuel status (fresh or burned).
              READ(15,101) STAT(1,1), STAT(2,1),
     c        STAT(3,1), STAT(4,1), STAT(5,1),
     c        STAT(6,1), STAT(7,1), STAT(8,1)
  101         FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
              READ(15,102) STAT(2,2), STAT(3,2),
     c        STAT(4,2), STAT(5,2), STAT(6,2),
     c        STAT(7,2), STAT(8,2)
  102         FORMAT(T3,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
              READ(15,103) STAT(3,3), STAT(4,3),
     c        STAT(5,3), STAT(6,3), STAT(7,3),
     c        STAT(8,3)
  103         FORMAT(T5,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
              READ(15,104) STAT(4,4), STAT(5,4),
     c        STAT(6,4), STAT(7,4)
  104         FORMAT(T7,A1,1X,A1,1X,A1,1X,A1)
              READ(15,105) STAT(5,5), STAT(6,5),
     c        STAT(7,5)
  105         FORMAT(T9,A1,1X,A1,1X,A1)
              READ(15,106) STAT(6,6)
  106         FORMAT(T11,A1)
          ELSEIF (QUARTER.EQ..TRUE.) THEN
*     Read in the fuel assembly archive identifiers for retrieval of
isotopics.
              READ(15,110) ASSYID(1,1), ASSYID(2,1), ASSYID(3,1),
     c        ASSYID(4,1), ASSYID(5,1), ASSYID(6,1), ASSYID(7,1),
     c        ASSYID(8,1)
  110         FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c        A5,1X,A5)
              READ(15,120) ASSYID(1,2), ASSYID(2,2), ASSYID(3,2),
     c        ASSYID(4,2), ASSYID(5,2), ASSYID(6,2), ASSYID(7,2),
     c        ASSYID(8,2)
  120         FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c        A5,1X,A5)
              READ(15,130) ASSYID(1,3), ASSYID(2,3), ASSYID(3,3),
     c        ASSYID(4,3), ASSYID(5,3), ASSYID(6,3), ASSYID(7,3),
     c        ASSYID(8,3)
  130         FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c        A5,1X,A5)
              READ(15,140) ASSYID(1,4), ASSYID(2,4), ASSYID(3,4),
     c        ASSYID(4,4), ASSYID(5,4), ASSYID(6,4), ASSYID(7,4)
  140         FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5)
              READ(15,150) ASSYID(1,5), ASSYID(2,5), ASSYID(3,5),
     c        ASSYID(4,5), ASSYID(5,5), ASSYID(6,5), ASSYID(7,5)
  150         FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5)
              READ(15,160) ASSYID(1,6), ASSYID(2,6), ASSYID(3,6),
     c        ASSYID(4,6), ASSYID(5,6), ASSYID(6,6)
  160         FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5)
              READ(15,170) ASSYID(1,7), ASSYID(2,7), ASSYID(3,7),
     c        ASSYID(4,7), ASSYID(5,7)
  170         FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5)
              READ(15,180) ASSYID(1,8), ASSYID(2,8), ASSYID(3,8)
  180         FORMAT(T1,A5,1X,A5,1X,A5)
*     Read in the number of different fuel assembly designs to be included
```

```
*     in the MCNP calculation.
            READ(15,*) NUMOFFADESIGNS
*     Read in the fuel assembly relative design designations.
            READ(15,*) DESNUM(1,1), DESNUM(2,1), DESNUM(3,1),
     c      DESNUM(4,1), DESNUM(5,1), DESNUM(6,1), DESNUM(7,1),
     c      DESNUM(8,1)
            READ(15,*) DESNUM(1,2), DESNUM(2,2), DESNUM(3,2),
     c      DESNUM(4,2), DESNUM(5,2), DESNUM(6,2), DESNUM(7,2),
     c      DESNUM(8,2)
            READ(15,*) DESNUM(1,3), DESNUM(2,3), DESNUM(3,3),
     c      DESNUM(4,3), DESNUM(5,3), DESNUM(6,3), DESNUM(7,3),
     c      DESNUM(8,3)
            READ(15,*) DESNUM(1,4), DESNUM(2,4), DESNUM(3,4),
     c      DESNUM(4,4), DESNUM(5,4), DESNUM(6,4), DESNUM(7,4)
            READ(15,*) DESNUM(1,5), DESNUM(2,5), DESNUM(3,5),
     c      DESNUM(4,5), DESNUM(5,5), DESNUM(6,5), DESNUM(7,5)
            READ(15,*) DESNUM(1,6), DESNUM(2,6), DESNUM(3,6),
     c      DESNUM(4,6), DESNUM(5,6), DESNUM(6,6)
            READ(15,*) DESNUM(1,7), DESNUM(2,7), DESNUM(3,7),
     c      DESNUM(4,7), DESNUM(5,7)
            READ(15,*) DESNUM(1,8), DESNUM(2,8), DESNUM(3,8)
            DO 200 C=1,50
               DO 190 R=1,50
                  IF (DESNUM(C,R).GT.NUMOFFADESIGNS) THEN
                     WRITE(*,*) 'THE FUEL ASSEMBLY DESIGN NUMBER ',
     c               'SPECIFIED FOR THE ASSEMBLY IN RELATIVE POSITION ',
     c               C,',',R,' IS LARGER THAN THE NUMBER OF FUEL ',
     c               'ASSEMBLY DESIGNS SPECIFIED.'
                     STOP
                  ENDIF
  190          CONTINUE
  200       CONTINUE
*     Read in the number of different insertion rod assembly bank
designations and
*     bank insertion heights for the statepoint calculation.  The insertion
height
*     values should be the distances (cm) between the bottom of the absorber
material
*     in the insertion rods and the bottom of the active fuel region.
            READ(15,*) NUMOFBANKS
            DO 220 BANK=1,NUMOFBANKS
               READ(15,210) BANKID(BANK), BANKDES(BANK),
     c         BANKHEIGHT(BANK)
  210          FORMAT(T1,I2,1X,A5,1X,F7.3)
  220       CONTINUE
*     Read in the insertion rod assembly core layout.
            READ(15,*) BANKNUM(1,1), BANKNUM(2,1), BANKNUM(3,1),
     c      BANKNUM(4,1), BANKNUM(5,1), BANKNUM(6,1), BANKNUM(7,1),
     c      BANKNUM(8,1)
            READ(15,*) BANKNUM(1,2), BANKNUM(2,2), BANKNUM(3,2),
     c      BANKNUM(4,2), BANKNUM(5,2), BANKNUM(6,2), BANKNUM(7,2),
     c      BANKNUM(8,2)
            READ(15,*) BANKNUM(1,3), BANKNUM(2,3), BANKNUM(3,3),
     c      BANKNUM(4,3), BANKNUM(5,3), BANKNUM(6,3), BANKNUM(7,3),
```

```
     c      BANKNUM(8,3)
            READ(15,*) BANKNUM(1,4), BANKNUM(2,4), BANKNUM(3,4),
     c      BANKNUM(4,4), BANKNUM(5,4), BANKNUM(6,4),
     c      BANKNUM(7,4)
            READ(15,*) BANKNUM(1,5), BANKNUM(2,5), BANKNUM(3,5),
     c      BANKNUM(4,5), BANKNUM(5,5), BANKNUM(6,5),
     c      BANKNUM(7,5)
            READ(15,*) BANKNUM(1,6), BANKNUM(2,6), BANKNUM(3,6),
     c      BANKNUM(4,6), BANKNUM(5,6), BANKNUM(6,6)
            READ(15,*) BANKNUM(1,7), BANKNUM(2,7), BANKNUM(3,7),
     c      BANKNUM(4,7), BANKNUM(5,7)
            READ(15,*) BANKNUM(1,8), BANKNUM(2,8), BANKNUM(3,8)
*    Read in initial enrichments if it is a BOC or BOL case.
               READ(15,*) ENRICHMENT(1,1), ENRICHMENT(2,1),
     c         ENRICHMENT(3,1), ENRICHMENT(4,1), ENRICHMENT(5,1),
     c         ENRICHMENT(6,1), ENRICHMENT(7,1), ENRICHMENT(8,1)
               READ(15,*) ENRICHMENT(1,2), ENRICHMENT(2,2),
     c         ENRICHMENT(3,2), ENRICHMENT(4,2), ENRICHMENT(5,2),
     c         ENRICHMENT(6,2), ENRICHMENT(7,2), ENRICHMENT(8,2)
               READ(15,*) ENRICHMENT(1,3), ENRICHMENT(2,3),
     c         ENRICHMENT(3,3), ENRICHMENT(4,3), ENRICHMENT(5,3),
     c         ENRICHMENT(6,3), ENRICHMENT(7,3), ENRICHMENT(8,3)
               READ(15,*) ENRICHMENT(1,4), ENRICHMENT(2,4),
     c         ENRICHMENT(3,4), ENRICHMENT(4,4), ENRICHMENT(5,4),
     c         ENRICHMENT(6,4), ENRICHMENT(7,4)
               READ(15,*) ENRICHMENT(1,5), ENRICHMENT(2,5),
     c         ENRICHMENT(3,5), ENRICHMENT(4,5), ENRICHMENT(5,5),
     c         ENRICHMENT(6,5), ENRICHMENT(7,5)
               READ(15,*) ENRICHMENT(1,6), ENRICHMENT(2,6),
     c         ENRICHMENT(3,6), ENRICHMENT(4,6), ENRICHMENT(5,6),
     c         ENRICHMENT(6,6)
               READ(15,*) ENRICHMENT(1,7), ENRICHMENT(2,7),
     c         ENRICHMENT(3,7), ENRICHMENT(4,7), ENRICHMENT(5,7)
               READ(15,*) ENRICHMENT(1,8), ENRICHMENT(2,8),
     c         ENRICHMENT(3,8)
*    Read in fuel status (fresh or burned).
            READ(15,221) STAT(1,1), STAT(2,1),
     c      STAT(3,1), STAT(4,1), STAT(5,1),
     c      STAT(6,1), STAT(7,1), STAT(8,1)
 221        FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,222) STAT(1,2), STAT(2,2),
     c      STAT(3,2), STAT(4,2), STAT(5,2),
     c      STAT(6,2), STAT(7,2), STAT(8,2)
 222        FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,223) STAT(1,3), STAT(2,3),
     c      STAT(3,3), STAT(4,3), STAT(5,3),
     c      STAT(6,3), STAT(7,3), STAT(8,3)
 223        FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,224) STAT(1,4), STAT(2,4),
     c      STAT(3,4), STAT(4,4), STAT(5,4),
     c      STAT(6,4), STAT(7,4)
 224        FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,225) STAT(1,5), STAT(2,5),
     c      STAT(3,5), STAT(4,5), STAT(5,5),
```

```
     c        STAT(6,5), STAT(7,5)
   225        FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
              READ(15,226) STAT(1,6), STAT(2,6),
     c        STAT(3,6), STAT(4,6), STAT(5,6),
     c        STAT(6,6)
   226        FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
              READ(15,227) STAT(1,7), STAT(2,7),
     c        STAT(3,7), STAT(4,7), STAT(5,7)
   227        FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1)
              READ(15,228) STAT(1,8), STAT(2,8),
     c        STAT(3,8)
   228        FORMAT(T1,A1,1X,A1,1X,A1)
            ELSEIF (FULL.EQ..TRUE.) THEN
*     Read in the fuel assembly archive identifiers for retrieval of
isotopics.
              READ(15,230) ASSYID(6,1), ASSYID(7,1), ASSYID(8,1),
     c        ASSYID(9,1), ASSYID(10,1)
   230        FORMAT(T31,A5,1X,A5,1X,A5,1X,A5,1X,A5)
              READ(15,240) ASSYID(4,2), ASSYID(5,2), ASSYID(6,2),
     c        ASSYID(7,2), ASSYID(8,2), ASSYID(9,2), ASSYID(10,2),
     c        ASSYID(11,2), ASSYID(12,2)
   240        FORMAT(T19,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c        A5,1X,A5,1X,A5)
              READ(15,250) ASSYID(3,3), ASSYID(4,3), ASSYID(5,3),
     c        ASSYID(6,3), ASSYID(7,3), ASSYID(8,3), ASSYID(9,3),
     c        ASSYID(10,3), ASSYID(11,3), ASSYID(12,3), ASSYID(13,3)
   250        FORMAT(T13,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c        A5,1X,A5,1X,A5,1X,A5,1X,A5)
              READ(15,260) ASSYID(2,4), ASSYID(3,4), ASSYID(4,4),
     c        ASSYID(5,4), ASSYID(6,4), ASSYID(7,4), ASSYID(8,4),
     c        ASSYID(9,4), ASSYID(10,4), ASSYID(11,4), ASSYID(12,4),
     c        ASSYID(13,4), ASSYID(14,4)
   260        FORMAT(T7,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c        A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5)
              READ(15,270) ASSYID(2,5), ASSYID(3,5), ASSYID(4,5),
     c        ASSYID(5,5), ASSYID(6,5), ASSYID(7,5), ASSYID(8,5),
     c        ASSYID(9,5), ASSYID(10,5), ASSYID(11,5), ASSYID(12,5),
     c        ASSYID(13,5), ASSYID(14,5)
   270        FORMAT(T7,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c        A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5)
              READ(15,280) ASSYID(1,6), ASSYID(2,6), ASSYID(3,6),
     c        ASSYID(4,6), ASSYID(5,6), ASSYID(6,6), ASSYID(7,6),
     c        ASSYID(8,6), ASSYID(9,6), ASSYID(10,6), ASSYID(11,6),
     c        ASSYID(12,6), ASSYID(13,6), ASSYID(14,6), ASSYID(15,6)
   280        FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c        A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5)
              READ(15,290) ASSYID(1,7), ASSYID(2,7), ASSYID(3,7),
     c        ASSYID(4,7), ASSYID(5,7), ASSYID(6,7), ASSYID(7,7),
     c        ASSYID(8,7), ASSYID(9,7), ASSYID(10,7), ASSYID(11,7),
     c        ASSYID(12,7), ASSYID(13,7), ASSYID(14,7), ASSYID(15,7)
   290        FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c        A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5)
              READ(15,300) ASSYID(1,8), ASSYID(2,8), ASSYID(3,8),
     c        ASSYID(4,8), ASSYID(5,8), ASSYID(6,8), ASSYID(7,8),
```

```
     c        DESNUM(5,4), DESNUM(6,4), DESNUM(7,4), DESNUM(8,4),
     c        DESNUM(9,4), DESNUM(10,4), DESNUM(11,4), DESNUM(12,4),
     c        DESNUM(13,4), DESNUM(14,4)
              READ(15,*) DESNUM(2,5), DESNUM(3,5), DESNUM(4,5),
     c        DESNUM(5,5), DESNUM(6,5), DESNUM(7,5), DESNUM(8,5),
     c        DESNUM(9,5), DESNUM(10,5), DESNUM(11,5), DESNUM(12,5),
     c        DESNUM(13,5), DESNUM(14,5)
              READ(15,*) DESNUM(1,6), DESNUM(2,6), DESNUM(3,6),
     c        DESNUM(4,6), DESNUM(5,6), DESNUM(6,6), DESNUM(7,6),
     c        DESNUM(8,6), DESNUM(9,6), DESNUM(10,6), DESNUM(11,6),
     c        DESNUM(12,6), DESNUM(13,6), DESNUM(14,6), DESNUM(15,6)
              READ(15,*) DESNUM(1,7), DESNUM(2,7), DESNUM(3,7),
     c        DESNUM(4,7), DESNUM(5,7), DESNUM(6,7), DESNUM(7,7),
     c        DESNUM(8,7), DESNUM(9,7), DESNUM(10,7), DESNUM(11,7),
     c        DESNUM(12,7), DESNUM(13,7), DESNUM(14,7), DESNUM(15,7)
              READ(15,*) DESNUM(1,8), DESNUM(2,8), DESNUM(3,8),
     c        DESNUM(4,8), DESNUM(5,8), DESNUM(6,8), DESNUM(7,8),
     c        DESNUM(8,8), DESNUM(9,8), DESNUM(10,8), DESNUM(11,8),
     c        DESNUM(12,8), DESNUM(13,8), DESNUM(14,8), DESNUM(15,8)
              READ(15,*) DESNUM(1,9), DESNUM(2,9), DESNUM(3,9),
     c        DESNUM(4,9), DESNUM(5,9), DESNUM(6,9), DESNUM(7,9),
     c        DESNUM(8,9), DESNUM(9,9), DESNUM(10,9), DESNUM(11,9),
     c        DESNUM(12,9), DESNUM(13,9), DESNUM(14,9), DESNUM(15,9)
              READ(15,*) DESNUM(1,10), DESNUM(2,10), DESNUM(3,10),
     c        DESNUM(4,10), DESNUM(5,10), DESNUM(6,10), DESNUM(7,10),
     c        DESNUM(8,10), DESNUM(9,10), DESNUM(10,10), DESNUM(11,10),
     c        DESNUM(12,10), DESNUM(13,10), DESNUM(14,10), DESNUM(15,10)
              READ(15,*) DESNUM(2,11), DESNUM(3,11), DESNUM(4,11),
     c        DESNUM(5,11), DESNUM(6,11), DESNUM(7,11), DESNUM(8,11),
     c        DESNUM(9,11), DESNUM(10,11), DESNUM(11,11), DESNUM(12,11),
     c        DESNUM(13,11), DESNUM(14,11)
              READ(15,*) DESNUM(2,12), DESNUM(3,12), DESNUM(4,12),
     c        DESNUM(5,12), DESNUM(6,12), DESNUM(7,12), DESNUM(8,12),
     c        DESNUM(9,12), DESNUM(10,12), DESNUM(11,12), DESNUM(12,12),
     c        DESNUM(13,12), DESNUM(14,12)
              READ(15,*) DESNUM(3,13), DESNUM(4,13), DESNUM(5,13),
     c        DESNUM(6,13), DESNUM(7,13), DESNUM(8,13), DESNUM(9,13),
     c        DESNUM(10,13), DESNUM(11,13), DESNUM(12,13), DESNUM(13,13)
              READ(15,*) DESNUM(4,14), DESNUM(5,14), DESNUM(6,14),
     c        DESNUM(7,14), DESNUM(8,14), DESNUM(9,14), DESNUM(10,14),
     c        DESNUM(11,14), DESNUM(12,14)
              READ(15,*) DESNUM(6,15), DESNUM(7,15), DESNUM(8,15),
     c        DESNUM(9,15), DESNUM(10,15)
              DO 390 C=1,50
                 DO 380 R=1,50
                    IF (DESNUM(C,R).GT.NUMOFFADESIGNS) THEN
                       WRITE(*,*) 'THE FUEL ASSEMBLY DESIGN NUMBER ',
     c                 'SPECIFIED FOR THE ASSEMBLY IN POSITION ',
     c                 C,',',R,' IS LARGER THAN THE NUMBER OF FUEL ',
     c                 'ASSEMBLY DESIGNS SPECIFIED.'
                       STOP
                    ENDIF
 380             CONTINUE
 390          CONTINUE
```

```
*     Read in the number of different insertion rod assembly bank
designations and
*     bank insertion heights for the statepoint calculation.  The insertion
height
*     values should be the distances (cm) between the bottom of the absorber
material
*     in the insertion rods and the bottom of the active fuel region.
         READ(15,*) NUMOFBANKS
         DO 392 BANK=1,NUMOFBANKS
            READ(15,391) BANKID(BANK), BANKDES(BANK),
     c      BANKHEIGHT(BANK)
 391        FORMAT(T1,I2,1X,A5,1X,F7.3)
 392     CONTINUE
*     Read in the insertion rod assembly core layout.
         READ(15,*) BANKNUM(6,1), BANKNUM(7,1), BANKNUM(8,1),
     c   BANKNUM(9,1), BANKNUM(10,1)
         READ(15,*).BANKNUM(4,2), BANKNUM(5,2), BANKNUM(6,2),
     c   BANKNUM(7,2), BANKNUM(8,2), BANKNUM(9,2), BANKNUM(10,2),
     c   BANKNUM(11,2), BANKNUM(12,2)
         READ(15,*) BANKNUM(3,3), BANKNUM(4,3), BANKNUM(5,3),
     c   BANKNUM(6,3), BANKNUM(7,3), BANKNUM(8,3), BANKNUM(9,3),
     c   BANKNUM(10,3), BANKNUM(11,3), BANKNUM(12,3), BANKNUM(13,3)
         READ(15,*) BANKNUM(2,4), BANKNUM(3,4), BANKNUM(4,4),
     c   BANKNUM(5,4), BANKNUM(6,4), BANKNUM(7,4), BANKNUM(8,4),
     c   BANKNUM(9,4), BANKNUM(10,4), BANKNUM(11,4), BANKNUM(12,4),
     c   BANKNUM(13,4), BANKNUM(14,4)
         READ(15,*) BANKNUM(2,5), BANKNUM(3,5), BANKNUM(4,5),
     c   BANKNUM(5,5), BANKNUM(6,5), BANKNUM(7,5), BANKNUM(8,5),
     c   BANKNUM(9,5), BANKNUM(10,5), BANKNUM(11,5), BANKNUM(12,5),
     c   BANKNUM(13,5), BANKNUM(14,5)
         READ(15,*) BANKNUM(1,6), BANKNUM(2,6), BANKNUM(3,6),
     c   BANKNUM(4,6), BANKNUM(5,6), BANKNUM(6,6), BANKNUM(7,6),
     c   BANKNUM(8,6), BANKNUM(9,6), BANKNUM(10,6), BANKNUM(11,6),
     c   BANKNUM(12,6), BANKNUM(13,6), BANKNUM(14,6), BANKNUM(15,6)
         READ(15,*) BANKNUM(1,7), BANKNUM(2,7), BANKNUM(3,7),
     c   BANKNUM(4,7), BANKNUM(5,7), BANKNUM(6,7), BANKNUM(7,7),
     c   BANKNUM(8,7), BANKNUM(9,7), BANKNUM(10,7), BANKNUM(11,7),
     c   BANKNUM(12,7), BANKNUM(13,7), BANKNUM(14,7), BANKNUM(15,7)
         READ(15,*) BANKNUM(1,8), BANKNUM(2,8), BANKNUM(3,8),
     c   BANKNUM(4,8), BANKNUM(5,8), BANKNUM(6,8), BANKNUM(7,8),
     c   BANKNUM(8,8), BANKNUM(9,8), BANKNUM(10,8), BANKNUM(11,8),
     c   BANKNUM(12,8), BANKNUM(13,8), BANKNUM(14,8), BANKNUM(15,8)
         READ(15,*) BANKNUM(1,9), BANKNUM(2,9), BANKNUM(3,9),
     c   BANKNUM(4,9), BANKNUM(5,9), BANKNUM(6,9), BANKNUM(7,9),
     c   BANKNUM(8,9), BANKNUM(9,9), BANKNUM(10,9), BANKNUM(11,9),
     c   BANKNUM(12,9), BANKNUM(13,9), BANKNUM(14,9), BANKNUM(15,9)
         READ(15,*) BANKNUM(1,10), BANKNUM(2,10), BANKNUM(3,10),
     c   BANKNUM(4,10), BANKNUM(5,10), BANKNUM(6,10), BANKNUM(7,10),
     c   BANKNUM(8,10), BANKNUM(9,10), BANKNUM(10,10),
     c   BANKNUM(11,10), BANKNUM(12,10), BANKNUM(13,10),
     c   BANKNUM(14,10), BANKNUM(15,10)
         READ(15,*) BANKNUM(2,11), BANKNUM(3,11), BANKNUM(4,11),
     c   BANKNUM(5,11), BANKNUM(6,11), BANKNUM(7,11), BANKNUM(8,11),
     c   BANKNUM(9,11), BANKNUM(10,11), BANKNUM(11,11),
```

```
     c      BANKNUM(12,11), BANKNUM(13,11), BANKNUM(14,11)
            READ(15,*) BANKNUM(2,12), BANKNUM(3,12), BANKNUM(4,12),
     c      BANKNUM(5,12), BANKNUM(6,12), BANKNUM(7,12), BANKNUM(8,12),
     c      BANKNUM(9,12), BANKNUM(10,12), BANKNUM(11,12),
     c      BANKNUM(12,12), BANKNUM(13,12), BANKNUM(14,12)
            READ(15,*) BANKNUM(3,13), BANKNUM(4,13), BANKNUM(5,13),
     c      BANKNUM(6,13), BANKNUM(7,13), BANKNUM(8,13), BANKNUM(9,13),
     c      BANKNUM(10,13), BANKNUM(11,13), BANKNUM(12,13),
     c      BANKNUM(13,13)
            READ(15,*) BANKNUM(4,14), BANKNUM(5,14), BANKNUM(6,14),
     c      BANKNUM(7,14), BANKNUM(8,14), BANKNUM(9,14),
     c      BANKNUM(10,14), BANKNUM(11,14), BANKNUM(12,14)
            READ(15,*) BANKNUM(6,15), BANKNUM(7,15), BANKNUM(8,15),
     c      BANKNUM(9,15), BANKNUM(10,15)
*    Read in initial enrichments if it is a BOC or BOL case.
               READ(15,*) ENRICHMENT(6,1), ENRICHMENT(7,1),
     c         ENRICHMENT(8,1), ENRICHMENT(9,1), ENRICHMENT(10,1)
               READ(15,*) ENRICHMENT(4,2), ENRICHMENT(5,2),
     c         ENRICHMENT(6,2), ENRICHMENT(7,2), ENRICHMENT(8,2),
     c         ENRICHMENT(9,2), ENRICHMENT(10,2), ENRICHMENT(11,2),
     c         ENRICHMENT(12,2)
               READ(15,*) ENRICHMENT(3,3), ENRICHMENT(4,3),
     c         ENRICHMENT(5,3), ENRICHMENT(6,3), ENRICHMENT(7,3),
     c         ENRICHMENT(8,3), ENRICHMENT(9,3), ENRICHMENT(10,3),
     c         ENRICHMENT(11,3), ENRICHMENT(12,3), ENRICHMENT(13,3)
               READ(15,*) ENRICHMENT(2,4), ENRICHMENT(3,4),
     c         ENRICHMENT(4,4), ENRICHMENT(5,4), ENRICHMENT(6,4),
     c         ENRICHMENT(7,4), ENRICHMENT(8,4), ENRICHMENT(9,4),
     c         ENRICHMENT(10,4), ENRICHMENT(11,4), ENRICHMENT(12,4),
     c         ENRICHMENT(13,4), ENRICHMENT(14,4)
               READ(15,*) ENRICHMENT(2,5), ENRICHMENT(3,5),
     c         ENRICHMENT(4,5), ENRICHMENT(5,5), ENRICHMENT(6,5),
     c         ENRICHMENT(7,5), ENRICHMENT(8,5), ENRICHMENT(9,5),
     c         ENRICHMENT(10,5), ENRICHMENT(11,5), ENRICHMENT(12,5),
     c         ENRICHMENT(13,5), ENRICHMENT(14,5)
               READ(15,*) ENRICHMENT(1,6), ENRICHMENT(2,6),
     c         ENRICHMENT(3,6), ENRICHMENT(4,6), ENRICHMENT(5,6),
     c         ENRICHMENT(6,6), ENRICHMENT(7,6), ENRICHMENT(8,6),
     c         ENRICHMENT(9,6), ENRICHMENT(10,6), ENRICHMENT(11,6),
     c         ENRICHMENT(12,6), ENRICHMENT(13,6), ENRICHMENT(14,6),
     c         ENRICHMENT(15,6)
               READ(15,*) ENRICHMENT(1,7), ENRICHMENT(2,7),
     c         ENRICHMENT(3,7), ENRICHMENT(4,7), ENRICHMENT(5,7),
     c         ENRICHMENT(6,7), ENRICHMENT(7,7), ENRICHMENT(8,7),
     c         ENRICHMENT(9,7), ENRICHMENT(10,7), ENRICHMENT(11,7),
     c         ENRICHMENT(12,7), ENRICHMENT(13,7), ENRICHMENT(14,7),
     c         ENRICHMENT(15,7)
               READ(15,*) ENRICHMENT(1,8), ENRICHMENT(2,8),
     c         ENRICHMENT(3,8), ENRICHMENT(4,8), ENRICHMENT(5,8),
     c         ENRICHMENT(6,8), ENRICHMENT(7,8), ENRICHMENT(8,8),
     c         ENRICHMENT(9,8), ENRICHMENT(10,8), ENRICHMENT(11,8),
     c         ENRICHMENT(12,8), ENRICHMENT(13,8), ENRICHMENT(14,8),
     c         ENRICHMENT(15,8)
               READ(15,*) ENRICHMENT(1,9), ENRICHMENT(2,9),
```

```
     c          ENRICHMENT(3,9), ENRICHMENT(4,9), ENRICHMENT(5,9),
     c          ENRICHMENT(6,9), ENRICHMENT(7,9), ENRICHMENT(8,9),
     c          ENRICHMENT(9,9), ENRICHMENT(10,9), ENRICHMENT(11,9),
     c          ENRICHMENT(12,9), ENRICHMENT(13,9), ENRICHMENT(14,9),
     c          ENRICHMENT(15,9)
                READ(15,*) ENRICHMENT(1,10), ENRICHMENT(2,10),
     c          ENRICHMENT(3,10), ENRICHMENT(4,10), ENRICHMENT(5,10),
     c          ENRICHMENT(6,10), ENRICHMENT(7,10), ENRICHMENT(8,10),
     c          ENRICHMENT(9,10), ENRICHMENT(10,10), ENRICHMENT(11,10),
     c          ENRICHMENT(12,10), ENRICHMENT(13,10), ENRICHMENT(14,10),
     c          ENRICHMENT(15,10)
                READ(15,*) ENRICHMENT(2,11), ENRICHMENT(3,11),
     c          ENRICHMENT(4,11), ENRICHMENT(5,11), ENRICHMENT(6,11),
     c          ENRICHMENT(7,11), ENRICHMENT(8,11), ENRICHMENT(9,11),
     c          ENRICHMENT(10,11), ENRICHMENT(11,11), ENRICHMENT(12,11),
     c          ENRICHMENT(13,11), ENRICHMENT(14,11)
                READ(15,*) ENRICHMENT(2,12), ENRICHMENT(3,12),
     c          ENRICHMENT(4,12), ENRICHMENT(5,12), ENRICHMENT(6,12),
     c          ENRICHMENT(7,12), ENRICHMENT(8,12), ENRICHMENT(9,12),
     c          ENRICHMENT(10,12), ENRICHMENT(11,12), ENRICHMENT(12,12),
     c          ENRICHMENT(13,12), ENRICHMENT(14,12)
                READ(15,*) ENRICHMENT(3,13), ENRICHMENT(4,13),
     c          ENRICHMENT(5,13), ENRICHMENT(6,13), ENRICHMENT(7,13),
     c          ENRICHMENT(8,13), ENRICHMENT(9,13), ENRICHMENT(10,13),
     c          ENRICHMENT(11,13), ENRICHMENT(12,13), ENRICHMENT(13,13)
                READ(15,*) ENRICHMENT(4,14), ENRICHMENT(5,14),
     c          ENRICHMENT(6,14), ENRICHMENT(7,14), ENRICHMENT(8,14),
     c          ENRICHMENT(9,14), ENRICHMENT(10,14), ENRICHMENT(11,14),
     c          ENRICHMENT(12,14)
                READ(15,*) ENRICHMENT(6,15), ENRICHMENT(7,15),
     c          ENRICHMENT(8,15), ENRICHMENT(9,15), ENRICHMENT(10,15)
*     Read in fuel status (fresh or burned).
            READ(15,393) STAT(6,1), STAT(7,1), STAT(8,1),
     c      STAT(9,1), STAT(10,1)
 393        FORMAT(T11,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,394) STAT(4,2), STAT(5,2), STAT(6,2),
     c      STAT(7,2), STAT(8,2), STAT(9,2), STAT(10,2),
     c      STAT(11,2), STAT(12,2)
 394        FORMAT(T7,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,
     c      A1,1X,A1,1X,A1)
            READ(15,395) STAT(3,3), STAT(4,3), STAT(5,3),
     c      STAT(6,3), STAT(7,3), STAT(8,3), STAT(9,3),
     c      STAT(10,3), STAT(11,3), STAT(12,3), STAT(13,3)
 395        FORMAT(T5,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,
     c      A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,396) STAT(2,4), STAT(3,4), STAT(4,4),
     c      STAT(5,4), STAT(6,4), STAT(7,4), STAT(8,4),
     c      STAT(9,4), STAT(10,4), STAT(11,4), STAT(12,4),
     c      STAT(13,4), STAT(14,4)
 396        FORMAT(T3,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,
     c      1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,397) STAT(2,5), STAT(3,5), STAT(4,5),
     c      STAT(5,5), STAT(6,5), STAT(7,5), STAT(8,5),
     c      STAT(9,5), STAT(10,5), STAT(11,5), STAT(12,5),
```

```
     c         STAT(13,5), STAT(14,5)
 397           FORMAT(T3,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,
     c         1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
               READ(15,398) STAT(1,6), STAT(2,6), STAT(3,6),
     c         STAT(4,6), STAT(5,6), STAT(6,6), STAT(7,6),
     c         STAT(8,6), STAT(9,6), STAT(10,6), STAT(11,6),
     c         STAT(12,6), STAT(13,6), STAT(14,6), STAT(15,6)
 398           FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,
     c         1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
               READ(15,399) STAT(1,7), STAT(2,7), STAT(3,7),
     c         STAT(4,7), STAT(5,7), STAT(6,7), STAT(7,7),
     c         STAT(8,7), STAT(9,7), STAT(10,7), STAT(11,7),
     c         STAT(12,7), STAT(13,7), STAT(14,7), STAT(15,7)
 399           FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,
     c         1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
               READ(15,400) STAT(1,8), STAT(2,8), STAT(3,8),
     c         STAT(4,8), STAT(5,8), STAT(6,8), STAT(7,8),
     c         STAT(8,8), STAT(9,8), STAT(10,8), STAT(11,8),
     c         STAT(12,8), STAT(13,8), STAT(14,8), STAT(15,8)
 400           FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,
     c         1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
               READ(15,401) STAT(1,9), STAT(2,9), STAT(3,9),
     c         STAT(4,9), STAT(5,9), STAT(6,9), STAT(7,9),
     c         STAT(8,9), STAT(9,9), STAT(10,9), STAT(11,9),
     c         STAT(12,9), STAT(13,9), STAT(14,9), STAT(15,9)
 401           FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,
     c         1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
               READ(15,402) STAT(1,10), STAT(2,10), STAT(3,10),
     c         STAT(4,10), STAT(5,10), STAT(6,10), STAT(7,10),
     c         STAT(8,10), STAT(9,10), STAT(10,10), STAT(11,10),
     c         STAT(12,10), STAT(13,10), STAT(14,10), STAT(15,10)
 402           FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,
     c         1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
               READ(15,403) STAT(2,11), STAT(3,11), STAT(4,11),
     c         STAT(5,11), STAT(6,11), STAT(7,11), STAT(8,11),
     c         STAT(9,11), STAT(10,11), STAT(11,11), STAT(12,11),
     c         STAT(13,11), STAT(14,11)
 403           FORMAT(T3,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,
     c         1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
               READ(15,404) STAT(2,12), STAT(3,12), STAT(4,12),
     c         STAT(5,12), STAT(6,12), STAT(7,12), STAT(8,12),
     c         STAT(9,12), STAT(10,12), STAT(11,12), STAT(12,12),
     c         STAT(13,12), STAT(14,12)
 404           FORMAT(T3,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,
     c         1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
               READ(15,405) STAT(3,13), STAT(4,13), STAT(5,13),
     c         STAT(6,13), STAT(7,13), STAT(8,13), STAT(9,13),
     c         STAT(10,13), STAT(11,13), STAT(12,13), STAT(13,13)
 405           FORMAT(T5,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,
     c         A1,1X,A1,1X,A1,1X,A1,1X,A1)
               READ(15,406) STAT(4,14), STAT(5,14), STAT(6,14),
     c         STAT(7,14), STAT(8,14), STAT(9,14), STAT(10,14),
     c         STAT(11,14), STAT(12,14)
 406           FORMAT(T7,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,
```

```
     c        A1,1X,A1,1X,A1)
              READ(15,407) STAT(6,15), STAT(7,15), STAT(8,15),
     c        STAT(9,15), STAT(10,15)
 407          FORMAT(T11,A1,1X,A1,1X,A1,1X,A1,1X,A1)
           ENDIF
        ELSEIF (WESTINGHOUSE.EQ..TRUE.) THEN
           CALL WESTONE(EIGHTH, QUARTER, FULL, ASSYID, NUMOFFADESIGNS,
     c     DESNUM, NUMOFBANKS, BANKDES, BANKID, BANKHEIGHT, BANKNUM,
     c     ENRICHMENT, STAT)
        ELSEIF (CE.EQ..TRUE.) THEN
           WRITE(*,*) 'The CE reactor option is not available in',
     c     ' this version of the code.'
           STOP
        ENDIF
*       Read in the number of regions below the lower end-fittings.
        READ(15,*) NUMOFLOWREG
*       Read in the information for the regions below the lower end-fittings.
        DO 440 REGION=1,NUMOFLOWREG
           READ(15,*) LOWERREGION(REGION,1),  ! Height of region (cm)
     c     LOWERREGION(REGION,2),             ! Density (g/cc) of lower region
     c     LOWERREGION(REGION,3)              ! Number of isotopes in region
           DO 430 ISOTOPE=1,LOWERREGION(REGION,3)
              READ(15,420) LOWERREGIONZAIDS(REGION,ISOTOPE), ! MCNP ZAID
     c        LOWERREGIONWTS(REGION,ISOTOPE) ! Isotope wt %
 420          FORMAT(T1,A9,1X,G15.13)
 430       CONTINUE
 440    CONTINUE
*       Read in the distance between the core baffle and the outer edge of
*       the fuel assembly outer unit cell boundary.
        READ(15,*) BAFFLESEPARATION
*       Read in the core baffle thickness (cm), density (g/cc), and
*       number of isotopes in the baffle material composition.
        READ(15,*) BAFFLETHICKNESS, BAFFLEDENSITY, BAFFLEISONUM
           DO 460 ISOTOPE=1,BAFFLEISONUM
              READ(15,450) BAFFLEZAIDS(ISOTOPE), ! MCNP ZAID
     c        BAFFLEWTS(ISOTOPE) ! Isotope wt %
 450          FORMAT(T1,A9,1X,G15.13)
 460       CONTINUE
*       Read in the core barrel inner radius from the center of the core.
        READ(15,*) BARRELIR
*       Read in the core barrel thickness (cm), density (g/cc), and
*       number of isotopes in the barrel material composition.
        READ(15,*) BARRELTHICKNESS, BARRELDENSITY, BARRELISONUM
           DO 480 ISOTOPE=1,BARRELISONUM
              READ(15,470) BARRELZAIDS(ISOTOPE), ! MCNP ZAID
     c        BARRELWTS(ISOTOPE) ! Isotope wt %
 470          FORMAT(T1,A9,1X,G15.13)
 480       CONTINUE
        IF (BANDW.EQ..TRUE.) THEN
*       Read in the thermal shield inner radius from the center of the core.
        READ(15,*) SHIELDIR
*       Read in the thermal shield thickness (cm), density (g/cc), and
*       number of isotopes in the thermal shield material composition.
        READ(15,*) SHIELDTHICKNESS, SHIELDDENSITY, SHIELDISONUM
```

```
          DO 492 ISOTOPE=1,SHIELDISONUM
             READ(15,490) SHIELDZAIDS(ISOTOPE), ! MCNP ZAID
     c       SHIELDWTS(ISOTOPE) ! Isotope wt %
  490        FORMAT(T1,A9,1X,G15.13)
  492     CONTINUE
       ELSEIF (WESTINGHOUSE.EQ..TRUE.) THEN
*      Read in the neutron pad inner radius from the center of the core.
       READ(15,*) SHIELDIR
*      Read in the neutron pad thickness (cm), density (g/cc), and
*      number of isotopes in the neutron pad material composition.
       READ(15,*) SHIELDTHICKNESS, SHIELDDENSITY, SHIELDISONUM
          DO 496 ISOTOPE=1,SHIELDISONUM
             READ(15,494) SHIELDZAIDS(ISOTOPE), ! MCNP ZAID
     c       SHIELDWTS(ISOTOPE) ! Isotope wt %
  494        FORMAT(T1,A9,1X,G15.13)
  496     CONTINUE
*      Read in the neutron pad bounding angles in degrees
*      (lowest to highest) for the neutron pad in the northeast quadrant
*      of the reactor core where the y-axis is 0 degrees.
       READ(15,*) NPLOWDEG, NPHIGHDEG
       ENDIF
*      Read in the pressure vessel cladding inner radius from the center of
the core.
       READ(15,*) PVCLADIR
*      Read in the pressure vessel cladding thickness (cm), density (g/cc),
and
*      number of isotopes in the pressure vessel cladding material
composition.
       READ(15,*) PVCLADTHICKNESS, PVCLADDENSITY, PVCLADISONUM
          DO 515 ISOTOPE=1,PVCLADISONUM
             READ(15,510) PVCLADZAIDS(ISOTOPE), ! MCNP ZAID
     c       PVCLADWTS(ISOTOPE) ! Isotope wt %
  510        FORMAT(T1,A9,1X,G15.13)
  515     CONTINUE
*      Read in the pressure vessel thickness (cm), density (g/cc), and
*      number of isotopes in the pressure vessel material composition.
       READ(15,*) PVTHICKNESS, PVDENSITY, PVISONUM
          DO 525 ISOTOPE=1,PVISONUM
             READ(15,520) PVZAIDS(ISOTOPE), ! MCNP ZAID
     c       PVWTS(ISOTOPE) ! Isotope wt %
  520        FORMAT(T1,A9,1X,G15.13)
  525     CONTINUE
       IF (BANDW.EQ..TRUE.) THEN
*      If BPRAs are present, read in the descriptions of the axial zones
*      above the upper end-fitting of the fuel assembly containing the BPRA.
       DO 560 BANK=1,NUMOFBANKS
          IF (BANKDES(BANK).EQ.'BPRA ') THEN
             READ(15,*) NUMREGABOVEBPRA
             DO 550 REGION=1,NUMREGABOVEBPRA ! Region 1 is the top-most axial
region
                READ(15,*) REGABOVEBPRA(REGION,1), ! Height (cm) of region
above BPRA
     c          REGABOVEBPRA(REGION,2),            ! Density (g/cc) of region
above BPRA
```

```
     c          REGABOVEBPRA(REGION,3)              ! Number of isotopes in
region above BPRA
                DO 540 ISOTOPE=1, REGABOVEBPRA(REGION,3)
                   READ(15,530) ABOVEBPRAZAIDS(REGION,ISOTOPE), ! MCNP ZAID
     c             ABOVEBPRAWTS(REGION,ISOTOPE)
  530              FORMAT(T1,A9,1X,G15.13)
  540           CONTINUE
  550        CONTINUE
             EXIT
          ENDIF
  560 CONTINUE
*     If CRAs are present, read in the descriptions of the axial zones
*     above the upper end-fitting of the fuel assembly containing the CRA.
      DO 600 BANK=1,NUMOFBANKS
          IF (BANKDES(BANK).EQ.'CRA  ') THEN
             READ(15,*) NUMREGABOVECRA
             DO 590 REGION=1,NUMREGABOVECRA
                READ(15,*) REGABOVECRA(REGION,1),   ! Height (cm) of region
above CRA
     c          REGABOVECRA(REGION,2),              ! Density (g/cc) of region
above CRA
     c          REGABOVECRA(REGION,3)               ! Number of isotopes in
region above CRA
                DO 580 ISOTOPE=1, REGABOVECRA(REGION,3)
                   READ(15,570) ABOVECRAZAIDS(REGION,ISOTOPE), ! MCNP ZAID
     c             ABOVECRAWTS(REGION,ISOTOPE)
  570              FORMAT(T1,A9,1X,G15.13)
  580           CONTINUE
  590        CONTINUE
             EXIT
          ENDIF
  600 CONTINUE
*     If APSRAs are present, read in the descriptions of the axial zones
*     above the upper end-fitting of the fuel assembly containing the APSRA.
      DO 640 BANK=1,NUMOFBANKS
          IF (BANKDES(BANK).EQ.'APSRA') THEN
             READ(15,*) NUMREGABOVEAPSRA
             DO 630 REGION=1,NUMREGABOVEAPSRA
                READ(15,*) REGABOVEAPSRA(REGION,1),   ! Height (cm) of region
above APSRA
     c          REGABOVEAPSRA(REGION,2),              ! Density (g/cc) of
region above APSRA
     c          REGABOVEAPSRA(REGION,3)               ! Number of isotopes in
region above APSRA
                DO 620 ISOTOPE=1, REGABOVEAPSRA(REGION,3)
                   READ(15,610) ABOVEAPSRAZAIDS(REGION,ISOTOPE), ! MCNP ZAID
     c             ABOVEAPSRAWTS(REGION,ISOTOPE)
  610              FORMAT(T1,A9,1X,G15.13)
  620           CONTINUE
  630        CONTINUE
             EXIT
          ENDIF
  640 CONTINUE
*     Read in the descriptions of the axial zones above the upper end-fitting
```

```
*     of the fuel assembly with no insertion rod assembly.
      READ(15,*) NUMREGABOVE
      DO 670 REGION=1,NUMREGABOVE
         READ(15,*) REGABOVE(REGION,1),   ! Height (cm) of region above assy
     c   REGABOVE(REGION,2),              ! Density (g/cc) of region above
assy
     c   REGABOVE(REGION,3)               ! Number of isotopes in region above
assy
         DO 660 ISOTOPE=1, REGABOVE(REGION,3)
            READ(15,650) ABOVEZAIDS(REGION,ISOTOPE), ! MCNP ZAID
     c      ABOVEWTS(REGION,ISOTOPE)
 650        FORMAT(T1,A9,1X,G15.13)
 660     CONTINUE
 670  CONTINUE
      ELSEIF (WESTINGHOUSE.EQ..TRUE.) THEN
*     If BPRAs are present, read in the descriptions of the axial zones at
and
*     above the upper end-fitting of the fuel assembly containing the BPRA.
      DO 674 BANK=1,NUMOFBANKS
         IF (BANKDES(BANK).EQ.'BPRA ') THEN
            READ(15,*) NUMREGABOVEBPRA
            DO 673 REGION=1,NUMREGABOVEBPRA ! Region 1 is the top-most axial
region
               READ(15,*) REGABOVEBPRA(REGION,1), ! Height (cm) of region
above BPRA
     c         REGABOVEBPRA(REGION,2),            ! Density (g/cc) of region
above BPRA
     c         REGABOVEBPRA(REGION,3)             ! Number of isotopes in
region above BPRA
               DO 672 ISOTOPE=1, REGABOVEBPRA(REGION,3)
                  READ(15,671) ABOVEBPRAZAIDS(REGION,ISOTOPE), ! MCNP ZAID
     c            ABOVEBPRAWTS(REGION,ISOTOPE)
 671              FORMAT(T1,A9,1X,G15.13)
 672           CONTINUE
 673        CONTINUE
            EXIT
         ENDIF
 674  CONTINUE
*     If CRAs are present, read in the descriptions of the axial zones at and
*     above the upper end-fitting of the fuel assembly containing the CRA.
      DO 678 BANK=1,NUMOFBANKS
         IF (BANKDES(BANK).EQ.'CRA  ') THEN
            READ(15,*) NUMREGABOVECRA
            DO 677 REGION=1,NUMREGABOVECRA
               READ(15,*) REGABOVECRA(REGION,1),  ! Height (cm) of region
above CRA
     c         REGABOVECRA(REGION,2),             ! Density (g/cc) of region
above CRA
     c         REGABOVECRA(REGION,3)              ! Number of isotopes in
region above CRA
               DO 676 ISOTOPE=1, REGABOVECRA(REGION,3)
                  READ(15,675) ABOVECRAZAIDS(REGION,ISOTOPE), ! MCNP ZAID
     c            ABOVECRAWTS(REGION,ISOTOPE)
 675              FORMAT(T1,A9,1X,G15.13)
```

```
  676          CONTINUE
  677        CONTINUE
             EXIT
           ENDIF
  678 CONTINUE
*     Read in the descriptions of the axial zones at and above the upper
*     end-fitting of the fuel assembly with no insertion rod assembly.
      READ(15,*) NUMREGABOVE
      DO 681 REGION=1,NUMREGABOVE
        READ(15,*) REGABOVE(REGION,1),   ! Height (cm) of region above assy
     c  REGABOVE(REGION,2),              ! Density (g/cc) of region above
assy
     c  REGABOVE(REGION,3)               ! Number of isotopes in region above
assy
        DO 680 ISOTOPE=1, REGABOVE(REGION,3)
          READ(15,679) ABOVEZAIDS(REGION,ISOTOPE), ! MCNP ZAID
     c    ABOVEWTS(REGION,ISOTOPE)
  679     FORMAT(T1,A9,1X,G15.13)
  680   CONTINUE
  681 CONTINUE
      ENDIF
*     Read in the specifications for each fuel assembly design.
*     Read in the dimension specifications for the fuel rods in the assembly
design.
      DO 810 DESIGN=1,NUMOFFADESIGNS
        READ(15,*) NUMOFMCNPFUELNODES(DESIGN)
        DO 682 NODE=1,NUMOFMCNPFUELNODES(DESIGN)
          READ(15,*) NODENUMBER, MCNPFUELHEIGHT(DESIGN,NODE)
  682   CONTINUE
        READ(15,*) RODNUM(DESIGN)
        READ(15,*) PINPITCH(DESIGN)
        READ(15,*) GRAMS(DESIGN)
        READ(15,*) FUELRADIUS(DESIGN), CLADRADIUS(DESIGN,1),
     c  CLADRADIUS(DESIGN,2)
        READ(15,*) ASSYPLENUM(DESIGN,1), ASSYPLENUM(DESIGN,2)
        READ(15,*) ENDCAPHEIGHT(DESIGN,1),
     c  ENDCAPHEIGHT(DESIGN,2)
        READ(15,*) ENDFITHEIGHT(DESIGN,1),
     c  ENDFITHEIGHT(DESIGN,2)
        READ(15,*) NUMOFSPACERS(DESIGN)
        DO 690 SPACER=1,NUMOFSPACERS(DESIGN)
          READ(15,*) SPACERNUMBER, SPACERHEIGHT(DESIGN,SPACER),
     c    SPACERDIST(DESIGN,SPACER), SPACERVOL(DESIGN,SPACER),
     c    SPACERMAT(DESIGN,SPACER)
          IF (SPACERMAT(DESIGN,SPACER).EQ.4) THEN
            BACKSPACE(15)
            READ(15,*) I, X, X, X, I, SPM4SS(DESIGN,SPACER),
     c      SPM4INC(DESIGN,SPACER)
          ELSEIF (SPACERMAT(DESIGN,SPACER).EQ.5) THEN
            BACKSPACE(15)
            READ(15,*) I, X, X, X, I, SPM4SS(DESIGN,SPACER),
     c      SPM4ZR(DESIGN,SPACER)
          ENDIF
  690   CONTINUE
```

```
*     Read in the material specifications for the fuel rods in the assembly
design.
            READ(15,*) CLADMATERIAL(DESIGN)
            READ(15,*) FRUPLENMAT(DESIGN,1), ! Fuel rod upper plenum density
     c      FRUPLENMAT(DESIGN,2)  ! Number of isotopes in fuel rod upper plenum
            DO 710 ISOTOPE=1,FRUPLENMAT(DESIGN,2)
               READ(15,700) FRUPLENZAIDS(DESIGN,ISOTOPE), ! MCNP ZAID
     c         FRUPLENWTS(DESIGN,ISOTOPE)
  700          FORMAT(T1,A9,1X,G15.13)
  710       CONTINUE
            READ(15,*) FRLPLENMAT(DESIGN,1), ! Fuel rod lower plenum density
     c      FRLPLENMAT(DESIGN,2)  ! Number of isotopes in fuel rod lower plenum
            DO 730 ISOTOPE=1,FRLPLENMAT(DESIGN,2)
               READ(15,720) FRLPLENZAIDS(DESIGN,ISOTOPE), ! MCNP ZAID
     c         FRLPLENWTS(DESIGN,ISOTOPE)
  720          FORMAT(T1,A9,1X,G15.13)
  730       CONTINUE
*     In Westinghouse designs the upper end-fitting material should correspond
*     only to the region below the top of either the fuel rod, BPR, or CR.
            READ(15,*) UEFMAT(DESIGN,1), ! Fuel rod assy upper end-fitting
density
     c      UEFMAT(DESIGN,2)  ! Number of isotopes in fuel rod upper end-fitting
            DO 750 ISOTOPE=1,UEFMAT(DESIGN,2)
               READ(15,740) UEFZAIDS(DESIGN,ISOTOPE), ! MCNP ZAID
     c         UEFWTS(DESIGN,ISOTOPE)
  740          FORMAT(T1,A9,1X,G15.13)
  750       CONTINUE
            READ(15,*) LEFMAT(DESIGN,1), ! Fuel rod assy lower end-fitting
density
     c      LEFMAT(DESIGN,2)  ! Number of isotopes in fuel rod lower end-fitting
            DO 770 ISOTOPE=1,LEFMAT(DESIGN,2)
               READ(15,760) LEFZAIDS(DESIGN,ISOTOPE), ! MCNP ZAID
     c         LEFWTS(DESIGN,ISOTOPE)
  760          FORMAT(T1,A9,1X,G15.13)
  770       CONTINUE
         IF (BANDW.EQ..TRUE.) THEN
*     Read in the dimension specifications for the guide tubes in the
assembly design.
            READ(15,*) GTDATA(DESIGN,1), ! Guide tube inner radius (cm)
     c      GTDATA(DESIGN,2), ! Guide tube outer radius (cm)
     c      GTDATA(DESIGN,3), ! Distance (cm) between guide tube upper end and
top of lower pad
     c      GTDATA(DESIGN,4) ! Distance (cm) between guide tube lower end and
top of lower pad
            READ(15,*) GTMAT(DESIGN)
*     Read in the dimension specifications for the instrument tubes in the
assembly design.
            READ(15,*) ITDATA(DESIGN,1), ! Instrument tube inner radius (cm)
     c      ITDATA(DESIGN,2), ! Instrument tube outer radius (cm)
     c      ITDATA(DESIGN,3), ! Distance (cm) between instrument tube upper end
and top of lower pad
     c      ITDATA(DESIGN,4) ! Distance (cm) between instrument tube lower end
and top of lower pad
            READ(15,*) ITMAT(DESIGN)
```

```
      ELSEIF (WESTINGHOUSE.EQ..TRUE.) THEN
*     Read in the dimension specifications for the guide tubes in the
assembly design.
         READ(15,*) GTSPLIT
         IF (GTSPLIT.EQ.1) THEN
         READ(15,*) NUMOFGTAXS(DESIGN)
         DO 772 GTAX=1,NUMOFGTAXS(DESIGN)
         READ(15,*) GTAXDATA(DESIGN,1,GTAX), ! Guide tube section inner
radius (cm)
     c   GTAXDATA(DESIGN,2,GTAX), ! Guide tube section outer radius (cm)
     c   GTAXDATA(DESIGN,3,GTAX), ! Distance (cm) between guide tube section
upper end and top of lower pad
     c   GTAXDATA(DESIGN,4,GTAX) ! Distance (cm) between guide tube lower end
and top of lower pad
         READ(15,*) GTAXMAT(DESIGN,GTAX)
  772    CONTINUE
         ELSEIF(GTSPLIT.NE.1) THEN
         READ(15,*) GTDATA(DESIGN,1), ! Guide tube inner radius (cm)
     c   GTDATA(DESIGN,2), ! Guide tube outer radius (cm)
     c   GTDATA(DESIGN,3), ! Distance (cm) between guide tube upper end and
top of lower pad
     c   GTDATA(DESIGN,4) ! Distance (cm) between guide tube lower end and
top of lower pad
         READ(15,*) GTMAT(DESIGN)
         ENDIF
*     Read in the dimension specifications for the instrument tubes in the
assembly design.
         READ(15,*) ITSPLIT
         IF (ITSPLIT.EQ.1) THEN
         READ(15,*) NUMOFITAXS(DESIGN)
         DO 800 ITAX=1,NUMOFITAXS(DESIGN)
         READ(15,*) ITAXDATA(DESIGN,1,ITAX), ! Instrument tube inner radius
(cm)
     c   ITAXDATA(DESIGN,2,ITAX), ! Instrument tube outer radius (cm)
     c   ITAXDATA(DESIGN,3,ITAX), ! Distance (cm) between instrument tube
upper end and top of lower pad
     c   ITAXDATA(DESIGN,4,ITAX) ! Distance (cm) between instrument tube
lower end and top of lower pad
         READ(15,*) ITAXMAT(DESIGN,ITAX)
  800    CONTINUE
         ELSEIF(ITSPLIT.NE.1) THEN
         READ(15,*) ITDATA(DESIGN,1), ! Instrument tube inner radius (cm)
     c   ITDATA(DESIGN,2), ! Instrument tube outer radius (cm)
     c   ITDATA(DESIGN,3), ! Distance (cm) between instrument tube upper end
and top of lower pad
     c   ITDATA(DESIGN,4) ! Distance (cm) between instrument tube lower end
and top of lower pad
         READ(15,*) ITMAT(DESIGN)
         ENDIF
      ENDIF
  810 CONTINUE
      IF (BANDW.EQ..TRUE.) THEN
*     Read in the specifications for each BPRA design.
      DO 846 BANKIDS=1,NUMOFBANKS
```

```
          IF (BANKDES(BANKIDS).EQ.'BPRA ') THEN
        READ(15,*) NUMDIFFBPRABANKS
        DO 844 BANK=1,NUMDIFFBPRABANKS
           READ(15,*) CURRENTBANK
           READ(15,*) NUMOFBPRANODES(CURRENTBANK)
           DO 820 NODE=1,NUMOFBPRANODES(CURRENTBANK)
              READ(15,*) NODENUMBER, MCNPBPRAHEIGHT(CURRENTBANK,NODE)
  820      CONTINUE
           READ(15,*) BOTBPNODEHEIGHT(CURRENTBANK) ! Distance between
*  the top of the lower pad and the bottom of the bottom BP node.
           READ(15,822) NONABSBP(CURRENTBANK)
  822      FORMAT(T1,A1)
           IF (NONABSBP(CURRENTBANK).EQ.'Y') THEN
              DO 826 NODE=1,NUMOFBPRANODES(CURRENTBANK)
                 READ(15,824) NODENUMBER, BPRABSNODE(CURRENTBANK,NODE)
  824            FORMAT(T1,I2,1X,A1)
  826         CONTINUE
           ENDIF
           READ(15,*) BPRDIM(CURRENTBANK,1), ! BP absorber radius (cm)
     c     BPRDIM(CURRENTBANK,2), ! BPR inner radius (cm)
     c     BPRDIM(CURRENTBANK,3) ! BPR outer radius (cm)
           READ(15,*) BPRPLEN(CURRENTBANK,1), ! BPR upper end-cap height (cm)
     c     BPRPLEN(CURRENTBANK,2) ! BPR lower end-cap height (cm)
           READ(15,*) BPABSMAT(CURRENTBANK) ! 1=AL2O3-B4C any other number is
not
           IF (BPABSMAT(CURRENTBANK).NE.1) THEN
              READ(15,*) BPMATDATA(CURRENTBANK,1), ! BP density (g/cc)
     c        BPMATDATA(CURRENTBANK,2) ! Number of isotopes in BP
              DO 830 ISOTOPE=1,BPMATDATA(CURRENTBANK,2)
                 READ(15,828) BPMATZAIDS(CURRENTBANK,ISOTOPE), ! MCNP ZAID
     c           BPMATWTS(CURRENTBANK,ISOTOPE)
  828            FORMAT(T1,A9,1X,G15.13)
  830         CONTINUE
           ELSEIF (BPABSMAT(CURRENTBANK).EQ.1) THEN
              READ(15,*) AL2O3B4CDENSITY(CURRENTBANK),
     c        B4CWTPCT(CURRENTBANK)
           ENDIF
           IF (NONABSBP(CURRENTBANK).EQ.'Y') THEN
              READ(15,*) BPNONABSMAT(CURRENTBANK) ! 1=AL2O3 any other number is
not
              IF (BPNONABSMAT(CURRENTBANK).NE.1) THEN
                 READ(15,*) NONBPMATDATA(CURRENTBANK,1), ! Non-absorbing BP
density (g/cc)
     c           NONBPMATDATA(CURRENTBANK,2) ! Number of isotopes in non-
absorbing BP
                 DO 834 ISOTOPE=1,NONBPMATDATA(CURRENTBANK,2)
                    READ(15,832) NONBPMATZAIDS(CURRENTBANK,ISOTOPE), ! MCNP
ZAID
     c              NONBPMATWTS(CURRENTBANK,ISOTOPE)
  832               FORMAT(T1,A9,1X,G15.13)
  834            CONTINUE
              ELSEIF (BPNONABSMAT(CURRENTBANK).EQ.1) THEN
                 READ(15,*) AL2O3DENSITY(CURRENTBANK)
              ENDIF
```

```
            ENDIF
            READ(15,*) BPRCLADMAT(CURRENTBANK) ! 1=ZIRC-4, 2=SS304, 3=INCONEL
            READ(15,*) BPRUPLENMAT(CURRENTBANK,1), ! BPR upper plenum density
     c      BPRUPLENMAT(CURRENTBANK,2)  ! Number of isotopes in BPR upper plenum
            DO 838 ISOTOPE=1,BPRUPLENMAT(CURRENTBANK,2)
               READ(15,836) BPRUPLENZAIDS(CURRENTBANK,ISOTOPE), ! MCNP ZAID
     c         BPRUPLENWTS(CURRENTBANK,ISOTOPE)
 836           FORMAT(T1,A9,1X,G15.13)
 838        CONTINUE
            READ(15,*) BPRLPLENMAT(CURRENTBANK,1), ! BPR lower plenum density
     c      BPRLPLENMAT(CURRENTBANK,2)  ! Number of isotopes in BPR lower plenum
            DO 842 ISOTOPE=1,BPRLPLENMAT(CURRENTBANK,2)
               READ(15,840) BPRLPLENZAIDS(CURRENTBANK,ISOTOPE), ! MCNP ZAID
     c         BPRLPLENWTS(CURRENTBANK,ISOTOPE)
 840           FORMAT(T1,A9,1X,G15.13)
 842        CONTINUE
 844     CONTINUE
            EXIT
            ENDIF
 846     CONTINUE
         ELSEIF(WESTINGHOUSE.EQ..TRUE.) THEN
*        Read in the specifications for each BPRA design.
         DO 918 BANKIDS=1,NUMOFBANKS
            IF (BANKDES(BANKIDS).EQ.'BPRA ') THEN
         READ(15,*) NUMDIFFBPRABANKS
         DO 916 BANK=1,NUMDIFFBPRABANKS
            READ(15,*) CURRENTBANK
            READ(15,*) NUMOFBPRANODES(CURRENTBANK)
            DO 848 NODE=1,NUMOFBPRANODES(CURRENTBANK)
               READ(15,*) NODENUMBER, MCNPBPRAHEIGHT(CURRENTBANK,NODE)
 848        CONTINUE
            READ(15,*) BOTBPNODEHEIGHT(CURRENTBANK) ! Distance between
*     the top of the lower pad and the bottom of the bottom BP node.
            READ(15,850) NONABSBP(CURRENTBANK)
 850        FORMAT(T1,A1)
            IF (NONABSBP(CURRENTBANK).EQ.'Y') THEN
               DO 854 NODE=1,NUMOFBPRANODES(CURRENTBANK)
                  READ(15,852) NODENUMBER, BPRABSNODE(CURRENTBANK,NODE)
 852              FORMAT(T1,I2,1X,A1)
 854           CONTINUE
            ENDIF
            READ(15,*) WBPRATYPE(CURRENTBANK)
*        WBPRATYPE=1: 4 BPR
*        WBPRATYPE=2: 8 BPR
*        WBPRATYPE=3: 9 BPR
*        WBPRATYPE=4: 10 BPR
*        WBPRATYPE=5: 12 BPR
*        WBPRATYPE=6: 16 BPR
*        WBPRATYPE=7: 20 BPR
            READ(15,*) WBPRA(CURRENTBANK)
*        WBPRA=1: solid, single absorber zone burnable poison rod
*        WBPRA=2: annular, single absorber zone burnable poison rod
*        WBPRA=3: solid, multiple absorber zone burnable poison rod
*        WBPRA=4: annular, multiple absorber zone burnable poison rod
```

```
      IF (WBPRA(CURRENTBANK).EQ.1) THEN
         READ(15,*) BPRDIM(CURRENTBANK,1), ! BP absorber radius (cm)
     c   BPRDIM(CURRENTBANK,2), ! BPR inner radius (cm)
     c   BPRDIM(CURRENTBANK,3) ! BPR outer radius (cm)
         READ(15,*) BPRPLEN(CURRENTBANK,1), ! BPR upper end-cap height (cm)
     c   BPRPLEN(CURRENTBANK,2) ! BPR lower end-cap height (cm)
         READ(15,*) BPABSMAT(CURRENTBANK) ! 1=AL203-B4C any other number is
not
         IF (BPABSMAT(CURRENTBANK).NE.1) THEN
            READ(15,*) BPMATDATA(CURRENTBANK,1), ! BP density (g/cc)
     c      BPMATDATA(CURRENTBANK,2) ! Number of isotopes in BP
            DO 858 ISOTOPE=1,BPMATDATA(CURRENTBANK,2)
               READ(15,856) BPMATZAIDS(CURRENTBANK,ISOTOPE), ! MCNP ZAID
     c         BPMATWTS(CURRENTBANK,ISOTOPE)
  856          FORMAT(T1,A9,1X,G15.13)
  858       CONTINUE
         ELSEIF (BPABSMAT(CURRENTBANK).EQ.1) THEN
            READ(15,*) AL203B4CDENSITY(CURRENTBANK),
     c      B4CWTPCT(CURRENTBANK)
         ENDIF
         IF (NONABSBP(CURRENTBANK).EQ.'Y') THEN
            READ(15,*) BPNONABSMAT(CURRENTBANK) ! 1=AL203 any other number is
not
            IF (BPNONABSMAT(CURRENTBANK).NE.1) THEN
               READ(15,*) NONBPMATDATA(CURRENTBANK,1), ! Non-absorbing BP
density (g/cc)
     c         NONBPMATDATA(CURRENTBANK,2) ! Number of isotopes in non-
absorbing BP
               DO 862 ISOTOPE=1,NONBPMATDATA(CURRENTBANK,2)
                  READ(15,860) NONBPMATZAIDS(CURRENTBANK,ISOTOPE), ! MCNP
ZAID
     c            NONBPMATWTS(CURRENTBANK,ISOTOPE)
  860             FORMAT(T1,A9,1X,G15.13)
  862          CONTINUE
            ELSEIF (BPNONABSMAT(CURRENTBANK).EQ.1) THEN
               READ(15,*) AL203DENSITY(CURRENTBANK)
            ENDIF
         ENDIF
         READ(15,*) BPRCLADMAT(CURRENTBANK) ! 1=ZIRC-4, 2=SS304, 3=INCONEL
         READ(15,*) BPRUPLENMAT(CURRENTBANK,1), ! BPR upper plenum density
     c   BPRUPLENMAT(CURRENTBANK,2)  ! Number of isotopes in BPR upper plenum
         DO 866 ISOTOPE=1,BPRUPLENMAT(CURRENTBANK,2)
            READ(15,864) BPRUPLENZAIDS(CURRENTBANK,ISOTOPE), ! MCNP ZAID
     c      BPRUPLENWTS(CURRENTBANK,ISOTOPE)
  864       FORMAT(T1,A9,1X,G15.13)
  866    CONTINUE
         READ(15,*) BPRLPLENMAT(CURRENTBANK,1), ! BPR lower plenum density
     c   BPRLPLENMAT(CURRENTBANK,2)  ! Number of isotopes in BPR lower plenum
         DO 870 ISOTOPE=1,BPRLPLENMAT(CURRENTBANK,2)
            READ(15,868) BPRLPLENZAIDS(CURRENTBANK,ISOTOPE), ! MCNP ZAID
     c      BPRLPLENWTS(CURRENTBANK,ISOTOPE)
  868       FORMAT(T1,A9,1X,G15.13)
  870    CONTINUE
      ELSEIF ((WBPRA(CURRENTBANK).EQ.2).OR.
```

```
     c    (WBPRA(CURRENTBANK).EQ.3)) THEN
          READ(15,*) BPRAXDIM(CURRENTBANK,1), ! BPR inner cladding inner
radius (cm)
     c    BPRAXDIM(CURRENTBANK,2), ! BPR inner cladding outer radius (cm)
     c    BPRAXDIM(CURRENTBANK,3), ! BP absorber inner radius (cm)
     c    BPRAXDIM(CURRENTBANK,4), ! BP absorber outer radius (cm)
     c    BPRAXDIM(CURRENTBANK,5), ! BPR outer cladding inner radius (cm)
     c    BPRAXDIM(CURRENTBANK,6) ! BPR outer cladding outer radius (cm)
          READ(15,*) BPRPLEN(CURRENTBANK,1), ! BPR upper end-cap height (cm)
     c    BPRPLEN(CURRENTBANK,2) ! BPR lower end-cap height (cm)
          READ(15,*) BPABSMAT(CURRENTBANK) ! 1=AL2O3-B4C any other number is
not
          IF (BPABSMAT(CURRENTBANK).NE.1) THEN
             READ(15,*) BPMATDATA(CURRENTBANK,1), ! BP density (g/cc)
     c       BPMATDATA(CURRENTBANK,2) ! Number of isotopes in BP
             DO 888 ISOTOPE=1,BPMATDATA(CURRENTBANK,2)
                READ(15,886) BPMATZAIDS(CURRENTBANK,ISOTOPE), ! MCNP ZAID
     c          BPMATWTS(CURRENTBANK,ISOTOPE)
  886           FORMAT(T1,A9,1X,G15.13)
  888        CONTINUE
          ELSEIF (BPABSMAT(CURRENTBANK).EQ.1) THEN
             READ(15,*) AL2O3B4CDENSITY(CURRENTBANK),
     c       B4CWTPCT(CURRENTBANK)
          ENDIF
          IF (NONABSBP(CURRENTBANK).EQ.'Y') THEN
             READ(15,*) BPNONABSMAT(CURRENTBANK) ! 1=AL2O3 any other number is
not
             IF (BPNONABSMAT(CURRENTBANK).NE.1) THEN
                READ(15,*) NONBPMATDATA(CURRENTBANK,1), ! Non-absorbing BP
density (g/cc)
     c          NONBPMATDATA(CURRENTBANK,2) ! Number of isotopes in non-
absorbing BP
                DO 892 ISOTOPE=1,NONBPMATDATA(CURRENTBANK,2)
                   READ(15,890) NONBPMATZAIDS(CURRENTBANK,ISOTOPE), ! MCNP
ZAID
     c             NONBPMATWTS(CURRENTBANK,ISOTOPE)
  890              FORMAT(T1,A9,1X,G15.13)
  892           CONTINUE
             ELSEIF (BPNONABSMAT(CURRENTBANK).EQ.1) THEN
                READ(15,*) AL2O3DENSITY(CURRENTBANK)
             ENDIF
          ENDIF
          READ(15,*) BPRCLADMAT(CURRENTBANK) ! 1=ZIRC-4, 2=SS304, 3=INCONEL
          READ(15,*) BPRUPLENMAT(CURRENTBANK,1), ! BPR upper plenum density
     c    BPRUPLENMAT(CURRENTBANK,2)  ! Number of isotopes in BPR upper plenum
          DO 896 ISOTOPE=1,BPRUPLENMAT(CURRENTBANK,2)
             READ(15,894) BPRUPLENZAIDS(CURRENTBANK,ISOTOPE), ! MCNP ZAID
     c       BPRUPLENWTS(CURRENTBANK,ISOTOPE)
  894        FORMAT(T1,A9,1X,G15.13)
  896     CONTINUE
          READ(15,*) BPRLPLENMAT(CURRENTBANK,1), ! BPR lower plenum density
     c    BPRLPLENMAT(CURRENTBANK,2)  ! Number of isotopes in BPR lower plenum
          DO 900 ISOTOPE=1,BPRLPLENMAT(CURRENTBANK,2)
             READ(15,898) BPRLPLENZAIDS(CURRENTBANK,ISOTOPE), ! MCNP ZAID
```

```
     c        BPRLPLENWTS(CURRENTBANK,ISOTOPE)
  898         FORMAT(T1,A9,1X,G15.13)
  900      CONTINUE
         ENDIF
  916 CONTINUE
           EXIT
      ENDIF
  918 CONTINUE
      ENDIF
      IF (BANDW.EQ..TRUE.) THEN
*     Read in the specifications for each CRA design.
      DO 934 BANKIDS=1,NUMOFBANKS
         IF (BANKDES(BANKIDS).EQ.'CRA  ') THEN
            READ(15,*) NUMDIFFCRABANKS
            DO 932 BANK=1,NUMDIFFCRABANKS
               READ(15,*) CURRENTBANK
               READ(15,*) CRADIM(CURRENTBANK,1), ! CR absorber material
radius (cm)
     c         CRADIM(CURRENTBANK,2), ! CR inner radius (cm)
     c         CRADIM(CURRENTBANK,3), ! CR outer radius (cm)
     c         CRADIM(CURRENTBANK,4), ! Distance (cm) between CR absorber
material
*                                     and the bottom of the active fuel.
     c         CRADIM(CURRENTBANK,5), ! CR end-cap height (cm)
     c         CRADIM(CURRENTBANK,6), ! CR lower plenum height (cm)
     c         CRADIM(CURRENTBANK,7)  ! CR upper plenum height (cm)
               READ(15,*) CRABSMAT(CURRENTBANK,1), ! Density (g/cc) of CR
absorber material
     c         CRABSMAT(CURRENTBANK,2) ! Number of isotopes in CR absorber
material
               DO 922 ISOTOPE=1,INT(CRABSMAT(CURRENTBANK,2))
                  READ(15,920) CRABSZAIDS(CURRENTBANK,ISOTOPE), ! MCNP ZAID
     c            CRABSWTS(CURRENTBANK,ISOTOPE) ! Isotope wt %
  920             FORMAT(T1,A9,1X,G15.13)
  922          CONTINUE
               READ(15,*) CRCLADMAT(CURRENTBANK) ! 1=ZIRC-4, 2=SS304,
3=INCONEL
               READ(15,*) CRUPLENMAT(CURRENTBANK,1), ! CR upper plenum
density
     c         CRUPLENMAT(CURRENTBANK,2)  ! Number of isotopes in CR upper
plenum
               DO 926 ISOTOPE=1,INT(CRUPLENMAT(CURRENTBANK,2))
                  READ(15,924) CRUPZS(CURRENTBANK,ISOTOPE), ! MCNP ZAID
     c            CRUPLENWTS(CURRENTBANK,ISOTOPE)
  924             FORMAT(T1,A9,1X,G15.13)
  926          CONTINUE
               READ(15,*) CRLPLENMAT(CURRENTBANK,1), ! CR lower plenum
density
     c         CRLPLENMAT(CURRENTBANK,2)  ! Number of isotopes in CR lower
plenum
               DO 930 ISOTOPE=1,INT(CRLPLENMAT(CURRENTBANK,2))
                  READ(15,928) CRLPLENZAIDS(CURRENTBANK,ISOTOPE), ! MCNP ZAID
     c            CRLPLENWTS(CURRENTBANK,ISOTOPE)
  928             FORMAT(T1,A9,1X,G15.13)
```

```
  930          CONTINUE
  932        CONTINUE
             EXIT
           ENDIF
  934 CONTINUE
        ELSEIF (WESTINGHOUSE.EQ..TRUE.) THEN
        READ(15,*) HYBRID
*      Hybrid=1: hybrid control rod used for some bank
*      Hybrid<>1: no hybrid control rod used for any bank
        IF (HYBRID.NE.1) THEN
        DO 950 BANKIDS=1,NUMOFBANKS
           IF (BANKDES(BANKIDS).EQ.'CRA   ') THEN
              READ(15,*) NUMDIFFCRABANKS
              DO 948 BANK=1,NUMDIFFCRABANKS
                 READ(15,*) CURRENTBANK
                 READ(15,*) CRADIM(CURRENTBANK,1), ! CR absorber material
radius (cm)
     c           CRADIM(CURRENTBANK,2), ! CR inner radius (cm)
     c           CRADIM(CURRENTBANK,3), ! CR outer radius (cm)
     c           CRADIM(CURRENTBANK,4), ! Distance (cm) between CR absorber
material
*                                         and the bottom of the active fuel.
     c           CRADIM(CURRENTBANK,5), ! CR end-cap height (cm)
     c           CRADIM(CURRENTBANK,6), ! CR lower plenum height (cm)
     c           CRADIM(CURRENTBANK,7)  ! CR upper plenum height (cm)
                 READ(15,*) CRABSMAT(CURRENTBANK,1), ! Density (g/cc) of CR
absorber material
     c           CRABSMAT(CURRENTBANK,2) ! Number of isotopes in CR absorber
material
                 DO 938 ISOTOPE=1,INT(CRABSMAT(CURRENTBANK,2))
                    READ(15,936) CRABSZAIDS(CURRENTBANK,ISOTOPE), ! MCNP ZAID
     c              CRABSWTS(CURRENTBANK,ISOTOPE) ! Isotope wt %
  936               FORMAT(T1,A9,1X,G15.13)
  938            CONTINUE
                 READ(15,*) CRCLADMAT(CURRENTBANK) ! 1=ZIRC-4, 2=SS304,
3=INCONEL
                 READ(15,*) CRUPLENMAT(CURRENTBANK,1), ! CR upper plenum
density
     c           CRUPLENMAT(CURRENTBANK,2)  ! Number of isotopes in CR upper
plenum
                 DO 942 ISOTOPE=1,INT(CRUPLENMAT(CURRENTBANK,2))
                    READ(15,940) CRUPZS(CURRENTBANK,ISOTOPE), ! MCNP ZAID
     c              CRUPLENWTS(CURRENTBANK,ISOTOPE)
  940               FORMAT(T1,A9,1X,G15.13)
  942            CONTINUE
                 READ(15,*) CRLPLENMAT(CURRENTBANK,1), ! CR lower plenum
density
     c           CRLPLENMAT(CURRENTBANK,2)  ! Number of isotopes in CR lower
plenum
                 DO 946 ISOTOPE=1,INT(CRLPLENMAT(CURRENTBANK,2))
                    READ(15,944) CRLPLENZAIDS(CURRENTBANK,ISOTOPE), ! MCNP ZAID
     c              CRLPLENWTS(CURRENTBANK,ISOTOPE)
  944               FORMAT(T1,A9,1X,G15.13)
  946            CONTINUE
```

```
  948         CONTINUE
              EXIT
           ENDIF
  950 CONTINUE
        ELSEIF (HYBRID.EQ.1) THEN
        DO 980 BANKIDS=1,NUMOFBANKS
           IF (BANKDES(BANKIDS).EQ.'CRA  ') THEN
              READ(15,*) NUMDIFFCRABANKS
              DO 978 BANK=1,NUMDIFFCRABANKS
                 READ(15,*) CURRENTBANK, HYBFLAG(CURRENTBANK)
*    HYBFLAG=1: This bank is a hybrid control rod
*    HYBFLAG<>1: This bank is not a hybrid control rod
              IF (HYBFLAG(CURRENTBANK).EQ.1) THEN
                 READ(15,*) NUMCRAXS(CURRENTBANK)
              DO 956 CRAX=1,NUMCRAXS(CURRENTBANK)
                 READ(15,*) CRAXDIM(CURRENTBANK,1,CRAX), ! CR absorber material
radius (cm)
     c           CRAXDIM(CURRENTBANK,2,CRAX), ! CR inner radius (cm)
     c           CRAXDIM(CURRENTBANK,3,CRAX), ! CR outer radius (cm)
     c           CRAXDIM(CURRENTBANK,4,CRAX), ! Distance (cm) between CR
absorber material
*                                               bottom and the bottom of the
active fuel.
     c           CRAXDIM(CURRENTBANK,5,CRAX)   ! Height of CR absorber section
                 READ(15,*) CRABSAXMAT(CURRENTBANK,1,CRAX), ! Density (g/cc) of
CR absorber material
     c           CRABSAXMAT(CURRENTBANK,2,CRAX) ! Number of isotopes in CR
absorber material
                 DO 954 ISOTOPE=1,INT(CRABSAXMAT(CURRENTBANK,2,CRAX))
                    READ(15,952) CRABSAXZAIDS(CURRENTBANK,ISOTOPE,CRAX), ! MCNP
ZAID
     c              CRABSAXWTS(CURRENTBANK,ISOTOPE,CRAX) ! Isotope wt %
  952               FORMAT(T1,A9,1X,G15.13)
  954            CONTINUE
                 READ(15,*) CRAXCLADMAT(CURRENTBANK,CRAX) ! 1=ZIRC-4, 2=SS304,
3=INCONEL
  956         CONTINUE
                 READ(15,*) CRADIM(CURRENTBANK,5), ! CR end-cap height (cm)
     c           CRADIM(CURRENTBANK,6), ! CR lower plenum height (cm)
     c           CRADIM(CURRENTBANK,7)  ! CR upper plenum height (cm)
                 READ(15,*) CRUPLENMAT(CURRENTBANK,1), ! CR upper plenum
density
     c           CRUPLENMAT(CURRENTBANK,2)   ! Number of isotopes in CR upper
plenum
                 DO 960 ISOTOPE=1,INT(CRUPLENMAT(CURRENTBANK,2))
                    READ(15,958) CRUPZS(CURRENTBANK,ISOTOPE), ! MCNP ZAID
     c              CRUPLENWTS(CURRENTBANK,ISOTOPE)
  958               FORMAT(T1,A9,1X,G15.13)
  960            CONTINUE
                 READ(15,*) CRLPLENMAT(CURRENTBANK,1), ! CR lower plenum
density
     c           CRLPLENMAT(CURRENTBANK,2)   ! Number of isotopes in CR lower
plenum
                 DO 964 ISOTOPE=1,INT(CRLPLENMAT(CURRENTBANK,2))
```

```
                READ(15,962) CRLPLENZAIDS(CURRENTBANK,ISOTOPE), ! MCNP ZAID
     c          CRLPLENWTS(CURRENTBANK,ISOTOPE)
  962           FORMAT(T1,A9,1X,G15.13)
  964         CONTINUE
           ELSEIF (HYBFLAG(CURRENTBANK).NE.1) THEN
              READ(15,*) CRADIM(CURRENTBANK,1), ! CR absorber material
radius (cm)
     c        CRADIM(CURRENTBANK,2), ! CR inner radius (cm)
     c        CRADIM(CURRENTBANK,3), ! CR outer radius (cm)
     c        CRADIM(CURRENTBANK,4), ! Distance (cm) between CR absorber
material
*                                   and the bottom of the active fuel.
     c        CRADIM(CURRENTBANK,5), ! CR end-cap height (cm)
     c        CRADIM(CURRENTBANK,6), ! CR lower plenum height (cm)
     c        CRADIM(CURRENTBANK,7)  ! CR upper plenum height (cm)
              READ(15,*) CRABSMAT(CURRENTBANK,1), ! Density (g/cc) of CR
absorber material
     c        CRABSMAT(CURRENTBANK,2) ! Number of isotopes in CR absorber
material
              DO 968 ISOTOPE=1,INT(CRABSMAT(CURRENTBANK,2))
                 READ(15,966) CRABSZAIDS(CURRENTBANK,ISOTOPE), ! MCNP ZAID
     c           CRABSWTS(CURRENTBANK,ISOTOPE) ! Isotope wt %
  966            FORMAT(T1,A9,1X,G15.13)
  968         CONTINUE
              READ(15,*) CRCLADMAT(CURRENTBANK) ! 1=ZIRC-4, 2=SS304,
3=INCONEL
              READ(15,*) CRUPLENMAT(CURRENTBANK,1), ! CR upper plenum
density
     c        CRUPLENMAT(CURRENTBANK,2)  ! Number of isotopes in CR upper
plenum
              DO 972 ISOTOPE=1,INT(CRUPLENMAT(CURRENTBANK,2))
                 READ(15,970) CRUPZS(CURRENTBANK,ISOTOPE), ! MCNP ZAID
     c           CRUPLENWTS(CURRENTBANK,ISOTOPE)
  970            FORMAT(T1,A9,1X,G15.13)
  972         CONTINUE
              READ(15,*) CRLPLENMAT(CURRENTBANK,1), ! CR lower plenum
density
     c        CRLPLENMAT(CURRENTBANK,2)  ! Number of isotopes in CR lower
plenum
              DO 976 ISOTOPE=1,INT(CRLPLENMAT(CURRENTBANK,2))
                 READ(15,974) CRLPLENZAIDS(CURRENTBANK,ISOTOPE), ! MCNP ZAID
     c           CRLPLENWTS(CURRENTBANK,ISOTOPE)
  974            FORMAT(T1,A9,1X,G15.13)
  976         CONTINUE
           ENDIF
  978      CONTINUE
           EXIT
        ENDIF
  980 CONTINUE
      ENDIF
      ENDIF
      IF (BANDW.EQ..TRUE.) THEN
*     Read in the specifications for each APSRA design.
      DO 1080 BANKIDS=1,NUMOFBANKS
```

```
         IF (BANKDES(BANKIDS).EQ.'APSRA') THEN
            READ(15,*) NUMDIFFAPSRABANKS
            DO 1070 BANK=1,NUMDIFFAPSRABANKS
               READ(15,*) CURRENTBANK
               READ(15,*) APSRADIM(CURRENTBANK,1), ! APSR absorber material
radius (cm)
     c         APSRADIM(CURRENTBANK,2), ! APSR clad inner radius (cm)
     c         APSRADIM(CURRENTBANK,3), ! APSR clad outer radius (cm)
     c         APSRADIM(CURRENTBANK,4), ! APSR follow rod inner radius (cm)
     c         APSRADIM(CURRENTBANK,5)  ! APSR follow rod outer radius (cm)
               READ(15,*) APSRIPLUGFRAC(CURRENTBANK) ! Volume fraction of
intermediate spacer in APSR
               READ(15,*) APSRADIM(CURRENTBANK,6), ! Distance (cm) between
APSR absorber material
*                                                     and the bottom of the
active fuel.
     c         APSRADIM(CURRENTBANK,7), ! APSR absorber region height (cm)
     c         APSRADIM(CURRENTBANK,8), ! APSR upper end-cap height (cm)
     c         APSRADIM(CURRENTBANK,9), ! APSR lower end-cap height (cm)
     c         APSRADIM(CURRENTBANK,10), ! APSR upper plenum height (cm)
     c         APSRADIM(CURRENTBANK,11)  ! APSR lower plenum height (cm)
               READ(15,*) APSRABSMAT(CURRENTBANK,1), ! Density (g/cc) of APSR
absorber material
     c         APSRABSMAT(CURRENTBANK,2) ! Number of isotopes in APSR
absorber material
               DO 1020 ISOTOPE=1,APSRABSMAT(CURRENTBANK,2)
                  READ(15,1010) APSRABSZAIDS(CURRENTBANK,ISOTOPE), ! MCNP
ZAID
     c            APSRABSWTS(CURRENTBANK,ISOTOPE) ! Isotope wt %
 1010             FORMAT(T1,A9,1X,G15.13)
 1020          CONTINUE
               READ(15,*) APSRCLADMAT(CURRENTBANK) ! 1=ZIRC-4, 2=SS304,
3=INCONEL
               READ(15,*) APSRFOLLOWMAT(CURRENTBANK) ! 1=ZIRC-4, 2=SS304,
3=INCONEL
               READ(15,*) APSRUPLENMAT(CURRENTBANK,1), ! APSR upper plenum
density
     c         APSRUPLENMAT(CURRENTBANK,2)  ! Number of isotopes in APSR
upper plenum
               DO 1040 ISOTOPE=1,APSRUPLENMAT(CURRENTBANK,2)
                  READ(15,1030) APSRUPLENZAIDS(CURRENTBANK,ISOTOPE), ! MCNP
ZAID
     c            APSRUPLENWTS(CURRENTBANK,ISOTOPE)
 1030             FORMAT(T1,A9,1X,G15.13)
 1040          CONTINUE
               READ(15,*) APSRLPLENMAT(CURRENTBANK,1), ! APSR lower plenum
density
     c         APSRLPLENMAT(CURRENTBANK,2)  ! Number of isotopes in APSR
lower plenum
               DO 1060 ISOTOPE=1,APSRLPLENMAT(CURRENTBANK,2)
                  READ(15,1050) APSRLPLENZAIDS(CURRENTBANK,ISOTOPE), ! MCNP
ZAID
     c            APSRLPLENWTS(CURRENTBANK,ISOTOPE)
 1050             FORMAT(T1,A9,1X,G15.13)
```

```
 1060          CONTINUE
 1070       CONTINUE
         EXIT
         ENDIF
 1080 CONTINUE
      ENDIF

      RETURN
      END

******************************************************************
*     SUBROUTINE MODDEN                                          *
*     This subroutine calculates the moderator density (g/cc)    *
*     from the moderator temperature (F) and the system          *
*     pressure (psi).                                            *
******************************************************************
      SUBROUTINE MODDEN (MODTEMP, PRESSURE, MODDENSITY)
*
      INTEGER CT3, COL1, COL2, ROW1, ROW2
*
      REAL MODTEMP, PRESSURE, MODDENSITY, DENDAT(29,10),
     c P1, P2, DENCOL(29)
*
*    Data input for table of subcooled water density (g/cc) at
*    various temperatures (F) and pressures (psia).
*    (REFERENCE: Radiation Shielding Information Center Number
*    CCC-545, "SCALE 4.2, Modular Code System for Performing
*    Standardized Computer Analyses for Licensing Evaluation,
*    Volume 1, Page S2.5.14, Table S2.5.2.)
*
      DATA ((DENDAT(E,Q),Q=1,10),E=1,29) /0.0,3000.0,2500.0,
     c 2000.0,1500.0,1000.0,
     c 800.0,600.0,400.0,200.0,50.0,1.0084,1.0069,1.0055,1.0040,
     c 1.0025,1.0019,
     c 1.0013,1.0007,1.000,100,1.0018,1.0004,0.9989,0.9975,0.9960,
     c 0.9954,0.9948,0.9942,0.9936,150.0,0.9893,0.9878,0.9864,0.9849,
     c 0.9834,0.9828,0.9822,0.9815,0.9809,200,0.9725,0.9709,0.9694,
     c 0.9679,0.9663,0.9656,0.9650,0.9644,0.9637,250.0,0.9522,0.9505,
     c 0.9489,0.9472,0.9455,0.9449,0.9442,0.9435,0.9428,300,0.9289,
     c 0.9271,0.9252,0.9234,0.9215,0.9208,0.9200,0.9192,0.9185,350.0,
     c 0.9026,0.9006,0.8985,0.8964,0.8943,0.8934,0.8925,0.8916,0,
     c 400.0,0.8733,0.8709,0.8685,0.8660,0.8634,0.8624,0.8613,0.8603,0,
     c 450.0,0.8405,0.8375,0.8345,0.8314,0.8281,0.8268,0.8255,0,0,
     c 500.0,0.8029,0.7992,0.7952,0.7911,0.7869,0.7851,0,0,0,
     c 510.0,0.7947,0.7907,0.7866,0.7822,0.7776,0,0,0,0,
     c 520.0,0.7862,0.7820,0.7776,0.7729,0.7680,0,0,0,0,
     c 530.0,0.7775,0.7729,0.7682,0.7632,0.7579,0,0,0,0,
     c 540.0,0.7683,0.7635,0.7584,0.7530,0.7472,0,0,0,0,
     c 550.0,0.7589,0.7537,0.7482,0.7423,0,0,0,0,0,
     c 560.0,0.7490,0.7434,0.7374,0.7310,0,0,0,0,0,
     c 570.0,0.7386,0.7326,0.7261,0.7190,0,0,0,0,0,
     c 580.0,0.7278,0.7212,0.7141,0.7062,0,0,0,0,0,
     c 590.0,0.7164,0.7092,0.7012,0.6923,0,0,0,0,0,
     c 600.0,0.7043,0.6963,0.6874,0,0,0,0,0,0,
```

```
     c 610.0,0.6915,0.6825,0.6724,0,0,0,0,0,0,
     c 620.0,0.6777,0.6676,0.6558,0,0,0,0,0,0,
     c 630.0,0.6629,0.6512,0.6370,0,0,0,0,0,0,
     c 640.0,0.6467,0.6329,0,0,0,0,0,0,0,
     c 650.0,0.6288,0.6119,0,0,0,0,0,0,0,
     c 660.0,0.6086,0.5866,0,0,0,0,0,0,0,
     c 670.0,0.5850,0,0,0,0,0,0,0,0,
     c 680.0,0.5559,0,0,0,0,0,0,0,0/
*
      DO 10 CT3=2,10
         IF ((PRESSURE.LT.DENDAT(1,CT3)).AND.
     c   (PRESSURE.GT.DENDAT(1,(CT3+1)))) THEN
            P1=DENDAT(1,CT3)
            P2=DENDAT(1,(CT3+1))
            COL1=CT3
            COL2=(CT3+1)
         ELSEIF (PRESSURE.EQ.DENDAT(1,CT3)) THEN
            P1=PRESSURE
            P2=DENDAT(1,(CT3+1))
            COL1=CT3
            COL2=(CT3+1)
         ENDIF
   10 CONTINUE
      DO 20 CT3=2,29
         DENCOL(CT3)=((PRESSURE-P2)*((DENDAT(CT3,COL1)
     c   -DENDAT(CT3,COL2))/(P1-P2)))+DENDAT(CT3,COL2)
   20 CONTINUE
      DO 30 CT3=2,29
         IF ((MODTEMP.GT.DENDAT(CT3,1)).AND.
     c   (MODTEMP.LT.DENDAT(CT3+1,1))) THEN
            ROW1=CT3
            ROW2=CT3+1
            MODDENSITY=(((MODTEMP-DENDAT(CT3,1))*
     c      (DENCOL(ROW2)-DENCOL(ROW1))))/(DENDAT(CT3+1,1)
     c      -DENDAT(CT3,1)))+DENCOL(ROW1)
         ELSEIF (MODTEMP.EQ.DENDAT(CT3,1)) THEN
            MODDENSITY=DENCOL(CT3)
         ENDIF
   30 CONTINUE

      RETURN
      END

*********************************************************************
*     SUBROUTINE INTROSECTION                                       *
*     This subroutine writes the introduction section of the MCNP   *
*     input deck.                                                   *
*********************************************************************
      SUBROUTINE INTROSECTION (BANDW, WESTINGHOUSE, CE, EIGHTH,
     c QUARTER, FULL, PREFIX, REACTOR, CYCLE, EFPD, NUMOFNODES,
     c NUMOFMCNPFUELNODES, INTROFILE)
*
      INTEGER NUMOFNODES, NUMOFMCNPFUELNODES(20), NUMSTPT1,
     c NUMSTPT2, NUMSTPT3
```

```
*
      REAL EFPD
*
      CHARACTER PREFIX*3, REACTOR*21, CYCLE*2, INTROFILE*16,
     c CHSTPT1*1, CHSTPT2*1, CHSTPT3*1
*
      LOGICAL BANDW, WESTINGHOUSE, CE, EIGHTH, QUARTER, FULL
*
*     Open file to contain the introduction section of the MCNP input deck.
      INTROFILE(1:3)=PREFIX
      INTROFILE(4:4)='C'
      INTROFILE(5:6)=CYCLE
      INTROFILE(7:7)='T'
      NUMSTPT1=INT(EFPD/100.0)
      CHSTPT1=CHAR(NUMSTPT1+48)
      NUMSTPT2=INT((EFPD-(NUMSTPT1*100))/10.0)
      CHSTPT2=CHAR(NUMSTPT2+48)
      NUMSTPT3=INT(EFPD-(NUMSTPT1*100)-
     c (NUMSTPT2*10))
      CHSTPT3=CHAR(NUMSTPT3+48)
      INTROFILE(8:8)=CHSTPT1
      INTROFILE(9:9)=CHSTPT2
      INTROFILE(10:10)=CHSTPT3
      INTROFILE(11:16)='.intro'
      OPEN(UNIT=20, FILE=INTROFILE, STATUS='UNKNOWN')
      REWIND(UNIT=20)
*     Write the introduction section of the MCNP input deck.
      WRITE(20,10) REACTOR, CYCLE, EFPD
   10 FORMAT(T1,A21,', CYCLE ',A2,', ',F5.1,' EFPD STATEPOINT')
      WRITE(20,12)
   12 FORMAT(T1,'C')
      WRITE(20,13)
   13 FORMAT(T1,'C   This MCNP input deck was generated',
     c ' by MACE Version 3.')
      WRITE(20,12)
      WRITE(20,14)
   14 FORMAT(T1,'C   PROBLEM DESCRIPTION')
      WRITE(20,16)
   16 FORMAT(T1,'C')
      IF (BANDW.EQ..TRUE.) THEN
         IF (EIGHTH.EQ..TRUE.) THEN
            WRITE(20,20)
   20       FORMAT(T1,'C   This is a B&W reactor design modeled',
     c      ' in eighth-core symmetry.')
         ELSEIF (QUARTER.EQ..TRUE.) THEN
            WRITE(20,30)
   30       FORMAT(T1,'C   This is a B&W reactor design modeled',
     c      ' in quarter-core symmetry.')
         ELSEIF (FULL.EQ..TRUE.) THEN
            WRITE(20,40)
   40       FORMAT(T1,'C   This is a B&W reactor design modeled',
     c      ' as a full core.')
         ENDIF
      ELSEIF (WESTINGHOUSE.EQ..TRUE.) THEN
```

```
        IF (EIGHTH.EQ..TRUE.) THEN
           WRITE(20,50)
  50       FORMAT(T1,'C   This is a Westinghouse reactor design',
     c     ' modeled in eighth-core symmetry.')
        ELSEIF (QUARTER.EQ..TRUE.) THEN
           WRITE(20,60)
  60       FORMAT(T1,'C   This is a Westinghouse reactor design',
     c     ' modeled in quarter-core symmetry.')
        ELSEIF (FULL.EQ..TRUE.) THEN
           WRITE(20,70)
  70       FORMAT(T1,'C   This is a Westinghouse reactor design',
     c     ' modeled as a full core.')
        ENDIF
      ELSEIF (CE.EQ..TRUE.) THEN
        IF (EIGHTH.EQ..TRUE.) THEN
           WRITE(20,80)
  80       FORMAT(T1,'C   This is a CE reactor design modeled',
     c     ' in eighth-core symmetry.')
        ELSEIF (QUARTER.EQ..TRUE.) THEN
           WRITE(20,90)
  90       FORMAT(T1,'C   This is a CE reactor design modeled',
     c     ' in quarter-core symmetry.')
        ELSEIF (FULL.EQ..TRUE.) THEN
           WRITE(20,100)
 100       FORMAT(T1,'C   This is a CE reactor design modeled',
     c     ' as a full core.')
        ENDIF
      ENDIF
      IF ((CYCLE.NE.'01').OR.(CYCLE.NE.'1A')) THEN
        WRITE(20,110)
 110    FORMAT(T1,'C   The depleted fuel and burnable poison')
        WRITE(20,120)
 120    FORMAT(T1,'C   isotopics were calculated with SAS2H.')
        WRITE(20,130)
 130    FORMAT(T1,'C   The SAS2H depletion calculations were')
        WRITE(20,140)
 140    FORMAT(T1,'C   performed on assemblies modeled as having')
        WRITE(20,150) NUMOFNODES
 150    FORMAT(T1,'C   ',I2,' unique axial fuel nodes.')
      ELSEIF (EFPD.NE.(0.0)) THEN
        WRITE(20,160)
 160    FORMAT(T1,'C   The depleted fuel and burnable poison')
        WRITE(20,170)
 170    FORMAT(T1,'C   isotopics were calculated with SAS2H.')
        WRITE(20,180)

 180    FORMAT(T1,'C   The SAS2H depletion calculations were')
        WRITE(20,190)
 190    FORMAT(T1,'C   performed on assemblies modeled as having')
        WRITE(20,200) NUMOFNODES
 200    FORMAT(T1,'C   ',I2,' unique axial fuel nodes.')
      ENDIF
      WRITE(20,210)
 210  FORMAT(T1,'C   This MCNP calculation models the fuel')
```

```
      WRITE(20,220) NUMOFMCNPFUELNODES(1)
  220 FORMAT(T1,'C   as having ',I2,' unique axial fuel node(s).')
      WRITE(20,230)
  230 FORMAT(T1,'C')
      CLOSE(UNIT=20)

      RETURN
      END

**********************************************************
*     This program retrieves the fuel and burnable       *
*     poison composition information from the depletion *
*     and decay calculation for the assembly.           *
**********************************************************
      SUBROUTINE FUEL (ASSYID, ISOREQUEST, CYCLE, EFPD,
     c NUMOFNODES, NODEHEIGHT, FUELRADIUS, DESNUM, GRAMS,
     c ENRICHMENT, RODNUM, STAT, BANKDES, BANKNUM,
     c BPRABSNODE, B4CWTPCT, AL2O3B4CDENSITY,
     c BPRDIM, FUELNODEDEN, BPDENTOGO, FDENPREF,
     c BPABSMAT, BPMATDATA, BPMATZAIDS, BPMATWTS,
     c BPNONABSMAT, NONBPMATDATA, NONBPMATZAIDS,
     c NONBPMATWTS, NONABSBP, BANDW, WESTINGHOUSE,
     c WBPRA, WBPRATYPE, BPRAXDIM, NUMOFBPRANODES)
*
      INTEGER BPML, NUMOFASSYS, ISOREQUEST, NUMOFNODES,
     c NODE, RODS(50,50), BPROD,
     c FUELNODEISONUM(50,50,50), ISO, RES, N, C, N1,
     c N2, NUMSTPT1, NUMSTPT2, NUMSTPT3, CT1, COLUMNSTART,
     c COLUMNEND, ISONUMBER, CARRYCOUNTER, CT2,
     c DESNUM(50,50), RODNUM(20), FUELNODEML(50,50,50),
     c BANKNUM(50,50), COLUMN, ROW, CO, RO,
     c BPABSMAT(20), BPNONABSMAT(20), WBPRA(20),
     c WBPRATYPE(20), NUMOFBPRANODES(20)
*
      REAL PI, EFPD, NODEHEIGHT(50), RAD(50,50), MASS(50,50),
     c RICH(50,50), BPRICH(50,50), BPDEN(50,50), BPRAD(50,50),
     c OXYGMS, WT234, WT235, WT236, WT238, WTOXY,
     c FUELNODEDEN(50,50,50), FUELNODECOMP(50,50,50,100),
     c ALWT, OWT, CWT, B10WT,
     c B11WT, ISOVALUE(1000), FUELISOVALUE(1000),
     c BPRAISOVALUE(2), BPNODECOMP(50,50,50,2), BPVOL,
     c BPMASSTOTAL, MASSTOTAL, OXYWT,
     c FUELRADIUS(20), BPDENTOGO(50,50,50),
     c GRAMS(20), ENRICHMENT(50,50), B4CWTPCT(20),
     c AL2O3B4CDENSITY(20), BPRDIM(20,3), TOTHEIGHT,
     c WT234INU, WT235INU, WT236INU, WT238INU,
     c UMASSPERMOL, OMASSPERMOL, CMASSPERMOLB4C,
     c BMASSPERMOLB4C, ALMASSPERMOLAL2O3, OMASSPERMOLAL2O3,
     c B10WTINB4C, B11WTINB4C, CWTINB4C, ALWTINAL2O3,
     c OWTINAL2O3, DENFRAC, TMASS, UWT, BPMATDATA(20,2),
     c BPMATWTS(20,35), NONBPMATDATA(20,2),
     c NONBPMATWTS(20,35), BPAREA, BPRAXDIM(20,6)
*
      CHARACTER ASSYID(50,50)*5, CYCLE*2, STAT(50,50)*1,
```

```
     c BP(50,50)*1, FUELNODEZAIDS(50,50,50,100)*9,
     c COMMAND1*11, COMMAND2*54, CHN1*1, CHN2*1,
     c CHSTPT1*1, CHSTPT2*1, CHSTPT3*1, FILENAME*8,
     c ISONAME(1000)*6, ROWFLAG*7, COL1*8, COL2*8,
     c COL3*8, COL4*8, COL5*8, COL6*8, COL7*8, COL8*8,
     c BPRAISONAME(2)*6, BPLABEL*14, FORMATLABEL*29,
     c BPRAZAID(50,50,50,2)*9, ACTINIDELABEL*9,
     c ISOLABEL*6, FISSPRODLABEL*29, FSORIG(84)*6,
     c FSZAID(84)*9, PIORIG(29)*6, PIZAID(29)*9,
     c PAORIG(14)*6, PAZAID(14)*9, AOORIG(10)*6,
     c AOZAID(10)*9, COMMAND3*11,
     c BPRABSNODE(20,50)*1, BANKDES(20)*5, FDENPREF*1,
     c BPMATZAIDS(20,35)*9, NONBPMATZAIDS(20,35)*9,
     c NONABSBP(20)*1
*
      LOGICAL ROWFLAGLOG, BPFIND, ACTINIDEFIND, FISSPRODFIND,
     c ASSYUNIQUE(50,50), BPGO(50,50,50), LEAVE, BPRUNIQUE(50,50),
     c BANDW, WESTINGHOUSE
*
      DATA (FSORIG(X),X=1,84)   /' h  3 ','he  4 ','li  6 ','li  7 ',
     c'be  9 ','as 75 ','kr 80 ','kr 82 ','kr 83 ','kr 84 ','kr 86 ',
     c' y 89 ','zr 93 ','nb 93 ','mo 95 ','tc 99 ','ru101 ','ru103 ',
     c'rh103 ','rh105 ','pd105 ','pd108 ','ag107 ','ag109 ','xe131 ',
     c'xe134 ','xe135 ','cs133 ','cs135 ','ba138 ','pr141 ','nd143 ',
     c'nd145 ','nd147 ','nd148 ','pm147 ','pm148 ','pm149 ','sm147 ',
     c'sm149 ','sm150 ','sm151 ','sm152 ','eu151 ','eu152 ','eu153 ',
     c'eu154 ','eu155 ','gd152 ','gd154 ','gd155 ','gd156 ','gd157 ',
     c'gd158 ','gd160 ','ho165 ','th232 ','pa233 ',' u233 ',' u234 ',
     c' u235 ',' u236 ',' u237 ',' u238 ','np235 ','np236 ','np237 ',
     c'np238 ','pu237 ','pu238 ','pu239 ','pu240 ','pu241 ','pu242 ',
     c'am241 ','am242m','am243 ','cm242 ','cm243 ','cm244 ','cm245 ',
     c'cm246 ','cm247 ','cm248 '/
      DATA (FSZAID(X),X=1,84) /' 1003.50c',' 2004.50c',' 3006.50c',
     c ' 3007.55c',' 4009.50c','33075.35c','36080.50c','36082.50c',
     c '36083.50c','36084.50c','36086.50c',
     c '39089.50c','40093.50c','41093.50c','42095.50c','43099.50c',
     c '44101.50c','44103.50c','45103.50c','45105.50c','46105.50c',
     c '46108.50c','47107.60c','47109.60c','54131.50c',
     c '54134.35c','54135.53c','55133.50c','55135.50c','56138.50c',
     c '59141.50c','60143.50c','60145.50c','60147.50c','60148.50c',
     c '61147.50c','61148.50c','61149.50c','62147.50c','62149.50c',
     c '62150.50c','62151.50c','62152.50c','63151.55c','63152.50c',
     c '63153.55c','63154.50c','63155.50c','64152.50c','64154.50c',
     c '64155.50c','64156.50c','64157.50c','64158.50c','64160.50c',
     c '67165.55c','90232.50c','91233.50c','92233.50c',
     c '92234.50c','92235.53c','92236.50c','92237.50c','92238.53c',
     c '93235.35c','93236.35c','93237.50c','93238.35c','94237.35c',
     c '94238.50c','94239.55c','94240.50c','94241.50c','94242.50c',
     c '95241.50c','95242.50c','95243.50c','96242.50c','96243.35c',
     c '96244.50c','96245.35c','96246.35c','96247.35c','96248.35c'/
      DATA (PIORIG(X),X=1,29)   /'mo 95 ','tc 99 ','ru101 ','rh103 ',
     c'ag109 ','nd143 ','nd145 ','sm147 ','sm149 ',
     c'sm150 ','sm151 ','sm152 ','eu151 ','eu153 ','gd155 ',
     c' u233 ',' u234 ',' u235 ',
```

```
     c' u236 ',' u238 ','np237 ','pu238 ','pu239 ','pu240 ','pu241 ',
     c'pu242 ','am241 ',
     c'am242m','am243 '/
      DATA (PIZAID(X),X=1,29) /'42095.50c','43099.50c',
     c '44101.50c','44103.50c','47109.60c',
     c '60143.50c','60145.50c','62147.50c','62149.50c','62150.50c',
     c '62151.50c','62152.50c','63151.55c','63153.55c','64155.50c',
     c '92233.50c','92234.50c','92235.53c','92236.50c','92238.53c',
     c '93237.50c','94238.50c','94239.55c','94240.50c','94241.50c',
     c '94242.50c','95241.50c','95242.50c','95243.50c'/
      DATA (PAORIG(X),X=1,14) /' u233 ',' u234 ',' u235 ',
     c' u236 ',' u238 ','np237 ','pu238 ','pu239 ','pu240 ','pu241 ',
     c'pu242 ','am241 ','am242m','am243 '/
      DATA (PAZAID(X),X=1,14) /'92233.50c','92234.50c','92235.53c',
     c '92236.50c','92238.53c','93237.50c','94238.50c','94239.55c',
     c '94240.50c','94241.50c','94242.50c','95241.50c','95242.50c',
     c '95243.50c'/
      DATA (AOORIG(X),X=1,10) /' u234 ',' u235 ',
     c' u236 ',' u238 ','pu238 ','pu239 ','pu240 ','pu241 ',
     c'pu242 ','am241 '/
      DATA (AOZAID(X),X=1,10) /'92234.50c','92235.53c',
     c '92236.50c','92238.53c','94238.50c','94239.55c',
     c '94240.50c','94241.50c','94242.50c','95241.50c'/
*
      PI=3.141592653
      FMN=6000
      BPROD=16
      BPML=3000
      NUMOFASSYS=0
      TOTHEIGHT=0.0
      DO 2 NODE=1,NUMOFNODES
         TOTHEIGHT=TOTHEIGHT+NODEHEIGHT(NODE)
    2 CONTINUE
      DO 55 ROW=1,50
         DO 50 COLUMN=1,50
            IF (ASSYID(COLUMN,ROW).NE.'      ') THEN
               ASSYUNIQUE(COLUMN,ROW)=.TRUE.
               IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                  LEAVE=.FALSE.
                  DO 10 RO=1,(ROW-1)
                     DO 5 CO=1,50
                        IF ((ASSYID(CO,RO).NE.'      ').AND.
     c                  (ASSYID(CO,RO).EQ.ASSYID(COLUMN,ROW))) THEN
                           ASSYUNIQUE(COLUMN,ROW)=.FALSE.
                           LEAVE=.TRUE.
                           EXIT
                        ENDIF
    5                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
   10             CONTINUE
                  IF (LEAVE.EQ..FALSE.) THEN
                     DO 20 RO=ROW,ROW
```

```
                  DO 15 CO=1,(COLUMN-1)
                     IF ((ASSYID(CO,RO).NE.'      ').AND.
     c               (ASSYID(CO,RO).EQ.ASSYID(COLUMN,ROW))) THEN
                        ASSYUNIQUE(COLUMN,ROW)=.FALSE.
                        LEAVE=.TRUE.
                        EXIT
                     ENDIF
   15             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
   20          CONTINUE
            ENDIF
         ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
            LEAVE=.FALSE.
            DO 30 RO=1,(ROW-1)
               DO 25 CO=1,50
                  IF ((ASSYID(CO,RO).NE.'      ').AND.
     c            (ASSYID(CO,RO).EQ.ASSYID(COLUMN,ROW))) THEN
                     ASSYUNIQUE(COLUMN,ROW)=.FALSE.
                     LEAVE=.TRUE.
                     EXIT
                  ENDIF
   25          CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
   30       CONTINUE
         ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
            LEAVE=.FALSE.
            DO 40 RO=1,1
               DO 35 CO=1,(COLUMN-1)
                  IF ((ASSYID(CO,RO).NE.'      ').AND.
     c            (ASSYID(CO,RO).EQ.ASSYID(COLUMN,ROW))) THEN
                     ASSYUNIQUE(COLUMN,ROW)=.FALSE.
                     LEAVE=.TRUE.
                     EXIT
                  ENDIF
   35          CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
   40       CONTINUE
         ENDIF
      ELSEIF (ASSYID(COLUMN,ROW).EQ.'      ') THEN
         ASSYUNIQUE(COLUMN,ROW)=.FALSE.
      ENDIF
   50   CONTINUE
   55 CONTINUE
      DO 65 ROW=1,50
         DO 60 COLUMN=1,50
            IF (ASSYUNIQUE(COLUMN,ROW).EQ..TRUE.) THEN
               NUMOFASSYS=NUMOFASSYS+1
               RAD(COLUMN,ROW)=FUELRADIUS(DESNUM(COLUMN,ROW))
```

```
                MASS(COLUMN,ROW)=GRAMS(DESNUM(COLUMN,ROW))
                RICH(COLUMN,ROW)=ENRICHMENT(COLUMN,ROW)
                RODS(COLUMN,ROW)=RODNUM(DESNUM(COLUMN,ROW))
              ENDIF
   60      CONTINUE
   65 CONTINUE
*  Determine if the assembly has a unique BPRA inserted.
      DO 1270 ROW=1,50
         DO 1260 COLUMN=1,50
            BPRUNIQUE(COLUMN,ROW)=.FALSE.
            IF (BANKNUM(COLUMN,ROW).NE.0) THEN
            IF (ASSYUNIQUE(COLUMN,ROW).EQ..TRUE.) THEN
               IF (BANKDES(BANKNUM(COLUMN,ROW)).EQ.'BPRA ') THEN
                  BPRUNIQUE(COLUMN,ROW)=.TRUE.
               ENDIF
            ELSEIF ((ASSYUNIQUE(COLUMN,ROW).EQ..FALSE.).AND.
     c      (BANKDES(BANKNUM(COLUMN,ROW)).EQ.'BPRA ')) THEN
               BPRUNIQUE(COLUMN,ROW)=.TRUE.
               LEAVE=.FALSE.
               IF (COLUMN.NE.1) THEN
                  DO 1245 RO=1,(ROW-1)
                     DO 1244 CO=1,50
                        IF ((ASSYID(COLUMN,ROW).NE.'      ').AND.
     c                     (ASSYID(COLUMN,ROW).EQ.ASSYID(CO,RO)).AND.
     c                     (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO))) THEN
                           BPRUNIQUE(COLUMN,ROW)=.FALSE.
                           LEAVE=.TRUE.
                           EXIT
                        ENDIF
 1244                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 1245             CONTINUE
                  IF (LEAVE.EQ..FALSE.) THEN
                     DO 1247 RO=ROW,ROW
                        DO 1246 CO=1,(COLUMN-1)
                           IF ((ASSYID(COLUMN,ROW).NE.'      ').AND.
     c                     (ASSYID(COLUMN,ROW).EQ.ASSYID(CO,RO)).AND.
     c                     (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO))) THEN
                              BPRUNIQUE(COLUMN,ROW)=.FALSE.
                              LEAVE=.TRUE.
                              EXIT
                           ENDIF
 1246                   CONTINUE
                        IF (LEAVE.EQ..TRUE.) THEN
                           EXIT
                        ENDIF
 1247                CONTINUE
                  ENDIF
               ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                  DO 1249 RO=1,(ROW-1)
                     DO 1248 CO=1,50
                        IF ((ASSYID(COLUMN,ROW).NE.'      ').AND.
```

```
     c                      (ASSYID(COLUMN,ROW).EQ.ASSYID(CO,RO)).AND.
     c                      (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO))) THEN
                            BPRUNIQUE(COLUMN,ROW)=.FALSE.
                            LEAVE=.TRUE.
                            EXIT
                         ENDIF
 1248                 CONTINUE
                      IF (LEAVE.EQ..TRUE.) THEN
                         EXIT
                      ENDIF
 1249              CONTINUE
                ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                   DO 1251 RO=1,1
                      DO 1250 CO=1,(COLUMN-1)
                         IF ((ASSYID(COLUMN,ROW).NE.'      ').AND.
     c                      (ASSYID(COLUMN,ROW).EQ.ASSYID(CO,RO)).AND.
     c                      (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO))) THEN
                            BPRUNIQUE(COLUMN,ROW)=.FALSE.
                            LEAVE=.TRUE.
                            EXIT
                         ENDIF
 1250                 CONTINUE
                      IF (LEAVE.EQ..TRUE.) THEN
                         EXIT
                      ENDIF
 1251              CONTINUE
                ENDIF
             ENDIF
             ENDIF
 1260     CONTINUE
 1270  CONTINUE
       DO 80 ROW=1,50
          DO 75 COLUMN=1,50
             IF (BANKNUM(COLUMN,ROW).NE.0) THEN
                IF (BANKDES(BANKNUM(COLUMN,ROW)).EQ.'BPRA ') THEN
                   BP(COLUMN,ROW)='B'
                   BPRICH(COLUMN,ROW)=B4CWTPCT(BANKNUM(COLUMN,ROW))
                   BPDEN(COLUMN,ROW)=
     c             AL2O3B4CDENSITY(BANKNUM(COLUMN,ROW))
                   BPRAD(COLUMN,ROW)=BPRDIM(BANKNUM(COLUMN,ROW),1)
                   DO 70 N=1,NUMOFNODES
                IF ((BPRABSNODE(BANKNUM(COLUMN,ROW),N).EQ.'N').AND.
     c          (BPRUNIQUE(COLUMN,ROW).EQ..TRUE.)) THEN
                   BPGO(COLUMN,ROW,N)=.TRUE.
                ELSEIF (BPRABSNODE(BANKNUM(COLUMN,ROW),N).EQ.'Y') THEN
                   BPGO(COLUMN,ROW,N)=.FALSE.
                ELSEIF ((NONABSBP(BANKNUM(COLUMN,ROW)).EQ.'N').AND.
     c          (BPRUNIQUE(COLUMN,ROW).EQ..TRUE.)) THEN
                   BPGO(COLUMN,ROW,N)=.TRUE.
                ENDIF
   70              CONTINUE
                ENDIF
             ENDIF
   75     CONTINUE
```

```
   80 CONTINUE
*
      OPEN(UNIT=60, FILE='fuel.out', STATUS='UNKNOWN')
      REWIND(UNIT=60)
      OPEN(UNIT=70, FILE='bp.out', STATUS='UNKNOWN')
      REWIND(UNIT=70)
      OPEN(UNIT=80, FILE='fuelden.out', STATUS='UNKNOWN')
      REWIND(UNIT=80)
      OPEN(UNIT=90, FILE='fuelch.out', STATUS='UNKNOWN')
      REWIND(UNIT=90)
      DO 700 ROW=1,50
         DO 690 COLUMN=1,50
            IF (ASSYUNIQUE(COLUMN,ROW).EQ..TRUE.) THEN
               WRITE(*,*) 'Now retrieving isotopics from ',
     c         ASSYID(COLUMN,ROW)
* Calculate the oxygen mass in the fuel
      WT234INU=.007731*(RICH(COLUMN,ROW)**1.0837)
      WT235INU=RICH(COLUMN,ROW)
      WT236INU=.0046*RICH(COLUMN,ROW)
      WT238INU=100.0-WT234INU-WT235INU-WT236INU
      UMASSPERMOL=((1.008664904*232.030*WT234INU)+
     c (1.008664904*233.025*WT235INU)+
     c (1.008664904*234.018*WT236INU)+
     c (1.008664904*236.006*WT238INU))/100.0
      OMASSPERMOL=(2*1.008664904*15.858)
      OXYGMS=(OMASSPERMOL/UMASSPERMOL)*
     c MASS(COLUMN,ROW)
      TMASS=OXYGMS+MASS(COLUMN,ROW)
      IF (STAT(COLUMN,ROW).EQ.'F') THEN
         DO 150 NODE=1,NUMOFNODES
            WTOXY=(OXYGMS/TMASS)*100.0
            UWT=100.0-WTOXY
            WT234=(WT234INU/100.0)*UWT
            WT235=(WT235INU/100.0)*UWT
            WT236=(WT236INU/100.0)*UWT
            WT238=(WT238INU/100.0)*UWT
            FUELNODEISONUM(COLUMN,ROW,NODE)=5
            FUELNODECOMP(COLUMN,ROW,NODE,1)=WT234
            FUELNODECOMP(COLUMN,ROW,NODE,2)=WT235
            FUELNODECOMP(COLUMN,ROW,NODE,3)=WT236
            FUELNODECOMP(COLUMN,ROW,NODE,4)=WT238
            FUELNODECOMP(COLUMN,ROW,NODE,5)=WTOXY
            FUELNODEZAIDS(COLUMN,ROW,NODE,1)='92234.50c'
            FUELNODEZAIDS(COLUMN,ROW,NODE,2)='92235.53c'
            FUELNODEZAIDS(COLUMN,ROW,NODE,3)='92236.50c'
            FUELNODEZAIDS(COLUMN,ROW,NODE,4)='92238.53c'
            FUELNODEZAIDS(COLUMN,ROW,NODE,5)=' 8016.50c'
            IF ((FDENPREF.EQ.'T').OR.(FDENPREF.EQ.'C')) THEN
               FUELNODEDEN(COLUMN,ROW,NODE)=10.41
            ELSE
               FUELNODEDEN(COLUMN,ROW,NODE)=
     c         ((MASS(COLUMN,ROW)+OXYGMS)*
     c         (NODEHEIGHT(NODE)/TOTHEIGHT))/
     c         (NODEHEIGHT(NODE)*(PI)*
```

```
     c          (RAD(COLUMN,ROW)*RAD(COLUMN,ROW))*RODS(COLUMN,ROW))
              ENDIF
              FUELNODEML(COLUMN,ROW,NODE)=FMN
              FMN=FMN+1
              DO 95 ISO=1,FUELNODEISONUM(COLUMN,ROW,NODE)
                 IF (ISO.EQ.1) THEN
                    WRITE(60,85) FUELNODEML(COLUMN,ROW,NODE),
     c              FUELNODEZAIDS(COLUMN,ROW,NODE,ISO),
     c              (-1*FUELNODECOMP(COLUMN,ROW,NODE,ISO)),
     c              NODE, ASSYID(COLUMN,ROW), RICH(COLUMN,ROW)
 85                 FORMAT(T1,'M',I4,T8,A9,T20,G14.8,3X,
     c              '$ Fuel composition for node ',I2,
     c              ' of assembly ',A5,', ',F7.3,' U-235 WT% IN U')
                 ELSE
                    WRITE(60,90) FUELNODEZAIDS(COLUMN,ROW,NODE,ISO),
     c              (-1*FUELNODECOMP(COLUMN,ROW,NODE,ISO))
 90                 FORMAT(T8,A9,T20,G14.8)
                 ENDIF
 95           CONTINUE
              WRITE(80,100) ASSYID(COLUMN,ROW), NODE,
     c        (-1*FUELNODEDEN(COLUMN,ROW,NODE))
100           FORMAT(T1,'Assembly ',A5,' node ',I2,
     c        ' density ',G14.8,' g/cc')
150        CONTINUE
        ELSE
           COMMAND1(1:11)='rm ./*.temp'
           RES=SYSTEM(COMMAND1)
           N=0
           DO 160 C=1,5
              IF (ASSYID(COLUMN,ROW)(C:C).NE.' ') THEN
                 N=N+1
              ENDIF
160        CONTINUE
           DO 670 NODE=1,NUMOFNODES
              COMMAND2(1:6)='cp  ./'
              DO 170 C=1,N
                 COMMAND2((C+6):(C+6))=ASSYID(COLUMN,ROW)(C:C)
170           CONTINUE
              N1=INT(NODE/10.0)
              CHN1=CHAR(N1+48)
              N2=INT(NODE-(N1*10))
              CHN2=CHAR(N2+48)
              COMMAND2((N+7):(N+14))='/??????N'
              COMMAND2((N+15):(N+15))=CHN1
              COMMAND2((N+16):(N+16))=CHN2
              COMMAND2((N+17):(N+26))='????????AC'
              COMMAND2((N+27):(N+28))=CYCLE
              COMMAND2((N+29):(N+29))='T'
              NUMSTPT1=INT(EFPD/100.0)
              CHSTPT1=CHAR(NUMSTPT1+48)
              NUMSTPT2=INT((EFPD-(NUMSTPT1*100))/10.0)
              CHSTPT2=CHAR(NUMSTPT2+48)
              NUMSTPT3=INT((EFPD-(NUMSTPT1*100)-
     c        (NUMSTPT2*10)))
```

```
            CHSTPT3=CHAR(NUMSTPT3+48)
            COMMAND2((N+30):(N+30))=CHSTPT1
            COMMAND2((N+31):(N+31))=CHSTPT2
            COMMAND2((N+32):(N+32))=CHSTPT3
            COMMAND2((N+33):(N+40))='.cut ./N'
            COMMAND2((N+41):(N+41))=CHN1
            COMMAND2((N+42):(N+42))=CHN2
            COMMAND2((N+43):(N+47))='.temp'
            DO 180 C=(N+48),54
               COMMAND2(C:C)=' '
 180        CONTINUE
            RES=SYSTEM(COMMAND2)
            FILENAME(1:8)=COMMAND2((N+40):(N+47))
            OPEN (UNIT=300, FILE=FILENAME, STATUS='OLD')
            REWIND(UNIT=300)
*
*
            DO 190 CT1=1,1000
               ISOVALUE(CT1)=0.0
               FUELISOVALUE(CT1)=0.0
               ISONAME(CT1)='        '
 190        CONTINUE
            ROWFLAGLOG=.FALSE.
            DO 210 WHILE (ROWFLAGLOG.EQ..FALSE.)
               READ (300,200) ROWFLAG, COL1, COL2, COL3,
     c         COL4, COL5, COL6, COL7, COL8
 200           FORMAT (T15,A7,T24,A8,T34,A8,T44,A8,T54,A8,
     c         T64,A8,T74,A8,T84,A8,T94,A8)
               IF (ROWFLAG.EQ.'initial') THEN
                  ROWFLAGLOG=.TRUE.
               ENDIF
 210        CONTINUE
            IF (COL1.NE.'          ') THEN
               COLUMNSTART=23
               COLUMNEND=32
            ENDIF
            IF (COL2.NE.'          ') THEN
               COLUMNSTART=33
               COLUMNEND=42
            ENDIF
            IF (COL3.NE.'          ') THEN
               COLUMNSTART=43
               COLUMNEND=52
            ENDIF
            IF (COL4.NE.'          ') THEN
               COLUMNSTART=53
               COLUMNEND=62
            ENDIF
            IF (COL5.NE.'          ') THEN
               COLUMNSTART=63
               COLUMNEND=72
            ENDIF
            IF (COL6.NE.'          ') THEN
               COLUMNSTART=73
```

```
            COLUMNEND=82
         ENDIF
         IF (COL7.NE.'          ') THEN
            COLUMNSTART=83
            COLUMNEND=92
         ENDIF
         IF (COL8.NE.'          ') THEN
            COLUMNSTART=93
            COLUMNEND=102
         ENDIF
* Get fuel composition data
         REWIND(300)
         ACTINIDEFIND=.FALSE.
         DO 370 WHILE (ACTINIDEFIND.EQ..FALSE.)
            READ(300,350) ACTINIDELABEL
 350        FORMAT (T103,A9)
            IF (ACTINIDELABEL.EQ.'actinides') THEN
               READ(300,*)
               READ(300,360) FORMATLABEL
 360           FORMAT (T46,A29)
               IF (FORMATLABEL.EQ.
     c         'nuclide concentrations, grams') THEN
                  ACTINIDEFIND=.TRUE.
               ENDIF
            ENDIF
 370     CONTINUE
         READ(300,*)
         READ(300,*)
         ISOLABEL='      '
         ISONUMBER=0
         DO 390 WHILE (ISOLABEL.NE.'tal   ')
            ISONUMBER=ISONUMBER+1
            READ(300,380) ISONAME(ISONUMBER),
     c      ISOVALUE(ISONUMBER)
 380        FORMAT(T6,A6,T<COLUMNSTART>,G10.2)
            ISOLABEL=ISONAME(ISONUMBER)
            IF (ISOLABEL.EQ.'tal   ') THEN
               ISONAME(ISONUMBER)='      '
               ISOVALUE(ISONUMBER)=0
            ENDIF
 390     CONTINUE
         ISONUMBER=ISONUMBER-1
         REWIND(300)
         FISSPRODFIND=.FALSE.
         DO 420 WHILE (FISSPRODFIND.EQ..FALSE.)
            READ(300,400) FISSPRODLABEL
 400        FORMAT (T96,A16)
            IF (FISSPRODLABEL.EQ.'fission products') THEN
               READ(300,*)
               READ(300,410) FORMATLABEL
 410           FORMAT (T46,A29)
               IF (FORMATLABEL.EQ.
     c         'nuclide concentrations, grams') THEN
                  FISSPRODFIND=.TRUE.
```

```
                 ENDIF
              ENDIF
 420       CONTINUE
           READ(300,*)
           READ(300,*)
           ISOLABEL='      '
           DO 440 WHILE (ISOLABEL.NE.'tal   ')
              ISONUMBER=ISONUMBER+1
              READ(300,430) ISONAME(ISONUMBER),
     c        ISOVALUE(ISONUMBER)
 430          FORMAT(T6,A6,T<COLUMNSTART>,G10.2)
              ISOLABEL=ISONAME(ISONUMBER)
              IF (ISOLABEL.EQ.'      ') THEN
                 ISONUMBER=ISONUMBER-1
                 READ(300,*)
                 READ(300,*)
                 READ(300,*)
                 READ(300,*)
                 READ(300,*)
              ENDIF
              IF (ISOLABEL.EQ.'tal   ') THEN
                 ISONAME(ISONUMBER)='      '
                 ISOVALUE(ISONUMBER)=0
              ENDIF
 440       CONTINUE
           ISONUMBER=ISONUMBER-1
           CARRYCOUNTER=0
           MASSTOTAL=OXYGMS*(NODEHEIGHT(NODE)/TOTHEIGHT)
           IF (ISOREQUEST.EQ.1) THEN
              DO 460 CT1=1,ISONUMBER
                 DO 450 CT2=1,84
                    IF (ISONAME(CT1).EQ.FSORIG(CT2)) THEN
                       CARRYCOUNTER=CARRYCOUNTER+1
                       FUELNODEZAIDS(COLUMN,ROW,NODE,
     c                 CARRYCOUNTER)=FSZAID(CT2)
                       FUELISOVALUE(CARRYCOUNTER)=
     c                 ISOVALUE(CT1)
                    ENDIF
 450             CONTINUE
 460          CONTINUE
              DO 470 CT1=1,CARRYCOUNTER
                 MASSTOTAL=MASSTOTAL+FUELISOVALUE(CT1)
 470          CONTINUE
              DO 480 CT1=1,CARRYCOUNTER
                 FUELNODECOMP(COLUMN,ROW,NODE,CT1)=
     c           (FUELISOVALUE(CT1)/MASSTOTAL)*100.0
 480          CONTINUE
              FUELNODEISONUM(COLUMN,ROW,NODE)=CARRYCOUNTER
              IF (FDENPREF.EQ.'T') THEN
                 DENFRAC=10.41/(((MASS(COLUMN,ROW)+OXYGMS)*
     c           (NODEHEIGHT(NODE)/TOTHEIGHT))/
     c           (NODEHEIGHT(NODE)*(PI)*
     c           (RAD(COLUMN,ROW)*RAD(COLUMN,ROW))*
     c           RODS(COLUMN,ROW)))
```

```
                  FUELNODEDEN(COLUMN,ROW,NODE)=(MASSTOTAL/
     c            (NODEHEIGHT(NODE)*(PI)*
     c            (RAD(COLUMN,ROW)*RAD(COLUMN,ROW))*
     c            RODS(COLUMN,ROW)))*DENFRAC
               ELSE
                  FUELNODEDEN(COLUMN,ROW,NODE)=(MASSTOTAL)/
     c            (NODEHEIGHT(NODE)*(PI)*
     c            (RAD(COLUMN,ROW)*RAD(COLUMN,ROW))*
     c            RODS(COLUMN,ROW))
               ENDIF
               OXYWT=OXYGMS*(NODEHEIGHT(NODE)/TOTHEIGHT)
     c         *100.0/MASSTOTAL
            ELSEIF (ISOREQUEST.EQ.2) THEN
               DO 500 CT1=1,ISONUMBER
                  DO 490 CT2=1,29
                     IF (ISONAME(CT1).EQ.PIORIG(CT2)) THEN
                        CARRYCOUNTER=CARRYCOUNTER+1
                        FUELNODEZAIDS(COLUMN,ROW,NODE,
     c                  CARRYCOUNTER)=PIZAID(CT2)
                        FUELISOVALUE(CARRYCOUNTER)=ISOVALUE(CT1)
                     ENDIF
 490              CONTINUE
 500           CONTINUE
               DO 510 CT1=1,CARRYCOUNTER
                  MASSTOTAL=MASSTOTAL+FUELISOVALUE(CT1)
 510           CONTINUE
               DO 520 CT1=1,CARRYCOUNTER
                  FUELNODECOMP(COLUMN,ROW,NODE,CT1)=
     c            (FUELISOVALUE(CT1)/MASSTOTAL)*100.0
 520           CONTINUE
               FUELNODEISONUM(COLUMN,ROW,NODE)=CARRYCOUNTER
               IF (FDENPREF.EQ.'T') THEN
                  DENFRAC=10.41/(((MASS(COLUMN,ROW)+OXYGMS)*
     c            (NODEHEIGHT(NODE)/TOTHEIGHT))/
     c            (NODEHEIGHT(NODE)*(PI)*
     c            (RAD(COLUMN,ROW)*RAD(COLUMN,ROW))*
     c            RODS(COLUMN,ROW)))
                  FUELNODEDEN(COLUMN,ROW,NODE)=(MASSTOTAL/
     c            (NODEHEIGHT(NODE)*(PI)*
     c            (RAD(COLUMN,ROW)*RAD(COLUMN,ROW))*
     c            RODS(COLUMN,ROW)))*DENFRAC
               ELSE
                  FUELNODEDEN(COLUMN,ROW,NODE)=(MASSTOTAL)/
     c            (NODEHEIGHT(NODE)*(PI)*
     c            (RAD(COLUMN,ROW)*RAD(COLUMN,ROW))*
     c            RODS(COLUMN,ROW))
               ENDIF
               OXYWT=OXYGMS*(NODEHEIGHT(NODE)/TOTHEIGHT)
     c         *100.0/MASSTOTAL
            ELSEIF (ISOREQUEST.EQ.3) THEN
               DO 540 CT1=1,ISONUMBER
                  DO 530 CT2=1,14
                     IF (ISONAME(CT1).EQ.PAORIG(CT2)) THEN
                        CARRYCOUNTER=CARRYCOUNTER+1
```

```
                          FUELNODEZAIDS(COLUMN,ROW,NODE,
     c                    CARRYCOUNTER)=PAZAID(CT2)
                          FUELISOVALUE(CARRYCOUNTER)=ISOVALUE(CT1)
                       ENDIF
 530                CONTINUE
 540             CONTINUE
                 DO 550 CT1=1,CARRYCOUNTER
                    MASSTOTAL=MASSTOTAL+FUELISOVALUE(CT1)
 550             CONTINUE
                 DO 560 CT1=1,CARRYCOUNTER
                    FUELNODECOMP(COLUMN,ROW,NODE,CT1)=
     c              (FUELISOVALUE(CT1)/MASSTOTAL)*100.0
 560             CONTINUE
                 FUELNODEISONUM(COLUMN,ROW,NODE)=CARRYCOUNTER
                 IF (FDENPREF.EQ.'T') THEN
                    DENFRAC=10.41/(((MASS(COLUMN,ROW)+OXYGMS)*
     c              (NODEHEIGHT(NODE)/TOTHEIGHT))/
     c              (NODEHEIGHT(NODE)*(PI)*
     c              (RAD(COLUMN,ROW)*RAD(COLUMN,ROW))*
     c              RODS(COLUMN,ROW)))
                    FUELNODEDEN(COLUMN,ROW,NODE)=(MASSTOTAL/
     c              (NODEHEIGHT(NODE)*(PI)*
     c              (RAD(COLUMN,ROW)*RAD(COLUMN,ROW))*
     c              RODS(COLUMN,ROW)))*DENFRAC
                 ELSE
                    FUELNODEDEN(COLUMN,ROW,NODE)=(MASSTOTAL)/
     c              (NODEHEIGHT(NODE)*(PI)*
     c              (RAD(COLUMN,ROW)*RAD(COLUMN,ROW))*
     c              RODS(COLUMN,ROW))
                 ENDIF
                 OXYWT=OXYGMS*(NODEHEIGHT(NODE)/TOTHEIGHT)
     c           *100.0/MASSTOTAL
              ELSEIF (ISOREQUEST.EQ.4) THEN
                 DO 580 CT1=1,ISONUMBER
                    DO 570 CT2=1,10
                       IF (ISONAME(CT1).EQ.AOORIG(CT2)) THEN
                          CARRYCOUNTER=CARRYCOUNTER+1
                          FUELNODEZAIDS(COLUMN,ROW,NODE,
     c                    CARRYCOUNTER)=AOZAID(CT2)
                          FUELISOVALUE(CARRYCOUNTER)=ISOVALUE(CT1)
                       ENDIF
 570                CONTINUE
 580             CONTINUE
                 DO 590 CT1=1,CARRYCOUNTER
                    MASSTOTAL=MASSTOTAL+FUELISOVALUE(CT1)
 590             CONTINUE
                 DO 600 CT1=1,CARRYCOUNTER
                    FUELNODECOMP(COLUMN,ROW,NODE,CT1)=
     c              (FUELISOVALUE(CT1)/MASSTOTAL)*100.0
 600             CONTINUE
                 FUELNODEISONUM(COLUMN,ROW,NODE)=CARRYCOUNTER
                 IF (FDENPREF.EQ.'T') THEN
                    DENFRAC=10.41/(((MASS(COLUMN,ROW)+OXYGMS)*
     c              (NODEHEIGHT(NODE)/TOTHEIGHT))/
```

```
     c              (NODEHEIGHT(NODE)*(PI)*
     c              (RAD(COLUMN,ROW)*RAD(COLUMN,ROW))*
     c              RODS(COLUMN,ROW)))
                    FUELNODEDEN(COLUMN,ROW,NODE)=(MASSTOTAL/
     c              (NODEHEIGHT(NODE)*(PI)*
     c              (RAD(COLUMN,ROW)*RAD(COLUMN,ROW))*
     c              RODS(COLUMN,ROW)))*DENFRAC
                 ELSE
                    FUELNODEDEN(COLUMN,ROW,NODE)=(MASSTOTAL)/
     c              (NODEHEIGHT(NODE)*(PI)*
     c              (RAD(COLUMN,ROW)*RAD(COLUMN,ROW))*
     c              RODS(COLUMN,ROW))
                 ENDIF
                 OXYWT=OXYGMS*(NODEHEIGHT(NODE)/TOTHEIGHT)
     c           *100.0/MASSTOTAL
              ENDIF
              FUELNODEML(COLUMN,ROW,NODE)=FMN
              FMN=FMN+1
              DO 650 ISO=1,FUELNODEISONUM(COLUMN,ROW,NODE)
                 IF (ISO.EQ.1) THEN
                    WRITE(60,610) FUELNODEML(COLUMN,ROW,NODE),
     c              (-1*OXYWT), NODE, ASSYID(COLUMN,ROW),
     c              RICH(COLUMN,ROW)
 610                FORMAT(T1,'M',I4,T8,' 8016.50c',T20,G14.8,3X,
     c              '$ Fuel isotopic composition for node ',I2,
     c              ' of assembly ',A5,', ',F7.3,' WT% U-235 IN U')
                    WRITE(60,620)
 620                FORMAT(T8,$)
                    WRITE(60,625) FUELNODEZAIDS(COLUMN,ROW,NODE,ISO),
     c              (-1*FUELNODECOMP(COLUMN,ROW,NODE,ISO))
 625                FORMAT(A9,3X,G14.6,3X,$)
                 ELSE
                    WRITE(60,630) FUELNODEZAIDS(COLUMN,ROW,NODE,ISO),
     c              (-1*FUELNODECOMP(COLUMN,ROW,NODE,ISO))
 630                FORMAT(A9,3X,G14.6,3X,$)
                 ENDIF
                 IF ((MOD(ISO,2).EQ.0).AND.
     c           (ISO.NE.FUELNODEISONUM(COLUMN,ROW,NODE))) THEN
                    WRITE(60,*)
                    WRITE(60,640)
 640                FORMAT(T8,$)
                 ENDIF
                 IF (ISO.EQ.FUELNODEISONUM(COLUMN,ROW,NODE)) THEN
                    WRITE(60,*)
                 ENDIF
                 IF (ISO.EQ.1) THEN
                    WRITE(90,*) 'NODE ',NODE,' ASSEMBLY ',
     c              ASSYID(COLUMN,ROW)
                 ENDIF
                 WRITE(90,645) FUELNODEZAIDS(COLUMN,ROW,NODE,ISO),
     c           FUELISOVALUE(ISO)
 645             FORMAT(T1,A9,3X,G14.8)
 650          CONTINUE
              WRITE(80,660) ASSYID(COLUMN,ROW), NODE,
```

```
     c       (-1*FUELNODEDEN(COLUMN,ROW,NODE))
 660         FORMAT(T1,'Assembly ',A5,' node ',I2,
     c       ' density ',G14.8,' g/cc')
             CLOSE(UNIT=300)
 670      CONTINUE
        ENDIF
          ENDIF
 690      CONTINUE
 700    CONTINUE
*  Retrieve BP isotopic compositions.
        DO 950 ROW=1,50
          DO 940 COLUMN=1,50
           IF (BANKNUM(COLUMN,ROW).NE.0) THEN
             DO 930 NODE=1,NUMOFBPRANODES(BANKNUM(COLUMN,ROW))
             IF ((BPGO(COLUMN,ROW,NODE).EQ..TRUE.).AND.
     c       (STAT(COLUMN,ROW).EQ.'F')) THEN
             IF ((BPABSMAT(BANKNUM(COLUMN,ROW))).EQ.1) THEN
                CMASSPERMOLB4C=1.008664904*11.8969
                BMASSPERMOLB4C=4*((1.008664904*9.9269*0.194)+
     c          (1.008664904*10.9147*0.806))
                ALMASSPERMOLAL2O3=2*1.008664904*26.75
                OMASSPERMOLAL2O3=3*1.008664904*15.858
                B10WTINB4C=((1.008664904*9.9269*0.194/100.0)/
     c          (1.008664904*((9.9269*0.194/100.0)+
     c          (10.9147*0.806/100.0))))*BMASSPERMOLB4C*100.0/
     c          (BMASSPERMOLB4C+CMASSPERMOLB4C)
                B11WTINB4C=((1.008664904*10.9147*0.806/100.0)/
     c          (1.008664904*((9.9269*0.194/100.0)+
     c          (10.9147*0.806/100.0))))*BMASSPERMOLB4C*100.0/
     c          (BMASSPERMOLB4C+CMASSPERMOLB4C)
                CWTINB4C=(CMASSPERMOLB4C/
     c          (BMASSPERMOLB4C+CMASSPERMOLB4C))*100.0
                ALWTINAL2O3=(ALMASSPERMOLAL2O3/
     c          (ALMASSPERMOLAL2O3+OMASSPERMOLAL2O3))*100.0
                OWTINAL2O3=(OMASSPERMOLAL2O3/
     c          (ALMASSPERMOLAL2O3+OMASSPERMOLAL2O3))*100.0
                B10WT=BPRICH(COLUMN,ROW)*B10WTINB4C/100.0
                B11WT=BPRICH(COLUMN,ROW)*B11WTINB4C/100.0
                CWT=BPRICH(COLUMN,ROW)*CWTINB4C/100.0
                ALWT=ALWTINAL2O3*(100.0-BPRICH(COLUMN,ROW))/100.0
                OWT=OWTINAL2O3*(100.0-BPRICH(COLUMN,ROW))/100.0
                DO 706 ISO=1,2
                   IF (ISO.EQ.1) THEN
                      WRITE(70,701) BPML, (-1*ALWT), NODE,
     c                ASSYID(COLUMN,ROW), BPRICH(COLUMN,ROW)
 701                  FORMAT(T1,'M',I4,T8,'13027.50c',T20,G14.8,3X,
     c                '$ Burnable poison composition for node ',
     c                I2,' of assembly ',A5,', ',F7.3,
     c                ' WT% B4C IN ORIGINAL B4C-AL2O3')
                   ELSE
                      WRITE(70,702) (-1*OWT)
 702                  FORMAT(T8,' 8016.50c',T20,G14.8)
                      WRITE(70,703) (-1*CWT)
 703                  FORMAT(T8,' 6000.50c',T20,G14.8)
```

```
                  WRITE(70,704) (-1*B10WT)
 704              FORMAT(T8,' 5010.50c',T20,G14.8)
                  WRITE(70,705) (-1*B11WT)
 705              FORMAT(T8,' 5011.56c',T20,G14.8)
               ENDIF
 706        CONTINUE
         ELSEIF ((BPABSMAT(BANKNUM(COLUMN,ROW))).NE.1) THEN
            DO 709 ISO=1,BPMATDATA(BANKNUM(COLUMN,ROW),2)
               IF (ISO.EQ.1) THEN
                  WRITE(70,707) BPML,
     c            BPMATZAIDS(BANKNUM(COLUMN,ROW),ISO),
     c            (-1*BPMATWTS(BANKNUM(COLUMN,ROW),ISO)), NODE,
     c            ASSYID(COLUMN,ROW)
 707              FORMAT(T1,'M',I4,T8,A9,T20,G14.8,3X,
     c            '$ Burnable poison composition for node ',
     c            I2,' of assembly ',A9)
               ELSE
                  WRITE(70,708)
     c            BPMATZAIDS(BANKNUM(COLUMN,ROW),ISO),
     c            (-1*BPMATWTS(BANKNUM(COLUMN,ROW),ISO))
 708              FORMAT(T8,A9,T20,G14.8)
               ENDIF
 709        CONTINUE
         ENDIF
            BPML=BPML+1
            IF ((BPABSMAT(BANKNUM(COLUMN,ROW))).EQ.1) THEN
               BPDENTOGO(COLUMN,ROW,NODE)=BPDEN(COLUMN,ROW)
            ELSEIF ((BPABSMAT(BANKNUM(COLUMN,ROW))).NE.1) THEN
               BPDENTOGO(COLUMN,ROW,NODE)=
     c         BPMATDATA(BANKNUM(COLUMN,ROW),1)
            ENDIF
         ELSEIF ((BPGO(COLUMN,ROW,NODE).EQ..TRUE.).AND.
     c   (STAT(COLUMN,ROW).NE.'F')) THEN
* Get B-10 and B-11 composition data for BPRA
      COMMAND1(1:11)='rm ./*.temp'
      RES=SYSTEM(COMMAND1)
      N=0
      DO 762 C=1,5
         IF (ASSYID(COLUMN,ROW)(C:C).NE.' ') THEN
            N=N+1
         ENDIF
 762  CONTINUE
         COMMAND2(1:6)='cp  ./'
         DO 764 C=1,N
            COMMAND2((C+6):(C+6))=ASSYID(COLUMN,ROW)(C:C)
 764     CONTINUE
         N1=INT(NODE/10.0)
         CHN1=CHAR(N1+48)
         N2=INT(NODE-(N1*10))
         CHN2=CHAR(N2+48)
         COMMAND2((N+7):(N+14))='/??????N'
         COMMAND2((N+15):(N+15))=CHN1
         COMMAND2((N+16):(N+16))=CHN2
         COMMAND2((N+17):(N+26))='????????AC'
```

```
            COMMAND2((N+27):(N+28))=CYCLE
            COMMAND2((N+29):(N+29))='T'
            NUMSTPT1=INT(EFPD/100.0)
            CHSTPT1=CHAR(NUMSTPT1+48)
            NUMSTPT2=INT((EFPD-(NUMSTPT1*100))/10.0)
            CHSTPT2=CHAR(NUMSTPT2+48)
            NUMSTPT3=INT((EFPD-(NUMSTPT1*100)-
     c      (NUMSTPT2*10)))
            CHSTPT3=CHAR(NUMSTPT3+48)
            COMMAND2((N+30):(N+30))=CHSTPT1
            COMMAND2((N+31):(N+31))=CHSTPT2
            COMMAND2((N+32):(N+32))=CHSTPT3
            COMMAND2((N+33):(N+40))='.cut ./N'
            COMMAND2((N+41):(N+41))=CHN1
            COMMAND2((N+42):(N+42))=CHN2
            COMMAND2((N+43):(N+47))='.temp'
            DO 766 C=(N+48),54
              COMMAND2(C:C)=' '
 766        CONTINUE
            RES=SYSTEM(COMMAND2)
            FILENAME(1:8)=COMMAND2((N+40):(N+47))
            OPEN (UNIT=300, FILE=FILENAME, STATUS='OLD')
*
*
            REWIND(UNIT=300)
              BPRAISOVALUE(1)=0.0
              BPRAISOVALUE(2)=0.0
              BPRAISONAME(1)='      '
              BPRAISONAME(2)='      '
              REWIND(300)
              BPFIND=.FALSE.
              DO 800 WHILE (BPFIND.EQ..FALSE.)
                READ(300,780) BPLABEL
 780            FORMAT (T98,A14)
                IF (BPLABEL.EQ.'light elements') THEN
                  READ(300,*)
                  READ(300,790) FORMATLABEL
 790              FORMAT (T46,A29)
                  IF (FORMATLABEL.EQ.
     c            'nuclide concentrations, grams') THEN
                    BPFIND=.TRUE.
                  ENDIF
                ENDIF
 800          CONTINUE
              DO 820 CT1=1,25
                READ (300,810) BPRAISONAME(1)
 810            FORMAT(T6,A6)
                IF (BPRAISONAME(1).EQ.' b 10 ') THEN
                  BACKSPACE(300)
                  EXIT
                ENDIF
 820          CONTINUE
              READ (300,830) BPRAISONAME(1), BPRAISOVALUE(1)
 830          FORMAT(T6,A6,T<COLUMNSTART>,G10.2)
```

```
          READ (300,840) BPRAISONAME(2), BPRAISOVALUE(2)
 840      FORMAT(T6,A6,T<COLUMNSTART>,G10.2)
          IF (BPRAISONAME(1).EQ.' b 10 ') THEN
             BPRAZAID(COLUMN,ROW,NODE,1)=' 5010.50c'
             BPNODECOMP(COLUMN,ROW,NODE,1)=BPRAISOVALUE(1)
          ENDIF
          IF (BPRAISONAME(2).EQ.' b 11 ') THEN
             BPRAZAID(COLUMN,ROW,NODE,2)=' 5011.56c'
             BPNODECOMP(COLUMN,ROW,NODE,2)=BPRAISOVALUE(2)
          ENDIF
        IF ((BPABSMAT(BANKNUM(COLUMN,ROW))).EQ.1) THEN
          CMASSPERMOLB4C=1.008664904*11.8969
          BMASSPERMOLB4C=4*((1.008664904*9.9269*0.194)+
     c    (1.008664904*10.9147*0.806))
          ALMASSPERMOLAL2O3=2*1.008664904*26.75
          OMASSPERMOLAL2O3=3*1.008664904*15.858
          CWTINB4C=(CMASSPERMOLB4C/
     c    (BMASSPERMOLB4C+CMASSPERMOLB4C))*100.0
          ALWTINAL2O3=(ALMASSPERMOLAL2O3/
     c    (ALMASSPERMOLAL2O3+OMASSPERMOLAL2O3))*100.0
          OWTINAL2O3=(OMASSPERMOLAL2O3/
     c    (ALMASSPERMOLAL2O3+OMASSPERMOLAL2O3))*100.0
          CWT=BPRICH(COLUMN,ROW)*CWTINB4C/100.0
          ALWT=ALWTINAL2O3*(100.0-BPRICH(COLUMN,ROW))/100.0
          OWT=OWTINAL2O3*(100.0-BPRICH(COLUMN,ROW))/100.0
          IF (BANDW.EQ..TRUE.) THEN
             BPVOL=(PI)*(BPRAD(COLUMN,ROW)**2)*
     c       NODEHEIGHT(NODE)*BPROD
          ELSEIF (WESTINGHOUSE.EQ..TRUE.) THEN
             IF (WBPRA(BANKNUM(COLUMN,ROW)).EQ.1) THEN
                BPAREA=(PI)*
     c          (BPRDIM(BANKNUM(COLUMN,ROW),1)**2)
             ELSEIF ((WBPRA(BANKNUM(COLUMN,ROW)).EQ.2).OR.
     c       (WBPRA(BANKNUM(COLUMN,ROW)).EQ.3)) THEN
                BPAREA=((PI)*
     c          (BPRAXDIM(BANKNUM(COLUMN,ROW),4)**2))-
     c          ((PI)*
     c          (BPRAXDIM(BANKNUM(COLUMN,ROW),3)**2))
             ENDIF
             IF (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.1) THEN
                BPVOL=BPAREA*4*NODEHEIGHT(NODE)
             ELSEIF (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.2) THEN
                BPVOL=BPAREA*8*NODEHEIGHT(NODE)
             ELSEIF (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.3) THEN
                BPVOL=BPAREA*9*NODEHEIGHT(NODE)
             ELSEIF (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.4) THEN
                BPVOL=BPAREA*10*NODEHEIGHT(NODE)
             ELSEIF (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.5) THEN
                BPVOL=BPAREA*12*NODEHEIGHT(NODE)
             ELSEIF (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.6) THEN
                BPVOL=BPAREA*16*NODEHEIGHT(NODE)
             ELSEIF (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.7) THEN
                BPVOL=BPAREA*20*NODEHEIGHT(NODE)
             ENDIF
```

```
                  ENDIF
                  BPMASSTOTAL=((ALWT+OWT+CWT)*BPDEN(COLUMN,ROW)*
     c            BPVOL/100.0)+BPNODECOMP(COLUMN,ROW,NODE,1)+
     c            BPNODECOMP(COLUMN,ROW,NODE,2)
                  ALWT=(ALWT*BPDEN(COLUMN,ROW)*BPVOL)/BPMASSTOTAL
                  OWT=(OWT*BPDEN(COLUMN,ROW)*BPVOL)/BPMASSTOTAL
                  CWT=(CWT*BPDEN(COLUMN,ROW)*BPVOL)/BPMASSTOTAL
                  B10WT=BPNODECOMP(COLUMN,ROW,NODE,1)*100.0/
     c            BPMASSTOTAL
                  B11WT=BPNODECOMP(COLUMN,ROW,NODE,2)*100.0/
     c            BPMASSTOTAL
                  DO 900 ISO=1,2
                     IF (ISO.EQ.1) THEN
                        WRITE(70,850) BPML, (-1*ALWT), NODE,
     c                  ASSYID(COLUMN,ROW), BPRICH(COLUMN,ROW)
 850                    FORMAT(T1,'M',I4,T8,'13027.50c',T20,G14.8,3X,
     c                  '$ Burnable poison composition for node ',
     c                  I2,' of assembly ',A5,', ',F7.3,
     c                  ' WT% B4C IN ORIGINAL B4C-AL2O3')
                     ELSE
                        WRITE(70,860) (-1*OWT)
 860                    FORMAT(T8,' 8016.50c',T20,G14.8)
                        WRITE(70,870) (-1*CWT)
 870                    FORMAT(T8,' 6000.50c',T20,G14.8)
                        WRITE(70,880) (-1*B10WT)
 880                    FORMAT(T8,' 5010.50c',T20,G14.8)
                        WRITE(70,890) (-1*B11WT)
 890                    FORMAT(T8,' 5011.56c',T20,G14.8)
                     ENDIF
 900              CONTINUE
               ELSEIF ((BPABSMAT(BANKNUM(COLUMN,ROW))).NE.1) THEN
                  IF (BANDW.EQ..TRUE.) THEN
                     BPVOL=(PI)*(BPRAD(COLUMN,ROW)**2)*
     c               NODEHEIGHT(NODE)*BPROD
                  ELSEIF (WESTINGHOUSE.EQ..TRUE.) THEN
                     IF (WBPRA(BANKNUM(COLUMN,ROW)).EQ.1) THEN
                        BPAREA=(PI)*
     c                  (BPRDIM(BANKNUM(COLUMN,ROW),1)**2)
                     ELSEIF ((WBPRA(BANKNUM(COLUMN,ROW)).EQ.2).OR.
     c               (WBPRA(BANKNUM(COLUMN,ROW)).EQ.3)) THEN
                        BPAREA=((PI)*
     c                  (BPRAXDIM(BANKNUM(COLUMN,ROW),4)**2))-
     c                  ((PI)*
     c                  (BPRAXDIM(BANKNUM(COLUMN,ROW),3)**2))
                     ENDIF
                     IF (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.1) THEN
                        BPVOL=BPAREA*4*NODEHEIGHT(NODE)
                     ELSEIF (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.2) THEN
                        BPVOL=BPAREA*8*NODEHEIGHT(NODE)
                     ELSEIF (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.3) THEN
                        BPVOL=BPAREA*9*NODEHEIGHT(NODE)
                     ELSEIF (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.4) THEN
                        BPVOL=BPAREA*10*NODEHEIGHT(NODE)
                     ELSEIF (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.5) THEN
```

```
                BPVOL=BPAREA*12*NODEHEIGHT(NODE)
             ELSEIF (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.6) THEN
                BPVOL=BPAREA*16*NODEHEIGHT(NODE)
             ELSEIF (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.7) THEN
                BPVOL=BPAREA*20*NODEHEIGHT(NODE)
             ENDIF
          ENDIF
          BPMASSTOTAL=0.0
          DO 902 ISO=1,BPMATDATA(BANKNUM(COLUMN,ROW),2)
             IF ((BPMATZAIDS(BANKNUM(COLUMN,ROW),ISO).NE.
     c          (' 5010.50C')).OR.
     c          (BPMATZAIDS(BANKNUM(COLUMN,ROW),ISO).NE.
     c          (' 5010.50c')).OR.
     c          (BPMATZAIDS(BANKNUM(COLUMN,ROW),ISO).NE.
     c          (' 5011.56C')).OR.
     c          (BPMATZAIDS(BANKNUM(COLUMN,ROW),ISO).NE.
     c          (' 5011.56c'))) THEN
                BPMASSTOTAL=BPMASSTOTAL+
     c          ((BPMATWTS(BANKNUM(COLUMN,ROW),ISO))*
     c          BPMATDATA(BANKNUM(COLUMN,ROW),1)*
     c          BPVOL/100.0)
             ENDIF
 902      CONTINUE
          BPMASSTOTAL=BPMASSTOTAL+
     c    BPNODECOMP(COLUMN,ROW,NODE,1)+
     c    BPNODECOMP(COLUMN,ROW,NODE,2)
          B10WT=BPNODECOMP(COLUMN,ROW,NODE,1)*100.0/
     c    BPMASSTOTAL
          B11WT=BPNODECOMP(COLUMN,ROW,NODE,2)*100.0/
     c    BPMASSTOTAL
          DO 916 ISO=1,BPMATDATA(BANKNUM(COLUMN,ROW),2)
             IF (ISO.EQ.1) THEN
                IF ((BPMATZAIDS(BANKNUM(COLUMN,ROW),ISO).NE.
     c          (' 5010.50C')).OR.
     c          (BPMATZAIDS(BANKNUM(COLUMN,ROW),ISO).NE.
     c          (' 5010.50c')).OR.
     c          (BPMATZAIDS(BANKNUM(COLUMN,ROW),ISO).NE.
     c          (' 5011.56C')).OR.
     c          (BPMATZAIDS(BANKNUM(COLUMN,ROW),ISO).NE.
     c          (' 5011.56c'))) THEN
                   WRITE(70,904) BPML,
     c             BPMATZAIDS(BANKNUM(COLUMN,ROW),ISO),
     c             (-1*BPMATWTS(BANKNUM(COLUMN,ROW),ISO)*
     c             BPMATDATA(BANKNUM(COLUMN,ROW),1)*
     c             BPVOL/BPMASSTOTAL), NODE,
     c             ASSYID(COLUMN,ROW)
 904               FORMAT(T1,'M',I4,T8,A9,T20,G14.8,3X,
     c             '$ Burnable poison composition for node ',
     c             I2,' of assembly ',A5)
                ELSEIF ((BPMATZAIDS(BANKNUM(COLUMN,ROW),ISO).EQ.
     c          (' 5010.50C')).OR.
     c          (BPMATZAIDS(BANKNUM(COLUMN,ROW),ISO).EQ.
     c          (' 5010.50c'))) THEN
                   WRITE(70,906) BPML, (-1*B10WT), NODE,
```

```
     c                    ASSYID(COLUMN,ROW)
 906                      FORMAT(T1,'M',I4,T8,' 5010.50c',T20,G14.8,3X,
     c                    '$ Burnable poison composition for node ',
     c                    I2,' of assembly ',A5)
                       ELSEIF ((BPMATZAIDS(BANKNUM(COLUMN,ROW),ISO).EQ.
     c                 (' 5011.56C')).OR.
     c                 (BPMATZAIDS(BANKNUM(COLUMN,ROW),ISO).EQ.
     c                 (' 5011.56c'))) THEN
                          WRITE(70,908) BPML, (-1*B11WT), NODE,
     c                    ASSYID(COLUMN,ROW)
 908                      FORMAT(T1,'M',I4,T8,' 5011.56c',T20,G14.8,3X,
     c                    '$ Burnable poison composition for node ',
     c                    I2,' of assembly ',A5)
                       ENDIF
                    ELSE
                       IF ((BPMATZAIDS(BANKNUM(COLUMN,ROW),ISO).EQ.
     c                 (' 5010.50C')).OR.
     c                 (BPMATZAIDS(BANKNUM(COLUMN,ROW),ISO).EQ.
     c                 (' 5010.50c'))) THEN
                          WRITE(70,910) (-1*B10WT)
 910                      FORMAT(T8,' 5010.50c',T20,G14.8)
                       ELSEIF ((BPMATZAIDS(BANKNUM(COLUMN,ROW),ISO).EQ.
     c                 (' 5011.56C')).OR.
     c                 (BPMATZAIDS(BANKNUM(COLUMN,ROW),ISO).EQ.
     c                 (' 5011.56c'))) THEN
                          WRITE(70,912) (-1*B11WT)
 912                      FORMAT(T8,' 5011.56c',T20,G14.8)
                       ELSE
                          WRITE(70,914)
     c                    BPMATZAIDS(BANKNUM(COLUMN,ROW),ISO),
     c                    (-1*BPMATWTS(BANKNUM(COLUMN,ROW),ISO)*
     c                    BPMATDATA(BANKNUM(COLUMN,ROW),1)*
     c                    BPVOL/BPMASSTOTAL)
 914                      FORMAT(T8,A9,T20,G14.8)
                       ENDIF
                    ENDIF
 916             CONTINUE
              ENDIF
                 BPML=BPML+1
                 BPDENTOGO(COLUMN,ROW,NODE)=BPMASSTOTAL/BPVOL
              ELSEIF (BPRUNIQUE(COLUMN,ROW).EQ..TRUE.) THEN
                 IF ((BPGO(COLUMN,ROW,NODE).EQ..FALSE.).AND.
     c           (BPRABSNODE(BANKNUM(COLUMN,ROW),NODE).EQ.'Y')) THEN
               IF (BPNONABSMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
                 ALMASSPERMOLAL2O3=2*1.008664904*26.75
                 OMASSPERMOLAL2O3=3*1.008664904*15.858
                 ALWTINAL2O3=(ALMASSPERMOLAL2O3/
     c           (ALMASSPERMOLAL2O3+OMASSPERMOLAL2O3))*100.0
                 OWTINAL2O3=(OMASSPERMOLAL2O3/
     c           (ALMASSPERMOLAL2O3+OMASSPERMOLAL2O3))*100.0
                 ALWT=ALWTINAL2O3
                 OWT=OWTINAL2O3
                 DO 922 ISO=1,2
                    IF (ISO.EQ.1) THEN
```

```
                    WRITE(70,918) BPML, (-1*ALWT), NODE,
     c              ASSYID(COLUMN,ROW)
 918                FORMAT(T1,'M',I4,T8,'13027.50c',T20,G14.8,3X,
     c              '$ Non-absorbing burnable poison',
     c              'composition for node ',
     c              I2,' of assembly ',A5)
                 ELSE
                    WRITE(70,920) (-1*OWT)
 920                FORMAT(T8,' 8016.50c',T20,G14.8)
                 ENDIF
 922          CONTINUE
            ELSEIF (BPNONABSMAT(BANKNUM(COLUMN,ROW)).NE.1) THEN
              DO 928 ISO=1,NONBPMATDATA(BANKNUM(COLUMN,ROW),2)
                 IF (ISO.EQ.1) THEN
                    WRITE(70,924) BPML,
     c              NONBPMATZAIDS(BANKNUM(COLUMN,ROW),ISO),
     c              (-1*NONBPMATWTS(BANKNUM(COLUMN,ROW),ISO)),
     c              NODE, ASSYID(COLUMN,ROW)
 924                FORMAT(T1,'M',I4,T8,A9,T20,G14.8,3X,
     c              '$ Non-absorbing burnable poison',
     c              'composition for node ',
     c              I2,' of assembly ',A5)
                 ELSE
                    WRITE(70,926)
     c              NONBPMATZAIDS(BANKNUM(COLUMN,ROW),ISO),
     c              (-1*NONBPMATWTS(BANKNUM(COLUMN,ROW),ISO))
 926                FORMAT(T8,A9,T20,G14.8)
                 ENDIF
 928          CONTINUE
            ENDIF
              BPML=BPML+1
              ENDIF
           ENDIF
           CLOSE(UNIT=300)
 930       CONTINUE
         ENDIF
 940   CONTINUE
 950 CONTINUE
     COMMAND3(1:11)='rm ./*.temp'
     RES=SYSTEM(COMMAND3)
     CLOSE(UNIT=60)
     CLOSE(UNIT=70)
     CLOSE(UNIT=80)
     CLOSE(UNIT=90)

     RETURN
     END

*******************************************************************
*    SUBROUTINE GEOSECTION                                        *
*    This subroutine writes the geometry section of the MCNP      *
*    input deck.                                                  *
*******************************************************************
     SUBROUTINE GEOSECTION (NUMREGABOVE, NUMOFLOWREG,
```

```
c SYSTEMWEST, BANKNUM,
c DESNUM, NUMOFMCNPFUELNODES,
c CLADMATERIAL, NUMREGABOVEBPRA, NUMOFBANKS,
c BANKID, NUMREGABOVECRA, NUMREGABOVEAPSRA,
c NUMOFSPACERS,
c EFPD, SURFVALUESPEC, PVCLADIR, PVCLADTHICKNESS,
c PVTHICKNESS, REGABOVE, SPACERDIST,
c ENDFITHEIGHT, LOWERREGION, PVDENSITY,
c PVCLADDENSITY, SHIELDIR, SHIELDTHICKNESS,
c MODDENSITY, SHIELDDENSITY, BARRELIR, BARRELTHICKNESS,
c BARRELDENSITY, PINPITCH, MCNPFUELHEIGHT,
c FUELRADIUS, CLADRADIUS, ASSYPLENUM,
c ENDCAPHEIGHT, FRUPLENMAT, FRLPLENMAT,
c LEFMAT, UEFMAT, REGABOVEBPRA,
c REGABOVECRA, REGABOVEAPSRA,
c SPACERHEIGHT,
c PREFIX, CYCLE, GEOFILE, SURFTYPESPEC,
c ASSYID, BANKDES,
c EIGHTH, QUARTER, FULL, BANDW, WESTINGHOUSE, CE,
c BOTBPNODEHEIGHT, MCNPBPRAHEIGHT,
c BPRDIM, BPRPLEN, BPRCLADMAT,
c GTDATA, GTMAT, NUMOFBPRANODES,
c BPRUPLENMAT, BPRLPLENMAT, ITDATA,
c ITMAT, CRADIM, CRABSMAT,
c CRUPLENMAT, CRLPLENMAT, CRCLADMAT, APSRADIM,
c APSRABSMAT, APSRCLADMAT, APSRUPLENMAT, APSRLPLENMAT,
c APSRIPLUGDEN, APSRIPLUGFRAC,
c BAFFLEDENSITY, PVOUTERSURF, SYSTEMSOUTH, SYSTEMTOP,
c SYSTEMBOTTOM, SN, ASSYPITCH, BAFFLESEPARATION,
c BAFFLETHICKNESS, NUMOFNODES, ISOREQUEST,
c NODEHEIGHT,
c GRAMS, ENRICHMENT, STAT, RODNUM, MN,
c LOWERREGIONZAIDS, SHIELDWTS, LOWERREGIONWTS,
c UEFWTS, BAFFLEZAIDS, PVZAIDS, LEFZAIDS, FRLPLENZAIDS,
c SHIELDZAIDS, PVCLADZAIDS, ABOVEWTS, FRUPLENZAIDS,
c PVCLADWTS, FRUPLENWTS, LEFWTS, PVWTS, BARRELWTS,
c ABOVEZAIDS, BARRELZAIDS, BAFFLEWTS, FRLPLENWTS,
c UEFZAIDS, SPACERMAT, SPACERVOL, PPMB, PVISONUM,
c PVCLADISONUM, SHIELDISONUM, BARRELISONUM, BAFFLEISONUM,
c CRABSZAIDS, CRABSWTS, CRUPZS, CRUPLENWTS,
c CRLPLENZAIDS, CRLPLENWTS, APSRUPLENWTS, APSRLPLENZAIDS,
c APSRUPLENZAIDS, APSRLPLENWTS, APSRABSWTS, APSRABSZAIDS,
c BPRLPLENZAIDS, BPRUPLENZAIDS, BPRLPLENWTS, BPRUPLENWTS,
c ABOVEBPRAZAIDS, ABOVEBPRAWTS, ABOVECRAZAIDS, ABOVECRAWTS,
c ABOVEAPSRAZAIDS, ABOVEAPSRAWTS, FUELNODEDEN, BPDENTOGO,
c BPRABSNODE, BPNONABSMAT, NONBPMATDATA, AL2O3DENSITY,
c PITCH, MATFILE, NUMOFFADESIGNS, GTAXDATA, GTSPLIT,
c NUMOFGTAXS, GTAXMAT, ITAXDATA, ITSPLIT,
c NUMOFITAXS, ITAXMAT, CRAXDIM, CRABSAXMAT,
c CRABSAXWTS, HYBRID, HYBFLAG, NUMCRAXS, CRAXCLADMAT,
c CRABSAXZAIDS, BPRAXDIM, WBPRATYPE, WBPRA,
c AL2O3B4CDENSITY, SPM4SS, SPM4INC, SPM4ZR)

  INTEGER
```

```
c NUMREGABOVE, NUMOFLOWREG, PVML, SYSTEMWEST, PVCML, BMODML,
c TSML, BARRELML, LOWREGML(10), BAFFLEML, BANKNUM(50,50),
c DESNUM(50,50), NUMOFMCNPFUELNODES(20),
c CLADMATERIAL(20), CLADML(50,50),
c FRUPML(50,50), FRLPML(50,50), FRLEFML(50,50), FRUEFML(50,50),
c NUMREGABOVEBPRA, FRUREGIONML(50,50,20), NUMOFBANKS,
c BANKID(20), NUMREGABOVECRA, NUMREGABOVEAPSRA,
c NUMOFSPACERS(20), HOMOSPACERML,
c NUMSTPT1, NUMSTPT2, NUMSTPT3, LN, MN, SN, PVIRSURF, PVORSURF,
c SYSTEMSOUTH, SYSTEMTOP, REGION, SYSTEMBOTTOM,
c PVCIRSURF, TSORSURF, TSIRSURF, BORSURF, BIRSURF, UN,
c FCOREBOTSURF, FCOREUNIV, LOWREGSURF(10), COLUMN, ROW,
c ASSYUNIV(50,50), COREWIDTHOLD, COREWIDTH, BAFQ1P1UNIV,
c BAFQ1P2UNIV, BAFQ1P3UNIV, BAFQ1P4UNIV, BAFQ2P1UNIV,
c BAFQ2P2UNIV, BAFQ2P3UNIV, BAFQ2P4UNIV, BAFQ3P1UNIV,
c BAFQ3P2UNIV, BAFQ3P3UNIV, BAFQ3P4UNIV, BAFQ4P1UNIV,
c BAFQ4P2UNIV, BAFQ4P3UNIV, BAFQ4P4UNIV, CLUCNORTHSURF,
c CLUCSOUTHSURF, CLUCEASTSURF, CLUCWESTSURF, LATWIDTH,
c COUNT, BAFACDNORTHSURF, BAFACDSOUTHSURF, BAFBCDEASTSURF,
c BAFBCDWESTSURF, BAFACDFGNORTHSURF, BAFACDFGSOUTHSURF,
c BAFBCDKLEASTSURF, BAFBCDKLWESTSURF, BAFHIJKLNORTHSURF,
c BAFHIJKLSOUTHSURF, BAFEFGIJEASTSURF, BAFEFGIJWESTSURF,
c FRUN, CRUN, BPRUN, APSRUN, GTUN, ITUN, FRUNIV(50,50),
c GTUNIV(50,50), ITUNIV(50,50), CRAUNIV(50,50),
c BPRAUNIV(50,50), APSRAUNIV(50,50), RO, CO, DNUM,
c SURFNUM, PITCHNORTH, PITCHSOUTH, PITCHEAST,
c PITCHWEST, MCNPNODE, Z, CURRENTSURFLABEL, V,
c TOPNODETOPSURF, NODETOPSURF, NODEBOTTOMSURF, RADIUS,
c CLADIRSURF, CLADORSURF, CLADTOPSURF, CLADBOTTOMSURF,
c PLENUMTOPSURF, PLENUMBOTTOMSURF, UEFBOTTOMSURF,
c UEFTOPSURF, REGIONTOPSURF, REGIONBOTTOMSURF, BN, SPN,
c SPACERTOPSURF, SPACERBOTTOMSURF, WATERREGIONTOPSURF,
c WATERREGIONBOTTOMSURF, BPNODEML,
c TOPBPNODETOPSURF, BPNODETOPSURF, BPNODEBOTTOMSURF,
c BPRADIUS, BPCLADIRSURF, BPCLADORSURF, BPCLADTOPSURF,
c BPCLADBOTTOMSURF, BPRCLADMAT(20), BPCLADML(50,50),
c BPRUPML(50,50), BPRLPML(50,50), GTTOPSURF,
c GTBOTSURF, GTORSURF, GTIRSURF, BPLEFTOPSURF,
c GTMAT(20), GTML(50,50),
c NUMOFBPRANODES(20), GTLEFML(50,50), GTUEFML(50,50),
c ITTOPSURF, ITBOTSURF, ITORSURF, ITIRSURF,
c ITLEFML(50,50), ITUEFML(50,50), ITMAT(20), ITML(50,50),
c CRABSSURF, CRCLADIRSURF, CRCLADORSURF, CRCLADTOPSURF,
c CRCLADBOTTOMSURF, CRABSML,
c CRCLADML(50,50), CRUPML(50,50), CRLPML(50,50),
c CRLEFTOPSURF, CRCLADMAT(20),
c APSRABSSURF, APSRCLADIRSURF, APSRCLADORSURF,
c APSRCLADBOTTOMSURF,
c APSRABSBOTTOMSURF, APSRABSTOPSURF, APSRUPTOPSURF,
c APSRIPLUGTOPSURF, APSRCLADTOPSURF, APSRABSML,
c APSRCLADML(50,50), APSRCLADMAT(20), APSRUPML(50,50),
c APSRLPML(50,50), APSRIPLUGML(50,50),
c APSRLEFTOPSURF,
c NUMOFNODES, ISOREQUEST, FMN, BPMN,
```

```
     c RODNUM(20), PVISONUM, PVCLADISONUM, SHIELDISONUM,
     c BARRELISONUM, BAFFLEISONUM, SPACERMAT(20,10),
     c HOMOSPACMLNUM(20,15), FUELNODEML, CRABSTOPSURF,
     c CRABSBOTTOMSURF, BPNONABSMAT(20), NUMOFFADESIGNS,
     c COMBOVAL(20), LGTS, LGTSOR, UGTS, UGTSOR,
     c GTSECTTOPSURF(5), GTSECTBOTSURF(5), GTSECTORSURF(5),
     c GTSECTIRSURF(5), GTSPLIT, NUMOFGTAXS(20),
     c GTAXMAT(20,5), GTAXML(50,50,5), LITS, LITSOR,
     c UITS, UITSOR,
     c ITSECTTOPSURF(5), ITSECTBOTSURF(5), ITSECTORSURF(5),
     c ITSECTIRSURF(5), ITSPLIT, NUMOFITAXS(20),
     c ITAXMAT(20,5), ITAXML(50,50,5), HYBRID, HYBFLAG(20),
     c NUMCRAXS(20), CRAXCLADMAT(20,20), WBPRA(20),
     c WBPRATYPE(20)
*
      REAL
     c EFPD, SURFVALUESPEC(200), PVCLADIR, PVCLADTHICKNESS,
     c PVTHICKNESS, REGABOVE(20,3), SPACERDIST(20,10),
     c ENDFITHEIGHT(20,2), LOWERREGION(10,3), PVDENSITY,
     c PVCLADDENSITY, SHIELDIR, SHIELDTHICKNESS,
     c MODDENSITY, SHIELDDENSITY, BARRELIR, BARRELTHICKNESS,
     c BARRELDENSITY, PINPITCH(20), MCNPFUELHEIGHT(20,50),
     c FUELRADIUS(20), CLADRADIUS(20,2), ASSYPLENUM(20,2),
     c ENDCAPHEIGHT(20,2), FRUPLENMAT(20,2), FRLPLENMAT(20,2),
     c LEFMAT(20,2), UEFMAT(20,2), REGABOVEBPRA(20,3),
     c REGABOVECRA(20,3), REGABOVEAPSRA(20,3),
     c SPACERHEIGHT(20,10), HOMOSPACERDEN(20,15), TEMP,
     c TOTFUELHEIGHT, CURRENTSURF, CLADRHO,
     c BOTBPNODEHEIGHT(20), MCNPBPRAHEIGHT(20,50),
     c BPRDIM(20,3), BPRPLEN(20,2), GTDATA(20,4),
     c BPRUPLENMAT(20,2), BPRLPLENMAT(20,2), ITDATA(20,4),
     c CRADIM(20,7), CRABSMAT(20,2), CRUPLENMAT(20,2),
     c CRLPLENMAT(20,2), APSRADIM(20,11), APSRABSMAT(20,2),
     c APSRUPLENMAT(20,2), APSRLPLENMAT(20,2),
     c APSRIPLUGDEN(20), APSRIPLUGFRAC(20), BAFFLEDENSITY,
     c ASSYPITCH, BAFFLESEPARATION, BAFFLETHICKNESS, PITCH,
     c NODEHEIGHT(50),
     c GRAMS(20), ENRICHMENT(50,50),
     c LOWERREGIONWTS(10,35), SHIELDWTS(35), UEFWTS(20,35),
     c ABOVEWTS(20,35), PVCLADWTS(35), FRUPLENWTS(20,35),
     c LEFWTS(20,35), PVWTS(35), BARRELWTS(35),
     c BAFFLEWTS(35), FRLPLENWTS(20,35),
     c SPACHEIGHT, PI, SPACERVOL(20,10),
     c SPACVOL, MODVOL, VAL1, VAL2, VAL3, VAL4, PPMB,
     c SPACMASS, MODMASS, SPACFRAC, MODFRAC,
     c CRABSWTS(20,35), CRUPLENWTS(20,35), CRLPLENWTS(20,35),
     c APSRUPLENWTS(20,35), APSRLPLENWTS(20,35),
     c APSRABSWTS(20,35), FUELNODEDEN(50,50,50),
     c BPRLPLENWTS(20,35), BPRUPLENWTS(20,35),
     c ABOVEBPRAWTS(20,35), ABOVECRAWTS(20,35),
     c ABOVEAPSRAWTS(20,35), VAL5, BPDENTOGO(50,50,50),
     c NONBPMATDATA(20,2), AL2O3DENSITY(20),
     c OWTINH2O, HWTINH2O, B10WTINH2O, B11WTINH2O, B10WTINB,
     c B11WTINB, OWT, HWT, B10WT, B11WT, CRWT, FEWT, ZRWT,
```

```
     c SNWT, CWT, NWT, SIWT, PWT, SWT, MNWT, NIWT, ALWT,
     c TIWT, COWT, CUWT, NBWT, MOWT, TAWT, BWTINH2O, TVOL,
     c CR50, CR52, CR53, CR54, FE54, FE56, FE57, FE58,
     c NI58, NI60, NI61, NI62, NI64, CU63, CU65,
     c GTAXDATA(20,4,5), ITAXDATA(20,4,5), CRAXDIM(20,5,20),
     c CRABSAXMAT(20,2,20), CRABSAXWTS(20,35,20),
     c BPRAXDIM(20,6), AL2O3B4CDENSITY(20),
     c SPM4SS(20,10), SPM4INC(20,10), SPM4ZR(20,10),
     c SPINCMASS, SPSSMASS, SPINCFRAC, SPSSFRAC
*
      CHARACTER
     c PREFIX*3, CYCLE*2, GEOFILE*14, SURFTYPESPEC(200)*2,
     c ASSYID(50,50)*5, BANKDES(20)*5,
     c CHSTPT1*1, CHSTPT2*1, CHSTPT3*1,
     c STAT(50,50)*1, MATFILE*14,
     c LOWERREGIONZAIDS(10,35)*9,
     c BAFFLEZAIDS(35)*9, PVZAIDS(35)*9, LEFZAIDS(20,35)*9,
     c FRLPLENZAIDS(20,35)*9, SHIELDZAIDS(35)*9,
     c PVCLADZAIDS(35)*9, FRUPLENZAIDS(20,35)*9,
     c ABOVEZAIDS(20,35)*9, BARRELZAIDS(35)*9,
     c UEFZAIDS(20,35)*9, CRABSZAIDS(20,35)*9,
     c CRUPZS(20,35)*9, CRLPLENZAIDS(20,35)*9,
     c APSRLPLENZAIDS(20,35)*9, APSRUPLENZAIDS(20,35)*9,
     c APSRABSZAIDS(20,35)*9, BPRLPLENZAIDS(20,35)*9,
     c BPRUPLENZAIDS(20,35)*9, ABOVEBPRAZAIDS(20,35)*9,
     c ABOVECRAZAIDS(20,35)*9, ABOVEAPSRAZAIDS(20,35)*9,
     c BPRABSNODE(20,50)*1, CRABSAXZAIDS(20,35,20)*9
*
      LOGICAL
     c EIGHTH, QUARTER, FULL, BANDW, WESTINGHOUSE, CE,
     c FUNIQUE(50,50), DUNIQUE(50,50), BPRAINSERTED, CRAINSERTED,
     c APSRAINSERTED, CRUNIQUE(50,50), APSRUNIQUE(50,50), BPRAFLAG,
     c CRAFLAG, APSRAFLAG, CLADMLUNIQUE, FRUPMLUNIQUE,
     c FRLPMLUNIQUE, FRLEFMLUNIQUE, FRUEFMLUNIQUE,
     c FRUREGIONMLUNIQUE, LEAVE, BPRUNIQUE(50,50), BPRUPMLUNIQUE,
     c BPRLPMLUNIQUE, CRUPMLUNIQUE, CRLPMLUNIQUE,
     c APSRUPMLUNIQUE, APSRLPMLUNIQUE, PLAINFLAG, COMBOFLAG(20),
     c GTWRITE(50,50)
*
      PI=3.1415926536
      FUELNODEML=0
      BPNODEML=3000
*     Open the file to contain the geometry specification section of the MCNP
input deck.
      GEOFILE(1:3)=PREFIX
      GEOFILE(4:4)='C'
      GEOFILE(5:6)=CYCLE
      GEOFILE(7:7)='T'
      NUMSTPT1=INT(EFPD/100.0)
      CHSTPT1=CHAR(NUMSTPT1+48)
      NUMSTPT2=INT((EFPD-(NUMSTPT1*100))/10.0)
      CHSTPT2=CHAR(NUMSTPT2+48)
      NUMSTPT3=INT(EFPD-(NUMSTPT1*100)-
     c (NUMSTPT2*10))
```

```
      CHSTPT3=CHAR(NUMSTPT3+48)
      GEOFILE(8:8)=CHSTPT1
      GEOFILE(9:9)=CHSTPT2
      GEOFILE(10:10)=CHSTPT3
      GEOFILE(11:14)='.geo'
      OPEN(UNIT=30, FILE=GEOFILE, STATUS='UNKNOWN')
      REWIND(UNIT=30)
*
*     Open file to contain the material section of the MCNP input deck.
      MATFILE(1:3)=PREFIX
      MATFILE(4:4)='C'
      MATFILE(5:6)=CYCLE
      MATFILE(7:7)='T'
      NUMSTPT1=INT(EFPD/100.0)
      CHSTPT1=CHAR(NUMSTPT1+48)
      NUMSTPT2=INT((EFPD-(NUMSTPT1*100))/10.0)
      CHSTPT2=CHAR(NUMSTPT2+48)
      NUMSTPT3=INT(EFPD-(NUMSTPT1*100)-
     c (NUMSTPT2*10))
      CHSTPT3=CHAR(NUMSTPT3+48)
      MATFILE(8:8)=CHSTPT1
      MATFILE(9:9)=CHSTPT2
      MATFILE(10:10)=CHSTPT3
      MATFILE(11:14)='.mat'
      OPEN(UNIT=200, FILE=MATFILE, STATUS='UNKNOWN')
      REWIND(UNIT=200)
*
      WRITE(200,*)
      WRITE(200,5)
    5 FORMAT(T1,'C   MATERIAL SPECIFICATIONS')
      WRITE(200,7)
    7 FORMAT(T1,'C')
      WRITE(30,10)
   10 FORMAT(T1,'C   GEOMETRY SPECIFICATIONS')
      WRITE(30,20)
   20 FORMAT(T1,'C')
*     Initiate line humber at 1.
      LN=1
*     Initiate material number at 1.
      MN=1000
*     Initiate surface number at 1.
      SN=1
*     Write the pressure vessel specification.
      WRITE(30,30)
   30 FORMAT(T1,'C   PRESSURE VESSEL')
      IF (EIGHTH.EQ..TRUE.) THEN
         PVIRSURF=SN ! Pressure vessel inner radius surface label
         SURFTYPESPEC(SN)='CZ'
         SURFVALUESPEC(SN)=PVCLADIR+PVCLADTHICKNESS
         SN=SN+1
         PVORSURF=SN ! Pressure vessel outer radius surface label
         SURFTYPESPEC(SN)='CZ'
         SURFVALUESPEC(SN)=PVCLADIR+PVCLADTHICKNESS+
     c   PVTHICKNESS
```

```
          PVOUTERSURF=SN
          SN=SN+1
          SYSTEMSOUTH=SN ! System's southern surface label
          SURFTYPESPEC(SN)='PY'
          SURFVALUESPEC(SN)=0.0
          SN=SN+1
          SYSTEMWEST=SN ! System's western surface label
          SURFTYPESPEC(SN)='P '
          SURFVALUESPEC(SN)=0
          SN=SN+1
          SYSTEMTOP=SN ! System's top surface label
          SURFTYPESPEC(SN)='PZ'
          TEMP=0
          DO 40 REGION=1,NUMREGABOVE
             TEMP=TEMP+REGABOVE(REGION,1)
   40     CONTINUE
          TEMP=TEMP+SPACERDIST(1,1)
          TEMP=TEMP+ENDFITHEIGHT(1,1)
          TEMP=TEMP+ENDFITHEIGHT(1,2)
          SURFVALUESPEC(SN)=TEMP
          SN=SN+1
          SYSTEMBOTTOM=SN ! System's bottom surface label
          SURFTYPESPEC(SN)='PZ'
          TEMP=0
          DO 50 REGION=1,NUMOFLOWREG
             TEMP=TEMP-LOWERREGION(REGION,1)
   50     CONTINUE
          SURFVALUESPEC(SN)=TEMP
          SN=SN+1
          PVML=MN
*  Write Pressure Vessel Material Spec
             DO 53 C=1,PVISONUM
                IF (C.EQ.1) THEN
                   WRITE(200,51) PVML, PVZAIDS(C), (-1*PVWTS(C))
   51         FORMAT(T1,'M',I4,T9,A9,3X,G14.6,'   $ Pressure Vessel')
                ELSE
                   WRITE(200,52) PVZAIDS(C), (-1*PVWTS(C))
   52              FORMAT(T9,A9,3X,G14.6)
                ENDIF
   53        CONTINUE
          MN=MN+1
          WRITE(30,60) LN, PVML, (-1*PVDENSITY), PVIRSURF,
     c    (-1*PVORSURF), SYSTEMSOUTH, SYSTEMWEST,
     c    (-1*SYSTEMTOP), SYSTEMBOTTOM
   60     FORMAT(T1,I4,T6,I4,T11,F7.3,T25,I4,1X,I4,
     c    1X,I4,1X,I4,1X,I4,1X,I4,' IMP:N=1')
          LN=LN+1
        ELSEIF (QUARTER.EQ..TRUE.) THEN
          PVIRSURF=SN ! Pressure vessel inner radius surface label
          SURFTYPESPEC(SN)='CZ'
          SURFVALUESPEC(SN)=PVCLADIR+PVCLADTHICKNESS
          SN=SN+1
          PVORSURF=SN ! Pressure vessel outer radius surface label
          SURFTYPESPEC(SN)='CZ'
```

```
         SURFVALUESPEC(SN)=PVCLADIR+PVCLADTHICKNESS+
     c   PVTHICKNESS
         SN=SN+1
         SYSTEMSOUTH=SN ! System's southern surface label
         SURFTYPESPEC(SN)='PY'
         SURFVALUESPEC(SN)=0.0
         SN=SN+1
         SYSTEMWEST=SN ! System's western surface label
         SURFTYPESPEC(SN)='PX'
         SURFVALUESPEC(SN)=0.0
         SN=SN+1
         SYSTEMTOP=SN ! System's top surface label
         SURFTYPESPEC(SN)='PZ'
         TEMP=0
         DO 70 REGION=1,NUMREGABOVE
            TEMP=TEMP+REGABOVE(REGION,1)
 70      CONTINUE
         TEMP=TEMP+SPACERDIST(1,1)
         TEMP=TEMP+ENDFITHEIGHT(1,1)
         TEMP=TEMP+ENDFITHEIGHT(1,2)
         SURFVALUESPEC(SN)=TEMP
         SN=SN+1
         SYSTEMBOTTOM=SN ! System's bottom surface label
         SURFTYPESPEC(SN)='PZ'
         TEMP=0
         DO 80 REGION=1,NUMOFLOWREG
            TEMP=TEMP-LOWERREGION(REGION,1)
 80      CONTINUE
         SURFVALUESPEC(SN)=TEMP
         SN=SN+1
         PVML=MN
*  Write Pressure Vessel Material Spec
            DO 83 C=1,PVISONUM
               IF (C.EQ.1) THEN
                  WRITE(200,81) PVML, PVZAIDS(C), (-1*PVWTS(C))
 81          FORMAT(T1,'M',I4,T9,A9,3X,G14.6,'  $ Pressure Vessel')
               ELSE
                  WRITE(200,82) PVZAIDS(C), (-1*PVWTS(C))
 82               FORMAT(T9,A9,3X,G14.6)
               ENDIF
 83         CONTINUE
         MN=MN+1
         WRITE(30,90) LN, PVML, (-1*PVDENSITY), PVIRSURF,
     c   (-1*PVORSURF), SYSTEMSOUTH, SYSTEMWEST,
     c   (-1*SYSTEMTOP), SYSTEMBOTTOM
 90      FORMAT(T1,I4,T6,I4,T11,F7.3,T25,I4,1X,I4,
     c   1X,I4,1X,I4,1X,I4,1X,I4,' IMP:N=1')
         LN=LN+1
      ELSEIF (FULL.EQ..TRUE.) THEN
         PVIRSURF=SN ! Pressure vessel inner radius surface label
         SURFTYPESPEC(SN)='CZ'
         SURFVALUESPEC(SN)=PVCLADIR+PVCLADTHICKNESS
         SN=SN+1
         PVORSURF=SN ! Pressure vessel outer radius surface label
```

```
            SURFTYPESPEC(SN)='CZ'
            SURFVALUESPEC(SN)=PVCLADIR+PVCLADTHICKNESS+
     c      PVTHICKNESS
            SN=SN+1
            SYSTEMTOP=SN ! System's top surface label
            SURFTYPESPEC(SN)='PZ'
            TEMP=0
            DO 100 REGION=1,NUMREGABOVE
               TEMP=TEMP+REGABOVE(REGION,1)
 100        CONTINUE
            TEMP=TEMP+SPACERDIST(1,1)
            TEMP=TEMP+ENDFITHEIGHT(1,1)
            TEMP=TEMP+ENDFITHEIGHT(1,2)
            SURFVALUESPEC(SN)=TEMP
            SN=SN+1
            SYSTEMBOTTOM=SN ! System's bottom surface label
            SURFTYPESPEC(SN)='PZ'
            TEMP=0
            DO 110 REGION=1,NUMOFLOWREG
               TEMP=TEMP-LOWERREGION(REGION,1)
 110        CONTINUE
            SURFVALUESPEC(SN)=TEMP
            SN=SN+1
            PVML=MN
*  Write Pressure Vessel Material Spec
               DO 113 C=1,PVISONUM
                  IF (C.EQ.1) THEN
                     WRITE(200,111) PVML, PVZAIDS(C), (-1*PVWTS(C))
 111              FORMAT(T1,'M',I4,T9,A9,3X,G14.6,'    $ Pressure Vessel')
                  ELSE
                     WRITE(200,112) PVZAIDS(C), (-1*PVWTS(C))
 112                 FORMAT(T9,A9,3X,G14.6)
                  ENDIF
 113           CONTINUE
            MN=MN+1
            WRITE(30,120) LN, PVML, (-1*PVDENSITY), PVIRSURF,
     c      (-1*PVORSURF), (-1*SYSTEMTOP), SYSTEMBOTTOM
 120        FORMAT(T1,I4,T6,I4,T11,F7.3,T25,I4,1X,I4,
     c      1X,I4,1X,I4,' IMP:N=1')
            LN=LN+1
         ENDIF
*        Write the pressure vessel cladding specification.
         WRITE(30,130)
 130     FORMAT(T1,'C    PRESSURE VESSEL CLADDING')
         IF ((EIGHTH.EQ..TRUE.).OR.(QUARTER.EQ..TRUE.)) THEN
            PVCIRSURF=SN ! Pressure vessel cladding inner radius surface label
            SURFTYPESPEC(SN)='CZ'
            SURFVALUESPEC(SN)=PVCLADIR
            SN=SN+1
            PVCML=MN
*  Check Pressure Vessel Cladding
               DO 133 C=1,PVCLADISONUM
                  IF (C.EQ.1) THEN
                     WRITE(200,131) PVCML, PVCLADZAIDS(C),
```

```
   c              (-1*PVCLADWTS(C))
 131              FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
   c              '    $ Pressure Vessel Clad')
               ELSE
                  WRITE(200,132) PVCLADZAIDS(C), (-1*PVCLADWTS(C))
 132              FORMAT(T9,A9,3X,G14.6)
               ENDIF
 133        CONTINUE
         MN=MN+1
         WRITE(30,140) LN, PVCML, (-1*PVCLADDENSITY), PVCIRSURF,
   c     (-1*PVIRSURF), SYSTEMSOUTH, SYSTEMWEST,
   c     (-1*SYSTEMTOP), SYSTEMBOTTOM
 140     FORMAT(T1,I4,T6,I4,T11,F7.3,T25,I4,1X,I4,
   c     1X,I4,1X,I4,1X,I4,1X,I4,' IMP:N=1')
         LN=LN+1
      ELSEIF (FULL.EQ..TRUE.) THEN
         PVCIRSURF=SN ! Pressure vessel cladding inner radius surface label
         SURFTYPESPEC(SN)='CZ'
         SURFVALUESPEC(SN)=PVCLADIR
         SN=SN+1
         PVCML=MN
*  Check Pressure Vessel Cladding
            DO 143 C=1,PVCLADISONUM
               IF (C.EQ.1) THEN
                  WRITE(200,141) PVCML, PVCLADZAIDS(C),
   c              (-1*PVCLADWTS(C))
 141              FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
   c              '    $ Pressure Vessel Clad')
               ELSE
                  WRITE(200,142) PVCLADZAIDS(C), (-1*PVCLADWTS(C))
 142              FORMAT(T9,A9,3X,G14.6)
               ENDIF
 143        CONTINUE
         MN=MN+1
         WRITE(30,150) LN, PVCML, (-1*PVCLADDENSITY), PVCIRSURF,
   c     (-1*PVIRSURF), (-1*SYSTEMTOP), SYSTEMBOTTOM
 150     FORMAT(T1,I4,T6,I4,T11,F7.3,T25,I4,1X,I4,
   c     1X,I4,1X,I4,' IMP:N=1')
         LN=LN+1
      ENDIF
*     Write the moderator region specification between the pressure vessel
and the thermal shield.
      IF (BANDW.EQ..TRUE.) THEN
      WRITE(30,152)
 152 FORMAT(T1,'C   BORATED MODERATOR BETWEEN THE ',
    c'PRESSURE VESSEL AND THERMAL SHIELD')
      ELSEIF (WESTINGHOUSE.EQ..TRUE.) THEN
      WRITE(30,154)
 154 FORMAT(T1,'C   BORATED MODERATOR BETWEEN THE ',
    c'PRESSURE VESSEL AND NEUTRON PAD REGION')
      ENDIF
      IF ((EIGHTH.EQ..TRUE.).OR.(QUARTER.EQ..TRUE.)) THEN
         TSORSURF=SN ! Thermal shield outer radius surface label
         SURFTYPESPEC(SN)='CZ'
```

```
        SURFVALUESPEC(SN)=SHIELDIR+SHIELDTHICKNESS
        SN=SN+1
        BMODML=MN
        BWTINH2O=((PPMB*1E-6)/(1.0+(PPMB*1E-6)))*100.0
        HWT=((1.008664904*0.999167*2.0)/
     c  ((1.008664904*0.999167*2.0)+
     c  (1.008664904*15.857510)))*(100.0-BWTINH2O)
        OWT=((1.008664904*15.857510)/
     c  ((1.008664904*0.999167*2.0)+
     c  (1.008664904*15.857510)))*(100.0-BWTINH2O)
        B10WT=((1.008664904*9.926922*0.194)/
     c  ((1.008664904*9.926922*0.194)+
     c  (1.008664904*10.914730*0.806)))*BWTINH2O
        B11WT=((1.008664904*10.914730*0.806)/
     c  ((1.008664904*9.926922*0.194)+
     c  (1.008664904*10.914730*0.806)))*BWTINH2O
* Check Borated Moderator
           DO 165 C=1,2
              IF (C.EQ.1) THEN
                 WRITE(200,161) BMODML,
     c           (-1*HWT)
 161             FORMAT(T1,'M',I4,T9,' 1001.50c',5X,G14.8,
     c           '      $ Borated Moderator')
              ELSE
                 WRITE(200,162)
     c           (-1*OWT)
 162             FORMAT(T9,' 8016.50c',5X,G14.8)
                 WRITE(200,163)
     c           (-1*B10WT)
 163             FORMAT(T9,' 5010.50c',5X,G14.8)
                 WRITE(200,164)
     c           (-1*B11WT)
 164             FORMAT(T9,' 5011.56c',5X,G14.8)
              ENDIF
 165       CONTINUE
        WRITE(200,166) BMODML
 166    FORMAT(T1,'MT',I4,T10,'LWTR.03T')
        MN=MN+1
        WRITE(30,170) LN, BMODML, (-1*MODDENSITY), TSORSURF,
     c  (-1*PVCIRSURF), SYSTEMSOUTH, SYSTEMWEST,
     c  (-1*SYSTEMTOP), SYSTEMBOTTOM
 170    FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,
     c  1X,I4,1X,I4,1X,I4,1X,I4,' IMP:N=1')
        LN=LN+1
      ELSEIF (FULL.EQ..TRUE.) THEN
        TSORSURF=SN ! Thermal shield outer radius surface label
        SURFTYPESPEC(SN)='CZ'
        SURFVALUESPEC(SN)=SHIELDIR+SHIELDTHICKNESS
        SN=SN+1
        BMODML=MN
        BWTINH2O=((PPMB*1E-6)/(1.0+(PPMB*1E-6)))*100.0
        HWT=((1.008664904*0.999167*2.0)/
     c  ((1.008664904*0.999167*2.0)+
     c  (1.008664904*15.857510)))*(100.0-BWTINH2O)
```

```
           OWT=((1.008664904*15.857510)/
     c     ((1.008664904*0.999167*2.0)+
     c     (1.008664904*15.857510)))*(100.0-BWTINH2O)
           B10WT=((1.008664904*9.926922*0.194)/
     c     ((1.008664904*9.926922*0.194)+
     c     (1.008664904*10.914730*0.806)))*BWTINH2O
           B11WT=((1.008664904*10.914730*0.806)/
     c     ((1.008664904*9.926922*0.194)+
     c     (1.008664904*10.914730*0.806)))*BWTINH2O
*   Check Borated Moderator
              DO 175 C=1,2
                 IF (C.EQ.1) THEN
                    WRITE(200,161) BMODML,
     c              (-1*HWT)
 171                FORMAT(T1,'M',I4,T9,' 1001.50c',5X,G14.8,
     c              '       $ Borated Moderator')
                 ELSE
                    WRITE(200,162)
     c              (-1*OWT)
 172                FORMAT(T9,' 8016.50c',5X,G14.8)
                    WRITE(200,163)
     c              (-1*B10WT)
 173                FORMAT(T9,' 5010.50c',5X,G14.8)
                    WRITE(200,164)
     c              (-1*B11WT)
 174                FORMAT(T9,' 5011.56c',5X,G14.8)
                 ENDIF
 175          CONTINUE
           WRITE(200,176) BMODML
 176       FORMAT(T1,'MT',I4,T10,'LWTR.03T')
           MN=MN+1
           WRITE(30,180) LN, BMODML, (-1*MODDENSITY), TSORSURF,
     c     (-1*PVCIRSURF), (-1*SYSTEMTOP), SYSTEMBOTTOM
 180       FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,
     c     1X,I4,1X,I4,' IMP:N=1')
           LN=LN+1
        ENDIF
*       Write the thermal shield specification.
        IF (BANDW.EQ..TRUE.) THEN
           WRITE(30,185)
 185       FORMAT(T1,'C   THERMAL SHIELD')
        ELSEIF (WESTINGHOUSE.EQ..TRUE.) THEN
           WRITE(30,190)
 190       FORMAT(T1,'C   NEUTRON PAD')
        ENDIF
        IF ((EIGHTH.EQ..TRUE.).OR.(QUARTER.EQ..TRUE.)) THEN
           TSIRSURF=SN ! Thermal shield or neutron pad inner radius surface
label
           SURFTYPESPEC(SN)='CZ'
           SURFVALUESPEC(SN)=SHIELDIR
           SN=SN+1
           TSML=MN
           IF (WESTINGHOUSE.EQ..TRUE.) THEN
              IF ((NPLOWDEG.GT.45).AND.(EIGHTH.EQ..TRUE.)) THEN
```

```
     c       1X,I4,1X,I4,1X,I4,1X,I4,' IMP:N=1')
             LN=LN+1
             IF ((EIGHTH.EQ..TRUE.).AND.(NPLOWDEG.GT.(45.0))) THEN
                WRITE(30,206) LN, BMODML, (-1*MODDENSITY), TSIRSURF,
     c          (-1*TSORSURF), (-1*NPSURF1), SYSTEMWEST,
     c          (-1*SYSTEMTOP), SYSTEMBOTTOM
 206            FORMAT(T1,I4,T6,I4,T11,F7.3,T25,I4,1X,I4,
     c          1X,I4,1X,I4,1X,I4,1X,I4,' IMP:N=1')
                LN=LN+1
             ENDIF
             IF ((QUARTER.EQ..TRUE.).AND.(NPLOWDEG.GT.(0.0))) THEN
                WRITE(30,207) LN, BMODML, (-1*MODDENSITY), TSIRSURF,
     c          (-1*TSORSURF), (-1*NPSURF1), SYSTEMWEST,
     c          (-1*SYSTEMTOP), SYSTEMBOTTOM
 207            FORMAT(T1,I4,T6,I4,T11,F7.3,T25,I4,1X,I4,
     c          1X,I4,1X,I4,1X,I4,1X,I4,' IMP:N=1')
                LN=LN+1
             ENDIF
             WRITE(30,208) LN, BMODML, (-1*MODDENSITY), TSIRSURF,
     c       (-1*TSORSURF), NPSURF2, SYSTEMSOUTH,
     c       (-1*SYSTEMTOP), SYSTEMBOTTOM
 208         FORMAT(T1,I4,T6,I4,T11,F7.3,T25,I4,1X,I4,
     c       1X,I4,1X,I4,1X,I4,1X,I4,' IMP:N=1')
             LN=LN+1
          ENDIF
       ELSEIF (FULL.EQ..TRUE.) THEN
          TSIRSURF=SN ! Thermal shield inner radius surface label
          SURFTYPESPEC(SN)='CZ'
          SURFVALUESPEC(SN)=SHIELDIR
          SN=SN+1
          TSML=MN
          IF (WESTINGHOUSE.EQ..TRUE.) THEN
             NPSURF1=SN ! Neutron pad radial surface label
             SURFTYPESPEC(SN)='P '
             SURFVALUESPEC(SN)=8000
             SN=SN+1
             TSML=MN
             NPSURF2=SN ! Neutron pad radial surface label
             SURFTYPESPEC(SN)='P '
             SURFVALUESPEC(SN)=8010
             SN=SN+1
             TSML=MN
             NPSURF3=SN ! Neutron pad radial surface label
             SURFTYPESPEC(SN)='P '
             SURFVALUESPEC(SN)=8020
             SN=SN+1
             TSML=MN
             NPSURF4=SN ! Neutron pad radial surface label
             SURFTYPESPEC(SN)='P '
             SURFVALUESPEC(SN)=8030
             SN=SN+1
             TSML=MN
          ENDIF
*   Check Thermal Shield
```

```
        IF (BANDW.EQ..TRUE.) THEN
           DO 214 C=1,SHIELDISONUM
              IF (C.EQ.1) THEN
                 WRITE(200,210) TSML, SHIELDZAIDS(C),
     c           (-1*SHIELDWTS(C))
 210             FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c           '    $ Thermal Shield')
              ELSE
                 WRITE(200,212) SHIELDZAIDS(C), (-1*SHIELDWTS(C))
 212             FORMAT(T9,A9,3X,G14.6)
              ENDIF
 214       CONTINUE
           MN=MN+1
           WRITE(30,216) LN, TSML, (-1*SHIELDDENSITY), TSIRSURF,
     c     (-1*TSORSURF), (-1*SYSTEMTOP), SYSTEMBOTTOM
 216       FORMAT(T1,I4,T6,I4,T11,F7.3,T25,I4,1X,I4,
     c     1X,I4,1X,I4,' IMP:N=1')
           LN=LN+1
        ELSEIF (WESTINGHOUSE.EQ..TRUE.) THEN
           DO 222 C=1,SHIELDISONUM
              IF (C.EQ.1) THEN
                 WRITE(200,218) TSML, SHIELDZAIDS(C),
     c           (-1*SHIELDWTS(C))
 218             FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c           '    $ Neutron Pad')
              ELSE
                 WRITE(200,220) SHIELDZAIDS(C), (-1*SHIELDWTS(C))
 220             FORMAT(T9,A9,3X,G14.6)
              ENDIF
 222       CONTINUE
           MN=MN+1
           WRITE(30,224) LN, TSML, (-1*SHIELDDENSITY), TSIRSURF,
     c     (-1*TSORSURF), (-1*SYSTEMTOP), SYSTEMBOTTOM,
     c     NPSURF1, (-1*NPSURF2)
 224       FORMAT(T1,I4,T6,I4,T11,F7.3,T25,I4,1X,I4,
     c     1X,I4,1X,I4,1X,I4,1X,I4,' IMP:N=1')
           LN=LN+1
           WRITE(30,226) LN, TSML, (-1*SHIELDDENSITY), TSIRSURF,
     c     (-1*TSORSURF), (-1*SYSTEMTOP), SYSTEMBOTTOM,
     c     NPSURF4, (-1*NPSURF3)
 226       FORMAT(T1,I4,T6,I4,T11,F7.3,T25,I4,1X,I4,
     c     1X,I4,1X,I4,1X,I4,1X,I4,' IMP:N=1')
           LN=LN+1
           WRITE(30,228) LN, TSML, (-1*SHIELDDENSITY), TSIRSURF,
     c     (-1*TSORSURF), (-1*SYSTEMTOP), SYSTEMBOTTOM,
     c     NPSURF2, (-1*NPSURF1)
 228       FORMAT(T1,I4,T6,I4,T11,F7.3,T25,I4,1X,I4,
     c     1X,I4,1X,I4,1X,I4,1X,I4,' IMP:N=1')
           LN=LN+1
           WRITE(30,230) LN, TSML, (-1*SHIELDDENSITY), TSIRSURF,
     c     (-1*TSORSURF), (-1*SYSTEMTOP), SYSTEMBOTTOM,
     c     NPSURF3, (-1*NPSURF4)
 230       FORMAT(T1,I4,T6,I4,T11,F7.3,T25,I4,1X,I4,
     c     1X,I4,1X,I4,1X,I4,1X,I4,' IMP:N=1')
```

```
            LN=LN+1
            WRITE(30,232) LN, BMODML, (-1*MODDENSITY), TSIRSURF,
     c      (-1*TSORSURF), (-1*NPSURF1), NPSURF3,
     c      (-1*SYSTEMTOP), SYSTEMBOTTOM
 232        FORMAT(T1,I4,T6,I4,T11,F7.3,T25,I4,1X,I4,
     c      1X,I4,1X,I4,1X,I4,1X,I4,' IMP:N=1')
            LN=LN+1
            WRITE(30,234) LN, BMODML, (-1*MODDENSITY), TSIRSURF,
     c      (-1*TSORSURF), NPSURF2, NPSURF4,
     c      (-1*SYSTEMTOP), SYSTEMBOTTOM
 234        FORMAT(T1,I4,T6,I4,T11,F7.3,T25,I4,1X,I4,
     c      1X,I4,1X,I4,1X,I4,1X,I4,' IMP:N=1')
            LN=LN+1
            WRITE(30,236) LN, BMODML, (-1*MODDENSITY), TSIRSURF,
     c      (-1*TSORSURF), (-1*NPSURF3), NPSURF1,
     c      (-1*SYSTEMTOP), SYSTEMBOTTOM
 236        FORMAT(T1,I4,T6,I4,T11,F7.3,T25,I4,1X,I4,
     c      1X,I4,1X,I4,1X,I4,1X,I4,' IMP:N=1')
            LN=LN+1
            WRITE(30,238) LN, BMODML, (-1*MODDENSITY), TSIRSURF,
     c      (-1*TSORSURF), (-1*NPSURF4), (-1*NPSURF2),
     c      (-1*SYSTEMTOP), SYSTEMBOTTOM
 238        FORMAT(T1,I4,T6,I4,T11,F7.3,T25,I4,1X,I4,
     c      1X,I4,1X,I4,1X,I4,1X,I4,' IMP:N=1')
            LN=LN+1
         ENDIF
      ENDIF
*     Write the moderator region specification between the thermal shield or
neutron pad and the core barrel.
      IF (BANDW.EQ..TRUE.) THEN
         WRITE(30,240)
 240     FORMAT(T1,'C     BORATED MODERATOR BETWEEN THE ',
     c   'THERMAL SHIELD AND CORE BARREL')
      ELSEIF (WESTINGHOUSE.EQ..TRUE.) THEN
         WRITE(30,242)
 242     FORMAT(T1,'C     BORATED MODERATOR BETWEEN THE ',
     c   'NEUTRON PAD AND CORE BARREL')
      ENDIF
      IF ((EIGHTH.EQ..TRUE.).OR.(QUARTER.EQ..TRUE.)) THEN
         BORSURF=SN ! Core barrel outer radius surface label
         SURFTYPESPEC(SN)='CZ'
         SURFVALUESPEC(SN)=BARRELIR+BARRELTHICKNESS
         SN=SN+1
         WRITE(30,244) LN, BMODML, (-1*MODDENSITY), BORSURF,
     c   (-1*TSIRSURF), SYSTEMSOUTH, SYSTEMWEST,
     c   (-1*SYSTEMTOP), SYSTEMBOTTOM
 244     FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,
     c   1X,I4,1X,I4,1X,I4,1X,I4,' IMP:N=1')
         LN=LN+1
      ELSEIF (FULL.EQ..TRUE.) THEN
         BORSURF=SN ! Core barrel outer radius surface label
         SURFTYPESPEC(SN)='CZ'
         SURFVALUESPEC(SN)=BARRELIR+BARRELTHICKNESS
         SN=SN+1
```

```
          WRITE(30,246) LN, BMODML, (-1*MODDENSITY), BORSURF,
     c    (-1*TSIRSURF), (-1*SYSTEMTOP), SYSTEMBOTTOM
 246      FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,
     c    1X,I4,1X,I4,' IMP:N=1')
          LN=LN+1
        ENDIF
*       Write the core barrel specification.
        WRITE(30,248)
 248    FORMAT(T1,'C    CORE BARREL')
        IF ((EIGHTH.EQ..TRUE.).OR.(QUARTER.EQ..TRUE.)) THEN
          BIRSURF=SN ! Core barrel inner radius surface label
          SURFTYPESPEC(SN)='CZ'
          SURFVALUESPEC(SN)=BARRELIR
          SN=SN+1
          BARRELML=MN
*   Check Core Barrel
            DO 253 C=1,BARRELISONUM
              IF (C.EQ.1) THEN
                WRITE(200,251) BARRELML, BARRELZAIDS(C),
     c          (-1*BARRELWTS(C))
 251            FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c          '     $ Core Barrel')
              ELSE
                WRITE(200,252) BARRELZAIDS(C), (-1*BARRELWTS(C))
 252            FORMAT(T9,A9,3X,G14.6)
              ENDIF
 253        CONTINUE
          MN=MN+1
          WRITE(30,260) LN, BARRELML, (-1*BARRELDENSITY), BIRSURF,
     c    (-1*BORSURF), SYSTEMSOUTH, SYSTEMWEST,
     c    (-1*SYSTEMTOP), SYSTEMBOTTOM
 260      FORMAT(T1,I4,T6,I4,T11,F7.3,T25,I4,1X,I4,
     c    1X,I4,1X,I4,1X,I4,1X,I4,' IMP:N=1')
          LN=LN+1
        ELSEIF (FULL.EQ..TRUE.) THEN
          BIRSURF=SN ! Core barrel inner radius surface label
          SURFTYPESPEC(SN)='CZ'
          SURFVALUESPEC(SN)=BARRELIR
          SN=SN+1
          BARRELML=MN
*   Check Core Barrel
            DO 263 C=1,BARRELISONUM
              IF (C.EQ.1) THEN
                WRITE(200,261) BARRELML, BARRELZAIDS(C),
     c          (-1*BARRELWTS(C))
 261            FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c          '     $ Core Barrel')
              ELSE
                WRITE(200,262) BARRELZAIDS(C), (-1*BARRELWTS(C))
 262            FORMAT(T9,A9,3X,G14.6)
              ENDIF
 263        CONTINUE
          MN=MN+1
          WRITE(30,270) LN, BARRELML, (-1*BARRELDENSITY), BIRSURF,
```

```
     c    (-1*BORSURF), (-1*SYSTEMTOP), SYSTEMBOTTOM
 270      FORMAT(T1,I4,T6,I4,T11,F7.3,T25,I4,1X,I4,
     c    1X,I4,1X,I4,' IMP:N=1')
          LN=LN+1
        ENDIF
*       Write the core lattice window specification.
        WRITE(30,280)
 280    FORMAT(T1,'C   CORE LATTICE WINDOW')
        UN=10
        IF ((EIGHTH.EQ..TRUE.).OR.(QUARTER.EQ..TRUE.)) THEN
          FCOREBOTSURF=SN ! Bottom surface of core lattice window (top of
lower pad)
          SURFTYPESPEC(SN)='PZ'
          SURFVALUESPEC(SN)=0.0
          SN=SN+1
          WRITE(30,290) LN, BMODML, (-1*MODDENSITY),
     c    (-1*BIRSURF), FCOREBOTSURF, (-1*SYSTEMTOP),
     c    SYSTEMSOUTH, SYSTEMWEST, UN
 290      FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,
     c    1X,I4,1X,I4,1X,I4,' IMP:N=1 FILL=',I2)
          LN=LN+1
          FCOREUNIV=UN
          UN=UN+10
        ELSEIF (FULL.EQ..TRUE.) THEN
          FCOREBOTSURF=SN ! Bottom surface of core lattice window (top of
lower pad)
          SURFTYPESPEC(SN)='PZ'
          SURFVALUESPEC(SN)=0.0
          SN=SN+1
          IF (BANDW.EQ..TRUE.) THEN
            WRITE(30,300) LN, BMODML, (-1*MODDENSITY),
     c      (-1*BIRSURF), FCOREBOTSURF, (-1*SYSTEMTOP),
     c      UN
 300        FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,
     c      1X,I4,' IMP:N=1 FILL=',I2,
     c      ' (-174.48784 -174.48784 0)')
            LN=LN+1
          ELSEIF (WESTINGHOUSE.EQ..TRUE.) THEN
            WRITE(30,302) LN, BMODML, (-1*MODDENSITY),
     c      (-1*BIRSURF), FCOREBOTSURF, (-1*SYSTEMTOP),
     c      UN
 302        FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,
     c      1X,I4,' IMP:N=1 FILL=',I2,
     c      ' (-172.02912 -172.02912 0)')
            LN=LN+1
          ENDIF
          FCOREUNIV=UN
          UN=UN+10
        ENDIF
*       Write the specifications for the regions below the core lattice window.
        WRITE(30,310)
 310    FORMAT(T1,'C   REGIONS BELOW THE CORE LATTICE WINDOW')
        TEMP=0
        DO 360 REGION=1,NUMOFLOWREG
```

```
         IF ((EIGHTH.EQ..TRUE.).OR.(QUARTER.EQ..TRUE.)) THEN
            IF (REGION.NE.NUMOFLOWREG) THEN
               LOWREGSURF(REGION)=SN ! Bottom surface of lower REGION
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=TEMP-LOWERREGION(REGION,1)
               TEMP=TEMP-LOWERREGION(REGION,1)
               SN=SN+1
            ELSE
               LOWREGSURF(REGION)=SYSTEMBOTTOM
            ENDIF
            LOWREGML(REGION)=MN
*  Check Lower Core Regions
               DO 313 D=1,LOWERREGION(REGION,3)
                  IF (D.EQ.1) THEN
                     WRITE(200,311) LOWREGML(REGION),
     c               LOWERREGIONZAIDS(REGION,D),
     c               (-1*LOWERREGIONWTS(REGION,D)), REGION
 311                 FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c               '    $ Lower Core Region ',I2)
                  ELSE
                     WRITE(200,312) LOWERREGIONZAIDS(REGION,D),
     c               (-1*LOWERREGIONWTS(REGION,D))
 312                 FORMAT(T9,A9,3X,G14.6)
                  ENDIF
 313           CONTINUE
               WRITE(200,314) LOWREGML(REGION)
 314           FORMAT(T1,'MT',I4,T9,'LWTR.03T')
            MN=MN+1
            IF (REGION.EQ.1) THEN
               WRITE(30,320) LN, LOWREGML(REGION),
     c         (-1*LOWERREGION(REGION,2)),
     c         (-1*BIRSURF), (-1*FCOREBOTSURF), LOWREGSURF(REGION),
     c         SYSTEMSOUTH, SYSTEMWEST, REGION
 320           FORMAT(T1,I4,T6,I4,T11,F7.3,T25,I4,1X,I4,
     c         1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1    $ Lower core region ',I2)
               LN=LN+1
            ELSE
               WRITE(30,330) LN, LOWREGML(REGION),
     c         (-1*LOWERREGION(REGION,2)),
     c         (-1*BIRSURF), (-1*LOWREGSURF(REGION-1)),
     c         LOWREGSURF(REGION), SYSTEMSOUTH, SYSTEMWEST, REGION
 330           FORMAT(T1,I4,T6,I4,T11,F7.3,T25,I4,1X,I4,
     c         1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1    $ Lower core region ',I2)
               LN=LN+1
            ENDIF
         ELSEIF (FULL.EQ..TRUE.) THEN
            IF (REGION.NE.NUMOFLOWREG) THEN
               LOWREGSURF(REGION)=SN ! Bottom surface of lower REGION
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=TEMP-LOWERREGION(REGION,1)
               TEMP=TEMP-LOWERREGION(REGION,1)
               SN=SN+1
```

```
            ELSE
               LOWREGSURF(REGION)=SYSTEMBOTTOM
            ENDIF
            LOWREGML(REGION)=MN
*  Check Lower Core Regions
               DO 333 D=1,LOWERREGION(REGION,3)
                  IF (D.EQ.1) THEN
                     WRITE(200,331) LOWREGML(REGION),
     c               LOWERREGIONZAIDS(REGION,D),
     c               (-1*LOWERREGIONWTS(REGION,D)), REGION
 331                 FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c               '    $ Lower Core Region ',I2)
                  ELSE
                     WRITE(200,332) LOWERREGIONZAIDS(REGION,D),
     c               (-1*LOWERREGIONWTS(REGION,D))
 332                 FORMAT(T9,A9,3X,G14.6)
                  ENDIF
 333           CONTINUE
               WRITE(200,334) LOWREGML(REGION)
 334           FORMAT(T1,'MT',I4,T9,'LWTR.03T')
            MN=MN+1
            IF (REGION.EQ.1) THEN
               WRITE(30,340) LN, LOWREGML(REGION),
     c         (-1*LOWERREGION(REGION,2)),
     c         (-1*BIRSURF), (-1*FCOREBOTSURF), LOWREGSURF(REGION),
     c         REGION
 340           FORMAT(T1,I4,T6,I4,T11,F7.3,T25,I4,1X,I4,
     c         1X,I4,' IMP:N=1   $ Lower core region ',I2)
               LN=LN+1
            ELSE
               WRITE(30,350) LN, LOWREGML(REGION),
     c         (-1*LOWERREGION(REGION,2)),
     c         (-1*BIRSURF), (-1*LOWREGSURF(REGION-1)),
     c         LOWREGSURF(REGION), REGION
 350           FORMAT(T1,I4,T6,I4,T11,F7.3,T25,I4,1X,I4,
     c         1X,I4,' IMP:N=1   $ Lower core region ',I2)
               LN=LN+1.
            ENDIF
         ENDIF
 360  CONTINUE
*     Write the zero importance outside world specification.
      WRITE(30,370)
 370  FORMAT(T1,'C   ZERO IMPORTANCE OUTSIDE WORLD')
      IF ((EIGHTH.EQ..TRUE.).OR.(QUARTER.EQ..TRUE.)) THEN
         WRITE(30,380) LN, PVORSURF,
     c   (-1*SYSTEMSOUTH), (-1*SYSTEMWEST),
     c   SYSTEMTOP, (-1*SYSTEMBOTTOM)
 380     FORMAT(T1,I4,'    0',T25,I4,':',I4,
     c   ':',I4,':',I4,':',I4,' IMP:N=0')
         LN=LN+1
      ELSEIF (FULL.EQ..TRUE.) THEN
         WRITE(30,390) LN, PVORSURF,
     c   SYSTEMTOP, (-1*SYSTEMBOTTOM)
 390     FORMAT(T1,I4,'    0',T25,I4,':',I4,
```

```
     c   ':',I4,' IMP:N=0')
           LN=LN+1
         ENDIF
*        Define and write the core lattice layout.
*        Step 1: Associate a universe identifier with each unique assembly
position.
         UN=100
         DO 392 COLUMN=1,50
            DO 391 ROW=1,50
               ASSYUNIV(COLUMN,ROW)=0
  391       CONTINUE
  392    CONTINUE
         COREWIDTHOLD=0
         DO 430 ROW=1,50
         COREWIDTH=0
            DO 420 COLUMN=1,50
               IF ((ASSYID(COLUMN,ROW).NE.'      ').AND.
     c         (ASSYUNIV(COLUMN,ROW).EQ.0)) THEN
                  IF (EIGHTH.EQ..TRUE.) THEN
                     ASSYUNIV(COLUMN,ROW)=UN
                     ASSYUNIV(ROW,COLUMN)=UN
                     UN=UN+1
                     COREWIDTH=COREWIDTH+1
                  ELSEIF (QUARTER.EQ..TRUE.) THEN
                     IF (ASSYID(COLUMN,ROW).EQ.
     c               ASSYID(ROW,COLUMN)) THEN
                        ASSYUNIV(COLUMN,ROW)=UN
                        ASSYUNIV(ROW,COLUMN)=UN
                        UN=UN+1
                        COREWIDTH=COREWIDTH+1
                     ELSEIF (ASSYID(COLUMN,ROW).NE.
     c               ASSYID(ROW,COLUMN)) THEN
                        ASSYUNIV(COLUMN,ROW)=UN
                        UN=UN+1
                        COREWIDTH=COREWIDTH+1
                     ENDIF
                  ELSEIF (FULL.EQ..TRUE.) THEN
                  IF ((ROW.NE.1).AND.(COLUMN.NE.1)) THEN
                     DO 394 RO=1,(ROW-1)
                        DO 393 CO=1,50
                           IF (ASSYID(COLUMN,ROW).EQ.ASSYID(CO,RO)) THEN
                              ASSYUNIV(COLUMN,ROW)=ASSYUNIV(CO,RO)
                              COREWIDTH=COREWIDTH+1
                              LEAVE=.TRUE.
                              EXIT
                           ENDIF
 393                    CONTINUE
                        IF (LEAVE.EQ..TRUE.) THEN
                           EXIT
                        ENDIF
 394                 CONTINUE
                     IF (LEAVE.EQ..FALSE.) THEN
                        DO 396 RO=ROW,ROW
                           DO 395 CO=1,(COLUMN-1)
```

```
                        IF (ASSYID(COLUMN,ROW).EQ.ASSYID(CO,RO)) THEN
                           ASSYUNIV(COLUMN,ROW)=ASSYUNIV(CO,RO)
                           COREWIDTH=COREWIDTH+1
                           LEAVE=.TRUE.
                           EXIT
                        ENDIF
395                  CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
396               CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 398 RO=1,(ROW-1)
                  DO 397 CO=1,50
                     IF (ASSYID(COLUMN,ROW).EQ.ASSYID(CO,RO)) THEN
                        ASSYUNIV(COLUMN,ROW)=ASSYUNIV(CO,RO)
                        COREWIDTH=COREWIDTH+1
                        LEAVE=.TRUE.
                        EXIT
                     ENDIF
397               CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
398            CONTINUE
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
               DO 400 RO=1,1
                  DO 399 CO=1,(COLUMN-1)
                     IF (ASSYID(COLUMN,ROW).EQ.ASSYID(CO,RO)) THEN
                        ASSYUNIV(COLUMN,ROW)=ASSYUNIV(CO,RO)
                        COREWIDTH=COREWIDTH+1
                        LEAVE=.TRUE.
                        EXIT
                     ENDIF
399               CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
400            CONTINUE
            ENDIF
            IF (ASSYUNIV(COLUMN,ROW).EQ.0) THEN
               ASSYUNIV(COLUMN,ROW)=UN
               UN=UN+1
               COREWIDTH=COREWIDTH+1
            ENDIF
            LEAVE=.FALSE.
            ENDIF
         ENDIF
420   CONTINUE
      IF (COREWIDTH.GT.COREWIDTHOLD) THEN
         COREWIDTHOLD=COREWIDTH
      ENDIF
430 CONTINUE
```

```
*       Step 2: Associate a universe identifier with each core baffle segment.
        UN=60
        IF ((EIGHTH.EQ..TRUE.).OR.(QUARTER.EQ..TRUE.)) THEN
           BAFQ1P1UNIV=UN ! Horizontal baffle segment
           UN=UN+1
           BAFQ1P2UNIV=UN ! Vertical baffle segment
           UN=UN+1
           BAFQ1P3UNIV=UN ! L-shaped baffle segment
           UN=UN+1
           BAFQ1P4UNIV=UN ! Corner piece baffle segment
           UN=UN+1
        ELSEIF (FULL.EQ..TRUE.) THEN
           BAFQ1P1UNIV=UN ! Horizontal baffle segment
           UN=UN+1
           BAFQ1P2UNIV=UN ! Vertical baffle segment
           UN=UN+1
           BAFQ1P3UNIV=UN ! L-shaped baffle segment
           UN=UN+1
           BAFQ1P4UNIV=UN ! Corner piece baffle segment
           UN=UN+1
           BAFQ2P1UNIV=BAFQ1P1UNIV ! Horizontal baffle segment
           BAFQ2P2UNIV=UN ! Vertical baffle segment
           UN=UN+1
           BAFQ2P3UNIV=UN ! L-shaped baffle segment
           UN=UN+1
           BAFQ2P4UNIV=UN ! Corner piece baffle segment
           UN=UN+1
           BAFQ3P1UNIV=UN ! Horizontal baffle segment
           UN=UN+1
           BAFQ3P2UNIV=BAFQ2P2UNIV ! Vertical baffle segment
           BAFQ3P3UNIV=UN ! L-shaped baffle segment
           UN=UN+1
           BAFQ3P4UNIV=UN ! Corner piece baffle segment
           UN=UN+1
           BAFQ4P1UNIV=BAFQ3P1UNIV ! Horizontal baffle segment
           BAFQ4P2UNIV=BAFQ1P2UNIV ! Vertical baffle segment
           BAFQ4P3UNIV=UN ! L-shaped baffle segment
           UN=UN+1
           BAFQ4P4UNIV=UN ! Corner piece baffle segment
           UN=UN+1
        ENDIF
*       Step 3: Define core lattice unit cell boundary surfaces.
        CLUCNORTHSURF=SN ! Northern surface of core lattice unit cell
        SURFTYPESPEC(SN)='PY'
        SURFVALUESPEC(SN)=(ASSYPITCH/2.0)
        SN=SN+1
        CLUCSOUTHSURF=SN ! Southern surface of core lattice unit cell
        SURFTYPESPEC(SN)='PY'
        SURFVALUESPEC(SN)=(-1*(ASSYPITCH/2.0))
        SN=SN+1
        CLUCEASTSURF=SN ! Eastern surface of core lattice unit cell
        SURFTYPESPEC(SN)='PX'
        SURFVALUESPEC(SN)=(ASSYPITCH/2.0)
        SN=SN+1
```

```
      CLUCWESTSURF=SN ! Western surface of core lattice unit cell
      SURFTYPESPEC(SN)='PX'
      SURFVALUESPEC(SN)=(-1*(ASSYPITCH/2.0))
      SN=SN+1
*     Step 4: Write the core lattice layout.
      WRITE(30,438)
  438 FORMAT(T1,'C   CORE LATTICE LAYOUT SPECIFICATION')
      IF ((EIGHTH.EQ..TRUE.).OR.(QUARTER.EQ..TRUE.)) THEN
         LATWIDTH=COREWIDTHOLD+6
         WRITE(30,440) LN, BMODML, (-1*MODDENSITY), (-1*CLUCEASTSURF),
     c   CLUCWESTSURF, (-1*CLUCNORTHSURF), CLUCSOUTHSURF, FCOREUNIV
 440     FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,1X,I4,1X,I4,
     c   ' IMP:N=1 LAT=1 U=',I2)
         LN=LN+1
         WRITE(30,450) (LATWIDTH-1), (LATWIDTH-1)
 450     FORMAT(T11,'FILL 0:',I2,' 0:',I2,' 0:0')
         COUNT=0
         WRITE(30,455)
 455     FORMAT(T11,$)
         DO 530 ROW=1,LATWIDTH
            DO 520 COLUMN=1,LATWIDTH
               IF (ASSYUNIV(COLUMN,ROW).NE.0) THEN
                  WRITE(30,460) ASSYUNIV(COLUMN,ROW)
 460              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
               ELSEIF ((ASSYUNIV(COLUMN,ROW).EQ.0).AND.
     c            (ROW.EQ.1)) THEN
                  IF ((ASSYUNIV((COLUMN+1),ROW).EQ.0).AND.
     c               (ASSYUNIV((COLUMN-1),ROW).NE.0)) THEN
                     WRITE(30,461) BAFQ1P2UNIV
 461                 FORMAT(I3,1X,$)
                     COUNT=COUNT+1
                  ELSEIF ((ASSYUNIV((COLUMN+1),ROW).EQ.0).AND.
     c               (ASSYUNIV((COLUMN-1),ROW).EQ.0)) THEN
                     WRITE(30,462) FCOREUNIV
 462                 FORMAT(I3,1X,$)
                     COUNT=COUNT+1
                  ENDIF
               ELSEIF ((ASSYUNIV(COLUMN,ROW).EQ.0).AND.
     c            (COLUMN.EQ.1)) THEN
                  IF ((ASSYUNIV((COLUMN+1),ROW).EQ.0).AND.
     c               (ASSYUNIV(COLUMN,(ROW+1)).EQ.0).AND.
     c               (ASSYUNIV(COLUMN,(ROW-1)).NE.0)) THEN
                     WRITE(30,463) BAFQ1P1UNIV
 463                 FORMAT(I3,1X,$)
                     COUNT=COUNT+1
                  ELSEIF ((ASSYUNIV((COLUMN+1),ROW).EQ.0).AND.
     c               (ASSYUNIV(COLUMN,(ROW+1)).EQ.0).AND.
     c               (ASSYUNIV(COLUMN,(ROW-1)).EQ.0)) THEN
                     WRITE(30,464) FCOREUNIV
 464                 FORMAT(I3,1X,$)
                     COUNT=COUNT+1
                  ENDIF
               ELSEIF ((ASSYUNIV(COLUMN,ROW).EQ.0).AND.
```

```
     c          (COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                IF ((ASSYUNIV((COLUMN-1),(ROW+1)).EQ.0).AND.
     c              (ASSYUNIV(COLUMN,(ROW+1)).EQ.0).AND.
     c              (ASSYUNIV((COLUMN+1),(ROW+1)).EQ.0).AND.
     c              (ASSYUNIV((COLUMN-1),ROW).NE.0).AND.
     c              (ASSYUNIV((COLUMN+1),ROW).EQ.0).AND.
     c              (ASSYUNIV((COLUMN-1),(ROW-1)).NE.0).AND.
     c              (ASSYUNIV(COLUMN,(ROW-1)).NE.0).AND.
     c              (ASSYUNIV((COLUMN+1),(ROW-1)).EQ.0)) THEN
                  WRITE(30,465) BAFQ1P3UNIV
 465              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
                ELSEIF ((ASSYUNIV((COLUMN-1),(ROW+1)).EQ.0).AND.
     c              (ASSYUNIV(COLUMN,(ROW+1)).EQ.0).AND.
     c              (ASSYUNIV((COLUMN+1),(ROW+1)).EQ.0).AND.
     c              (ASSYUNIV((COLUMN-1),ROW).NE.0).AND.
     c              (ASSYUNIV((COLUMN+1),ROW).EQ.0).AND.
     c              (ASSYUNIV((COLUMN-1),(ROW-1)).NE.0).AND.
     c              (ASSYUNIV(COLUMN,(ROW-1)).NE.0).AND.
     c              (ASSYUNIV((COLUMN+1),(ROW-1)).NE.0)) THEN
                  WRITE(30,466) BAFQ1P3UNIV
 466              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
                ELSEIF ((ASSYUNIV((COLUMN-1),(ROW+1)).NE.0).AND.
     c              (ASSYUNIV(COLUMN,(ROW+1)).EQ.0).AND.
     c              (ASSYUNIV((COLUMN+1),(ROW+1)).EQ.0).AND.
     c              (ASSYUNIV((COLUMN-1),ROW).NE.0).AND.
     c              (ASSYUNIV((COLUMN+1),ROW).EQ.0).AND.
     c              (ASSYUNIV((COLUMN-1),(ROW-1)).NE.0).AND.
     c              (ASSYUNIV(COLUMN,(ROW-1)).NE.0).AND.
     c              (ASSYUNIV((COLUMN+1),(ROW-1)).EQ.0)) THEN
                  WRITE(30,467) BAFQ1P3UNIV
 467              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
                ELSEIF ((ASSYUNIV((COLUMN-1),(ROW+1)).NE.0).AND.
     c              (ASSYUNIV(COLUMN,(ROW+1)).EQ.0).AND.
     c              (ASSYUNIV((COLUMN+1),(ROW+1)).EQ.0).AND.
     c              (ASSYUNIV((COLUMN-1),ROW).NE.0).AND.
     c              (ASSYUNIV((COLUMN+1),ROW).EQ.0).AND.
     c              (ASSYUNIV((COLUMN-1),(ROW-1)).NE.0).AND.
     c              (ASSYUNIV(COLUMN,(ROW-1)).NE.0).AND.
     c              (ASSYUNIV((COLUMN+1),(ROW-1)).NE.0)) THEN
                  WRITE(30,468) BAFQ1P3UNIV
 468              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
                ELSEIF ((ASSYUNIV((COLUMN-1),(ROW+1)).EQ.0).AND.
     c              (ASSYUNIV(COLUMN,(ROW+1)).EQ.0).AND.
     c              (ASSYUNIV((COLUMN+1),(ROW+1)).EQ.0).AND.
     c              (ASSYUNIV((COLUMN-1),ROW).EQ.0).AND.
     c              (ASSYUNIV((COLUMN+1),ROW).EQ.0).AND.
     c              (ASSYUNIV((COLUMN-1),(ROW-1)).NE.0).AND.
     c              (ASSYUNIV(COLUMN,(ROW-1)).NE.0).AND.
     c              (ASSYUNIV((COLUMN+1),(ROW-1)).NE.0)) THEN
                  WRITE(30,469) BAFQ1P1UNIV
```

```
469         FORMAT(I3,1X,$)
            COUNT=COUNT+1
         ELSEIF ((ASSYUNIV((COLUMN-1),(ROW+1)).EQ.0).AND.
     c        (ASSYUNIV(COLUMN,(ROW+1)).EQ.0).AND.
     c        (ASSYUNIV((COLUMN+1),(ROW+1)).EQ.0).AND.
     c        (ASSYUNIV((COLUMN-1),ROW).EQ.0).AND.
     c        (ASSYUNIV((COLUMN+1),ROW).EQ.0).AND.
     c        (ASSYUNIV((COLUMN-1),(ROW-1)).NE.0).AND.
     c        (ASSYUNIV(COLUMN,(ROW-1)).NE.0).AND.
     c        (ASSYUNIV((COLUMN+1),(ROW-1)).EQ.0)) THEN
            WRITE(30,470) BAFQ1P1UNIV
470         FORMAT(I3,1X,$)
            COUNT=COUNT+1
         ELSEIF ((ASSYUNIV((COLUMN-1),(ROW+1)).EQ.0).AND.
     c        (ASSYUNIV(COLUMN,(ROW+1)).EQ.0).AND.
     c        (ASSYUNIV((COLUMN+1),(ROW+1)).EQ.0).AND.
     c        (ASSYUNIV((COLUMN-1),ROW).EQ.0).AND.
     c        (ASSYUNIV((COLUMN+1),ROW).EQ.0).AND.
     c        (ASSYUNIV((COLUMN-1),(ROW-1)).EQ.0).AND.
     c        (ASSYUNIV(COLUMN,(ROW-1)).NE.0).AND.
     c        (ASSYUNIV((COLUMN+1),(ROW-1)).NE.0)) THEN
            WRITE(30,471) BAFQ1P1UNIV
471         FORMAT(I3,1X,$)
            COUNT=COUNT+1
         ELSEIF ((ASSYUNIV((COLUMN-1),(ROW+1)).EQ.0).AND.
     c        (ASSYUNIV(COLUMN,(ROW+1)).EQ.0).AND.
     c        (ASSYUNIV((COLUMN+1),(ROW+1)).EQ.0).AND.
     c        (ASSYUNIV((COLUMN-1),ROW).NE.0).AND.
     c        (ASSYUNIV((COLUMN+1),ROW).EQ.0).AND.
     c        (ASSYUNIV((COLUMN-1),(ROW-1)).NE.0).AND.
     c        (ASSYUNIV(COLUMN,(ROW-1)).EQ.0).AND.
     c        (ASSYUNIV((COLUMN+1),(ROW-1)).EQ.0)) THEN
            WRITE(30,472) BAFQ1P2UNIV
472         FORMAT(I3,1X,$)
            COUNT=COUNT+1
         ELSEIF ((ASSYUNIV((COLUMN-1),(ROW+1)).NE.0).AND.
     c        (ASSYUNIV(COLUMN,(ROW+1)).EQ.0).AND.
     c        (ASSYUNIV((COLUMN+1),(ROW+1)).EQ.0).AND.
     c        (ASSYUNIV((COLUMN-1),ROW).NE.0).AND.
     c        (ASSYUNIV((COLUMN+1),ROW).EQ.0).AND.
     c        (ASSYUNIV((COLUMN-1),(ROW-1)).NE.0).AND.
     c        (ASSYUNIV(COLUMN,(ROW-1)).EQ.0).AND.
     c        (ASSYUNIV((COLUMN+1),(ROW-1)).EQ.0)) THEN
            WRITE(30,473) BAFQ1P2UNIV
473         FORMAT(I3,1X,$)
            COUNT=COUNT+1
         ELSEIF ((ASSYUNIV((COLUMN-1),(ROW+1)).NE.0).AND.
     c        (ASSYUNIV(COLUMN,(ROW+1)).EQ.0).AND.
     c        (ASSYUNIV((COLUMN+1),(ROW+1)).EQ.0).AND.
     c        (ASSYUNIV((COLUMN-1),ROW).NE.0).AND.
     c        (ASSYUNIV((COLUMN+1),ROW).EQ.0).AND.
     c        (ASSYUNIV((COLUMN-1),(ROW-1)).EQ.0).AND.
     c        (ASSYUNIV(COLUMN,(ROW-1)).EQ.0).AND.
     c        (ASSYUNIV((COLUMN+1),(ROW-1)).EQ.0)) THEN
```

```
                        WRITE(30,474) BAFQ1P2UNIV
 474                    FORMAT(I3,1X,$)
                        COUNT=COUNT+1
                     ELSEIF ((ASSYUNIV((COLUMN-1),(ROW+1)).EQ.0).AND.
     c                    (ASSYUNIV(COLUMN,(ROW+1)).EQ.0).AND.
     c                    (ASSYUNIV((COLUMN+1),(ROW+1)).EQ.0).AND.
     c                    (ASSYUNIV((COLUMN-1),ROW).EQ.0).AND.
     c                    (ASSYUNIV((COLUMN+1),ROW).EQ.0).AND.
     c                    (ASSYUNIV((COLUMN-1),(ROW-1)).NE.0).AND.
     c                    (ASSYUNIV(COLUMN,(ROW-1)).EQ.0).AND.
     c                    (ASSYUNIV((COLUMN+1),(ROW-1)).EQ.0)) THEN
                        WRITE(30,475) BAFQ1P4UNIV
 475                    FORMAT(I3,1X,$)
                        COUNT=COUNT+1
                     ELSEIF ((ASSYUNIV((COLUMN-1),(ROW+1)).EQ.0).AND.
     c                    (ASSYUNIV(COLUMN,(ROW+1)).EQ.0).AND.
     c                    (ASSYUNIV((COLUMN+1),(ROW+1)).EQ.0).AND.
     c                    (ASSYUNIV((COLUMN-1),ROW).EQ.0).AND.
     c                    (ASSYUNIV((COLUMN+1),ROW).EQ.0).AND.
     c                    (ASSYUNIV((COLUMN-1),(ROW-1)).EQ.0).AND.
     c                    (ASSYUNIV(COLUMN,(ROW-1)).EQ.0).AND.
     c                    (ASSYUNIV((COLUMN+1),(ROW-1)).EQ.0)) THEN
                        WRITE(30,476) FCOREUNIV
 476                    FORMAT(I3,1X,$)
                        COUNT=COUNT+1
                     ENDIF
                  ENDIF
                  IF ((COLUMN.EQ.LATWIDTH)
     c               .AND.(ROW.EQ.LATWIDTH)) THEN
                     WRITE(30,*)
                  ELSEIF ((COLUMN.EQ.LATWIDTH).AND.
     c               (ROW.NE.LATWIDTH)) THEN
                     WRITE(30,*)
                     WRITE(30,492)
 492                 FORMAT(T11,$)
                  ENDIF
 520           CONTINUE
 530        CONTINUE
         ELSEIF (FULL.EQ..TRUE.) THEN
            LATWIDTH=COREWIDTHOLD+8
            WRITE(30,440) LN, BMODML, (-1*MODDENSITY), (-1*CLUCEASTSURF),
     c      CLUCWESTSURF,(-1*CLUCNORTHSURF), CLUCSOUTHSURF,  FCOREUNIV
 540        FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 LAT=1 U=',I2)
            LN=LN+1
            WRITE(30,550) (LATWIDTH-4), (LATWIDTH-4)
 550        FORMAT(T11,'FILL -3:',I2,' -3:',I2,' 0:0')
            COUNT=0
            WRITE(30,555)
 555        FORMAT(T11,$)
            DO 710 COREROW=-3,(LATWIDTH-4)
               DO 700 CORECOLUMN=-3,(LATWIDTH-4)
                  IF ((COREROW.LE.-1).OR.(CORECOLUMN.LE.-1)) THEN
                     WRITE(30,560) FCOREUNIV
```

```
 560        FORMAT(I3,1X,$)
            COUNT=COUNT+1
         ELSEIF ((COREROW.EQ.(LATWIDTH-4)).OR.
     c   (CORECOLUMN.EQ.(LATWIDTH-4))) THEN
            WRITE(30,565) FCOREUNIV
 565        FORMAT(I3,1X,$)
            COUNT=COUNT+1
         ELSEIF ((COREROW.EQ.0).AND.(CORECOLUMN.EQ.0)) THEN
            WRITE(30,600) FCOREUNIV
 600        FORMAT(I3,1X,$)
            COUNT=COUNT+1
         ELSEIF ((COREROW.EQ.1).AND.(CORECOLUMN.EQ.1)) THEN
            WRITE(30,601) FCOREUNIV
 601        FORMAT(I3,1X,$)
            COUNT=COUNT+1
         ELSEIF ((COREROW.EQ.0).AND.(CORECOLUMN.EQ.1)) THEN
            WRITE(30,601) FCOREUNIV
            COUNT=COUNT+1
         ELSEIF ((COREROW.EQ.1).AND.(CORECOLUMN.EQ.0)) THEN
            WRITE(30,601) FCOREUNIV
            COUNT=COUNT+1
         ELSEIF ((COREROW.GT.1).AND.(CORECOLUMN.EQ.0)) THEN
            IF ((ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).NE.0)) THEN
               WRITE(30,602) BAFQ3P4UNIV
 602           FORMAT(I3,1X,$)
               COUNT=COUNT+1
        ELSEIF ((ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).NE.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).NE.0)) THEN
               WRITE(30,603) BAFQ2P2UNIV
 603           FORMAT(I3,1X,$)
               COUNT=COUNT+1
        ELSEIF ((ASSYUNIV((CORECOLUMN+1),(COREROW-1)).NE.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).NE.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).NE.0)) THEN
               WRITE(30,604) BAFQ2P2UNIV
 604           FORMAT(I3,1X,$)
               COUNT=COUNT+1
        ELSEIF ((ASSYUNIV((CORECOLUMN+1),(COREROW-1)).NE.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).NE.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
               WRITE(30,605) BAFQ2P2UNIV
 605           FORMAT(I3,1X,$)
               COUNT=COUNT+1
        ELSEIF ((ASSYUNIV((CORECOLUMN+1),(COREROW-1)).NE.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
               WRITE(30,606) BAFQ2P4UNIV
 606           FORMAT(I3,1X,$)
               COUNT=COUNT+1
        ELSEIF ((ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
```

```
     c             (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                  WRITE(30,607) FCOREUNIV
 607              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
                ENDIF
              ELSEIF ((COREROW.GT.1).AND.(CORECOLUMN.EQ.1).AND.
     c             (ASSYUNIV(CORECOLUMN,COREROW).EQ.0)) THEN
                IF ((ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c             (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c             (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
     c             (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c             (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).NE.0)) THEN
                  WRITE(30,608) BAFQ3P4UNIV
 608              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
                ELSEIF ((ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c             (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c             (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
     c             (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).NE.0).AND.
     c             (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).NE.0)) THEN
                  WRITE(30,609) BAFQ2P2UNIV
 609              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
                ELSEIF ((ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c             (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).NE.0).AND.
     c             (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).NE.0).AND.
     c             (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).NE.0).AND.
     c             (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).NE.0)) THEN
                  WRITE(30,610) BAFQ3P3UNIV
 610              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
                ELSEIF ((ASSYUNIV((CORECOLUMN+0),(COREROW-1)).NE.0).AND.
     c             (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c             (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).NE.0).AND.
     c             (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).NE.0).AND.
     c             (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).NE.0)) THEN
                  WRITE(30,611) BAFQ2P3UNIV
 611              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
                ELSEIF ((ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c             (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c             (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).NE.0).AND.
     c             (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).NE.0).AND.
     c             (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                  WRITE(30,612) BAFQ2P2UNIV
 612              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
                ELSEIF ((ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c             (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c             (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).NE.0).AND.
     c             (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c             (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                  WRITE(30,613) BAFQ2P4UNIV
 613              FORMAT(I3,1X,$)
```

```
                  COUNT=COUNT+1
              ELSEIF ((ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c                (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c                (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
     c                (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c                (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                  WRITE(30,614) FCOREUNIV
 614              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
                ENDIF
              ELSEIF ((COREROW.EQ.0).AND.(CORECOLUMN.GT.1)) THEN
                IF ((ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).NE.0)) THEN
                  WRITE(30,615) BAFQ3P4UNIV
 615              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
              ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).NE.0)) THEN
                  WRITE(30,616) BAFQ3P1UNIV
 616              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
              ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW+1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).NE.0)) THEN
                  WRITE(30,617) BAFQ3P1UNIV
 617              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
              ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW+1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                  WRITE(30,618) BAFQ3P1UNIV
 618              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
              ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW+1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                  WRITE(30,619) BAFQ4P4UNIV
 619              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
              ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                  WRITE(30,620) FCOREUNIV
 620              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
                ENDIF
              ELSEIF ((COREROW.EQ.1).AND.(CORECOLUMN.GT.1).AND.
     c              (ASSYUNIV(CORECOLUMN,COREROW).EQ.0)) THEN
                IF ((ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
```

```
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).NE.0)) THEN
                  WRITE(30,621) BAFQ3P4UNIV
 621              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
            ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).NE.0)) THEN
                  WRITE(30,622) BAFQ3P1UNIV
 622              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
            ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).NE.0)) THEN
                  WRITE(30,623) BAFQ3P3UNIV
 623              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
            ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW+0)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).NE.0)) THEN
                  WRITE(30,624) BAFQ4P3UNIV
 624              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
            ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                  WRITE(30,625) BAFQ4P1UNIV
 625              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
            ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                  WRITE(30,626) BAFQ4P4UNIV
 626              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
            ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                  WRITE(30,627) FCOREUNIV
 627              FORMAT(I3,1X,$)
                  COUNT=COUNT+1
               ENDIF
            ELSEIF ((COREROW.GE.1).AND.(CORECOLUMN.GE.1).AND.
```

```
     c          (ASSYUNIV(CORECOLUMN,COREROW).NE.0)) THEN
                   WRITE(30,570) ASSYUNIV(CORECOLUMN,COREROW)
 570               FORMAT(I3,1X,$)
                   COUNT=COUNT+1
                ELSEIF ((COREROW.GT.1).AND.(CORECOLUMN.GT.1).AND.
     c          (ASSYUNIV(CORECOLUMN,COREROW).EQ.0)) THEN
                   IF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).NE.0).AND.
     c                (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c                (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
     c                (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).NE.0).AND.
     c                (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c                (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c                (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c                (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                     WRITE(30,571) BAFQ1P2UNIV
 571                 FORMAT(I3,1X,$)
                     COUNT=COUNT+1
                  ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).NE.0).AND.
     c                (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).NE.0).AND.
     c                (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
     c                (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).NE.0).AND.
     c                (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c                (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c                (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c                (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                     WRITE(30,572) BAFQ1P3UNIV
 572                 FORMAT(I3,1X,$)
                     COUNT=COUNT+1
                  ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).NE.0).AND.
     c                (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).NE.0).AND.
     c                (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
     c                (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).NE.0).AND.
     c                (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c                (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).NE.0).AND.
     c                (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c                (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                     WRITE(30,573) BAFQ1P3UNIV
 573                 FORMAT(I3,1X,$)
                     COUNT=COUNT+1
                  ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).NE.0).AND.
     c                (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).NE.0).AND.
     c                (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).NE.0).AND.
     c                (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).NE.0).AND.
     c                (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c                (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c                (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c                (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                     WRITE(30,574) BAFQ1P3UNIV
 574                 FORMAT(I3,1X,$)
                     COUNT=COUNT+1
                  ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).NE.0).AND.
     c                (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c                (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
     c                (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
```

```
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                   WRITE(30,575) BAFQ1P4UNIV
 575               FORMAT(I3,1X,$)
                   COUNT=COUNT+1
           ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                   WRITE(30,576) BAFQ1P1UNIV
 576               FORMAT(I3,1X,$)
                   COUNT=COUNT+1
           ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                   WRITE(30,577) BAFQ1P1UNIV
 577               FORMAT(I3,1X,$)
                   COUNT=COUNT+1
           ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                   WRITE(30,578) BAFQ1P2UNIV
 578               FORMAT(I3,1X,$)
                   COUNT=COUNT+1
           ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                   WRITE(30,579) BAFQ4P2UNIV
 579               FORMAT(I3,1X,$)
                   COUNT=COUNT+1
           ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
```

```
     c          (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).NE.0).AND.
     c          (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                WRITE(30,580) BAFQ4P4UNIV
 580            FORMAT(I3,1X,$)
                COUNT=COUNT+1
        ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                WRITE(30,581) FCOREUNIV
 581            FORMAT(I3,1X,$)
                COUNT=COUNT+1
        ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).NE.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).NE.0).AND.
     c          (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).NE.0).AND.
     c          (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).NE.0)) THEN
                WRITE(30,582) BAFQ2P3UNIV
 582            FORMAT(I3,1X,$)
                COUNT=COUNT+1
        ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).NE.0).AND.
     c          (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).NE.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).NE.0).AND.
     c          (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).NE.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                WRITE(30,583) BAFQ4P3UNIV
 583            FORMAT(I3,1X,$)
                COUNT=COUNT+1
        ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).NE.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).NE.0).AND.
     c          (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).NE.0).AND.
     c          (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                WRITE(30,584) BAFQ4P3UNIV
 584            FORMAT(I3,1X,$)
                COUNT=COUNT+1
        ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).EQ.0).AND.
     c          (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
```

```
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                   WRITE(30,585) BAFQ4P1UNIV
 585               FORMAT(I3,1X,$)
                   COUNT=COUNT+1
            ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).NE.0)) THEN
                   WRITE(30,586) BAFQ4P3UNIV
 586               FORMAT(I3,1X,$)
                   COUNT=COUNT+1
            ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                   WRITE(30,587) BAFQ2P3UNIV
 587               FORMAT(I3,1X,$)
                   COUNT=COUNT+1
            ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).NE.0)) THEN
                   WRITE(30,588) BAFQ4P1UNIV
 588               FORMAT(I3,1X,$)
                   COUNT=COUNT+1
            ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c              (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).NE.0).AND.
     c              (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).NE.0)) THEN
                   WRITE(30,589) BAFQ3P1UNIV
 589               FORMAT(I3,1X,$)
                   COUNT=COUNT+1
            ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).EQ.0).AND.
```

```
     c               (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).NE.0)) THEN
                    WRITE(30,590) BAFQ3P4UNIV
 590                FORMAT(I3,1X,$)
                    COUNT=COUNT+1
             ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).NE.0).AND.
     c               (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).NE.0).AND.
     c               (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).NE.0).AND.
     c               (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).NE.0)) THEN
                    WRITE(30,591) BAFQ3P3UNIV
 591                FORMAT(I3,1X,$)
                    COUNT=COUNT+1
             ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).NE.0).AND.
     c               (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).NE.0).AND.
     c               (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).NE.0)) THEN
                    WRITE(30,592) BAFQ3P3UNIV
 592                FORMAT(I3,1X,$)
                    COUNT=COUNT+1
             ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).NE.0).AND.
     c               (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).NE.0).AND.
     c               (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
                    WRITE(30,593) BAFQ2P1UNIV
 593                FORMAT(I3,1X,$)
                    COUNT=COUNT+1
             ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).NE.0).AND.
     c               (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c               (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).NE.0)) THEN
                    WRITE(30,594) BAFQ3P2UNIV
 594                FORMAT(I3,1X,$)
                    COUNT=COUNT+1
```

```
      ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).EQ.0).AND.
     c        (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c        (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).NE.0).AND.
     c        (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c        (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).NE.0).AND.
     c        (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c        (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).NE.0).AND.
     c        (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).NE.0)) THEN
              WRITE(30,595) BAFQ3P3UNIV
595           FORMAT(I3,1X,$)
              COUNT=COUNT+1
      ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).EQ.0).AND.
     c        (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c        (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).NE.0).AND.
     c        (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c        (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).NE.0).AND.
     c        (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c        (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c        (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).NE.0)) THEN
              WRITE(30,596) BAFQ2P2UNIV
596           FORMAT(I3,1X,$)
              COUNT=COUNT+1
      ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).EQ.0).AND.
     c        (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c        (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).NE.0).AND.
     c        (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c        (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).NE.0).AND.
     c        (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c        (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c        (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
              WRITE(30,597) BAFQ2P2UNIV
597           FORMAT(I3,1X,$)
              COUNT=COUNT+1
      ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).EQ.0).AND.
     c        (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).EQ.0).AND.
     c        (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).NE.0).AND.
     c        (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c        (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).EQ.0).AND.
     c        (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c        (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c        (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
              WRITE(30,598) BAFQ2P4UNIV
598           FORMAT(I3,1X,$)
              COUNT=COUNT+1
      ELSEIF ((ASSYUNIV((CORECOLUMN-1),(COREROW-1)).NE.0).AND.
     c        (ASSYUNIV((CORECOLUMN+0),(COREROW-1)).NE.0).AND.
     c        (ASSYUNIV((CORECOLUMN+1),(COREROW-1)).NE.0).AND.
     c        (ASSYUNIV((CORECOLUMN-1),(COREROW+0)).EQ.0).AND.
     c        (ASSYUNIV((CORECOLUMN+1),(COREROW+0)).NE.0).AND.
     c        (ASSYUNIV((CORECOLUMN-1),(COREROW+1)).EQ.0).AND.
     c        (ASSYUNIV((CORECOLUMN+0),(COREROW+1)).EQ.0).AND.
     c        (ASSYUNIV((CORECOLUMN+1),(COREROW+1)).EQ.0)) THEN
              WRITE(30,599) BAFQ2P3UNIV
599           FORMAT(I3,1X,$)
```

```
                    COUNT=COUNT+1
                 ENDIF
              ENDIF
              IF ((MOD(COUNT,10).EQ.0).OR.
     c           (CORECOLUMN.EQ.(LATWIDTH-4))) THEN
                 WRITE(30,*)
                 IF (COREROW.LT.(LATWIDTH-4)) THEN
                    WRITE(30,697)
 697                FORMAT(T11,$)
                 ELSEIF ((COREROW.EQ.(LATWIDTH-4)).AND.
     c           (CORECOLUMN.LT.(LATWIDTH-4))) THEN
                    WRITE(30,698)
 698                FORMAT(T11,$)
                 ENDIF
              ENDIF
 700       CONTINUE
 710    CONTINUE
      ENDIF
*     Write the specifications for the universes which define the baffle
plate unit cells.
      WRITE(30,720)
 720  FORMAT(T1,'C   CORE BAFFLE UNIVERSE SPECIFICATIONS')
      IF ((EIGHTH.EQ..TRUE.).OR.(QUARTER.EQ..TRUE.)) THEN
         BAFACDNORTHSURF=SN ! Nothern surface of baffle sections A, C, & D.
         SURFTYPESPEC(SN)='PY'
         SURFVALUESPEC(SN)=((ASSYPITCH/2.0)-
     c   (BAFFLESEPARATION+BAFFLETHICKNESS))*(-1.0)
         SN=SN+1
         BAFACDSOUTHSURF=SN ! Southern surface of baffle sections A, C, & D.
         SURFTYPESPEC(SN)='PY'
         SURFVALUESPEC(SN)=((ASSYPITCH/2.0)-
     c   BAFFLESEPARATION)*(-1.0)
         SN=SN+1
         BAFBCDEASTSURF=SN ! Eastern surface of baffle sections B, C, & D.
         SURFTYPESPEC(SN)='PX'
         SURFVALUESPEC(SN)=((ASSYPITCH/2.0)-
     c   (BAFFLESEPARATION+BAFFLETHICKNESS))*(-1.0)
         SN=SN+1
         BAFBCDWESTSURF=SN ! Western surface of baffle sections B, C, & D.
         SURFTYPESPEC(SN)='PX'
         SURFVALUESPEC(SN)=((ASSYPITCH/2.0)-
     c   BAFFLESEPARATION)*(-1.0)
         SN=SN+1
         BAFFLEML=MN
*  Check Core Baffle
            DO 723 C=1,BAFFLEISONUM
               IF (C.EQ.1) THEN
                  WRITE(200,721) BAFFLEML, BAFFLEZAIDS(C),
     c            (-1*BAFFLEWTS(C))
 721              FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c            '   $ Core Baffle')
               ELSE
                  WRITE(200,722) BAFFLEZAIDS(C), (-1*BAFFLEWTS(C))
 722              FORMAT(T9,A9,3X,G14.6)
```

```
              ENDIF
 723        CONTINUE
         MN=MN+1
*     Write universe for baffle section A.
         WRITE(30,730) LN, BMODML, (-1*MODDENSITY), BAFACDNORTHSURF,
     c   BAFQ1P1UNIV
 730     FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,' IMP:N=1 U=',I2,
     c   '    $ Vertical baffle section water region')
         LN=LN+1
         WRITE(30,740) LN, BMODML, (-1*MODDENSITY),
     c   (-1*BAFACDSOUTHSURF), BAFQ1P1UNIV
 740     FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,' IMP:N=1 U=',I2,
     c   '    $ Vertical baffle section water region')
         LN=LN+1
         WRITE(30,750) LN, BAFFLEML, (-1*BAFFLEDENSITY),
     c   (-1*BAFACDNORTHSURF), BAFACDSOUTHSURF, BAFQ1P1UNIV
 750     FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,' IMP:N=1 U=',I2,
     c   '    $ Vertical baffle section baffle region')
         LN=LN+1
*     Write universe for baffle section B.
         WRITE(30,760) LN, BMODML, (-1*MODDENSITY), BAFBCDEASTSURF,
     c   BAFQ1P2UNIV
 760     FORMAT(T1,I4,T6,I4;T11,F7.4,T25,I4,' IMP:N=1 U=',I2,
     c   '    $ Horizontal baffle section water region')
         LN=LN+1
         WRITE(30,770) LN, BMODML, (-1*MODDENSITY), (-1*BAFBCDWESTSURF),
     c   BAFQ1P2UNIV
 770     FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,' IMP:N=1 U=',I2,
     c   '    $ Horizontal baffle section water region')
         LN=LN+1
         WRITE(30,780) LN, BAFFLEML, (-1*BAFFLEDENSITY),
     c   (-1*BAFBCDEASTSURF), BAFBCDWESTSURF, BAFQ1P2UNIV
 780     FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,' IMP:N=1 U=',I2,
     c   '    $ Horizontal baffle section baffle region')
         LN=LN+1
*     Write universe for baffle section C.
         WRITE(30,790) LN, BMODML, (-1*MODDENSITY), BAFBCDEASTSURF,
     c   BAFACDNORTHSURF, BAFQ1P3UNIV
 790     FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,' IMP:N=1 U=',I2,
     c   '    $ L-shaped baffle section water region')
         LN=LN+1
         WRITE(30,800) LN, BMODML, (-1*MODDENSITY), (-1*BAFBCDWESTSURF),
     c   (-1*BAFACDSOUTHSURF), BAFQ1P3UNIV
 800     FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,':',I4,' IMP:N=1 U=',I2,
     c   '    $ L-shaped baffle section water region')
         LN=LN+1
         WRITE(30,810) LN, BAFFLEML, (-1*BAFFLEDENSITY),
     c   (-1*BAFBCDEASTSURF), (-1*BAFACDNORTHSURF), BAFBCDWESTSURF,
     c   BAFACDSOUTHSURF, BAFQ1P3UNIV
 810     FORMAT(T1,I4,T6,I4,T11,F7.4,T25,'(',I4,':',I4,') (',I4,1X,I4,
     c   ') IMP:N=1 U=',I2,'    $ L-shaped baffle section baffle region')
         LN=LN+1
*     Write universe for baffle section D.
         WRITE(30,820) LN, BMODML, (-1*MODDENSITY), BAFBCDEASTSURF,
```

```
     c    BAFACDNORTHSURF, BAFQ1P4UNIV
   820    FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,':',I4,' IMP:N=1 U=',I2,
     c    '   $ Corner baffle section water region')
          LN=LN+1
          WRITE(30,830) LN, BMODML, (-1*MODDENSITY), (-1*BAFBCDWESTSURF),
     c    (-1*BAFACDSOUTHSURF), BAFQ1P4UNIV
   830    FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,' IMP:N=1 U=',I2,
     c    '   $ Corner baffle section water region')
          LN=LN+1
          WRITE(30,840) LN, BAFFLEML, (-1*BAFFLEDENSITY),
     c    (-1*BAFBCDEASTSURF), (-1*BAFACDNORTHSURF), BAFBCDWESTSURF,
     c    BAFACDSOUTHSURF, BAFQ1P4UNIV
   840    FORMAT(T1,I4,T6,I4,T11,F7.4,T25,'(',I4,1X,I4,') (',I4,':',I4,
     c    ') IMP:N=1 U=',I2,'    $ Corner baffle section baffle region')
          LN=LN+1
      ELSEIF (FULL.EQ..TRUE.) THEN
          BAFACDFGNORTHSURF=SN ! Nothern surface of baffle sections A, C, D,
F, & G.
          SURFTYPESPEC(SN)='PY'
          SURFVALUESPEC(SN)=(-1.0*(ASSYPITCH/2.0))+
     c    (BAFFLESEPARATION+BAFFLETHICKNESS)
          SN=SN+1
          BAFACDFGSOUTHSURF=SN ! Southern surface of baffle sections A, C, D,
F, & G.
          SURFTYPESPEC(SN)='PY'
          SURFVALUESPEC(SN)=(-1.0*(ASSYPITCH/2.0))+
     c    BAFFLESEPARATION
          SN=SN+1
          BAFBCDKLEASTSURF=SN ! Eastern surface of baffle sections B, C, D, K,
& L.
          SURFTYPESPEC(SN)='PX'
          SURFVALUESPEC(SN)=(-1.0*(ASSYPITCH/2.0))+
     c    (BAFFLESEPARATION+BAFFLETHICKNESS)
          SN=SN+1
          BAFBCDKLWESTSURF=SN ! Western surface of baffle sections B, C, D, K,
& L.
          SURFTYPESPEC(SN)='PX'
          SURFVALUESPEC(SN)=(-1.0*(ASSYPITCH/2.0))+
     c    BAFFLESEPARATION
          SN=SN+1
          BAFHIJKLNORTHSURF=SN ! Nothern surface of baffle sections H, I, J,
K, & L.
          SURFTYPESPEC(SN)='PY'
          SURFVALUESPEC(SN)=(ASSYPITCH/2.0)-
     c    BAFFLESEPARATION
          SN=SN+1
          BAFHIJKLSOUTHSURF=SN ! Southern surface of baffle sections H, I, J,
K, & L.
          SURFTYPESPEC(SN)='PY'
          SURFVALUESPEC(SN)=(ASSYPITCH/2.0)-
     c    (BAFFLESEPARATION+BAFFLETHICKNESS)
          SN=SN+1
          BAFEFGIJEASTSURF=SN ! Eastern surface of baffle sections E, F, G, I,
& J.
```

```
          SURFTYPESPEC(SN)='PX'
          SURFVALUESPEC(SN)=(ASSYPITCH/2.0)-
     c    BAFFLESEPARATION
          SN=SN+1
          BAFEFGIJWESTSURF=SN ! Western surface of baffle sections E, F, G, I,
& J.
          SURFTYPESPEC(SN)='PX'
          SURFVALUESPEC(SN)=(ASSYPITCH/2.0)-
     c    (BAFFLESEPARATION+BAFFLETHICKNESS)
          SN=SN+1
          BAFFLEML=MN
*  Check Core Baffle
             DO 843 C=1,BAFFLEISONUM
                IF (C.EQ.1) THEN
                   WRITE(200,841) BAFFLEML, BAFFLEZAIDS(C),
     c             (-1*BAFFLEWTS(C))
 841               FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c             '    $ Core Baffle')
                ELSE
                   WRITE(200,842) BAFFLEZAIDS(C), (-1*BAFFLEWTS(C))
 842               FORMAT(T9,A9,3X,G14.6)
                ENDIF
 843         CONTINUE
          MN=MN+1
*     Write universe for baffle section A.
          WRITE(30,850) LN, BMODML, (-1*MODDENSITY), BAFACDFGNORTHSURF,
     c    BAFQ1P1UNIV
 850      FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,' IMP:N=1 U=',I2)
          LN=LN+1
          WRITE(30,860) LN, BMODML, (-1*MODDENSITY),
     c    (-1*BAFACDFGSOUTHSURF), BAFQ1P1UNIV
 860      FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,' IMP:N=1 U=',I2)
          LN=LN+1
          WRITE(30,870) LN, BAFFLEML, (-1*BAFFLEDENSITY),
     c    (-1*BAFACDFGNORTHSURF), BAFACDFGSOUTHSURF, BAFQ1P1UNIV
 870      FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,' IMP:N=1 U=',I2)
          LN=LN+1
*     Write universe for baffle section B.
          WRITE(30,880) LN, BMODML, (-1*MODDENSITY), BAFBCDKLEASTSURF,
     c    BAFQ1P2UNIV
 880      FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,' IMP:N=1 U=',I2)
          LN=LN+1
          WRITE(30,890) LN, BMODML, (-1*MODDENSITY),
     c    (-1*BAFBCDKLWESTSURF), BAFQ1P2UNIV
 890      FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,' IMP:N=1 U=',I2)
          LN=LN+1
          WRITE(30,900) LN, BAFFLEML, (-1*BAFFLEDENSITY),
     c    (-1*BAFBCDKLEASTSURF), BAFBCDKLWESTSURF, BAFQ1P2UNIV
 900      FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,' IMP:N=1 U=',I2)
          LN=LN+1
*     Write universe for baffle section C.
          WRITE(30,910) LN, BMODML, (-1*MODDENSITY), BAFBCDKLEASTSURF,
     c    BAFACDFGNORTHSURF, BAFQ1P3UNIV
 910      FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,' IMP:N=1 U=',I2)
```

```
      LN=LN+1
      WRITE(30,920) LN, BMODML, (-1*MODDENSITY),
     c (-1*BAFBCDKLWESTSURF), (-1*BAFACDFGSOUTHSURF), BAFQ1P3UNIV
 920  FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,':',I4,' IMP:N=1 U=',I2)
      LN=LN+1
      WRITE(30,930) LN, BAFFLEML, (-1*BAFFLEDENSITY),
     c (-1*BAFBCDKLEASTSURF), (-1*BAFACDFGNORTHSURF),
     c BAFBCDKLWESTSURF, BAFACDFGSOUTHSURF, BAFQ1P3UNIV
 930  FORMAT(T1,I4,T6,I4,T11,F7.4,T25,'(',I4,':',I4,') (',I4,1X,I4,
     c ') IMP:N=1 U=',I2)
      LN=LN+1
*    Write universe for baffle section D.
      WRITE(30,940) LN, BMODML, (-1*MODDENSITY), BAFBCDKLEASTSURF,
     c BAFACDFGNORTHSURF, BAFQ1P4UNIV
 940  FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,':',I4,' IMP:N=1 U=',I2)
      LN=LN+1
      WRITE(30,950) LN, BMODML, (-1*MODDENSITY),
     c (-1*BAFBCDKLWESTSURF), (-1*BAFACDFGSOUTHSURF), BAFQ1P4UNIV
 950  FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,' IMP:N=1 U=',I2)
      LN=LN+1
      WRITE(30,960) LN, BAFFLEML, (-1*BAFFLEDENSITY),
     c (-1*BAFBCDKLEASTSURF), (-1*BAFACDFGNORTHSURF),
     c BAFBCDKLWESTSURF, BAFACDFGSOUTHSURF, BAFQ1P4UNIV
 960  FORMAT(T1,I4,T6,I4,T11,F7.4,T25,'(',I4,1X,I4,') (',I4,':',I4,
     c ') IMP:N=1 U=',I2)
      LN=LN+1
*    Write universe for baffle section E.
      WRITE(30,970) LN, BMODML, (-1*MODDENSITY), BAFEFGIJEASTSURF,
     c BAFQ2P2UNIV
 970  FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,' IMP:N=1 U=',I2)
      LN=LN+1
      WRITE(30,980) LN, BMODML, (-1*MODDENSITY),
     c (-1*BAFEFGIJWESTSURF), BAFQ2P2UNIV
 980  FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,' IMP:N=1 U=',I2)
      LN=LN+1
      WRITE(30,990) LN, BAFFLEML, (-1*BAFFLEDENSITY),
     c (-1*BAFEFGIJEASTSURF), BAFEFGIJWESTSURF, BAFQ2P2UNIV
 990  FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,' IMP:N=1 U=',I2)
      LN=LN+1
*    Write universe for baffle section F.
      WRITE(30,1000) LN, BMODML, (-1*MODDENSITY),
     c (-1*BAFEFGIJWESTSURF), BAFACDFGNORTHSURF, BAFQ2P3UNIV
1000  FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,' IMP:N=1 U=',I2)
      LN=LN+1
      WRITE(30,1010) LN, BMODML, (-1*MODDENSITY), BAFEFGIJEASTSURF,
     c (-1*BAFACDFGSOUTHSURF), BAFQ2P3UNIV
1010  FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,':',I4,' IMP:N=1 U=',I2)
      LN=LN+1
      WRITE(30,1020) LN, BAFFLEML, (-1*BAFFLEDENSITY),
     c BAFEFGIJWESTSURF, (-1*BAFACDFGNORTHSURF),
     c (-1*BAFEFGIJEASTSURF), BAFACDFGSOUTHSURF, BAFQ2P3UNIV
1020  FORMAT(T1,I4,T6,I4,T11,F7.4,T25,'(',I4,':',I4,') (',I4,1X,I4,
     c ') IMP:N=1 U=',I2)
      LN=LN+1
```

```
*     Write universe for baffle section G.
        WRITE(30,1030) LN, BMODML, (-1*MODDENSITY),
     c  (-1*BAFEFGIJWESTSURF), BAFACDFGNORTHSURF, BAFQ2P4UNIV
 1030   FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,':',I4,' IMP:N=1 U=',I2)
        LN=LN+1
        WRITE(30,1040) LN, BMODML, (-1*MODDENSITY), BAFEFGIJEASTSURF,
     c  (-1*BAFACDFGSOUTHSURF), BAFQ2P4UNIV
 1040   FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,' IMP:N=1 U=',I2)
        LN=LN+1
        WRITE(30,1050) LN, BAFFLEML, (-1*BAFFLEDENSITY),
     c  BAFEFGIJWESTSURF, (-1*BAFACDFGNORTHSURF),
     c  (-1*BAFEFGIJEASTSURF), BAFACDFGSOUTHSURF, BAFQ2P4UNIV
 1050   FORMAT(T1,I4,T6,I4,T11,F7.4,T25,'(',I4,1X,I4,') (',I4,':',I4,
     c  ') IMP:N=1 U=',I2)
        LN=LN+1
*     Write universe for baffle section H.
        WRITE(30,1060) LN, BMODML, (-1*MODDENSITY), BAFHIJKLNORTHSURF,
     c  BAFQ3P1UNIV
 1060   FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,' IMP:N=1 U=',I2)
        LN=LN+1
        WRITE(30,1070) LN, BMODML, (-1*MODDENSITY),
     c  (-1*BAFHIJKLSOUTHSURF), BAFQ3P1UNIV
 1070   FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,' IMP:N=1 U=',I2)
        LN=LN+1
        WRITE(30,1080) LN, BAFFLEML, (-1*BAFFLEDENSITY),
     c  (-1*BAFHIJKLNORTHSURF), BAFHIJKLSOUTHSURF, BAFQ3P1UNIV
 1080   FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,' IMP:N=1 U=',I2)
        LN=LN+1
*     Write universe for baffle section I.
        WRITE(30,1090) LN, BMODML, (-1*MODDENSITY),
     c  (-1*BAFEFGIJWESTSURF), (-1*BAFHIJKLSOUTHSURF), BAFQ3P3UNIV
 1090   FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,' IMP:N=1 U=',I2)
        LN=LN+1
        WRITE(30,1100) LN, BMODML, (-1*MODDENSITY), BAFEFGIJEASTSURF,
     c  BAFHIJKLNORTHSURF, BAFQ3P3UNIV
 1100   FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,':',I4,' IMP:N=1 U=',I2)
        LN=LN+1
        WRITE(30,1110) LN, BAFFLEML, (-1*BAFFLEDENSITY),
     c  BAFEFGIJWESTSURF, BAFHIJKLSOUTHSURF, (-1*BAFEFGIJEASTSURF),
     c  (-1*BAFHIJKLNORTHSURF), BAFQ3P3UNIV
 1110   FORMAT(T1,I4,T6,I4,T11,F7.4,T25,'(',I4,':',I4,') (',I4,1X,I4,
     c  ') IMP:N=1 U=',I2)
        LN=LN+1
*     Write universe for baffle section J.
        WRITE(30,1120) LN, BMODML, (-1*MODDENSITY),
     c  (-1*BAFEFGIJWESTSURF), (-1*BAFHIJKLSOUTHSURF), BAFQ3P4UNIV
 1120   FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,':',I4,' IMP:N=1 U=',I2)
        LN=LN+1
        WRITE(30,1130) LN, BMODML, (-1*MODDENSITY), BAFEFGIJEASTSURF,
     c  BAFHIJKLNORTHSURF, BAFQ3P4UNIV
 1130   FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,' IMP:N=1 U=',I2)
        LN=LN+1
        WRITE(30,1140) LN, BAFFLEML, (-1*BAFFLEDENSITY),
     c  BAFEFGIJWESTSURF, BAFHIJKLSOUTHSURF, (-1*BAFEFGIJEASTSURF),
```

```
     c    (-1*BAFHIJKLNORTHSURF), BAFQ3P4UNIV
 1140     FORMAT(T1,I4,T6,I4,T11,F7.4,T25,'(',I4,1X,I4,') (',I4,':',I4,
     c    ') IMP:N=1 U=',I2)
          LN=LN+1
*        Write universe for baffle section K.
          WRITE(30,1150) LN, BMODML, (-1*MODDENSITY), BAFBCDKLEASTSURF,
     c    (-1*BAFHIJKLSOUTHSURF), BAFQ4P3UNIV
 1150     FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,' IMP:N=1 U=',I2)
          LN=LN+1
          WRITE(30,1160) LN, BMODML, (-1*MODDENSITY),
     c    (-1*BAFBCDKLWESTSURF), BAFHIJKLNORTHSURF, BAFQ4P3UNIV
 1160     FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,':',I4,' IMP:N=1 U=',I2)
          LN=LN+1
          WRITE(30,1170) LN, BAFFLEML, (-1*BAFFLEDENSITY),
     c    (-1*BAFBCDKLEASTSURF), BAFHIJKLSOUTHSURF, BAFBCDKLWESTSURF,
     c    (-1*BAFHIJKLNORTHSURF), BAFQ4P3UNIV
 1170.    FORMAT(T1,I4,T6,I4,T11,F7.4,T25,'(',I4,':',I4,') (',I4,1X,I4,
     c    ') IMP:N=1 U=',I2)
          LN=LN+1
*        Write universe for baffle section L.
          WRITE(30,1180) LN, BMODML, (-1*MODDENSITY), BAFBCDKLEASTSURF,
     c    (-1*BAFHIJKLSOUTHSURF), BAFQ4P4UNIV
 1180     FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,':',I4,' IMP:N=1 U=',I2)
          LN=LN+1
          WRITE(30,1190) LN, BMODML, (-1*MODDENSITY),
     c    (-1*BAFBCDKLWESTSURF), BAFHIJKLNORTHSURF, BAFQ4P4UNIV
 1190     FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,' IMP:N=1 U=',I2)
          LN=LN+1
          WRITE(30,1200) LN, BAFFLEML, (-1*BAFFLEDENSITY),
     c    (-1*BAFBCDKLEASTSURF), BAFHIJKLSOUTHSURF, BAFBCDKLWESTSURF,
     c    (-1*BAFHIJKLNORTHSURF), BAFQ4P4UNIV
 1200     FORMAT(T1,I4,T6,I4,T11,F7.4,T25,'(',I4,1X,I4,') (',I4,':',I4,
     c    ') IMP:N=1 U=',I2)
          LN=LN+1
        ENDIF
        DO 1202 C=1,NUMOFFADESIGNS
          COMBOFLAG(C)=.FALSE.
 1202   CONTINUE
*       Write the specifications for the fuel assembly lattices.
        WRITE(30,1210)
 1210   FORMAT(T1,'C')
        WRITE(30,1220)
 1220   FORMAT(T1,'C    FUEL ASSEMBLY LATTICE SPECIFICATIONS')
        WRITE(30,1230)
 1230   FORMAT(T1,'C')
*
*
        FRUN=400
        CRUN=700
        BPRUN=800
        APSRUN=900
        GTUN=1000
        ITUN=1100
        DO 1234 COLUMN=1,50
```

```
         DO 1232 ROW=1,50
            FRUNIV(COLUMN,ROW)=0
            GTUNIV(COLUMN,ROW)=0
            ITUNIV(COLUMN,ROW)=0
            CRAUNIV(COLUMN,ROW)=0
            BPRAUNIV(COLUMN,ROW)=0
            APSRAUNIV(COLUMN,ROW)=0
            GTWRITE(COLUMN,ROW)=.FALSE.
 1232    CONTINUE
 1234 CONTINUE
      DO 1900 ROW=1,50
         DO 1890 COLUMN=1,50
         IF (ASSYID(COLUMN,ROW).NE.'      ') THEN
*  Determine if the fuel composition in the assembly is unique.
*  Assign the appropriate universe id to the fuel rods in the assembly.
            FUNIQUE(COLUMN,ROW)=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 1237 RO=1,(ROW-1)
                  DO 1236 CO=1,50
                     IF (ASSYID(COLUMN,ROW).EQ.ASSYID(CO,RO)) THEN
                        FUNIQUE(COLUMN,ROW)=.FALSE.
                        LEAVE=.TRUE.
                        FRUNIV(COLUMN,ROW)=FRUNIV(CO,RO)
                        EXIT
                     ENDIF
 1236             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 1237          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 1239 RO=ROW,ROW
                     DO 1238 CO=1,(COLUMN-1)
                        IF (ASSYID(COLUMN,ROW).EQ.ASSYID(CO,RO)) THEN
                           FUNIQUE(COLUMN,ROW)=.FALSE.
                           LEAVE=.TRUE.
                           FRUNIV(COLUMN,ROW)=FRUNIV(CO,RO)
                           EXIT
                        ENDIF
 1238                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 1239             CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 1241 RO=1,(ROW-1)
                  DO 1240 CO=1,50
                     IF (ASSYID(COLUMN,ROW).EQ.ASSYID(CO,RO)) THEN
                        FUNIQUE(COLUMN,ROW)=.FALSE.
                        LEAVE=.TRUE.
                        FRUNIV(COLUMN,ROW)=FRUNIV(CO,RO)
                        EXIT
```

```
                  ENDIF
 1240          CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
 1241       CONTINUE
         ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
            DO 1243 RO=1,1
               DO 1242 CO=1,(COLUMN-1)
                  IF (ASSYID(COLUMN,ROW).EQ.ASSYID(CO,RO)) THEN
                     FUNIQUE(COLUMN,ROW)=.FALSE.
                     LEAVE=.TRUE.
                     FRUNIV(COLUMN,ROW)=FRUNIV(CO,RO)
                     EXIT
                  ENDIF
 1242          CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
 1243       CONTINUE
         ENDIF
         IF (FUNIQUE(COLUMN,ROW).EQ..TRUE.) THEN
            FRUNIV(COLUMN,ROW)=FRUN
            FRUN=FRUN+1
         ENDIF
*  Determine if the assembly has a unique BPRA inserted.
         BPRUNIQUE(COLUMN,ROW)=.FALSE.
         IF (BANKNUM(COLUMN,ROW).NE.0) THEN
         IF (FUNIQUE(COLUMN,ROW).EQ..TRUE.) THEN
            IF (BANKDES(BANKNUM(COLUMN,ROW)).EQ.'BPRA ') THEN
               BPRUNIQUE(COLUMN,ROW)=.TRUE.
               BPRAUNIV(COLUMN,ROW)=BPRUN
               BPRUN=BPRUN+1
            ENDIF
         ELSEIF ((FUNIQUE(COLUMN,ROW).EQ..FALSE.).AND.
     c   (BANKDES(BANKNUM(COLUMN,ROW)).EQ.'BPRA ')) THEN
            BPRUNIQUE(COLUMN,ROW)=.TRUE.
            LEAVE=.FALSE.
            IF (COLUMN.NE.1) THEN
               DO 1245 RO=1,(ROW-1)
                  DO 1244 CO=1,50
                     IF ((ASSYID(COLUMN,ROW).NE.'      ').AND.
     c                  (ASSYID(COLUMN,ROW).EQ.ASSYID(CO,RO)).AND.
     c                  (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO))) THEN
                        BPRUNIQUE(COLUMN,ROW)=.FALSE.
                        LEAVE=.TRUE.
                        BPRAUNIV(COLUMN,ROW)=BPRAUNIV(CO,RO)
                        EXIT
                     ENDIF
 1244             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 1245          CONTINUE
```

```
              IF (LEAVE.EQ..FALSE.) THEN
                 DO 1247 RO=ROW,ROW
                    DO 1246 CO=1,(COLUMN-1)
                       IF ((ASSYID(COLUMN,ROW).NE.'      ').AND.
     c                 (ASSYID(COLUMN,ROW).EQ.ASSYID(CO,RO)).AND.
     c                 (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO))) THEN
                          BPRUNIQUE(COLUMN,ROW)=.FALSE.
                          LEAVE=.TRUE.
                          BPRAUNIV(COLUMN,ROW)=BPRAUNIV(CO,RO)
                          EXIT
                       ENDIF
 1246               CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
 1247            CONTINUE
              ENDIF
           ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
              DO 1249 RO=1,(ROW-1)
                 DO 1248 CO=1,50
                    IF ((ASSYID(COLUMN,ROW).NE.'      ').AND.
     c                 (ASSYID(COLUMN,ROW).EQ.ASSYID(CO,RO)).AND.
     c                 (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO))) THEN
                       BPRUNIQUE(COLUMN,ROW)=.FALSE.
                       LEAVE=.TRUE.
                       BPRAUNIV(COLUMN,ROW)=BPRAUNIV(CO,RO)
                       EXIT
                    ENDIF
 1248            CONTINUE
                 IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                 ENDIF
 1249         CONTINUE
           ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
              DO 1251 RO=1,1
                 DO 1250 CO=1,(COLUMN-1)
                    IF ((ASSYID(COLUMN,ROW).NE.'      ').AND.
     c                 (ASSYID(COLUMN,ROW).EQ.ASSYID(CO,RO)).AND.
     c                 (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO))) THEN
                       BPRUNIQUE(COLUMN,ROW)=.FALSE.
                       LEAVE=.TRUE.
                       BPRAUNIV(COLUMN,ROW)=BPRAUNIV(CO,RO)
                       EXIT
                    ENDIF
 1250            CONTINUE
                 IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                 ENDIF
 1251         CONTINUE
           ENDIF
           IF (BPRUNIQUE(COLUMN,ROW).EQ..TRUE.) THEN
              BPRAUNIV(COLUMN,ROW)=BPRUN
              BPRUN=BPRUN+1
           ENDIF
```

```
            ENDIF
            ENDIF
*  Determine if the assembly design is unique.
*  If the assembly design is unique, additional GT and IT universes will be
defined.
            DUNIQUE(COLUMN,ROW)=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 1253 RO=1,(ROW-1)
                  DO 1252 CO=1,50
                     IF (DESNUM(COLUMN,ROW).EQ.DESNUM(CO,RO)) THEN
                        DUNIQUE(COLUMN,ROW)=.FALSE.
                        ITUNIV(COLUMN,ROW)=ITUNIV(CO,RO)
                        LEAVE=.TRUE.
                        EXIT
                     ENDIF
 1252             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 1253          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 1255 RO=ROW,ROW
                     DO 1254 CO=1,(COLUMN-1)
                        IF (DESNUM(COLUMN,ROW).EQ.DESNUM(CO,RO)) THEN
                           DUNIQUE(COLUMN,ROW)=.FALSE.
                           ITUNIV(COLUMN,ROW)=ITUNIV(CO,RO)
                           LEAVE=.TRUE.
                           EXIT
                        ENDIF
 1254                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 1255             CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 1257 RO=1,(ROW-1)
                  DO 1256 CO=1,50
                     IF (DESNUM(COLUMN,ROW).EQ.DESNUM(CO,RO)) THEN
                        DUNIQUE(COLUMN,ROW)=.FALSE.
                        ITUNIV(COLUMN,ROW)=ITUNIV(CO,RO)
                        LEAVE=.TRUE.
                        EXIT
                     ENDIF
 1256             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 1257          CONTINUE
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
               DO 1259 RO=1,1
                  DO 1258 CO=1,(COLUMN-1)
                     IF (DESNUM(COLUMN,ROW).EQ.DESNUM(CO,RO)) THEN
```

```
                          DUNIQUE(COLUMN,ROW)=.FALSE.
                          ITUNIV(COLUMN,ROW)=ITUNIV(CO,RO)
                          LEAVE=.TRUE.
                          EXIT
                       ENDIF
 1258               CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
 1259            CONTINUE
              ENDIF
              IF (BANKNUM(COLUMN,ROW).EQ.0) THEN
                 IF (COMBOFLAG(DESNUM(COLUMN,ROW)).EQ..FALSE.) THEN
                    COMBOFLAG(DESNUM(COLUMN,ROW))=.TRUE.
                    GTUNIV(COLUMN,ROW)=GTUN
                    GTWRITE(COLUMN,ROW)=.TRUE.
                    COMBOVAL(DESNUM(COLUMN,ROW))=GTUN
                    GTUN=GTUN+1
                 ELSEIF (COMBOFLAG(DESNUM(COLUMN,ROW)).EQ..TRUE.) THEN
                    GTUNIV(COLUMN,ROW)=COMBOVAL(DESNUM(COLUMN,ROW))
                 ENDIF
              ELSEIF (WESTINGHOUSE.EQ..TRUE.) THEN
                 IF ((WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.1).OR.
     c              (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.2).OR.
     c              (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.3).OR.
     c              (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.4).OR.
     c              (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.5).OR.
     c              (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.6).OR.
     c              (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.7)) THEN
                       IF (COMBOFLAG(DESNUM(COLUMN,ROW)).EQ.
     c                 .FALSE.) THEN
                          COMBOFLAG(DESNUM(COLUMN,ROW))=.TRUE.
                          GTUNIV(COLUMN,ROW)=GTUN
                          GTWRITE(COLUMN,ROW)=.TRUE.
                          COMBOVAL(DESNUM(COLUMN,ROW))=GTUN
                          GTUN=GTUN+1
                       ELSEIF (COMBOFLAG(DESNUM(COLUMN,ROW))
     c                 .EQ..TRUE.) THEN
                          GTUNIV(COLUMN,ROW)=
     c                    COMBOVAL(DESNUM(COLUMN,ROW))
                       ENDIF
                 ENDIF
              ENDIF
              IF (DUNIQUE(COLUMN,ROW).EQ..TRUE.) THEN
                 ITUNIV(COLUMN,ROW)=ITUN
                 ITUN=ITUN+1
              ENDIF
* Determine if the assembly has a unique CRA inserted.
* If the assembly contains a unique CRA, give the CRA an appropriate
universe id.
              CRUNIQUE(COLUMN,ROW)=.FALSE.
              IF (BANKNUM(COLUMN,ROW).NE.0) THEN
              IF ((DUNIQUE(COLUMN,ROW).EQ..TRUE.).AND.
     c        (BANKDES(BANKNUM(COLUMN,ROW)).EQ.'CRA  ')) THEN
```

```
              CRUNIQUE(COLUMN,ROW)=.TRUE.
              CRAUNIV(COLUMN,ROW)=CRUN
              CRUN=CRUN+1
           ELSEIF ((DUNIQUE(COLUMN,ROW).EQ..FALSE.).AND.
     c     (BANKDES(BANKNUM(COLUMN,ROW)).EQ.'CRA   ')) THEN
              CRUNIQUE(COLUMN,ROW)=.TRUE.
              LEAVE=.FALSE.
              IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                 DO 1261 RO=1,(ROW-1)
                    DO 1260 CO=1,50
                       IF (DESNUM(COLUMN,ROW).EQ.DESNUM(CO,RO)) THEN
                          CRUNIQUE(COLUMN,ROW)=.FALSE.
                          LEAVE=.TRUE.
                          CRAUNIV(COLUMN,ROW)=CRAUNIV(CO,RO)
                          EXIT
                       ENDIF
 1260               CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
 1261            CONTINUE
                 IF (LEAVE.EQ..FALSE.) THEN
                    DO 1264 RO=ROW,ROW
                       DO 1263 CO=1,(COLUMN-1)
                          IF (DESNUM(COLUMN,ROW).EQ.
     c                    DESNUM(CO,RO)) THEN
                             CRUNIQUE(COLUMN,ROW)=.FALSE.
                             LEAVE=.TRUE.
                             CRAUNIV(COLUMN,ROW)=CRAUNIV(CO,RO)
                             EXIT
                          ENDIF
 1263                  CONTINUE
                       IF (LEAVE.EQ..TRUE.) THEN
                          EXIT
                       ENDIF
 1264               CONTINUE
                 ENDIF
              ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                 DO 1266 RO=1,(ROW-1)
                    DO 1265 CO=1,50
                       IF (DESNUM(COLUMN,ROW).EQ.DESNUM(CO,RO)) THEN
                          CRUNIQUE(COLUMN,ROW)=.FALSE.
                          LEAVE=.TRUE.
                          CRAUNIV(COLUMN,ROW)=CRAUNIV(CO,RO)
                          EXIT
                       ENDIF
 1265               CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
 1266            CONTINUE
              ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                 DO 1268 RO=1,1
                    DO 1267 CO=1,(COLUMN-1)
```

```
                        IF (DESNUM(COLUMN,ROW).EQ.DESNUM(CO,RO)) THEN
                           CRUNIQUE(COLUMN,ROW)=.FALSE.
                           LEAVE=.TRUE.
                           CRAUNIV(COLUMN,ROW)=CRAUNIV(CO,RO)
                           EXIT
                        ENDIF
 1267                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 1268             CONTINUE
               ENDIF
               IF (CRUNIQUE(COLUMN,ROW).EQ..TRUE.) THEN
                  CRAUNIV(COLUMN,ROW)=CRUN
                  CRUN=CRUN+1
               ENDIF
            ENDIF
            ENDIF
*  Determine if the assembly has a unique APSRA inserted.
*  If the assembly contains a unique APSRA, give the APSRA an appropriate
universe id.
            APSRUNIQUE(COLUMN,ROW)=.FALSE.
            IF (BANKNUM(COLUMN,ROW).NE.0) THEN
            IF ((DUNIQUE(COLUMN,ROW).EQ..TRUE.).AND.
     c      (BANKDES(BANKNUM(COLUMN,ROW)).EQ.'APSRA')) THEN
               IF (BANKDES(BANKNUM(COLUMN,ROW)).EQ.'APSRA') THEN
                  APSRUNIQUE(COLUMN,ROW)=.TRUE.
                  APSRAUNIV(COLUMN,ROW)=APSRUN
                  APSRUN=APSRUN+1
               ENDIF
            ELSEIF ((DUNIQUE(COLUMN,ROW).EQ..FALSE.).AND.
     c      (BANKDES(BANKNUM(COLUMN,ROW)).EQ.'APSRA')) THEN
               APSRUNIQUE(COLUMN,ROW)=.TRUE.
               LEAVE=.FALSE.
               IF (COLUMN.NE.1) THEN
                  DO 1270 RO=1,(ROW-1)
                     DO 1269 CO=1,50
                        IF (DESNUM(COLUMN,ROW).EQ.DESNUM(CO,RO)) THEN
                           APSRUNIQUE(COLUMN,ROW)=.FALSE.
                           LEAVE=.TRUE.
                           APSRAUNIV(COLUMN,ROW)=APSRAUNIV(CO,RO)
                           EXIT
                        ENDIF
 1269                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 1270             CONTINUE
                  IF (LEAVE.EQ..FALSE.) THEN
                     DO 1272 RO=ROW,ROW
                        DO 1271 CO=1,(COLUMN-1)
                           IF (DESNUM(COLUMN,ROW).EQ.
     c                     DESNUM(CO,RO)) THEN
                              APSRUNIQUE(COLUMN,ROW)=.FALSE.
```

```
                                 LEAVE=.TRUE.
                                 APSRAUNIV(COLUMN,ROW)=APSRAUNIV(CO,RO)
                                 EXIT
                              ENDIF
 1271                      CONTINUE
                           IF (LEAVE.EQ..TRUE.) THEN
                              EXIT
                           ENDIF
 1272                   CONTINUE
                     ENDIF
                  ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                     DO 1274 RO=1,(ROW-1)
                        DO 1273 CO=1,50
                           IF (DESNUM(COLUMN,ROW).EQ.DESNUM(CO,RO)) THEN
                              APSRUNIQUE(COLUMN,ROW)=.FALSE.
                              LEAVE=.TRUE.
                              APSRAUNIV(COLUMN,ROW)=APSRAUNIV(CO,RO)
                              EXIT
                           ENDIF
 1273                   CONTINUE
                        IF (LEAVE.EQ..TRUE.) THEN
                           EXIT
                        ENDIF
 1274                CONTINUE
                  ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                     DO 1276 RO=1,1
                        DO 1275 CO=1,(COLUMN-1)
                           IF (DESNUM(COLUMN,ROW).EQ.DESNUM(CO,RO)) THEN
                              APSRUNIQUE(COLUMN,ROW)=.FALSE.
                              LEAVE=.TRUE.
                              APSRAUNIV(COLUMN,ROW)=APSRAUNIV(CO,RO)
                              EXIT
                           ENDIF
 1275                   CONTINUE
                        IF (LEAVE.EQ..TRUE.) THEN
                           EXIT
                        ENDIF
 1276                CONTINUE
                  ENDIF
                  IF (APSRUNIQUE(COLUMN,ROW).EQ..TRUE.) THEN
                     APSRAUNIV(COLUMN,ROW)=APSRUN
                     APSRUN=APSRUN+1
                  ENDIF
               ENDIF
               ENDIF
*  Write the assembly layout specification if any of the following are true:
*  FUNIQUE(COLUMN,ROW)=.TRUE., DUNIQUE(COLUMN,ROW)=.TRUE.,
BPRUNIQUE(COLUMN,ROW)=.TRUE.,
*  CRUNIQUE(COLUMN,ROW)=.TRUE., APSRUNIQUE(COLUMN,ROW)=.TRUE.
               IF ((FUNIQUE(COLUMN,ROW).EQ..TRUE.)
     c            .OR.(DUNIQUE(COLUMN,ROW).EQ..TRUE.)
     c            .OR.(BPRUNIQUE(COLUMN,ROW).EQ..TRUE.)
     c            .OR.(CRUNIQUE(COLUMN,ROW).EQ..TRUE.)
     c            .OR.(APSRUNIQUE(COLUMN,ROW).EQ..TRUE.)) THEN
```

```
*  Write the header for the assembly layout specification.
                    IF (BANKNUM(COLUMN,ROW).NE.0) THEN
                    IF (BANKDES(BANKNUM(COLUMN,ROW)).EQ.'BPRA ') THEN
                       WRITE(30,1277) ASSYID(COLUMN,ROW)
 1277                  FORMAT(T1,'C   ASSEMBLY ',A5,' LAYOUT WITH ',
     c                 'BPRA INSERTED')
                    ELSEIF (BANKDES(BANKNUM(COLUMN,ROW)).EQ.'CRA  ') THEN
                       WRITE(30,1278) ASSYID(COLUMN,ROW)
 1278                  FORMAT(T1,'C   ASSEMBLY ',A5,' LAYOUT WITH ',
     c                 'CRA INSERTED')
                    ELSEIF (BANKDES(BANKNUM(COLUMN,ROW)).EQ.'APSRA') THEN
                       WRITE(30,1280) ASSYID(COLUMN,ROW)
 1280                  FORMAT(T1,'C   ASSEMBLY ',A5,' LAYOUT WITH ',
     c                 'APSRA INSERTED')
                    ENDIF
                    ENDIF
                    IF (BANKNUM(COLUMN,ROW).EQ.0) THEN
                       WRITE(30,1282) ASSYID(COLUMN,ROW)
 1282                  FORMAT(T1,'C   ASSEMBLY ',A5,' LAYOUT WITH ',
     c                 'NO RODDED INSERTION ASSEMBLY')
                    ENDIF
*  Define the pin pitch surfaces.
                    IF (DESNUM(COLUMN,ROW).NE.0) THEN
                    DNUM=DESNUM(COLUMN,ROW)
                    PITCH=PINPITCH(DNUM)
                    ENDIF
                    VAL5=(PITCH/2.0)
                    DO 1284 SURFNUM=1,SN-1
                       IF ((SURFTYPESPEC(SURFNUM).EQ.'PY').AND.
     c                    (SURFVALUESPEC(SURFNUM).EQ.VAL5)) THEN
                          PITCHNORTH=SURFNUM ! Northern surface of unit pin cell
                          PITCHSOUTH=SURFNUM+1 ! Southern surface of unit pin cell
                          PITCHEAST=SURFNUM+2 ! Eastern surface of unit pin cell
                          PITCHWEST=SURFNUM+3 ! Western surface of unit pin cell
                          IF (SURFNUM.NE.(SN-1)) THEN
                             EXIT
                          ENDIF
                       ELSEIF(SURFNUM.EQ.(SN-1)) THEN
                          PITCHNORTH=SN ! Northern surface of unit pin cell
                          SURFTYPESPEC(SN)='PY'
                          SURFVALUESPEC(SN)=(PITCH/2.0)
                          SN=SN+1
                          PITCHSOUTH=SN ! Southern surface of unit pin cell
                          SURFTYPESPEC(SN)='PY'
                          SURFVALUESPEC(SN)=(-1*(PITCH/2.0))
                          SN=SN+1
                          PITCHEAST=SN ! Eastern surface of unit pin cell
                          SURFTYPESPEC(SN)='PX'
                          SURFVALUESPEC(SN)=(PITCH/2.0)
                          SN=SN+1
                          PITCHWEST=SN ! Western surface of unit pin cell
                          SURFTYPESPEC(SN)='PX'
                          SURFVALUESPEC(SN)=(-1*(PITCH/2.0))
                          SN=SN+1
```

```
                  ENDIF
 1284          CONTINUE
*  Write the basic assembly layout specification cards.
               WRITE(30,1286) LN, BMODML, (-1*MODDENSITY),
    c          (-1*PITCHEAST), PITCHWEST, (-1*PITCHNORTH),
    c          PITCHSOUTH, ASSYUNIV(COLUMN,ROW)
 1286          FORMAT(T1,I4,T6,I4,T11,F7.4,T25,I4,1X,I4,1X,I4,1X,I4,
    c          ' IMP:N=1 LAT=1 U=',I3)
               LN=LN+1
               IF (BANDW.EQ..TRUE.) THEN
                  WRITE(30,1288)
 1288             FORMAT(T11,'FILL -9:9 -9:9 0:0')
               ELSEIF (WESTINGHOUSE.EQ..TRUE.) THEN
                  WRITE(30,1290)
 1290             FORMAT(T11,'FILL -10:10 -10:10 0:0')
               ENDIF
*  Determine whether or not this assembly contains a BPRA, CRA, or APSRA.
               BPRAINSERTED=.FALSE.
               CRAINSERTED=.FALSE.
               APSRAINSERTED=.FALSE.
               IF (BANKNUM(COLUMN,ROW).NE.0) THEN
                  IF (BANKDES(BANKNUM(COLUMN,ROW)).EQ.'BPRA ') THEN
                     BPRAINSERTED=.TRUE.
                  ELSEIF (BANKDES(BANKNUM(COLUMN,ROW)).EQ.'CRA  ') THEN
                     CRAINSERTED=.TRUE.
                  ELSEIF (BANKDES(BANKNUM(COLUMN,ROW)).EQ.'APSRA') THEN
                     APSRAINSERTED=.TRUE.
                  ENDIF
               ENDIF
         IF (BANDW.EQ..TRUE.) THEN
*  Write the assembly lattice layout specification.
               WRITE(30,1300) ASSYUNIV(COLUMN,ROW) ! First framing row
 1300          FORMAT(T11,I3,' 18R')
               WRITE(30,1302) ASSYUNIV(COLUMN,ROW) ! Second framing row
 1302          FORMAT(T11,I3,' 18R')
               WRITE(30,1304) ASSYUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW), !
Fuel row 1
    c          ASSYUNIV(COLUMN,ROW)
 1304          FORMAT(T11,I3,' 1R ',I3,' 14R ',I3,' 1R')
               WRITE(30,1306) ASSYUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW), !
Fuel row 2
    c          ASSYUNIV(COLUMN,ROW)
 1306          FORMAT(T11,I3,' 1R ',I3,' 14R ',I3,' 1R')
               IF ((CRAINSERTED.EQ..FALSE.).AND.
    c             (APSRAINSERTED.EQ..FALSE.).AND.
    c             (BPRAINSERTED.EQ..FALSE.)) THEN
                  WRITE(30,1308) ASSYUNIV(COLUMN,ROW), ! Fuel row 3
    c             FRUNIV(COLUMN,ROW),
    c             GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
    c             GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
    c             ASSYUNIV(COLUMN,ROW)
 1308             FORMAT(T11,I3,' 1R ',I3,' 4R ',I4,1X,I3,
    c             ' 2R ',I4,1X,I3,' 4R ',I3,' 1R ')
                  WRITE(30,1310) ASSYUNIV(COLUMN,ROW), ! Fuel row 4
```

```
     c              FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1310               FORMAT(T11,I3,' 1R ',I3,' 2R ',I4,1X,I3,
     c              ' 6R ',I4,1X,I3,' 2R ',I3,' 1R ')
                    WRITE(30,1312) ASSYUNIV(COLUMN,ROW), ! Fuel row 5
     c              FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1312               FORMAT(T11,I3,' 1R ',I3,' 14R ',I3,' 1R')
                    WRITE(30,1314) ASSYUNIV(COLUMN,ROW), ! Fuel row 6
     c              FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1314               FORMAT(T11,I3,' 1R ',I3,' 1R ',I4,1X,I3,
     c              ' 1R ',I4,1X,I3,' 2R ',I4,1X,I3,' 1R ',
     c              I4,1X,I3,' 1R ',I3,' 1R ')
                    WRITE(30,1316) ASSYUNIV(COLUMN,ROW), ! Fuel row 7
     c              FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1316               FORMAT(T11,I3,' 1R ',I3,' 14R ',I3,' 1R')
                    WRITE(30,1318) ASSYUNIV(COLUMN,ROW), ! Fuel row 8
     c              FRUNIV(COLUMN,ROW),
     c              ITUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1318               FORMAT(T11,I3,' 1R ',I3,' 6R ',I4,1X,I3,
     c              ' 6R ',I3,' 1R ')
                    WRITE(30,1320) ASSYUNIV(COLUMN,ROW), ! Fuel row 9
     c              FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1320               FORMAT(T11,I3,' 1R ',I3,' 14R ',I3,' 1R')
                    WRITE(30,1322) ASSYUNIV(COLUMN,ROW), ! Fuel row 10
     c              FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1322               FORMAT(T11,I3,' 1R ',I3,' 1R ',I4,1X,I3,
     c              ' 1R ',I4,1X,I3,' 2R ',I4,1X,I3,' 1R ',
     c              I4,1X,I3,' 1R ',I3,' 1R ')
                    WRITE(30,1324) ASSYUNIV(COLUMN,ROW), ! Fuel row 11
     c              FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1324               FORMAT(T11,I3,' 1R ',I3,' 14R ',I3,' 1R')
                    WRITE(30,1326) ASSYUNIV(COLUMN,ROW), ! Fuel row 12
     c              FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1326               FORMAT(T11,I3,' 1R ',I3,' 2R ',I4,1X,I3,
```

```
     c             ' 6R ',I4,1X,I3,' 2R ',I3,' 1R ')
                   WRITE(30,1328) ASSYUNIV(COLUMN,ROW), ! Fuel row 13
     c             FRUNIV(COLUMN,ROW),
     c             GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c             GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c             ASSYUNIV(COLUMN,ROW)
 1328              FORMAT(T11,I3,' 1R ',I3,' 4R ',I4,1X,I3,
     c             ' 2R ',I4,1X,I3,' 4R ',I3,' 1R ')
                ELSEIF (BPRAINSERTED.EQ..TRUE.) THEN
                   WRITE(30,1330) ASSYUNIV(COLUMN,ROW), ! Fuel row 3
     c             FRUNIV(COLUMN,ROW),
     c             BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c             BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c             ASSYUNIV(COLUMN,ROW)
 1330              FORMAT(T11,I3,' 1R ',I3,' 4R ',I3,1X,I3,
     c             ' 2R ',I3,1X,I3,' 4R ',I3,' 1R ')
                   WRITE(30,1332) ASSYUNIV(COLUMN,ROW), ! Fuel row 4
     c             FRUNIV(COLUMN,ROW),
     c             BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c             BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c             ASSYUNIV(COLUMN,ROW)
 1332              FORMAT(T11,I3,' 1R ',I3,' 2R ',I3,1X,I3,
     c             ' 6R ',I3,1X,I3,' 2R ',I3,' 1R ')
                   WRITE(30,1334) ASSYUNIV(COLUMN,ROW), ! Fuel row 5
     c             FRUNIV(COLUMN,ROW),
     c           - ASSYUNIV(COLUMN,ROW)
 1334              FORMAT(T11,I3,' 1R ',I3,' 14R ',I3,' 1R')
                   WRITE(30,1336) ASSYUNIV(COLUMN,ROW), ! Fuel row 6
     c             FRUNIV(COLUMN,ROW),
     c             BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c             BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN;ROW),
     c             BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c             BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c             ASSYUNIV(COLUMN,ROW)
 1336              FORMAT(T11,I3,' 1R ',I3,' 1R ',I3,1X,I3,
     c             ' 1R ',I3,1X,I3,' 2R ',I3,1X,I3,' 1R ',
     c             I3,1X,I3,' 1R ',I3,' 1R ')
                   WRITE(30,1338) ASSYUNIV(COLUMN,ROW), ! Fuel row 7
     c             FRUNIV(COLUMN,ROW),
     c             ASSYUNIV(COLUMN,ROW)
 1338              FORMAT(T11,I3,' 1R ',I3,' 14R ',I3,' 1R')
                   WRITE(30,1340) ASSYUNIV(COLUMN,ROW), ! Fuel row 8
     c             FRUNIV(COLUMN,ROW),
     c             ITUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c             ASSYUNIV(COLUMN,ROW)
 1340              FORMAT(T11,I3,' 1R ',I3,' 6R ',I4,1X,I3,
     c             ' 6R ',I3,' 1R ')
                   WRITE(30,1342) ASSYUNIV(COLUMN,ROW), ! Fuel row 9
     c             FRUNIV(COLUMN,ROW),
     c             ASSYUNIV(COLUMN,ROW)
 1342              FORMAT(T11,I3,' 1R ',I3,' 14R ',I3,' 1R')
                   WRITE(30,1344) ASSYUNIV(COLUMN,ROW), ! Fuel row 10
     c             FRUNIV(COLUMN,ROW),
     c             BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
```

```
     c              BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1344               FORMAT(T11,I3,' 1R ',I3,' 1R ',I3,1X,I3,
     c              ' 1R ',I3,1X,I3,' 2R ',I3,1X,I3,' 1R ',
     c              I3,1X,I3,' 1R ',I3,' 1R ')
                    WRITE(30,1346) ASSYUNIV(COLUMN,ROW), ! Fuel row 11
     c              FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1346               FORMAT(T11,I3,' 1R ',I3,' 14R ',I3,' 1R')
                    WRITE(30,1348) ASSYUNIV(COLUMN,ROW), ! Fuel row 12
     c              FRUNIV(COLUMN,ROW),
     c              BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1348               FORMAT(T11,I3,' 1R ',I3,' 2R ',I3,1X,I3,
     c              ' 6R ',I3,1X,I3,' 2R ',I3,' 1R ')
                    WRITE(30,1350) ASSYUNIV(COLUMN,ROW), ! Fuel row 13
     c              FRUNIV(COLUMN,ROW),
     c              BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1350               FORMAT(T11,I3,' 1R ',I3,' 4R ',I3,1X,I3,
     c              ' 2R ',I3,1X,I3,' 4R ',I3,' 1R ')
                 ELSEIF (CRAINSERTED.EQ..TRUE.) THEN
                    WRITE(30,1352) ASSYUNIV(COLUMN,ROW), ! Fuel row 3
     c              FRUNIV(COLUMN,ROW),
     c              CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1352               FORMAT(T11,I3,' 1R ',I3,' 4R ',I3,1X,I3,
     c              ' 2R ',I3,1X,I3,' 4R ',I3,' 1R ')
                    WRITE(30,1354) ASSYUNIV(COLUMN,ROW), ! Fuel row 4
     c              FRUNIV(COLUMN,ROW),
     c              CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1354               FORMAT(T11,I3,' 1R ',I3,' 2R ',I3,1X,I3,
     c              ' 6R ',I3,1X,I3,' 2R ',I3,' 1R ')
                    WRITE(30,1356) ASSYUNIV(COLUMN,ROW), ! Fuel row 5
     c              FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1356               FORMAT(T11,I3,' 1R ',I3,' 14R ',I3,' 1R')
                    WRITE(30,1358) ASSYUNIV(COLUMN,ROW), ! Fuel row 6
     c              FRUNIV(COLUMN,ROW),
     c              CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1358               FORMAT(T11,I3,' 1R ',I3,' 1R ',I3,1X,I3,
     c              ' 1R ',I3,1X,I3,' 2R ',I3,1X,I3,' 1R ',
     c              I3,1X,I3,' 1R ',I3,' 1R ')
```

```
               WRITE(30,1360) ASSYUNIV(COLUMN,ROW), ! Fuel row 7
     c         FRUNIV(COLUMN,ROW),
     c         ASSYUNIV(COLUMN,ROW)
 1360          FORMAT(T11,I3,' 1R ',I3,' 14R ',I3,' 1R')
               WRITE(30,1362) ASSYUNIV(COLUMN,ROW), ! Fuel row 8
     c         FRUNIV(COLUMN,ROW),
     c         ITUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c         ASSYUNIV(COLUMN,ROW)
 1362          FORMAT(T11,I3,' 1R ',I3,' 6R ',I4,1X,I3,
     c         ' 6R ',I3,' 1R ')
               WRITE(30,1364) ASSYUNIV(COLUMN,ROW), ! Fuel row 9
     c         FRUNIV(COLUMN,ROW),
     c         ASSYUNIV(COLUMN,ROW)
 1364          FORMAT(T11,I3,' 1R ',I3,' 14R ',I3,' 1R')
               WRITE(30,1366) ASSYUNIV(COLUMN,ROW), ! Fuel row 10
     c         FRUNIV(COLUMN,ROW),
     c         CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c         CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c         CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c         CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c         ASSYUNIV(COLUMN,ROW)
 1366          FORMAT(T11,I3,' 1R ',I3,' 1R ',I3,1X,I3,
     c         ' 1R ',I3,1X,I3,' 2R ',I3,1X,I3,' 1R ',
     c         I3,1X,I3,' 1R ',I3,' 1R ')
               WRITE(30,1368) ASSYUNIV(COLUMN,ROW), ! Fuel row 11
     c         FRUNIV(COLUMN,ROW),
     c         ASSYUNIV(COLUMN,ROW)
 1368          FORMAT(T11,I3,' 1R ',I3,' 14R ',I3,' 1R')
               WRITE(30,1370) ASSYUNIV(COLUMN,ROW), ! Fuel row 12
     c         FRUNIV(COLUMN,ROW),
     c         CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c         CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c         ASSYUNIV(COLUMN,ROW)
 1370          FORMAT(T11,I3,' 1R ',I3,' 2R ',I3,1X,I3,
     c         ' 6R ',I3,1X,I3,' 2R ',I3,' 1R ')
               WRITE(30,1372) ASSYUNIV(COLUMN,ROW), ! Fuel row 13
     c         FRUNIV(COLUMN,ROW),
     c         CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c         CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c         ASSYUNIV(COLUMN,ROW)
 1372          FORMAT(T11,I3,' 1R ',I3,' 4R ',I3,1X,I3,
     c         ' 2R ',I3,1X,I3,' 4R ',I3,' 1R ')
            ELSEIF (APSRAINSERTED.EQ..TRUE.) THEN
               WRITE(30,1374) ASSYUNIV(COLUMN,ROW), ! Fuel row 3
     c         FRUNIV(COLUMN,ROW),
     c         APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c         APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c         ASSYUNIV(COLUMN,ROW)
 1374          FORMAT(T11,I3,' 1R ',I3,' 4R ',I3,1X,I3,
     c         ' 2R ',I3,1X,I3,' 4R ',I3,' 1R ')
               WRITE(30,1376) ASSYUNIV(COLUMN,ROW), ! Fuel row 4
     c         FRUNIV(COLUMN,ROW),
     c         APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c         APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
```

```
     c          ASSYUNIV(COLUMN,ROW)
 1376           FORMAT(T11,I3,' 1R ',I3,' 2R ',I3,1X,I3,
     c          ' 6R ',I3,1X,I3,' 2R ',I3,' 1R ')
                WRITE(30,1378) ASSYUNIV(COLUMN,ROW), ! Fuel row 5
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
 1378           FORMAT(T11,I3,' 1R ',I3,' 14R ',I3,' 1R')
                WRITE(30,1380) ASSYUNIV(COLUMN,ROW), ! Fuel row 6
     c          FRUNIV(COLUMN,ROW),
     c          APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
 1380           FORMAT(T11,I3,' 1R ',I3,' 1R ',I3,1X,I3,
     c          ' 1R ',I3,1X,I3,' 2R ',I3,1X,I3,' 1R ',
     c          I3,1X,I3,' 1R ',I3,' 1R ')
                WRITE(30,1382) ASSYUNIV(COLUMN,ROW), ! Fuel row 7
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
 1382           FORMAT(T11,I3,' 1R ',I3,' 14R ',I3,' 1R')
                WRITE(30,1384) ASSYUNIV(COLUMN,ROW), ! Fuel row 8
     c          FRUNIV(COLUMN,ROW),
     c          ITUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
 1384           FORMAT(T11,I3,' 1R ',I3,' 6R ',I4,1X,I3,
     c          ' 6R ',I3,' 1R ')
                WRITE(30,1386) ASSYUNIV(COLUMN,ROW), ! Fuel row 9
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
 1386           FORMAT(T11,I3,' 1R ',I3,' 14R ',I3,' 1R')
                WRITE(30,1388) ASSYUNIV(COLUMN,ROW), ! Fuel row 10
     c          FRUNIV(COLUMN,ROW),
     c          APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
 1388           FORMAT(T11,I3,' 1R ',I3,' 1R ',I3,1X,I3,
     c          ' 1R ',I3,1X,I3,' 2R ',I3,1X,I3,' 1R ',
     c          I3,1X,I3,' 1R ',I3,' 1R ')
                WRITE(30,1390) ASSYUNIV(COLUMN,ROW), ! Fuel row 11
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
 1390           FORMAT(T11,I3,' 1R ',I3,' 14R ',I3,' 1R')
                WRITE(30,1392) ASSYUNIV(COLUMN,ROW), ! Fuel row 12
     c          FRUNIV(COLUMN,ROW),
     c          APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
 1392           FORMAT(T11,I3,' 1R ',I3,' 2R ',I3,1X,I3,
     c          ' 6R ',I3,1X,I3,' 2R ',I3,' 1R ')
                WRITE(30,1394) ASSYUNIV(COLUMN,ROW), ! Fuel row 13
     c          FRUNIV(COLUMN,ROW),
```

```
     c              APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1394               FORMAT(T11,I3,' 1R ',I3,' 4R ',I3,1X,I3,
     c              ' 2R ',I3,1X,I3,' 4R ',I3,' 1R ')
                 ENDIF
                 WRITE(30,1396) ASSYUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW), !
Fuel row 14
     c           ASSYUNIV(COLUMN,ROW)
 1396            FORMAT(T11,I3,' 1R ',I3,' 14R ',I3,' 1R')
                 WRITE(30,1398) ASSYUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW), !
Fuel row 15
     c           ASSYUNIV(COLUMN,ROW)
 1398            FORMAT(T11,I3,' 1R ',I3,' 14R ',I3,' 1R')
                 WRITE(30,1400) ASSYUNIV(COLUMN,ROW) ! Second-to-last framing
row
 1400            FORMAT(T11,I3,' 18R')
                 WRITE(30,1402) ASSYUNIV(COLUMN,ROW) ! Last framing row
 1402            FORMAT(T11,I3,' 18R')
*
              ELSEIF (WESTINGHOUSE.EQ..TRUE.) THEN
*
                 WRITE(30,1404) ASSYUNIV(COLUMN,ROW) ! First framing row
 1404            FORMAT(T11,I3,' 20R')
                 WRITE(30,1406) ASSYUNIV(COLUMN,ROW) ! Second framing row
 1406            FORMAT(T11,I3,' 20R')
                 WRITE(30,1408) ASSYUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW), !
Fuel row 1
     c           ASSYUNIV(COLUMN,ROW)
 1408            FORMAT(T11,I3,' 1R ',I3,' 16R ',I3,' 1R')
                 WRITE(30,1410) ASSYUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW), !
Fuel row 2
     c           ASSYUNIV(COLUMN,ROW)
 1410            FORMAT(T11,I3,' 1R ',I3,' 16R ',I3,' 1R')
                 IF ((CRAINSERTED.EQ..FALSE.).AND.
     c              (APSRAINSERTED.EQ..FALSE.).AND.
     c              (BPRAINSERTED.EQ..FALSE.)) THEN
                    WRITE(30,1412) ASSYUNIV(COLUMN,ROW), ! Fuel row 3
     c              FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1412               FORMAT(T11,I3,' 1R ',I3,' 4R ',I4,1X,I3,
     c              ' 1R ',I4,1X,I3,' 1R ',I4,1X,I3,' 4R ',I3,' 1R ')
                    WRITE(30,1414) ASSYUNIV(COLUMN,ROW), ! Fuel row 4
     c              FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1414               FORMAT(T11,I3,' 1R ',I3,' 2R ',I4,1X,I3,
     c              ' 8R ',I4,1X,I3,' 2R ',I3,' 1R ')
                    WRITE(30,1416) ASSYUNIV(COLUMN,ROW), ! Fuel row 5
     c              FRUNIV(COLUMN,ROW),
```

```
     c              ASSYUNIV(COLUMN,ROW)
 1416               FORMAT(T11,I3,' 1R ',I3,' 16R ',I3,' 1R')
                    WRITE(30,1418) ASSYUNIV(COLUMN,ROW), ! Fuel row 6
     c              FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW)
                    WRITE(30,1419) FRUNIV(COLUMN,ROW), ! Fuel row 6
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1418               FORMAT(T11,I3,' 1R ',I3,' 1R ',I4,1X,I3,
     c              ' 1R ',I4,1X,I3,' 1R ',I4)
 1419               FORMAT(T11,I3,' 1R ',
     c              I4,1X,I3,' 1R ',I4,1X,I3,' 1R ',I3,' 1R ')
                    WRITE(30,1420) ASSYUNIV(COLUMN,ROW), ! Fuel row 7
     c              FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1420               FORMAT(T11,I3,' 1R ',I3,' 16R ',I3,' 1R')
                    WRITE(30,1422) ASSYUNIV(COLUMN,ROW), ! Fuel row 8
     c              FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1422               FORMAT(T11,I3,' 1R ',I3,' 16R ',I3,' 1R')
                    WRITE(30,1424) ASSYUNIV(COLUMN,ROW), ! Fuel row 9
     c              FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c              FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c              FRUNIV(COLUMN,ROW), ITUNIV(COLUMN,ROW)
                    WRITE(30,1425)  ! Fuel row 9
     c              FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c              FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c              FRUNIV(COLUMN,ROW), ASSYUNIV(COLUMN,ROW)
 1424               FORMAT(T11,I3,' 1R ',I3,' 1R ',I4,1X,
     c              I3,' 1R ',I4,1X,I3,' 1R ',I4)
 1425               FORMAT(T11,I3,' 1R ',
     c              I4,1X,I3,' 1R ',I4,1X,I3,' 1R ',I3,' 1R ')
                    WRITE(30,1426) ASSYUNIV(COLUMN,ROW), ! Fuel row 10
     c              FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1426               FORMAT(T11,I3,' 1R ',I3,' 16R ',I3,' 1R')
                    WRITE(30,1428) ASSYUNIV(COLUMN,ROW), ! Fuel row 11
     c              FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1428               FORMAT(T11,I3,' 1R ',I3,' 16R ',I3,' 1R')
                    WRITE(30,1430) ASSYUNIV(COLUMN,ROW), ! Fuel row 12
     c              FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW)
                    WRITE(30,1431) FRUNIV(COLUMN,ROW), ! Fuel row 12
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
 1430               FORMAT(T11,I3,' 1R ',I3,' 1R ',I4,1X,I3,
     c              ' 1R ',I4,1X,I3,' 1R ',I4)
```

```
1431            FORMAT(T11,I3,' 1R ',
     c          I4,1X,I3,' 1R ',I4,1X,I3,' 1R ',I3,' 1R ')
                WRITE(30,1432) ASSYUNIV(COLUMN,ROW), ! Fuel row 13
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
1432            FORMAT(T11,I3,' 1R ',I3,' 16R ',I3,' 1R')
                WRITE(30,1434) ASSYUNIV(COLUMN,ROW), ! Fuel row 14
     c          FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
1434            FORMAT(T11,I3,' 1R ',I3,' 2R ',I4,1X,I3,
     c          ' 8R ',I4,1X,I3,' 2R ',I3,' 1R ')
                WRITE(30,1436) ASSYUNIV(COLUMN,ROW), ! Fuel row 15
     c          FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
1436            FORMAT(T11,I3,' 1R ',I3,' 4R ',I4,1X,I3,
     c          ' 1R ',I4,1X,I3,' 1R ',I4,1X,I3,' 4R ',I3,' 1R ')
             ELSEIF (BPRAINSERTED.EQ..TRUE.) THEN
             IF (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.1) THEN ! 4 BPR assembly
                WRITE(30,1412) ASSYUNIV(COLUMN,ROW), ! Fuel row 3
     c          FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1414) ASSYUNIV(COLUMN,ROW), ! Fuel row 4
     c          FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1416) ASSYUNIV(COLUMN,ROW), ! Fuel row 5
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1418) ASSYUNIV(COLUMN,ROW), ! Fuel row 6
     c          FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW)
                WRITE(30,1419) FRUNIV(COLUMN,ROW), ! Fuel row 6
     c          BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1420) ASSYUNIV(COLUMN,ROW), ! Fuel row 7
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1422) ASSYUNIV(COLUMN,ROW), ! Fuel row 8
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1424) ASSYUNIV(COLUMN,ROW), ! Fuel row 9
     c          FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
```

```
     c          FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), ITUNIV(COLUMN,ROW)
                WRITE(30,1425) ! Fuel row 9
     c          FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), ASSYUNIV(COLUMN,ROW)
                WRITE(30,1426) ASSYUNIV(COLUMN,ROW), ! Fuel row 10
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1428) ASSYUNIV(COLUMN,ROW), ! Fuel row 11
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1430) ASSYUNIV(COLUMN,ROW), ! Fuel row 12
     c          FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW)
                WRITE(30,1431) ! Fuel row 12
     c          FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), ASSYUNIV(COLUMN,ROW)
                WRITE(30,1432) ASSYUNIV(COLUMN,ROW), ! Fuel row 13
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1434) ASSYUNIV(COLUMN,ROW), ! Fuel row 14
     c          FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1436) ASSYUNIV(COLUMN,ROW), ! Fuel row 15
     c          FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
             ELSEIF (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.2) THEN ! 8 BPR
assembly
                WRITE(30,1412) ASSYUNIV(COLUMN,ROW), ! Fuel row 3
     c          FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1414) ASSYUNIV(COLUMN,ROW), ! Fuel row 4
     c          FRUNIV(COLUMN,ROW),
     c          BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1416) ASSYUNIV(COLUMN,ROW), ! Fuel row 5
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1418) ASSYUNIV(COLUMN,ROW), ! Fuel row 6
     c          FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
```

```
     c          BPRAUNIV(COLUMN,ROW)
                WRITE(30,1419) FRUNIV(COLUMN,ROW), ! Fuel row 6
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1420) ASSYUNIV(COLUMN,ROW), ! Fuel row 7
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1422) ASSYUNIV(COLUMN,ROW), ! Fuel row 8
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1424) ASSYUNIV(COLUMN,ROW), ! Fuel row 9
     c          FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), ITUNIV(COLUMN,ROW)
                WRITE(30,1425) ! Fuel row 9
     c          FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), ASSYUNIV(COLUMN,ROW)
                WRITE(30,1426) ASSYUNIV(COLUMN,ROW), ! Fuel row 10
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1428) ASSYUNIV(COLUMN,ROW), ! Fuel row 11
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1430) ASSYUNIV(COLUMN,ROW), ! Fuel row 12
     c          FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW)
                WRITE(30,1431) ! Fuel row 12
     c          FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), ASSYUNIV(COLUMN,ROW)
                WRITE(30,1432) ASSYUNIV(COLUMN,ROW), ! Fuel row 13
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1434) ASSYUNIV(COLUMN,ROW), ! Fuel row 14
     c          FRUNIV(COLUMN,ROW),
     c          BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1436) ASSYUNIV(COLUMN,ROW), ! Fuel row 15
     c          FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
            ELSEIF (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.3) THEN ! 9 BPR
assembly
                WRITE(30,1412) ASSYUNIV(COLUMN,ROW), ! Fuel row 3
     c          FRUNIV(COLUMN,ROW),
     c          BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
```

```
c         ASSYUNIV(COLUMN,ROW)
          WRITE(30,1414) ASSYUNIV(COLUMN,ROW), ! Fuel row 4
c         FRUNIV(COLUMN,ROW),
c         BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
c         GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
c         ASSYUNIV(COLUMN,ROW)
          WRITE(30,1416) ASSYUNIV(COLUMN,ROW), ! Fuel row 5
c         FRUNIV(COLUMN,ROW),
c         ASSYUNIV(COLUMN,ROW)
          WRITE(30,1418) ASSYUNIV(COLUMN,ROW), ! Fuel row 6
c         FRUNIV(COLUMN,ROW),
c         BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
c         GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
c         BPRAUNIV(COLUMN,ROW)
          WRITE(30,1419) FRUNIV(COLUMN,ROW), ! Fuel row 6
c         GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
c         GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
c         ASSYUNIV(COLUMN,ROW)
          WRITE(30,1420) ASSYUNIV(COLUMN,ROW), ! Fuel row 7
c         FRUNIV(COLUMN,ROW),
c         ASSYUNIV(COLUMN,ROW)
          WRITE(30,1422) ASSYUNIV(COLUMN,ROW), ! Fuel row 8
c         FRUNIV(COLUMN,ROW),
c         ASSYUNIV(COLUMN,ROW)
          WRITE(30,1424) ASSYUNIV(COLUMN,ROW), ! Fuel row 9
c         FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW),
c         FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW),
c         FRUNIV(COLUMN,ROW), ITUNIV(COLUMN,ROW)
          WRITE(30,1425) ! Fuel row 9
c         FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
c         FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
c         FRUNIV(COLUMN,ROW), ASSYUNIV(COLUMN,ROW)
          WRITE(30,1426) ASSYUNIV(COLUMN,ROW), ! Fuel row 10
c         FRUNIV(COLUMN,ROW),
c         ASSYUNIV(COLUMN,ROW)
          WRITE(30,1428) ASSYUNIV(COLUMN,ROW), ! Fuel row 11
c         FRUNIV(COLUMN,ROW),
c         ASSYUNIV(COLUMN,ROW)
          WRITE(30,1430) ASSYUNIV(COLUMN,ROW), ! Fuel row 12
c         FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW),
c         FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
c         FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW).
          WRITE(30,1431) ! Fuel row 12
c         FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
c         FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
c         FRUNIV(COLUMN,ROW), ASSYUNIV(COLUMN,ROW)
          WRITE(30,1432) ASSYUNIV(COLUMN,ROW), ! Fuel row 13
c         FRUNIV(COLUMN,ROW),
c         ASSYUNIV(COLUMN,ROW)
          WRITE(30,1434) ASSYUNIV(COLUMN,ROW), ! Fuel row 14
c         FRUNIV(COLUMN,ROW),
c         GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
c         GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
c         ASSYUNIV(COLUMN,ROW)
```

```
                WRITE(30,1436) ASSYUNIV(COLUMN,ROW), ! Fuel row 15
     c          FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
             ELSEIF (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.4) THEN ! 10 BPR
assembly
                WRITE(30,1412) ASSYUNIV(COLUMN,ROW), ! Fuel row 3
     c          FRUNIV(COLUMN,ROW),
     c          BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1414) ASSYUNIV(COLUMN,ROW), ! Fuel row 4
     c          FRUNIV(COLUMN,ROW),
     c          BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1416) ASSYUNIV(COLUMN,ROW), ! Fuel row 5
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1418) ASSYUNIV(COLUMN,ROW), ! Fuel row 6
     c          FRUNIV(COLUMN,ROW),
     c          BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW)
                WRITE(30,1419) FRUNIV(COLUMN,ROW),! Fuel row 6
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1420) ASSYUNIV(COLUMN,ROW), ! Fuel row 7
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1422) ASSYUNIV(COLUMN,ROW), ! Fuel row 8
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1424) ASSYUNIV(COLUMN,ROW), ! Fuel row 9
     c          FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), ITUNIV(COLUMN,ROW)
                WRITE(30,1425) ! Fuel row 9
     c          FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), ASSYUNIV(COLUMN,ROW)
                WRITE(30,1426) ASSYUNIV(COLUMN,ROW), ! Fuel row 10
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1428) ASSYUNIV(COLUMN,ROW), ! Fuel row 11
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1430) ASSYUNIV(COLUMN,ROW), ! Fuel row 12
     c          FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
```

```
     c            FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW)
                  WRITE(30,1431) ! Fuel row 12
     c            FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c            FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c            FRUNIV(COLUMN,ROW), ASSYUNIV(COLUMN,ROW)
                  WRITE(30,1432) ASSYUNIV(COLUMN,ROW), ! Fuel row 13
     c            FRUNIV(COLUMN,ROW),
     c            ASSYUNIV(COLUMN,ROW)
                  WRITE(30,1434) ASSYUNIV(COLUMN,ROW), ! Fuel row 14
     c            FRUNIV(COLUMN,ROW),
     c            BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            ASSYUNIV(COLUMN,ROW)
                  WRITE(30,1436) ASSYUNIV(COLUMN,ROW), ! Fuel row 15
     c            FRUNIV(COLUMN,ROW),
     c            BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            ASSYUNIV(COLUMN,ROW)
               ELSEIF (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.5) THEN ! 12 BPR
assembly
                  WRITE(30,1412) ASSYUNIV(COLUMN,ROW), ! Fuel row 3
     c            FRUNIV(COLUMN,ROW),
     c            BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            ASSYUNIV(COLUMN,ROW)
                  WRITE(30,1414) ASSYUNIV(COLUMN,ROW), ! Fuel row 4
     c            FRUNIV(COLUMN,ROW),
     c            GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            ASSYUNIV(COLUMN,ROW)
                  WRITE(30,1416) ASSYUNIV(COLUMN,ROW), ! Fuel row 5
     c            FRUNIV(COLUMN,ROW),
     c            ASSYUNIV(COLUMN,ROW)
                  WRITE(30,1418) ASSYUNIV(COLUMN,ROW), ! Fuel row 6
     c            FRUNIV(COLUMN,ROW),
     c            BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            BPRAUNIV(COLUMN,ROW)
                  WRITE(30,1419) FRUNIV(COLUMN,ROW), ! Fuel row 6
     c            GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            ASSYUNIV(COLUMN,ROW)
                  WRITE(30,1420) ASSYUNIV(COLUMN,ROW), ! Fuel row 7
     c            FRUNIV(COLUMN,ROW),
     c            ASSYUNIV(COLUMN,ROW)
                  WRITE(30,1422) ASSYUNIV(COLUMN,ROW), ! Fuel row 8
     c            FRUNIV(COLUMN,ROW),
     c            ASSYUNIV(COLUMN,ROW)
                  WRITE(30,1424) ASSYUNIV(COLUMN,ROW), ! Fuel row 9
     c            FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c            FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW),
     c            FRUNIV(COLUMN,ROW), ITUNIV(COLUMN,ROW)
```

```
                  WRITE(30,1425) ! Fuel row 9
     c            FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW),
     c            FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c            FRUNIV(COLUMN,ROW), ASSYUNIV(COLUMN,ROW)
                  WRITE(30,1426) ASSYUNIV(COLUMN,ROW), ! Fuel row 10
     c            FRUNIV(COLUMN,ROW),
     c            ASSYUNIV(COLUMN,ROW)
                  WRITE(30,1428) ASSYUNIV(COLUMN,ROW), ! Fuel row 11
     c            FRUNIV(COLUMN,ROW),
     c            ASSYUNIV(COLUMN,ROW)
                  WRITE(30,1430) ASSYUNIV(COLUMN,ROW), ! Fuel row 12
     c            FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW),
     c            FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c            FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW)
                  WRITE(30,1431) ! Fuel row 12
     c            FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c            FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW),
     c            FRUNIV(COLUMN,ROW), ASSYUNIV(COLUMN,ROW)
                  WRITE(30,1432) ASSYUNIV(COLUMN,ROW), ! Fuel row 13
     c            FRUNIV(COLUMN,ROW),
     c            ASSYUNIV(COLUMN,ROW)
                  WRITE(30,1434) ASSYUNIV(COLUMN,ROW), ! Fuel row 14
     c            FRUNIV(COLUMN,ROW),
     c            GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            ASSYUNIV(COLUMN,ROW)
                  WRITE(30,1436) ASSYUNIV(COLUMN,ROW), ! Fuel row 15
     c            FRUNIV(COLUMN,ROW),
     c            BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            ASSYUNIV(COLUMN,ROW)
               ELSEIF (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.6) THEN ! 16 BPR
assembly
                  WRITE(30,1412) ASSYUNIV(COLUMN,ROW), ! Fuel row 3
     c            FRUNIV(COLUMN,ROW),
     c            BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            ASSYUNIV(COLUMN,ROW)
                  WRITE(30,1414) ASSYUNIV(COLUMN,ROW), ! Fuel row 4
     c            FRUNIV(COLUMN,ROW),
     c            BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            ASSYUNIV(COLUMN,ROW)
                  WRITE(30,1416) ASSYUNIV(COLUMN,ROW), ! Fuel row 5
     c            FRUNIV(COLUMN,ROW),
     c            ASSYUNIV(COLUMN,ROW)
                  WRITE(30,1418) ASSYUNIV(COLUMN,ROW), ! Fuel row 6
     c            FRUNIV(COLUMN,ROW),
     c            BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c            BPRAUNIV(COLUMN,ROW)
                  WRITE(30,1419) FRUNIV(COLUMN,ROW), ! Fuel row 6
```

```
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1420) ASSYUNIV(COLUMN,ROW), ! Fuel row 7
     c              FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1422) ASSYUNIV(COLUMN,ROW), ! Fuel row 8
     c              FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1424) ASSYUNIV(COLUMN,ROW), ! Fuel row 9
     c              FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c              FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW),
     c              FRUNIV(COLUMN,ROW), ITUNIV(COLUMN,ROW)
                    WRITE(30,1425) ! Fuel row 9
     c              FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW),
     c              FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c              FRUNIV(COLUMN,ROW), ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1426) ASSYUNIV(COLUMN,ROW), ! Fuel row 10
     c              FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1428) ASSYUNIV(COLUMN,ROW), ! Fuel row 11
     c              FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1430) ASSYUNIV(COLUMN,ROW), ! Fuel row 12
     c              FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW),
     c              FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c              FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW)
                    WRITE(30,1431) ! Fuel row 12
     c              FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c              FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW),
     c              FRUNIV(COLUMN,ROW), ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1432) ASSYUNIV(COLUMN,ROW), ! Fuel row 13
     c              FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1434) ASSYUNIV(COLUMN,ROW), ! Fuel row 14
     c              FRUNIV(COLUMN,ROW),
     c              BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1436) ASSYUNIV(COLUMN,ROW), ! Fuel row 15
     c              FRUNIV(COLUMN,ROW),
     c              BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
                ELSEIF (WBPRATYPE(BANKNUM(COLUMN,ROW)).EQ.7) THEN
                    WRITE(30,1412) ASSYUNIV(COLUMN,ROW), ! Fuel row 3
     c              FRUNIV(COLUMN,ROW),
     c              BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1414) ASSYUNIV(COLUMN,ROW), ! Fuel row 4
     c              FRUNIV(COLUMN,ROW),
```

```
     c              BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1416) ASSYUNIV(COLUMN,ROW), ! Fuel row 5
     c              FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1418) ASSYUNIV(COLUMN,ROW), ! Fuel row 6
     c              FRUNIV(COLUMN,ROW),
     c              BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              BPRAUNIV(COLUMN,ROW)
                    WRITE(30,1419) FRUNIV(COLUMN,ROW), ! Fuel row 6
     c              GTUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1420) ASSYUNIV(COLUMN,ROW), ! Fuel row 7
     c              FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1422) ASSYUNIV(COLUMN,ROW), ! Fuel row 8
     c              FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1424) ASSYUNIV(COLUMN,ROW), ! Fuel row 9
     c              FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW),
     c              FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW),
     c              FRUNIV(COLUMN,ROW), ITUNIV(COLUMN,ROW)
                    WRITE(30,1425) ! Fuel row 9
     c              FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW),
     c              FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW),
     c              FRUNIV(COLUMN,ROW), ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1426) ASSYUNIV(COLUMN,ROW), ! Fuel row 10
     c              FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1428) ASSYUNIV(COLUMN,ROW), ! Fuel row 11
     c              FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1430) ASSYUNIV(COLUMN,ROW), ! Fuel row 12
     c              FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW),
     c              FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c              FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW)
                    WRITE(30,1431) ! Fuel row 12
     c              FRUNIV(COLUMN,ROW), GTUNIV(COLUMN,ROW),
     c              FRUNIV(COLUMN,ROW), BPRAUNIV(COLUMN,ROW),
     c              FRUNIV(COLUMN,ROW), ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1432) ASSYUNIV(COLUMN,ROW), ! Fuel row 13
     c              FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1434) ASSYUNIV(COLUMN,ROW), ! Fuel row 14
     c              FRUNIV(COLUMN,ROW),
     c              BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1436) ASSYUNIV(COLUMN,ROW), ! Fuel row 15
     c              FRUNIV(COLUMN,ROW),
     c              BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
```

```
     c          BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          BPRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
             ENDIF
             ELSEIF (CRAINSERTED.EQ..TRUE.) THEN
                WRITE(30,1412) ASSYUNIV(COLUMN,ROW), ! Fuel row 3
     c          FRUNIV(COLUMN,ROW),
     c          CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1414) ASSYUNIV(COLUMN,ROW), ! Fuel row 4
     c          FRUNIV(COLUMN,ROW),
     c          CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1416) ASSYUNIV(COLUMN,ROW), ! Fuel row 5
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1418) ASSYUNIV(COLUMN,ROW), ! Fuel row 6
     c          FRUNIV(COLUMN,ROW),
     c          CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          CRAUNIV(COLUMN,ROW)
                WRITE(30,1419) FRUNIV(COLUMN,ROW), ! Fuel row 6
     c          CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1420) ASSYUNIV(COLUMN,ROW), ! Fuel row 7
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1422) ASSYUNIV(COLUMN,ROW), ! Fuel row 8
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1424) ASSYUNIV(COLUMN,ROW), ! Fuel row 9
     c          FRUNIV(COLUMN,ROW), CRAUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), CRAUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), ITUNIV(COLUMN,ROW)
                WRITE(30,1425) ! Fuel row 9
     c          FRUNIV(COLUMN,ROW), CRAUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), CRAUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), ASSYUNIV(COLUMN,ROW)
                WRITE(30,1426) ASSYUNIV(COLUMN,ROW), ! Fuel row 10
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1428) ASSYUNIV(COLUMN,ROW), ! Fuel row 11
     c          FRUNIV(COLUMN,ROW),
     c          ASSYUNIV(COLUMN,ROW)
                WRITE(30,1430) ASSYUNIV(COLUMN,ROW), ! Fuel row 12
     c          FRUNIV(COLUMN,ROW), CRAUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), CRAUNIV(COLUMN,ROW),
     c          FRUNIV(COLUMN,ROW), CRAUNIV(COLUMN,ROW)
                WRITE(30,1431) ! Fuel row 12
     c          FRUNIV(COLUMN,ROW), CRAUNIV(COLUMN,ROW),
```

```
c          FRUNIV(COLUMN,ROW), CRAUNIV(COLUMN,ROW),
c          FRUNIV(COLUMN,ROW), ASSYUNIV(COLUMN,ROW)
           WRITE(30,1432) ASSYUNIV(COLUMN,ROW), ! Fuel row 13
c          FRUNIV(COLUMN,ROW),
c          ASSYUNIV(COLUMN,ROW)
           WRITE(30,1434) ASSYUNIV(COLUMN,ROW), ! Fuel row 14
c          FRUNIV(COLUMN,ROW),
c          CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
c          CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
c          ASSYUNIV(COLUMN,ROW)
           WRITE(30,1436) ASSYUNIV(COLUMN,ROW), ! Fuel row 15
c          FRUNIV(COLUMN,ROW),
c          CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
c          CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
c          CRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
c          ASSYUNIV(COLUMN,ROW)
         ELSEIF (APSRAINSERTED.EQ..TRUE.) THEN
           WRITE(30,1412) ASSYUNIV(COLUMN,ROW), ! Fuel row 3
c          FRUNIV(COLUMN,ROW),
c          APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
c          APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
c          APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
c          ASSYUNIV(COLUMN,ROW)
           WRITE(30,1414) ASSYUNIV(COLUMN,ROW), ! Fuel row 4
c          FRUNIV(COLUMN,ROW),
c          APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
c          APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
c          ASSYUNIV(COLUMN,ROW)
           WRITE(30,1416) ASSYUNIV(COLUMN,ROW), ! Fuel row 5
c          FRUNIV(COLUMN,ROW),
c          ASSYUNIV(COLUMN,ROW)
           WRITE(30,1418) ASSYUNIV(COLUMN,ROW), ! Fuel row 6
c          FRUNIV(COLUMN,ROW),
c          APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
c          APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
c          APSRAUNIV(COLUMN,ROW)
           WRITE(30,1419) FRUNIV(COLUMN,ROW), ! Fuel row 6
c          APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
c          APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
c          ASSYUNIV(COLUMN,ROW)
           WRITE(30,1420) ASSYUNIV(COLUMN,ROW), ! Fuel row 7
c          FRUNIV(COLUMN,ROW),
c          ASSYUNIV(COLUMN,ROW)
           WRITE(30,1422) ASSYUNIV(COLUMN,ROW), ! Fuel row 8
c          FRUNIV(COLUMN,ROW),
c          ASSYUNIV(COLUMN,ROW)
           WRITE(30,1424) ASSYUNIV(COLUMN,ROW), ! Fuel row 9
c          FRUNIV(COLUMN,ROW), APSRAUNIV(COLUMN,ROW),
c          FRUNIV(COLUMN,ROW), APSRAUNIV(COLUMN,ROW),
c          FRUNIV(COLUMN,ROW), ITUNIV(COLUMN,ROW)
           WRITE(30,1425) ! Fuel row 9
c          FRUNIV(COLUMN,ROW), APSRAUNIV(COLUMN,ROW),
c          FRUNIV(COLUMN,ROW), APSRAUNIV(COLUMN,ROW),
c          FRUNIV(COLUMN,ROW), ASSYUNIV(COLUMN,ROW)
```

```
                    WRITE(30,1426) ASSYUNIV(COLUMN,ROW), ! Fuel row 10
     c              FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1428) ASSYUNIV(COLUMN,ROW), ! Fuel row 11
     c              FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1430) ASSYUNIV(COLUMN,ROW), ! Fuel row 12
     c              FRUNIV(COLUMN,ROW), APSRAUNIV(COLUMN,ROW),
     c              FRUNIV(COLUMN,ROW), APSRAUNIV(COLUMN,ROW),
     c              FRUNIV(COLUMN,ROW), APSRAUNIV(COLUMN,ROW)
                    WRITE(30,1431) ! Fuel row 12
     c              FRUNIV(COLUMN,ROW), APSRAUNIV(COLUMN,ROW),
     c              FRUNIV(COLUMN,ROW), APSRAUNIV(COLUMN,ROW),
     c              FRUNIV(COLUMN,ROW), ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1432) ASSYUNIV(COLUMN,ROW), ! Fuel row 13
     c              FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1434) ASSYUNIV(COLUMN,ROW), ! Fuel row 14
     c              FRUNIV(COLUMN,ROW),
     c              APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
                    WRITE(30,1436) ASSYUNIV(COLUMN,ROW), ! Fuel row 15
     c              FRUNIV(COLUMN,ROW),
     c              APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              APSRAUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW),
     c              ASSYUNIV(COLUMN,ROW)
                 ENDIF
                 WRITE(30,1438) ASSYUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW), !
Fuel row 16
     c           ASSYUNIV(COLUMN,ROW)
 1438            FORMAT(T11,I3,' 1R ',I3,' 16R ',I3,' 1R')
                 WRITE(30,1440) ASSYUNIV(COLUMN,ROW), FRUNIV(COLUMN,ROW), !
Fuel row 17
     c           ASSYUNIV(COLUMN,ROW)
 1440            FORMAT(T11,I3,' 1R ',I3,' 16R ',I3,' 1R')
                 WRITE(30,1442) ASSYUNIV(COLUMN,ROW) ! Second-to-last framing
row
 1442            FORMAT(T11,I3,' 20R')
                 WRITE(30,1444) ASSYUNIV(COLUMN,ROW) ! Last framing row
 1444            FORMAT(T11,I3,' 20R')
*
             ELSEIF (CE.EQ..TRUE.) THEN
*
             WRITE(30,*) 'THE CE MODELING OPTION IS NOT AVAILABLE.'
*
             ENDIF
             ENDIF
          ENDIF
 1890     CONTINUE
 1900 CONTINUE
*
*
```

```
      FMN=6000
      BPMN=3000
      BPRAFLAG=.FALSE.
      CRAFLAG=.FALSE.
      APSRAFLAG=.FALSE.
      PLAINFLAG=.FALSE.
*  Write the specifications for the fuel rod universes that are
*  required to fill the assembly layout specification previously defined.
      DO 2845 ROW=1,50
         DO 2844 COLUMN=1,50
*     Write the fuel rod universe specification for the assembly if it
*     contains a unique fuel material and upper region combination.
*
      IF (FUNIQUE(COLUMN,ROW).EQ..TRUE.) THEN
*  Write the fuel rod specification header.
         WRITE(30,1911)
 1911    FORMAT(T1,'C')
         WRITE(30,1912) ASSYID(COLUMN,ROW)
 1912    FORMAT(T1,
     c   'C   FUEL ROD UNIVERSE SPECIFICATION FOR ASSEMBLY ',A5)
         WRITE(30,1913)
 1913    FORMAT(T1,'C')
*     Determine what upper region specification is to be used
*     (i.e. base fuel assembly, BPRA, CRA, APSRA).
*     Determine whether or not this assembly contains a BPRA, CRA, or APSRA.
            BPRAINSERTED=.FALSE.
            CRAINSERTED=.FALSE.
            APSRAINSERTED=.FALSE.
            IF (BANKNUM(COLUMN,ROW).NE.0) THEN
               IF (BANKDES(BANKNUM(COLUMN,ROW)).EQ.'BPRA ') THEN
                  BPRAINSERTED=.TRUE.
               ELSEIF (BANKDES(BANKNUM(COLUMN,ROW)).EQ.'CRA  ') THEN
                  CRAINSERTED=.TRUE.
               ELSEIF (BANKDES(BANKNUM(COLUMN,ROW)).EQ.'APSRA') THEN
                  APSRAINSERTED=.TRUE.
               ENDIF
            ENDIF
          DO 1920 MCNPNODE=1,NUMOFMCNPFUELNODES(DESNUM(COLUMN,ROW))
*     Define the fuel node bounding surfaces.
            IF (MCNPNODE.EQ.1) THEN
               TOTFUELHEIGHT=0.0
               DO 1914 Z=1,NUMOFMCNPFUELNODES(DESNUM(COLUMN,ROW))
                  TOTFUELHEIGHT=TOTFUELHEIGHT+
     c            MCNPFUELHEIGHT(DESNUM(COLUMN,ROW),Z)
 1914          CONTINUE
               CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c         TOTFUELHEIGHT
               CURRENTSURFLABEL=0
               DO 1915 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
```

```
            ENDIF
 1915     CONTINUE
          IF (CURRENTSURFLABEL.EQ.0) THEN
            TOPNODETOPSURF=SN
            SURFTYPESPEC(SN)='PZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
          ELSE
            TOPNODETOPSURF=CURRENTSURFLABEL
          ENDIF
          NODETOPSURF=TOPNODETOPSURF
          CURRENTSURF=SURFVALUESPEC(NODETOPSURF)-
     c    MCNPFUELHEIGHT(DESNUM(COLUMN,ROW),MCNPNODE)
          CURRENTSURFLABEL=0
          DO 1916 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'PZ') THEN
        IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                CURRENTSURFLABEL=V
                EXIT
              ENDIF
            ENDIF
 1916     CONTINUE
          IF (CURRENTSURFLABEL.EQ.0) THEN
            NODEBOTTOMSURF=SN
            SURFTYPESPEC(SN)='PZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
          ELSE
            NODEBOTTOMSURF=CURRENTSURFLABEL
          ENDIF
          CURRENTSURF=FUELRADIUS(DESNUM(COLUMN,ROW))
          CURRENTSURFLABEL=0
          DO 1917 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'CZ') THEN
        IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                CURRENTSURFLABEL=V
                EXIT
              ENDIF
            ENDIF
 1917     CONTINUE
          IF (CURRENTSURFLABEL.EQ.0) THEN
            RADIUS=SN
            SURFTYPESPEC(SN)='CZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
          ELSE
            RADIUS=CURRENTSURFLABEL
          ENDIF
        ELSEIF (MCNPNODE.NE.1) THEN
          NODETOPSURF=NODEBOTTOMSURF
          CURRENTSURF=SURFVALUESPEC(NODETOPSURF)-
     c    MCNPFUELHEIGHT(DESNUM(COLUMN,ROW),MCNPNODE)
          CURRENTSURFLABEL=0
          DO 1918 V=1,(SN-1)
```

```
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                           CURRENTSURFLABEL=V
                           EXIT
                        ENDIF
                     ENDIF
 1918             CONTINUE
                  IF (CURRENTSURFLABEL.EQ.0) THEN
                     NODEBOTTOMSURF=SN
                     SURFTYPESPEC(SN)='PZ'
                     SURFVALUESPEC(SN)=CURRENTSURF
                     SN=SN+1
                  ELSE
                     NODEBOTTOMSURF=CURRENTSURFLABEL
                  ENDIF
               ENDIF
               FUELNODEML=FMN
               FMN=FMN+1
*     Write the fuel node cells in this fuel rod universe.
               WRITE(30,1919) LN, FUELNODEML,
     c         (-1*FUELNODEDEN(COLUMN,ROW,MCNPNODE)), (-1*RADIUS),
     c         (-1*NODETOPSURF), NODEBOTTOMSURF, FRUNIV(COLUMN,ROW),
     c         MCNPNODE
 1919          FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Fuel node ',I2)
               LN=LN+1
 1920       CONTINUE
*     Define the fuel rod cladding inner radius.
                  CURRENTSURF=CLADRADIUS(DESNUM(COLUMN,ROW),1)
                  CURRENTSURFLABEL=0
                  DO 1921 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'CZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                           CURRENTSURFLABEL=V
                           EXIT
                        ENDIF
                     ENDIF
 1921             CONTINUE
                  IF (CURRENTSURFLABEL.EQ.0) THEN
                     CLADIRSURF=SN
                     SURFTYPESPEC(SN)='CZ'
                     SURFVALUESPEC(SN)=CURRENTSURF
                     SN=SN+1
                  ELSE
                     CLADIRSURF=CURRENTSURFLABEL
                  ENDIF
*     Define the fuel rod cladding outer radius.
                  CURRENTSURF=CLADRADIUS(DESNUM(COLUMN,ROW),2)
                  CURRENTSURFLABEL=0
                  DO 1922 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'CZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                           CURRENTSURFLABEL=V
                           EXIT
```

```
                  ENDIF
               ENDIF
 1922       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               CLADORSURF=SN
               SURFTYPESPEC(SN)='CZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               CLADORSURF=CURRENTSURFLABEL
            ENDIF
*     Define the fuel rod cladding top surface.
            CURRENTSURF=TOTFUELHEIGHT+
     c      ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c      ASSYPLENUM(DESNUM(COLUMN,ROW),1)-
     c      ENDCAPHEIGHT(DESNUM(COLUMN,ROW),1)
            CURRENTSURFLABEL=0
            DO 1923 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 1923       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               CLADTOPSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               CLADTOPSURF=CURRENTSURFLABEL
            ENDIF
*     Define the fuel rod cladding bottom surface.
            CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)-
     c      ASSYPLENUM(DESNUM(COLUMN,ROW),2)+
     c      ENDCAPHEIGHT(DESNUM(COLUMN,ROW),2)
            CURRENTSURFLABEL=0
            DO 1924 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 1924       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               CLADBOTTOMSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               CLADBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
```

```
*     Define the fuel rod upper plenum top surface.
                CURRENTSURF=TOTFUELHEIGHT+
     c          ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c          ASSYPLENUM(DESNUM(COLUMN,ROW),1)
                CURRENTSURFLABEL=0
                DO 1925 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 1925           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   PLENUMTOPSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   PLENUMTOPSURF=CURRENTSURFLABEL
                ENDIF
*     Define the fuel rod lower plenum bottom surface.
                CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)-
     c          ASSYPLENUM(DESNUM(COLUMN,ROW),2)
                CURRENTSURFLABEL=0
                DO 1926 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 1926           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   PLENUMBOTTOMSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   PLENUMBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
*     Define the upper end-fitting bottom surface.
                CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c          ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
                CURRENTSURFLABEL=0
                DO 1927 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 1927           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
```

```
                  UEFBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  UEFBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
*     Define the upper end-fitting top surface.
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),1)+
     c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 1928 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 1928          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  UEFTOPSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  UEFTOPSURF=CURRENTSURFLABEL
               ENDIF
*     Write the fuel-to-cladding gap cell in this fuel rod universe.
            WRITE(30,1929) LN, (-1*CLADIRSURF), RADIUS,
     c      (-1*TOPNODETOPSURF),
     c      NODEBOTTOMSURF, FRUNIV(COLUMN,ROW)
 1929       FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Fuel-to-cladding gap')
            LN=LN+1
*     Write the fuel cladding cell in this fuel rod universe.
*     Determine if the fuel rod cladding material specification has
*     previously been defined.  If it has been previously defined, determine
*     the cladding material specification label.
            CLADMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 1931 RO=1,(ROW-1)
                  DO 1930 CO=1,50
                     IF (DESNUM(CO,RO).NE.0) THEN
                        IF (CLADMATERIAL(DESNUM(COLUMN,ROW)).EQ.
     c                  CLADMATERIAL(DESNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CLADML(COLUMN,ROW)=CLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 1930             CONTINUE
```

```
            IF (LEAVE.EQ..TRUE.) THEN
               EXIT
            ENDIF
 1931     CONTINUE
          IF (LEAVE.EQ..FALSE.) THEN
            DO 1933 RO=ROW,ROW
              DO 1932 CO=1,(COLUMN-1)
                IF (DESNUM(CO,RO).NE.0) THEN
                  IF (CLADMATERIAL(DESNUM(COLUMN,ROW)).EQ.
     c            CLADMATERIAL(DESNUM(CO,RO))) THEN
                    CLADMLUNIQUE=.FALSE.
                    LEAVE=.TRUE.
                    CLADML(COLUMN,ROW)=CLADML(CO,RO)
                    EXIT
                  ENDIF
                ENDIF
 1932         CONTINUE
              IF (LEAVE.EQ..TRUE.) THEN
                EXIT
              ENDIF
 1933       CONTINUE
          ENDIF
        ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
          DO 1935 RO=1,(ROW-1)
            DO 1934 CO=1,50
              IF (DESNUM(CO,RO).NE.0) THEN
                IF (CLADMATERIAL(DESNUM(COLUMN,ROW)).EQ.
     c          CLADMATERIAL(DESNUM(CO,RO))) THEN
                  CLADMLUNIQUE=.FALSE.
                  LEAVE=.TRUE.
                  CLADML(COLUMN,ROW)=CLADML(CO,RO)
                  EXIT
                ENDIF
              ENDIF
 1934       CONTINUE
            IF (LEAVE.EQ..TRUE.) THEN
              EXIT
            ENDIF
 1935     CONTINUE
        ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
          DO 1937 RO=1,1
            DO 1936 CO=1,(COLUMN-1)
              IF (DESNUM(CO,RO).NE.0) THEN
                IF (CLADMATERIAL(DESNUM(COLUMN,ROW)).EQ.
     c          CLADMATERIAL(DESNUM(CO,RO))) THEN
                  CLADMLUNIQUE=.FALSE.
                  LEAVE=.TRUE.
                  CLADML(COLUMN,ROW)=CLADML(CO,RO)
                  EXIT
                ENDIF
              ENDIF
 1936       CONTINUE
            IF (LEAVE.EQ..TRUE.) THEN
              EXIT
```

```
                 ENDIF
 1937          CONTINUE
             ENDIF
             IF (CLADMLUNIQUE.EQ..TRUE.) THEN
               CLADML(COLUMN,ROW)=MN
*  Check Fuel Rod Cladding
                 IF (CLADMATERIAL(DESNUM(COLUMN,ROW)).EQ.1) THEN
                   DO 1943 C=1,2
                     IF (C.EQ.1) THEN
                       WRITE(200,9300) CLADML(COLUMN,ROW)
 9300                FORMAT(T1,'M',I4,T9,' 8016.50c      -0.120',
     c                '               $ Zirc-4 Cladding')
                     ELSEIF (C.EQ.2) THEN
                       WRITE(200,9301)
 9301                  FORMAT(T9,'24050.60c      -0.004')
                       WRITE(200,7000)
 7000                  FORMAT(T9,'24052.60c      -0.084')
                       WRITE(200,7001)
 7001                  FORMAT(T9,'24053.60c      -0.010')
                       WRITE(200,7002)
 7002                  FORMAT(T9,'24054.60c      -0.002')
                       WRITE(200,9302)
 9302                  FORMAT(T9,'26054.60c      -0.011')
                       WRITE(200,7003)
 7003                  FORMAT(T9,'26056.60c      -0.184')
                       WRITE(200,7004)
 7004                  FORMAT(T9,'26057.60c      -0.004')
                       WRITE(200,7005)
 7005                  FORMAT(T9,'26058.60c      -0.001')
                       WRITE(200,9303)
 9303                  FORMAT(T9,'40000.60c      -98.180')
                       WRITE(200,9304)
 9304                  FORMAT(T9,'50000.35c      -1.400')
                     ENDIF
 1943              CONTINUE
                 ELSEIF (CLADMATERIAL(DESNUM(COLUMN,ROW))
     c           .EQ.2) THEN
                   DO 1948 C=1,2
                     IF (C.EQ.1) THEN
                       WRITE(200,9305) CLADML(COLUMN,ROW)
 9305                FORMAT(T1,'M',I4,T9,'6000.50c       -0.080',
     c                '               $ SS304 Cladding')
                     ELSEIF (C.EQ.2) THEN
                       WRITE(200,9306)
 9306                  FORMAT(T9,'7014.50c       -0.100')
                       WRITE(200,9307)
 9307                  FORMAT(T9,'14000.50c      -0.750')
                       WRITE(200,9308)
 9308                  FORMAT(T9,'15031.50c      -0.045')
                       WRITE(200,9309)
 9309                  FORMAT(T9,'16032.50c      -0.030')
                       WRITE(200,9310)
 9310                  FORMAT(T9,'24050.60c      -0.793')
                       WRITE(200,7006)
```

```
7006          FORMAT(T9,'24052.60c      -15.903')
              WRITE(200,7007)
7007          FORMAT(T9,'24053.60c      -1.838')
              WRITE(200,7008)
7008          FORMAT(T9,'24054.60c      -0.466')
              WRITE(200,9311)
9311          FORMAT(T9,'25055.50c      -2.000')
              WRITE(200,9312)
9312          FORMAT(T9,'26054.60c      -3.918')
              WRITE(200,7009)
7009          FORMAT(T9,'26056.60c      -63.156')
              WRITE(200,7010)
7010          FORMAT(T9,'26057.60c      -1.472')
              WRITE(200,7011)
7011          FORMAT(T9,'26058.60c      -0.200')
              WRITE(200,9313)
9313          FORMAT(T9,'28058.60c      -6.234')
              WRITE(200,7012)
7012          FORMAT(T9,'28060.60c      -2.465')
              WRITE(200,7013)
7013          FORMAT(T9,'28061.60c      -0.109')
              WRITE(200,7014)
7014          FORMAT(T9,'28062.60c      -0.350')
              WRITE(200,7015)
7015          FORMAT(T9,'28064.60c      -0.092')
          ENDIF
1948      CONTINUE
        ELSEIF (CLADMATERIAL(DESNUM(COLUMN,ROW))
     c  .EQ.3) THEN
          DO 1954 C=1,2
            IF (C.EQ.1) THEN
              WRITE(200,9314) CLADML(COLUMN,ROW)
9314        FORMAT(T1,'M',I4,T9,'6000.50c      -0.080',
     c        '                 $ Inconel Cladding')
            ELSEIF (C.EQ.2) THEN
              WRITE(200,9315)
9315          FORMAT(T9,'14000.50c      -0.350')
              WRITE(200,9316)
9316          FORMAT(T9,'15031.50c      -0.015')
              WRITE(200,9317)
9317          FORMAT(T9,'16032.50c      -0.015')
              WRITE(200,9318)
9318          FORMAT(T9,'24050.60c      -0.793')
              WRITE(200,7016)
7016          FORMAT(T9,'24052.60c      -15.903')
              WRITE(200,7017)
7017          FORMAT(T9,'24053.60c      -1.838')
              WRITE(200,7018)
7018          FORMAT(T9,'24054.60c      -0.466')
              WRITE(200,9319)
9319          FORMAT(T9,'25055.50c      -0.350')
              WRITE(200,9320)
9320          FORMAT(T9,'26054.60c      -0.958')
              WRITE(200,7019)
```

```
7019                      FORMAT(T9,'26056.60c      -15.442')
                          WRITE(200,7020)
7020                      FORMAT(T9,'26057.60c      -0.360')
                          WRITE(200,7021)
7021                      FORMAT(T9,'26058.60c      -0.049')
                          WRITE(200,9321)
9321                      FORMAT(T9,'28058.60c      -35.382')
                          WRITE(200,7022)
7022                      FORMAT(T9,'28060.60c      -13.993')
                          WRITE(200,7023)
7023                      FORMAT(T9,'28061.60c      -0.616')
                          WRITE(200,7024)
7024                      FORMAT(T9,'28062.60c      -1.989')
                          WRITE(200,7025)
7025                      FORMAT(T9,'28064.60c      -0.520')
                          WRITE(200,9322)
9322                      FORMAT(T9,'5010.50c       -1.078E-3')
                          WRITE(200,9323)
9323                      FORMAT(T9,'5011.56c       -4.925E-3')
                          WRITE(200,9324)
9324                      FORMAT(T9,'13027.50c      -0.500')
                          WRITE(200,9325)
9325                      FORMAT(T9,'22000.50c      -0.900')
                          WRITE(200,9326)
9326                      FORMAT(T9,'27059.50c      -1.000')
                          WRITE(200,9327)
9327                      FORMAT(T9,'29063.60c      -0.205')
                          WRITE(200,7026)
7026                      FORMAT(T9,'29065.60c      -0.095')
                          WRITE(200,9328)
9328                      FORMAT(T9,'41093.50c      -2.563')
                          WRITE(200,9329)
9329                      FORMAT(T9,'42000.50c      -3.050')
                          WRITE(200,9330)
9330                      FORMAT(T9,'73181.50c      -2.563')
                        ENDIF
1954                  CONTINUE
                    ENDIF
                  MN=MN+1
                ENDIF
                IF (CLADMATERIAL(DESNUM(COLUMN,ROW)).EQ.1) THEN
                  CLADRHO=6.56
                ELSEIF (CLADMATERIAL(DESNUM(COLUMN,ROW)).EQ.2) THEN
                  CLADRHO=7.90
                ELSEIF (CLADMATERIAL(DESNUM(COLUMN,ROW)).EQ.3) THEN
                  CLADRHO=8.19
                ENDIF
                WRITE(30,1956) LN, CLADML(COLUMN,ROW), (-1*CLADRHO),
     c          CLADIRSURF,
     c          (-1*CLADORSURF), (-1*CLADTOPSURF), CLADBOTTOMSURF,
     c          FRUNIV(COLUMN,ROW)
1956            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Fuel rod cladding')
                LN=LN+1
```

```
*     Write the fuel rod upper plenum cell in this fuel rod universe.
*     Determine if the fuel rod upper plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the upper plenum material specification label.
            FRUPMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 2151 RO=1,(ROW-1)
                  DO 2150 CO=1,50
                     IF (DESNUM(CO,RO).NE.0) THEN
                        IF (DESNUM(COLUMN,ROW).EQ.DESNUM(CO,RO)) THEN
                           FRUPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           FRUPML(COLUMN,ROW)=FRUPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 2150             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 2151          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 2153 RO=ROW,ROW
                     DO 2152 CO=1,(COLUMN-1)
                        IF (DESNUM(CO,RO).NE.0) THEN
                           IF (DESNUM(COLUMN,ROW).EQ.
     c                     DESNUM(CO,RO)) THEN
                              FRUPMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              FRUPML(COLUMN,ROW)=FRUPML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 2152                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 2153             CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 2155 RO=1,(ROW-1)
                  DO 2154 CO=1,50
                     IF (DESNUM(CO,RO).NE.0) THEN
                        IF (DESNUM(COLUMN,ROW).EQ.
     c                     DESNUM(CO,RO)) THEN
                           FRUPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           FRUPML(COLUMN,ROW)=FRUPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 2154             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
```

```
                        EXIT
                     ENDIF
 2155             CONTINUE
               ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                  DO 2157 RO=1,1
                     DO 2156 CO=1,(COLUMN-1)
                        IF (DESNUM(CO,RO).NE.0) THEN
                           IF (DESNUM(COLUMN,ROW).EQ.
     c                        DESNUM(CO,RO)) THEN
                              FRUPMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              FRUPML(COLUMN,ROW)=FRUPML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 2156                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 2157             CONTINUE
               ENDIF
               IF (FRUPMLUNIQUE.EQ..TRUE.) THEN
                  FRUPML(COLUMN,ROW)=MN
*  Check Fuel Rod Upper Plenum Regions
                     DO 2160 C=1,FRUPLENMAT(DESNUM(COLUMN,ROW),2)
                        IF (C.EQ.1) THEN
                           WRITE(200,2158) FRUPML(COLUMN,ROW),
     c                     FRUPLENZAIDS(DESNUM(COLUMN,ROW),C),
     c                     (-1*FRUPLENWTS(DESNUM(COLUMN,ROW),C))
 2158                      FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                     '    $ Fuel Rod Upper Plenum')
                        ELSE
                           WRITE(200,2159)
     c                     FRUPLENZAIDS(DESNUM(COLUMN,ROW),C),
     c                     (-1*FRUPLENWTS(DESNUM(COLUMN,ROW),C))
 2159                      FORMAT(T9,A9,3X,G14.6)
                        ENDIF
 2160                CONTINUE
                  MN=MN+1
               ENDIF
               WRITE(30,2190) LN, FRUPML(COLUMN,ROW),
     c         (-1*FRUPLENMAT(DESNUM(COLUMN,ROW),1)), TOPNODETOPSURF,
     c         (-1*PLENUMTOPSURF), (-1*CLADIRSURF), FRUNIV(COLUMN,ROW)
 2190          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'    $ Fuel rod upper plenum')
               LN=LN+1
               WRITE(30,2200) LN, FRUPML(COLUMN,ROW),
     c         (-1*FRUPLENMAT(DESNUM(COLUMN,ROW),1)), CLADTOPSURF,
     c         (-1*PLENUMTOPSURF), CLADIRSURF, (-1*CLADORSURF),
     c         FRUNIV(COLUMN,ROW)
 2200          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'    $ Fuel rod upper plenum')
               LN=LN+1
*     Write the fuel rod lower plenum cell in this fuel rod universe.
```

```
*     Determine if the fuel rod lower plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the lower plenum material specification label.
            FRLPMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 2211 RO=1,(ROW-1)
                  DO 2210 CO=1,50
                     IF (DESNUM(CO,RO).NE.0) THEN
                        IF (DESNUM(COLUMN,ROW).EQ.DESNUM(CO,RO)) THEN
                           FRLPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           FRLPML(COLUMN,ROW)=FRLPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 2210             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 2211          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 2213 RO=ROW,ROW
                     DO 2212 CO=1,(COLUMN-1)
                        IF (DESNUM(CO,RO).NE.0) THEN
                           IF (DESNUM(COLUMN,ROW).EQ.
     c                     DESNUM(CO,RO)) THEN
                              FRLPMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              FRLPML(COLUMN,ROW)=FRLPML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 2212                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 2213             CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 2215 RO=1,(ROW-1)
                  DO 2214 CO=1,50
                     IF (DESNUM(CO,RO).NE.0) THEN
                        IF (DESNUM(COLUMN,ROW).EQ.
     c                  DESNUM(CO,RO)) THEN
                        FRLPMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        FRLPML(COLUMN,ROW)=FRLPML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
 2214          CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
```

```
                  ENDIF
 2215           CONTINUE
              ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                DO 2217 RO=1,1
                  DO 2216 CO=1,(COLUMN-1)
                    IF (DESNUM(CO,RO).NE.0) THEN
                      IF (DESNUM(COLUMN,ROW).EQ.
     c                  DESNUM(CO,RO)) THEN
                        FRLPMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        FRLPML(COLUMN,ROW)=FRLPML(CO,RO)
                        EXIT
                      ENDIF
                    ENDIF
 2216             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                  ENDIF
 2217           CONTINUE
              ENDIF
              IF (FRLPMLUNIQUE.EQ..TRUE.) THEN
                FRLPML(COLUMN,ROW)=MN
*  Check Fuel Rod Lower Plenum Regions
                  DO 2220 C=1,FRLPLENMAT(DESNUM(COLUMN,ROW),2)
                    IF (C.EQ.1) THEN
                      WRITE(200,2218) FRLPML(COLUMN,ROW),
     c                FRLPLENZAIDS(DESNUM(COLUMN,ROW),C),
     c                (-1*FRLPLENWTS(DESNUM(COLUMN,ROW),C))
 2218                 FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                '   $ Fuel Rod Lower Plenum')
                    ELSE
                      WRITE(200,2219)
     c                FRLPLENZAIDS(DESNUM(COLUMN,ROW),C),
     c                (-1*FRLPLENWTS(DESNUM(COLUMN,ROW),C))
 2219                 FORMAT(T9,A9,3X,G14.6)
                    ENDIF
 2220             CONTINUE
                MN=MN+1
              ENDIF
              WRITE(30,2250) LN, FRLPML(COLUMN,ROW),
     c        (-1*FRLPLENMAT(DESNUM(COLUMN,ROW),1)), PLENUMBOTTOMSURF,
     c        (-1*NODEBOTTOMSURF), (-1*CLADIRSURF), FRUNIV(COLUMN,ROW)
 2250         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Fuel rod lower plenum')
              LN=LN+1
              WRITE(30,2260) LN, FRLPML(COLUMN,ROW),
     c        (-1*FRLPLENMAT(DESNUM(COLUMN,ROW),1)), PLENUMBOTTOMSURF,
     c        (-1*CLADBOTTOMSURF), CLADIRSURF, (-1*CLADORSURF),
     c        FRUNIV(COLUMN,ROW)
 2260         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Fuel rod lower plenum')
              LN=LN+1
*     Write the lower end-fitting cell specification for this fuel rod
universe.
```

```
*     Determine if the fuel rod lower end-fitting material specification has
*     previously been defined.  If it has been previously defined, determine
*     the lower end-fitting material specification label.
            FRLEFMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 2271 RO=1,(ROW-1)
                  DO 2270 CO=1,50
                     IF (DESNUM(CO,RO).NE.0) THEN
                        IF (DESNUM(COLUMN,ROW).EQ.DESNUM(CO,RO)) THEN
                           FRLEFMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           FRLEFML(COLUMN,ROW)=FRLEFML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 2270             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 2271          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 2273 RO=ROW,ROW
                     DO 2272 CO=1,(COLUMN-1)
                        IF (DESNUM(CO,RO).NE.0) THEN
                           IF (DESNUM(COLUMN,ROW).EQ.
     c                     DESNUM(CO,RO)) THEN
                              FRLEFMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              FRLEFML(COLUMN,ROW)=FRLEFML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 2272                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 2273             CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 2275 RO=1,(ROW-1)
                  DO 2274 CO=1,50
                     IF (DESNUM(CO,RO).NE.0) THEN
                        IF (DESNUM(COLUMN,ROW).EQ.
     c                  DESNUM(CO,RO)) THEN
                           FRLEFMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           FRLEFML(COLUMN,ROW)=FRLEFML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 2274             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
```

```
              ENDIF
 2275       CONTINUE
         ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
            DO 2277 RO=1,1
               DO 2276 CO=1,(COLUMN-1)
                  IF (DESNUM(CO,RO).NE.0) THEN
                     IF (DESNUM(COLUMN,ROW).EQ.
     c                  DESNUM(CO,RO)) THEN
                        FRLEFMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        FRLEFML(COLUMN,ROW)=FRLEFML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
 2276          CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
 2277       CONTINUE
         ENDIF
         IF (FRLEFMLUNIQUE.EQ..TRUE.) THEN
            FRLEFML(COLUMN,ROW)=MN
*  Check Fuel Rod Lower End-Fitting Regions
               DO 2280 C=1,LEFMAT(DESNUM(COLUMN,ROW),2)
                  IF (C.EQ.1) THEN
                     WRITE(200,2278) FRLEFML(COLUMN,ROW),
     c               LEFZAIDS(DESNUM(COLUMN,ROW),C),
     c               (-1*LEFWTS(DESNUM(COLUMN,ROW),C))
 2278                FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c               '    $ Fuel Rod Lower End Fitting')
                  ELSE
                     WRITE(200,2279) LEFZAIDS(DESNUM(COLUMN,ROW),C),
     c               (-1*LEFWTS(DESNUM(COLUMN,ROW),C))
 2279                FORMAT(T9,A9,3X,G14.6)
                  ENDIF
 2280          CONTINUE
            WRITE(200,2281) FRLEFML(COLUMN,ROW)
 2281       FORMAT(T1,'MT',I4,T9,'LWTR.03T')
            MN=MN+1
         ENDIF
         IF ((SURFVALUESPEC(PLENUMBOTTOMSURF).GT.(0.0)).AND.
     c      (SURFVALUESPEC(PLENUMBOTTOMSURF).LT.
     c      SURFVALUESPEC(NODEBOTTOMSURF))) THEN
            WRITE(30,2285) LN, FRLEFML(COLUMN,ROW),
     c      (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*NODEBOTTOMSURF),
     c      CLADORSURF, FRUNIV(COLUMN,ROW)
 2285       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'    $ Assembly lower end-fitting')
            LN=LN+1
            WRITE(30,2290) LN, FRLEFML(COLUMN,ROW),
     c      (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*PLENUMBOTTOMSURF),
     c      (-1*CLADORSURF), FRUNIV(COLUMN,ROW)
 2290       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'    $ Assembly lower end-fitting')
```

```
              LN=LN+1
           ELSEIF (SURFVALUESPEC(PLENUMBOTTOMSURF).LT.(0.0)) THEN
              WRITE(30,2295) LN, FRLEFML(COLUMN,ROW),
     c        (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*NODEBOTTOMSURF),
     c        CLADORSURF, FRUNIV(COLUMN,ROW)
 2295         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Assembly lower end-fitting')
              LN=LN+1
           ELSEIF (SURFVALUESPEC(PLENUMBOTTOMSURF).GE.
     c        SURFVALUESPEC(NODEBOTTOMSURF)) THEN
              WRITE(30,2300) LN, FRLEFML(COLUMN,ROW),
     c        (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*NODEBOTTOMSURF),
     c        FRUNIV(COLUMN,ROW)
 2300         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
     c        ' IMP:N=1 U=',I3,'   $ Assembly lower end-fitting')
              LN=LN+1
           ENDIF
*     Write the upper end-fitting cell specification for this fuel rod
universe.
*     Determine if the fuel rod upper end-fitting material specification has
*     previously been defined.  If it has been previously defined, determine
*     the upper end-fitting material specification label.
           FRUEFMLUNIQUE=.TRUE.
           LEAVE=.FALSE.
           IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
              DO 2321 RO=1,(ROW-1)
                 DO 2320 CO=1,50
                    IF (DESNUM(CO,RO).NE.0) THEN
                       IF (DESNUM(COLUMN,ROW).EQ.DESNUM(CO,RO)) THEN
                          FRUEFMLUNIQUE=.FALSE..
                          LEAVE=.TRUE.
                          FRUEFML(COLUMN,ROW)=FRUEFML(CO,RO)
                          EXIT
                       ENDIF
                    ENDIF
 2320            CONTINUE
                 IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                 ENDIF
 2321         CONTINUE
              IF (LEAVE.EQ..FALSE.) THEN
                 DO 2323 RO=ROW,ROW
                    DO 2322 CO=1,(COLUMN-1)
                       IF (DESNUM(CO,RO).NE.0) THEN
                          IF (DESNUM(COLUMN,ROW).EQ.
     c                    DESNUM(CO,RO)) THEN
                             FRUEFMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             FRUEFML(COLUMN,ROW)=FRUEFML(CO,RO)
                             EXIT
                          ENDIF
                       ENDIF
 2322               CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
```

```
                        EXIT
                     ENDIF
 2323             CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 2325 RO=1,(ROW-1)
                  DO 2324 CO=1,50
                     IF (DESNUM(CO,RO).NE.0) THEN
                        IF (DESNUM(COLUMN,ROW).EQ.
     c                     DESNUM(CO,RO)) THEN
                           FRUEFMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           FRUEFML(COLUMN,ROW)=FRUEFML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 2324             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 2325          CONTINUE
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
               DO 2327 RO=1,1
                  DO 2326 CO=1,(COLUMN-1)
                     IF (DESNUM(CO,RO).NE.0) THEN
                        IF (DESNUM(COLUMN,ROW).EQ.
     c                     DESNUM(CO,RO)) THEN
                           FRUEFMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           FRUEFML(COLUMN,ROW)=FRUEFML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 2326             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 2327          CONTINUE
            ENDIF
            IF (FRUEFMLUNIQUE.EQ..TRUE.) THEN
               FRUEFML(COLUMN,ROW)=MN
*  Check Fuel Rod Upper End-Fitting Regions
                  DO 2330 C=1,UEFMAT(DESNUM(COLUMN,ROW),2)
                     IF (C.EQ.1) THEN
                        WRITE(200,2328) FRUEFML(COLUMN,ROW),
     c                  UEFZAIDS(DESNUM(COLUMN,ROW),C),
     c                  (-1*UEFWTS(DESNUM(COLUMN,ROW),C))
 2328                   FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                  '    $ Fuel Rod Upper End Fitting')
                     ELSE
                        WRITE(200,2329) UEFZAIDS(DESNUM(COLUMN,ROW),C),
     c                  (-1*UEFWTS(DESNUM(COLUMN,ROW),C))
 2329                   FORMAT(T9,A9,3X,G14.6)
                     ENDIF
```

```
2330        CONTINUE
          WRITE(200,2331) FRUEFML(COLUMN,ROW)
2331      FORMAT(T1,'MT',I4,T9,'LWTR.03T')
          MN=MN+1
        ENDIF
        IF (SURFVALUESPEC(PLENUMTOPSURF).LT.
     c  SURFVALUESPEC(UEFBOTTOMSURF)) THEN
          WRITE(30,2333) LN, FRUEFML(COLUMN,ROW),
     c    (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c    (-1*UEFTOPSURF), FRUNIV(COLUMN,ROW)
2333      FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c    ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
          LN=LN+1
        ELSEIF ((SURFVALUESPEC(PLENUMTOPSURF).GT.
     c  SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c  (SURFVALUESPEC(PLENUMTOPSURF).LT.
     c  SURFVALUESPEC(UEFTOPSURF))) THEN
          WRITE(30,2335) LN, FRUEFML(COLUMN,ROW),
     c    (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c    (-1*UEFTOPSURF), CLADORSURF, FRUNIV(COLUMN,ROW)
2335      FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c    ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
          LN=LN+1
          WRITE(30,2340) LN, FRUEFML(COLUMN,ROW),
     c    (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), PLENUMTOPSURF,
     c    (-1*UEFTOPSURF), (-1*CLADORSURF), FRUNIV(COLUMN,ROW)
2340      FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c    ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
          LN=LN+1
        ELSEIF (SURFVALUESPEC(PLENUMTOPSURF).EQ.
     c  SURFVALUESPEC(UEFBOTTOMSURF)) THEN
          WRITE(30,2345) LN, FRUEFML(COLUMN,ROW),
     c    (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c    (-1*UEFTOPSURF), FRUNIV(COLUMN,ROW)
2345      FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c    ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
          LN=LN+1
        ELSEIF (SURFVALUESPEC(PLENUMTOPSURF).EQ.
     c  SURFVALUESPEC(UEFTOPSURF)) THEN
          WRITE(30,2350) LN, FRUEFML(COLUMN,ROW),
     c    (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c    (-1*UEFTOPSURF), CLADORSURF, FRUNIV(COLUMN,ROW)
2350      FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c    ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
          LN=LN+1
        ENDIF
*     Loop through the regions above the fuel rod (i.e. the appropriate upper
core regions)
*     Define the upper region lower surface.
        IF (BPRAINSERTED.EQ..TRUE.) THEN
          DO 2450 REGION=1,NUMREGABOVEBPRA
*     Determine the current upper region's lower surface specification.
            IF (REGION.EQ.1) THEN
              REGIONTOPSURF=SYSTEMTOP
```

```
                    CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
     c              REGABOVEBPRA(REGION,1)
                 ENDIF
                 CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
     c           REGABOVEBPRA(REGION,1)
                 IF (REGION.EQ.NUMREGABOVEBPRA) THEN
                    REGIONBOTTOMSURF=UEFTOPSURF
                 ELSE
                    CURRENTSURFLABEL=0
                    DO 2370 V=1,(SN-1)
                       IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                             CURRENTSURFLABEL=V
                             EXIT
                          ENDIF
                       ENDIF
 2370               CONTINUE
                    IF (CURRENTSURFLABEL.EQ.0) THEN
                       REGIONBOTTOMSURF=SN
                       SURFTYPESPEC(SN)='PZ'
                       SURFVALUESPEC(SN)=CURRENTSURF
                       SN=SN+1
                    ELSE
                       REGIONBOTTOMSURF=CURRENTSURFLABEL
                    ENDIF
                 ENDIF
*     Determine if the fuel rod upper region material specification has
*     previously been defined.  If it has been previously defined, determine
*     the upper region material specification label.
                 IF (BPRAFLAG.EQ..FALSE.) THEN
                    FRUREGIONMLUNIQUE=.TRUE.
                    FRUREGIONML(COLUMN,ROW,REGION)=MN
*  Check Upper Core Region in the Fuel Rod Universe
                    DO 2373 D=1,REGABOVEBPRA(REGION,3)
                       IF (D.EQ.1) THEN
                          WRITE(200,2371) MN,
     c                    ABOVEBPRAZAIDS(REGION,D),
     c                    (-1*ABOVEBPRAWTS(REGION,D)), REGION,
     c                    ASSYID(COLUMN,ROW)
 2371                     FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                    '    $ Upper Core Region ',I2,
     c                    ' in Assembly ', A5)
                       ELSE
                          WRITE(200,2372) ABOVEBPRAZAIDS(REGION,D),
     c                    (-1*ABOVEBPRAWTS(REGION,D))
 2372                     FORMAT(T9,A9,3X,G14.6)
                       ENDIF
 2373               CONTINUE
                    WRITE(200,2374) MN
 2374               FORMAT(T1,'MT',I4,T9,'LWTR.03T')
                    MN=MN+1
                 ELSE
                    FRUREGIONMLUNIQUE=.FALSE.
                    IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
```

```
                     LEAVE=.FALSE.
                     DO 2382 RO=1,(ROW-1)
                        DO 2381 CO=1,50
                           DO 2380 BN=1,NUMOFBANKS
                              IF ((BANKNUM(CO,RO).NE.
     c                           0).AND.
     c                           (BANKDES(BANKNUM(CO,RO)).EQ.
     c                           'BPRA ')) THEN
                                 FRUREGIONML(COLUMN,ROW,REGION)=
     c                           FRUREGIONML(CO,RO,REGION)
                                 LEAVE=.TRUE.
                                 EXIT
                              ENDIF
 2380                      CONTINUE
                           IF (LEAVE.EQ..TRUE.) THEN
                              EXIT
                           ENDIF
 2381                   CONTINUE
                        IF (LEAVE.EQ..TRUE.) THEN
                           EXIT
                        ENDIF
 2382                CONTINUE
                     IF (LEAVE.EQ..FALSE.) THEN
                        DO 2385 RO=ROW,ROW
                           DO 2384 CO=1,(COLUMN-1)
                              DO 2383 BN=1,NUMOFBANKS
                                 IF ((BANKNUM(CO,RO).NE.
     c                           0).AND.
     c                           (BANKDES(BANKNUM(CO,RO)).EQ.
     c                           'BPRA ')) THEN
                                    FRUREGIONML(COLUMN,ROW,REGION)=
     c                              FRUREGIONML(CO,RO,REGION)
                                    LEAVE=.TRUE.
                                    EXIT
                                 ENDIF
 2383                         CONTINUE
                              IF (LEAVE.EQ..TRUE.) THEN
                                 EXIT
                              ENDIF
 2384                      CONTINUE
                           IF (LEAVE.EQ..TRUE.) THEN
                              EXIT
                           ENDIF
 2385                   CONTINUE
                     ENDIF
                  ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                     LEAVE=.FALSE.
                     DO 2388 RO=1,(ROW-1)
                        DO 2387 CO=1,50
                           DO 2386 BN=1,NUMOFBANKS
                              IF ((BANKNUM(CO,RO).NE.0)
     c                        .AND.(BANKDES(BANKNUM(CO,RO)).EQ.
     c                        'BPRA ')) THEN
                                 FRUREGIONML(COLUMN,ROW,REGION)=
```

```
     c                                     FRUREGIONML(CO,RO,REGION)
                                           LEAVE=.TRUE.
                                           EXIT
                                        ENDIF
 2386                                CONTINUE
                                     IF (LEAVE.EQ..TRUE.) THEN
                                        EXIT
                                     ENDIF
 2387                             CONTINUE
                                  IF (LEAVE.EQ..TRUE.) THEN
                                     EXIT
                                  ENDIF
 2388                          CONTINUE
                            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                               LEAVE=.FALSE.
                               DO 2391 RO=1,1
                                  DO 2390 CO=1,(COLUMN-1)
                                     DO 2389 BN=1,NUMOFBANKS
                                        IF ((BANKNUM(CO,RO).NE.0)
     c                                  .AND.(BANKDES(BANKNUM(CO,RO)).EQ.
     c                                  'BPRA ')) THEN
                                           FRUREGIONML(COLUMN,ROW,REGION)=
     c                                     FRUREGIONML(CO,RO,REGION)
                                           LEAVE=.TRUE.
                                           EXIT
                                        ENDIF
 2389                                CONTINUE
                                     IF (LEAVE.EQ..TRUE.) THEN
                                        EXIT
                                     ENDIF
 2390                             CONTINUE
                                  IF (LEAVE.EQ..TRUE.) THEN
                                     EXIT
                                  ENDIF
 2391                          CONTINUE
                            ENDIF
                         ENDIF
*    Write the cell specification for the fuel rod universe upper region.
                      IF (REGION.EQ.1) THEN
                         WRITE(30,2440) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c                   (-1*REGABOVEBPRA(REGION,2)),
     c                   REGIONBOTTOMSURF, FRUNIV(COLUMN,ROW), REGION
 2440                    FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
     c                   ' IMP:N=1 U=',I3,'   $ Upper-core region ',I2)
                         LN=LN+1
                         REGIONTOPSURF=REGIONBOTTOMSURF
                      ELSE
                         WRITE(30,2445) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c                   (-1*REGABOVEBPRA(REGION,2)), (-1*REGIONTOPSURF),
     c                   REGIONBOTTOMSURF, FRUNIV(COLUMN,ROW), REGION
 2445                    FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c                   ' IMP:N=1 U=',I3,'   $ Upper-core region ',I2)
                         LN=LN+1
                         REGIONTOPSURF=REGIONBOTTOMSURF
```

```
                 ENDIF
 2450            CONTINUE
                 BPRAFLAG=.TRUE.
              ELSEIF (CRAINSERTED.EQ..TRUE.) THEN
                 DO 2540 REGION=1,NUMREGABOVECRA
*     Determine the current upper region's lower surface specification.
                    IF (REGION.EQ.1) THEN
                       REGIONTOPSURF=SYSTEMTOP
                       CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
     c                 REGABOVECRA(REGION,1)
                    ENDIF
                    CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
     c              REGABOVECRA(REGION,1)
                    IF (REGION.EQ.NUMREGABOVECRA) THEN
                       REGIONBOTTOMSURF=UEFTOPSURF
                    ELSE
                    CURRENTSURFLABEL=0
                    DO 2460 V=1,(SN-1)
                       IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                  IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                             CURRENTSURFLABEL=V
                             EXIT
                          ENDIF
                       ENDIF
 2460               CONTINUE
                    IF (CURRENTSURFLABEL.EQ.0) THEN
                       REGIONBOTTOMSURF=SN
                       SURFTYPESPEC(SN)='PZ'
                       SURFVALUESPEC(SN)=CURRENTSURF
                       SN=SN+1
                    ELSE
                       REGIONBOTTOMSURF=CURRENTSURFLABEL
                    ENDIF
                    ENDIF
*     Determine if the fuel rod upper region material specification has
*     previously been defined.  If it has been previously defined, determine
*     the upper region material specification label.
                    IF (CRAFLAG.EQ..FALSE.) THEN
                       FRUREGIONMLUNIQUE=.TRUE.
                       FRUREGIONML(COLUMN,ROW,REGION)=MN
*  Check Upper Core Region in the Fuel Rod Universe
                       DO 2463 D=1,REGABOVECRA(REGION,3)
                          IF (D.EQ.1) THEN
                             WRITE(200,2461) MN,
     c                       ABOVECRAZAIDS(REGION,D),
     c                       (-1*ABOVECRAWTS(REGION,D)), REGION,
     c                       ASSYID(COLUMN,ROW)
 2461                        FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                       '    $ Upper Core Region ',I2,
     c                       ' in Assembly ', A5)
                          ELSE
                             WRITE(200,2462) ABOVECRAZAIDS(REGION,D),
     c                       (-1*ABOVECRAWTS(REGION,D))
 2462                        FORMAT(T9,A9,3X,G14.6)
```

```
                        ENDIF
 2463               CONTINUE
                    WRITE(200,2464) MN
 2464               FORMAT(T1,'MT',I4,T9,'LWTR.03T')
                    MN=MN+1
                ELSE
                    FRUREGIONMLUNIQUE=.FALSE.
                    IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                        LEAVE=.FALSE.
                        DO 2472 RO=1,(ROW-1)
                            DO 2471 CO=1,50
                                DO 2470 BN=1,NUMOFBANKS
                                    IF ((BANKNUM(CO,RO).NE.0)
     c                              .AND.(BANKDES(BANKNUM(CO,RO)).EQ.
     c                              'CRA  ')) THEN
                                        FRUREGIONML(COLUMN,ROW,REGION)=
     c                                  FRUREGIONML(CO,RO,REGION)
                                        LEAVE=.TRUE.
                                        EXIT
                                    ENDIF
 2470                           CONTINUE
                                IF (LEAVE.EQ..TRUE.) THEN
                                    EXIT
                                ENDIF
 2471                       CONTINUE
                            IF (LEAVE.EQ..TRUE.) THEN
                                EXIT
                            ENDIF
 2472                   CONTINUE
                        IF (LEAVE.EQ..FALSE.) THEN
                            DO 2475 RO=ROW,ROW
                                DO 2474 CO=1,(COLUMN-1)
                                    DO 2473 BN=1,NUMOFBANKS
                                        IF ((BANKNUM(CO,RO).NE.0)
     c                                  .AND.(BANKDES(BANKNUM(CO,RO)).EQ.
     c                                  'CRA  ')) THEN
                                            FRUREGIONML(COLUMN,ROW,REGION)=
     c                                      FRUREGIONML(CO,RO,REGION)
                                            LEAVE=.TRUE.
                                            EXIT
                                        ENDIF
 2473                               CONTINUE
                                    IF (LEAVE.EQ..TRUE.) THEN
                                        EXIT
                                    ENDIF
 2474                           CONTINUE
                                IF (LEAVE.EQ..TRUE.) THEN
                                    EXIT
                                ENDIF
 2475                       CONTINUE
                        ENDIF
                    ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                        LEAVE=.FALSE.
                        DO 2478 RO=1,(ROW-1)
```

```
                            DO 2477 CO=1,50
                               DO 2476 BN=1,NUMOFBANKS
                                  IF ((BANKNUM(CO,RO).NE.0)
     c                               .AND.(BANKDES(BANKNUM(CO,RO)).EQ.
     c                               'CRA  ')) THEN
                                     FRUREGIONML(COLUMN,ROW,REGION)=
     c                               FRUREGIONML(CO,RO,REGION)
                                     LEAVE=.TRUE.
                                     EXIT
                                  ENDIF
 2476                          CONTINUE
                               IF (LEAVE.EQ..TRUE.) THEN
                                  EXIT
                               ENDIF
 2477                       CONTINUE
                            IF (LEAVE.EQ..TRUE.) THEN
                               EXIT
                            ENDIF
 2478                    CONTINUE
                      ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                         LEAVE=.FALSE.
                         DO 2481 RO=1,1
                            DO 2480 CO=1,(COLUMN-1)
                               DO 2479 BN=1,NUMOFBANKS
                                  IF ((BANKNUM(CO,RO).NE.0)
     c                               .AND.(BANKDES(BANKNUM(CO,RO)).EQ.
     c                               'CRA  ')) THEN
                                     FRUREGIONML(COLUMN,ROW,REGION)=
     c                               FRUREGIONML(CO,RO,REGION)
                                     LEAVE=.TRUE.
                                     EXIT
                                  ENDIF
 2479                          CONTINUE
                               IF (LEAVE.EQ..TRUE.) THEN
                                  EXIT
                               ENDIF
 2480                       CONTINUE
                            IF (LEAVE.EQ..TRUE.) THEN
                               EXIT
                            ENDIF
 2481                    CONTINUE
                      ENDIF
                   ENDIF
*     Write the cell specification for the fuel rod universe upper region.
                IF (REGION.EQ.1) THEN
                   WRITE(30,2530) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c             (-1*REGABOVECRA(REGION,2)),
     c             REGIONBOTTOMSURF, FRUNIV(COLUMN,ROW), REGION
 2530              FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
     c             ' IMP:N=1 U=',I3,'    $ Upper-core region ',I2)
                   LN=LN+1
                   REGIONTOPSURF=REGIONBOTTOMSURF
                ELSE
                   WRITE(30,2535) LN, FRUREGIONML(COLUMN,ROW,REGION),
```

```
     c             (-1*REGABOVECRA(REGION,2)), (-1*REGIONTOPSURF),
     c             REGIONBOTTOMSURF, FRUNIV(COLUMN,ROW), REGION
 2535              FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c             ' IMP:N=1 U=',I3,'   $ Upper-core region ',I2)
                   LN=LN+1
                   REGIONTOPSURF=REGIONBOTTOMSURF
                ENDIF
 2540           CONTINUE
                CRAFLAG=.TRUE.
             ELSEIF (APSRAINSERTED.EQ..TRUE.) THEN
                DO 2630 REGION=1,NUMREGABOVEAPSRA
*    Determine the current upper region's lower surface specification.
                   IF (REGION.EQ.1) THEN
                      REGIONTOPSURF=SYSTEMTOP
                      CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
     c                REGABOVEAPSRA(REGION,1)
                   ENDIF
                   CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
     c             REGABOVEAPSRA(REGION,1)
                   IF (REGION.EQ.NUMREGABOVEAPSRA) THEN
                      REGIONBOTTOMSURF=UEFTOPSURF
                   ELSE
                   CURRENTSURFLABEL=0
                   DO 2550 V=1,(SN-1)
                      IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                            CURRENTSURFLABEL=V
                            EXIT
                         ENDIF
                      ENDIF
 2550              CONTINUE
                   IF (CURRENTSURFLABEL.EQ.0) THEN
                      REGIONBOTTOMSURF=SN
                      SURFTYPESPEC(SN)='PZ'
                      SURFVALUESPEC(SN)=CURRENTSURF
                      SN=SN+1
                   ELSE
                      REGIONBOTTOMSURF=CURRENTSURFLABEL
                   ENDIF
                   ENDIF
*    Determine if the fuel rod upper region material specification has
*    previously been defined.  If it has been previously defined, determine
*    the upper region material specification label.
                   IF (APSRAFLAG.EQ..FALSE.) THEN
                      FRUREGIONMLUNIQUE=.TRUE.
                      FRUREGIONML(COLUMN,ROW,REGION)=MN
*  Check Upper Core Region in the Fuel Rod Universe
                      DO 2553 D=1,REGABOVEAPSRA(REGION,3)
                         IF (D.EQ.1) THEN
                            WRITE(200,2551) MN,
     c                      ABOVEAPSRAZAIDS(REGION,D),
     c                      (-1*ABOVEAPSRAWTS(REGION,D)), REGION,
     c                      ASSYID(COLUMN,ROW)
 2551                       FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
```

```
     c                        '    $ Upper Core Region ',I2,
     c                        ' in Assembly ', A5)
                            ELSE
                              WRITE(200,2552) ABOVEAPSRAZAIDS(REGION,D),
     c                        (-1*ABOVEAPSRAWTS(REGION,D))
 2552                         FORMAT(T9,A9,3X,G14.6)
                            ENDIF
 2553                     CONTINUE
                          WRITE(200,2554) MN
 2554                     FORMAT(T1,'MT',I4,T9,'LWTR.03T')
                          MN=MN+1
                        ELSE
                          FRUREGIONMLUNIQUE=.FALSE.
                          IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                            LEAVE=.FALSE.
                            DO 2562 RO=1,(ROW-1)
                              DO 2561 CO=1,50
                                DO 2560 BN=1,NUMOFBANKS
                                  IF ((BANKNUM(CO,RO).NE.0)
     c                            .AND.(BANKDES(BANKNUM(CO,RO)).EQ.
     c                            'APSRA')) THEN
                                    FRUREGIONML(COLUMN,ROW,REGION)=
     c                              FRUREGIONML(CO,RO,REGION)
                                    LEAVE=.TRUE.
                                    EXIT
                                  ENDIF
 2560                           CONTINUE
                                IF (LEAVE.EQ..TRUE.) THEN
                                  EXIT
                                ENDIF
 2561                         CONTINUE
                              IF (LEAVE.EQ..TRUE.) THEN
                                EXIT
                              ENDIF
 2562                       CONTINUE
                            IF (LEAVE.EQ..FALSE.) THEN
                              DO 2565 RO=ROW,ROW
                                DO 2564 CO=1,(COLUMN-1)
                                  DO 2563 BN=1,NUMOFBANKS
                                    IF ((BANKNUM(CO,RO).NE.0)
     c                              .AND.(BANKDES(BANKNUM(CO,RO)).EQ.
     c                              'APSRA')) THEN
                                      FRUREGIONML(COLUMN,ROW,REGION)=
     c                                FRUREGIONML(CO,RO,REGION)
                                      LEAVE=.TRUE.
                                      EXIT
                                    ENDIF
 2563                             CONTINUE
                                  IF (LEAVE.EQ..TRUE.) THEN
                                    EXIT
                                  ENDIF
 2564                           CONTINUE
                                IF (LEAVE.EQ..TRUE.) THEN
                                  EXIT
```

```
                    ENDIF
 2565             CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               LEAVE=.FALSE.
               DO 2568 RO=1,(ROW-1)
                  DO 2567 CO=1,50
                     DO 2566 BN=1,NUMOFBANKS
                        IF ((BANKNUM(CO,RO).NE.0)
     c                  .AND.(BANKDES(BANKNUM(CO,RO)).EQ.
     c                  'APSRA')) THEN
                           FRUREGIONML(COLUMN,ROW,REGION)=
     c                     FRUREGIONML(CO,RO,REGION)
                           LEAVE=.TRUE.
                           EXIT
                        ENDIF
 2566                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 2567             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 2568          CONTINUE
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
               LEAVE=.FALSE.
               DO 2571 RO=1,1
                  DO 2570 CO=1,(COLUMN-1)
                     DO 2569 BN=1,NUMOFBANKS
                        IF ((BANKNUM(CO,RO).NE.0)
     c                  .AND.(BANKDES(BANKNUM(CO,RO)).EQ.
     c                  'APSRA')) THEN
                           FRUREGIONML(COLUMN,ROW,REGION)=
     c                     FRUREGIONML(CO,RO,REGION)
                           LEAVE=.TRUE.
                           EXIT
                        ENDIF
 2569                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 2570             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 2571          CONTINUE
            ENDIF
         ENDIF
*     Write the cell specification for the fuel rod universe upper region.
         IF (REGION.EQ.1) THEN
            WRITE(30,2620) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c      (-1*REGABOVEAPSRA(REGION,2)),
     c      REGIONBOTTOMSURF, FRUNIV(COLUMN,ROW), REGION
```

```
 2620              FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
     c             ' IMP:N=1 U=',I3,'   $ Upper-core region ',I2)
                   LN=LN+1
                   REGIONTOPSURF=REGIONBOTTOMSURF
                ELSE
                   WRITE(30,2625) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c             (-1*REGABOVEAPSRA(REGION,2)), (-1*REGIONTOPSURF),
     c             REGIONBOTTOMSURF, FRUNIV(COLUMN,ROW), REGION
 2625              FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c             ' IMP:N=1 U=',I3,'   $ Upper-core region ',I2)
                   LN=LN+1
                   REGIONTOPSURF=REGIONBOTTOMSURF
                ENDIF
 2630           CONTINUE
                APSRAFLAG=.TRUE.
             ELSE
                DO 2720 REGION=1,NUMREGABOVE
*     Determine the current upper region's lower surface specification.
                   IF (REGION.EQ.1) THEN
                      REGIONTOPSURF=SYSTEMTOP
                      CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
     c                REGABOVE(REGION,1)
                   ENDIF
                   CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
     c             REGABOVE(REGION,1)
                   IF (REGION.EQ.NUMREGABOVE) THEN
                      REGIONBOTTOMSURF=UEFTOPSURF
                   ELSE
                   CURRENTSURFLABEL=0
                   DO 2640 V=1,(SN-1)
                      IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                            CURRENTSURFLABEL=V
                            EXIT
                         ENDIF
                      ENDIF
 2640              CONTINUE
                   IF (CURRENTSURFLABEL.EQ.0) THEN
                      REGIONBOTTOMSURF=SN
                      SURFTYPESPEC(SN)='PZ'
                      SURFVALUESPEC(SN)=CURRENTSURF
                      SN=SN+1
                   ELSE
                      REGIONBOTTOMSURF=CURRENTSURFLABEL
                   ENDIF
                   ENDIF
*     Determine if the fuel rod upper region material specification has
*     previously been defined.  If it has been previously defined, determine
*     the upper region material specification label.
                   IF (PLAINFLAG.EQ..FALSE.) THEN
                      FRUREGIONMLUNIQUE=.TRUE.
                      FRUREGIONML(COLUMN,ROW,REGION)=MN
*  Check Upper Core Region in the Fuel Rod Universe
                      DO 2643 D=1,REGABOVE(REGION,3)
```

```
                 IF (D.EQ.1) THEN.
                    WRITE(200,2641) MN,
     c              ABOVEZAIDS(REGION,D),
     c              (-1*ABOVEWTS(REGION,D)), REGION,
     c              ASSYID(COLUMN,ROW)
 2641               FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c              '    $ Upper Core Region ',I2,
     c              ' in Assembly ', A5)
                 ELSE
                    WRITE(200,2642) ABOVEZAIDS(REGION,D),
     c              (-1*ABOVEWTS(REGION,D))
 2642               FORMAT(T9,A9,3X,G14.6)
                 ENDIF
 2643         CONTINUE
              WRITE(200,2644) MN
 2644         FORMAT(T1,'MT',I4,T9,'LWTR.03T')
              MN=MN+1
           ELSE
              FRUREGIONMLUNIQUE=.FALSE.
              IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                 LEAVE=.FALSE.
                 DO 2652 RO=1,(ROW-1)
                    DO 2651 CO=1,50
                       DO 2650 BN=1,NUMOFBANKS
                          IF ((BANKNUM(CO,RO).EQ.0).AND.
     c                    (ASSYID(CO,RO).NE.'     ')) THEN
                             FRUREGIONML(COLUMN,ROW,REGION)=
     c                       FRUREGIONML(CO,RO,REGION)
                             LEAVE=.TRUE.
                             EXIT
                          ENDIF
 2650                  CONTINUE
                       IF (LEAVE.EQ..TRUE.) THEN
                          EXIT
                       ENDIF
 2651               CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
 2652            CONTINUE
                 IF (LEAVE.EQ..FALSE.) THEN
                    DO 2655 RO=ROW,ROW
                       DO 2654 CO=1,(COLUMN-1)
                          DO 2653 BN=1,NUMOFBANKS
                             IF ((BANKNUM(CO,RO).EQ.0).AND.
     c                    (ASSYID(CO,RO).NE.'     ')) THEN
                                FRUREGIONML(COLUMN,ROW,REGION)=
     c                          FRUREGIONML(CO,RO,REGION)
                                LEAVE=.TRUE.
                                EXIT
                             ENDIF
 2653                     CONTINUE
                          IF (LEAVE.EQ..TRUE.) THEN
                             EXIT
```

```
                    ENDIF
 2654             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                  ENDIF
 2655           CONTINUE
              ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
              LEAVE=.FALSE.
              DO 2658 RO=1,(ROW-1)
                DO 2657 CO=1,50
                  DO 2656 BN=1,NUMOFBANKS
                    IF ((BANKNUM(CO,RO).EQ.0).AND.
     c              (ASSYID(CO,RO).NE.'      ')) THEN
                      FRUREGIONML(COLUMN,ROW,REGION)=
     c                FRUREGIONML(CO,RO,REGION)
                      LEAVE=.TRUE.
                      EXIT
                    ENDIF
 2656             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                  ENDIF
 2657           CONTINUE
                IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
                ENDIF
 2658         CONTINUE
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
              LEAVE=.FALSE.
              DO 2661 RO=1,1
                DO 2660 CO=1,(COLUMN-1)
                  DO 2659 BN=1,NUMOFBANKS
                    IF ((BANKNUM(CO,RO).EQ.0).AND.
     c              (ASSYID(CO,RO).NE.'      ')) THEN
                      FRUREGIONML(COLUMN,ROW,REGION)=
     c                FRUREGIONML(CO,RO,REGION)
                      LEAVE=.TRUE.
                      EXIT
                    ENDIF
 2659             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                  ENDIF
 2660           CONTINUE
                IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
                ENDIF
 2661         CONTINUE
            ENDIF
          ENDIF
*    Write the cell specification for the fuel rod universe upper region.
        IF (REGION.EQ.1) THEN
          WRITE(30,2710) LN, FRUREGIONML(COLUMN,ROW,REGION),
```

```
     c            (-1*REGABOVE(REGION,2)),
     c            REGIONBOTTOMSURF, FRUNIV(COLUMN,ROW), REGION
 2710             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
     c            ' IMP:N=1 U=',I3,'    $ Upper-core region ',I2)
                  LN=LN+1
                  REGIONTOPSURF=REGIONBOTTOMSURF
               ELSE
                  WRITE(30,2715) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c            (-1*REGABOVE(REGION,2)), (-1*REGIONTOPSURF),
     c            REGIONBOTTOMSURF, FRUNIV(COLUMN,ROW), REGION
 2715             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c            ' IMP:N=1 U=',I3,'    $ Upper-core region ',I2)
                  LN=LN+1
                  REGIONTOPSURF=REGIONBOTTOMSURF
               ENDIF
 2720          CONTINUE
               PLAINFLAG=.TRUE.
            ENDIF
            SPACHEIGHT=0.0
*     Loop through the spacer and moderator regions along the axial
*     length of the fuel rod (from top to bottom).
         DO 2722 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
            SPACHEIGHT=SPACHEIGHT+SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
 2722    CONTINUE
         DO 2843 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
*     Define the homogenized spacer region bounding surfaces.
            IF (SPN.EQ.1) THEN
               SPACERTOPSURF=UEFBOTTOMSURF
               CURRENTSURF=SURFVALUESPEC(UEFBOTTOMSURF)-
     c         SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
               CURRENTSURFLABEL=0
               DO 2730 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 2730          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  SPACERBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  SPACERBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
               WATERREGIONTOPSURF=SPACERBOTTOMSURF
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
               CURRENTSURFLABEL=0
               DO 2740 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
```

```
                  EXIT
               ENDIF
            ENDIF
2740     CONTINUE
         IF (CURRENTSURFLABEL.EQ.0) THEN
            WATERREGIONBOTTOMSURF=SN
            SURFTYPESPEC(SN)='PZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
         ELSE
            WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
         ENDIF
      ELSEIF ((SPN.NE.1).AND.(SPN.NE.
     c      NUMOFSPACERS(DESNUM(COLUMN,ROW)))) THEN
         SPACERTOPSURF=WATERREGIONBOTTOMSURF
         CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c      SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
         CURRENTSURFLABEL=0
         DO 2750 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'PZ') THEN
       IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
2750     CONTINUE
         IF (CURRENTSURFLABEL.EQ.0) THEN
            SPACERBOTTOMSURF=SN
            SURFTYPESPEC(SN)='PZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
         ELSE
            SPACERBOTTOMSURF=CURRENTSURFLABEL
         ENDIF
         WATERREGIONTOPSURF=SPACERBOTTOMSURF
         CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
         CURRENTSURFLABEL=0
         DO 2760 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'PZ') THEN
       IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
2760     CONTINUE
         IF (CURRENTSURFLABEL.EQ.0) THEN
            WATERREGIONBOTTOMSURF=SN
            SURFTYPESPEC(SN)='PZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
         ELSE
            WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
         ENDIF
      ELSEIF (SPN.EQ.NUMOFSPACERS(DESNUM(COLUMN,ROW))) THEN
```

```
              SPACERTOPSURF=WATERREGIONBOTTOMSURF
              CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c        SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
              CURRENTSURFLABEL=0
              DO 2761 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 2761         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 SPACERBOTTOMSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 SPACERBOTTOMSURF=CURRENTSURFLABEL
              ENDIF
              WATERREGIONTOPSURF=SPACERBOTTOMSURF
              WATERREGIONBOTTOMSURF=NODEBOTTOMSURF
           ENDIF
           IF (DUNIQUE(COLUMN,ROW).EQ..TRUE.) THEN
           HOMOSPACERML=MN
           MN=MN+1
           HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN)=HOMOSPACERML
              CR50=0.04173684
              CR52=0.837
              CR53=0.09673684
              CR54=0.02452632
              FE54=0.05699324
              FE56=0.91868499
              FE57=0.02141247
              FE58=0.00290930
              NI58=0.67394595
              NI60=0.26648649
              NI61=0.01178378
              NI62=0.03783784
              NI64=0.00994594
              CU63=0.683
              CU65=0.317
*
* Zircaloy Spacer Grid Specification
*
           IF (SPACERMAT(DESNUM(COLUMN,ROW),SPN).EQ.1) THEN
              VAL1=(ASSYPITCH**2)*
     c        SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
              VAL2=RODNUM(DESNUM(COLUMN,ROW))*(PI)*
     c        (CLADRADIUS(DESNUM(COLUMN,ROW),2)**2)*
     c        SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
              IF (BANDW.EQ..TRUE.) THEN
                 VAL3=16.0*(GTDATA(DESNUM(COLUMN,ROW),2)**2)*
     c           (PI)*SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
```

```
                 ELSEIF ((WESTINGHOUSE.EQ..TRUE.).AND.
     c           (GTSPLIT.NE.1)) THEN
                    VAL3=24.0*(GTDATA(DESNUM(COLUMN,ROW),2)**2)*
     c              (PI)*SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                 ELSEIF ((WESTINGHOUSE.EQ..TRUE.).AND.
     c           (GTSPLIT.EQ.1)) THEN
                    DO 2762 SCTN=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
                       IF ((SURFVALUESPEC(SPACERBOTTOMSURF).LT.
     c                 GTAXDATA(DESNUM(COLUMN,ROW),3,SCTN)).AND.
     c                 (SURFVALUESPEC(SPACERBOTTOMSURF).GE.
     c                 GTAXDATA(DESNUM(COLUMN,ROW),4,SCTN))) THEN
                          VAL3=24.0*
     c                    (GTAXDATA(DESNUM(COLUMN,ROW),2,SCTN)**2)*
     c                    (PI)*SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                       ENDIF
 2762               CONTINUE
                 ENDIF
                 VAL4=(ITDATA(DESNUM(COLUMN,ROW),2)**2)*(PI)*
     c           SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                 TVOL=VAL1-VAL2-VAL3-VAL4
                 SPACVOL=SPACERVOL(DESNUM(COLUMN,ROW),SPN)
                 MODVOL=TVOL-SPACVOL
                 HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)=
     c           ((SPACVOL*6.56)+(MODVOL*MODDENSITY))/
     c           (MODVOL+SPACVOL)
                 SPACMASS=SPACVOL*6.56
                 MODMASS=MODVOL*MODDENSITY
                 SPACFRAC=SPACMASS/(SPACMASS+MODMASS)
                 MODFRAC=MODMASS/(SPACMASS+MODMASS)
           BWTINH2O=((PPMB*1E-6)/(1.0+(PPMB*1E-6)))*100.0
           HWTINH2O=((1.008664904*0.999167*2.0)/
     c     ((1.008664904*0.999167*2.0)+
     c     (1.008664904*15.857510)))*(100.0-BWTINH2O)
           OWTINH2O=((1.008664904*15.857510)/
     c     ((1.008664904*0.999167*2.0)+
     c     (1.008664904*15.857510)))*(100.0-BWTINH2O)
           B10WTINH2O=((1.008664904*9.926922*0.194)/
     c     ((1.008664904*9.926922*0.194)+
     c     (1.008664904*10.914730*0.806)))*BWTINH2O
           B11WTINH2O=((1.008664904*10.914730*0.806)/
     c     ((1.008664904*9.926922*0.194)+
     c     (1.008664904*10.914730*0.806)))*BWTINH2O
                 OWT=(OWTINH2O*MODFRAC)+(0.120*SPACFRAC)
                 HWT=HWTINH2O*MODFRAC
                 B10WT=B10WTINH2O*MODFRAC
                 B11WT=B11WTINH2O*MODFRAC
                 CRWT=0.100*SPACFRAC
                 FEWT=0.200*SPACFRAC
                 ZRWT=98.180*SPACFRAC
                 SNWT=1.400*SPACFRAC
                 WRITE(200,2771) HOMOSPACERML, (-1*HWT),
     c           SPN, ASSYID(COLUMN,ROW)
 2771            FORMAT(T1,'M',I4,T9,' 1001.50c'3X,G14.8,
     c '    $ Homogenized Zirc-4 Spacer Number ',I2,' in Assembly ',A5)
```

```
            WRITE(200,2772) -1*OWT
 2772       FORMAT(T9,' 8016.50c',3X,G14.8)
            WRITE(200,2773) -1*B10WT
 2773       FORMAT(T9,' 5010.50c',3X,G14.8)
            WRITE(200,2774) -1*B11WT
 2774       FORMAT(T9,' 5011.56c',3X,G14.8)
            WRITE(200,2775) -1*CRWT*CR50
 2775       FORMAT(T9,'24050.60c',3X,G14.8)
            WRITE(200,7050) -1*CRWT*CR52
 7050       FORMAT(T9,'24052.60c',3X,G14.8)
            WRITE(200,7051) -1*CRWT*CR53
 7051       FORMAT(T9,'24053.60c',3X,G14.8)
            WRITE(200,7052) -1*CRWT*CR54
 7052       FORMAT(T9,'24054.60c',3X,G14.8)
            WRITE(200,2776) -1*FEWT*FE54
 2776       FORMAT(T9,'26054.60c',3X,G14.8)
            WRITE(200,7053) -1*FEWT*FE56
 7053       FORMAT(T9,'26056.60c',3X,G14.8)
            WRITE(200,7054) -1*FEWT*FE57
 7054       FORMAT(T9,'26057.60c',3X,G14.8)
            WRITE(200,7055) -1*FEWT*FE58
 7055       FORMAT(T9,'26058.60c',3X,G14.8)
            WRITE(200,2777) -1*ZRWT
 2777       FORMAT(T9,'40093.50c',3X,G14.8)
            WRITE(200,2778) -1*SNWT
 2778       FORMAT(T9,'50000.35c',3X,G14.8)
*
* Stainless Steel Spacer Grid Specification
*
          ELSEIF (SPACERMAT(DESNUM(COLUMN,ROW),SPN).EQ.2) THEN
            VAL1=(ASSYPITCH**2)*
     c      SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
            VAL2=RODNUM(DESNUM(COLUMN,ROW))*(PI)*
     c      (CLADRADIUS(DESNUM(COLUMN,ROW),2)**2)*
     c      SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
            IF (BANDW.EQ..TRUE.) THEN
              VAL3=16.0*(GTDATA(DESNUM(COLUMN,ROW),2)**2)*
     c        (PI)*SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
            ELSEIF ((WESTINGHOUSE.EQ..TRUE.).AND.
     c      (GTSPLIT.NE.1)) THEN
              VAL3=24.0*(GTDATA(DESNUM(COLUMN,ROW),2)**2)*
     c        (PI)*SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
            ELSEIF ((WESTINGHOUSE.EQ..TRUE.).AND.
     c      (GTSPLIT.EQ.1)) THEN
              DO 2763 SCTN=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
                IF ((SURFVALUESPEC(SPACERBOTTOMSURF).LT.
     c          GTAXDATA(DESNUM(COLUMN,ROW),3,SCTN)).AND.
     c          (SURFVALUESPEC(SPACERBOTTOMSURF).GE.
     c          GTAXDATA(DESNUM(COLUMN,ROW),4,SCTN))) THEN
                  VAL3=24.0*
     c            (GTAXDATA(DESNUM(COLUMN,ROW),2,SCTN)**2)*
     c            (PI)*SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                ENDIF
 2763         CONTINUE
```

```
            ENDIF
            VAL4=(ITDATA(DESNUM(COLUMN,ROW),2)**2)*(PI)*
     c      SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
            TVOL=VAL1-VAL2-VAL3-VAL4
            SPACVOL=SPACERVOL(DESNUM(COLUMN,ROW),SPN)
            MODVOL=TVOL-SPACVOL
            HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)=
     c      ((SPACVOL*7.90)+(MODVOL*MODDENSITY))/
     c      (MODVOL+SPACVOL)
            SPACMASS=SPACVOL*7.90
            MODMASS=MODVOL*MODDENSITY
            SPACFRAC=SPACMASS/(SPACMASS+MODMASS)
            MODFRAC=MODMASS/(SPACMASS+MODMASS)
        BWTINH2O=((PPMB*1E-6)/(1.0+(PPMB*1E-6)))*100.0
        HWTINH2O=((1.008664904*0.999167*2.0)/
     c  ((1.008664904*0.999167*2.0)+
     c  (1.008664904*15.857510)))*(100.0-BWTINH2O)
        OWTINH2O=((1.008664904*15.857510)/
     c  ((1.008664904*0.999167*2.0)+
     c  (1.008664904*15.857510)))*(100.0-BWTINH2O)
        B10WTINH2O=((1.008664904*9.926922*0.194)/
     c  ((1.008664904*9.926922*0.194)+
     c  (1.008664904*10.914730*0.806)))*BWTINH2O
        B11WTINH2O=((1.008664904*10.914730*0.806)/
     c  ((1.008664904*9.926922*0.194)+
     c  (1.008664904*10.914730*0.806)))*BWTINH2O
            OWT=OWTINH2O*MODFRAC
            HWT=HWTINH2O*MODFRAC
            B10WT=B10WTINH2O*MODFRAC
            B11WT=B11WTINH2O*MODFRAC
            CWT=0.08*SPACFRAC
            NWT=0.100*SPACFRAC
            SIWT=0.75*SPACFRAC
            PWT=0.045*SPACFRAC
            SWT=0.030*SPACFRAC
            CRWT=19.000*SPACFRAC
            MNWT=2.000*SPACFRAC
            FEWT=68.745*SPACFRAC
            NIWT=9.250*SPACFRAC
            WRITE(200,2779) HOMOSPACERML, (-1*HWT),
     c      SPN, ASSYID(COLUMN,ROW)
2779        FORMAT(T1,'M',I4,T9,' 1001.50c'3X,G14.8,
     c '   $ Homogenized SS304 Spacer Number ',I2,' in Assembly ',A5)
            WRITE(200,2780) -1*OWT
2780        FORMAT(T9,' 8016.50c',3X,G14.8)
            WRITE(200,2781) -1*B10WT
2781        FORMAT(T9,' 5010.50c',3X,G14.8)
            WRITE(200,2782) -1*B11WT
2782        FORMAT(T9,' 5011.56c',3X,G14.8)
            WRITE(200,2783) -1*CWT
2783        FORMAT(T9,'6000.50c',3X,G14.8)
            WRITE(200,2784) -1*NWT
2784        FORMAT(T9,'7014.50c',3X,G14.8)
            WRITE(200,2785) -1*SIWT
```

```
2785          FORMAT(T9,'14000.50c',3X,G14.8)
              WRITE(200,2786) -1*PWT
2786          FORMAT(T9,'15031.50c',3X,G14.8)
              WRITE(200,2787) -1*SWT
2787          FORMAT(T9,'16032.50c',3X,G14.8)
              WRITE(200,2788) -1*CRWT*CR50
2788          FORMAT(T9,'24050.60c',3X,G14.8)
              WRITE(200,7056) -1*CRWT*CR52
7056          FORMAT(T9,'24052.60c',3X,G14.8)
              WRITE(200,7057) -1*CRWT*CR53
7057          FORMAT(T9,'24053.60c',3X,G14.8)
              WRITE(200,7058) -1*CRWT*CR54
7058          FORMAT(T9,'24054.60c',3X,G14.8)
              WRITE(200,2789) -1*MNWT
2789          FORMAT(T9,'25055.50c',3X,G14.8)
              WRITE(200,2790) -1*FEWT*FE54
2790          FORMAT(T9,'26054.60c',3X,G14.8)
              WRITE(200,7059) -1*FEWT*FE56
7059          FORMAT(T9,'26056.60c',3X,G14.8)
              WRITE(200,7060) -1*FEWT*FE57
7060          FORMAT(T9,'26057.60c',3X,G14.8)
              WRITE(200,7061) -1*FEWT*FE58
7061          FORMAT(T9,'26058.60c',3X,G14.8)
              WRITE(200,2791) -1*NIWT*NI58
2791          FORMAT(T9,'28058.60c',3X,G14.8)
              WRITE(200,7062) -1*NIWT*NI60
7062          FORMAT(T9,'28060.60c',3X,G14.8)
              WRITE(200,7063) -1*NIWT*NI61
7063          FORMAT(T9,'28061.60c',3X,G14.8)
              WRITE(200,7064) -1*NIWT*NI62
7064          FORMAT(T9,'28062.60c',3X,G14.8)
              WRITE(200,7065) -1*NIWT*NI64
7065          FORMAT(T9,'28064.60c',3X,G14.8)
*
* Inconel Spacer Grid Specification
*
          ELSEIF (SPACERMAT(DESNUM(COLUMN,ROW),SPN).EQ.3) THEN
             VAL1=(ASSYPITCH**2)*
     c       SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
             VAL2=RODNUM(DESNUM(COLUMN,ROW))*(PI)*
     c       (CLADRADIUS(DESNUM(COLUMN,ROW),2)**2)*
     c       SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
             IF (BANDW.EQ..TRUE.) THEN
                VAL3=16.0*(GTDATA(DESNUM(COLUMN,ROW),2)**2)*
     c          (PI)*SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
             ELSEIF ((WESTINGHOUSE.EQ..TRUE.).AND.
     c       (GTSPLIT.NE.1)) THEN
                VAL3=24.0*(GTDATA(DESNUM(COLUMN,ROW),2)**2)*
     c          (PI)*SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
             ELSEIF ((WESTINGHOUSE.EQ..TRUE.).AND.
     c       (GTSPLIT.EQ.1)) THEN
                DO 2764 SCTN=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
                   IF ((SURFVALUESPEC(SPACERBOTTOMSURF).LT.
     c             GTAXDATA(DESNUM(COLUMN,ROW),3,SCTN)).AND.
```

```
     c                   (SURFVALUESPEC(SPACERBOTTOMSURF).GE.
     c                   GTAXDATA(DESNUM(COLUMN,ROW),4,SCTN))) THEN
                            VAL3=24.0*
     c                      (GTAXDATA(DESNUM(COLUMN,ROW),2,SCTN)**2)*
     c                      (PI)*SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                         ENDIF
 2764                 CONTINUE
                   ENDIF
                   VAL4=(ITDATA(DESNUM(COLUMN,ROW),2)**2)*(PI)*
     c             SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                   TVOL=VAL1-VAL2-VAL3-VAL4
                   SPACVOL=SPACERVOL(DESNUM(COLUMN,ROW),SPN)
                   MODVOL=TVOL-SPACVOL
                   HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)=
     c             ((SPACVOL*8.19)+(MODVOL*MODDENSITY))/
     c             (MODVOL+SPACVOL)
                   SPACMASS=SPACVOL*8.19
                   MODMASS=MODVOL*MODDENSITY
                   SPACFRAC=SPACMASS/(SPACMASS+MODMASS)
                   MODFRAC=MODMASS/(SPACMASS+MODMASS)
            BWTINH2O=((PPMB*1E-6)/(1.0+(PPMB*1E-6)))*100.0
            HWTINH2O=((1.008664904*0.999167*2.0)/
     c      ((1.008664904*0.999167*2.0)+
     c      (1.008664904*15.857510)))*(100.0-BWTINH2O)
            OWTINH2O=((1.008664904*15.857510)/
     c      ((1.008664904*0.999167*2.0)+
     c      (1.008664904*15.857510)))*(100.0-BWTINH2O)
            B10WTINH2O=((1.008664904*9.926922*0.194)/
     c      ((1.008664904*9.926922*0.194)+
     c      (1.008664904*10.914730*0.806)))*BWTINH2O
            B11WTINH2O=((1.008664904*10.914730*0.806)/
     c      ((1.008664904*9.926922*0.194)+
     c      (1.008664904*10.914730*0.806)))*BWTINH2O
                   B10WTINB=((1.008664904*9.926922*0.194)/
     c             ((1.008664904*9.926922*0.194)+
     c             (1.008664904*10.914730*0.806)))*100.0
                   B11WTINB=((1.008664904*10.914730*0.806)/
     c             ((1.008664904*9.926922*0.194)+
     c             (1.008664904*10.914730*0.806)))*100.0
                   OWT=OWTINH2O*MODFRAC
                   HWT=HWTINH2O*MODFRAC
                   B10WT=(B10WTINH2O*MODFRAC)+
     c             (B10WTINB*0.00006*SPACFRAC)
                   B11WT=(B11WTINH2O*MODFRAC)+
     c             (B11WTINB*0.00006*SPACFRAC)
                   CWT=0.080*SPACFRAC
                   SIWT=0.350*SPACFRAC
                   PWT=0.015*SPACFRAC
                   SWT=0.015*SPACFRAC
                   CRWT=19.000*SPACFRAC
                   MNWT=0.350*SPACFRAC
                   FEWT=16.809*SPACFRAC
                   NIWT=52.500*SPACFRAC
                   ALWT=0.500*SPACFRAC
```

```
                  TIWT=0.900*SPACFRAC
                  COWT=1.000*SPACFRAC
                  CUWT=0.300*SPACFRAC
                  NBWT=2.5625*SPACFRAC
                  MOWT=3.050*SPACFRAC
                  TAWT=2.5625*SPACFRAC
                  WRITE(200,2792) HOMOSPACERML, (-1*HWT),
     c            SPN, ASSYID(COLUMN,ROW)
2792              FORMAT(T1,'M',I4,T9,' 1001.50c'3X,G14.8,
     c '    $ Homogenized Inconel Spacer Number ',I2,' in Assembly ',A5)
                  WRITE(200,2793) -1*OWT
2793              FORMAT(T9,' 8016.50c',3X,G14.8)
                  WRITE(200,2794) -1*B10WT
2794              FORMAT(T9,' 5010.50c',3X,G14.8)
                  WRITE(200,2795) -1*B11WT
2795              FORMAT(T9,' 5011.56c',3X,G14.8)
                  WRITE(200,2796) -1*CWT
2796              FORMAT(T9,'6000.50c',3X,G14.8)
                  WRITE(200,2797) -1*SIWT
2797              FORMAT(T9,'14000.50c',3X,G14.8)
                  WRITE(200,2798) -1*PWT
2798              FORMAT(T9,'15031.50c',3X,G14.8)
                  WRITE(200,2799) -1*SWT
2799              FORMAT(T9,'16032.50c',3X,G14.8)
                  WRITE(200,2800) -1*CRWT*CR50
2800              FORMAT(T9,'24050.60c',3X,G14.8)
                  WRITE(200,7066) -1*CRWT*CR52
7066              FORMAT(T9,'24052.60c',3X,G14.8)
                  WRITE(200,7067) -1*CRWT*CR53
7067              FORMAT(T9,'24053.60c',3X,G14.8)
                  WRITE(200,7068) -1*CRWT*CR54
7068              FORMAT(T9,'24054.60c',3X,G14.8)
                  WRITE(200,2801) -1*MNWT
2801              FORMAT(T9,'25055.50c',3X,G14.8)
                  WRITE(200,2802) -1*FEWT*FE54
2802              FORMAT(T9,'26054.60c',3X,G14.8)
                  WRITE(200,7069) -1*FEWT*FE56
7069              FORMAT(T9,'26056.60c',3X,G14.8)
                  WRITE(200,7070) -1*FEWT*FE57
7070              FORMAT(T9,'26057.60c',3X,G14.8)
                  WRITE(200,7071) -1*FEWT*FE58
7071              FORMAT(T9,'26058.60c',3X,G14.8)
                  WRITE(200,2803) -1*NIWT*NI58
2803              FORMAT(T9,'28058.60c',3X,G14.8)
                  WRITE(200,7072) -1*NIWT*NI60
7072              FORMAT(T9,'28060.60c',3X,G14.8)
                  WRITE(200,7073) -1*NIWT*NI61
7073              FORMAT(T9,'28061.60c',3X,G14.8)
                  WRITE(200,7074) -1*NIWT*NI62
7074              FORMAT(T9,'28062.60c',3X,G14.8)
                  WRITE(200,7075) -1*NIWT*NI64
7075              FORMAT(T9,'28064.60c',3X,G14.8)
                  WRITE(200,2804) -1*ALWT
2804              FORMAT(T9,'13027.50c',3X,G14.8)
```

```
            WRITE(200,2805) -1*TIWT
 2805       FORMAT(T9,'22000.50c',3X,G14.8)
            WRITE(200,2806) -1*COWT
 2806       FORMAT(T9,'27059.50c',3X,G14.8)
            WRITE(200,2807) -1*CUWT*CU63
 2807       FORMAT(T9,'29063.60c',3X,G14.8)
            WRITE(200,7076) -1*CUWT*CU65
 7076       FORMAT(T9,'29065.60c',3X,G14.8)
            WRITE(200,2808) -1*NBWT
 2808       FORMAT(T9,'41093.50c',3X,G14.8)
            WRITE(200,2809) -1*MOWT
 2809       FORMAT(T9,'42000.50c',3X,G14.8)
            WRITE(200,2810) -1*TAWT
 2810       FORMAT(T9,'73181.50c',3X,G14.8)
*
* Stainless Steel/Inconel Spacer Grid Specification
*
          ELSEIF (SPACERMAT(DESNUM(COLUMN,ROW),SPN).EQ.4) THEN
            VAL1=(ASSYPITCH**2)*
     c      SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
            VAL2=RODNUM(DESNUM(COLUMN,ROW))*(PI)*
     c      (CLADRADIUS(DESNUM(COLUMN,ROW),2)**2)*
     c      SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
            IF (BANDW.EQ..TRUE.) THEN
               VAL3=16.0*(GTDATA(DESNUM(COLUMN,ROW),2)**2)*
     c         (PI)*SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
            ELSEIF ((WESTINGHOUSE.EQ..TRUE.).AND.
     c      (GTSPLIT.NE.1)) THEN
               VAL3=24.0*(GTDATA(DESNUM(COLUMN,ROW),2)**2)*
     c         (PI)*SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
            ELSEIF ((WESTINGHOUSE.EQ..TRUE.).AND.
     c      (GTSPLIT.EQ.1)) THEN
               DO 2765 SCTN=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
                  IF ((SURFVALUESPEC(SPACERBOTTOMSURF).LT.
     c            GTAXDATA(DESNUM(COLUMN,ROW),3,SCTN)).AND.
     c            (SURFVALUESPEC(SPACERBOTTOMSURF).GE.
     c            GTAXDATA(DESNUM(COLUMN,ROW),4,SCTN))) THEN
                     VAL3=24.0*
     c               (GTAXDATA(DESNUM(COLUMN,ROW),2,SCTN)**2)*
     c               (PI)*SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                  ENDIF
 2765          CONTINUE
            ENDIF
            VAL4=(ITDATA(DESNUM(COLUMN,ROW),2)**2)*(PI)*
     c      SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
            TVOL=VAL1-VAL2-VAL3-VAL4
            SPACVOL=SPACERVOL(DESNUM(COLUMN,ROW),SPN)
            MODVOL=TVOL-SPACVOL
            HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)=
     c      ((SPACVOL*SPM4INC(DESNUM(COLUMN,ROW),SPN)*8.19)+
     c      (SPACVOL*SPM4SS(DESNUM(COLUMN,ROW),SPN)*7.90)+
     c      (MODVOL*MODDENSITY))/(MODVOL+SPACVOL)
            SPACMASS=SPACVOL*((SPM4INC(DESNUM(COLUMN,ROW),SPN)*
     c      8.19)+(SPM4SS(DESNUM(COLUMN,ROW),SPN)*7.90))
```

```
                  SPINCMASS=SPACVOL*(SPM4INC(DESNUM(COLUMN,ROW),SPN)*
     c            8.19)
                  SPSSMASS=SPACVOL*(SPM4SS(DESNUM(COLUMN,ROW),SPN)*
     c            7.90)
                  MODMASS=MODVOL*MODDENSITY
                  SPACFRAC=SPACMASS/(SPACMASS+MODMASS)
                  MODFRAC=MODMASS/(SPACMASS+MODMASS)
                  SPINCFRAC=SPINCMASS/(SPACMASS+MODMASS)
                  SPSSFRAC=SPSSMASS/(SPACMASS+MODMASS)
           BWTINH2O=((PPMB*1E-6)/(1.0+(PPMB*1E-6)))*100.0
           HWTINH2O=((1.008664904*0.999167*2.0)/
     c     ((1.008664904*0.999167*2.0)+
     c     (1.008664904*15.857510)))*(100.0-BWTINH2O)
           OWTINH2O=((1.008664904*15.857510)/
     c     ((1.008664904*0.999167*2.0)+
     c     (1.008664904*15.857510)))*(100.0-BWTINH2O)
           B10WTINH2O=((1.008664904*9.926922*0.194)/
     c     ((1.008664904*9.926922*0.194)+
     c     (1.008664904*10.914730*0.806)))*BWTINH2O
           B11WTINH2O=((1.008664904*10.914730*0.806)/
     c     ((1.008664904*9.926922*0.194)+
     c     (1.008664904*10.914730*0.806)))*BWTINH2O
                  B10WTINB=((1.008664904*9.926922*0.194)/
     c            ((1.008664904*9.926922*0.194)+
     c            (1.008664904*10.914730*0.806)))*100.0
                  B11WTINB=((1.008664904*10.914730*0.806)/
     c            ((1.008664904*9.926922*0.194)+
     c            (1.008664904*10.914730*0.806)))*100.0
                  OWT=OWTINH2O*MODFRAC
                  HWT=HWTINH2O*MODFRAC
                  B10WT=(B10WTINH2O*MODFRAC)+
     c            (B10WTINB*0.00006*SPINCFRAC)
                  B11WT=(B11WTINH2O*MODFRAC)+
     c            (B11WTINB*0.00006*SPINCFRAC)
                  CWT=(0.080*SPINCFRAC)+(0.080*SPSSFRAC)
                  NWT=(0.100*SPSSFRAC)
                  SIWT=(0.350*SPINCFRAC)+(0.750*SPSSFRAC)
                  PWT=(0.015*SPINCFRAC)+(0.045*SPSSFRAC)
                  SWT=(0.015*SPINCFRAC)+(0.030*SPSSFRAC)
                  CRWT=(19.000*SPINCFRAC)+(19.0*SPSSFRAC)
                  MNWT=(0.350*SPINCFRAC)+(2.0*SPSSFRAC)
                  FEWT=(16.809*SPINCFRAC)+(68.745*SPSSFRAC)
                  NIWT=(52.500*SPINCFRAC)+(9.250*SPSSFRAC)
                  ALWT=(0.500*SPINCFRAC)
                  TIWT=(0.900*SPINCFRAC)
                  COWT=(1.000*SPINCFRAC)
                  CUWT=(0.300*SPINCFRAC)
                  NBWT=(2.5625*SPINCFRAC)
                  MOWT=(3.050*SPINCFRAC)
                  TAWT=(2.5625*SPINCFRAC)
                  WRITE(200,8000) HOMOSPACERML, (-1*HWT),
     c            SPN, ASSYID(COLUMN,ROW)
8000              FORMAT(T1,'M',I4,T9,' 1001.50c'3X,G14.8,
     c '     $ Homogenized SS304 & Inconel Spacer Number ',
```

```
     c I2,' in Assembly ',A5)
                 WRITE(200,8001) -1*OWT
 8001            FORMAT(T9,' 8016.50c',3X,G14.8)
                 WRITE(200,8002) -1*B10WT
 8002            FORMAT(T9,' 5010.50c',3X,G14.8)
                 WRITE(200,8003) -1*B11WT
 8003            FORMAT(T9,' 5011.56c',3X,G14.8)
                 WRITE(200,8004) -1*CWT
 8004            FORMAT(T9,'6000.50c',3X,G14.8)
                 WRITE(200,8005) -1*NWT
 8005            FORMAT(T9,'7014.50c',3X,G14.8)
                 WRITE(200,8006) -1*SIWT
 8006            FORMAT(T9,'14000.50c',3X,G14.8)
                 WRITE(200,8007) -1*PWT
 8007            FORMAT(T9,'15031.50c',3X,G14.8)
                 WRITE(200,8008) -1*SWT
 8008            FORMAT(T9,'16032.50c',3X,G14.8)
                 WRITE(200,8009) -1*CRWT*CR50
 8009            FORMAT(T9,'24050.60c',3X,G14.8)
                 WRITE(200,8010) -1*CRWT*CR52
 8010            FORMAT(T9,'24052.60c',3X,G14.8)
                 WRITE(200,8011) -1*CRWT*CR53
 8011            FORMAT(T9,'24053.60c',3X,G14.8)
                 WRITE(200,8012) -1*CRWT*CR54
 8012            FORMAT(T9,'24054.60c',3X,G14.8)
                 WRITE(200,8013) -1*MNWT
 8013            FORMAT(T9,'25055.50c',3X,G14.8)
                 WRITE(200,8014) -1*FEWT*FE54
 8014            FORMAT(T9,'26054.60c',3X,G14.8)
                 WRITE(200,8015) -1*FEWT*FE56
 8015            FORMAT(T9,'26056.60c',3X,G14.8)
                 WRITE(200,8016) -1*FEWT*FE57
 8016            FORMAT(T9,'26057.60c',3X,G14.8)
                 WRITE(200,8017) -1*FEWT*FE58
 8017            FORMAT(T9,'26058.60c',3X,G14.8)
                 WRITE(200,8018) -1*NIWT*NI58
 8018            FORMAT(T9,'28058.60c',3X,G14.8)
                 WRITE(200,8019) -1*NIWT*NI60
 8019            FORMAT(T9,'28060.60c',3X,G14.8)
                 WRITE(200,8020) -1*NIWT*NI61
 8020            FORMAT(T9,'28061.60c',3X,G14.8)
                 WRITE(200,8021) -1*NIWT*NI62
 8021            FORMAT(T9,'28062.60c',3X,G14.8)
                 WRITE(200,8022) -1*NIWT*NI64
 8022            FORMAT(T9,'28064.60c',3X,G14.8)
                 WRITE(200,8023) -1*ALWT
 8023            FORMAT(T9,'13027.50c',3X,G14.8)
                 WRITE(200,8024) -1*TIWT
 8024            FORMAT(T9,'22000.50c',3X,G14.8)
                 WRITE(200,8025) -1*COWT
 8025            FORMAT(T9,'27059.50c',3X,G14.8)
                 WRITE(200,8026) -1*CUWT*CU63
 8026            FORMAT(T9,'29063.60c',3X,G14.8)
                 WRITE(200,8027) -1*CUWT*CU65
```

```
 8027          FORMAT(T9,'29065.60c',3X,G14.8)
               WRITE(200,8028) -1*NBWT
 8028          FORMAT(T9,'41093.50c',3X,G14.8)
               WRITE(200,8029) -1*MOWT
 8029          FORMAT(T9,'42000.50c',3X,G14.8)
               WRITE(200,8030) -1*TANT
 8030          FORMAT(T9,'73181.50c',3X,G14.8)
*
* Stainless Steel/Zircaloy Spacer Grid Specification
*
            ELSEIF (SPACERMAT(DESNUM(COLUMN,ROW),SPN).EQ.5) THEN
               VAL1=(ASSYPITCH**2)*
     c         SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
               VAL2=RODNUM(DESNUM(COLUMN,ROW))*(PI)*
     c         (CLADRADIUS(DESNUM(COLUMN,ROW),2)**2)*
     c         SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
               IF (BANDW.EQ..TRUE.) THEN
                  VAL3=16.0*(GTDATA(DESNUM(COLUMN,ROW),2)**2)*
     c            (PI)*SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
               ELSEIF ((WESTINGHOUSE.EQ..TRUE.).AND.
     c         (GTSPLIT.NE.1)) THEN
                  VAL3=24.0*(GTDATA(DESNUM(COLUMN,ROW),2)**2)*
     c            (PI)*SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
               ELSEIF ((WESTINGHOUSE.EQ..TRUE.).AND.
     c         (GTSPLIT.EQ.1)) THEN
                  DO 2766 SCTN=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
                     IF ((SURFVALUESPEC(SPACERBOTTOMSURF).LT.
     c               GTAXDATA(DESNUM(COLUMN,ROW),3,SCTN)).AND.
     c               (SURFVALUESPEC(SPACERBOTTOMSURF).GE.
     c               GTAXDATA(DESNUM(COLUMN,ROW),4,SCTN))) THEN
                        VAL3=24.0*
     c                  (GTAXDATA(DESNUM(COLUMN,ROW),2,SCTN)**2)*
     c                  (PI)*SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                     ENDIF
2766              CONTINUE
               ENDIF
               VAL4=(ITDATA(DESNUM(COLUMN,ROW),2)**2)*(PI)*
     c         SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
               TVOL=VAL1-VAL2-VAL3-VAL4
               SPACVOL=SPACERVOL(DESNUM(COLUMN,ROW),SPN)
               MODVOL=TVOL-SPACVOL
               HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)=
     c         ((SPACVOL*SPM4SS(DESNUM(COLUMN,ROW),SPN)*7.90)+
     c         (SPACVOL*SPM4ZR(DESNUM(COLUMN,ROW),SPN)*6.56)+
     c         (MODVOL*MODDENSITY))/(MODVOL+SPACVOL)
               SPACMASS=(SPACVOL*
     c         SPM4SS(DESNUM(COLUMN,ROW),SPN)*7.90)+
     c         (SPACVOL*SPM4ZR(DESNUM(COLUMN,ROW),SPN)*6.56)
               MODMASS=MODVOL*MODDENSITY
               SPACFRAC=SPACMASS/(SPACMASS+MODMASS)
               MODFRAC=MODMASS/(SPACMASS+MODMASS)
               SPSSMASS=SPACVOL*(SPM4SS(DESNUM(COLUMN,ROW),SPN)*
     c         7.90)
               SPZRMASS=SPACVOL*(SPM4ZR(DESNUM(COLUMN,ROW),SPN)*
```

```
     c          6.56)
                SPSSFRAC=SPSSMASS/(SPACMASS+MODMASS)
                SPZRFRAC=SPZRMASS/(SPACMASS+MODMASS)
          BWTINH2O=((PPMB*1E-6)/(1.0+(PPMB*1E-6)))*100.0
          HWTINH2O=((1.008664904*0.999167*2.0)/
     c    ((1.008664904*0.999167*2.0)+
     c    (1.008664904*15.857510)))*(100.0-BWTINH2O)
          OWTINH2O=((1.008664904*15.857510)/
     c    ((1.008664904*0.999167*2.0)+
     c    (1.008664904*15.857510)))*(100.0-BWTINH2O)
          B10WTINH2O=((1.008664904*9.926922*0.194)/
     c    ((1.008664904*9.926922*0.194)+
     c    (1.008664904*10.914730*0.806)))*BWTINH2O
          B11WTINH2O=((1.008664904*10.914730*0.806)/
     c    ((1.008664904*9.926922*0.194)+
     c    (1.008664904*10.914730*0.806)))*BWTINH2O
                B10WTINB=((1.008664904*9.926922*0.194)/
     c          ((1.008664904*9.926922*0.194)+
     c          (1.008664904*10.914730*0.806)))*100.0
                B11WTINB=((1.008664904*10.914730*0.806)/
     c          ((1.008664904*9.926922*0.194)+
     c          (1.008664904*10.914730*0.806)))*100.0
                OWT=(OWTINH2O*MODFRAC)+(0.120*SPZRFRAC)
                HWT=HWTINH2O*MODFRAC
                B10WT=(B10WTINH2O*MODFRAC)
                B11WT=(B11WTINH2O*MODFRAC)
                CWT=0.080*SPSSFRAC
                NWT=0.100*SPSSFRAC
                SIWT=0.750*SPSSFRAC
                PWT=0.045*SPSSFRAC
                SWT=0.030*SPSSFRAC
                CRWT=(19.000*SPSSFRAC)+(0.100*SPZRFRAC)
                MNWT=2.0*SPSSFRAC
                FEWT=(68.745*SPSSFRAC)+(0.200*SPZRFRAC)
                NIWT=9.25*SPSSFRAC
                ZRWT=98.180*SPZRFRAC
                SNWT=1.40*SPZRFRAC
                WRITE(200,8031) HOMOSPACERML, (-1*HWT),
     c          SPN, ASSYID(COLUMN,ROW)
8031            FORMAT(T1,'M',I4,T9,' 1001.50c'3X,G14.8,
     c '  $ Homogenized SS304 & Zirc-4 Spacer Number ',
     c I2,' in Assembly ',A5)
                WRITE(200,8032) -1*OWT
8032            FORMAT(T9,' 8016.50c',3X,G14.8)
                WRITE(200,8033) -1*B10WT
8033            FORMAT(T9,' 5010.50c',3X,G14.8)
                WRITE(200,8034) -1*B11WT
8034            FORMAT(T9,' 5011.56c',3X,G14.8)
                WRITE(200,8035) -1*CWT
8035            FORMAT(T9,'6000.50c',3X,G14.8)
                WRITE(200,8036) -1*NWT
8036            FORMAT(T9,'7014.50c',3X,G14.8)
                WRITE(200,8037) -1*SIWT
8037            FORMAT(T9,'14000.50c',3X,G14.8)
```

```
                WRITE(200,8038) -1*PWT
 8038           FORMAT(T9,'15031.50c',3X,G14.8)
                WRITE(200,8039) -1*SWT
 8039           FORMAT(T9,'16032.50c',3X,G14.8)
                WRITE(200,8040) -1*CRWT*CR50
 8040           FORMAT(T9,'24050.60c',3X,G14.8)
                WRITE(200,8041) -1*CRWT*CR52
 8041           FORMAT(T9,'24052.60c',3X,G14.8)
                WRITE(200,8042) -1*CRWT*CR53
 8042           FORMAT(T9,'24053.60c',3X,G14.8)
                WRITE(200,8043) -1*CRWT*CR54
 8043           FORMAT(T9,'24054.60c',3X,G14.8)
                WRITE(200,8044) -1*MNWT
 8044           FORMAT(T9,'25055.50c',3X,G14.8)
                WRITE(200,8045) -1*FEWT*FE54
 8045           FORMAT(T9,'26054.60c',3X,G14.8)
                WRITE(200,8046) -1*FEWT*FE56
 8046           FORMAT(T9,'26056.60c',3X,G14.8)
                WRITE(200,8047) -1*FEWT*FE57
 8047           FORMAT(T9,'26057.60c',3X,G14.8)
                WRITE(200,8048) -1*FEWT*FE58
 8048           FORMAT(T9,'26058.60c',3X,G14.8)
                WRITE(200,8049) -1*NIWT*NI58
 8049           FORMAT(T9,'28058.60c',3X,G14.8)
                WRITE(200,8050) -1*NIWT*NI60
 8050           FORMAT(T9,'28060.60c',3X,G14.8)
                WRITE(200,8051) -1*NIWT*NI61
 8051           FORMAT(T9,'28061.60c',3X,G14.8)
                WRITE(200,8052) -1*NIWT*NI62
 8052           FORMAT(T9,'28062.60c',3X,G14.8)
                WRITE(200,8053) -1*NIWT*NI64
 8053           FORMAT(T9,'28064.60c',3X,G14.8)
                WRITE(200,8054) -1*ZRWT
 8054           FORMAT(T9,'40093.50c',3X,G14.8)
                WRITE(200,8055) -1*SNWT
 8055           FORMAT(T9,'50000.35c',3X,G14.8)
             ENDIF
             WRITE(200,2838) HOMOSPACERML
 2838        FORMAT(T1,'MT',I4,T9,'LWTR.03T')
             ENDIF
*     Write the current homogenized spacer region cell in this fuel rod
universe.
             IF ((SURFVALUESPEC(PLENUMTOPSURF).LT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.(SPN.EQ.1)) THEN
                WRITE(30,2839) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c          (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)), CLADORSURF,
     c          (-1*SPACERTOPSURF), SPACERBOTTOMSURF, FRUNIV(COLUMN,ROW),
     c          SPN
 2839           FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '    $ Homogenized region for spacer ',I2)
                LN=LN+1
                WRITE(30,2840) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c          (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
```

```
     c          (-1*CLADORSURF),
     c          (-1*UEFBOTTOMSURF), PLENUMTOPSURF, FRUNIV(COLUMN,ROW),
     c          SPN
 2840           FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '   $ Homogenized region for spacer ',I2)
                LN=LN+1
              ELSE
                WRITE(30,2841) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c          (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)), CLADORSURF,
     c          (-1*SPACERTOPSURF), SPACERBOTTOMSURF, FRUNIV(COLUMN,ROW),
     c          SPN
 2841           FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '   $ Homogenized region for spacer ',I2)
                LN=LN+1
              ENDIF
*    Write the water region cell below the current homogenized spacer cell
in this fuel rod universe.
              WRITE(30,2842) LN, BMODML, (-1*MODDENSITY), CLADORSURF,
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        FRUNIV(COLUMN,ROW)
 2842         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Borated moderator region')
              LN=LN+1
 2843       CONTINUE
        ENDIF
 2844     CONTINUE
 2845 CONTINUE
*  Write the specifications for the GT universes that are
*  required to fill the assembly layout specifications previously defined.
        DO 3340 ROW=1,50
          DO 3330 COLUMN=1,50
*    Write the GT universe specification if the assembly design is unique.
*
        IF (GTWRITE(COLUMN,ROW).EQ..TRUE.) THEN
*  Write the BPR specification header.
          WRITE(30,2846)
 2846     FORMAT(T1,'C')
          WRITE(30,2847) ASSYID(COLUMN,ROW)
 2847     FORMAT(T1,
    c     'C   GUIDE TUBE UNIVERSE SPECIFICATION FOR ASSEMBLY ',A5)
          WRITE(30,2850)
 2850     FORMAT(T1,'C')
*    Define the upper end-fitting bottom surface.
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
    c          ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 2860 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
```

```
            ENDIF
 2860     CONTINUE
          IF (CURRENTSURFLABEL.EQ.0) THEN
            UEFBOTTOMSURF=SN
            SURFTYPESPEC(SN)='PZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
          ELSE
            UEFBOTTOMSURF=CURRENTSURFLABEL
          ENDIF
*    Define the upper end-fitting top surface.
          CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
    c     ENDFITHEIGHT(DESNUM(COLUMN,ROW),1)+
    c     ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
          CURRENTSURFLABEL=0
          DO 2870 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'PZ') THEN
        IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                CURRENTSURFLABEL=V
                EXIT
              ENDIF
            ENDIF
 2870     CONTINUE
          IF (CURRENTSURFLABEL.EQ.0) THEN
            UEFTOPSURF=SN.
            SURFTYPESPEC(SN)='PZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
          ELSE
            UEFTOPSURF=CURRENTSURFLABEL
          ENDIF
       IF (GTSPLIT.NE.1) THEN
*    Define the GT top surface.
          CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),3)
          IF (CURRENTSURF.GT.SURFVALUESPEC(UEFTOPSURF)) THEN
            CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
          ENDIF
          CURRENTSURFLABEL=0
          DO 2880 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'PZ') THEN
        IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                CURRENTSURFLABEL=V
                EXIT
              ENDIF
            ENDIF
 2880     CONTINUE
          IF (CURRENTSURFLABEL.EQ.0) THEN
            GTTOPSURF=SN
            SURFTYPESPEC(SN)='PZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
          ELSE
            GTTOPSURF=CURRENTSURFLABEL
          ENDIF
```

```
*     Define the GT bottom surface.
            CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),4)
            CURRENTSURFLABEL=0
            DO 2890 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 2890       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               GTBOTSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               GTBOTSURF=CURRENTSURFLABEL
            ENDIF
*     Define the GT outer radius surface.
            CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),2)
            CURRENTSURFLABEL=0
            DO 2900 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'CZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 2900       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               GTORSURF=SN
               SURFTYPESPEC(SN)='CZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               GTORSURF=CURRENTSURFLABEL
            ENDIF
*     Define the GT inner radius surface.
            CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),1)
            CURRENTSURFLABEL=0
            DO 2910 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'CZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 2910       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               GTIRSURF=SN
               SURFTYPESPEC(SN)='CZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
```

```
                    ELSE
                       GTIRSURF=CURRENTSURFLABEL
                    ENDIF
                 ELSEIF (GTSPLIT.EQ.1) THEN
*     Define the guide tube axial sections for this design.
                    DO 2920 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
*     Define the GT section top surface.
                    CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),3,SECT)
                    IF (CURRENTSURF.GT.SURFVALUESPEC(UEFTOPSURF)) THEN
                       CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                    ENDIF
                    CURRENTSURFLABEL=0
                    DO 2912 V=1,(SN-1)
                       IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                  IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                             CURRENTSURFLABEL=V
                             EXIT
                          ENDIF
                       ENDIF
 2912               CONTINUE
                    IF (CURRENTSURFLABEL.EQ.0) THEN
                       GTSECTTOPSURF(SECT)=SN
                       SURFTYPESPEC(SN)='PZ'
                       SURFVALUESPEC(SN)=CURRENTSURF
                       SN=SN+1
                    ELSE
                       GTSECTTOPSURF(SECT)=CURRENTSURFLABEL
                    ENDIF
*     Define the GT section bottom surface.
                    CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),4,SECT)
                    CURRENTSURFLABEL=0
                    DO 2914 V=1,(SN-1)
                       IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                  IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                             CURRENTSURFLABEL=V
                             EXIT
                          ENDIF
                       ENDIF
 2914               CONTINUE
                    IF (CURRENTSURFLABEL.EQ.0) THEN
                       GTSECTBOTSURF(SECT)=SN
                       SURFTYPESPEC(SN)='PZ'
                       SURFVALUESPEC(SN)=CURRENTSURF
                       SN=SN+1
                    ELSE
                       GTSECTBOTSURF(SECT)=CURRENTSURFLABEL
                    ENDIF
*     Define the GT section outer radius surface.
                    CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),2,SECT)
                    CURRENTSURFLABEL=0
                    DO 2916 V=1,(SN-1)
                       IF (SURFTYPESPEC(V).EQ.'CZ') THEN
                  IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                             CURRENTSURFLABEL=V
```

```
                    EXIT
                 ENDIF
              ENDIF
2916       CONTINUE
           IF (CURRENTSURFLABEL.EQ.0) THEN
              GTSECTORSURF(SECT)=SN
              SURFTYPESPEC(SN)='CZ'
              SURFVALUESPEC(SN)=CURRENTSURF
              SN=SN+1
           ELSE
              GTSECTORSURF(SECT)=CURRENTSURFLABEL
           ENDIF
*     Define the GT section inner radius surface.
           CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),1,SECT)
           CURRENTSURFLABEL=0
           DO 2918 V=1,(SN-1)
              IF (SURFTYPESPEC(V).EQ.'CZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                 ENDIF
              ENDIF
2918       CONTINUE
           IF (CURRENTSURFLABEL.EQ.0) THEN
              GTSECTIRSURF(SECT)=SN
              SURFTYPESPEC(SN)='CZ'
              SURFVALUESPEC(SN)=CURRENTSURF
              SN=SN+1
           ELSE
              GTSECTIRSURF(SECT)=CURRENTSURFLABEL
           ENDIF
2920       CONTINUE
        ENDIF
*     Define the lower end-fitting top surface.
           CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
           CURRENTSURFLABEL=0
           DO 2922 V=1,(SN-1)
              IF (SURFTYPESPEC(V).EQ.'PZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                 ENDIF
              ENDIF
2922       CONTINUE
           IF (CURRENTSURFLABEL.EQ.0) THEN
              LEFTOPSURF=SN
              SURFTYPESPEC(SN)='PZ'
              SURFVALUESPEC(SN)=CURRENTSURF
              SN=SN+1
           ELSE
              LEFTOPSURF=CURRENTSURFLABEL
           ENDIF
*     Write the lower end-fitting cell specification for this empty GT
universe.
```

```
*     Determine if the lower end-fitting material specification has
*     previously been defined.  If it has been previously defined, determine
*     the lower end-fitting material specification label.
            FRLEFMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 2931 RO=1,(ROW-1)
                  DO 2930 CO=1,50
                        IF (GTWRITE(COLUMN,ROW).EQ.GTWRITE(CO,RO)) THEN
                           FRLEFMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTLEFML(COLUMN,ROW)=GTLEFML(CO,RO)
                           EXIT
                        ENDIF
 2930             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 2931          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 2933 RO=ROW,ROW
                     DO 2932 CO=1,(COLUMN-1)
                           IF (GTWRITE(COLUMN,ROW).EQ.
     c                     GTWRITE(CO,RO)) THEN
                              FRLEFMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              GTLEFML(COLUMN,ROW)=GTLEFML(CO,RO)
                              EXIT
                           ENDIF
 2932                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 2933             CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 2935 RO=1,(ROW-1)
                  DO 2934 CO=1,50
                        IF (GTWRITE(COLUMN,ROW).EQ.
     c                     GTWRITE(CO,RO)) THEN
                           FRLEFMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTLEFML(COLUMN,ROW)=GTLEFML(CO,RO)
                           EXIT
                        ENDIF
 2934             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 2935          CONTINUE
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
               DO 2937 RO=1,1
                  DO 2936 CO=1,(COLUMN-1)
                        IF (GTWRITE(COLUMN,ROW).EQ.
```

```
     c                          GTWRITE(CO,RO)) THEN
                                FRLEFMLUNIQUE=.FALSE.
                                LEAVE=.TRUE.
                                GTLEFML(COLUMN,ROW)=GTLEFML(CO,RO)
                                EXIT
                             ENDIF
 2936                  CONTINUE
                       IF (LEAVE.EQ..TRUE.) THEN
                          EXIT
                       ENDIF
 2937               CONTINUE
                 ENDIF
                 IF (FRLEFMLUNIQUE.EQ..TRUE.) THEN
                    GTLEFML(COLUMN,ROW)=MN
*  Check Guide Tube Lower End-Fitting Regions
                       DO 2940 C=1,LEFMAT(DESNUM(COLUMN,ROW),2)
                          IF (C.EQ.1) THEN
                             WRITE(200,2938) GTLEFML(COLUMN,ROW),
     c                       LEFZAIDS(DESNUM(COLUMN,ROW),C),
     c                       (-1*LEFWTS(DESNUM(COLUMN,ROW),C))
 2938                        FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                       '    $ Guide Tube Lower End Fitting')
                          ELSE
                             WRITE(200,2939) LEFZAIDS(DESNUM(COLUMN,ROW),C),
     c                       (-1*LEFWTS(DESNUM(COLUMN,ROW),C))
 2939                        FORMAT(T9,A9,3X,G14.6)
                          ENDIF
 2940                  CONTINUE
                    WRITE(200,2941) GTLEFML(COLUMN,ROW)
 2941               FORMAT(T1,'MT',I4,T9,'LWTR.03T')
                    MN=MN+1
                 ENDIF
              IF (GTSPLIT.NE.1) THEN
                 IF (SURFVALUESPEC(GTBOTSURF).GE.
     c           ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)) THEN
                    WRITE(30,2970) LN, GTLEFML(COLUMN,ROW),
     c              (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*LEFTOPSURF),
     c              GTUNIV(COLUMN,ROW)
 2970               FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,' IMP:N=1 U=',I4,
     c              '    $ Guide tube lower end-fitting')
                    LN=LN+1
                 ELSE
                    WRITE(30,2980) LN, GTLEFML(COLUMN,ROW),
     c              (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*LEFTOPSURF),
     c              GTORSURF, GTUNIV(COLUMN,ROW)
 2980               FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c              ' IMP:N=1 U=',I4,'    $ Guide tube lower end-fitting')
                    LN=LN+1
                    WRITE(30,2990) LN, GTLEFML(COLUMN,ROW),
     c              (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*GTBOTSURF),
     c              (-1*GTORSURF), GTUNIV(COLUMN,ROW)
 2990               FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c              ' IMP:N=1 U=',I4,'    $ Guide tube lower end-fitting')
                    LN=LN+1
```

```
            ENDIF
         ELSEIF (GTSPLIT.EQ.1) THEN
         LGTS=GTSECTBOTSURF(1)
         LGTSOR=GTSECTORSURF(1)
         DO 2992 SECT=2,NUMOFGTAXS(DESNUM(COLUMN,ROW))
            IF (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c      SURFVALUESPEC(LGTS)) THEN
               LGTS=GTSECTBOTSURF(SECT)
               LGTSOR=GTSECTORSURF(SECT)
            ENDIF
 2992    CONTINUE
            IF (SURFVALUESPEC(LGTS).GE.
     c      ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)) THEN
               WRITE(30,2994) LN, GTLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*LEFTOPSURF),
     c         GTUNIV(COLUMN,ROW)
 2994          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,' IMP:N=1 U=',I4,
     c         '   $ Guide tube lower end-fitting')
               LN=LN+1
            ELSE
               WRITE(30,2996) LN, GTLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*LEFTOPSURF),
     c         LGTSOR, GTUNIV(COLUMN,ROW)
 2996          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I4,'   $ Guide tube lower end-fitting')
               LN=LN+1
               WRITE(30,2998) LN, GTLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*LGTS),
     c         (-1*LGTSOR), GTUNIV(COLUMN,ROW)
 2998          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I4,'   $ Guide tube lower end-fitting')
               LN=LN+1
            ENDIF
         ENDIF
*     Write the upper end-fitting cell specification for this empty GT
universe.
*     Determine if the GT universe upper end-fitting material specification
has
*     previously been defined.  If it has been previously defined, determine
*     the upper end-fitting material specification label.
            FRUEFMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 3001 RO=1,(ROW-1)
                  DO 3000 CO=1,50
                        IF (GTWRITE(COLUMN,ROW).EQ.GTWRITE(CO,RO)) THEN
                           FRUEFMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTUEFML(COLUMN,ROW)=GTUEFML(CO,RO)
                           EXIT
                        ENDIF
 3000             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
```

```
                ENDIF
 3001        CONTINUE
             IF (LEAVE.EQ..FALSE.) THEN
                DO 3003 RO=1,(ROW-1)
                   DO 3002 CO=1,50
                         IF (GTWRITE(COLUMN,ROW).EQ.
     c                   GTWRITE(CO,RO)) THEN
                            FRUEFMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            GTUEFML(COLUMN,ROW)=GTUEFML(CO,RO)
                            EXIT
                         ENDIF
 3002              CONTINUE
                   IF (LEAVE.EQ..TRUE.) THEN
                      EXIT
                   ENDIF
 3003           CONTINUE
             ENDIF
          ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
             DO 3005 RO=1,(ROW-1)
                DO 3004 CO=1,50
                      IF (GTWRITE(COLUMN,ROW).EQ.
     c                   GTWRITE(CO,RO)) THEN
                         FRUEFMLUNIQUE=.FALSE.
                         LEAVE=.TRUE.
                         GTUEFML(COLUMN,ROW)=GTUEFML(CO,RO)
                         EXIT
                      ENDIF
 3004           CONTINUE
                IF (LEAVE.EQ..TRUE.) THEN
                   EXIT
                ENDIF
 3005        CONTINUE
          ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
             DO 3007 RO=1,1
                DO 3006 CO=1,(COLUMN-1)
                      IF (GTWRITE(COLUMN,ROW).EQ.
     c                   GTWRITE(CO,RO)) THEN
                         FRUEFMLUNIQUE=.FALSE.
                         LEAVE=.TRUE.
                         GTUEFML(COLUMN,ROW)=GTUEFML(CO,RO)
                         EXIT
                      ENDIF
 3006           CONTINUE
                IF (LEAVE.EQ..TRUE.) THEN
                   EXIT
                ENDIF
 3007        CONTINUE
          ENDIF
          IF (FRUEFMLUNIQUE.EQ..TRUE.) THEN
             GTUEFML(COLUMN,ROW)=MN
*  Check Guide Tube Upper End-Fitting Regions
                DO 3010 C=1,UEFMAT(DESNUM(COLUMN,ROW),2)
                   IF (C.EQ.1) THEN
```

```
                      WRITE(200,3008) GTUEFML(COLUMN,ROW),
     c                UEFZAIDS(DESNUM(COLUMN,ROW),C),
     c                (-1*UEFWTS(DESNUM(COLUMN,ROW),C))
 3008                 FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                '    $ Guide Tube Upper End Fitting')
                   ELSE
                      WRITE(200,3009) UEFZAIDS(DESNUM(COLUMN,ROW),C),
     c                (-1*UEFWTS(DESNUM(COLUMN,ROW),C))
 3009                 FORMAT(T9,A9,3X,G14.6)
                   ENDIF
 3010           CONTINUE
             WRITE(200,3011) GTUEFML(COLUMN,ROW)
 3011        FORMAT(T1,'MT',I4,T9,'LWTR.03T')
             MN=MN+1
          ENDIF
       IF (GTSPLIT.NE.1) THEN
          IF ((SURFVALUESPEC(GTTOPSURF).LE.
     c    SURFVALUESPEC(UEFBOTTOMSURF))) THEN
             WRITE(30,3012) LN, GTUEFML(COLUMN,ROW),
     c       (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c       (-1*UEFTOPSURF), GTUNIV(COLUMN,ROW)
 3012        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'    $ Guide tube upper end-fitting')
             LN=LN+1
          ELSEIF ((SURFVALUESPEC(GTTOPSURF).GT.
     c    SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c    (SURFVALUESPEC(GTTOPSURF).LT.
     c    SURFVALUESPEC(UEFTOPSURF))) THEN
             WRITE(30,3013) LN, GTUEFML(COLUMN,ROW),
     c       (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c       (-1*UEFTOPSURF), GTORSURF, GTUNIV(COLUMN,ROW)
 3013        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'    $ Guide tube upper end-fitting')
             LN=LN+1
             WRITE(30,3014) LN, GTUEFML(COLUMN,ROW),
     c       (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c       (-1*UEFTOPSURF), (-1*GTORSURF),
     c       GTUNIV(COLUMN,ROW)
 3014        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'    $ Guide tube upper end-fitting')
             LN=LN+1
          ELSEIF (SURFVALUESPEC(GTTOPSURF).EQ.
     c    SURFVALUESPEC(UEFTOPSURF)) THEN
             WRITE(30,3015) LN, GTUEFML(COLUMN,ROW),
     c       (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c       (-1*UEFTOPSURF), GTORSURF, GTUNIV(COLUMN,ROW)
 3015        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'    $ Guide tube upper end-fitting')
             LN=LN+1
          ENDIF
       ELSEIF (GTSPLIT.EQ.1) THEN
       UGTS=GTSECTTOPSURF(1)
       UGTSOR=GTSECTORSURF(1)
       DO 3016 SECT=2,NUMOFGTAXS(DESNUM(COLUMN,ROW))
```

```
                IF (SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c          SURFVALUESPEC(UGTS)) THEN
                   UGTS=GTSECTTOPSURF(SECT)
                   UGTSOR=GTSECTORSURF(SECT)
                ENDIF
 3016        CONTINUE
                IF ((SURFVALUESPEC(UGTS).LE.
     c          SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                   WRITE(30,3017) LN, GTUEFML(COLUMN,ROW),
     c             (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c             (-1*UEFTOPSURF), GTUNIV(COLUMN,ROW)
 3017              FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c             ' IMP:N=1 U=',I4,'    $ Guide tube upper end-fitting')
                   LN=LN+1
                ELSEIF ((SURFVALUESPEC(UGTS).GT.
     c          SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c          (SURFVALUESPEC(UGTS).LT.
     c          SURFVALUESPEC(UEFTOPSURF))) THEN
                   WRITE(30,3018) LN, GTUEFML(COLUMN,ROW),
     c             (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c             (-1*UEFTOPSURF), UGTSOR, GTUNIV(COLUMN,ROW)
 3018              FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c             ' IMP:N=1 U=',I4,'    $ Guide tube upper end-fitting')
                   LN=LN+1
                   WRITE(30,3019) LN, GTUEFML(COLUMN,ROW),
     c             (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UGTS,
     c             (-1*UEFTOPSURF), (-1*UGTSOR),
     c             GTUNIV(COLUMN,ROW)
 3019              FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c             ' IMP:N=1 U=',I4,'    $ Guide tube upper end-fitting')
                   LN=LN+1
                ELSEIF (SURFVALUESPEC(GTTOPSURF).EQ.
     c          SURFVALUESPEC(UEFTOPSURF)) THEN
                   WRITE(30,3020) LN, GTUEFML(COLUMN,ROW),
     c             (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c             (-1*UEFTOPSURF), UGTSOR, GTUNIV(COLUMN,ROW)
 3020              FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c             ' IMP:N=1 U=',I4,'    $ Guide tube upper end-fitting')
                   LN=LN+1
                ENDIF
             ENDIF
*     Write the GT material cell in this GT universe.
             IF (GTSPLIT.NE.1) THEN
*     Determine if the GT material specification has
*     previously been defined.  If it has been previously defined, determine
*     the material specification label.
                CLADMLUNIQUE=.TRUE.
                LEAVE=.FALSE.
                IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                   DO 3071 RO=1,(ROW-1)
                      DO 3070 CO=1,50
                         IF ((DESNUM(CO,RO).NE.0).AND.
     c                   (BANKNUM(CO,RO).EQ.0)) THEN
                            IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
```

```
     c                   GTMAT(DESNUM(CO,RO))) THEN
                            CLADMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            GTML(COLUMN,ROW)=GTML(CO,RO)
                            EXIT
                         ENDIF
                      ENDIF
 3070              CONTINUE
                   IF (LEAVE.EQ..TRUE.) THEN
                      EXIT
                   ENDIF
 3071           CONTINUE
                IF (LEAVE.EQ..FALSE.) THEN
                   DO 3073 RO=ROW,ROW
                      DO 3072 CO=1,(COLUMN-1)
                         IF ((DESNUM(CO,RO).NE.0).AND.
     c                   (BANKNUM(CO,RO).EQ.0)) THEN
                            IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                      GTMAT(DESNUM(CO,RO))) THEN
                               CLADMLUNIQUE=.FALSE.
                               LEAVE=.TRUE.
                               GTML(COLUMN,ROW)=GTML(CO,RO)
                               EXIT
                            ENDIF
                         ENDIF
 3072                 CONTINUE
                      IF (LEAVE.EQ..TRUE.) THEN
                         EXIT
                      ENDIF
 3073              CONTINUE
                ENDIF
             ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                DO 3075 RO=1,(ROW-1)
                   DO 3074 CO=1,50
                      IF ((DESNUM(CO,RO).NE.0).AND.
     c                (BANKNUM(CO,RO).EQ.0)) THEN
                         IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                   GTMAT(DESNUM(CO,RO))) THEN
                            CLADMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            GTML(COLUMN,ROW)=GTML(CO,RO)
                            EXIT
                         ENDIF
                      ENDIF
 3074              CONTINUE
                   IF (LEAVE.EQ..TRUE.) THEN
                      EXIT
                   ENDIF
 3075           CONTINUE
             ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                DO 3077 RO=1,1
                   DO 3076 CO=1,(COLUMN-1)
                      IF ((DESNUM(CO,RO).NE.0).AND.
     c                (BANKNUM(CO,RO).EQ.0)) THEN
```

```
                        IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                  GTMAT(DESNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTML(COLUMN,ROW)=GTML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 3076             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 3077          CONTINUE
            ENDIF
            IF (CLADMLUNIQUE.EQ..TRUE.) THEN
               GTML(COLUMN,ROW)=MN
*  Check Guide Tube Material
                  IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.1) THEN
                     DO 3080 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9300) GTML(COLUMN,ROW)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9301)
                           WRITE(200,7000)
                           WRITE(200,7001)
                           WRITE(200,7002)
                           WRITE(200,9302)
                           WRITE(200,7003)
                           WRITE(200,7004)
                           WRITE(200,7005)
                           WRITE(200,9303)
                           WRITE(200,9304)
                        ENDIF
 3080                CONTINUE
                  ELSEIF (GTMAT(DESNUM(COLUMN,ROW))
     c            .EQ.2) THEN
                     DO 3082 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9305) GTML(COLUMN,ROW)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9306)
                           WRITE(200,9307)
                           WRITE(200,9308)
                           WRITE(200,9309)
                           WRITE(200,9310)
                           WRITE(200,7006)
                           WRITE(200,7007)
                           WRITE(200,7008)
                           WRITE(200,9311)
                           WRITE(200,9312)
                           WRITE(200,7009)
                           WRITE(200,7010)
                           WRITE(200,7011)
                           WRITE(200,9313)
```

```
                  WRITE(200,7012)
                  WRITE(200,7013)
                  WRITE(200,7014)
                  WRITE(200,7015)
               ENDIF
3082        CONTINUE
         ELSEIF (GTMAT(DESNUM(COLUMN,ROW))
     c   .EQ.3) THEN
            DO 3084 C=1,2
               IF (C.EQ.1) THEN
                  WRITE(200,9314) GTML(COLUMN,ROW)
               ELSEIF (C.EQ.2) THEN
                  WRITE(200,9315)
                  WRITE(200,9316)
                  WRITE(200,9317)
                  WRITE(200,9318)
                  WRITE(200,7016)
                  WRITE(200,7017)
                  WRITE(200,7018)
                  WRITE(200,9319)
                  WRITE(200,9320)
                  WRITE(200,7019)
                  WRITE(200,7020)
                  WRITE(200,7021)
                  WRITE(200,9321)
                  WRITE(200,7022)
                  WRITE(200,7023)
                  WRITE(200,7024)
                  WRITE(200,7025)
                  WRITE(200,9322)
                  WRITE(200,9323)
                  WRITE(200,9324)
                  WRITE(200,9325)
                  WRITE(200,9326)
                  WRITE(200,9327)
                  WRITE(200,7026)
                  WRITE(200,9328)
                  WRITE(200,9329)
                  WRITE(200,9330)
               ENDIF
3084        CONTINUE
         ENDIF
      MN=MN+1
   ENDIF
   IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.1) THEN
      CLADRHO=6.56
   ELSEIF (GTMAT(DESNUM(COLUMN,ROW)).EQ.2) THEN
      CLADRHO=7.90
   ELSEIF (GTMAT(DESNUM(COLUMN,ROW)).EQ.3) THEN
      CLADRHO=8.19
   ENDIF
   WRITE(30,3110) LN, GTML(COLUMN,ROW), (-1*CLADRHO),
c  GTIRSURF,
c  (-1*GTORSURF), (-1*GTTOPSURF), GTBOTSURF,
```

```
     c        GTUNIV(COLUMN,ROW)
 3110         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'   $ Guide tube')
              LN=LN+1
*     Write the moderator cells within the GT in this GT universe.
              WRITE(30,3112) LN, BMODML, (-1*MODDENSITY),
     c        (-1*GTIRSURF), (-1*GTTOPSURF), GTBOTSURF,
     c        GTUNIV(COLUMN,ROW)
 3112         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,
     c        '    $ Borated moderator inside guide tube')
              LN=LN+1
          ELSEIF (GTSPLIT.EQ.1) THEN
           DO 3118 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
*     Determine if the GT material specification has
*     previously been defined.  If it has been previously defined, determine
*     the material specification label.
              CLADMLUNIQUE=.TRUE.
              LEAVE=.FALSE.
              IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                 DO 3031 RO=1,(ROW-1)
                    DO 3030 CO=1,50
                       IF ((DESNUM(CO,RO).NE.0).AND.
     c                 (BANKNUM(CO,RO).EQ.0)) THEN
                          IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                    GTMAT(DESNUM(CO,RO))) THEN
                             CLADMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                             EXIT
                          ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                    GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                             CLADMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                             EXIT
                          ENDIF
                       ENDIF
 3030               CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
 3031            CONTINUE
                 IF (LEAVE.EQ..FALSE.) THEN
                    DO 3033 RO=ROW,ROW
                       DO 3032 CO=1,(COLUMN-1)
                          IF ((DESNUM(CO,RO).NE.0).AND.
     c                    (BANKNUM(CO,RO).EQ.0)) THEN
                             IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                       GTMAT(DESNUM(CO,RO))) THEN
                                CLADMLUNIQUE=.FALSE.
                                LEAVE=.TRUE.
                                GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                                EXIT
```

```
                         ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                   GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                            CLADMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                            EXIT
                         ENDIF
                      ENDIF
 3032              CONTINUE
                   IF (LEAVE.EQ..TRUE.) THEN
                      EXIT
                   ENDIF
 3033           CONTINUE
             ENDIF
          ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
             DO 3035 RO=1,(ROW-1)
                DO 3034 CO=1,50
                   IF ((DESNUM(CO,RO).NE.0).AND.
     c                (BANKNUM(CO,RO).EQ.0)) THEN
                      IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                GTMAT(DESNUM(CO,RO))) THEN
                         CLADMLUNIQUE=.FALSE.
                         LEAVE=.TRUE.
                         GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                         EXIT
                      ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                         CLADMLUNIQUE=.FALSE.
                         LEAVE=.TRUE.
                         GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                         EXIT
                      ENDIF
                   ENDIF
 3034           CONTINUE
                IF (LEAVE.EQ..TRUE.) THEN
                   EXIT
                ENDIF
 3035        CONTINUE
          ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
             DO 3037 RO=1,1
                DO 3036 CO=1,(COLUMN-1)
                   IF ((DESNUM(CO,RO).NE.0).AND.
     c                (BANKNUM(CO,RO).EQ.0)) THEN
                      IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                GTMAT(DESNUM(CO,RO))) THEN
                         CLADMLUNIQUE=.FALSE.
                         LEAVE=.TRUE.
                         GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                         EXIT
                      ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                         CLADMLUNIQUE=.FALSE.
                         LEAVE=.TRUE.
                         GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
```

```
                              EXIT
                           ENDIF
                        ENDIF
 3036                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 3037             CONTINUE
               ENDIF
               IF (CLADMLUNIQUE.EQ..TRUE.) THEN
                  GTAXML(COLUMN,ROW,SECT)=MN
*  Check Guide Tube Material
                     IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.1) THEN
                        DO 3038 C=1,2
                           IF (C.EQ.1) THEN
                              WRITE(200,9300) GTAXML(COLUMN,ROW,SECT)
                           ELSEIF (C.EQ.2) THEN
                              WRITE(200,9301)
                              WRITE(200,7000)
                              WRITE(200,7001)
                              WRITE(200,7002)
                              WRITE(200,9302)
                              WRITE(200,7003)
                              WRITE(200,7004)
                              WRITE(200,7005)
                              WRITE(200,9303)
                              WRITE(200,9304)
                           ENDIF
 3038                   CONTINUE
                     ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT)
     c               .EQ.2) THEN
                        DO 3039 C=1,2
                           IF (C.EQ.1) THEN
                              WRITE(200,9305) GTAXML(COLUMN,ROW,SECT)
                           ELSEIF (C.EQ.2) THEN
                              WRITE(200,9306)
                              WRITE(200,9307)
                              WRITE(200,9308)
                              WRITE(200,9309)
                              WRITE(200,9310)
                              WRITE(200,7006)
                              WRITE(200,7007)
                              WRITE(200,7008)
                              WRITE(200,9311)
                              WRITE(200,9312)
                              WRITE(200,7009)
                              WRITE(200,7010)
                              WRITE(200,7011)
                              WRITE(200,9313)
                              WRITE(200,7012)
                              WRITE(200,7013)
                              WRITE(200,7014)
                              WRITE(200,7015)
                           ENDIF
```

```
3039                CONTINUE
                 ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT)
    c            .EQ.3) THEN
                    DO 3040 C=1,2
                       IF (C.EQ.1) THEN
                          WRITE(200,9314) GTAXML(COLUMN,ROW,SECT)
                       ELSEIF (C.EQ.2) THEN
                          WRITE(200,9315)
                          WRITE(200,9316)
                          WRITE(200,9317)
                          WRITE(200,9318)
                          WRITE(200,7016)
                          WRITE(200,7017)
                          WRITE(200,7018)
                          WRITE(200,9319)
                          WRITE(200,9320)
                          WRITE(200,7019)
                          WRITE(200,7020)
                          WRITE(200,7021)
                          WRITE(200,9321)
                          WRITE(200,7022)
                          WRITE(200,7023)
                          WRITE(200,7024)
                          WRITE(200,7025)
                          WRITE(200,9322)
                          WRITE(200,9323)
                          WRITE(200,9324)
                          WRITE(200,9325)
                          WRITE(200,9326)
                          WRITE(200,9327)
                          WRITE(200,7026)
                          WRITE(200,9328)
                          WRITE(200,9329)
                          WRITE(200,9330)
                       ENDIF
3040                CONTINUE
                 ENDIF
              MN=MN+1
           ENDIF
           IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.1) THEN
              CLADRHO=6.56
           ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.2) THEN
              CLADRHO=7.90
           ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.3) THEN
              CLADRHO=8.19
           ENDIF
           WRITE(30,3114) LN, GTAXML(COLUMN,ROW,SECT), (-1*CLADRHO),
    c      GTSECTIRSURF(SECT),
    c      (-1*GTSECTORSURF(SECT)), (-1*GTSECTTOPSURF(SECT)),
    c      GTSECTBOTSURF(SECT), GTUNIV(COLUMN,ROW)
3114       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
    c      ' IMP:N=1 U=',I4,'   $ Guide tube')
           LN=LN+1
*     Write the moderator cells within the GT in this GT universe.
```

```
             WRITE(30,3116) LN, BMODML, (-1*MODDENSITY),
     c       (-1*GTSECTIRSURF(SECT)), (-1*GTSECTTOPSURF(SECT)),
     c       GTSECTBOTSURF(SECT), GTUNIV(COLUMN,ROW)
 3116        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,
     c       '    $ Borated moderator inside guide tube')
             LN=LN+1
 3118      CONTINUE
          ENDIF
*    Loop through the regions above the GT (i.e. the appropriate upper core
regions)
*    Define the upper region lower surface.
                DO 3228 REGION=1,NUMREGABOVE
*    Determine the current upper region's lower surface specification.
                  IF (REGION.EQ.1) THEN
                     REGIONTOPSURF=SYSTEMTOP
                     CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
     c               REGABOVE(REGION,1)
                  ENDIF
                  CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
     c            REGABOVE(REGION,1)
                  IF (REGION.EQ.NUMREGABOVE) THEN
                     REGIONBOTTOMSURF=UEFTOPSURF
                  ELSE
                  CURRENTSURFLABEL=0
                  DO 3220 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                           CURRENTSURFLABEL=V
                           EXIT
                        ENDIF
                     ENDIF
 3220             CONTINUE
                  IF (CURRENTSURFLABEL.EQ.0) THEN
                     REGIONBOTTOMSURF=SN
                     SURFTYPESPEC(SN)='PZ'
                     SURFVALUESPEC(SN)=CURRENTSURF
                     SN=SN+1
                  ELSE
                     REGIONBOTTOMSURF=CURRENTSURFLABEL
                  ENDIF
                  ENDIF
*    Write the cell specification for the GT universe upper region.
               IF (REGION.EQ.1) THEN
                  WRITE(30,3224) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c            (-1*REGABOVE(REGION,2)),
     c            REGIONBOTTOMSURF, GTUNIV(COLUMN,ROW), REGION
 3224             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
     c            ' IMP:N=1 U=',I4,'   $ Upper-core region ',I2)
                  LN=LN+1
                  REGIONTOPSURF=REGIONBOTTOMSURF
               ELSE
                  WRITE(30,3226) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c            (-1*REGABOVE(REGION,2)), (-1*REGIONTOPSURF),
```

```
     c        REGIONBOTTOMSURF, GTUNIV(COLUMN,ROW), REGION
 3226         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'   $ Upper-core region ',I2)
              LN=LN+1
              REGIONTOPSURF=REGIONBOTTOMSURF
           ENDIF
 3228      CONTINUE
         SPACHEIGHT=0.0
*    Loop through the spacer and moderator regions along the axial
*    length of the GT (from top to bottom).
       DO 3230 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
         SPACHEIGHT=SPACHEIGHT+SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
 3230  CONTINUE
       DO 3320 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
*    Define the homogenized spacer region bounding surfaces.
         IF (SPN.EQ.1) THEN
           SPACERTOPSURF=UEFBOTTOMSURF
           CURRENTSURF=SURFVALUESPEC(UEFBOTTOMSURF)-
     c     SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
           CURRENTSURFLABEL=0
           DO 3232 V=1,(SN-1)
              IF (SURFTYPESPEC(V).EQ.'PZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                 ENDIF
              ENDIF
 3232      CONTINUE
           IF (CURRENTSURFLABEL.EQ.0) THEN
              SPACERBOTTOMSURF=SN
              SURFTYPESPEC(SN)='PZ'
              SURFVALUESPEC(SN)=CURRENTSURF
              SN=SN+1
           ELSE
              SPACERBOTTOMSURF=CURRENTSURFLABEL
           ENDIF
           WATERREGIONTOPSURF=SPACERBOTTOMSURF
           CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
           CURRENTSURFLABEL=0
           DO 3234 V=1,(SN-1)
              IF (SURFTYPESPEC(V).EQ.'PZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                 ENDIF
              ENDIF
 3234      CONTINUE
           IF (CURRENTSURFLABEL.EQ.0) THEN
              WATERREGIONBOTTOMSURF=SN
              SURFTYPESPEC(SN)='PZ'
              SURFVALUESPEC(SN)=CURRENTSURF
              SN=SN+1
           ELSE
              WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
```

```
            ENDIF
         ELSEIF ((SPN.NE.1).AND.(SPN.NE.
     c   NUMOFSPACERS(DESNUM(COLUMN,ROW)))) THEN
            SPACERTOPSURF=WATERREGIONBOTTOMSURF
            CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c      SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
            CURRENTSURFLABEL=0
            DO 3236 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                   CURRENTSURFLABEL=V
                   EXIT
                 ENDIF
               ENDIF
 3236       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               SPACERBOTTOMSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               SPACERBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
            WATERREGIONTOPSURF=SPACERBOTTOMSURF
            CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
            CURRENTSURFLABEL=0
            DO 3238 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                   CURRENTSURFLABEL=V
                   EXIT
                 ENDIF
               ENDIF
 3238       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               WATERREGIONBOTTOMSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
         ELSEIF (SPN.EQ.NUMOFSPACERS(DESNUM(COLUMN,ROW))) THEN
            SPACERTOPSURF=WATERREGIONBOTTOMSURF
            CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c      SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
            CURRENTSURFLABEL=0
            DO 3240 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                   CURRENTSURFLABEL=V
                   EXIT
                 ENDIF
               ENDIF
```

```
 3240            CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
                    SPACERBOTTOMSURF=SN
                    SURFTYPESPEC(SN)='PZ'
                    SURFVALUESPEC(SN)=CURRENTSURF
                    SN=SN+1
                 ELSE
                    SPACERBOTTOMSURF=CURRENTSURFLABEL
                 ENDIF
                 WATERREGIONTOPSURF=SPACERBOTTOMSURF
                 WATERREGIONBOTTOMSURF=NODEBOTTOMSURF
              ENDIF
*     Write the current homogenized spacer region cell in this GT universe.
           IF (GTSPLIT.NE.1) THEN
              WRITE(30,3242) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c        (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)), GTORSURF,
     c        (-1*SPACERTOPSURF), SPACERBOTTOMSURF, GTUNIV(COLUMN,ROW),
     c        SPN
 3242         FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,
     c        '    $ Homogenized region for spacer ',I2)
              LN=LN+1
*     Write the water region cell below the current homogenized spacer cell
in this GT universe.
              WRITE(30,3244) LN, BMODML, (-1*MODDENSITY), GTORSURF,
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        GTUNIV(COLUMN,ROW)
 3244         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
           ELSEIF (GTSPLIT.EQ.1) THEN
            DO 3270 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
              IF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(SPACERTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
              WRITE(30,3246) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c        (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c        GTSECTORSURF(SECT),
     c        (-1*SPACERTOPSURF), SPACERBOTTOMSURF, GTUNIV(COLUMN,ROW),
     c        SPN
 3246         FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,
     c        '    $ Homogenized region for spacer ',I2)
              LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c        SURFVALUESPEC(SPACERTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
              WRITE(30,3248) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c        (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c        GTSECTORSURF(SECT),
     c        (-1*SPACERTOPSURF), SPACERBOTTOMSURF, GTUNIV(COLUMN,ROW),
     c        SPN
```

```
3248        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,
     c      '   $ Homogenized region for spacer ',I2)
            LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c      SURFVALUESPEC(SPACERTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c      SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
            WRITE(30,3250) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c      (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c      GTSECTORSURF(SECT),
     c      (-1*SPACERTOPSURF), SPACERBOTTOMSURF, GTUNIV(COLUMN,ROW),
     c      SPN
3250        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,
     c      '   $ Homogenized region for spacer ',I2)
            LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c      SURFVALUESPEC(SPACERTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c      SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
            WRITE(30,3252) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c      (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c      GTSECTORSURF(SECT),
     c      (-1*SPACERTOPSURF), SPACERBOTTOMSURF, GTUNIV(COLUMN,ROW),
     c      SPN
3252        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,
     c      '   $ Homogenized region for spacer ',I2)
            LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c      SURFVALUESPEC(SPACERTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).GT.
     c      SURFVALUESPEC(SPACERBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c      SURFVALUESPEC(SPACERTOPSURF))) THEN
            WRITE(30,3254) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c      (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c      GTSECTORSURF(SECT),
     c      (-1*SPACERTOPSURF), GTSECTBOTSURF(SECT),
     c      GTUNIV(COLUMN,ROW), SPN
3254        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,
     c      '   $ Homogenized region for spacer ',I2)
            LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).LT.
     c      SURFVALUESPEC(SPACERTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c      SURFVALUESPEC(SPACERBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c      SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
            WRITE(30,3256) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c      (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c      GTSECTORSURF(SECT),
```

```
     c        (-1*GTSECTTOPSURF(SECT)), SPACERBOTTOMSURF,
     c        GTUNIV(COLUMN,ROW), SPN
 3256         FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,
     c        '   $ Homogenized region for spacer ',I2)
              LN=LN+1
               ENDIF
*      Write the water region cell below the current homogenized spacer cell
in this GT universe.
               IF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,3258) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        GTUNIV(COLUMN,ROW)
 3258         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'   $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,3260) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        GTUNIV(COLUMN,ROW)
 3260         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'   $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,3262) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        GTUNIV(COLUMN,ROW)
 3262         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'   $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,3264) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        GTUNIV(COLUMN,ROW)
 3264         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'   $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
```

```
     c         SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c         (SURFVALUESPEC(GTSECTBOTSURF(SECT)).GT.
     c         SURFVALUESPEC(WATERREGIONBOTTOMSURF)).AND.
     c         (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c         SURFVALUESPEC(WATERREGIONTOPSURF))) THEN
               WRITE(30,3266) LN, BMODML, (-1*MODDENSITY),
     c         GTSECTORSURF(SECT),
     c         (-1*WATERREGIONTOPSURF), GTSECTBOTSURF(SECT),
     c         GTUNIV(COLUMN,ROW)
 3266          FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I4,'    $ Borated moderator region')
               LN=LN+1
                ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).LT.
     c         SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c         (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c         SURFVALUESPEC(WATERREGIONBOTTOMSURF)).AND.
     c         (SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c         SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
               WRITE(30,3268) LN, BMODML, (-1*MODDENSITY),
     c         GTSECTORSURF(SECT),
     c         (-1*GTSECTTOPSURF(SECT)), WATERREGIONBOTTOMSURF,
     c         GTUNIV(COLUMN,ROW)
 3268          FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I4,'    $ Borated moderator region')
               LN=LN+1
                ENDIF
 3270          CONTINUE
            ENDIF
 3320       CONTINUE
         ENDIF
 3330       CONTINUE
 3340    CONTINUE
*  Write the specifications for the IT universes that are
*  required to fill the assembly layout specifications previously defined.
         DO 3858 ROW=1,50
            DO 3856 COLUMN=1,50
*        Write the IT universe specification if the assembly design is unique.
*
         IF (DUNIQUE(COLUMN,ROW).EQ..TRUE.) THEN
*  Write the IT specification header.
            WRITE(30,3350)
 3350       FORMAT(T1,'C')
            WRITE(30,3352) ASSYID(COLUMN,ROW)
 3352       FORMAT(T1,
    c       'C   INSTRUMENT TUBE UNIVERSE SPECIFICATION FOR ASSEMBLY ',A5)
            WRITE(30,3354)
 3354       FORMAT(T1,'C')
*        Define the upper end-fitting bottom surface.
                  CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
    c             ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
                  CURRENTSURFLABEL=0
                  DO 3356 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
```

```
                    CURRENTSURFLABEL=V
                    EXIT
                 ENDIF
              ENDIF
 3356      CONTINUE
           IF (CURRENTSURFLABEL.EQ.0) THEN
              UEFBOTTOMSURF=SN
              SURFTYPESPEC(SN)='PZ'
              SURFVALUESPEC(SN)=CURRENTSURF
              SN=SN+1
           ELSE
              UEFBOTTOMSURF=CURRENTSURFLABEL
           ENDIF
*     Define the upper end-fitting top surface.
           CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c     ENDFITHEIGHT(DESNUM(COLUMN,ROW),1)+
     c     ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
           CURRENTSURFLABEL=0
           DO 3358 V=1,(SN-1)
              IF (SURFTYPESPEC(V).EQ.'PZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                 ENDIF
              ENDIF
 3358      CONTINUE
           IF (CURRENTSURFLABEL.EQ.0) THEN
              UEFTOPSURF=SN
              SURFTYPESPEC(SN)='PZ'
              SURFVALUESPEC(SN)=CURRENTSURF
              SN=SN+1
           ELSE
              UEFTOPSURF=CURRENTSURFLABEL
           ENDIF
        IF (ITSPLIT.NE.1) THEN
*     Define the IT top surface.
           ITTOPSURF=UEFBOTTOMSURF
*     Define the IT bottom surface.
           CURRENTSURF=ITDATA(DESNUM(COLUMN,ROW),4)
           CURRENTSURFLABEL=0
           DO 3360 V=1,(SN-1)
              IF (SURFTYPESPEC(V).EQ.'PZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                 ENDIF
              ENDIF
 3360      CONTINUE
           IF (CURRENTSURFLABEL.EQ.0) THEN
              ITBOTSURF=SN
              SURFTYPESPEC(SN)='PZ'
              SURFVALUESPEC(SN)=CURRENTSURF
              SN=SN+1
           ELSE
```

```
                  ITBOTSURF=CURRENTSURFLABEL
               ENDIF
*     Define the IT outer radius surface.
               CURRENTSURF=ITDATA(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 3362 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 3362          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  ITORSURF=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  ITORSURF=CURRENTSURFLABEL
               ENDIF
*     Define the IT inner radius surface.
               CURRENTSURF=ITDATA(DESNUM(COLUMN,ROW),1)
               CURRENTSURFLABEL=0
               DO 3364 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 3364          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  ITIRSURF=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  ITIRSURF=CURRENTSURFLABEL
               ENDIF
            ELSEIF (ITSPLIT.EQ.1) THEN
             DO 3374 SECT=1,NUMOFITAXS(DESNUM(COLUMN,ROW))
*     Define the IT top surface.
               CURRENTSURF=ITAXDATA(DESNUM(COLUMN,ROW),3,SECT)
               CURRENTSURFLABEL=0
               DO 3366 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 3366          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
```

```
                  ITSECTTOPSURF(SECT)=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  ITSECTTOPSURF(SECT)=CURRENTSURFLABEL
               ENDIF
*     Define the IT bottom surface.
               CURRENTSURF=ITAXDATA(DESNUM(COLUMN,ROW),4,SECT)
               CURRENTSURFLABEL=0
               DO 3368 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 3368          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  ITSECTBOTSURF(SECT)=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  ITSECTBOTSURF(SECT)=CURRENTSURFLABEL
               ENDIF
*     Define the IT outer radius surface.
               CURRENTSURF=ITAXDATA(DESNUM(COLUMN,ROW),2,SECT)
               CURRENTSURFLABEL=0
               DO 3370 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 3370          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  ITSECTORSURF(SECT)=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  ITSECTORSURF(SECT)=CURRENTSURFLABEL
               ENDIF
*     Define the IT inner radius surface.
               CURRENTSURF=ITAXDATA(DESNUM(COLUMN,ROW),1,SECT)
               CURRENTSURFLABEL=0
               DO 3372 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
```

```
                    ENDIF
 3372            CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
                    ITSECTIRSURF(SECT)=SN
                    SURFTYPESPEC(SN)='CZ'
                    SURFVALUESPEC(SN)=CURRENTSURF
                    SN=SN+1
                 ELSE
                    ITSECTIRSURF(SECT)=CURRENTSURFLABEL
                 ENDIF
 3374         CONTINUE
             ENDIF
*     Define the lower end-fitting top surface.
                 CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
                 CURRENTSURFLABEL=0
                 DO 3376 V=1,(SN-1)
                    IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                          CURRENTSURFLABEL=V
                          EXIT
                       ENDIF
                    ENDIF
 3376            CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
                    LEFTOPSURF=SN
                    SURFTYPESPEC(SN)='PZ'
                    SURFVALUESPEC(SN)=CURRENTSURF
                    SN=SN+1
                 ELSE
                    LEFTOPSURF=CURRENTSURFLABEL
                 ENDIF
*     Write the lower end-fitting cell specification for this empty IT
universe.
*     Determine if the lower end-fitting material specification has
*     previously been defined.  If it has been previously defined, determine
*     the lower end-fitting material specification label.
            FRLEFMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 3380 RO=1,(ROW-1)
                  DO 3378 CO=1,50
                     IF (DESNUM(CO,RO).NE.0) THEN
                        IF (DESNUM(COLUMN,ROW).EQ.DESNUM(CO,RO)) THEN
                           FRLEFMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           ITLEFML(COLUMN,ROW)=ITLEFML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 3378             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 3380          CONTINUE
```

```
            IF (LEAVE.EQ..FALSE.) THEN
               DO 3384 RO=ROW,ROW
                  DO 3382 CO=1,(COLUMN-1)
                     IF (DESNUM(CO,RO).NE.0) THEN
                        IF (DESNUM(COLUMN,ROW).EQ.
     c                  DESNUM(CO,RO)) THEN
                           FRLEFMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           ITLEFML(COLUMN,ROW)=ITLEFML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 3382             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 3384          CONTINUE
            ENDIF
         ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
            DO 3388 RO=1,(ROW-1)
               DO 3386 CO=1,50
                  IF (DESNUM(CO,RO).NE.0) THEN
                     IF (DESNUM(COLUMN,ROW).EQ.
     c               DESNUM(CO,RO)) THEN
                        FRLEFMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        ITLEFML(COLUMN,ROW)=ITLEFML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
 3386          CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
 3388       CONTINUE
         ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
            DO 3392 RO=1,1
               DO 3390 CO=1,(COLUMN-1)
                  IF (DESNUM(CO,RO).NE.0) THEN
                     IF (DESNUM(COLUMN,ROW).EQ.
     c               DESNUM(CO,RO)) THEN
                        FRLEFMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        ITLEFML(COLUMN,ROW)=ITLEFML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
 3390          CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
 3392       CONTINUE
         ENDIF
         IF (FRLEFMLUNIQUE.EQ..TRUE.) THEN
```

```
                ITLEFML(COLUMN,ROW)=MN
*  Check Instrument Tube Lower End-Fitting Regions
                  DO 3398 C=1,LEFMAT(DESNUM(COLUMN,ROW),2)
                    IF (C.EQ.1) THEN
                      WRITE(200,3394) ITLEFML(COLUMN,ROW),
     c                LEFZAIDS(DESNUM(COLUMN,ROW),C),
     c                (-1*LEFWTS(DESNUM(COLUMN,ROW),C))
 3394                 FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                '   $ Instrument Tube Lower End Fitting')
                    ELSE
                      WRITE(200,3396) LEFZAIDS(DESNUM(COLUMN,ROW),C),
     c                (-1*LEFWTS(DESNUM(COLUMN,ROW),C))
 3396                 FORMAT(T9,A9,3X,G14.6)
                    ENDIF
 3398             CONTINUE
                WRITE(200,3400) ITLEFML(COLUMN,ROW)
 3400           FORMAT(T1,'MT',I4,T9,'LWTR.03T')
                MN=MN+1
              ENDIF
            IF (ITSPLIT.NE.1) THEN
              IF (SURFVALUESPEC(ITBOTSURF).GE.
     c        ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)) THEN
                WRITE(30,3402) LN, ITLEFML(COLUMN,ROW),
     c          (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*LEFTOPSURF),
     c          ITUNIV(COLUMN,ROW)
 3402           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,' IMP:N=1 U=',I4,
     c          '   $ Lower end-fitting region')
                LN=LN+1
              ELSE
                WRITE(30,3404) LN, ITLEFML(COLUMN,ROW),
     c          (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*LEFTOPSURF),
     c          ITORSURF, ITUNIV(COLUMN,ROW)
 3404           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c          ' IMP:N=1 U=',I4,'   $ Lower end-fitting region')
                LN=LN+1
                WRITE(30,3406) LN, ITLEFML(COLUMN,ROW),
     c          (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*ITBOTSURF),
     c          (-1*ITORSURF), ITUNIV(COLUMN,ROW)
 3406           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c          ' IMP:N=1 U=',I4,'   $ Lower end-fitting region')
                LN=LN+1
              ENDIF
            ELSEIF (ITSPLIT.EQ.1) THEN
            LITS=ITSECTBOTSURF(1)
            LITSOR=ITSECTBOTSURF(1)
            DO 3408 SECT=2,NUMOFITAXS(DESNUM(COLUMN,ROW))
              IF (SURFVALUESPEC(ITSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(LITS)) THEN
                LITS=ITSECTBOTSURF(SECT)
                LITSOR=ITSECTORSURF(SECT)
              ENDIF
 3408       CONTINUE
              IF (SURFVALUESPEC(LITS).GE.
     c        ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)) THEN
```

```
                 WRITE(30,3410) LN, ITLEFML(COLUMN,ROW),
     c           (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*LEFTOPSURF),
     c           ITUNIV(COLUMN,ROW)
 3410            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,' IMP:N=1 U=',I4,
     c           '    $ Instrument tube lower end-fitting')
                 LN=LN+1
              ELSE
                 WRITE(30,3412) LN, ITLEFML(COLUMN,ROW),
     c           (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*LEFTOPSURF),
     c           LITSOR, ITUNIV(COLUMN,ROW)
 3412            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c           ' IMP:N=1 U=',I4,
     c           '    $ Instrument tube lower end-fitting')
                 LN=LN+1
                 WRITE(30,3414) LN, ITLEFML(COLUMN,ROW),
     c           (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*LITS),
     c           (-1*LITSOR), ITUNIV(COLUMN,ROW)
 3414            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c           ' IMP:N=1 U=',I4,
     c           '    $ Instrument tube lower end-fitting')
                 LN=LN+1
              ENDIF
           ENDIF
*     Write the upper end-fitting cell specification for this empty IT
universe.
*     Determine if the IT universe upper end-fitting material specification
has
*     previously been defined.  If it has been previously defined, determine
*     the upper end-fitting material specification label.
              FRUEFMLUNIQUE=.TRUE.
              LEAVE=.FALSE.
              IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                 DO 3418 RO=1,(ROW-1)
                    DO 3416 CO=1,50
                       IF (DESNUM(CO,RO).NE.0) THEN
                          IF (DESNUM(COLUMN,ROW).EQ.DESNUM(CO,RO)) THEN
                             FRUEFMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             ITUEFML(COLUMN,ROW)=ITUEFML(CO,RO)
                             EXIT
                          ENDIF
                       ENDIF
 3416               CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
 3418            CONTINUE
                 IF (LEAVE.EQ..FALSE.) THEN
                    DO 3422 RO=1,(ROW-1)
                       DO 3420 CO=1,50
                          IF (DESNUM(CO,RO).NE.0) THEN
                             IF (DESNUM(COLUMN,ROW).EQ.
     c                       DESNUM(CO,RO)) THEN
                                FRUEFMLUNIQUE=.FALSE.
```

```
                        LEAVE=.TRUE.
                        ITUEFML(COLUMN,ROW)=ITUEFML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
 3420          CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
 3422       CONTINUE
         ENDIF
      ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
         DO 3426 RO=1,(ROW-1)
            DO 3424 CO=1,50
               IF (DESNUM(CO,RO).NE.0) THEN
                  IF (DESNUM(COLUMN,ROW).EQ.
     c               DESNUM(CO,RO)) THEN
                     FRUEFMLUNIQUE=.FALSE.
                     LEAVE=.TRUE.
                     ITUEFML(COLUMN,ROW)=ITUEFML(CO,RO)
                     EXIT
                  ENDIF
               ENDIF
 3424       CONTINUE
            IF (LEAVE.EQ..TRUE.) THEN
               EXIT
            ENDIF
 3426    CONTINUE
      ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
         DO 3430 RO=1,1
            DO 3428 CO=1,(COLUMN-1)
               IF (DESNUM(CO,RO).NE.0) THEN
                  IF (DESNUM(COLUMN,ROW).EQ.
     c               DESNUM(CO,RO)) THEN
                     FRUEFMLUNIQUE=.FALSE.
                     LEAVE=.TRUE.
                     ITUEFML(COLUMN,ROW)=ITUEFML(CO,RO)
                     EXIT
                  ENDIF
               ENDIF
 3428       CONTINUE
            IF (LEAVE.EQ..TRUE.) THEN
               EXIT
            ENDIF
 3430    CONTINUE
      ENDIF
      IF (FRUEFMLUNIQUE.EQ..TRUE.) THEN
         ITUEFML(COLUMN,ROW)=MN
*  Check Instrument Tube Upper End-Fitting Regions
            DO 3436 C=1,UEFMAT(DESNUM(COLUMN,ROW),2)
               IF (C.EQ.1) THEN
                  WRITE(200,3432) ITUEFML(COLUMN,ROW),
     c            UEFZAIDS(DESNUM(COLUMN,ROW),C),
     c            (-1*UEFWTS(DESNUM(COLUMN,ROW),C))
```

```
 3432                    FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                   '    $ Instrument Tube Upper End Fitting')
                      ELSE
                         WRITE(200,3434) UEFZAIDS(DESNUM(COLUMN,ROW),C),
     c                   (-1*UEFWTS(DESNUM(COLUMN,ROW),C))
 3434                    FORMAT(T9,A9,3X,G14.6)
                      ENDIF
 3436              CONTINUE
                WRITE(200,3438) ITUEFML(COLUMN,ROW)
 3438           FORMAT(T1,'MT',I4,T9,'LWTR.03T')
                MN=MN+1
             ENDIF
          IF (ITSPLIT.NE.1) THEN
             IF ((SURFVALUESPEC(ITTOPSURF).LE.
     c       SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                WRITE(30,3440) LN, ITUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), ITUNIV(COLUMN,ROW)
 3440           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c          ' IMP:N=1 U=',I4,'    $ Upper end-fitting region')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(ITTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(ITTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,3442) LN, ITUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), ITORSURF, ITUNIV(COLUMN,ROW)
 3442           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I4,'    $ Upper end-fitting region')
                LN=LN+1
                WRITE(30,3444) LN, ITUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), ITTOPSURF,
     c          (-1*UEFTOPSURF), (-1*ITORSURF),
     c          ITUNIV(COLUMN,ROW)
 3444           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I4,'    $ Upper end-fitting region')
                LN=LN+1
             ELSEIF (SURFVALUESPEC(ITTOPSURF).EQ.
     c       SURFVALUESPEC(UEFTOPSURF)) THEN
                WRITE(30,3446) LN, ITUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), ITORSURF, ITUNIV(COLUMN,ROW)
 3446           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I4,'    $ Upper end-fitting region')
                LN=LN+1
             ENDIF
          ELSEIF (ITSPLIT.EQ.1) THEN
          UITS=ITSECTTOPSURF(1)
          UITSOR=ITSECTORSURF(1)
          DO 3448 SECT=2,NUMOFITAXS(DESNUM(COLUMN,ROW))
             IF (SURFVALUESPEC(ITSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(UITS)) THEN
                UITS=ITSECTTOPSURF(SECT)
```

```
                  UITSOR=ITSECTORSURF(SECT)
               ENDIF
 3448       CONTINUE
               IF ((SURFVALUESPEC(UITS).LE.
     c         SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                  WRITE(30,3450) LN, ITUEFML(COLUMN,ROW),
     c            (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c            (-1*UEFTOPSURF), ITUNIV(COLUMN,ROW)
 3450             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c            ' IMP:N=1 U=',I4,
     c            '    $ Instrument tube upper end-fitting')
                  LN=LN+1
               ELSEIF ((SURFVALUESPEC(UITS).GT.
     c         SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c         (SURFVALUESPEC(UITS).LT.
     c         SURFVALUESPEC(UEFTOPSURF))) THEN
                  WRITE(30,3452) LN, ITUEFML(COLUMN,ROW),
     c            (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c            (-1*UEFTOPSURF), UITSOR, ITUNIV(COLUMN,ROW)
 3452             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c            ' IMP:N=1 U=',I4,
     c            '    $ Instrument tube upper end-fitting')
                  LN=LN+1
                  WRITE(30,3454) LN, ITUEFML(COLUMN,ROW),
     c            (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UITS,
     c            (-1*UEFTOPSURF), (-1*UITSOR),
     c            ITUNIV(COLUMN,ROW)
 3454             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c            ' IMP:N=1 U=',I4,
     c            '    $ Instrument tube upper end-fitting')
                  LN=LN+1
               ELSEIF (SURFVALUESPEC(ITTOPSURF).EQ.
     c         SURFVALUESPEC(UEFTOPSURF)) THEN
                  WRITE(30,3456) LN, ITUEFML(COLUMN,ROW),
     c            (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c            (-1*UEFTOPSURF), UITSOR, ITUNIV(COLUMN,ROW)
 3456             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c            ' IMP:N=1 U=',I4,
     c            '    $ Instrument tube upper end-fitting')
                  LN=LN+1
               ENDIF
            ENDIF
*     Write the IT material cell in this IT universe.
            IF (ITSPLIT.NE.1) THEN
*     Determine if the IT material specification has
*     previously been defined.  If it has been previously defined, determine
*     the material specification label.
               CLADMLUNIQUE=.TRUE.
               LEAVE=.FALSE.
               IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                  DO 3680 RO=1,(ROW-1)
                     DO 3670 CO=1,50
                        IF (DESNUM(CO,RO).NE.0) THEN
                           IF (ITMAT(DESNUM(COLUMN,ROW)).EQ.
```

```
     c                       ITMAT(DESNUM(CO,RO))) THEN
                                CLADMLUNIQUE=.FALSE.
                                LEAVE=.TRUE.
                                ITML(COLUMN,ROW)=ITML(CO,RO)
                                EXIT
                             ENDIF
                          ENDIF
 3670                  CONTINUE
                       IF (LEAVE.EQ..TRUE.) THEN
                          EXIT
                       ENDIF
 3680               CONTINUE
                    IF (LEAVE.EQ..FALSE.) THEN
                       DO 3700 RO=ROW,ROW
                          DO 3690 CO=1,(COLUMN-1)
                             IF (DESNUM(CO,RO).NE.0) THEN
                                IF (ITMAT(DESNUM(COLUMN,ROW)).EQ.
     c                          ITMAT(DESNUM(CO,RO))) THEN
                                   CLADMLUNIQUE=.FALSE.
                                   LEAVE=.TRUE.
                                   ITML(COLUMN,ROW)=ITML(CO,RO)
                                   EXIT
                                ENDIF
                             ENDIF
 3690                     CONTINUE
                          IF (LEAVE.EQ..TRUE.) THEN
                             EXIT
                          ENDIF
 3700                  CONTINUE
                    ENDIF
                 ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                    DO 3720 RO=1,(ROW-1)
                       DO 3710 CO=1,50
                          IF (DESNUM(CO,RO).NE.0) THEN
                             IF (ITMAT(DESNUM(COLUMN,ROW)).EQ.
     c                       ITMAT(DESNUM(CO,RO))) THEN
                                CLADMLUNIQUE=.FALSE.
                                LEAVE=.TRUE.
                                ITML(COLUMN,ROW)=ITML(CO,RO)
                                EXIT
                             ENDIF
                          ENDIF
 3710                  CONTINUE
                       IF (LEAVE.EQ..TRUE.) THEN
                          EXIT
                       ENDIF
 3720               CONTINUE
                 ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                    DO 3740 RO=1,1
                       DO 3730 CO=1,(COLUMN-1)
                          IF (DESNUM(CO,RO).NE.0) THEN
                             IF (ITMAT(DESNUM(COLUMN,ROW)).EQ.
     c                       ITMAT(DESNUM(CO,RO))) THEN
                                CLADMLUNIQUE=.FALSE.
```

```
                              LEAVE=.TRUE.
                              ITML(COLUMN,ROW)=ITML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 3730                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 3740             CONTINUE
               ENDIF
               IF (CLADMLUNIQUE.EQ..TRUE.) THEN
                  ITML(COLUMN,ROW)=MN
*  Check Instrument Tube Material
                     IF (ITMAT(DESNUM(COLUMN,ROW)).EQ.1) THEN
                        DO 3742 C=1,2
                           IF (C.EQ.1) THEN
                              WRITE(200,9300) ITML(COLUMN,ROW)
                           ELSEIF (C.EQ.2) THEN
                              WRITE(200,9301)
                              WRITE(200,7000)
                              WRITE(200,7001)
                              WRITE(200,7002)
                              WRITE(200,9302)
                              WRITE(200,7003)
                              WRITE(200,7004)
                              WRITE(200,7005)
                              WRITE(200,9303)
                              WRITE(200,9304)
                           ENDIF
 3742                   CONTINUE
                     ELSEIF (ITMAT(DESNUM(COLUMN,ROW))
     c               .EQ.2) THEN
                        DO 3744 C=1,2
                           IF (C.EQ.1) THEN
                              WRITE(200,9305) ITML(COLUMN,ROW)
                           ELSEIF (C.EQ.2) THEN
                              WRITE(200,9306)
                              WRITE(200,9307)
                              WRITE(200,9308)
                              WRITE(200,9309)
                              WRITE(200,9310)
                              WRITE(200,7006)
                              WRITE(200,7007)
                              WRITE(200,7008)
                              WRITE(200,9311)
                              WRITE(200,9312)
                              WRITE(200,7009)
                              WRITE(200,7010)
                              WRITE(200,7011)
                              WRITE(200,9313)
                              WRITE(200,7012)
                              WRITE(200,7013)
                              WRITE(200,7014)
```

```
                        WRITE(200,7015)
                     ENDIF
 3744             CONTINUE
               ELSEIF (ITMAT(DESNUM(COLUMN,ROW))
     c         .EQ.3) THEN
                  DO 3746 C=1,2
                     IF (C.EQ.1) THEN
                        WRITE(200,9314) ITML(COLUMN,ROW)
                     ELSEIF (C.EQ.2) THEN
                        WRITE(200,9315)
                        WRITE(200,9316)
                        WRITE(200,9317)
                        WRITE(200,9318)
                        WRITE(200,7016)
                        WRITE(200,7017)
                        WRITE(200,7018)
                        WRITE(200,9319)
                        WRITE(200,9320)
                        WRITE(200,7019)
                        WRITE(200,7020)
                        WRITE(200,7021)
                        WRITE(200,9321)
                        WRITE(200,7022)
                        WRITE(200,7023)
                        WRITE(200,7024)
                        WRITE(200,7025)
                        WRITE(200,9322)
                        WRITE(200,9323)
                        WRITE(200,9324)
                        WRITE(200,9325)
                        WRITE(200,9326)
                        WRITE(200,9327)
                        WRITE(200,7026)
                        WRITE(200,9328)
                        WRITE(200,9329)
                        WRITE(200,9330)
                     ENDIF
 3746             CONTINUE
               ENDIF
            MN=MN+1
         ENDIF
         IF (ITMAT(DESNUM(COLUMN,ROW)).EQ.1) THEN
            CLADRHO=6.56
         ELSEIF (ITMAT(DESNUM(COLUMN,ROW)).EQ.2) THEN
            CLADRHO=7.90
         ELSEIF (ITMAT(DESNUM(COLUMN,ROW)).EQ.3) THEN
            CLADRHO=8.19
         ENDIF
         WRITE(30,3748) LN, ITML(COLUMN,ROW), (-1*CLADRHO),
     c   ITIRSURF,
     c   (-1*ITORSURF), (-1*ITTOPSURF), ITBOTSURF,
     c   ITUNIV(COLUMN,ROW)
 3748    FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c   ' IMP:N=1 U=',I4,'   $ Instrument tube')
```

```
            LN=LN+1
*     Write the moderator cells within the IT in this IT universe.
            WRITE(30,3750) LN, BMODML, (-1*MODDENSITY),
     c      (-1*ITIRSURF), (-1*ITTOPSURF), ITBOTSURF,
     c      ITUNIV(COLUMN,ROW)
 3750       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,'   $ Borated moderator')
            LN=LN+1
          ELSEIF (ITSPLIT.EQ.1) THEN
           DO 3778 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
*     Determine if the IT material specification has
*     previously been defined.  If it has been previously defined, determine
*     the material specification label.
            CLADMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 3754 RO=1,(ROW-1)
                  DO 3752 CO=1,50
                     IF ((DESNUM(CO,RO).NE.0).AND.
     c               (BANKNUM(CO,RO).EQ.0)) THEN
                        IF (ITAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                  ITMAT(DESNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           ITAXML(COLUMN,ROW,SECT)=ITML(CO,RO)
                           EXIT
                        ELSEIF (ITAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                  ITAXMAT(DESNUM(CO,RO),SECT)) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           ITAXML(COLUMN,ROW,SECT)=ITAXML(CO,RO,SECT)
                           EXIT
                        ENDIF
                     ENDIF
 3752             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 3754          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 3758 RO=ROW,ROW
                     DO 3756 CO=1,(COLUMN-1)
                        IF ((DESNUM(CO,RO).NE.0).AND.
     c                  (BANKNUM(CO,RO).EQ.0)) THEN
                           IF (ITAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                     ITMAT(DESNUM(CO,RO))) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              ITAXML(COLUMN,ROW,SECT)=ITML(CO,RO)
                              EXIT
                           ELSEIF (ITAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                     ITAXMAT(DESNUM(CO,RO),SECT)) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
```

```
                          ITAXML(COLUMN,ROW,SECT)=ITAXML(CO,RO,SECT)
                          EXIT
                       ENDIF
                    ENDIF
3756             CONTINUE
                 IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                 ENDIF
3758          CONTINUE
           ENDIF
        ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
           DO 3762 RO=1,(ROW-1)
              DO 3760 CO=1,50
                 IF ((DESNUM(CO,RO).NE.0).AND.
     c              (BANKNUM(CO,RO).EQ.0)) THEN
                    IF (ITAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c              ITMAT(DESNUM(CO,RO))) THEN
                       CLADMLUNIQUE=.FALSE.
                       LEAVE=.TRUE.
                       ITAXML(COLUMN,ROW,SECT)=ITML(CO,RO)
                       EXIT
                    ELSEIF (ITAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c              ITAXMAT(DESNUM(CO,RO),SECT)) THEN
                       CLADMLUNIQUE=.FALSE.
                       LEAVE=.TRUE.
                       ITAXML(COLUMN,ROW,SECT)=ITAXML(CO,RO,SECT)
                       EXIT
                    ENDIF
                 ENDIF
3760          CONTINUE
              IF (LEAVE.EQ..TRUE.) THEN
                 EXIT
              ENDIF
3762       CONTINUE
        ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
           DO 3766 RO=1,1
              DO 3764 CO=1,(COLUMN-1)
                 IF ((DESNUM(CO,RO).NE.0).AND.
     c              (BANKNUM(CO,RO).EQ.0)) THEN
                    IF (ITAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c              ITMAT(DESNUM(CO,RO))) THEN
                       CLADMLUNIQUE=.FALSE.
                       LEAVE=.TRUE.
                       ITAXML(COLUMN,ROW,SECT)=ITML(CO,RO)
                       EXIT
                    ELSEIF (ITAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c              ITAXMAT(DESNUM(CO,RO),SECT)) THEN
                       CLADMLUNIQUE=.FALSE.
                       LEAVE=.TRUE.
                       ITAXML(COLUMN,ROW,SECT)=ITAXML(CO,RO,SECT)
                       EXIT
                    ENDIF
                 ENDIF
3764          CONTINUE
```

```
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 3766          CONTINUE
            ENDIF
            IF (CLADMLUNIQUE.EQ..TRUE.) THEN
               ITAXML(COLUMN,ROW,SECT)=MN
*  Check Guide Tube Material
                  IF (ITAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.1) THEN
                     DO 3768 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9300) ITAXML(COLUMN,ROW,SECT)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9301)
                           WRITE(200,7000)
                           WRITE(200,7001)
                           WRITE(200,7002)
                           WRITE(200,9302)
                           WRITE(200,7003)
                           WRITE(200,7004)
                           WRITE(200,7005)
                           WRITE(200,9303)
                           WRITE(200,9304)
                        ENDIF
 3768                CONTINUE
                  ELSEIF (ITAXMAT(DESNUM(COLUMN,ROW),SECT)
     c            ..EQ.2) THEN
                     DO 3770 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9305) ITAXML(COLUMN,ROW,SECT)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9306)
                           WRITE(200,9307)
                           WRITE(200,9308)
                           WRITE(200,9309)
                           WRITE(200,9310)
                           WRITE(200,7006)
                           WRITE(200,7007)
                           WRITE(200,7008)
                           WRITE(200,9311)
                           WRITE(200,9312)
                           WRITE(200,7009)
                           WRITE(200,7010)
                           WRITE(200,7011)
                           WRITE(200,9313)
                           WRITE(200,7012)
                           WRITE(200,7013)
                           WRITE(200,7014)
                           WRITE(200,7015)
                        ENDIF
 3770                CONTINUE
                  ELSEIF (ITAXMAT(DESNUM(COLUMN,ROW),SECT)
     c            .EQ.3) THEN
                     DO 3772 C=1,2
```

```
                              IF (C.EQ.1) THEN
                                 WRITE(200,9314) ITAXML(COLUMN,ROW,SECT)
                              ELSEIF (C.EQ.2) THEN
                                 WRITE(200,9315)
                                 WRITE(200,9316)
                                 WRITE(200,9317)
                                 WRITE(200,9318)
                                 WRITE(200,7016)
                                 WRITE(200,7017)
                                 WRITE(200,7018)
                                 WRITE(200,9319)
                                 WRITE(200,9320)
                                 WRITE(200,7019)
                                 WRITE(200,7020)
                                 WRITE(200,7021)
                                 WRITE(200,9321)
                                 WRITE(200,7022)
                                 WRITE(200,7023)
                                 WRITE(200,7024)
                                 WRITE(200,7025)
                                 WRITE(200,9322)
                                 WRITE(200,9323)
                                 WRITE(200,9324)
                                 WRITE(200,9325)
                                 WRITE(200,9326)
                                 WRITE(200,9327)
                                 WRITE(200,7026)
                                 WRITE(200,9328)
                                 WRITE(200,9329)
                                 WRITE(200,9330)
                              ENDIF
 3772                      CONTINUE
                        ENDIF
                     MN=MN+1
                  ENDIF
                  IF (ITAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.1) THEN
                     CLADRHO=6.56
                  ELSEIF (ITAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.2) THEN
                     CLADRHO=7.90
                  ELSEIF (ITAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.3) THEN
                     CLADRHO=8.19
                  ENDIF
                  WRITE(30,3774) LN, ITAXML(COLUMN,ROW,SECT), (-1*CLADRHO),
     c            ITSECTIRSURF(SECT),
     c            (-1*ITSECTORSURF(SECT)), (-1*ITSECTTOPSURF(SECT)),
     c            ITSECTBOTSURF(SECT), ITUNIV(COLUMN,ROW)
 3774             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c            ' IMP:N=1 U=',I4,'   $ Instrument tube')
                  LN=LN+1
*     Write the moderator cells within the IT in this IT universe.
                  WRITE(30,3776) LN, BMODML, (-1*MODDENSITY),
     c            (-1*ITSECTIRSURF(SECT)), (-1*ITSECTTOPSURF(SECT)),
     c            ITSECTBOTSURF(SECT), ITUNIV(COLUMN,ROW)
 3776             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
```

```
     c        ' IMP:N=1 U=',I4,
     c        '   $ Borated moderator inside instrument tube')
              LN=LN+1
 3778       CONTINUE
          ENDIF
*     Loop through the regions above the IT (i.e. the appropriate upper core
regions)
*     Define the upper region lower surface.
              IF (BANKNUM(COLUMN,ROW).NE.0) THEN
              IF (BANKDES(BANKNUM(COLUMN,ROW)).EQ.'BPRA ') THEN
                DO 3786 REGION=1,NUMREGABOVEBPRA
*     Determine the current upper region's lower surface specification.
                    IF (REGION.EQ.1) THEN
                      REGIONTOPSURF=SYSTEMTOP
                      CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
     c                REGABOVEBPRA(REGION,1)
                    ENDIF
                    CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
     c              REGABOVEBPRA(REGION,1)
                    IF (REGION.EQ.NUMREGABOVEBPRA) THEN
                      REGIONBOTTOMSURF=UEFTOPSURF
                    ELSE
                    CURRENTSURFLABEL=0
                    DO 3780 V=1,(SN-1)
                      IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                 IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                          CURRENTSURFLABEL=V
                          EXIT
                        ENDIF
                      ENDIF
 3780               CONTINUE
                    IF (CURRENTSURFLABEL.EQ.0) THEN
                      REGIONBOTTOMSURF=SN
                      SURFTYPESPEC(SN)='PZ'
                      SURFVALUESPEC(SN)=CURRENTSURF
                      SN=SN+1
                    ELSE
                      REGIONBOTTOMSURF=CURRENTSURFLABEL
                    ENDIF
                    ENDIF
*     Write the cell specification for the IT universe upper region.
                 IF (REGION.EQ.1) THEN
                    WRITE(30,3782) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c              (-1*REGABOVEBPRA(REGION,2)),
     c              REGIONBOTTOMSURF, ITUNIV(COLUMN,ROW), REGION
 3782               FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
     c              ' IMP:N=1 U=',I4,'    $ Upper core region ',I2)
                    LN=LN+1
                    REGIONTOPSURF=REGIONBOTTOMSURF
                 ELSE
                    WRITE(30,3784) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c              (-1*REGABOVEBPRA(REGION,2)), (-1*REGIONTOPSURF),
     c              REGIONBOTTOMSURF, ITUNIV(COLUMN,ROW), REGION
 3784               FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
```

```
     c                     ' IMP:N=1 U=',I4,'   $ Upper core region ',I2)
                  LN=LN+1
                  REGIONTOPSURF=REGIONBOTTOMSURF
               ENDIF
 3786          CONTINUE
            ELSEIF (BANKDES(BANKNUM(COLUMN,ROW)).EQ.'CRA  ') THEN
               DO 3794 REGION=1,NUMREGABOVECRA
*      Determine the current upper region's lower surface specification.
                  IF (REGION.EQ.1) THEN
                     REGIONTOPSURF=SYSTEMTOP
                     CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
     c               REGABOVECRA(REGION,1)
                  ENDIF
                  CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
     c            REGABOVECRA(REGION,1)
                  IF (REGION.EQ.NUMREGABOVECRA) THEN
                     REGIONBOTTOMSURF=UEFTOPSURF
                  ELSE
                  CURRENTSURFLABEL=0
                  DO 3788 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                           CURRENTSURFLABEL=V
                           EXIT
                        ENDIF
                     ENDIF
 3788             CONTINUE
                  IF (CURRENTSURFLABEL.EQ.0) THEN
                     REGIONBOTTOMSURF=SN
                     SURFTYPESPEC(SN)='PZ'
                     SURFVALUESPEC(SN)=CURRENTSURF
                     SN=SN+1
                  ELSE
                     REGIONBOTTOMSURF=CURRENTSURFLABEL
                  ENDIF
                  ENDIF
*      Write the cell specification for the IT universe upper region.
               IF (REGION.EQ.1) THEN
                  WRITE(30,3790) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c            (-1*REGABOVECRA(REGION,2)),
     c            REGIONBOTTOMSURF, ITUNIV(COLUMN,ROW), REGION
 3790             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
     c            ' IMP:N=1 U=',I4,'   $ Upper core region ',I2)
                  LN=LN+1
                  REGIONTOPSURF=REGIONBOTTOMSURF
               ELSE
                  WRITE(30,3792) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c            (-1*REGABOVECRA(REGION,2)), (-1*REGIONTOPSURF),
     c            REGIONBOTTOMSURF, ITUNIV(COLUMN,ROW), REGION
 3792             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c            ' IMP:N=1 U=',I4,'   $ Upper core region ',I2)
                  LN=LN+1
                  REGIONTOPSURF=REGIONBOTTOMSURF
               ENDIF
```

```
 3794         CONTINUE
           ELSEIF (BANKDES(BANKNUM(COLUMN,ROW)).EQ.'APSRA') THEN
              DO 3802 REGION=1,NUMREGABOVEAPSRA
*     Determine the current upper region's lower surface specification.
                 IF (REGION.EQ.1) THEN
                    REGIONTOPSURF=SYSTEMTOP
                    CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
     c              REGABOVEAPSRA(REGION,1)
                 ENDIF
                 CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
     c           REGABOVEAPSRA(REGION,1)
                 IF (REGION.EQ.NUMREGABOVEAPSRA) THEN
                    REGIONBOTTOMSURF=UEFTOPSURF
                 ELSE
                 CURRENTSURFLABEL=0
                 DO 3796 V=1,(SN-1)
                    IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                          CURRENTSURFLABEL=V
                          EXIT
                       ENDIF
                    ENDIF
 3796            CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
                    REGIONBOTTOMSURF=SN
                    SURFTYPESPEC(SN)='PZ'
                    SURFVALUESPEC(SN)=CURRENTSURF
                    SN=SN+1
                 ELSE
                    REGIONBOTTOMSURF=CURRENTSURFLABEL
                 ENDIF
                 ENDIF
*     Write the cell specification for the IT universe upper region.
              IF (REGION.EQ.1) THEN
                 WRITE(30,3798) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c           (-1*REGABOVEAPSRA(REGION,2)),
     c           REGIONBOTTOMSURF, ITUNIV(COLUMN,ROW), REGION
 3798            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
     c           ' IMP:N=1 U=',I4,'    $ Upper core region ',I2)
                 LN=LN+1
                 REGIONTOPSURF=REGIONBOTTOMSURF
              ELSE
                 WRITE(30,3800) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c           (-1*REGABOVEAPSRA(REGION,2)), (-1*REGIONTOPSURF),
     c           REGIONBOTTOMSURF, ITUNIV(COLUMN,ROW), REGION
 3800            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c           ' IMP:N=1 U=',I4,'    $ Upper core region ',I2)
                 LN=LN+1
                 REGIONTOPSURF=REGIONBOTTOMSURF
              ENDIF
 3802         CONTINUE
           ENDIF
           ENDIF
           IF ((BANKNUM(COLUMN,ROW).EQ.0).AND.
```

```
     c       (ASSYID(COLUMN,ROW).NE.'      ')) THEN
                DO 3810 REGION=1,NUMREGABOVE
*     Determine the current upper region's lower surface specification.
                   IF (REGION.EQ.1) THEN
                      REGIONTOPSURF=SYSTEMTOP
                      CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
     c                REGABOVE(REGION,1)
                   ENDIF
                   CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
     c             REGABOVE(REGION,1)
                   IF (REGION.EQ.NUMREGABOVE) THEN
                      REGIONBOTTOMSURF=UEFTOPSURF
                   ELSE
                   CURRENTSURFLABEL=0
                   DO 3804 V=1,(SN-1)
                      IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                 IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                            CURRENTSURFLABEL=V
                            EXIT
                         ENDIF
                      ENDIF
 3804              CONTINUE
                   IF (CURRENTSURFLABEL.EQ.0) THEN
                      REGIONBOTTOMSURF=SN
                      SURFTYPESPEC(SN)='PZ'
                      SURFVALUESPEC(SN)=CURRENTSURF
                      SN=SN+1
                   ELSE
                      REGIONBOTTOMSURF=CURRENTSURFLABEL
                   ENDIF
                   ENDIF
*     Write the cell specification for the IT universe upper region.
                IF (REGION.EQ.1) THEN
                   WRITE(30,3806) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c             (-1*REGABOVE(REGION,2)),
     c             REGIONBOTTOMSURF, ITUNIV(COLUMN,ROW), REGION
 3806              FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
     c             ' IMP:N=1 U=',I4,'   $ Upper core region ',I2)
                   LN=LN+1
                   REGIONTOPSURF=REGIONBOTTOMSURF
                ELSE
                   WRITE(30,3808) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c             (-1*REGABOVE(REGION,2)), (-1*REGIONTOPSURF),
     c             REGIONBOTTOMSURF, ITUNIV(COLUMN,ROW), REGION
 3808              FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c             ' IMP:N=1 U=',I4,'   $ Upper core region ',I2)
                   LN=LN+1
                   REGIONTOPSURF=REGIONBOTTOMSURF
                ENDIF
 3810           CONTINUE
             ENDIF
             SPACHEIGHT=0.0
*     Loop through the spacer and moderator regions along the axial
*     length of the IT (from top to bottom).
```

```
          DO 3812 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
            SPACHEIGHT=SPACHEIGHT+SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
 3812     CONTINUE
          DO 3854 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
*      Define the homogenized spacer region bounding surfaces.
              IF (SPN.EQ.1) THEN
                SPACERTOPSURF=UEFBOTTOMSURF
                CURRENTSURF=SURFVALUESPEC(UEFBOTTOMSURF)-
     c          SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                CURRENTSURFLABEL=0
                DO 3814 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                    ENDIF
                  ENDIF
 3814           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                  SPACERBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
                ELSE
                  SPACERBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
                WATERREGIONTOPSURF=SPACERBOTTOMSURF
                CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
                CURRENTSURFLABEL=0
                DO 3816 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                    ENDIF
                  ENDIF
 3816           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                  WATERREGIONBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
                ELSE
                  WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
              ELSEIF ((SPN.NE.1).AND.(SPN.NE.
     c        NUMOFSPACERS(DESNUM(COLUMN,ROW)))) THEN
                SPACERTOPSURF=WATERREGIONBOTTOMSURF
                CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c          SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                CURRENTSURFLABEL=0
                DO 3818 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
```

```
                    CURRENTSURFLABEL=V
                    EXIT
                 ENDIF
              ENDIF
 3818      CONTINUE
           IF (CURRENTSURFLABEL.EQ.0) THEN
              SPACERBOTTOMSURF=SN
              SURFTYPESPEC(SN)='PZ'
              SURFVALUESPEC(SN)=CURRENTSURF
              SN=SN+1
           ELSE
              SPACERBOTTOMSURF=CURRENTSURFLABEL
           ENDIF
           WATERREGIONTOPSURF=SPACERBOTTOMSURF
           CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
           CURRENTSURFLABEL=0
           DO 3820 V=1,(SN-1)
              IF (SURFTYPESPEC(V).EQ.'PZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                 ENDIF
              ENDIF
 3820      CONTINUE
           IF (CURRENTSURFLABEL.EQ.0) THEN
              WATERREGIONBOTTOMSURF=SN
              SURFTYPESPEC(SN)='PZ'
              SURFVALUESPEC(SN)=CURRENTSURF
              SN=SN+1
           ELSE
              WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
           ENDIF
        ELSEIF (SPN.EQ.NUMOFSPACERS(DESNUM(COLUMN,ROW))) THEN
           SPACERTOPSURF=WATERREGIONBOTTOMSURF
           CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c     SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
           CURRENTSURFLABEL=0
           DO 3822 V=1,(SN-1)
              IF (SURFTYPESPEC(V).EQ.'PZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                 ENDIF
              ENDIF
 3822      CONTINUE
           IF (CURRENTSURFLABEL.EQ.0) THEN
              SPACERBOTTOMSURF=SN
              SURFTYPESPEC(SN)='PZ'
              SURFVALUESPEC(SN)=CURRENTSURF
              SN=SN+1
           ELSE
              SPACERBOTTOMSURF=CURRENTSURFLABEL
           ENDIF
           WATERREGIONTOPSURF=SPACERBOTTOMSURF
```

```
              WATERREGIONBOTTOMSURF=NODEBOTTOMSURF
           ENDIF
*     Write the current homogenized spacer region cell in this IT universe.
        IF (ITSPLIT.NE.1) THEN
           WRITE(30,3824) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c     (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)), ITORSURF,
     c     (-1*SPACERTOPSURF), SPACERBOTTOMSURF, ITUNIV(COLUMN,ROW),
     c     SPN
 3824      FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c     ' IMP:N=1 U=',I4,
     c     '    $ Homogenized region for spacer ',I2)
           LN=LN+1
*     Write the water region cell below the current homogenized spacer cell
in this IT universe.
           WRITE(30,3826) LN, BMODML, (-1*MODDENSITY), ITORSURF,
     c     (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c     ITUNIV(COLUMN,ROW)
 3826      FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c     ' IMP:N=1 U=',I4,'    $ Borated moderator')
           LN=LN+1
        ELSEIF (ITSPLIT.EQ.1) THEN
         DO 3852 SECT=1,NUMOFITAXS(DESNUM(COLUMN,ROW))
           IF ((SURFVALUESPEC(ITSECTTOPSURF(SECT)).GT.
     c     SURFVALUESPEC(SPACERTOPSURF)).AND.
     c     (SURFVALUESPEC(ITSECTBOTSURF(SECT)).LT.
     c     SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
           WRITE(30,3828) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c     (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c     ITSECTORSURF(SECT),
     c     (-1*SPACERTOPSURF), SPACERBOTTOMSURF, ITUNIV(COLUMN,ROW),
     c     SPN
 3828      FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c     ' IMP:N=1 U=',I4,
     c     '    $ Homogenized region for spacer ',I2)
           LN=LN+1
           ELSEIF ((SURFVALUESPEC(ITSECTTOPSURF(SECT)).EQ.
     c     SURFVALUESPEC(SPACERTOPSURF)).AND.
     c     (SURFVALUESPEC(ITSECTBOTSURF(SECT)).LT.
     c     SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
           WRITE(30,3830) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c     (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c     ITSECTORSURF(SECT),
     c     (-1*SPACERTOPSURF), SPACERBOTTOMSURF, ITUNIV(COLUMN,ROW),
     c     SPN
 3830      FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c     ' IMP:N=1 U=',I4,
     c     '    $ Homogenized region for spacer ',I2)
           LN=LN+1
           ELSEIF ((SURFVALUESPEC(ITSECTTOPSURF(SECT)).EQ.
     c     SURFVALUESPEC(SPACERTOPSURF)).AND.
     c     (SURFVALUESPEC(ITSECTBOTSURF(SECT)).EQ.
     c     SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
           WRITE(30,3832) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c     (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
```

```
     c        ITSECTORSURF(SECT),
     c        (-1*SPACERTOPSURF), SPACERBOTTOMSURF, ITUNIV(COLUMN,ROW),
     c        SPN
 3832         FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,
     c        '   $ Homogenized region for spacer ',I2)
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(ITSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(SPACERTOPSURF)).AND. .
     c        (SURFVALUESPEC(ITSECTBOTSURF(SECT)).EQ.
     c        SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
              WRITE(30,3834) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c        (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c        ITSECTORSURF(SECT),
     c        (-1*SPACERTOPSURF), SPACERBOTTOMSURF, ITUNIV(COLUMN,ROW),
     c        SPN
 3834         FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,
     c        '   $ Homogenized region for spacer ',I2)
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(ITSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(SPACERTOPSURF)).AND.
     c        (SURFVALUESPEC(ITSECTBOTSURF(SECT)).GT.
     c        SURFVALUESPEC(SPACERBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(ITSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(SPACERTOPSURF))) THEN
              WRITE(30,3836) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c        (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c        ITSECTORSURF(SECT),
     c        (-1*SPACERTOPSURF), ITSECTBOTSURF(SECT),
     c        ITUNIV(COLUMN,ROW), SPN
 3836         FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,
     c        '   $ Homogenized region for spacer ',I2)
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(ITSECTTOPSURF(SECT)).LT.
     c        SURFVALUESPEC(SPACERTOPSURF)).AND.
     c        (SURFVALUESPEC(ITSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(SPACERBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(ITSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
              WRITE(30,3838) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c        (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c        ITSECTORSURF(SECT),
     c        (-1*ITSECTTOPSURF(SECT)), SPACERBOTTOMSURF,
     c        ITUNIV(COLUMN,ROW), SPN
 3838         FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,
     c        '   $ Homogenized region for spacer ',I2)
              LN=LN+1
               ENDIF
*     Write the water region cell below the current homogenized spacer cell
in this GT universe.
              IF ((SURFVALUESPEC(ITSECTTOPSURF(SECT)).GT.
```

```
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(ITSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,3840) LN, BMODML, (-1*MODDENSITY),
     c        ITSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        ITUNIV(COLUMN,ROW)
 3840         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(ITSECTTOPSURF(SECT)).EQ.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(ITSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,3842) LN, BMODML, (-1*MODDENSITY),
     c        ITSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        ITUNIV(COLUMN,ROW)
 3842         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(ITSECTTOPSURF(SECT)).EQ.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(ITSECTBOTSURF(SECT)).EQ.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,3844) LN, BMODML, (-1*MODDENSITY),
     c        ITSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        ITUNIV(COLUMN,ROW)
 3844         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(ITSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(ITSECTBOTSURF(SECT)).EQ.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,3846) LN, BMODML, (-1*MODDENSITY),
     c        ITSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        ITUNIV(COLUMN,ROW)
 3846         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(ITSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(ITSECTBOTSURF(SECT)).GT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(ITSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(WATERREGIONTOPSURF))) THEN
              WRITE(30,3848) LN, BMODML, (-1*MODDENSITY),
     c        ITSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), ITSECTBOTSURF(SECT),
     c        ITUNIV(COLUMN,ROW)
 3848         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
```

```
     c      ' IMP:N=1 U=',I4,'    $ Borated moderator region')
            LN=LN+1
             ELSEIF ((SURFVALUESPEC(ITSECTTOPSURF(SECT)).LT.
     c      SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c      (SURFVALUESPEC(ITSECTBOTSURF(SECT)).LT.
     c      SURFVALUESPEC(WATERREGIONBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(ITSECTTOPSURF(SECT)).GT.
     c      SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
            WRITE(30,3850) LN, BMODML, (-1*MODDENSITY),
     c      ITSECTORSURF(SECT),
     c      (-1*ITSECTTOPSURF(SECT)), WATERREGIONBOTTOMSURF,
     c      ITUNIV(COLUMN,ROW)
 3850       FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,'    $ Borated moderator region')
            LN=LN+1
             ENDIF
 3852       CONTINUE
         ENDIF
 3854    CONTINUE
      ENDIF
 3856    CONTINUE
 3858 CONTINUE
*  Write the specifications for the CR universes that are
*  required to fill the assembly layout specifications previously defined.
      IF ((BANDW.EQ..TRUE.).OR.
     c ((WESTINGHOUSE.EQ..TRUE.).AND.
     c ((GTSPLIT.NE.1).AND.(HYBRID.NE.1)))) THEN
      DO 5050 ROW=1,50
         DO 5040 COLUMN=1,50
*     Write the CR universe specification for the assembly if it
*     contains a unique CR material or unique CR position.
*
      IF (CRUNIQUE(COLUMN,ROW).EQ..TRUE.) THEN
*  Write the CR specification header.
         WRITE(30,4020)
 4020    FORMAT(T1,'C')
         WRITE(30,4030) ASSYID(COLUMN,ROW)
 4030    FORMAT(T1,
     c   'C   CONTROL ROD UNIVERSE SPECIFICATION FOR ASSEMBLY ',A5)
         WRITE(30,4040)
 4040    FORMAT(T1,'C')
*     Define the upper end-fitting bottom surface.
              CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c        ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
              CURRENTSURFLABEL=0
              DO 4042 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                  ENDIF
                ENDIF
 4042         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
```

```
                  UEFBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  UEFBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
*     Define the upper end-fitting top surface.
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),1)+
     c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 4044 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 4044          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  UEFTOPSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  UEFTOPSURF=CURRENTSURFLABEL
               ENDIF
               CRABSML=MN
*     Define the CR absorber radius.
               CURRENTSURF=CRADIM(BANKNUM(COLUMN,ROW),1)
               CURRENTSURFLABEL=0
               DO 4046 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 4046          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  CRABSSURF=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  CRABSSURF=CURRENTSURFLABEL
               ENDIF
*     Define the CR absorber top surface.
               CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c         CRADIM(BANKNUM(COLUMN,ROW),4)+
     c         CRADIM(BANKNUM(COLUMN,ROW),5)
               IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                  CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
```

```
            ENDIF
            CURRENTSURFLABEL=0
            DO 4048 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 4048       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               CRABSTOPSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               CRABSTOPSURF=CURRENTSURFLABEL
            ENDIF
*     Define the CR absorber bottom surface.
            CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c      CRADIM(BANKNUM(COLUMN,ROW),4)
            IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
               CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
            ENDIF
            CURRENTSURFLABEL=0
            DO 4050 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 4050       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               CRABSBOTTOMSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               CRABSBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
*     Define the CR cladding inner radius.
            CURRENTSURF=CRADIM(BANKNUM(COLUMN,ROW),2)
            CURRENTSURFLABEL=0
            DO 4060 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'CZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 4060       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               CRCLADIRSURF=SN
```

```
              SURFTYPESPEC(SN)='CZ'
              SURFVALUESPEC(SN)=CURRENTSURF
              SN=SN+1
           ELSE
              CRCLADIRSURF=CURRENTSURFLABEL
           ENDIF
*     Define the CR cladding outer radius.
           CURRENTSURF=CRADIM(BANKNUM(COLUMN,ROW),3)
           CURRENTSURFLABEL=0
           DO 4070 V=1,(SN-1)
              IF (SURFTYPESPEC(V).EQ.'CZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                 ENDIF
              ENDIF
 4070      CONTINUE
           IF (CURRENTSURFLABEL.EQ.0) THEN
              CRCLADORSURF=SN
              SURFTYPESPEC(SN)='CZ'
              SURFVALUESPEC(SN)=CURRENTSURF
              SN=SN+1
           ELSE
              CRCLADORSURF=CURRENTSURFLABEL
           ENDIF
*     Define the CR cladding top surface.
           CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c     CRADIM(BANKNUM(COLUMN,ROW),4)+
     c     CRADIM(BANKNUM(COLUMN,ROW),5)+
     c     CRADIM(BANKNUM(COLUMN,ROW),7)
           IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
              CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
           ENDIF
           CURRENTSURFLABEL=0
           DO 4080 V=1,(SN-1)
              IF (SURFTYPESPEC(V).EQ.'PZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                 ENDIF
              ENDIF
 4080      CONTINUE
           IF (CURRENTSURFLABEL.EQ.0) THEN
              CRCLADTOPSURF=SN
              SURFTYPESPEC(SN)='PZ'
              SURFVALUESPEC(SN)=CURRENTSURF
              SN=SN+1
           ELSE
              CRCLADTOPSURF=CURRENTSURFLABEL
           ENDIF
*     Define the CR cladding bottom surface.
           CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c     CRADIM(BANKNUM(COLUMN,ROW),4)-
     c     CRADIM(BANKNUM(COLUMN,ROW),6)
```

```
            IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
               CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
            ENDIF
            CURRENTSURFLABEL=0
            DO 4090 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                   CURRENTSURFLABEL=V
                   EXIT
                 ENDIF
               ENDIF
4090        CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               CRCLADBOTTOMSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               CRCLADBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
        IF (SURFVALUESPEC(CRABSBOTTOMSURF).LT.
     c  SURFVALUESPEC(UEFTOPSURF)) THEN
* Check Control Rod Absorber Material
            DO 4123 C=1,CRABSMAT(BANKNUM(COLUMN,ROW),2)
               IF (C.EQ.1) THEN
                 WRITE(200,4121) CRABSML,
     c           CRABSZAIDS(BANKNUM(COLUMN,ROW),C),
     c           (-1*CRABSWTS(BANKNUM(COLUMN,ROW),C)),
     c           ASSYID(COLUMN,ROW)
4121             FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c           '    $ Control Rod Absorber Material in Assembly ',
     c           A5)
               ELSE
                 WRITE(200,4122) CRABSZAIDS(BANKNUM(COLUMN,ROW),C),
     c           (-1*CRABSWTS(BANKNUM(COLUMN,ROW),C))
4122             FORMAT(T9,A9,3X,G14.6)
               ENDIF
4123        CONTINUE
            MN=MN+1
*    Write the CR absorber cell in this CR universe.
           WRITE(30,4130) LN, CRABSML,
     c     (-1*CRABSMAT(BANKNUM(COLUMN,ROW),1)), (-1*CRABSSURF),
     c     (-1*CRABSTOPSURF), CRABSBOTTOMSURF,
     c     CRAUNIV(COLUMN,ROW)
4130       FORMAT(T1,I4,T6,I4,T11,F10.6,T25,I4,1X,I4,1X,I4,
     c     ' IMP:N=1 U=',I3,'    $ Control rod absorber material')
           LN=LN+1
*    Write the absorber-to-cladding gap cell in this CR universe.
           WRITE(30,4140) LN, (-1*CRCLADIRSURF), CRABSSURF,
     c     (-1*CRABSTOPSURF),
     c     CRABSBOTTOMSURF, CRAUNIV(COLUMN,ROW)
4140       FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c     ' IMP:N=1 U=',I3,'    $ Absorber-to-cladding gap')
           LN=LN+1
```

```
         ENDIF
*      Write the CR cladding cell in this CR universe.
*      Determine if the CR cladding material specification has
*      previously been defined.  If it has been previously defined, determine
*      the cladding material specification label.
            CLADMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 4160 RO=1,(ROW-1)
                  DO 4150 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA   ') THEN
                        IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  CRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRCLADML(COLUMN,ROW)=CRCLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
 4150             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 4160          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 4180 RO=ROW,ROW
                     DO 4170 CO=1,(COLUMN-1)
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA   ') THEN
                        IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  CRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRCLADML(COLUMN,ROW)=CRCLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
 4170                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 4180             CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 4200 RO=1,(ROW-1)
                  DO 4190 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA   ') THEN
                        IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  CRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
```

```
                                    LEAVE=.TRUE.
                                    CRCLADML(COLUMN,ROW)=CRCLADML(CO,RO)
                                    EXIT
                                 ENDIF
                              ENDIF
                           ENDIF
 4190                   CONTINUE
                        IF (LEAVE.EQ..TRUE.) THEN
                           EXIT
                        ENDIF
 4200                CONTINUE
                  ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                     DO 4220 RO=1,1
                        DO 4210 CO=1,(COLUMN-1)
                           IF (BANKNUM(CO,RO).NE.0) THEN
                           IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA  ') THEN
                              IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                        CRCLADMAT(BANKNUM(CO,RO))) THEN
                                 CLADMLUNIQUE=.FALSE.
                                 LEAVE=.TRUE.
                                 CRCLADML(COLUMN,ROW)=CRCLADML(CO,RO)
                                 EXIT
                              ENDIF
                           ENDIF
                           ENDIF
 4210                   CONTINUE
                        IF (LEAVE.EQ..TRUE.) THEN
                           EXIT
                        ENDIF
 4220                CONTINUE
                  ENDIF
               IF (SURFVALUESPEC(CRCLADBOTTOMSURF).LT.
     c         SURFVALUESPEC(UEFTOPSURF)) THEN
                  IF (CLADMLUNIQUE.EQ..TRUE.) THEN
                     CRCLADML(COLUMN,ROW)=MN
*  Check Control Rod Cladding
                        IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
                           DO 4222 C=1,2
                              IF (C.EQ.1) THEN
                                 WRITE(200,9300) CRCLADML(COLUMN,ROW)
                              ELSEIF (C.EQ.2) THEN
                                 WRITE(200,9301)
                                 WRITE(200,7000)
                                 WRITE(200,7001)
                                 WRITE(200,7002)
                                 WRITE(200,9302)
                                 WRITE(200,7003)
                                 WRITE(200,7004)
                                 WRITE(200,7005)
                                 WRITE(200,9303)
                                 WRITE(200,9304)
                              ENDIF
 4222                      CONTINUE
                        ELSEIF (CRCLADMAT(BANKNUM(COLUMN,ROW))
```

```
     c              .EQ.2) THEN
                       DO 4224 C=1,2
                          IF (C.EQ.1) THEN
                             WRITE(200,9305) CRCLADML(COLUMN,ROW)
                          ELSEIF (C.EQ.2) THEN
                             WRITE(200,9306)
                             WRITE(200,9307)
                             WRITE(200,9308)
                             WRITE(200,9309)
                             WRITE(200,9310)
                             WRITE(200,7006)
                             WRITE(200,7007)
                             WRITE(200,7008)
                             WRITE(200,9311)
                             WRITE(200,9312)
                             WRITE(200,7009)
                             WRITE(200,7010)
                             WRITE(200,7011)
                             WRITE(200,9313)
                             WRITE(200,7012)
                             WRITE(200,7013)
                             WRITE(200,7014)
                             WRITE(200,7015)
                          ENDIF
 4224                  CONTINUE
                    ELSEIF (CRCLADMAT(BANKNUM(COLUMN,ROW))
     c              .EQ.3) THEN
                       DO 4226 C=1,2
                          IF (C.EQ.1) THEN
                             WRITE(200,9314) CRCLADML(COLUMN,ROW)
                          ELSEIF (C.EQ.2) THEN
                             WRITE(200,9315)
                             WRITE(200,9316)
                             WRITE(200,9317)
                             WRITE(200,9318)
                             WRITE(200,7016)
                             WRITE(200,7017)
                             WRITE(200,7018)
                             WRITE(200,9319)
                             WRITE(200,9320)
                             WRITE(200,7019)
                             WRITE(200,7020)
                             WRITE(200,7021)
                             WRITE(200,9321)
                             WRITE(200,7022)
                             WRITE(200,7023)
                             WRITE(200,7024)
                             WRITE(200,7025)
                             WRITE(200,9322)
                             WRITE(200,9323)
                             WRITE(200,9324)
                             WRITE(200,9325)
                             WRITE(200,9326)
                             WRITE(200,9327)
```

```
                           WRITE(200,7026)
                           WRITE(200,9328)
                           WRITE(200,9329)
                           WRITE(200,9330)
                        ENDIF
 4226                CONTINUE
                  ENDIF
               MN=MN+1
            ENDIF
            IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
               CLADRHO=6.56
            ELSEIF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.2) THEN
               CLADRHO=7.90
            ELSEIF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.3) THEN
               CLADRHO=8.19
            ENDIF
            WRITE(30,4238) LN, CRCLADML(COLUMN,ROW), (-1*CLADRHO),
     c      CRCLADIRSURF,
     c      (-1*CRCLADORSURF), (-1*CRCLADTOPSURF), CRCLADBOTTOMSURF,
     c      CRAUNIV(COLUMN,ROW)
 4238       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Control rod cladding')
            LN=LN+1
         ENDIF
*     Write the CR upper plenum cell in this CR universe.
*     Determine if the CR upper plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the upper plenum material specification label.
            CRUPMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 4250 RO=1,(ROW-1)
                  DO 4240 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                           CRUPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 4240             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 4250          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 4270 RO=ROW,ROW
                     DO 4260 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
                           IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                              CRUPMLUNIQUE=.FALSE.
```

```
                                LEAVE=.TRUE.
                                CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                                EXIT
                             ENDIF
                          ENDIF
4260                   CONTINUE
                       IF (LEAVE.EQ..TRUE.) THEN
                          EXIT
                       ENDIF
4270                CONTINUE
                 ENDIF
              ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                 DO 4290 RO=1,(ROW-1)
                    DO 4280 CO=1,50
                       IF (BANKNUM(CO,RO).NE.0) THEN
                          IF (BANKNUM(COLUMN,ROW).EQ.
     c                       BANKNUM(CO,RO)) THEN
                             CRUPMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                             EXIT
                          ENDIF
                       ENDIF
4280                CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
4290             CONTINUE
              ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                 DO 4310 RO=1,1
                    DO 4300 CO=1,(COLUMN-1)
                       IF (BANKNUM(CO,RO).NE.0) THEN
                          IF (BANKNUM(COLUMN,ROW).EQ.
     c                       BANKNUM(CO,RO)) THEN
                             CRUPMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                             EXIT
                          ENDIF
                       ENDIF
4300                CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
4310             CONTINUE
              ENDIF
           IF (SURFVALUESPEC(CRABSTOPSURF).LT.
     c     SURFVALUESPEC(UEFTOPSURF)) THEN
              IF (CRUPMLUNIQUE.EQ..TRUE.) THEN
                 CRUPML(COLUMN,ROW)=MN
*  Check Control Rod Upper Plenum Regions
                    DO 4313 C=1,CRUPLENMAT(BANKNUM(COLUMN,ROW),2)
                       IF (C.EQ.1) THEN
                          WRITE(200,4311) CRUPML(COLUMN,ROW),
```

```
     c                    CRUPZS(BANKNUM(COLUMN,ROW),C),
     c                    (-1*CRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 4311                     FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                    '    $ Control Rod Upper Plenum')
                       ELSE
                          WRITE(200,4312)
     c                    CRUPZS(BANKNUM(COLUMN,ROW),C),
     c                    (-1*CRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 4312                     FORMAT(T9,A9,3X,G14.6)
                       ENDIF
 4313               CONTINUE
                 MN=MN+1
              ENDIF
              WRITE(30,4320) LN, CRUPML(COLUMN,ROW),
     c        (-1*CRUPLENMAT(BANKNUM(COLUMN,ROW),1)),
     c        CRABSTOPSURF,
     c        (-1*CRCLADTOPSURF), (-1*CRCLADIRSURF),
     c        CRAUNIV(COLUMN,ROW)
 4320         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Control rod upper plenum')
              LN=LN+1
           ENDIF
*     Write the CR lower plenum cell in this CR universe.
*     Determine if the CR lower plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the lower plenum material specification label.
              CRLPMLUNIQUE=.TRUE.
              LEAVE=.FALSE.
              IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                 DO 4340 RO=1,(ROW-1)
                    DO 4330 CO=1,50
                       IF (BANKNUM(CO,RO).NE.0) THEN
                          IF (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO)) THEN
                             CRLPMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                             EXIT
                          ENDIF
                       ENDIF
 4330               CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
 4340            CONTINUE
                 IF (LEAVE.EQ..FALSE.) THEN
                    DO 4360 RO=ROW,ROW
                       DO 4350 CO=1,(COLUMN-1)
                          IF (BANKNUM(CO,RO).NE.0) THEN
                             IF (BANKNUM(COLUMN,ROW).EQ.
     c                       BANKNUM(CO,RO)) THEN
                                CRLPMLUNIQUE=.FALSE.
                                LEAVE=.TRUE.
                                CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                                EXIT
```

```
                         ENDIF
                      ENDIF
4350               CONTINUE
                   IF (LEAVE.EQ..TRUE.) THEN
                      EXIT
                   ENDIF
4360            CONTINUE
             ENDIF
          ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
             DO 4380 RO=1,(ROW-1)
                DO 4370 CO=1,50
                   IF (BANKNUM(CO,RO).NE.0) THEN
                      IF (BANKNUM(COLUMN,ROW).EQ.
     c                   BANKNUM(CO,RO)) THEN
                         CRLPMLUNIQUE=.FALSE.
                         LEAVE=.TRUE.
                         CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                         EXIT
                      ENDIF
                   ENDIF
4370            CONTINUE
                IF (LEAVE.EQ..TRUE.) THEN
                   EXIT
                ENDIF
4380         CONTINUE
          ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
             DO 4400 RO=1,1
                DO 4390 CO=1,(COLUMN-1)
                   IF (BANKNUM(CO,RO).NE.0) THEN
                      IF (BANKNUM(COLUMN,ROW).EQ.
     c                   BANKNUM(CO,RO)) THEN
                         CRLPMLUNIQUE=.FALSE.
                         LEAVE=.TRUE.
                         CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                         EXIT
                      ENDIF
                   ENDIF
4390            CONTINUE
                IF (LEAVE.EQ..TRUE.) THEN
                   EXIT
                ENDIF
4400         CONTINUE
          ENDIF
       IF (SURFVALUESPEC(CRCLADBOTTOMSURF).LT.
     c SURFVALUESPEC(UEFTOPSURF)) THEN
          IF (CRLPMLUNIQUE.EQ..TRUE.) THEN
             CRLPML(COLUMN,ROW)=MN
*  Check Control Rod Lower Plenum Regions
                DO 4403 C=1,CRLPLENMAT(BANKNUM(COLUMN,ROW),2)
                   IF (C.EQ.1) THEN
                      WRITE(200,4401) CRLPML(COLUMN,ROW),
     c                CRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                (-1*CRLPLENWTS(BANKNUM(COLUMN,ROW),C))
4401                  FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
```

```
     c                    '   $ Control Rod Lower Plenum')
                        ELSE
                           WRITE(200,4402)
     c                     CRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                     (-1*CRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 4402                      FORMAT(T9,A9,3X,G14.6)
                        ENDIF
 4403                CONTINUE
                  MN=MN+1
               ENDIF
               WRITE(30,4410) LN, CRLPML(COLUMN,ROW),
     c         (-1*CRLPLENMAT(BANKNUM(COLUMN,ROW),1)), CRCLADBOTTOMSURF,
     c         (-1*CRABSBOTTOMSURF), (-1*CRCLADIRSURF),
     c         CRAUNIV(COLUMN,ROW)
 4410          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Control rod lower plenum')
               LN=LN+1
            ENDIF
*        Define the GT top surface.
                  CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),3)
                  IF (CURRENTSURF.GT.SURFVALUESPEC(UEFTOPSURF)) THEN
                     CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c               ENDFITHEIGHT(DESNUM(COLUMN,ROW),1)
                  ENDIF
                  CURRENTSURFLABEL=0
                  DO 4420 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                           CURRENTSURFLABEL=V
                           EXIT
                        ENDIF
                     ENDIF
 4420             CONTINUE
                  IF (CURRENTSURFLABEL.EQ.0) THEN
                     GTTOPSURF=SN
                     SURFTYPESPEC(SN)='PZ'
                     SURFVALUESPEC(SN)=CURRENTSURF
                     SN=SN+1
                  ELSE
                     GTTOPSURF=CURRENTSURFLABEL
                  ENDIF
*        Define the GT bottom surface.
                  CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),4)
                  CURRENTSURFLABEL=0
                  DO 4430 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                           CURRENTSURFLABEL=V
                           EXIT
                        ENDIF
                     ENDIF
 4430             CONTINUE
                  IF (CURRENTSURFLABEL.EQ.0) THEN
                     GTBOTSURF=SN
```

```
            SURFTYPESPEC(SN)='PZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
         ELSE
            GTBOTSURF=CURRENTSURFLABEL
         ENDIF
*     Define the GT outer radius surface.
         CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),2)
         CURRENTSURFLABEL=0
         DO 4440 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'CZ') THEN
       IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
 4440    CONTINUE
         IF (CURRENTSURFLABEL.EQ.0) THEN
            GTORSURF=SN
            SURFTYPESPEC(SN)='CZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
         ELSE
            GTORSURF=CURRENTSURFLABEL
         ENDIF
*     Define the GT inner radius surface.
         CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),1)
         CURRENTSURFLABEL=0
         DO 4450 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'CZ') THEN
       IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
 4450    CONTINUE
         IF (CURRENTSURFLABEL.EQ.0) THEN
            GTIRSURF=SN
            SURFTYPESPEC(SN)='CZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
         ELSE
            GTIRSURF=CURRENTSURFLABEL
         ENDIF
*     Define the lower end-fitting top surface.
         CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
         CURRENTSURFLABEL=0
         DO 4460 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'PZ') THEN
       IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
```

```
4460          CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 CRLEFTOPSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 CRLEFTOPSURF=CURRENTSURFLABEL
              ENDIF
*     Write the lower end-fitting cell specification for this CR universe.
           IF (SURFVALUESPEC(GTBOTSURF).GE.
     c     ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)) THEN
              WRITE(30,4550) LN, FRLEFML(COLUMN,ROW),
     c        (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*CRLEFTOPSURF),
     c        CRAUNIV(COLUMN,ROW)
4550          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,' IMP:N=1 U=',I3,
     c        '    $ Lower end-fitting')
              LN=LN+1
           ELSE
              WRITE(30,4560) LN, FRLEFML(COLUMN,ROW),
     c        (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*CRLEFTOPSURF),
     c        GTORSURF, CRAUNIV(COLUMN,ROW)
4560          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'    $ Lower end-fitting')
              LN=LN+1
              WRITE(30,4570) LN, FRLEFML(COLUMN,ROW),
     c        (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*GTBOTSURF),
     c        (-1*GTORSURF), CRAUNIV(COLUMN,ROW)
4570          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'    $ Lower end-fitting')
              LN=LN+1
           ENDIF
*     Write the upper end-fitting cell specification for this CR universe.
           IF ((SURFVALUESPEC(GTTOPSURF).LE.
     c     SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c     (SURFVALUESPEC(CRCLADTOPSURF).GE.
     c     SURFVALUESPEC(UEFTOPSURF))) THEN
              WRITE(30,4572) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c        (-1*UEFTOPSURF), CRCLADORSURF, CRAUNIV(COLUMN,ROW)
4572          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
              LN=LN+1
           ELSEIF ((SURFVALUESPEC(GTTOPSURF).LE.
     c     SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c     (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c     SURFVALUESPEC(UEFTOPSURF)).AND.
     c     (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c     SURFVALUESPEC(UEFBOTTOMSURF))) THEN
              WRITE(30,4574) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c        (-1*UEFTOPSURF), CRCLADORSURF, CRAUNIV(COLUMN,ROW)
4574          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
```

```
            LN=LN+1
            WRITE(30,4576) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c      (-1*UEFTOPSURF), (-1*CRCLADORSURF),
     c      CRAUNIV(COLUMN,ROW)
 4576       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
            LN=LN+1
         ELSEIF ((SURFVALUESPEC(GTTOPSURF).LE.
     c   SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c   (SURFVALUESPEC(CRCLADTOPSURF).LE.
     c   SURFVALUESPEC(UEFBOTTOMSURF))) THEN
            WRITE(30,4578) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c      (-1*UEFTOPSURF), CRAUNIV(COLUMN,ROW)
 4578       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
            LN=LN+1
         ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c   SURFVALUESPEC(UEFTOPSURF)).AND.
     c   (SURFVALUESPEC(GTTOPSURF).GT.
     c   SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c   (SURFVALUESPEC(CRCLADTOPSURF).GE.
     c   SURFVALUESPEC(UEFTOPSURF))) THEN
            WRITE(30,4580) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c      (-1*UEFTOPSURF), GTORSURF, CRAUNIV(COLUMN,ROW)
 4580       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
            LN=LN+1
            WRITE(30,4582) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c      (-1*UEFTOPSURF), (-1*GTORSURF), CRCLADORSURF,
     c      CRAUNIV(COLUMN,ROW)
 4582       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      1X,I4,' IMP:N=1 U=',I3,'    $ Upper end-fitting')
            LN=LN+1
         ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c   SURFVALUESPEC(UEFTOPSURF)).AND.
     c   (SURFVALUESPEC(GTTOPSURF).GT.
     c   SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c   (SURFVALUESPEC(CRCLADTOPSURF).GE.
     c   SURFVALUESPEC(GTTOPSURF)).AND.
     c   (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c   SURFVALUESPEC(UEFTOPSURF))) THEN
            WRITE(30,4584) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c      (-1*UEFTOPSURF), GTORSURF, CRAUNIV(COLUMN,ROW)
 4584       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
            LN=LN+1
            WRITE(30,4586) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c      (-1*UEFTOPSURF), CRCLADORSURF, (-1*GTORSURF),
```

```
     c          CRAUNIV(COLUMN,ROW)
 4586           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          1X,I4,' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
                WRITE(30,4588) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*CRCLADORSURF),
     c          CRAUNIV(COLUMN,ROW)
 4588           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).EQ.
     c       SURFVALUESPEC(GTTOPSURF))) THEN
                WRITE(30,4590) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTORSURF, CRAUNIV(COLUMN,ROW)
 4590           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
                WRITE(30,4592) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTORSURF),
     c          CRAUNIV(COLUMN,ROW)
 4592           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c       SURFVALUESPEC(GTTOPSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                WRITE(30,4594) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTORSURF, CRAUNIV(COLUMN,ROW)
 4594           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
                WRITE(30,4596) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTORSURF),
     c          CRAUNIV(COLUMN,ROW)
 4596           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
                WRITE(30,4598) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), (-1*GTTOPSURF),
     c          CRCLADTOPSURF, (-1*CRCLADORSURF),
```

```
     c           CRAUNIV(COLUMN,ROW)
 4598            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                 LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c        SURFVALUESPEC(UEFTOPSURF)).AND.
     c        (SURFVALUESPEC(GTTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(CRCLADTOPSURF).LE.
     c        SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                 WRITE(30,4600) LN, FRUEFML(COLUMN,ROW),
     c           (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c           (-1*UEFTOPSURF), GTORSURF, CRAUNIV(COLUMN,ROW)
 4600            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                 LN=LN+1
                 WRITE(30,4602) LN, FRUEFML(COLUMN,ROW),
     c           (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c           (-1*UEFTOPSURF), (-1*GTORSURF),
     c           CRAUNIV(COLUMN,ROW)
 4602            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                 LN=LN+1
                 WRITE(30,4604) LN, FRUEFML(COLUMN,ROW),
     c           (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), (-1*GTTOPSURF),
     c           UEFBOTTOMSURF, (-1*CRCLADORSURF),
     c           CRAUNIV(COLUMN,ROW)
 4604            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                 LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTTOPSURF).GE.
     c        SURFVALUESPEC(UEFTOPSURF)).AND.
     c        (SURFVALUESPEC(CRCLADTOPSURF).GE.
     c        SURFVALUESPEC(UEFTOPSURF))) THEN
                 WRITE(30,4606) LN, FRUEFML(COLUMN,ROW),
     c           (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c           (-1*UEFTOPSURF), GTORSURF, CRAUNIV(COLUMN,ROW)
 4606            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                 LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTTOPSURF).GE.
     c        SURFVALUESPEC(UEFTOPSURF)).AND.
     c        (SURFVALUESPEC(CRCLADTOPSURF).LE.
     c        SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                 WRITE(30,4608) LN, FRUEFML(COLUMN,ROW),
     c           (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c           (-1*UEFTOPSURF), GTORSURF, CRAUNIV(COLUMN,ROW)
 4608            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                 LN=LN+1
                 WRITE(30,4610) LN, FRUEFML(COLUMN,ROW),
     c           (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c           (-1*UEFTOPSURF), (-1*CRCLADORSURF), CRAUNIV(COLUMN,ROW)
 4610            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
```

```
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTTOPSURF).GE.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,4612) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTORSURF, CRAUNIV(COLUMN,ROW)
 4612           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
                WRITE(30,4614) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*CRCLADORSURF), CRAUNIV(COLUMN,ROW)
 4614           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
             ENDIF
*     Write the GT material cell in this CR universe.
*     Determine if the GT material specification has
*     previously been defined.  If it has been previously defined, determine
*     the material specification label.
             CLADMLUNIQUE=.TRUE.
             LEAVE=.FALSE.
             IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                DO 4710 RO=1,(ROW-1)
                   DO 4700 CO=1,50
                      IF (DESNUM(CO,RO).NE.0) THEN
                         IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                   GTMAT(DESNUM(CO,RO))) THEN
                            CLADMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            GTML(COLUMN,ROW)=GTML(CO,RO)
                            EXIT
                         ENDIF
                      ENDIF
 4700              CONTINUE
                   IF (LEAVE.EQ..TRUE.) THEN
                      EXIT
                   ENDIF
 4710           CONTINUE
                IF (LEAVE.EQ..FALSE.) THEN
                   DO 4730 RO=ROW,ROW
                      DO 4720 CO=1,(COLUMN-1)
                      IF (DESNUM(CO,RO).NE.0) THEN
                         IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                   GTMAT(DESNUM(CO,RO))) THEN
                            CLADMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            GTML(COLUMN,ROW)=GTML(CO,RO)
                            EXIT
```

```
                    ENDIF
                 ENDIF
 4720            CONTINUE
                 IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                 ENDIF
 4730         CONTINUE
           ENDIF
        ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
           DO 4750 RO=1,(ROW-1)
              DO 4740 CO=1,50
                 IF (DESNUM(CO,RO).NE.0) THEN
                    IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c              GTMAT(DESNUM(CO,RO))) THEN
                       CLADMLUNIQUE=.FALSE.
                       LEAVE=.TRUE.
                       GTML(COLUMN,ROW)=GTML(CO,RO)
                       EXIT
                    ENDIF
                 ENDIF
 4740         CONTINUE
              IF (LEAVE.EQ..TRUE.) THEN
                 EXIT
              ENDIF
 4750      CONTINUE
        ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
           DO 4770 RO=1,1
              DO 4760 CO=1,(COLUMN-1)
                 IF (DESNUM(CO,RO).NE.0) THEN
                    IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c              GTMAT(DESNUM(CO,RO))) THEN
                       CLADMLUNIQUE=.FALSE.
                       LEAVE=.TRUE.
                       GTML(COLUMN,ROW)=GTML(CO,RO)
                       EXIT
                    ENDIF
                 ENDIF
 4760         CONTINUE
              IF (LEAVE.EQ..TRUE.) THEN
                 EXIT
              ENDIF
 4770      CONTINUE
        ENDIF
        IF (CLADMLUNIQUE.EQ..TRUE.) THEN
           GTML(COLUMN,ROW)=MN
* Check Guide Tube Material
              IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.1) THEN
                 DO 4772 C=1,2
                    IF (C.EQ.1) THEN
                       WRITE(200,9300) GTML(COLUMN,ROW)
                    ELSEIF (C.EQ.2) THEN
                       WRITE(200,9301)
                       WRITE(200,7000)
                       WRITE(200,7001)
```

```
                  WRITE(200,7002)
                  WRITE(200,9302)
                  WRITE(200,7003)
                  WRITE(200,7004)
                  WRITE(200,7005)
                  WRITE(200,9303)
                  WRITE(200,9304)
               ENDIF
4772        CONTINUE
         ELSEIF (GTMAT(DESNUM(COLUMN,ROW))
     c   .EQ.2) THEN
            DO 4774 C=1,2
               IF (C.EQ.1) THEN
                  WRITE(200,9305) GTML(COLUMN,ROW)
               ELSEIF (C.EQ.2) THEN
                  WRITE(200,9306)
                  WRITE(200,9307)
                  WRITE(200,9308)
                  WRITE(200,9309)
                  WRITE(200,9310)
                  WRITE(200,7006)
                  WRITE(200,7007)
                  WRITE(200,7008)
                  WRITE(200,9311)
                  WRITE(200,9312)
                  WRITE(200,7009)
                  WRITE(200,7010)
                  WRITE(200,7011)
                  WRITE(200,9313)
                  WRITE(200,7012)
                  WRITE(200,7013)
                  WRITE(200,7014)
                  WRITE(200,7015)
               ENDIF
4774        CONTINUE
         ELSEIF (GTMAT(DESNUM(COLUMN,ROW))
     c   .EQ.3) THEN
            DO 4776 C=1,2
               IF (C.EQ.1) THEN
                  WRITE(200,9314) GTML(COLUMN,ROW)
               ELSEIF (C.EQ.2) THEN
                  WRITE(200,9315)
                  WRITE(200,9316)
                  WRITE(200,9317)
                  WRITE(200,9318)
                  WRITE(200,7016)
                  WRITE(200,7017)
                  WRITE(200,7018)
                  WRITE(200,9319)
                  WRITE(200,9320)
                  WRITE(200,7019)
                  WRITE(200,7020)
                  WRITE(200,7021)
                  WRITE(200,9321)
```

```
                              WRITE(200,7022)
                              WRITE(200,7023)
                              WRITE(200,7024)
                              WRITE(200,7025)
                              WRITE(200,9322)
                              WRITE(200,9323)
                              WRITE(200,9324)
                              WRITE(200,9325)
                              WRITE(200,9326)
                              WRITE(200,9327)
                              WRITE(200,7026)
                              WRITE(200,9328)
                              WRITE(200,9329)
                              WRITE(200,9330)
                           ENDIF
 4776                   CONTINUE
                     ENDIF
                  MN=MN+1
               ENDIF
               IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.1) THEN
                  CLADRHO=6.56
               ELSEIF (GTMAT(DESNUM(COLUMN,ROW)).EQ.2) THEN
                  CLADRHO=7.90
               ELSEIF (GTMAT(DESNUM(COLUMN,ROW)).EQ.3) THEN
                  CLADRHO=8.19
               ENDIF
               WRITE(30,4788) LN, GTML(COLUMN,ROW), (-1*CLADRHO),
     c         GTIRSURF,
     c         (-1*GTORSURF), (-1*GTTOPSURF), GTBOTSURF,
     c         CRAUNIV(COLUMN,ROW)
 4788          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Guide tube region')
               LN=LN+1
*     Write the moderator cells within the GT in this CR universe.
               WRITE(30,4790) LN, BMODML, (-1*MODDENSITY),
     c         (-1*GTIRSURF),
     c         CRCLADORSURF, (-1*GTTOPSURF), GTBOTSURF,
     c         CRAUNIV(COLUMN,ROW)
 4790          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,
     c         '   $ Borated moderator inside guide tube')
               LN=LN+1
               WRITE(30,4800) LN, BMODML, (-1*MODDENSITY),
     c         (-1*CRCLADORSURF), (-1*CRCLADBOTTOMSURF), GTBOTSURF,
     c         CRAUNIV(COLUMN,ROW)
 4800          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,
     c         '   $ Borated moderator inside guide tube')
               LN=LN+1
*     Loop through the regions above the CR (i.e. the appropriate upper core
regions)
*     Define the upper region lower surface.
                  DO 4950 REGION=1,NUMREGABOVECRA
*     Determine the current upper region's lower surface specification.
```

```
                  IF (REGION.EQ.1) THEN
                     REGIONTOPSURF=SYSTEMTOP
                     CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
     c               REGABOVECRA(REGION,1)
                  ENDIF
                  CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
     c            REGABOVECRA(REGION,1)
                  IF (REGION.EQ.NUMREGABOVECRA) THEN
                     REGIONBOTTOMSURF=UEFTOPSURF
                  ELSE
                  CURRENTSURFLABEL=0
                  DO 4810 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                       ENDIF
                    ENDIF
 4810             CONTINUE
                  IF (CURRENTSURFLABEL.EQ.0) THEN
                    REGIONBOTTOMSURF=SN
                    SURFTYPESPEC(SN)='PZ'
                    SURFVALUESPEC(SN)=CURRENTSURF
                    SN=SN+1
                  ELSE
                    REGIONBOTTOMSURF=CURRENTSURFLABEL
                  ENDIF
                  ENDIF
*    Write the cell specification for the CR universe upper region.
               IF (REGION.EQ.1) THEN
                  WRITE(30,4940) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c            (-1*REGABOVECRA(REGION,2)),
     c            REGIONBOTTOMSURF, CRAUNIV(COLUMN,ROW), REGION
 4940             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
     c            ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                  LN=LN+1
                  REGIONTOPSURF=REGIONBOTTOMSURF
               ELSE
                  WRITE(30,4945) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c            (-1*REGABOVECRA(REGION,2)), (-1*REGIONTOPSURF),
     c            REGIONBOTTOMSURF, CRAUNIV(COLUMN,ROW), REGION
 4945             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c            ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                  LN=LN+1
                  REGIONTOPSURF=REGIONBOTTOMSURF
               ENDIF
 4950          CONTINUE
               SPACHEIGHT=0.0
*    Loop through the spacer and moderator regions along the axial
*    length of the GT (from top to bottom).
          DO 4951 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
             SPACHEIGHT=SPACHEIGHT+SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
 4951     CONTINUE
          DO 5030 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
```

```
*     Define the homogenized spacer region bounding surfaces.
          IF (SPN.EQ.1) THEN
             SPACERTOPSURF=UEFBOTTOMSURF
             CURRENTSURF=SURFVALUESPEC(UEFBOTTOMSURF)-
     c       SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
             CURRENTSURFLABEL=0
             DO 4960 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
 4960        CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                SPACERBOTTOMSURF=SN
                SURFTYPESPEC(SN)='PZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                SPACERBOTTOMSURF=CURRENTSURFLABEL
             ENDIF
             WATERREGIONTOPSURF=SPACERBOTTOMSURF
             CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
             CURRENTSURFLABEL=0
             DO 4970 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
 4970        CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                WATERREGIONBOTTOMSURF=SN
                SURFTYPESPEC(SN)='PZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
             ENDIF
          ELSEIF ((SPN.NE.1).AND.(SPN.NE.
     c    NUMOFSPACERS(DESNUM(COLUMN,ROW)))) THEN
             SPACERTOPSURF=WATERREGIONBOTTOMSURF
             CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c       SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
             CURRENTSURFLABEL=0
             DO 4980 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
```

```
 4980        CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                SPACERBOTTOMSURF=SN
                SURFTYPESPEC(SN)='PZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                SPACERBOTTOMSURF=CURRENTSURFLABEL
             ENDIF
             WATERREGIONTOPSURF=SPACERBOTTOMSURF
             CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
             CURRENTSURFLABEL=0
             DO 4990 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
 4990        CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                WATERREGIONBOTTOMSURF=SN
                SURFTYPESPEC(SN)='PZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
             ENDIF
          ELSEIF (SPN.EQ.NUMOFSPACERS(DESNUM(COLUMN,ROW))) THEN
             SPACERTOPSURF=WATERREGIONBOTTOMSURF
             CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c       SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
             CURRENTSURFLABEL=0
             DO 5000 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
 5000        CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                SPACERBOTTOMSURF=SN
                SURFTYPESPEC(SN)='PZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                SPACERBOTTOMSURF=CURRENTSURFLABEL
             ENDIF
             WATERREGIONTOPSURF=SPACERBOTTOMSURF
             WATERREGIONBOTTOMSURF=NODEBOTTOMSURF
          ENDIF
*     Write the current homogenized spacer region cell in this CR universe.
          WRITE(30,5026) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
```

```
     c        (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)), GTORSURF,
     c        (-1*SPACERTOPSURF), SPACERBOTTOMSURF, CRAUNIV(COLUMN,ROW),
     c        SPN
 5026         FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,
     c        '    $ Homogenized region for spacer ',I2)
              LN=LN+1
*      Write the water region cell below the current homogenized spacer cell
in this CR universe.
              WRITE(30,5027) LN, BMODML, (-1*MODDENSITY), GTORSURF,
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        CRAUNIV(COLUMN,ROW)
 5027         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'    $ Borated moderator')
              LN=LN+1
 5030       CONTINUE
         ENDIF
 5040     CONTINUE
 5050 CONTINUE
         ELSEIF (WESTINGHOUSE.EQ..TRUE.) THEN
            CALL WESTCRA(SPACERDIST, ENDFITHEIGHT, SURFVALUESPEC,
     c CRADIM, CRABSMAT, CRABSWTS, CRUPLENMAT, CRUPLENWTS,
     c CRLPLENMAT, CRLPLENWTS, GTDATA, LEFMAT,
     c UEFMAT, MODDENSITY, REGABOVECRA, SPACERHEIGHT,
     c HOMOSPACERDEN, MN, LN, CRAUNIV, BANKNUM, CRCLADMAT,
     c CRCLADML, CRUPML, CRLPML, FRLEFML, FRUEFML, DESNUM,
     c GTMAT, GTML, BMODML, NUMREGABOVECRA, SYSTEMTOP,
     c FRUREGIONML, NUMOFSPACERS, HOMOSPACMLNUM, ASSYID,
     c SURFTYPESPEC, CRABSZAIDS, BANKDES, CRUPZS, CRLPLENZAIDS,
     c CRUNIQUE, HYBRID, SN, CRAXCLADML, GTAXML, NUMCRAXS,
     c GTAXMAT, NUMOFGTAXS, CRAXCLADMAT, GTUNIV,
     c CRABSAXWTS, CRAXDIM, GTAXDATA, CRABSAXMAT,
     c CRABSAXZAIDS, GTSPLIT, NODEBOTTOMSURF)
        ENDIF
        IF (BANDW.EQ..TRUE.) THEN
*  Write the specifications for the APSR universes that are
*  required to fill the assembly layout specifications previously defined.
        DO 6150 ROW=1,50
           DO 6140 COLUMN=1,50
*       Write the APSR universe specification for the assembly if it
*       contains a unique APSR material or unique APSR position.
*
        IF (APSRUNIQUE(COLUMN,ROW).EQ..TRUE.) THEN
*  Write the CR specification header.
           WRITE(30,5060)
 5060      FORMAT(T1,'C')
           WRITE(30,5070) ASSYID(COLUMN,ROW)
 5070      FORMAT(T1,
    c      'C   AXIAL POWER SHAPING ROD UNIVERSE',
    c      ' SPECIFICATION FOR ASSEMBLY ',A5)
           WRITE(30,5080)
 5080      FORMAT(T1,'C')
*       Define the upper end-fitting bottom surface.
                CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
```

```
     c          ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
                CURRENTSURFLABEL=0
                DO 5082 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 5082           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   UEFBOTTOMSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   UEFBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
*    Define the upper end-fitting top surface.
                CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c          ENDFITHEIGHT(DESNUM(COLUMN,ROW),1)+
     c          ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
                CURRENTSURFLABEL=0
                DO 5084 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 5084           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   UEFTOPSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   UEFTOPSURF=CURRENTSURFLABEL
                ENDIF
                APSRCLADTOPSURF=UEFTOPSURF
                APSRABSML=MN
*    Define the APSR absorber radius.
                CURRENTSURF=APSRADIM(BANKNUM(COLUMN,ROW),1)
                CURRENTSURFLABEL=0
                DO 5090 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 5090           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   APSRABSSURF=SN
```

```
               SURFTYPESPEC(SN)='CZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               APSRABSSURF=CURRENTSURFLABEL
            ENDIF
*     Define the APSR cladding inner radius.
            CURRENTSURF=APSRADIM(BANKNUM(COLUMN,ROW),2)
            CURRENTSURFLABEL=0
            DO 5100 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'CZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
5100        CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               APSRCLADIRSURF=SN
               SURFTYPESPEC(SN)='CZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               APSRCLADIRSURF=CURRENTSURFLABEL
            ENDIF
*     Define the APSR cladding outer radius.
            CURRENTSURF=APSRADIM(BANKNUM(COLUMN,ROW),3)
            CURRENTSURFLABEL=0
            DO 5110 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'CZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
5110        CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               APSRCLADORSURF=SN
               SURFTYPESPEC(SN)='CZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               APSRCLADORSURF=CURRENTSURFLABEL
            ENDIF
*     Define the APSR cladding bottom surface.
            CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c      APSRADIM(BANKNUM(COLUMN,ROW),6)-
     c      APSRADIM(BANKNUM(COLUMN,ROW),11)-
     c      APSRADIM(BANKNUM(COLUMN,ROW),9)
            IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
               CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
            ENDIF
            CURRENTSURFLABEL=0
            DO 5120 V=1,(SN-1)
```

```
            IF (SURFTYPESPEC(V).EQ.'PZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
5120       CONTINUE
           IF (CURRENTSURFLABEL.EQ.0) THEN
              APSRCLADBOTTOMSURF=SN
              SURFTYPESPEC(SN)='PZ'
              SURFVALUESPEC(SN)=CURRENTSURF
              SN=SN+1
           ELSE
              APSRCLADBOTTOMSURF=CURRENTSURFLABEL
           ENDIF
*     Define the APSR absorber bottom surface.
           CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c     APSRADIM(BANKNUM(COLUMN,ROW),6)
           IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
              CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
           ENDIF
           CURRENTSURFLABEL=0
           DO 5140 V=1,(SN-1)
              IF (SURFTYPESPEC(V).EQ.'PZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
5140       CONTINUE
           IF (CURRENTSURFLABEL.EQ.0) THEN
              APSRABSBOTTOMSURF=SN
              SURFTYPESPEC(SN)='PZ'
              SURFVALUESPEC(SN)=CURRENTSURF
              SN=SN+1
           ELSE
              APSRABSBOTTOMSURF=CURRENTSURFLABEL
           ENDIF
*     Define the APSR absorber top surface.
           CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c     APSRADIM(BANKNUM(COLUMN,ROW),6)+
     c     APSRADIM(BANKNUM(COLUMN,ROW),7)
           IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
              CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
           ENDIF
           CURRENTSURFLABEL=0
           DO 5150 V=1,(SN-1)
              IF (SURFTYPESPEC(V).EQ.'PZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
5150       CONTINUE
```

```
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   APSRABSTOPSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   APSRABSTOPSURF=CURRENTSURFLABEL
                ENDIF
*     Define the APSR upper plenum top surface..
                CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c          APSRADIM(BANKNUM(COLUMN,ROW),6)+
     c          APSRADIM(BANKNUM(COLUMN,ROW),7)+
     c          APSRADIM(BANKNUM(COLUMN,ROW),10)
                IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                   CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                ENDIF
                CURRENTSURFLABEL=0
                DO 5160 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 5160           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   APSRUPTOPSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   APSRUPTOPSURF=CURRENTSURFLABEL
                ENDIF
*     Define the APSR intermediate plug top surface.
                CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c          APSRADIM(BANKNUM(COLUMN,ROW),6)+
     c          APSRADIM(BANKNUM(COLUMN,ROW),7)+
     c          APSRADIM(BANKNUM(COLUMN,ROW),10)+
     c          APSRADIM(BANKNUM(COLUMN,ROW),8)
                IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                   CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                ENDIF
                CURRENTSURFLABEL=0
                DO 5170 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 5170           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   APSRIPLUGTOPSURF=SN
                   SURFTYPESPEC(SN)='PZ'
```

```
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  APSRIPLUGTOPSURF=CURRENTSURFLABEL
               ENDIF
            IF (SURFVALUESPEC(APSRABSBOTTOMSURF).LT.
     c      SURFVALUESPEC(UEFTOPSURF)) THEN
*  Check Axial Power Shaping Rod Absorber Material
               DO 5193 C=1,APSRABSMAT(BANKNUM(COLUMN,ROW),2)
                  IF (C.EQ.1) THEN
                     WRITE(200,5191) APSRABSML,
     c               APSRABSZAIDS(BANKNUM(COLUMN,ROW),C),
     c               (-1*APSRABSWTS(BANKNUM(COLUMN,ROW),C)),
     c               ASSYID(COLUMN,ROW)
 5191                FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c               '    $ Axial Power Shaping Rod Absorber',
     c               ' Material in Assembly ',A5)
                  ELSE
                     WRITE(200,5192)
     c               APSRABSZAIDS(BANKNUM(COLUMN,ROW),C),
     c               (-1*APSRABSWTS(BANKNUM(COLUMN,ROW),C))
 5192                FORMAT(T9,A9,3X,G14.6)
                  ENDIF
 5193          CONTINUE
               MN=MN+1
*     Write the APSR absorber cell in this APSR universe.
            WRITE(30,5200) LN, APSRABSML,
     c      (-1*APSRABSMAT(BANKNUM(COLUMN,ROW),1)), (-1*APSRABSSURF),
     c      (-1*APSRABSTOPSURF), APSRABSBOTTOMSURF,
     c      APSRAUNIV(COLUMN,ROW)
 5200       FORMAT(T1,I4,T6,I4,T11,F10.6,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,
     c      '    $ Axial power shaping rod absorber region')
            LN=LN+1
*     Write the absorber-to-cladding gap cell in this APSR universe.
            WRITE(30,5210) LN, (-1*APSRCLADIRSURF), APSRABSSURF,
     c      (-1*APSRABSTOPSURF),
     c      APSRABSBOTTOMSURF, APSRAUNIV(COLUMN,ROW)
 5210       FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'    $ Absorber-to-cladding gap')
            LN=LN+1
         ENDIF
*     Write the APSR cladding cell in this APSR universe.
*     Determine if the APSR cladding material specification has
*     previously been defined.  If it has been previously defined, determine
*     the cladding material specification label.
            CLADMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 5230 RO=1,(ROW-1)
                  DO 5220 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'APSRA') THEN
                        IF (APSRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
```

```
     c                      APSRCLADMAT(BANKNUM(CO,RO))) THEN
                               CLADMLUNIQUE=.FALSE.
                               LEAVE=.TRUE.
                               APSRCLADML(COLUMN,ROW)=APSRCLADML(CO,RO)
                               EXIT
                            ENDIF
                         ENDIF
                         ENDIF
    5220              CONTINUE
                      IF (LEAVE.EQ..TRUE.) THEN
                         EXIT
                      ENDIF
    5230           CONTINUE
                   IF (LEAVE.EQ..FALSE.) THEN
                      DO 5250 RO=ROW,ROW
                         DO 5240 CO=1,(COLUMN-1)
                         IF (BANKNUM(CO,RO).NE.0) THEN
                         IF (BANKDES(BANKNUM(CO,RO)).EQ.'APSRA') THEN
                            IF (APSRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                      APSRCLADMAT(BANKNUM(CO,RO))) THEN
                               CLADMLUNIQUE=.FALSE.
                               LEAVE=.TRUE.
                               APSRCLADML(COLUMN,ROW)=APSRCLADML(CO,RO)
                               EXIT
                            ENDIF
                         ENDIF
                         ENDIF
    5240                 CONTINUE
                         IF (LEAVE.EQ..TRUE.) THEN
                            EXIT
                         ENDIF
    5250              CONTINUE
                   ENDIF
                ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                   DO 5270 RO=1,(ROW-1)
                      DO 5260 CO=1,50
                         IF (BANKNUM(CO,RO).NE.0) THEN
                         IF (BANKDES(BANKNUM(CO,RO)).EQ.'APSRA') THEN
                            IF (APSRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                      APSRCLADMAT(BANKNUM(CO,RO))) THEN
                               CLADMLUNIQUE=.FALSE.
                               LEAVE=.TRUE.
                               APSRCLADML(COLUMN,ROW)=APSRCLADML(CO,RO)
                               EXIT
                            ENDIF
                         ENDIF
                         ENDIF
    5260              CONTINUE
                      IF (LEAVE.EQ..TRUE.) THEN
                         EXIT
                      ENDIF
    5270           CONTINUE
                ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                   DO 5290 RO=1,1
```

```
                DO 5280 CO=1,(COLUMN-1)
                  IF (BANKNUM(CO,RO).NE.0) THEN
                  IF (BANKDES(BANKNUM(CO,RO)).EQ.'APSRA') THEN
                    IF (APSRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c              APSRCLADMAT(BANKNUM(CO,RO))) THEN
                      CLADMLUNIQUE=.FALSE.
                      LEAVE=.TRUE.
                      APSRCLADML(COLUMN,ROW)=APSRCLADML(CO,RO)
                      EXIT
                    ENDIF
                  ENDIF
                  ENDIF
 5280           CONTINUE
                IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
                ENDIF
 5290         CONTINUE
            ENDIF
            IF (SURFVALUESPEC(APSRCLADBOTTOMSURF).LT.
     c      SURFVALUESPEC(UEFTOPSURF)) THEN
            IF (CLADMLUNIQUE.EQ..TRUE.) THEN
              APSRCLADML(COLUMN,ROW)=MN
* Check Axial Power Shaping Rod Cladding
                IF (APSRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
                  DO 5292 C=1,2
                    IF (C.EQ.1) THEN
                      WRITE(200,9300) APSRCLADML(COLUMN,ROW)
                    ELSEIF (C.EQ.2) THEN
                      WRITE(200,9301)
                      WRITE(200,7000)
                      WRITE(200,7001)
                      WRITE(200,7002)
                      WRITE(200,9302)
                      WRITE(200,7003)
                      WRITE(200,7004)
                      WRITE(200,7005)
                      WRITE(200,9303)
                      WRITE(200,9304)
                    ENDIF
 5292             CONTINUE
                ELSEIF (APSRCLADMAT(BANKNUM(COLUMN,ROW))
     c          .EQ.2) THEN
                  DO 5294 C=1,2
                    IF (C.EQ.1) THEN
                      WRITE(200,9305) APSRCLADML(COLUMN,ROW)
                    ELSEIF (C.EQ.2) THEN
                      WRITE(200,9306)
                      WRITE(200,9307)
                      WRITE(200,9308)
                      WRITE(200,9309)
                      WRITE(200,9310)
                      WRITE(200,7006)
                      WRITE(200,7007)
                      WRITE(200,7008)
```

```
                    WRITE(200,9311)
                    WRITE(200,9312)
                    WRITE(200,7009)
                    WRITE(200,7010)
                    WRITE(200,7011)
                    WRITE(200,9313)
                    WRITE(200,7012)
                    WRITE(200,7013)
                    WRITE(200,7014)
                    WRITE(200,7015)
                 ENDIF
5294          CONTINUE
           ELSEIF (APSRCLADMAT(BANKNUM(COLUMN,ROW))
     c     .EQ.3) THEN
              DO 5296 C=1,2
                 IF (C.EQ.1) THEN
                    WRITE(200,9314) APSRCLADML(COLUMN,ROW)
                 ELSEIF (C.EQ.2) THEN
                    WRITE(200,9315)
                    WRITE(200,9316)
                    WRITE(200,9317)
                    WRITE(200,9318)
                    WRITE(200,7016)
                    WRITE(200,7017)
                    WRITE(200,7018)
                    WRITE(200,9319)
                    WRITE(200,9320)
                    WRITE(200,7019)
                    WRITE(200,7020)
                    WRITE(200,7021)
                    WRITE(200,9321)
                    WRITE(200,7022)
                    WRITE(200,7023)
                    WRITE(200,7024)
                    WRITE(200,7025)
                    WRITE(200,9322)
                    WRITE(200,9323)
                    WRITE(200,9324)
                    WRITE(200,9325)
                    WRITE(200,9326)
                    WRITE(200,9327)
                    WRITE(200,7026)
                    WRITE(200,9328)
                    WRITE(200,9329)
                    WRITE(200,9330)
                 ENDIF
5296          CONTINUE
           ENDIF
        MN=MN+1
     ENDIF
     IF (APSRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
        CLADRHO=6.56
     ELSEIF (APSRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.2) THEN
        CLADRHO=7.90
```

```
          ELSEIF (APSRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.3) THEN
             CLADRHO=8.19
          ENDIF
          WRITE(30,5308) LN, APSRCLADML(COLUMN,ROW), (-1*CLADRHO),
     c    APSRCLADIRSURF,
     c    (-1*APSRCLADORSURF), (-1*APSRCLADTOPSURF),
     c    APSRCLADBOTTOMSURF, APSRAUNIV(COLUMN,ROW)
 5308     FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c    ' IMP:N=1 U=',I3,
     c    '   $ Axial power shaping rod cladding')
          LN=LN+1
          ENDIF
*     Write the APSR upper plenum cell in this APSR universe.
*     Determine if the APSR upper plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the upper plenum material specification label.
          APSRUPMLUNIQUE=.TRUE.
          LEAVE=.FALSE.
          IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
             DO 5320 RO=1,(ROW-1)
                DO 5310 CO=1,50
                   IF (BANKNUM(CO,RO).NE.0) THEN
                      IF (BANKNUM(COLUMN,ROW).EQ.
     c                   BANKNUM(CO,RO)) THEN
                         APSRUPMLUNIQUE=.FALSE.
                         LEAVE=.TRUE.
                         APSRUPML(COLUMN,ROW)=APSRUPML(CO,RO)
                         EXIT
                      ENDIF
                   ENDIF
 5310           CONTINUE
                IF (LEAVE.EQ..TRUE.) THEN
                   EXIT
                ENDIF
 5320        CONTINUE
             IF (LEAVE.EQ..FALSE.) THEN
                DO 5340 RO=ROW,ROW
                   DO 5330 CO=1,(COLUMN-1)
                      IF (BANKNUM(CO,RO).NE.0) THEN
                         IF (BANKNUM(COLUMN,ROW).EQ.
     c                   BANKNUM(CO,RO)) THEN
                            APSRUPMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            APSRUPML(COLUMN,ROW)=APSRUPML(CO,RO)
                            EXIT
                         ENDIF
                      ENDIF
 5330              CONTINUE
                   IF (LEAVE.EQ..TRUE.) THEN
                      EXIT
                   ENDIF
 5340           CONTINUE
             ENDIF
          ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
```

```
                DO 5360 RO=1,(ROW-1)
                   DO 5350 CO=1,50
                      IF (BANKNUM(CO,RO).NE.0) THEN
                         IF (BANKNUM(COLUMN,ROW).EQ.
     c                      BANKNUM(CO,RO)) THEN
                            APSRUPMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            APSRUPML(COLUMN,ROW)=APSRUPML(CO,RO)
                            EXIT
                         ENDIF
                      ENDIF
 5350              CONTINUE
                   IF (LEAVE.EQ..TRUE.) THEN
                      EXIT
                   ENDIF
 5360           CONTINUE
             ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                DO 5380 RO=1,1
                   DO 5370 CO=1,(COLUMN-1)
                      IF (BANKNUM(CO,RO).NE.0) THEN
                         IF (BANKNUM(COLUMN,ROW).EQ.
     c                      BANKNUM(CO,RO)) THEN
                            APSRUPMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            APSRUPML(COLUMN,ROW)=APSRUPML(CO,RO)
                            EXIT
                         ENDIF
                      ENDIF
 5370              CONTINUE
                   IF (LEAVE.EQ..TRUE.) THEN
                      EXIT
                   ENDIF
 5380           CONTINUE
             ENDIF
          IF (SURFVALUESPEC(APSRABSTOPSURF).LT.
     c    SURFVALUESPEC(UEFTOPSURF)) THEN
             IF (APSRUPMLUNIQUE.EQ..TRUE.) THEN
                APSRUPML(COLUMN,ROW)=MN
*  Check Axial Power Shaping Rod Upper Plenum Regions
                   DO 5383 C=1,APSRUPLENMAT(BANKNUM(COLUMN,ROW),2)
                      IF (C.EQ.1) THEN
                         WRITE(200,5381) APSRUPML(COLUMN,ROW),
     c                   APSRUPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                   (-1*APSRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 5381                 FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                   '    $ Axial Power Shaping Rod Upper Plenum')
                      ELSE
                         WRITE(200,5382)
     c                   APSRUPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                   (-1*APSRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 5382                    FORMAT(T9,A9,3X,G14.6)
                      ENDIF
 5383              CONTINUE
                MN=MN+1
```

```
          ENDIF
          WRITE(30,5390) LN, APSRUPML(COLUMN,ROW),
     c    (-1*APSRUPLENMAT(BANKNUM(COLUMN,ROW),1)),
     c    APSRABSTOPSURF,
     c    (-1*APSRUPTOPSURF), (-1*APSRCLADIRSURF),
     c    APSRAUNIV(COLUMN,ROW)
 5390     FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c    ' IMP:N=1 U=',I3,
     c    '    $ Axial power shaping rod upper plenum')
          LN=LN+1
        ENDIF
*     Write the APSR lower plenum cell in this APSR universe.
*     Determine if the APSR lower plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the lower plenum material specification label.
        IF (APSRADIM(BANKNUM(COLUMN,ROW),11).GT.(0.0)) THEN
          APSRLPMLUNIQUE=.TRUE.
          LEAVE=.FALSE.
          IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
            DO 5410 RO=1,(ROW-1)
              DO 5400 CO=1,50
                IF (BANKNUM(CO,RO).NE.0) THEN
                  IF (BANKNUM(COLUMN,ROW).EQ.
     c            BANKNUM(CO,RO)) THEN
                    APSRLPMLUNIQUE=.FALSE.
                    LEAVE=.TRUE.
                    APSRLPML(COLUMN,ROW)=APSRLPML(CO,RO)
                    EXIT
                  ENDIF
                ENDIF
 5400         CONTINUE
              IF (LEAVE.EQ..TRUE.) THEN
                EXIT
              ENDIF
 5410       CONTINUE
            IF (LEAVE.EQ..FALSE.) THEN
              DO 5430 RO=ROW,ROW
                DO 5420 CO=1,(COLUMN-1)
                  IF (BANKNUM(CO,RO).NE.0) THEN
                    IF (BANKNUM(COLUMN,ROW).EQ.
     c              BANKNUM(CO,RO)) THEN
                      APSRLPMLUNIQUE=.FALSE.
                      LEAVE=.TRUE.
                      APSRLPML(COLUMN,ROW)=APSRLPML(CO,RO)
                      EXIT
                    ENDIF
                  ENDIF
 5420           CONTINUE
                IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
                ENDIF
 5430         CONTINUE
            ENDIF
          ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
```

```
                DO 5450 RO=1,(ROW-1)
                   DO 5440 CO=1,50
                      IF (BANKNUM(CO,RO).NE.0) THEN
                         IF (BANKNUM(COLUMN,ROW).EQ.
     c                      BANKNUM(CO,RO)) THEN
                            APSRLPMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            APSRLPML(COLUMN,ROW)=APSRLPML(CO,RO)
                            EXIT
                         ENDIF
                      ENDIF
 5440              CONTINUE
                   IF (LEAVE.EQ..TRUE.) THEN
                      EXIT
                   ENDIF
 5450           CONTINUE
             ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                DO 5470 RO=1,1
                   DO 5460 CO=1,(COLUMN-1)
                      IF (BANKNUM(CO,RO).NE.0) THEN
                         IF (BANKNUM(COLUMN,ROW).EQ.
     c                      BANKNUM(CO,RO)) THEN
                            APSRLPMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            APSRLPML(COLUMN,ROW)=APSRLPML(CO,RO)
                            EXIT
                         ENDIF
                      ENDIF
 5460              CONTINUE
                   IF (LEAVE.EQ..TRUE.) THEN
                      EXIT
                   ENDIF
 5470           CONTINUE
             ENDIF
          IF (SURFVALUESPEC(APSRCLADBOTTOMSURF).LT.
     c    SURFVALUESPEC(UEFTOPSURF)) THEN
             IF (APSRLPMLUNIQUE.EQ..TRUE.) THEN
                APSRLPML(COLUMN,ROW)=MN
*  Check Axial Power Shaping Rod Lower Plenum Regions
                DO 5473 C=1,APSRLPLENMAT(BANKNUM(COLUMN,ROW),2)
                   IF (C.EQ.1) THEN
                      WRITE(200,5471) APSRLPML(COLUMN,ROW),
     c                APSRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                (-1*APSRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 5471                 FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                '    $ Axial Power Shaping Rod Lower Plenum')
                   ELSE
                      WRITE(200,5472)
     c                APSRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                (-1*APSRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 5472                 FORMAT(T9,A9,3X,G14.6)
                   ENDIF
 5473           CONTINUE
                MN=MN+1
```

```
            ENDIF
            WRITE(30,5480) LN, APSRLPML(COLUMN,ROW),
     c      (-1*APSRLPLENMAT(BANKNUM(COLUMN,ROW),1)),
     c      APSRCLADBOTTOMSURF,
     c      (-1*APSRABSBOTTOMSURF), (-1*APSRCLADIRSURF),
     c      APSRAUNIV(COLUMN,ROW)
 5480       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,
     c      '    $ Axial power shaping rod lower plenum')
            LN=LN+1
         ENDIF
         ENDIF
         IF (SURFVALUESPEC(APSRUPTOPSURF).LT.
     c   SURFVALUESPEC(UEFTOPSURF)) THEN
*     Calculate the APSR intermediate plug cell density.
            APSRIPLUGML(COLUMN,ROW)=MN
            IF (APSRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
               APSRIPLUGDEN(BANKNUM(COLUMN,ROW))=
     c         ((APSRIPLUGFRAC(BANKNUM(COLUMN,ROW))*6.56)+
     c         ((1.0-APSRIPLUGFRAC(BANKNUM(COLUMN,ROW)))*
     c         MODDENSITY))
               SPACMASS=APSRIPLUGFRAC(BANKNUM(COLUMN,ROW))*6.56
               MODMASS=(1.0-APSRIPLUGFRAC(BANKNUM(COLUMN,ROW)))*
     c         MODDENSITY
               SPACFRAC=SPACMASS/(SPACMASS+MODMASS)
               MODFRAC=MODMASS/(SPACMASS+MODMASS)
         BWTINH2O=((PPMB*1E-6)/(1.0+(PPMB*1E-6)))*100.0
         HWTINH2O=((1.008664904*0.999167*2.0)/
     c   ((1.008664904*0.999167*2.0)+
     c   (1.008664904*15.857510)))*(100.0-BWTINH2O)
         OWTINH2O=((1.008664904*15.857510)/
     c   ((1.008664904*0.999167*2.0)+
     c   (1.008664904*15.857510)))*(100.0-BWTINH2O)
         B10WTINH2O=((1.008664904*9.926922*0.194)/
     c   ((1.008664904*9.926922*0.194)+
     c   (1.008664904*10.914730*0.806)))*BWTINH2O
         B11WTINH2O=((1.008664904*10.914730*0.806)/
     c   ((1.008664904*9.926922*0.194)+
     c   (1.008664904*10.914730*0.806)))*BWTINH2O
               OWT=(OWTINH2O*MODFRAC)+(0.120*SPACFRAC)
               HWT=HWTINH2O*MODFRAC
               B10WT=(B10WTINH2O*MODFRAC)
               B11WT=(B11WTINH2O*MODFRAC)
               CRWT=0.100*SPACFRAC
               CR50=0.04173684
               CR52=0.837
               CR53=0.09673684
               CR54=0.02452632
               FEWT=0.200*SPACFRAC
               FE54=0.05699324
               FE56=0.91868499
               FE57=0.02141247
               FE58=0.00290930
               ZRWT=98.180*SPACFRAC
```

```
                SNWT=1.400*SPACFRAC
                WRITE(200,5491) APSRIPLUGML(COLUMN,ROW),
     c          (-1*HWT), ASSYID(COLUMN,ROW)
 5491           FORMAT(T1,'M',I4,T9,' 1001.50c'3X,G14.8,
     c          '    $ APSR Intermediate Zirc-4 Plug in Assembly ',A5)
                WRITE(200,5492) -1*OWT
 5492           FORMAT(T9,' 8016.50c',3X,G14.8)
                WRITE(200,5493) -1*B10WT
 5493           FORMAT(T9,' 5010.50c',3X,G14.8)
                WRITE(200,5494) -1*B11WT
 5494           FORMAT(T9,' 5011.56c',3X,G14.8)
                WRITE(200,5495) -1*CRWT*CR50
 5495           FORMAT(T9,'24050.60c',3X,G14.8)
                WRITE(200,7077) -1*CRWT*CR52
 7077           FORMAT(T9,'24052.60c',3X,G14.8)
                WRITE(200,7078) -1*CRWT*CR53
 7078           FORMAT(T9,'24053.60c',3X,G14.8)
                WRITE(200,7079) -1*CRWT*CR54
 7079           FORMAT(T9,'24054.60c',3X,G14.8)
                WRITE(200,5499) -1*FEWT*FE54
 5499           FORMAT(T9,'26054.60c',3X,G14.8)
                WRITE(200,7080) -1*FEWT*FE56
 7080           FORMAT(T9,'26056.60c',3X,G14.8)
                WRITE(200,7081) -1*FEWT*FE57
 7081           FORMAT(T9,'26057.60c',3X,G14.8)
                WRITE(200,7082) -1*FEWT*FE58
 7082           FORMAT(T9,'26058.60c',3X,G14.8)
                WRITE(200,5503) -1*ZRWT
 5503           FORMAT(T9,'40093.50c',3X,G14.8)
                WRITE(200,5504) -1*SNWT
 5504           FORMAT(T9,'50000.35c',3X,G14.8)
             ELSEIF (APSRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.2) THEN
                APSRIPLUGDEN(BANKNUM(COLUMN,ROW))=
     c          ((APSRIPLUGFRAC(BANKNUM(COLUMN,ROW))*7.9)+
     c          ((1.0-APSRIPLUGFRAC(BANKNUM(COLUMN,ROW)))*
     c          MODDENSITY))
                SPACMASS=APSRIPLUGFRAC(BANKNUM(COLUMN,ROW))*7.9
                MODMASS=(1.0-APSRIPLUGFRAC(BANKNUM(COLUMN,ROW)))*
     c          MODDENSITY
                SPACFRAC=SPACMASS/(SPACMASS+MODMASS)
                MODFRAC=MODMASS/(SPACMASS+MODMASS)
          BWTINH2O=((PPMB*1E-6)/(1.0+(PPMB*1E-6)))*100.0
          HWTINH2O=((1.008664904*0.999167*2.0)/
     c    ((1.008664904*0.999167*2.0)+
     c    (1.008664904*15.857510)))*(100.0-BWTINH2O)
          OWTINH2O=((1.008664904*15.857510)/
     c    ((1.008664904*0.999167*2.0)+
     c    (1.008664904*15.857510)))*(100.0-BWTINH2O)
          B10WTINH2O=((1.008664904*9.926922*0.194)/
     c    ((1.008664904*9.926922*0.194)+
     c    (1.008664904*10.914730*0.806)))*BWTINH2O
          B11WTINH2O=((1.008664904*10.914730*0.806)/
     c    ((1.008664904*9.926922*0.194)+
     c    (1.008664904*10.914730*0.806)))*BWTINH2O
```

```
                OWT=OWTINH2O*MODFRAC
                HWT=HWTINH2O*MODFRAC
                B10WT=B10WTINH2O*MODFRAC
                B11WT=B11WTINH2O*MODFRAC
                CWT=0.08*SPACFRAC
                NWT=0.100*SPACFRAC
                SIWT=0.75*SPACFRAC
                PWT=0.045*SPACFRAC
                SWT=0.030*SPACFRAC
                CRWT=19.000*SPACFRAC
                CR50=0.04173684
                CR52=0.837
                CR53=0.09673684
                CR54=0.02452632
                MNWT=2.000*SPACFRAC
                FEWT=68.745*SPACFRAC
                FE54=0.05699324
                FE56=0.91868499
                FE57=0.02141247
                FE58=0.00290930
                NIWT=9.250*SPACFRAC
                NI58=0.67394595
                NI60=0.26648649
                NI61=0.01178378
                NI62=0.03783784
                NI64=0.00994594
                WRITE(200,5505) APSRIPLUGML(COLUMN,ROW),
     c          (-1*HWT), ASSYID(COLUMN,ROW)
5505            FORMAT(T1,'M',I4,T9,' 1001.50c'3X,G14.8,
     c          '   $ APSR Intermediate SS304 Plug in Assembly ',A5)
                WRITE(200,5506) -1*OWT
5506            FORMAT(T9,' 8016.50c',3X,G14.8)
                WRITE(200,5507) -1*B10WT
5507            FORMAT(T9,' 5010.50c',3X,G14.8)
                WRITE(200,5508) -1*B11WT
5508            FORMAT(T9,' 5011.56c',3X,G14.8)
                WRITE(200,5509) -1*CWT
5509            FORMAT(T9,'6000.50c',3X,G14.8)
                WRITE(200,5510) -1*NWT
5510            FORMAT(T9,'7014.50c',3X,G14.8)
                WRITE(200,5511) -1*SIWT
5511            FORMAT(T9,'14000.50c',3X,G14.8)
                WRITE(200,5512) -1*PWT
5512            FORMAT(T9,'15031.50c',3X,G14.8)
                WRITE(200,5513) -1*SWT
5513            FORMAT(T9,'16032.50c',3X,G14.8)
                WRITE(200,5514) -1*CRWT*CR50
5514            FORMAT(T9,'24050.60c',3X,G14.8)
                WRITE(200,7084) -1*CRWT*CR52
7084            FORMAT(T9,'24052.60c',3X,G14.8)
                WRITE(200,7085) -1*CRWT*CR53
7085            FORMAT(T9,'24053.60c',3X,G14.8)
                WRITE(200,7086) -1*CRWT*CR54
7086            FORMAT(T9,'24054.60c',3X,G14.8)
```

```
            WRITE(200,5518) -1*MNWT
 5518       FORMAT(T9,'25055.50c',3X,G14.8)
            WRITE(200,5519) -1*FEWT*FE54
 5519       FORMAT(T9,'26054.60c',3X,G14.8)
            WRITE(200,7087) -1*FEWT*FE56
 7087       FORMAT(T9,'26056.60c',3X,G14.8)
            WRITE(200,7088) -1*FEWT*FE57
 7088       FORMAT(T9,'26057.60c',3X,G14.8)
            WRITE(200,7089) -1*FEWT*FE58
 7089       FORMAT(T9,'26058.60c',3X,G14.8)
            WRITE(200,5523) -1*NIWT*NI58
 5523       FORMAT(T9,'28058.60c',3X,G14.8)
            WRITE(200,7090) -1*NIWT*NI60
 7090       FORMAT(T9,'28060.60c',3X,G14.8)
            WRITE(200,7091) -1*NIWT*NI61
 7091       FORMAT(T9,'28061.60c',3X,G14.8)
            WRITE(200,7092) -1*NIWT*NI62
 7092       FORMAT(T9,'28062.60c',3X,G14.8)
            WRITE(200,7093) -1*NIWT*NI64
 7093       FORMAT(T9,'28064.60c',3X,G14.8)
         ELSEIF (APSRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.3) THEN
            APSRIPLUGDEN(BANKNUM(COLUMN,ROW))=
     c      ((APSRIPLUGFRAC(BANKNUM(COLUMN,ROW))*8.19)+
     c      ((1.0-APSRIPLUGFRAC(BANKNUM(COLUMN,ROW)))*
     c      MODDENSITY))
            SPACMASS=APSRIPLUGFRAC(BANKNUM(COLUMN,ROW))*8.19
            MODMASS=(1.0-APSRIPLUGFRAC(BANKNUM(COLUMN,ROW)))*
     c      MODDENSITY
            SPACFRAC=SPACMASS/(SPACMASS+MODMASS)
            MODFRAC=MODMASS/(SPACMASS+MODMASS)
      BWTINH2O=((PPMB*1E-6)/(1.0+(PPMB*1E-6)))*100.0
      HWTINH2O=((1.008664904*0.999167*2.0)/
     c ((1.008664904*0.999167*2.0)+
     c (1.008664904*15.857510)))*(100.0-BWTINH2O)
      OWTINH2O=((1.008664904*15.857510)/
     c ((1.008664904*0.999167*2.0)+
     c (1.008664904*15.857510)))*(100.0-BWTINH2O)
      B10WTINH2O=((1.008664904*9.926922*0.194)/
     c ((1.008664904*9.926922*0.194)+
     c (1.008664904*10.914730*0.806)))*BWTINH2O
      B11WTINH2O=((1.008664904*10.914730*0.806)/
     c ((1.008664904*9.926922*0.194)+
     c (1.008664904*10.914730*0.806)))*BWTINH2O
            B10WTINB=((1.008664904*9.926922*0.194)/
     c      ((1.008664904*9.926922*0.194)+
     c      (1.008664904*10.914730*0.806)))*100.0
            B11WTINB=((1.008664904*10.914730*0.806)/
     c      ((1.008664904*9.926922*0.194)+
     c      (1.008664904*10.914730*0.806)))*100.0
            OWT=OWTINH2O*MODFRAC
            HWT=HWTINH2O*MODFRAC
            B10WT=(B10WTINH2O*MODFRAC)+(B10WTINB*0.006*SPACFRAC)
            B11WT=(B11WTINH2O*MODFRAC)+(B11WTINB*0.006*SPACFRAC)
            CWT=0.080*SPACFRAC
```

```
            SIWT=0.350*SPACFRAC
            PWT=0.015*SPACFRAC
            SWT=0.015*SPACFRAC
            CRWT=19.000*SPACFRAC
            CR50=0.04173684
            CR52=0.837
            CR53=0.09673684
            CR54=0.02452632
            FE54=0.05699324
            FE56=0.91868499
            FE57=0.02141247
            FE58=0.00290930
            NI58=0.67394595
            NI60=0.26648649
            NI61=0.01178378
            NI62=0.03783784
            NI64=0.00994594
            CU63=0.683
            CU65=0.317
            MNWT=0.350*SPACFRAC
            FEWT=16.809*SPACFRAC
            NIWT=52.500*SPACFRAC
            ALWT=0.500*SPACFRAC
            TIWT=0.900*SPACFRAC
            COWT=1.000*SPACFRAC
            CUWT=0.300*SPACFRAC
            NBWT=2.563*SPACFRAC
            MOWT=3.050*SPACFRAC
            TAWT=2.563*SPACFRAC
            WRITE(200,5528) APSRIPLUGML(COLUMN,ROW),
     c      (-1*HWT), ASSYID(COLUMN,ROW)
5528        FORMAT(T1,'M',I4,T9,' 1001.50c'3X,G14.8,
     c      '   $ APSR Intermediate Inconel Plug in Assembly ',A5)
            WRITE(200,5529) -1*OWT
5529        FORMAT(T9,' 8016.50c',3X,G14.8)
            WRITE(200,5530) -1*B10WT
5530        FORMAT(T9,' 5010.50c',3X,G14.8)
            WRITE(200,5531) -1*B11WT
5531        FORMAT(T9,' 5011.56c',3X,G14.8)
            WRITE(200,5532) -1*CWT
5532        FORMAT(T9,'6000.50c',3X,G14.8)
            WRITE(200,5533) -1*SIWT
5533        FORMAT(T9,'14000.50c',3X,G14.8)
            WRITE(200,5534) -1*PWT
5534        FORMAT(T9,'15031.50c',3X,G14.8)
            WRITE(200,5535) -1*SWT
5535        FORMAT(T9,'16032.50c',3X,G14.8)
            WRITE(200,5536) -1*CRWT*CR50
5536        FORMAT(T9,'24050.60c',3X,G14.8)
            WRITE(200,7094) -1*CRWT*CR52
7094        FORMAT(T9,'24052.60c',3X,G14.8)
            WRITE(200,7095) -1*CRWT*CR53
7095        FORMAT(T9,'24053.60c',3X,G14.8)
            WRITE(200,7096) -1*CRWT*CR54
```

```
 7096         FORMAT(T9,'24054.60c',3X,G14.8)
              WRITE(200,5540) -1*MNWT
 5540         FORMAT(T9,'25055.50c',3X,G14.8)
              WRITE(200,5541) -1*FEWT*FE54
 5541         FORMAT(T9,'26054.60c',3X,G14.8)
              WRITE(200,7097) -1*FEWT*FE56
 7097         FORMAT(T9,'26056.60c',3X,G14.8)
              WRITE(200,7098) -1*FEWT*FE57
 7098         FORMAT(T9,'26057.60c',3X,G14.8)
              WRITE(200,7099) -1*FEWT*FE58
 7099         FORMAT(T9,'26058.60c',3X,G14.8)
              WRITE(200,5545) -1*NIWT*NI58
 5545         FORMAT(T9,'28058.60c',3X,G14.8)
              WRITE(200,7100) -1*NIWT*NI60
 7100         FORMAT(T9,'28060.60c',3X,G14.8)
              WRITE(200,7101) -1*NIWT*NI61
 7101         FORMAT(T9,'28061.60c',3X,G14.8)
              WRITE(200,7102) -1*NIWT*NI62
 7102         FORMAT(T9,'28062.60c',3X,G14.8)
              WRITE(200,7103) -1*NIWT*NI64
 7103         FORMAT(T9,'28064.60c',3X,G14.8)
              WRITE(200,5550) -1*ALWT
 5550         FORMAT(T9,'13027.50c',3X,G14.8)
              WRITE(200,5551) -1*TIWT
 5551         FORMAT(T9,'22000.50c',3X,G14.8)
              WRITE(200,5552) -1*COWT
 5552         FORMAT(T9,'27059.50c',3X,G14.8)
              WRITE(200,5553) -1*CUWT*CU63
 5553         FORMAT(T9,'29063.60c',3X,G14.8)
              WRITE(200,7104) -1*CUWT*CU65
 7104         FORMAT(T9,'29065.60c',3X,G14.8)
              WRITE(200,5555) -1*NBWT
 5555         FORMAT(T9,'41093.50c',3X,G14.8)
              WRITE(200,5556) -1*MOWT
 5556         FORMAT(T9,'42000.50c',3X,G14.8)
              WRITE(200,5557) -1*TAWT
 5557         FORMAT(T9,'73181.50c',3X,G14.8)
            ENDIF
            WRITE(200,5558) APSRIPLUGML(COLUMN,ROW)
 5558       FORMAT(T1,'MT',I4,T9,'LWTR.03T')
            MN=MN+1
*      Write the APSR intermediate plug cell in this APSR universe.
            WRITE(30,5559) LN, APSRIPLUGML(COLUMN,ROW),
     c      (-1*APSRIPLUGDEN(BANKNUM(COLUMN,ROW))),
     c      APSRUPTOPSURF,
     c      (-1*APSRIPLUGTOPSURF), (-1*APSRCLADIRSURF),
     c      APSRAUNIV(COLUMN,ROW)
 5559       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,
     c      '    $ Axial power shaping rod intermediate plug region')
            LN=LN+1
          ENDIF
          IF (SURFVALUESPEC(APSRIPLUGTOPSURF).LT.
     c    SURFVALUESPEC(UEFTOPSURF)) THEN
```

```
*     Write the water above the intermediate plug cell in this APSR universe.
            WRITE(30,5560) LN, BMODML,
     c      (-1*MODDENSITY), APSRIPLUGTOPSURF,
     c      (-1*APSRCLADTOPSURF), (-1*APSRCLADIRSURF),
     c      APSRAUNIV(COLUMN,ROW)
 5560       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'    $ Borated moderator inside APSR')
            LN=LN+1
         ENDIF
*     Define the GT top surface.
               CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),3)
               IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                  CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
               ENDIF
               CURRENTSURFLABEL=0
               DO 5561 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 5561          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  GTTOPSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTTOPSURF=CURRENTSURFLABEL
               ENDIF
*     Define the GT bottom surface.
               CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),4)
               CURRENTSURFLABEL=0
               DO 5562 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 5562          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  GTBOTSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTBOTSURF=CURRENTSURFLABEL
               ENDIF
*     Define the GT outer radius surface.
               CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 5563 V=1,(SN-1)
```

```
                IF (SURFTYPESPEC(V).EQ.'CZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
5563         CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                GTORSURF=SN
                SURFTYPESPEC(SN)='CZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                GTORSURF=CURRENTSURFLABEL
             ENDIF
*     Define the GT inner radius surface.
             CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),1)
             CURRENTSURFLABEL=0
             DO 5564 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'CZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
5564         CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                GTIRSURF=SN
                SURFTYPESPEC(SN)='CZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                GTIRSURF=CURRENTSURFLABEL
             ENDIF
*     Define the lower end-fitting top surface.
             CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
             CURRENTSURFLABEL=0
             DO 5565 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
5565         CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                APSRLEFTOPSURF=SN
                SURFTYPESPEC(SN)='PZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                APSRLEFTOPSURF=CURRENTSURFLABEL
             ENDIF
*     Write the lower end-fitting cell specification for this APSR universe.
```

```
          IF (SURFVALUESPEC(GTBOTSURF).GE.
     c    ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)) THEN
             WRITE(30,5650) LN, FRLEFML(COLUMN,ROW),
     c       (-1*LEFMAT(DESNUM(COLUMN,ROW),1)),
     c       (-1*APSRLEFTOPSURF), APSRAUNIV(COLUMN,ROW)
 5650        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,' IMP:N=1 U=',I3,
     c       '   $ Assembly lower end-fitting')
             LN=LN+1
          ELSE
             WRITE(30,5660) LN, FRLEFML(COLUMN,ROW),
     c       (-1*LEFMAT(DESNUM(COLUMN,ROW),1)),
     c       (-1*APSRLEFTOPSURF), GTORSURF, APSRAUNIV(COLUMN,ROW)
 5660        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'   $ Assembly lower end-fitting')
             LN=LN+1
             WRITE(30,5670) LN, FRLEFML(COLUMN,ROW),
     c       (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*GTBOTSURF),
     c       (-1*GTORSURF), APSRAUNIV(COLUMN,ROW)
 5670        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'   $ Assembly lower end-fitting')
             LN=LN+1
          ENDIF
*     Write the upper end-fitting cell specification for this APSR universe.
          IF (SURFVALUESPEC(GTTOPSURF).LE.
     c    SURFVALUESPEC(UEFBOTTOMSURF)) THEN
             WRITE(30,5760) LN, FRUEFML(COLUMN,ROW),
     c       (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c       (-1*UEFTOPSURF), APSRCLADORSURF, APSRAUNIV(COLUMN,ROW)
 5760        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
             LN=LN+1
          ELSEIF ((SURFVALUESPEC(GTTOPSURF).GT.
     c    SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c    (SURFVALUESPEC(GTTOPSURF).LT.
     c    SURFVALUESPEC(UEFTOPSURF))) THEN
             WRITE(30,5770) LN, FRUEFML(COLUMN,ROW),
     c       (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c       (-1*UEFTOPSURF), GTORSURF, APSRAUNIV(COLUMN,ROW)
 5770        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
             LN=LN+1
             WRITE(30,5780) LN, FRUEFML(COLUMN,ROW),
     c       (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c       (-1*UEFTOPSURF), (-1*GTORSURF),
     c       APSRCLADORSURF, APSRAUNIV(COLUMN,ROW)
 5780        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       1X,I4,' IMP:N=1 U=',I3,
     c       '   $ Assembly upper end-fitting')
             LN=LN+1
          ELSEIF (SURFVALUESPEC(GTTOPSURF).GE.
     c    SURFVALUESPEC(UEFTOPSURF)) THEN
             WRITE(30,5790) LN, FRUEFML(COLUMN,ROW),
     c       (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c       (-1*UEFTOPSURF), GTORSURF, APSRAUNIV(COLUMN,ROW)
```

```
 5790         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
              LN=LN+1
           ENDIF
*      Write the GT material cell in this APSR universe.
*      Determine if the GT material specification has
*      previously been defined.  If it has been previously defined, determine
*      the material specification label.
           CLADMLUNIQUE=.TRUE.
           LEAVE=.FALSE.
           IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
              DO 5810 RO=1,(ROW-1)
                 DO 5800 CO=1,50
                    IF (DESNUM(CO,RO).NE.0) THEN
                       IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                 GTMAT(DESNUM(CO,RO))) THEN
                          CLADMLUNIQUE=.FALSE.
                          LEAVE=.TRUE.
                          GTML(COLUMN,ROW)=GTML(CO,RO)
                          EXIT
                       ENDIF
                    ENDIF
 5800            CONTINUE
                 IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                 ENDIF
 5810         CONTINUE
              IF (LEAVE.EQ..FALSE.) THEN
                 DO 5830 RO=ROW,ROW
                    DO 5820 CO=1,(COLUMN-1)
                    IF (DESNUM(CO,RO).NE.0) THEN
                       IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                 GTMAT(DESNUM(CO,RO))) THEN
                          CLADMLUNIQUE=.FALSE.
                          LEAVE=.TRUE.
                          GTML(COLUMN,ROW)=GTML(CO,RO)
                          EXIT
                       ENDIF
                    ENDIF
 5820               CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
 5830            CONTINUE
              ENDIF
           ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
              DO 5850 RO=1,(ROW-1)
                 DO 5840 CO=1,50
                    IF (DESNUM(CO,RO).NE.0) THEN
                       IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                 GTMAT(DESNUM(CO,RO))) THEN
                          CLADMLUNIQUE=.FALSE.
                          LEAVE=.TRUE.
                          GTML(COLUMN,ROW)=GTML(CO,RO)
```

```
                        EXIT
                     ENDIF
                  ENDIF
 5840          CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
 5850       CONTINUE
         ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
            DO 5870 RO=1,1
               DO 5860 CO=1,(COLUMN-1)
                  IF (DESNUM(CO,RO).NE.0) THEN
                     IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c               GTMAT(DESNUM(CO,RO))) THEN
                        CLADMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        GTML(COLUMN,ROW)=GTML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
 5860          CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
 5870       CONTINUE
         ENDIF
         IF (CLADMLUNIQUE.EQ..TRUE.) THEN
            GTML(COLUMN,ROW)=MN
*  Check Guide Tube Material
               IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.1) THEN
                  DO 5872 C=1,2
                     IF (C.EQ.1) THEN
                        WRITE(200,9300) GTML(COLUMN,ROW)
                     ELSEIF (C.EQ.2) THEN
                        WRITE(200,9301)
                        WRITE(200,7000)
                        WRITE(200,7001)
                        WRITE(200,7002)
                        WRITE(200,9302)
                        WRITE(200,7003)
                        WRITE(200,7004)
                        WRITE(200,7005)
                        WRITE(200,9303)
                        WRITE(200,9304)
                     ENDIF
 5872             CONTINUE
               ELSEIF (GTMAT(DESNUM(COLUMN,ROW))
     c         .EQ.2) THEN
                  DO 5874 C=1,2
                     IF (C.EQ.1) THEN
                        WRITE(200,9305) GTML(COLUMN,ROW)
                     ELSEIF (C.EQ.2) THEN
                        WRITE(200,9306)
                        WRITE(200,9307)
```

```
                         WRITE(200,9308)
                         WRITE(200,9309)
                         WRITE(200,9310)
                         WRITE(200,7006)
                         WRITE(200,7007)
                         WRITE(200,7008)
                         WRITE(200,9311)
                         WRITE(200,9312)
                         WRITE(200,7009)
                         WRITE(200,7010)
                         WRITE(200,7011)
                         WRITE(200,9313)
                         WRITE(200,7012)
                         WRITE(200,7013)
                         WRITE(200,7014)
                         WRITE(200,7015)
                      ENDIF
5874               CONTINUE
                ELSEIF (GTMAT(DESNUM(COLUMN,ROW))
     c          .EQ.3) THEN
                   DO 5876 C=1,2
                      IF (C.EQ.1) THEN
                         WRITE(200,9314) GTML(COLUMN,ROW)
                      ELSEIF (C.EQ.2) THEN
                         WRITE(200,9315)
                         WRITE(200,9316)
                         WRITE(200,9317)
                         WRITE(200,9318)
                         WRITE(200,7016)
                         WRITE(200,7017)
                         WRITE(200,7018)
                         WRITE(200,9319)
                         WRITE(200,9320)
                         WRITE(200,7019)
                         WRITE(200,7020)
                         WRITE(200,7021)
                         WRITE(200,9321)
                         WRITE(200,7022)
                         WRITE(200,7023)
                         WRITE(200,7024)
                         WRITE(200,7025)
                         WRITE(200,9322)
                         WRITE(200,9323)
                         WRITE(200,9324)
                         WRITE(200,9325)
                         WRITE(200,9326)
                         WRITE(200,9327)
                         WRITE(200,7026)
                         WRITE(200,9328)
                         WRITE(200,9329)
                         WRITE(200,9330)
                      ENDIF
5876               CONTINUE
                ENDIF
```

```
               MN=MN+1
            ENDIF
            IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.1) THEN
               CLADRHO=6.56
            ELSEIF (GTMAT(DESNUM(COLUMN,ROW)).EQ.2) THEN
               CLADRHO=7.90
            ELSEIF (GTMAT(DESNUM(COLUMN,ROW)).EQ.3) THEN
               CLADRHO=8.19
            ENDIF
            WRITE(30,5888) LN, GTML(COLUMN,ROW), (-1*CLADRHO),
     c      GTIRSURF,
     c      (-1*GTORSURF), (-1*GTTOPSURF), GTBOTSURF,
     c      APSRAUNIV(COLUMN,ROW)
 5888       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Guide tube region')
            LN=LN+1
*     Write the moderator cells within the GT in this APSR universe.
            WRITE(30,5890) LN, BMODML, (-1*MODDENSITY),
     c      (-1*GTIRSURF),
     c      APSRCLADORSURF, (-1*GTTOPSURF), GTBOTSURF,
     c      APSRAUNIV(COLUMN,ROW)
 5890       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,
     c      '    $ Borated moderator inside guide tube')
            LN=LN+1
            WRITE(30,5900) LN, BMODML, (-1*MODDENSITY),
     c      (-1*APSRCLADORSURF), (-1*APSRCLADBOTTOMSURF), GTBOTSURF,
     c      APSRAUNIV(COLUMN,ROW)
 5900       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,
     c      '    $ Borated moderator inside guide tube')
            LN=LN+1
*     Loop through the regions above the APSR (i.e. the appropriate upper
core regions)
*     Define the upper region lower surface.
               DO 6050 REGION=1,NUMREGABOVEAPSRA
*     Determine the current upper region's lower surface specification.
                  IF (REGION.EQ.1) THEN
                     REGIONTOPSURF=SYSTEMTOP
                     CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
     c               REGABOVEAPSRA(REGION,1)
                  ENDIF
                  CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
     c            REGABOVEAPSRA(REGION,1)
                  IF (REGION.EQ.NUMREGABOVEAPSRA) THEN
                     REGIONBOTTOMSURF=UEFTOPSURF
                  ELSE
                  CURRENTSURFLABEL=0
                  DO 5910 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                           CURRENTSURFLABEL=V
                           EXIT
                        ENDIF
```

```
                    ENDIF
 5910            CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
                    REGIONBOTTOMSURF=SN
                    SURFTYPESPEC(SN)='PZ'
                    SURFVALUESPEC(SN)=CURRENTSURF
                    SN=SN+1
                 ELSE
                    REGIONBOTTOMSURF=CURRENTSURFLABEL
                 ENDIF
                 ENDIF
*     Write the cell specification for the APSR universe upper region.
               IF (REGION.EQ.1) THEN
                 WRITE(30,6040) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c           (-1*REGABOVEAPSRA(REGION,2)),
     c           REGIONBOTTOMSURF, APSRAUNIV(COLUMN,ROW), REGION
 6040            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
     c           ' IMP:N=1 U=',I3,'    $ Upper core region ',I2)
                 LN=LN+1
                 REGIONTOPSURF=REGIONBOTTOMSURF
               ELSE
                 WRITE(30,6045) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c           (-1*REGABOVEAPSRA(REGION,2)), (-1*REGIONTOPSURF),
     c           REGIONBOTTOMSURF, APSRAUNIV(COLUMN,ROW), REGION
 6045            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'    $ Upper core region ',I2)
                 LN=LN+1
                 REGIONTOPSURF=REGIONBOTTOMSURF
               ENDIF
 6050          CONTINUE
          SPACHEIGHT=0.0
*     Loop through the spacer and moderator regions along the axial
*     length of the GT (from top to bottom).
          DO 6051 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
             SPACHEIGHT=SPACHEIGHT+SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
 6051     CONTINUE
          DO 6130 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
*     Define the homogenized spacer region bounding surfaces.
             IF (SPN.EQ.1) THEN
               SPACERTOPSURF=UEFBOTTOMSURF
               CURRENTSURF=SURFVALUESPEC(UEFBOTTOMSURF)-
     c         SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
               CURRENTSURFLABEL=0
               DO 6060 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                   ENDIF
                ENDIF
 6060          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  SPACERBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
```

```
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                SPACERBOTTOMSURF=CURRENTSURFLABEL
             ENDIF
             WATERREGIONTOPSURF=SPACERBOTTOMSURF
             CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
             CURRENTSURFLABEL=0
             DO 6070 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
6070         CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                WATERREGIONBOTTOMSURF=SN
                SURFTYPESPEC(SN)='PZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
             ENDIF
          ELSEIF ((SPN.NE.1).AND.(SPN.NE.
     c    NUMOFSPACERS(DESNUM(COLUMN,ROW)))) THEN
             SPACERTOPSURF=WATERREGIONBOTTOMSURF
             CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c       SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
             CURRENTSURFLABEL=0
             DO 6080 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
6080         CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                SPACERBOTTOMSURF=SN
                SURFTYPESPEC(SN)='PZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                SPACERBOTTOMSURF=CURRENTSURFLABEL
             ENDIF
             WATERREGIONTOPSURF=SPACERBOTTOMSURF
             CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
             CURRENTSURFLABEL=0
             DO 6090 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
```

```
                    ENDIF
                 ENDIF
 6090         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 WATERREGIONBOTTOMSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
              ENDIF
           ELSEIF (SPN.EQ.NUMOFSPACERS(DESNUM(COLUMN,ROW))) THEN
              SPACERTOPSURF=WATERREGIONBOTTOMSURF
              CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c        SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
              CURRENTSURFLABEL=0
              DO 6100 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 6100         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 SPACERBOTTOMSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 SPACERBOTTOMSURF=CURRENTSURFLABEL
              ENDIF
              WATERREGIONTOPSURF=SPACERBOTTOMSURF
              WATERREGIONBOTTOMSURF=NODEBOTTOMSURF
           ENDIF
*      Write the current homogenized spacer region cell in this APSR universe.
           WRITE(30,6126) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c     (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)), GTORSURF,
     c     (-1*SPACERTOPSURF), SPACERBOTTOMSURF, APSRAUNIV(COLUMN,ROW),
     c     SPN
 6126      FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c     ' IMP:N=1 U=',I3,'   $ Homogenized region for spacer ',I2)
           LN=LN+1
*      Write the water region cell below the current homogenized spacer cell
in this APSR universe.
           WRITE(30,6127) LN, BMODML, (-1*MODDENSITY), GTORSURF,
     c     (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c     APSRAUNIV(COLUMN,ROW)
 6127      FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c     ' IMP:N=1 U=',I3,'   $ Borated moderator')
           LN=LN+1
 6130    CONTINUE
      ENDIF
 6140    CONTINUE
```

```
 6150 CONTINUE
      ENDIF
*  Write the specifications for the BPR universes that are
*  required to fill the assembly layout specifications previously defined.
      DO 9290 ROW=1,50
         DO 9280 COLUMN=1,50
*     Write the BPR universe specification for the assembly if it
*     contains a unique BP material.
*
      IF (BPRUNIQUE(COLUMN,ROW).EQ..TRUE.) THEN
*  Write the BPR specification header.
         WRITE(30,9072)
 9072    FORMAT(T1,'C')
         WRITE(30,9074) ASSYID(COLUMN,ROW)
 9074    FORMAT(T1,
     c   'C   BPR UNIVERSE SPECIFICATION FOR ASSEMBLY ',A5)
         WRITE(30,9076)
 9076    FORMAT(T1,'C')
      IF ((BANDW.EQ..TRUE.).OR.
     c ((WESTINGHOUSE.EQ..TRUE.).AND.
     c (WBPRA(BANKNUM(COLUMN,ROW)).EQ.1).AND.
     c (GTSPLIT.NE.1))) THEN
         DO 9090 MCNPNODE=1,NUMOFBPRANODES(BANKNUM(COLUMN,ROW))
*     Define the upper end-fitting bottom surface.
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 9077 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 9077          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  UEFBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  UEFBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
*     Define the upper end-fitting top surface.
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),1)+
     c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 9078 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
```

```
                 ENDIF
 9078         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 UEFTOPSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 UEFTOPSURF=CURRENTSURFLABEL
              ENDIF
*     Define the BP node bounding surfaces.
           IF (MCNPNODE.EQ.1) THEN
              TOTBPHEIGHT=0.0
              DO 9079 Z=1,NUMOFBPRANODES(BANKNUM(COLUMN,ROW))
                 TOTBPHEIGHT=TOTBPHEIGHT+
     c           MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),Z)
 9079         CONTINUE
              CURRENTSURF=BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))+
     c        TOTBPHEIGHT
              IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                 CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
              ENDIF
              CURRENTSURFLABEL=0
              DO 9080 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 9080         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 TOPBPNODETOPSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 TOPBPNODETOPSURF=CURRENTSURFLABEL
              ENDIF
              BPNODETOPSURF=TOPBPNODETOPSURF
              CURRENTSURF=SURFVALUESPEC(BPNODETOPSURF)-
     c        MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),MCNPNODE)
              IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                 CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
              ENDIF
              CURRENTSURFLABEL=0
              DO 9082 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 9082         CONTINUE
```

```
                  IF (CURRENTSURFLABEL.EQ.0) THEN
                     BPNODEBOTTOMSURF=SN
                     SURFTYPESPEC(SN)='PZ'
                     SURFVALUESPEC(SN)=CURRENTSURF
                     SN=SN+1
                  ELSE
                     BPNODEBOTTOMSURF=CURRENTSURFLABEL
                  ENDIF
                  CURRENTSURF=BPRDIM(BANKNUM(COLUMN,ROW),1)
                  CURRENTSURFLABEL=0
                  DO 9083 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'CZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                           CURRENTSURFLABEL=V
                           EXIT
                        ENDIF
                     ENDIF
9083              CONTINUE
                  IF (CURRENTSURFLABEL.EQ.0) THEN
                     BPRADIUS=SN
                     SURFTYPESPEC(SN)='CZ'
                     SURFVALUESPEC(SN)=CURRENTSURF
                     SN=SN+1
                  ELSE
                     BPRADIUS=CURRENTSURFLABEL
                  ENDIF
               ELSEIF (MCNPNODE.NE.1) THEN
                  BPNODETOPSURF=BPNODEBOTTOMSURF
                  CURRENTSURF=SURFVALUESPEC(BPNODETOPSURF)-
     c            MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),MCNPNODE)
                  IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                     CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                  ENDIF
                  CURRENTSURFLABEL=0
                  DO 9084 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                           CURRENTSURFLABEL=V
                           EXIT
                        ENDIF
                     ENDIF
9084              CONTINUE
                  IF (CURRENTSURFLABEL.EQ.0) THEN
                     BPNODEBOTTOMSURF=SN
                     SURFTYPESPEC(SN)='PZ'
                     SURFVALUESPEC(SN)=CURRENTSURF
                     SN=SN+1
                  ELSE
                     BPNODEBOTTOMSURF=CURRENTSURFLABEL
                  ENDIF
               ENDIF
            IF (SURFVALUESPEC(BPNODEBOTTOMSURF).LT.
     c      SURFVALUESPEC(UEFTOPSURF)) THEN
*    Write the BP node cells in this BPR universe.
```

```
            IF ((BPRABSNODE(BANKNUM(COLUMN,ROW),MCNPNODE).EQ.'Y').AND.
     c         (BPNONABSMAT(BANKNUM(COLUMN,ROW)).EQ.1)) THEN
               WRITE(30,9085) LN, BPNODEML,
     c         (-1*AL2O3DENSITY(BANKNUM(COLUMN,ROW))), (-1*BPRADIUS),
     c         (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c         BPRAUNIV(COLUMN,ROW), MCNPNODE
 9085          FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Burnable poison node ',I2)
               LN=LN+1
               BPNODEML=BPNODEML+1
            ELSEIF ((BPRABSNODE(BANKNUM(COLUMN,ROW),MCNPNODE).EQ.'Y').AND.
     c         (BPNONABSMAT(BANKNUM(COLUMN,ROW)).NE.1)) THEN
               WRITE(30,9086) LN, BPNODEML,
     c         (-1*NONBPMATDATA(BANKNUM(COLUMN,ROW),1)), (-1*BPRADIUS),
     c         (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c         BPRAUNIV(COLUMN,ROW), MCNPNODE
 9086          FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Burnable poison node ',I2)
               LN=LN+1
               BPNODEML=BPNODEML+1
            ELSE
               WRITE(30,9088) LN, BPNODEML,
     c         (-1*BPDENTOGO(COLUMN,ROW,MCNPNODE)), (-1*BPRADIUS),
     c         (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c         BPRAUNIV(COLUMN,ROW), MCNPNODE
 9088          FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Burnable poison node ',I2)
               LN=LN+1
               BPNODEML=BPNODEML+1
            ENDIF
            ENDIF
 9090        CONTINUE
*     Define the BPR cladding inner radius.
                 CURRENTSURF=BPRDIM(BANKNUM(COLUMN,ROW),2)
                 CURRENTSURFLABEL=0
                 DO 9092 V=1,(SN-1)
                    IF (SURFTYPESPEC(V).EQ.'CZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                    ENDIF
 9092            CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
                    BPCLADIRSURF=SN
                    SURFTYPESPEC(SN)='CZ'
                    SURFVALUESPEC(SN)=CURRENTSURF
                    SN=SN+1
                 ELSE
                    BPCLADIRSURF=CURRENTSURFLABEL
                 ENDIF
*     Define the BPR cladding outer radius.
                 CURRENTSURF=BPRDIM(BANKNUM(COLUMN,ROW),3)
                 CURRENTSURFLABEL=0
```

```
              DO 9094 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'CZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
9094          CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 BPCLADORSURF=SN
                 SURFTYPESPEC(SN)='CZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 BPCLADORSURF=CURRENTSURFLABEL
              ENDIF
*     Define the BPR cladding top surface.
              CURRENTSURF=TOTBPHEIGHT+
     c        BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))+
     c        BPRPLEN(BANKNUM(COLUMN,ROW),1)
              IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                 CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
              ENDIF
              CURRENTSURFLABEL=0
              DO 9096 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
9096          CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 BPCLADTOPSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 BPCLADTOPSURF=CURRENTSURFLABEL
              ENDIF
*     Define the BPR cladding bottom surface.
              CURRENTSURF=BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))-
     c        BPRPLEN(BANKNUM(COLUMN,ROW),2)
              CURRENTSURFLABEL=0
              DO 9098 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
9098          CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 BPCLADBOTTOMSURF=SN
```

```
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPCLADBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
*     Write the BP-to-cladding gap cell in this BPR universe.
            WRITE(30,9104) LN, (-1*BPCLADIRSURF), BPRADIUS,
     c      (-1*TOPBPNODETOPSURF),
     c      BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 9104       FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,
     c      '    $ Burnable poison-to-cladding gap')
            LN=LN+1
*     Write the BPR cladding cell in this BPR universe.
*     Determine if the BPR cladding material specification has
*     previously been defined. If it has been previously defined, determine
*     the cladding material specification label.
            CLADMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 9107 RO=1,(ROW-1)
                  DO 9106 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                        IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  BPRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
 9106             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 9107          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 9109 RO=ROW,ROW
                     DO 9108 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                           IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                     BPRCLADMAT(BANKNUM(CO,RO))) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
                        ENDIF
 9108                CONTINUE
```

```
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 9109             CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 9111 RO=1,(ROW-1)
                  DO 9110 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                        IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  BPRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
 9110             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 9111          CONTINUE
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
               DO 9113 RO=1,1
                  DO 9112 CO=1,(COLUMN-1)
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).NE.'BPRA ') THEN
                        IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  BPRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
 9112             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 9113          CONTINUE
            ENDIF
            IF (CLADMLUNIQUE.EQ..TRUE.) THEN
               BPCLADML(COLUMN,ROW)=MN
*  Check BPR Cladding Material
                  IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
                     DO 9115 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9300) BPCLADML(COLUMN,ROW)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9301)
                           WRITE(200,7000)
```

```
                    WRITE(200,7001)
                    WRITE(200,7002)
                    WRITE(200,9302)
                    WRITE(200,7003)
                    WRITE(200,7004)
                    WRITE(200,7005)
                    WRITE(200,9303)
                    WRITE(200,9304)
                 ENDIF
9115          CONTINUE
           ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW))
     c     .EQ.2) THEN
              DO 9117 C=1,2
                 IF (C.EQ.1) THEN
                    WRITE(200,9305) BPCLADML(COLUMN,ROW)
                 ELSEIF (C.EQ.2) THEN
                    WRITE(200,9306)
                    WRITE(200,9307)
                    WRITE(200,9308)
                    WRITE(200,9309)
                    WRITE(200,9310)
                    WRITE(200,7006)
                    WRITE(200,7007)
                    WRITE(200,7008)
                    WRITE(200,9311)
                    WRITE(200,9312)
                    WRITE(200,7009)
                    WRITE(200,7010)
                    WRITE(200,7011)
                    WRITE(200,9313)
                    WRITE(200,7012)
                    WRITE(200,7013)
                    WRITE(200,7014)
                    WRITE(200,7015)
                 ENDIF
9117          CONTINUE
           ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW))
     c     .EQ.3) THEN
              DO 9119 C=1,2
                 IF (C.EQ.1) THEN
                    WRITE(200,9314) BPCLADML(COLUMN,ROW)
                 ELSEIF (C.EQ.2) THEN
                    WRITE(200,9315)
                    WRITE(200,9316)
                    WRITE(200,9317)
                    WRITE(200,9318)
                    WRITE(200,7016)
                    WRITE(200,7017)
                    WRITE(200,7018)
                    WRITE(200,9319)
                    WRITE(200,9320)
                    WRITE(200,7019)
                    WRITE(200,7020)
                    WRITE(200,7021)
```

```
                        WRITE(200,9321)
                        WRITE(200,7022)
                        WRITE(200,7023)
                        WRITE(200,7024)
                        WRITE(200,7025)
                        WRITE(200,9322)
                        WRITE(200,9323)
                        WRITE(200,9324)
                        WRITE(200,9325)
                        WRITE(200,9326)
                        WRITE(200,9327)
                        WRITE(200,7026)
                        WRITE(200,9328)
                        WRITE(200,9329)
                        WRITE(200,9330)
                     ENDIF
 9119             CONTINUE
               ENDIF
            MN=MN+1
         ENDIF
         IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
            CLADRHO=6.56
         ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.2) THEN
            CLADRHO=7.90
         ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.3) THEN
            CLADRHO=8.19
         ENDIF
         WRITE(30,9131) LN, BPCLADML(COLUMN,ROW), (-1*CLADRHO),
     c   BPCLADIRSURF,
     c   (-1*BPCLADORSURF), (-1*BPCLADTOPSURF), BPCLADBOTTOMSURF,
     c   BPRAUNIV(COLUMN,ROW)
 9131    FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c   ' IMP:N=1 U=',I3,'   $ BPR cladding')
         LN=LN+1
*     Write the BPR upper plenum cell in this BPR universe.
*     Determine if the BPR upper plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the upper plenum material specification label.
         BPRUPMLUNIQUE=.TRUE.
         LEAVE=.FALSE.
         IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
            DO 9133 RO=1,(ROW-1)
               DO 9132 CO=1,50
                  IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKNUM(COLUMN,ROW).EQ.
     c                  BANKNUM(CO,RO)) THEN
                        BPRUPMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
 9132          CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
```

```
                    EXIT
                 ENDIF
 9133         CONTINUE
              IF (LEAVE.EQ..FALSE.) THEN
                 DO 9135 RO=ROW,ROW
                    DO 9134 CO=1,(COLUMN-1)
                       IF (BANKNUM(CO,RO).NE.0) THEN
                          IF (BANKNUM(COLUMN,ROW).EQ.
     c                    BANKNUM(CO,RO)) THEN
                             BPRUPMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                             EXIT
                          ENDIF
                       ENDIF
 9134               CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
 9135            CONTINUE
              ENDIF
           ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
              DO 9137 RO=1,(ROW-1)
                 DO 9136 CO=1,50
                    IF (BANKNUM(CO,RO).NE.0) THEN
                       IF (BANKNUM(COLUMN,ROW).EQ.
     c                 BANKNUM(CO,RO)) THEN
                          BPRUPMLUNIQUE=.FALSE.
                          LEAVE=.TRUE.
                          BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                          EXIT
                       ENDIF
                    ENDIF
 9136            CONTINUE
                 IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                 ENDIF
 9137         CONTINUE
           ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
              DO 9139 RO=1,1
                 DO 9138 CO=1,(COLUMN-1)
                    IF (BANKNUM(CO,RO).NE.0) THEN
                       IF (BANKNUM(COLUMN,ROW).EQ.
     c                 BANKNUM(CO,RO)) THEN
                          BPRUPMLUNIQUE=.FALSE.
                          LEAVE=.TRUE.
                          BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                          EXIT
                       ENDIF
                    ENDIF
 9138            CONTINUE
                 IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                 ENDIF
```

```
 9139          CONTINUE
            ENDIF
         IF (SURFVALUESPEC(TOPBPNODETOPSURF).LT.
     c   SURFVALUESPEC(UEFTOPSURF)) THEN
            IF (BPRUPMLUNIQUE.EQ..TRUE.) THEN
               BPRUPML(COLUMN,ROW)=MN
*  Check Burnable Poison Rod Upper Plenum Regions
                  DO 9143 C=1,BPRUPLENMAT(BANKNUM(COLUMN,ROW),2)
                     IF (C.EQ.1) THEN
                        WRITE(200,9140) BPRUPML(COLUMN,ROW),
     c                  BPRUPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                  (-1*BPRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 9140                   FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                  '    $ Burnable Poison Rod Upper Plenum')
                     ELSE
                        WRITE(200,9142)
     c                  BPRUPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                  (-1*BPRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 9142                   FORMAT(T9,A9,3X,G14.6)
                     ENDIF
 9143             CONTINUE
               MN=MN+1
            ENDIF
            WRITE(30,9144) LN, BPRUPML(COLUMN,ROW),
     c      (-1*BPRUPLENMAT(BANKNUM(COLUMN,ROW),1)),
     c      TOPBPNODETOPSURF,
     c      (-1*BPCLADTOPSURF), (-1*BPCLADIRSURF),
     c      BPRAUNIV(COLUMN,ROW)
 9144       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'    $ BPR upper plenum region')
            LN=LN+1
         ENDIF
*     Write the BPR lower plenum cell (lower end plug) in this BPR universe.
*     Determine if the BPR lower plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the lower plenum material specification label.
            BPRLPMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 9146 RO=1,(ROW-1)
                  DO 9145 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO)) THEN
                           BPRLPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 9145             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 9146          CONTINUE
```

```
                 IF (LEAVE.EQ..FALSE.) THEN
                    DO 9148 RO=ROW,ROW
                       DO 9147 CO=1,(COLUMN-1)
                          IF (BANKNUM(CO,RO).NE.0) THEN
                             IF (BANKNUM(COLUMN,ROW).EQ.
     c                       BANKNUM(CO,RO)) THEN
                                BPRLPMLUNIQUE=.FALSE.
                                LEAVE=.TRUE.
                                BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                                EXIT
                             ENDIF
                          ENDIF
 9147                  CONTINUE
                       IF (LEAVE.EQ..TRUE.) THEN
                          EXIT
                       ENDIF
 9148               CONTINUE
                 ENDIF
              ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                 DO 9150 RO=1,(ROW-1)
                    DO 9149 CO=1,50
                       IF (BANKNUM(CO,RO).NE.0) THEN
                          IF (BANKNUM(COLUMN,ROW).EQ.
     c                    BANKNUM(CO,RO)) THEN
                             BPRLPMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                             EXIT
                          ENDIF
                       ENDIF
 9149               CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
 9150            CONTINUE
              ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                 DO 9152 RO=1,1
                    DO 9151 CO=1,(COLUMN-1)
                       IF (BANKNUM(CO,RO).NE.0) THEN
                          IF (BANKNUM(COLUMN,ROW).EQ.
     c                    BANKNUM(CO,RO)) THEN
                             BPRLPMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                             EXIT
                          ENDIF
                       ENDIF
 9151               CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
 9152            CONTINUE
              ENDIF
              IF (BPRLPMLUNIQUE.EQ..TRUE.) THEN
```

```
                 BPRLPML(COLUMN,ROW)=MN
*  Check Burnable Poison Rod Lower Plenum Regions
                   DO 9155 C=1,BPRLPLENMAT(BANKNUM(COLUMN,ROW),2)
                     IF (C.EQ.1) THEN
                       WRITE(200,9153) BPRLPML(COLUMN,ROW),
     c                 BPRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                 (-1*BPRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 9153                  FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                 '    $ Burnable Poison Rod Lower Plenum')
                     ELSE
                       WRITE(200,9154)
     c                 BPRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                 (-1*BPRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 9154                  FORMAT(T9,A9,3X,G14.6)
                     ENDIF
 9155              CONTINUE
                 MN=MN+1
               ENDIF
               WRITE(30,9156) LN, BPRLPML(COLUMN,ROW),
     c         (-1*BPRLPLENMAT(BANKNUM(COLUMN,ROW),1)), BPCLADBOTTOMSURF,
     c         (-1*BPNODEBOTTOMSURF), (-1*BPCLADIRSURF),
     c         BPRAUNIV(COLUMN,ROW)
 9156          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ BPR lower plenum region')
               LN=LN+1
*    Define the GT top surface.
                 CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),3)
                 IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                   CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                 ENDIF
                 CURRENTSURFLABEL=0
                 DO 9157 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                     ENDIF
                   ENDIF
 9157            CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
                   GTTOPSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                 ELSE
                   GTTOPSURF=CURRENTSURFLABEL
                 ENDIF
*    Define the GT bottom surface.
                 CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),4)
                 CURRENTSURFLABEL=0
                 DO 9158 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
```

```
                  EXIT
               ENDIF
            ENDIF
 9158    CONTINUE
         IF (CURRENTSURFLABEL.EQ.0) THEN
            GTBOTSURF=SN
            SURFTYPESPEC(SN)='PZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
         ELSE
            GTBOTSURF=CURRENTSURFLABEL
         ENDIF
*     Define the GT outer radius surface.
         CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),2)
         CURRENTSURFLABEL=0
         DO 9159 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'CZ') THEN
        IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
 9159    CONTINUE
         IF (CURRENTSURFLABEL.EQ.0) THEN
            GTORSURF=SN
            SURFTYPESPEC(SN)='CZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
         ELSE
            GTORSURF=CURRENTSURFLABEL
         ENDIF
*     Define the GT inner radius surface.
         CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),1)
         CURRENTSURFLABEL=0
         DO 9160 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'CZ') THEN
        IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
 9160    CONTINUE
         IF (CURRENTSURFLABEL.EQ.0) THEN
            GTIRSURF=SN
            SURFTYPESPEC(SN)='CZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
         ELSE
            GTIRSURF=CURRENTSURFLABEL
         ENDIF
*     Define the lower end-fitting top surface.
         CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
         CURRENTSURFLABEL=0
         DO 9161 V=1,(SN-1)
```

```
            IF (SURFTYPESPEC(V).EQ.'PZ') THEN
        IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
 9161    CONTINUE
         IF (CURRENTSURFLABEL.EQ.0) THEN
            BPLEFTOPSURF=SN
            SURFTYPESPEC(SN)='PZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
         ELSE
            BPLEFTOPSURF=CURRENTSURFLABEL
         ENDIF
*    Write the lower end-fitting cell specification for this BPR universe.
        IF (SURFVALUESPEC(GTBOTSURF).GE.
     c  ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)) THEN
           WRITE(30,9162) LN, FRLEFML(COLUMN,ROW),
     c     (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*BPLEFTOPSURF),
     c     BPRAUNIV(COLUMN,ROW)
 9162      FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,' IMP:N=1 U=',I3,
     c     '    $ Assembly lower end-fitting')
           LN=LN+1
        ELSE
           WRITE(30,9163) LN, FRLEFML(COLUMN,ROW),
     c     (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*BPLEFTOPSURF),
     c     GTORSURF, BPRAUNIV(COLUMN,ROW)
 9163      FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c     ' IMP:N=1 U=',I3,'    $ Assembly lower end-fitting')
           LN=LN+1
           WRITE(30,9164) LN, FRLEFML(COLUMN,ROW),
     c     (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*GTBOTSURF),
     c     (-1*GTORSURF), BPRAUNIV(COLUMN,ROW)
 9164      FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c     ' IMP:N=1 U=',I3,'    $ Assembly lower end-fitting')
           LN=LN+1
        ENDIF
*    Write the upper end-fitting cell specification for this BPR universe.
        IF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c  SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c  (SURFVALUESPEC(GTTOPSURF).LE.
     c  SURFVALUESPEC(UEFBOTTOMSURF))) THEN
           WRITE(30,9170) LN, FRUEFML(COLUMN,ROW),
     c     (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c     (-1*UEFTOPSURF), BPRAUNIV(COLUMN,ROW)
 9170      FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c     ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
           LN=LN+1
        ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c  SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c  (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c  SURFVALUESPEC(UEFTOPSURF)).AND.
     c  (SURFVALUESPEC(GTTOPSURF).LE.
```

```
     c       SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                WRITE(30,9172) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), BPCLADORSURF, BPRAUNIV(COLUMN,ROW)
 9172           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,9174) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), BPCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*BPCLADORSURF),
     c          BPRAUNIV(COLUMN,ROW)
 9174           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c       SURFVALUESPEC(GTTOPSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                WRITE(30,9176) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
 9176           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,9178) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTORSURF),
     c          BPRAUNIV(COLUMN,ROW)
 9178           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,9180) LN, BMODML,
     c          (-1*MODDENSITY), BPCLADTOPSURF,
     c          (-1*GTTOPSURF), (-1*GTIRSURF),
     c          BPRAUNIV(COLUMN,ROW)
 9180           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c       SURFVALUESPEC(GTTOPSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).EQ.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,9182) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
 9182           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
             IF (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c       SURFVALUESPEC(GTTOPSURF)) THEN
                WRITE(30,9184) LN, BMODML,
     c          (-1*MODDENSITY), BPCLADTOPSURF,
```

```
     c          (-1*UEFTOPSURF), (-1*GTIRSURF),
     c          BPRAUNIV(COLUMN,ROW)
9184            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
             ENDIF
             ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).LT.
     c       SURFVALUESPEC(BPCLADTOPSURF))) THEN
                WRITE(30,9186) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
9186            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,9188) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTORSURF),
     c          BPCLADORSURF, BPRAUNIV(COLUMN,ROW)
9188            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          1X,I4,' IMP:N=1 U=',I3,
     c          '   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,9190) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), BPCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*BPCLADORSURF),
     c          BPRAUNIV(COLUMN,ROW)
9190            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).EQ.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).LT.
     c       SURFVALUESPEC(BPCLADTOPSURF))) THEN
                WRITE(30,9192) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
9192            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,9194) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTORSURF),
     c          BPCLADORSURF, BPRAUNIV(COLUMN,ROW)
9194            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          1X,I4,' IMP:N=1 U=',I3,
     c          '   $ Assembly upper end-fitting')
```

```
                  LN=LN+1
               ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c         SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c         (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c         SURFVALUESPEC(UEFTOPSURF)).AND.
     c         (SURFVALUESPEC(GTTOPSURF).EQ.
     c         SURFVALUESPEC(BPCLADTOPSURF))) THEN
                  WRITE(30,9196) LN, FRUEFML(COLUMN,ROW),
     c            (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c            (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
 9196             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c            ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                  LN=LN+1
                  WRITE(30,9198) LN, FRUEFML(COLUMN,ROW),
     c            (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c            (-1*UEFTOPSURF), (-1*GTORSURF),
     c            BPCLADORSURF, BPRAUNIV(COLUMN,ROW)
 9198             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c            1X,I4,' IMP:N=1 U=',I3,
     c            '   $ Assembly upper end-fitting')
                  LN=LN+1
               ENDIF
*     Write the GT material cell in this BPR universe.
*     Determine if the GT material specification has
*     previously been defined.  If it has been previously defined, determine
*     the material specification label.
               CLADMLUNIQUE=.TRUE.
               LEAVE=.FALSE.
               IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                  DO 9201 RO=1,(ROW-1)
                     DO 9200 CO=1,50
                        IF (DESNUM(CO,RO).NE.0) THEN
                           IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                     GTMAT(DESNUM(CO,RO))) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              GTML(COLUMN,ROW)=GTML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 9200                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 9201             CONTINUE
                  IF (LEAVE.EQ..FALSE.) THEN
                     DO 9203 RO=ROW,ROW
                        DO 9202 CO=1,(COLUMN-1)
                           IF (DESNUM(CO,RO).NE.0) THEN
                              IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                        GTMAT(DESNUM(CO,RO))) THEN
                                 CLADMLUNIQUE=.FALSE.
                                 LEAVE=.TRUE.
                                 GTML(COLUMN,ROW)=GTML(CO,RO)
```

```
                          EXIT
                       ENDIF
                    ENDIF
 9202            CONTINUE
                 IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                 ENDIF
 9203         CONTINUE
           ENDIF
        ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
           DO 9205 RO=1,(ROW-1)
              DO 9204 CO=1,50
                 IF (DESNUM(CO,RO).NE.0) THEN
                    IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c              GTMAT(DESNUM(CO,RO))) THEN
                       CLADMLUNIQUE=.FALSE.
                       LEAVE=.TRUE.
                       GTML(COLUMN,ROW)=GTML(CO,RO)
                       EXIT
                    ENDIF
                 ENDIF
 9204         CONTINUE
              IF (LEAVE.EQ..TRUE.) THEN
                 EXIT
              ENDIF
 9205      CONTINUE
        ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
           DO 9207 RO=1,1
              DO 9206 CO=1,(COLUMN-1)
                 IF (DESNUM(CO,RO).NE.0) THEN
                    IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c              GTMAT(DESNUM(CO,RO))) THEN
                       CLADMLUNIQUE=.FALSE.
                       LEAVE=.TRUE.
                       GTML(COLUMN,ROW)=GTML(CO,RO)
                       EXIT
                    ENDIF
                 ENDIF
 9206         CONTINUE
              IF (LEAVE.EQ..TRUE.) THEN
                 EXIT
              ENDIF
 9207      CONTINUE
        ENDIF
        IF (CLADMLUNIQUE.EQ..TRUE.) THEN
           GTML(COLUMN,ROW)=MN
*  Check Guide Tube Material
              IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.1) THEN
                 DO 9210 C=1,2
                    IF (C.EQ.1) THEN
                       WRITE(200,9300) GTML(COLUMN,ROW)
                    ELSEIF (C.EQ.2) THEN
                       WRITE(200,9301)
                       WRITE(200,7000)
```

```
                        WRITE(200,7001)
                        WRITE(200,7002)
                        WRITE(200,9302)
                        WRITE(200,7003)
                        WRITE(200,7004)
                        WRITE(200,7005)
                        WRITE(200,9303)
                        WRITE(200,9304)
                     ENDIF
9210              CONTINUE
               ELSEIF (GTMAT(DESNUM(COLUMN,ROW))
     c         .EQ.2) THEN
                  DO 9212 C=1,2
                     IF (C.EQ.1) THEN
                        WRITE(200,9305) GTML(COLUMN,ROW)
                     ELSEIF (C.EQ.2) THEN
                        WRITE(200,9306)
                        WRITE(200,9307)
                        WRITE(200,9308)
                        WRITE(200,9309)
                        WRITE(200,9310)
                        WRITE(200,7006)
                        WRITE(200,7007)
                        WRITE(200,7008)
                        WRITE(200,9311)
                        WRITE(200,9312)
                        WRITE(200,7009)
                        WRITE(200,7010)
                        WRITE(200,7011)
                        WRITE(200,9313)
                        WRITE(200,7012)
                        WRITE(200,7013)
                        WRITE(200,7014)
                        WRITE(200,7015)
                     ENDIF
9212              CONTINUE
               ELSEIF (GTMAT(DESNUM(COLUMN,ROW))
     c         .EQ.3) THEN
                  DO 9214 C=1,2
                     IF (C.EQ.1) THEN
                        WRITE(200,9314) GTML(COLUMN,ROW)
                     ELSEIF (C.EQ.2) THEN
                        WRITE(200,9315)
                        WRITE(200,9316)
                        WRITE(200,9317)
                        WRITE(200,9318)
                        WRITE(200,7016)
                        WRITE(200,7017)
                        WRITE(200,7018)
                        WRITE(200,9319)
                        WRITE(200,9320)
                        WRITE(200,7019)
                        WRITE(200,7020)
                        WRITE(200,7021)
```

```
                        WRITE(200,9321)
                        WRITE(200,7022)
                        WRITE(200,7023)
                        WRITE(200,7024)
                        WRITE(200,7025)
                        WRITE(200,9322)
                        WRITE(200,9323)
                        WRITE(200,9324)
                        WRITE(200,9325)
                        WRITE(200,9326)
                        WRITE(200,9327)
                        WRITE(200,7026)
                        WRITE(200,9328)
                        WRITE(200,9329)
                        WRITE(200,9330)
                     ENDIF
9214              CONTINUE
               ENDIF
            MN=MN+1
         ENDIF
         IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.1) THEN
            CLADRHO=6.56
         ELSEIF (GTMAT(DESNUM(COLUMN,ROW)).EQ.2) THEN
            CLADRHO=7.90
         ELSEIF (GTMAT(DESNUM(COLUMN,ROW)).EQ.3) THEN
            CLADRHO=8.19
         ENDIF
         WRITE(30,9225) LN, GTML(COLUMN,ROW), (-1*CLADRHO),
    c    GTIRSURF,
    c    (-1*GTORSURF), (-1*GTTOPSURF), GTBOTSURF,
    c    BPRAUNIV(COLUMN,ROW)
9225     FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
    c    ' IMP:N=1 U=',I3,'   $ Guide tube region')
         LN=LN+1
*     Write the moderator cells within the GT in this BPR universe.
         IF (SURFVALUESPEC(BPCLADTOPSURF).GE.
    c    SURFVALUESPEC(GTTOPSURF)) THEN
            WRITE(30,9226) LN, BMODML, (-1*MODDENSITY),
    c       (-1*GTIRSURF),
    c       BPCLADORSURF, (-1*GTTOPSURF), GTBOTSURF,
    c       BPRAUNIV(COLUMN,ROW)
9226        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
    c       ' IMP:N=1 U=',I3,
    c       '   $ Borated moderator inside guide tube')
            LN=LN+1
         ELSEIF (SURFVALUESPEC(BPCLADTOPSURF).LT.
    c    SURFVALUESPEC(GTTOPSURF)) THEN
            WRITE(30,9227) LN, BMODML, (-1*MODDENSITY),
    c       (-1*GTIRSURF),
    c       BPCLADORSURF, (-1*BPCLADTOPSURF), GTBOTSURF,
    c       BPRAUNIV(COLUMN,ROW)
9227        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
    c       ' IMP:N=1 U=',I3,
    c       '   $ Borated moderator inside guide tube')
```

```
               LN=LN+1
            ENDIF
            WRITE(30,9228) LN, BMODML, (-1*MODDENSITY),
     c      (-1*BPCLADORSURF), (-1*BPCLADBOTTOMSURF), GTBOTSURF,
     c      BPRAUNIV(COLUMN,ROW)
 9228       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Borated moderator inside guide tube')
            LN=LN+1
*     Loop through the regions above the BPR (i.e. the appropriate upper core
regions)
*     Define the upper region lower surface.
               DO 9232 REGION=1,NUMREGABOVEBPRA
*     Determine the current upper region's lower surface specification.
                  IF (REGION.EQ.1) THEN
                     REGIONTOPSURF=SYSTEMTOP
                     CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
     c               REGABOVEBPRA(REGION,1)
                  ENDIF
                  CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
     c            REGABOVEBPRA(REGION,1)
                  IF (REGION.EQ.NUMREGABOVEBPRA) THEN
                     REGIONBOTTOMSURF=UEFTOPSURF
                  ELSE
                  CURRENTSURFLABEL=0
                  DO 9229 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                           CURRENTSURFLABEL=V
                           EXIT
                        ENDIF
                     ENDIF
 9229             CONTINUE
                  IF (CURRENTSURFLABEL.EQ.0) THEN
                     REGIONBOTTOMSURF=SN
                     SURFTYPESPEC(SN)='PZ'
                     SURFVALUESPEC(SN)=CURRENTSURF
                     SN=SN+1
                  ELSE
                     REGIONBOTTOMSURF=CURRENTSURFLABEL
                  ENDIF
                  ENDIF
*     Write the cell specification for the BPR universe upper region.
               IF (REGION.EQ.1) THEN
                  WRITE(30,9230) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c            (-1*REGABOVEBPRA(REGION,2)),
     c            REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 9230             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
     c            ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                  LN=LN+1
                  REGIONTOPSURF=REGIONBOTTOMSURF
               ELSE
                  WRITE(30,9231) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c            (-1*REGABOVEBPRA(REGION,2)), (-1*REGIONTOPSURF),
     c            REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
```

```
9231             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
    c            ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                 LN=LN+1
                 REGIONTOPSURF=REGIONBOTTOMSURF
              ENDIF
 9232         CONTINUE
         SPACHEIGHT=0.0
*     Loop through the spacer and moderator regions along the axial
*     length of the BPR (from top to bottom).
         DO 9233 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
            SPACHEIGHT=SPACHEIGHT+SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
 9233    CONTINUE
         DO 9270 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
*     Define the homogenized spacer region bounding surfaces.
           IF (SPN.EQ.1) THEN
              SPACERTOPSURF=UEFBOTTOMSURF
              CURRENTSURF=SURFVALUESPEC(UEFBOTTOMSURF)-
    c         SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
              CURRENTSURFLABEL=0
              DO 9234 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                 ENDIF
              ENDIF
 9234         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 SPACERBOTTOMSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 SPACERBOTTOMSURF=CURRENTSURFLABEL
              ENDIF
              WATERREGIONTOPSURF=SPACERBOTTOMSURF
              CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
              CURRENTSURFLABEL=0
              DO 9236 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                 ENDIF
              ENDIF
 9236         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 WATERREGIONBOTTOMSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
              ENDIF
```

```
            ELSEIF ((SPN.NE.1).AND.(SPN.NE.
     c      NUMOFSPACERS(DESNUM(COLUMN,ROW)))) THEN
               SPACERTOPSURF=WATERREGIONBOTTOMSURF
               CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c         SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
               CURRENTSURFLABEL=0
               DO 9238 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 9238          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  SPACERBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  SPACERBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
               WATERREGIONTOPSURF=SPACERBOTTOMSURF
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
               CURRENTSURFLABEL=0
               DO 9240 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 9240          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  WATERREGIONBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
            ELSEIF (SPN.EQ.NUMOFSPACERS(DESNUM(COLUMN,ROW))) THEN
               SPACERTOPSURF=WATERREGIONBOTTOMSURF
               CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c         SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
               CURRENTSURFLABEL=0
               DO 9242 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 9242          CONTINUE
```

```
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 SPACERBOTTOMSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 SPACERBOTTOMSURF=CURRENTSURFLABEL
              ENDIF
              WATERREGIONTOPSURF=SPACERBOTTOMSURF
              WATERREGIONBOTTOMSURF=NODEBOTTOMSURF
            ENDIF
*    Write the current homogenized spacer region cell in this BPR universe.
            WRITE(30,9268) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c      (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)), GTORSURF,
     c      (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c      SPN
 9268       FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Homogenized region for spacer ',I2)
            LN=LN+1
*    Write the water region cell below the current homogenized spacer cell
in this BPR universe.
            WRITE(30,9269) LN, BMODML, (-1*MODDENSITY), GTORSURF,
     c      (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW)
 9269       FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Borated moderator')
            LN=LN+1
 9270     CONTINUE
      ELSE
         CALL WESTBPR(BANKNUM, BMODML, BPCLADML, BPNODEML,
    c BPRAUNIV, BPRCLADMAT, BPRLPML, BPRUPML, COLUMN, DESNUM,
    c FRLEFML, FRUEFML, FRUREGIONML, GTAXMAT, GTAXML, GTDATA,
    c GTMAT, GTML, GTSPLIT, HOMOSPACMLNUM, LN, MN,
    c NUMOFBPRANODES, NUMOFGTAXS, NUMOFSPACERS,
    c NUMREGABOVEBPRA, ROW, SN, SYSTEMTOP, WBPRA,
    c WBPRATYPE, AL2O3B4CDENSITY, AL2O3DENSITY,
    c BOTBPNODEHEIGHT, BPDENTOGO, BPNONABSMAT,
    c BPRAXDIM, BPRDIM, BPRLPLENMAT, BPRLPLENWTS,
    c BPRPLEN, BPRUPLENMAT, BPRUPLENWTS, ENDFITHEIGHT,
    c GTAXDATA, HOMOSPACERDEN, LEFMAT, MCNPBPRAHEIGHT,
    c MODDENSITY, NONBPMATDATA, REGABOVEBPRA,
    c SPACERDIST, SPACERHEIGHT, SURFVALUESPEC, UEFMAT,
    c BANKDES, BPRABSNODE, BPRLPLENZAIDS,
    c BPRUPLENZAIDS, CURRENTSURFLABEL, SURFTYPESPEC,
    c NODEBOTTOMSURF)
      ENDIF
      ENDIF
9280     CONTINUE
9290  CONTINUE
      CLOSE(UNIT=30)

      RETURN
      END
```

```
*****************************************************************
*     SUBROUTINE SURFSECTION                                    *
*     This subroutine writes the introduction section of the MCNP *
*     input deck.                                               *
*****************************************************************
      SUBROUTINE SURFSECTION (SN, EFPD, SURFVALUESPEC, SURFTYPESPEC,
     c PREFIX, CYCLE, SURFFILE, PVOUTERSURF, SYSTEMSOUTH,
     c SYSTEMWEST, SYSTEMTOP, SYSTEMBOTTOM, NPLOWDEG, NPHIGHDEG)
*
      INTEGER SN, NUMSTPT1, NUMSTPT2, NUMSTPT3, PVOUTERSURF,
     c SYSTEMSOUTH, SYSTEMWEST, SYSTEMTOP, SYSTEMBOTTOM,
     c SURFCOUNT
*
      REAL EFPD, SURFVALUESPEC(200), NPLOWDEG, NPHIGHDEG, A, PI
*
      CHARACTER SURFTYPESPEC(200)*2, PREFIX*3, CYCLE*2, SURFFILE*15,
     c CHSTPT1*1, CHSTPT2*1, CHSTPT3*1
*
      PI=3.14159265359
*     Open the file to contain the surface specification section of the MCNP
input deck.
      SURFFILE(1:3)=PREFIX
      SURFFILE(4:4)='C'
      SURFFILE(5:6)=CYCLE
      SURFFILE(7:7)='T'
      NUMSTPT1=INT(EFPD/100.0)
      CHSTPT1=CHAR(NUMSTPT1+48)
      NUMSTPT2=INT((EFPD-(NUMSTPT1*100))/10.0)
      CHSTPT2=CHAR(NUMSTPT2+48)
      NUMSTPT3=INT(EFPD-(NUMSTPT1*100)-
     c (NUMSTPT2*10))
      CHSTPT3=CHAR(NUMSTPT3+48)
      SURFFILE(8:8)=CHSTPT1
      SURFFILE(9:9)=CHSTPT2
      SURFFILE(10:10)=CHSTPT3
      SURFFILE(11:15)='.surf'
      OPEN(UNIT=40, FILE=SURFFILE, STATUS='UNKNOWN')
      REWIND(UNIT=40)
      WRITE(40,*)
      WRITE(40,10)
   10 FORMAT(T1,'C   SURFACE SPECIFICATIONS')
      WRITE(40,20)
   20 FORMAT(T1,'C')
      DO 70 SURFCOUNT=1,(SN-1)
         IF (SURFTYPESPEC(SURFCOUNT).NE.'P ') THEN
            IF ((SURFCOUNT.EQ.PVOUTERSURF).OR.
     c      (SURFCOUNT.EQ.SYSTEMSOUTH).OR.
     c      (SURFCOUNT.EQ.SYSTEMWEST).OR.
     c      (SURFCOUNT.EQ.SYSTEMTOP).OR.
     c      (SURFCOUNT.EQ.SYSTEMBOTTOM)) THEN
               WRITE(40,30) SURFCOUNT, SURFTYPESPEC(SURFCOUNT),
     c         SURFVALUESPEC(SURFCOUNT)
   30          FORMAT(T1,I4,'*',T8,A2,T13,F11.6)
            ELSE
```

```
                WRITE(40,40) SURFCOUNT, SURFTYPESPEC(SURFCOUNT),
     c          SURFVALUESPEC(SURFCOUNT)
 40             FORMAT(T1,I4,T8,A2,T13,F11.6)
             ENDIF
          ELSEIF (SURFTYPESPEC(SURFCOUNT).EQ.'P ') THEN
             IF ((SURFCOUNT.EQ.PVOUTERSURF).OR.
     c       (SURFCOUNT.EQ.SYSTEMSOUTH).OR.
     c       (SURFCOUNT.EQ.SYSTEMWEST).OR.
     c       (SURFCOUNT.EQ.SYSTEMTOP).OR.
     c       (SURFCOUNT.EQ.SYSTEMBOTTOM)) THEN
                WRITE(40,50) SURFCOUNT, SURFTYPESPEC(SURFCOUNT)
 50             FORMAT(T1,I4,'*',T8,A2,T13,'1 -1 0 0')
             ELSEIF (SURFVALUESPEC(SURFCOUNT).EQ.8000) THEN
                NPLOWDEG=NPLOWDEG*(PI/180)
                A=1/(TAN(NPLOWDEG))
                WRITE(40,60) SURFCOUNT, SURFTYPESPEC(SURFCOUNT), A
 60             FORMAT(T1,I4,T8,A2,T13,F8.4,1X,'-1 0 0')
             ELSEIF (SURFVALUESPEC(SURFCOUNT).EQ.8010) THEN
                NPHIGHDEG=NPHIGHDEG*(PI/180)
                A=1/(TAN(NPHIGHDEG))
                WRITE(40,80) SURFCOUNT, SURFTYPESPEC(SURFCOUNT), A
 80             FORMAT(T1,I4,T8,A2,T13,F8.4,1X,'-1 0 0')
             ELSEIF (SURFVALUESPEC(SURFCOUNT).EQ.8020) THEN
                A=1/(TAN(NPLOWDEG))
                WRITE(40,90) SURFCOUNT, SURFTYPESPEC(SURFCOUNT), A
 90             FORMAT(T1,I4,T8,A2,T13,F8.4,1X,'1 0 0')
             ELSEIF (SURFVALUESPEC(SURFCOUNT).EQ.8030) THEN
                A=1/(TAN(NPHIGHDEG))
                WRITE(40,100) SURFCOUNT, SURFTYPESPEC(SURFCOUNT), A
 100            FORMAT(T1,I4,T8,A2,T13,F8.4,1X,'1 0 0')
             ENDIF
          ENDIF
 70   CONTINUE
      CLOSE(UNIT=40)

      RETURN
      END

*************************************************************
*     This subroutine writes the control and initial source *
*     specification for this MCNP calculation.              *
*************************************************************
      SUBROUTINE CONTROL (NPERCYC, TOTCYCS, GARBCYCS,
     c PITCH, ASSYPITCH, NUMOFNODES, NODEHEIGHT,
     c BANDW, WESTINGHOUSE, CE, EIGHTH, QUARTER, FULL,
     c PREFIX, CYCLE, EFPD, CONTFILE)
*
      INTEGER NUMOFNODES, COLUMN, ROW, NUMSTPT1, NUMSTPT2,
     c NUMSTPT3, NODE, NPERCYC, TOTCYCS, GARBCYCS
*
      REAL PITCH, ASSYPITCH, NODEHEIGHT(50), EFPD, TOTALHEIGHT,
     c XVAL, YVAL, ZVAL, TH
*
      CHARACTER PREFIX*3, CYCLE*2, CHSTPT1*1,
```

```
     c CHSTPT2*1, CHSTPT3*1, CONTFILE*15
*
      LOGICAL BANDW, WESTINGHOUSE, CE, EIGHTH, QUARTER, FULL
*
*  Open the file to contain the control and source specifications.
      CONTFILE(1:3)=PREFIX
      CONTFILE(4:4)='C'
      CONTFILE(5:6)=CYCLE
      CONTFILE(7:7)='T'
      NUMSTPT1=INT(EFPD/100.0)
      CHSTPT1=CHAR(NUMSTPT1+48)
      NUMSTPT2=INT((EFPD-(NUMSTPT1*100))/10.0)
      CHSTPT2=CHAR(NUMSTPT2+48)
      NUMSTPT3=INT(EFPD-(NUMSTPT1*100)-
     c (NUMSTPT2*10))
      CHSTPT3=CHAR(NUMSTPT3+48)
      CONTFILE(8:8)=CHSTPT1
      CONTFILE(9:9)=CHSTPT2
      CONTFILE(10:10)=CHSTPT3
      CONTFILE(11:15)='.cont'
      OPEN(UNIT=230, FILE=CONTFILE, STATUS='UNKNOWN')
      REWIND(UNIT=230)
*
*  Write the control card specifications
      WRITE(230,1)
    1 FORMAT(T1,'C')
      WRITE(230,2)
    2 FORMAT(T1,'C   CONTROL CARD SPECIFICATIONS')
      WRITE(230,3)
    3 FORMAT(T1,'C')
      WRITE(230,10)
   10 FORMAT(T1,'MODE N')
      WRITE(230,20) NPERCYC, GARBCYCS, TOTCYCS
   20 FORMAT(T1,'KCODE  ',I6,3X,'1.0',3X,I3,3X,I6)
*
*  Write the source specifications
      WRITE(230,30)
   30 FORMAT(T1,'C')
      WRITE(230,40)
   40 FORMAT(T1,'C   INITIAL SOURCE SPECIFICATIONS')
      WRITE(230,50)
   50 FORMAT(T1,'C')
      WRITE(230,60)
   60 FORMAT(T1,'KSRC',T8,$)
      TOTALHEIGHT=0.0
      DO 70 NODE=1,NUMOFNODES
         TOTALHEIGHT=TOTALHEIGHT+NODEHEIGHT(NODE)
   70 CONTINUE
      TH=TOTALHEIGHT
      IF ((BANDW.EQ..TRUE.).AND.(EIGHTH.EQ..TRUE.)) THEN
         DO 130 ROW=1,6
            IF (ROW.LE.3) THEN
               DO 90 COLUMN=ROW,8
                  TH=TOTALHEIGHT
```

```
                   DO 80 NODE=1,NUMOFNODES
                      XVAL=((COLUMN-1)*ASSYPITCH)+PITCH
                      YVAL=((ROW-1)*ASSYPITCH)+PITCH
                      ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                      TH=TH-NODEHEIGHT(NODE)
                      WRITE(230,72) XVAL, YVAL, ZVAL
72                    FORMAT(F7.3,' ',F7.3,' ',F7.3,' ',$)
                      IF ((MOD(NODE,2).EQ.0).AND.
     c                (NODE.NE.NUMOFNODES)) THEN
                         WRITE(230,*)
                         WRITE(230,75)
75                       FORMAT(T8,$)
                      ENDIF
                      IF (NODE.EQ.NUMOFNODES) THEN
                         WRITE(230,*)
                         WRITE(230,75)
                      ENDIF
80                 CONTINUE
90              CONTINUE
             ELSEIF ((ROW.GT.3).AND.(ROW.LE.5)) THEN
                DO 110 COLUMN=ROW,7
                   TH=TOTALHEIGHT
                   DO 100 NODE=1,NUMOFNODES
                      XVAL=((COLUMN-1)*ASSYPITCH)+PITCH
                      YVAL=((ROW-1)*ASSYPITCH)+PITCH
                      ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                      TH=TH-NODEHEIGHT(NODE)
                      WRITE(230,92) XVAL, YVAL, ZVAL
92                    FORMAT(F7.3,' ',F7.3,' ',F7.3,' ',$)
                      IF ((MOD(NODE,2).EQ.0).AND.
     c                (NODE.NE.NUMOFNODES)) THEN
                         WRITE(230,*)
                         WRITE(230,95)
95                       FORMAT(T8,$)
                      ENDIF
                      IF (NODE.EQ.NUMOFNODES) THEN
                         WRITE(230,*)
                         WRITE(230,95)
                      ENDIF
100                CONTINUE
110             CONTINUE
             ELSEIF (ROW.EQ.6) THEN
                DO 125 COLUMN=ROW,6
                   TH=TOTALHEIGHT
                   DO 120 NODE=1,NUMOFNODES
                      XVAL=((COLUMN-1)*ASSYPITCH)+PITCH
                      YVAL=((ROW-1)*ASSYPITCH)+PITCH
                      ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                      TH=TH-NODEHEIGHT(NODE)
                      WRITE(230,112) XVAL, YVAL, ZVAL
112                   FORMAT(F7.3,' ',F7.3,' ',F7.3,' ',$)
                      IF (((MOD(NODE,2).EQ.0).AND.
     c                (NODE.NE.NUMOFNODES)).OR.((NODE.EQ.NUMOFNODES)
     c                .AND.(COLUMN.NE.6))) THEN
```

```
                        WRITE(230,*)
                        WRITE(230,115)
 115                    FORMAT(T8,$)
                     ENDIF
                     IF ((NODE.EQ.NUMOFNODES).AND.(COLUMN.EQ.6)) THEN
                        WRITE(230,*)
                     ENDIF
 120              CONTINUE
 125           CONTINUE
            ENDIF
 130     CONTINUE
         WRITE(230,132)
 132     FORMAT(T1,'PRINT')
         WRITE(230,*)
         WRITE(230,*)
      ELSEIF ((BANDW.EQ..TRUE.).AND.(QUARTER.EQ..TRUE.)) THEN
         DO 250 ROW=1,8
            IF (ROW.LE.3) THEN
               DO 160 COLUMN=1,8
                  TH=TOTALHEIGHT
                  DO 150 NODE=1,NUMOFNODES
                     XVAL=((COLUMN-1)*ASSYPITCH)+PITCH
                     YVAL=((ROW-1)*ASSYPITCH)+PITCH
                     ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                     TH=TH-NODEHEIGHT(NODE)
                     WRITE(230,142) XVAL, YVAL, ZVAL
 142                 FORMAT(F7.3,' ',F7.3,' ',F7.3,' ',$)
                     IF ((MOD(NODE,2).EQ.0).AND.
     c               (NODE.NE.NUMOFNODES)) THEN
                        WRITE(230,*)
                        WRITE(230,145)
 145                    FORMAT(T8,$)
                     ENDIF
                     IF (NODE.EQ.NUMOFNODES) THEN
                        WRITE(230,*)
                        WRITE(230,145)
                     ENDIF
 150              CONTINUE
 160           CONTINUE
            ELSEIF ((ROW.GT.3).AND.(ROW.LE.5)) THEN
               DO 180 COLUMN=1,7
                  TH=TOTALHEIGHT
                  DO 170 NODE=1,NUMOFNODES
                     XVAL=((COLUMN-1)*ASSYPITCH)+PITCH
                     YVAL=((ROW-1)*ASSYPITCH)+PITCH
                     ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                     TH=TH-NODEHEIGHT(NODE)
                     WRITE(230,162) XVAL, YVAL, ZVAL
 162                 FORMAT(F7.3,' ',F7.3,' ',F7.3,' ',$)
                     IF ((MOD(NODE,2).EQ.0).AND.
     c               (NODE.NE.NUMOFNODES)) THEN
                        WRITE(230,*)
                        WRITE(230,165)
 165                    FORMAT(T8,$)
```

```
                  ENDIF
                  IF (NODE.EQ.NUMOFNODES) THEN
                     WRITE(230,*)
                     WRITE(230,165)
                  ENDIF
170            CONTINUE
180         CONTINUE
         ELSEIF (ROW.EQ.6) THEN
            DO 200 COLUMN=1,6
               TH=TOTALHEIGHT
               DO 190 NODE=1,NUMOFNODES
                  XVAL=((COLUMN-1)*ASSYPITCH)+PITCH
                  YVAL=((ROW-1)*ASSYPITCH)+PITCH
                  ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                  TH=TH-NODEHEIGHT(NODE)
                  WRITE(230,182) XVAL, YVAL, ZVAL
182               FORMAT(F7.3,' ',F7.3,' ',F7.3,' ',$)
                  IF ((MOD(NODE,2).EQ.0).AND.
     c            (NODE.NE.NUMOFNODES)) THEN
                     WRITE(230,*)
                     WRITE(230,185)
185                  FORMAT(T8,$)
                  ENDIF
                  IF (NODE.EQ.NUMOFNODES) THEN
                     WRITE(230,*)
                     WRITE(230,185)
                  ENDIF
190            CONTINUE
200         CONTINUE
         ELSEIF (ROW.EQ.7) THEN
            DO 220 COLUMN=1,5
               TH=TOTALHEIGHT
               DO 210 NODE=1,NUMOFNODES
                  XVAL=((COLUMN-1)*ASSYPITCH)+PITCH
                  YVAL=((ROW-1)*ASSYPITCH)+PITCH
                  ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                  TH=TH-NODEHEIGHT(NODE)
                  WRITE(230,202) XVAL, YVAL, ZVAL
202               FORMAT(F7.3,' ',F7.3,' ',F7.3,' ',$)
                  IF ((MOD(NODE,2).EQ.0).AND.
     c            (NODE.NE.NUMOFNODES)) THEN
                     WRITE(230,*)
                     WRITE(230,205)
205                  FORMAT(T8,$)
                  ENDIF
                  IF (NODE.EQ.NUMOFNODES) THEN
                     WRITE(230,*)
                     WRITE(230,205)
                  ENDIF
210            CONTINUE
220         CONTINUE
         ELSEIF (ROW.EQ.8) THEN
            DO 240 COLUMN=1,3
               TH=TOTALHEIGHT
```

```
                 DO 230 NODE=1,NUMOFNODES
                    XVAL=((COLUMN-1)*ASSYPITCH)+PITCH
                    YVAL=((ROW-1)*ASSYPITCH)+PITCH
                    ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                    TH=TH-NODEHEIGHT(NODE)
                    WRITE(230,222) XVAL, YVAL, ZVAL
222                 FORMAT(F7.3,' ',F7.3,' ',F7.3,' ',$)
                    IF (((MOD(NODE,2).EQ.0).AND.
     c              (NODE.NE.NUMOFNODES)).OR.((NODE.EQ.NUMOFNODES)
     c              .AND.(COLUMN.NE.3))) THEN
                       WRITE(230,*)
                       WRITE(230,225)
225                    FORMAT(T8,$)
                    ENDIF
                    IF ((NODE.EQ.NUMOFNODES).AND.
     c              (COLUMN.EQ.3)) THEN
                       WRITE(230,*)
                    ENDIF
230              CONTINUE
240           CONTINUE
           ENDIF
250     CONTINUE
        WRITE(230,252)
252     FORMAT(T1,'PRINT')
        WRITE(230,*)
        WRITE(230,*)
      ELSEIF ((BANDW.EQ..TRUE.).AND.(FULL.EQ..TRUE.)) THEN
        DO 374 ROW=1,15
           IF (ROW.EQ.1) THEN
              DO 260 COLUMN=6,10
                 TH=TOTALHEIGHT
                 DO 258 NODE=1,NUMOFNODES
                    XVAL=((COLUMN-8)*ASSYPITCH)+PITCH
                    YVAL=(7*ASSYPITCH)+PITCH
                    ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                    TH=TH-NODEHEIGHT(NODE)
                    WRITE(230,254) XVAL, YVAL, ZVAL
254                 FORMAT(F8.3,' ',F8.3,' ',F8.3,' ',$)
                    IF ((MOD(NODE,2).EQ.0).AND.
     c              (NODE.NE.NUMOFNODES)) THEN
                       WRITE(230,*)
                       WRITE(230,256)
256                    FORMAT(T8,$)
                    ENDIF
                    IF (NODE.EQ.NUMOFNODES) THEN
                       WRITE(230,*)
                       WRITE(230,256)
                    ENDIF
258              CONTINUE
260           CONTINUE
           ELSEIF (ROW.EQ.2) THEN
              DO 268 COLUMN=4,12
                 TH=TOTALHEIGHT
                 DO 266 NODE=1,NUMOFNODES
```

```
                  XVAL=((COLUMN-8)*ASSYPITCH)+PITCH
                  YVAL=(6*ASSYPITCH)+PITCH
                  ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                  TH=TH-NODEHEIGHT(NODE)
                  WRITE(230,262) XVAL, YVAL, ZVAL
 262              FORMAT(F8.3,' ',F8.3,' ',F8.3,' ',$)
                  IF ((MOD(NODE,2).EQ.0).AND.
     c            (NODE.NE.NUMOFNODES)) THEN
                     WRITE(230,*)
                     WRITE(230,264)
 264                 FORMAT(T8,$)
                  ENDIF
                  IF (NODE.EQ.NUMOFNODES) THEN
                     WRITE(230,*)
                     WRITE(230,264)
                  ENDIF
 266           CONTINUE
 268        CONTINUE
         ELSEIF (ROW.EQ.3) THEN
            DO 276 COLUMN=3,13
               TH=TOTALHEIGHT
               DO 274 NODE=1,NUMOFNODES
                  XVAL=((COLUMN-8)*ASSYPITCH)+PITCH
                  YVAL=(5*ASSYPITCH)+PITCH
                  ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                  TH=TH-NODEHEIGHT(NODE)
                  WRITE(230,270) XVAL, YVAL, ZVAL
 270              FORMAT(F8.3,' ',F8.3,' ',F8.3,' ',$)
                  IF ((MOD(NODE,2).EQ.0).AND.
     c            (NODE.NE.NUMOFNODES)) THEN
                     WRITE(230,*)
                     WRITE(230,272)
 272                 FORMAT(T8,$)
                  ENDIF
                  IF (NODE.EQ.NUMOFNODES) THEN
                     WRITE(230,*)
                     WRITE(230,272)
                  ENDIF
 274           CONTINUE
 276        CONTINUE
         ELSEIF (ROW.EQ.4) THEN
            DO 284 COLUMN=2,14
               TH=TOTALHEIGHT
               DO 282 NODE=1,NUMOFNODES
                  XVAL=((COLUMN-8)*ASSYPITCH)+PITCH
                  YVAL=(4*ASSYPITCH)+PITCH
                  ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                  TH=TH-NODEHEIGHT(NODE)
                  WRITE(230,278) XVAL, YVAL, ZVAL
 278              FORMAT(F8.3,' ',F8.3,' ',F8.3,' ',$)
                  IF ((MOD(NODE,2).EQ.0).AND.
     c            (NODE.NE.NUMOFNODES)) THEN
                     WRITE(230,*)
                     WRITE(230,280)
```

```
280                 FORMAT(T8,$)
                 ENDIF
                 IF (NODE.EQ.NUMOFNODES) THEN
                    WRITE(230,*)
                    WRITE(230,280)
                 ENDIF
282           CONTINUE
284        CONTINUE
        ELSEIF (ROW.EQ.5) THEN
           DO 292 COLUMN=2,14
              TH=TOTALHEIGHT
              DO 290 NODE=1,NUMOFNODES
                 XVAL=((COLUMN-8)*ASSYPITCH)+PITCH
                 YVAL=(3*ASSYPITCH)+PITCH
                 ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                 TH=TH-NODEHEIGHT(NODE)
                 WRITE(230,286) XVAL, YVAL, ZVAL
286              FORMAT(F8.3,' ',F8.3,' ',F8.3,' ',$)
                 IF ((MOD(NODE,2).EQ.0).AND.
     c           (NODE.NE.NUMOFNODES)) THEN
                    WRITE(230,*)
                    WRITE(230,288)
288                 FORMAT(T8,$)
                 ENDIF
                 IF (NODE.EQ.NUMOFNODES) THEN
                    WRITE(230,*)
                    WRITE(230,288)
                 ENDIF
290           CONTINUE
292        CONTINUE
        ELSEIF (ROW.EQ.6) THEN
           DO 300 COLUMN=1,15
              TH=TOTALHEIGHT
              DO 298 NODE=1,NUMOFNODES
                 XVAL=((COLUMN-8)*ASSYPITCH)+PITCH
                 YVAL=(2*ASSYPITCH)+PITCH
                 ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                 TH=TH-NODEHEIGHT(NODE)
                 WRITE(230,294) XVAL, YVAL, ZVAL
294              FORMAT(F8.3,' ',F8.3,' ',F8.3,' ',$)
                 IF ((MOD(NODE,2).EQ.0).AND.
     c           (NODE.NE.NUMOFNODES)) THEN
                    WRITE(230,*)
                    WRITE(230,296)
296                 FORMAT(T8,$)
                 ENDIF
                 IF (NODE.EQ.NUMOFNODES) THEN
                    WRITE(230,*)
                    WRITE(230,296)
                 ENDIF
298           CONTINUE
300        CONTINUE
        ELSEIF (ROW.EQ.7) THEN
           DO 308 COLUMN=1,15
```

```
                 TH=TOTALHEIGHT
                 DO 306 NODE=1,NUMOFNODES
                    XVAL=((COLUMN-8)*ASSYPITCH)+PITCH
                    YVAL=(1*ASSYPITCH)+PITCH
                    ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                    TH=TH-NODEHEIGHT(NODE)
                    WRITE(230,302) XVAL, YVAL, ZVAL
 302                FORMAT(F8.3,' ',F8.3,' ',F8.3,' ',$)
                    IF ((MOD(NODE,2).EQ.0).AND.
     c              (NODE.NE.NUMOFNODES)) THEN
                       WRITE(230,*)
                       WRITE(230,304)
 304                   FORMAT(T8,$)
                    ENDIF
                    IF (NODE.EQ.NUMOFNODES) THEN
                       WRITE(230,*)
                       WRITE(230,304)
                    ENDIF
 306             CONTINUE
 308          CONTINUE
           ELSEIF (ROW.EQ.8) THEN
              DO 316 COLUMN=1,15
                 TH=TOTALHEIGHT
                 DO 314 NODE=1,NUMOFNODES
                    XVAL=((COLUMN-8)*ASSYPITCH)+PITCH
                    YVAL=(0*ASSYPITCH)+PITCH
                    ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                    TH=TH-NODEHEIGHT(NODE)
                    WRITE(230,310) XVAL, YVAL, ZVAL
 310                FORMAT(F8.3,' ',F8.3,' ',F8.3,' ',$)
                    IF ((MOD(NODE,2).EQ.0).AND.
     c              (NODE.NE.NUMOFNODES)) THEN
                       WRITE(230,*)
                       WRITE(230,312)
 312                   FORMAT(T8,$)
                    ENDIF
                    IF (NODE.EQ.NUMOFNODES) THEN
                       WRITE(230,*)
                       WRITE(230,312)
                    ENDIF
 314             CONTINUE
 316          CONTINUE
           ELSEIF (ROW.EQ.9) THEN
              DO 324 COLUMN=1,15
                 TH=TOTALHEIGHT
                 DO 322 NODE=1,NUMOFNODES
                    XVAL=((COLUMN-8)*ASSYPITCH)+PITCH
                    YVAL=(-1*ASSYPITCH)+PITCH
                    ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                    TH=TH-NODEHEIGHT(NODE)
                    WRITE(230,318) XVAL, YVAL, ZVAL
 318                FORMAT(F8.3,' ',F8.3,' ',F8.3,' ',$)
                    IF ((MOD(NODE,2).EQ.0).AND.
     c              (NODE.NE.NUMOFNODES)) THEN
```

```
                    WRITE(230,*)
                    WRITE(230,320)
320                 FORMAT(T8,$)
                 ENDIF
                 IF (NODE.EQ.NUMOFNODES) THEN
                    WRITE(230,*)
                    WRITE(230,320)
                 ENDIF
322           CONTINUE
324        CONTINUE
        ELSEIF (ROW.EQ.10) THEN
           DO 332 COLUMN=1,15
              TH=TOTALHEIGHT
              DO 330 NODE=1,NUMOFNODES
                 XVAL=((COLUMN-8)*ASSYPITCH)+PITCH
                 YVAL=(-2*ASSYPITCH)+PITCH
                 ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                 TH=TH-NODEHEIGHT(NODE)
                 WRITE(230,326) XVAL, YVAL, ZVAL
326              FORMAT(F8.3,' ',F8.3,' ',F8.3,' ',$)
                 IF ((MOD(NODE,2).EQ.0).AND.
     c           (NODE.NE.NUMOFNODES)) THEN
                    WRITE(230,*)
                    WRITE(230,328)
328                 FORMAT(T8,$)
                 ENDIF
                 IF (NODE.EQ.NUMOFNODES) THEN
                    WRITE(230,*)
                    WRITE(230,328)
                 ENDIF
330           CONTINUE
332        CONTINUE
        ELSEIF (ROW.EQ.11) THEN
           DO 340 COLUMN=2,14
              TH=TOTALHEIGHT
              DO 338 NODE=1,NUMOFNODES
                 XVAL=((COLUMN-8)*ASSYPITCH)+PITCH
                 YVAL=(-3*ASSYPITCH)+PITCH
                 ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                 TH=TH-NODEHEIGHT(NODE)
                 WRITE(230,334) XVAL, YVAL, ZVAL
334              FORMAT(F8.3,' ',F8.3,' ',F8.3,' ',$)
                 IF ((MOD(NODE,2).EQ.0).AND.
     c           (NODE.NE.NUMOFNODES)) THEN
                    WRITE(230,*)
                    WRITE(230,336)
336                 FORMAT(T8,$)
                 ENDIF
                 IF (NODE.EQ.NUMOFNODES) THEN
                    WRITE(230,*)
                    WRITE(230,336)
                 ENDIF
338           CONTINUE
340        CONTINUE
```

```
          ELSEIF (ROW.EQ.12) THEN
             DO 348 COLUMN=2,14
                TH=TOTALHEIGHT
                DO 346 NODE=1,NUMOFNODES
                   XVAL=((COLUMN-8)*ASSYPITCH)+PITCH
                   YVAL=(-4*ASSYPITCH)+PITCH
                   ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                   TH=TH-NODEHEIGHT(NODE)
                   WRITE(230,342) XVAL, YVAL, ZVAL
342                FORMAT(F8.3,' ',F8.3,' ',F8.3,' ',$)
                   IF ((MOD(NODE,2).EQ.0).AND.
     c             (NODE.NE.NUMOFNODES)) THEN
                      WRITE(230,*)
                      WRITE(230,344)
344                   FORMAT(T8,$)
                   ENDIF
                   IF (NODE.EQ.NUMOFNODES) THEN
                      WRITE(230,*)
                      WRITE(230,344)
                   ENDIF
346             CONTINUE
348          CONTINUE
          ELSEIF (ROW.EQ.13) THEN
             DO 356 COLUMN=3,13
                TH=TOTALHEIGHT
                DO 354 NODE=1,NUMOFNODES
                   XVAL=((COLUMN-8)*ASSYPITCH)+PITCH
                   YVAL=(-5*ASSYPITCH)+PITCH
                   ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                   TH=TH-NODEHEIGHT(NODE)
                   WRITE(230,350) XVAL, YVAL, ZVAL
350                FORMAT(F8.3,' ',F8.3,' ',F8.3,' ',$)
                   IF ((MOD(NODE,2).EQ.0).AND.
     c             (NODE.NE.NUMOFNODES)) THEN
                      WRITE(230,*)
                      WRITE(230,352)
352                   FORMAT(T8,$)
                   ENDIF
                   IF (NODE.EQ.NUMOFNODES) THEN
                      WRITE(230,*)
                      WRITE(230,352)
                   ENDIF
354             CONTINUE
356          CONTINUE
          ELSEIF (ROW.EQ.14) THEN
             DO 364 COLUMN=4,12
                TH=TOTALHEIGHT
                DO 362 NODE=1,NUMOFNODES
                   XVAL=((COLUMN-8)*ASSYPITCH)+PITCH
                   YVAL=(-6*ASSYPITCH)+PITCH
                   ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                   TH=TH-NODEHEIGHT(NODE)
                   WRITE(230,358) XVAL, YVAL, ZVAL
358                FORMAT(F8.3,' ',F8.3,' ',F8.3,' ',$)
```

```
                      IF ((MOD(NODE,2).EQ.0).AND.
     c                (NODE.NE.NUMOFNODES)) THEN
                         WRITE(230,*)
                         WRITE(230,360)
 360                     FORMAT(T8,$)
                      ENDIF
                      IF (NODE.EQ.NUMOFNODES) THEN
                         WRITE(230,*)
                         WRITE(230,360)
                      ENDIF
 362              CONTINUE
 364            CONTINUE
             ELSEIF (ROW.EQ.15) THEN
                DO 372 COLUMN=6,10
                   TH=TOTALHEIGHT
                   DO 370 NODE=1,NUMOFNODES
                      XVAL=((COLUMN-8)*ASSYPITCH)+PITCH
                      YVAL=(-7*ASSYPITCH)+PITCH
                      ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                      TH=TH-NODEHEIGHT(NODE)
                      WRITE(230,366) XVAL, YVAL, ZVAL
 366                  FORMAT(F8.3,' ',F8.3,' ',F8.3,' ',$)
                      IF (((MOD(NODE,2).EQ.0).AND.
     c                (NODE.NE.NUMOFNODES)).OR.((NODE.EQ.NUMOFNODES)
     c                .AND.(COLUMN.NE.10))) THEN
                         WRITE(230,*)
                         WRITE(230,368)
 368                     FORMAT(T8,$)
                      ENDIF
                      IF ((NODE.EQ.NUMOFNODES).AND.(COLUMN.EQ.10)) THEN
                         WRITE(230,*)
                      ENDIF
 370               CONTINUE
 372            CONTINUE
             ENDIF
 374      CONTINUE
          WRITE(230,376)
 376      FORMAT(T1,'PRINT')
          WRITE(230,*)
          WRITE(230,*)
       ELSEIF ((WESTINGHOUSE.EQ..TRUE.).AND.
     c (EIGHTH.EQ..TRUE.)) THEN
          DO 460 ROW=1,6
             IF (ROW.LE.4) THEN
                DO 410 COLUMN=ROW,8
                   TH=TOTALHEIGHT
                   DO 400 NODE=1,NUMOFNODES
                      XVAL=((COLUMN-1)*ASSYPITCH)+(2*PITCH)
                      YVAL=((ROW-1)*ASSYPITCH)+PITCH
                      ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                      TH=TH-NODEHEIGHT(NODE)
                      WRITE(230,380) XVAL, YVAL, ZVAL
 380                  FORMAT(F7.3,' ',F7.3,' ',F7.3,' ',$)
                      IF ((MOD(NODE,2).EQ.0).AND.
```

```
     c            (NODE.NE.NUMOFNODES)) THEN
                     WRITE(230,*)
                     WRITE(230,390)
 390                 FORMAT(T8,$)
                  ENDIF
                  IF (NODE.EQ.NUMOFNODES) THEN
                     WRITE(230,*)
                     WRITE(230,390)
                  ENDIF
 400           CONTINUE
 410        CONTINUE
         ELSEIF ((ROW.GT.4).AND.(ROW.LE.6)) THEN
            DO 450 COLUMN=ROW,7
               TH=TOTALHEIGHT
               DO 440 NODE=1,NUMOFNODES
                  XVAL=((COLUMN-1)*ASSYPITCH)+(2*PITCH)
                  YVAL=((ROW-1)*ASSYPITCH)+PITCH
                  ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                  TH=TH-NODEHEIGHT(NODE)
                  WRITE(230,420) XVAL, YVAL, ZVAL
 420              FORMAT(F7.3,' ',F7.3,' ',F7.3,' ',$)
                  IF ((MOD(NODE,2).EQ.0).AND.
     c            (NODE.NE.NUMOFNODES)) THEN
                     WRITE(230,*)
                     WRITE(230,430)
 430                 FORMAT(T8,$)
                  ENDIF
                  IF ((NODE.EQ.NUMOFNODES).AND.(ROW.LT.6)
     c            .AND.(COLUMN.LT.7)) THEN
                     WRITE(230,*)
                     WRITE(230,430)
                  ELSEIF ((NODE.EQ.NUMOFNODES).AND.
     c            (ROW.EQ.6).AND.(COLUMN.EQ.7)) THEN
                     WRITE(230,*)
                  ENDIF
 440           CONTINUE
 450        CONTINUE
         ENDIF
 460  CONTINUE
      WRITE(230,470)
 470  FORMAT(T1,'PRINT')
      WRITE(230,*)
      WRITE(230,*)
   ELSEIF ((WESTINGHOUSE.EQ..TRUE.).AND.
     c (QUARTER.EQ..TRUE.)) THEN
      DO 640 ROW=1,8
         IF (ROW.LE.4) THEN
            DO 510 COLUMN=1,8
               TH=TOTALHEIGHT
               DO 500 NODE=1,NUMOFNODES
                  XVAL=((COLUMN-1)*ASSYPITCH)+PITCH
                  YVAL=((ROW-1)*ASSYPITCH)+PITCH
                  ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                  TH=TH-NODEHEIGHT(NODE)
```

```
                    WRITE(230,480) XVAL, YVAL, ZVAL
 480                FORMAT(F7.3,' ',F7.3,' ',F7.3,' ',$)
                    IF ((MOD(NODE,2).EQ.0).AND.
     c              (NODE.NE.NUMOFNODES)) THEN
                       WRITE(230,*)
                       WRITE(230,490)
 490                   FORMAT(T8,$)
                    ENDIF
                    IF (NODE.EQ.NUMOFNODES) THEN
                       WRITE(230,*)
                       WRITE(230,490)
                    ENDIF
 500             CONTINUE
 510          CONTINUE
           ELSEIF ((ROW.GT.4).AND.(ROW.LE.6)) THEN
              DO 550 COLUMN=1,7
                 TH=TOTALHEIGHT
                 DO 540 NODE=1,NUMOFNODES
                    XVAL=((COLUMN-1)*ASSYPITCH)+PITCH
                    YVAL=((ROW-1)*ASSYPITCH)+PITCH
                    ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                    TH=TH-NODEHEIGHT(NODE)
                    WRITE(230,520) XVAL, YVAL, ZVAL
 520                FORMAT(F7.3,' ',F7.3,' ',F7.3,' ',$)
                    IF ((MOD(NODE,2).EQ.0).AND.
     c              (NODE.NE.NUMOFNODES)) THEN
                       WRITE(230,*)
                       WRITE(230,530)
 530                   FORMAT(T8,$)
                    ENDIF
                    IF (NODE.EQ.NUMOFNODES) THEN
                       WRITE(230,*)
                       WRITE(230,530)
                    ENDIF
 540             CONTINUE
 550          CONTINUE
           ELSEIF (ROW.EQ.7) THEN
              DO 590 COLUMN=1,6
                 TH=TOTALHEIGHT
                 DO 580 NODE=1,NUMOFNODES
                    XVAL=((COLUMN-1)*ASSYPITCH)+PITCH
                    YVAL=((ROW-1)*ASSYPITCH)+PITCH
                    ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                    TH=TH-NODEHEIGHT(NODE)
                    WRITE(230,560) XVAL, YVAL, ZVAL
 560                FORMAT(F7.3,' ',F7.3,' ',F7.3,' ',$)
                    IF ((MOD(NODE,2).EQ.0).AND.
     c              (NODE.NE.NUMOFNODES)) THEN
                       WRITE(230,*)
                       WRITE(230,570)
 570                   FORMAT(T8,$)
                    ENDIF
                    IF (NODE.EQ.NUMOFNODES) THEN
                       WRITE(230,*)
```

```
                          WRITE(230,570)
                       ENDIF
 580                CONTINUE
 590             CONTINUE
              ELSEIF (ROW.EQ.8) THEN
                 DO 630 COLUMN=1,4
                    TH=TOTALHEIGHT
                    DO 620 NODE=1,NUMOFNODES
                       XVAL=((COLUMN-1)*ASSYPITCH)+PITCH
                       YVAL=((ROW-1)*ASSYPITCH)+PITCH
                       ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                       TH=TH-NODEHEIGHT(NODE)
                       WRITE(230,600) XVAL, YVAL, ZVAL
 600                   FORMAT(F7.3,' ',F7.3,' ',F7.3,' ',$)
                       IF (((MOD(NODE,2).EQ.0).AND.
     c                 (NODE.NE.NUMOFNODES)).OR.((NODE.EQ.NUMOFNODES)
     c                 .AND.(COLUMN.NE.4))) THEN
                          WRITE(230,*)
                          WRITE(230,610)
 610                      FORMAT(T8,$)
                       ENDIF
                       IF ((NODE.EQ.NUMOFNODES).AND.(COLUMN.EQ.4)) THEN
                          WRITE(230,*)
                       ENDIF
 620                CONTINUE
 630             CONTINUE
              ENDIF
 640       CONTINUE
           WRITE(230,650)
 650       FORMAT(T1,'PRINT')
           WRITE(230,*)
           WRITE(230,*)
        ELSEIF ((WESTINGHOUSE.EQ..TRUE.).AND.
     c (FULL.EQ..TRUE.)) THEN
           DO 1020 ROW=1,15
              IF (ROW.EQ.1) THEN
                 DO 690 COLUMN=5,11
                    TH=TOTALHEIGHT
                    DO 680 NODE=1,NUMOFNODES
                       XVAL=((COLUMN-8)*ASSYPITCH)+PITCH
                       YVAL=(7*ASSYPITCH)+PITCH
                       ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                       TH=TH-NODEHEIGHT(NODE)
                       WRITE(230,660) XVAL, YVAL, ZVAL
 660                   FORMAT(F8.3,' ',F8.3,' ',F8.3,' ',$)
                       IF ((MOD(NODE,2).EQ.0).AND.
     c                 (NODE.NE.NUMOFNODES)) THEN
                          WRITE(230,*)
                          WRITE(230,670)
 670                      FORMAT(T8,$)
                       ENDIF
                       IF (NODE.EQ.NUMOFNODES) THEN
                          WRITE(230,*)
                          WRITE(230,670)
```

```
                    ENDIF
 680            CONTINUE
 690         CONTINUE
          ELSEIF (ROW.EQ.2) THEN
             DO 730 COLUMN=3,13
                TH=TOTALHEIGHT
                DO 720 NODE=1,NUMOFNODES
                   XVAL=((COLUMN-8)*ASSYPITCH)+PITCH
                   YVAL=(6*ASSYPITCH)+PITCH
                   ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                   TH=TH-NODEHEIGHT(NODE)
                   WRITE(230,700) XVAL, YVAL, ZVAL
 700               FORMAT(F8.3,' ',F8.3,' ',F8.3,' ',$)
                   IF ((MOD(NODE,2).EQ.0).AND.
     c             (NODE.NE.NUMOFNODES)) THEN
                      WRITE(230,*)
                      WRITE(230,710)
 710                  FORMAT(T8,$)
                   ENDIF
                   IF (NODE.EQ.NUMOFNODES) THEN
                      WRITE(230,*)
                      WRITE(230,710)
                   ENDIF
 720            CONTINUE
 730         CONTINUE
          ELSEIF (ROW.EQ.3) THEN
             DO 770 COLUMN=2,14
                TH=TOTALHEIGHT
                DO 760 NODE=1,NUMOFNODES
                   XVAL=((COLUMN-8)*ASSYPITCH)+PITCH
                   YVAL=(5*ASSYPITCH)+PITCH
                   ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                   TH=TH-NODEHEIGHT(NODE)
                   WRITE(230,740) XVAL, YVAL, ZVAL
 740               FORMAT(F8.3,' ',F8.3,' ',F8.3,' ',$)
                   IF ((MOD(NODE,2).EQ.0).AND.
     c             (NODE.NE.NUMOFNODES)) THEN
                      WRITE(230,*)
                      WRITE(230,750)
 750                  FORMAT(T8,$)
                   ENDIF
                   IF (NODE.EQ.NUMOFNODES) THEN
                      WRITE(230,*)
                      WRITE(230,750)
                   ENDIF
 760            CONTINUE
 770         CONTINUE
          ELSEIF (ROW.EQ.4) THEN
             DO 810 COLUMN=2,14
                TH=TOTALHEIGHT
                DO 800 NODE=1,NUMOFNODES
                   XVAL=((COLUMN-8)*ASSYPITCH)+PITCH
                   YVAL=(4*ASSYPITCH)+PITCH
                   ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
```

```
                TH=TH-NODEHEIGHT(NODE)
                WRITE(230,780) XVAL, YVAL, ZVAL
780             FORMAT(F8.3,' ',F8.3,' ',F8.3,' ',$)
                IF ((MOD(NODE,2).EQ.0).AND.
     c          (NODE.NE.NUMOFNODES)) THEN
                   WRITE(230,*)
                   WRITE(230,790)
790                FORMAT(T8,$)
                ENDIF
                IF (NODE.EQ.NUMOFNODES) THEN
                   WRITE(230,*)
                   WRITE(230,790)
                ENDIF
800          CONTINUE
810       CONTINUE
       ELSEIF ((ROW.GE.5).AND.(ROW.LE.11)) THEN
          DO 850 COLUMN=1,15
             TH=TOTALHEIGHT
             DO 840 NODE=1,NUMOFNODES
                XVAL=((COLUMN-8)*ASSYPITCH)+PITCH
                YVAL=(3*ASSYPITCH)+PITCH
                ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                TH=TH-NODEHEIGHT(NODE)
                WRITE(230,820) XVAL, YVAL, ZVAL
820             FORMAT(F8.3,' ',F8.3,' ',F8.3,' ',$)
                IF ((MOD(NODE,2).EQ.0).AND.
     c          (NODE.NE.NUMOFNODES)) THEN
                   WRITE(230,*)
                   WRITE(230,830)
830                FORMAT(T8,$)
                ENDIF
                IF (NODE.EQ.NUMOFNODES) THEN
                   WRITE(230,*)
                   WRITE(230,830)
                ENDIF
840          CONTINUE
850       CONTINUE
       ELSEIF (ROW.EQ.12) THEN
          DO 890 COLUMN=2,14
             TH=TOTALHEIGHT
             DO 880 NODE=1,NUMOFNODES
                XVAL=((COLUMN-8)*ASSYPITCH)+PITCH
                YVAL=(2*ASSYPITCH)+PITCH
                ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                TH=TH-NODEHEIGHT(NODE)
                WRITE(230,860) XVAL, YVAL, ZVAL
860             FORMAT(F8.3,' ',F8.3,' ',F8.3,' ',$)
                IF ((MOD(NODE,2).EQ.0).AND.
     c          (NODE.NE.NUMOFNODES)) THEN
                   WRITE(230,*)
                   WRITE(230,870)
870                FORMAT(T8,$)
                ENDIF
                IF (NODE.EQ.NUMOFNODES) THEN
```

```
                  WRITE(230,*)
                  WRITE(230,870)
               ENDIF
 880        CONTINUE
 890     CONTINUE
      ELSEIF (ROW.EQ.13) THEN
         DO 930 COLUMN=2,14
            TH=TOTALHEIGHT
            DO 920 NODE=1,NUMOFNODES
               XVAL=((COLUMN-8)*ASSYPITCH)+PITCH
               YVAL=(1*ASSYPITCH)+PITCH
               ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
               TH=TH-NODEHEIGHT(NODE)
               WRITE(230,900) XVAL, YVAL, ZVAL
 900           FORMAT(F8.3,' ',F8.3,' ',F8.3,' ',$)
               IF ((MOD(NODE,2).EQ.0).AND.
     c         (NODE.NE.NUMOFNODES)) THEN
                  WRITE(230,*)
                  WRITE(230,910)
 910              FORMAT(T8,$)
               ENDIF
               IF (NODE.EQ.NUMOFNODES) THEN
                  WRITE(230,*)
                  WRITE(230,910)
               ENDIF
 920        CONTINUE
 930     CONTINUE
      ELSEIF (ROW.EQ.14) THEN
         DO 970 COLUMN=3,13
            TH=TOTALHEIGHT
            DO 960 NODE=1,NUMOFNODES
               XVAL=((COLUMN-8)*ASSYPITCH)+PITCH
               YVAL=(0*ASSYPITCH)+PITCH
               ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
               TH=TH-NODEHEIGHT(NODE)
               WRITE(230,940) XVAL, YVAL, ZVAL
 940           FORMAT(F8.3,' ',F8.3,' ',F8.3,' ',$)
               IF ((MOD(NODE,2).EQ.0).AND.
     c         (NODE.NE.NUMOFNODES)) THEN
                  WRITE(230,*)
                  WRITE(230,950)
 950              FORMAT(T8,$)
               ENDIF
               IF (NODE.EQ.NUMOFNODES) THEN
                  WRITE(230,*)
                  WRITE(230,950)
               ENDIF
 960        CONTINUE
 970     CONTINUE
      ELSEIF (ROW.EQ.15) THEN
         DO 1010 COLUMN=5,11
            TH=TOTALHEIGHT
            DO 1000 NODE=1,NUMOFNODES
               XVAL=((COLUMN-8)*ASSYPITCH)+PITCH
```

```
                        YVAL=(-1*ASSYPITCH)+PITCH
                        ZVAL=TH-(NODEHEIGHT(NODE)/2.0)
                        TH=TH-NODEHEIGHT(NODE)
                        WRITE(230,980) XVAL, YVAL, ZVAL
 980                    FORMAT(F8.3,' ',F8.3,' ',F8.3,' ',$)
                        IF (((MOD(NODE,2).EQ.0).AND.
     c                  (NODE.NE.NUMOFNODES)).OR.((NODE.EQ.NUMOFNODES)
     c                  .AND.(COLUMN.NE.11))) THEN
                           WRITE(230,*)
                           WRITE(230,990)
 990                       FORMAT(T8,$)
                        ENDIF
                        IF ((NODE.EQ.NUMOFNODES).AND.(COLUMN.EQ.11)) THEN
                           WRITE(230,*)
                        ENDIF
1000                 CONTINUE
1010              CONTINUE
               ENDIF
1020        CONTINUE
            WRITE(230,1030)
1030        FORMAT(T1,'PRINT')
            WRITE(230,*)
            WRITE(230,*)
         ELSEIF (CE.EQ..TRUE.) THEN
            WRITE(*,*) 'THE CE SOURCE SPECIFICATION IS',
     c      ' NOT AVAILABLE AT THIS TIME.'
         ENDIF

         RETURN
         END

**************************************************************************
*****
         SUBROUTINE WESTONE(EIGHTH, QUARTER, FULL, ASSYID, NUMOFFADESIGNS,
     c   DESNUM, NUMOFBANKS, BANKDES, BANKID, BANKHEIGHT, BANKNUM,
     c   ENRICHMENT, STAT)
*
         INTEGER NUMOFFADESIGNS, DESNUM(50,50), NUMOFBANKS,
     c   BANKID(20), BANKNUM(50,50), C, R, BANK
*
         REAL BANKHEIGHT(20), ENRICHMENT(50,50)
*
         CHARACTER ASSYID(50,50)*5, STAT(50,50)*1, BANKDES(20)*5
*
         LOGICAL EIGHTH, QUARTER, FULL
*
            IF (EIGHTH.EQ..TRUE.) THEN
*        Read in the fuel assembly archive identifiers for retrieval of
isotopics.
               READ(15,10) ASSYID(1,1), ASSYID(2,1), ASSYID(3,1),
     c         ASSYID(4,1), ASSYID(5,1), ASSYID(6,1), ASSYID(7,1),
     c         ASSYID(8,1)
 10            FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c         A5,1X,A5)
```

```
      READ(15,20) ASSYID(2,2), ASSYID(3,2), ASSYID(4,2),
     c ASSYID(5,2), ASSYID(6,2), ASSYID(7,2), ASSYID(8,2)
20    FORMAT(T1,5X,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c A5,1X,A5)
      READ(15,30) ASSYID(3,3), ASSYID(4,3), ASSYID(5,3),
     c ASSYID(6,3), ASSYID(7,3), ASSYID(8,3)
30    FORMAT(T1,5X,1X,5X,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c A5,1X,A5)
      READ(15,40) ASSYID(4,4), ASSYID(5,4), ASSYID(6,4),
     c ASSYID(7,4), ASSYID(8,4)
40    FORMAT(T1,5X,1X,5X,1X,5X,1X,A5,1X,A5,1X,A5,1X,A5,
     c 1X,A5)
      READ(15,50) ASSYID(5,5), ASSYID(6,5), ASSYID(7,5)
50    FORMAT(T1,5X,1X,5X,1X,5X,1X,5X,1X,A5,1X,A5,1X,A5)
      READ(15,60) ASSYID(6,6), ASSYID(7,6)
60    FORMAT(T1,5X,1X,5X,1X,5X,1X,5X,1X,5X,1X,A5,1X,A5)
*     Read in the number of different fuel assembly designs to be included
*     in the MCNP calculation.
      READ(15,*) NUMOFFADESIGNS
*     Read in the fuel assembly relative design designations.
      READ(15,*) DESNUM(1,1), DESNUM(2,1), DESNUM(3,1),
     c DESNUM(4,1), DESNUM(5,1), DESNUM(6,1), DESNUM(7,1),
     c DESNUM(8,1)
      READ(15,*) DESNUM(2,2), DESNUM(3,2), DESNUM(4,2),
     c DESNUM(5,2), DESNUM(6,2), DESNUM(7,2), DESNUM(8,2)
      READ(15,*) DESNUM(3,3), DESNUM(4,3), DESNUM(5,3),
     c DESNUM(6,3), DESNUM(7,3), DESNUM(8,3)
      READ(15,*) DESNUM(4,4), DESNUM(5,4), DESNUM(6,4),
     c DESNUM(7,4), DESNUM(8,4)
      READ(15,*) DESNUM(5,5), DESNUM(6,5), DESNUM(7,5)
      READ(15,*) DESNUM(6,6), DESNUM(7,6)
      DO 80 C=1,50
         DO 70 R=1,50
            IF (DESNUM(C,R).GT.NUMOFFADESIGNS) THEN
               WRITE(*,*) 'THE FUEL ASSEMBLY DESIGN NUMBER ',
     c         'SPECIFIED FOR THE ASSEMBLY IN RELATIVE POSITION ',
     c         C,',',R,' IS LARGER THAN THE NUMBER OF FUEL ',
     c         'ASSEMBLY DESIGNS SPECIFIED.'
               STOP
            ENDIF
70       CONTINUE
80    CONTINUE
*     Read in the number of different insertion rod assembly bank
designations and
*     bank insertion heights for the statepoint calculation.  The insertion
height
*     values should be the distances (cm) between the bottom of the absorber
material
*     in the insertion rods and the bottom of the active fuel region.
      READ(15,*) NUMOFBANKS
      DO 100 BANK=1,NUMOFBANKS
         READ(15,90) BANKID(BANK), BANKDES(BANK),
     c   BANKHEIGHT(BANK)
90       FORMAT(T1,I2,1X,A5,1X,F7.3)
```

```
  100       CONTINUE
*     Read in the insertion rod assembly core layout.
            READ(15,*) BANKNUM(1,1), BANKNUM(2,1), BANKNUM(3,1),
     c      BANKNUM(4,1), BANKNUM(5,1), BANKNUM(6,1), BANKNUM(7,1),
     c      BANKNUM(8,1)
            READ(15,*) BANKNUM(2,2), BANKNUM(3,2), BANKNUM(4,2),
     c      BANKNUM(5,2), BANKNUM(6,2), BANKNUM(7,2), BANKNUM(8,2)
            READ(15,*) BANKNUM(3,3), BANKNUM(4,3), BANKNUM(5,3),
     c      BANKNUM(6,3), BANKNUM(7,3), BANKNUM(8,3)
            READ(15,*) BANKNUM(4,4), BANKNUM(5,4), BANKNUM(6,4),
     c      BANKNUM(7,4), BANKNUM(8,4)
            READ(15,*) BANKNUM(5,5), BANKNUM(6,5), BANKNUM(7,5)
            READ(15,*) BANKNUM(6,6), BANKNUM(7,6)
*     Read in initial enrichments.
            READ(15,*) ENRICHMENT(1,1), ENRICHMENT(2,1),
     c      ENRICHMENT(3,1), ENRICHMENT(4,1), ENRICHMENT(5,1),
     c      ENRICHMENT(6,1), ENRICHMENT(7,1), ENRICHMENT(8,1)
            READ(15,*) ENRICHMENT(2,2), ENRICHMENT(3,2),
     c      ENRICHMENT(4,2), ENRICHMENT(5,2), ENRICHMENT(6,2),
     c      ENRICHMENT(7,2), ENRICHMENT(8,2)
            READ(15,*) ENRICHMENT(3,3), ENRICHMENT(4,3),
     c      ENRICHMENT(5,3), ENRICHMENT(6,3), ENRICHMENT(7,3),
     c      ENRICHMENT(8,3)
            READ(15,*) ENRICHMENT(4,4), ENRICHMENT(5,4),
     c      ENRICHMENT(6,4), ENRICHMENT(7,4), ENRICHMENT(8,4)
            READ(15,*) ENRICHMENT(5,5), ENRICHMENT(6,5),
     c      ENRICHMENT(7,5)
            READ(15,*) ENRICHMENT(6,6), ENRICHMENT(7,6)
*     Read in fuel status (fresh or burned).
            READ(15,101) STAT(1,1), STAT(2,1),
     c      STAT(3,1), STAT(4,1), STAT(5,1),
     c      STAT(6,1), STAT(7,1), STAT(8,1)
  101       FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,102) STAT(2,2), STAT(3,2),
     c      STAT(4,2), STAT(5,2), STAT(6,2),
     c      STAT(7,2), STAT(8,2)
  102       FORMAT(T3,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,103) STAT(3,3), STAT(4,3),
     c      STAT(5,3), STAT(6,3), STAT(7,3),
     c      STAT(8,3)
  103       FORMAT(T5,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,104) STAT(4,4), STAT(5,4),
     c      STAT(6,4), STAT(7,4), STAT(8,4)
  104       FORMAT(T7,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,105) STAT(5,5), STAT(6,5),
     c      STAT(7,5)
  105       FORMAT(T9,A1,1X,A1,1X,A1)
            READ(15,106) STAT(6,6), STAT(7,6)
  106       FORMAT(T11,A1,1X,A1)
        ELSEIF (QUARTER.EQ..TRUE.) THEN
*     Read in the fuel assembly archive identifiers for retrieval of
isotopics.
            READ(15,110) ASSYID(1,1), ASSYID(2,1), ASSYID(3,1),
     c      ASSYID(4,1), ASSYID(5,1), ASSYID(6,1), ASSYID(7,1),
```

```
     c        ASSYID(8,1)
   110        FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c        A5,1X,A5)
              READ(15,120) ASSYID(1,2), ASSYID(2,2), ASSYID(3,2),
     c        ASSYID(4,2), ASSYID(5,2), ASSYID(6,2), ASSYID(7,2),
     c        ASSYID(8,2)
   120        FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c        A5,1X,A5)
              READ(15,130) ASSYID(1,3), ASSYID(2,3), ASSYID(3,3),
     c        ASSYID(4,3), ASSYID(5,3), ASSYID(6,3), ASSYID(7,3),
     c        ASSYID(8,3)
   130        FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c        A5,1X,A5)
              READ(15,140) ASSYID(1,4), ASSYID(2,4), ASSYID(3,4),
     c        ASSYID(4,4), ASSYID(5,4), ASSYID(6,4), ASSYID(7,4),
     c        ASSYID(8,4)
   140        FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,
     c        1X,A5)
              READ(15,150) ASSYID(1,5), ASSYID(2,5), ASSYID(3,5),
     c        ASSYID(4,5), ASSYID(5,5), ASSYID(6,5), ASSYID(7,5)
   150        FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5)
              READ(15,160) ASSYID(1,6), ASSYID(2,6), ASSYID(3,6),
     c        ASSYID(4,6), ASSYID(5,6), ASSYID(6,6), ASSYID(7,6)
   160        FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5)
              READ(15,170) ASSYID(1,7), ASSYID(2,7), ASSYID(3,7),
     c        ASSYID(4,7), ASSYID(5,7), ASSYID(6,7)
   170        FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5)
              READ(15,180) ASSYID(1,8), ASSYID(2,8), ASSYID(3,8),
     c        ASSYID(4,8)
   180        FORMAT(T1,A5,1X,A5,1X,A5,1X,A5)
*     Read in the number of different fuel assembly designs to be included
*     in the MCNP calculation.
              READ(15,*) NUMOFFADESIGNS
*     Read in the fuel assembly relative design designations.
              READ(15,*) DESNUM(1,1), DESNUM(2,1), DESNUM(3,1),
     c        DESNUM(4,1), DESNUM(5,1), DESNUM(6,1), DESNUM(7,1),
     c        DESNUM(8,1)
              READ(15,*) DESNUM(1,2), DESNUM(2,2), DESNUM(3,2),
     c        DESNUM(4,2), DESNUM(5,2), DESNUM(6,2), DESNUM(7,2),
     c        DESNUM(8,2)
              READ(15,*) DESNUM(1,3), DESNUM(2,3), DESNUM(3,3),
     c        DESNUM(4,3), DESNUM(5,3), DESNUM(6,3), DESNUM(7,3),
     c        DESNUM(8,3)
              READ(15,*) DESNUM(1,4), DESNUM(2,4), DESNUM(3,4),
     c        DESNUM(4,4), DESNUM(5,4), DESNUM(6,4), DESNUM(7,4),
     c        DESNUM(8,4)
              READ(15,*) DESNUM(1,5), DESNUM(2,5), DESNUM(3,5),
     c        DESNUM(4,5), DESNUM(5,5), DESNUM(6,5), DESNUM(7,5)
              READ(15,*) DESNUM(1,6), DESNUM(2,6), DESNUM(3,6),
     c        DESNUM(4,6), DESNUM(5,6), DESNUM(6,6), DESNUM(7,6)
              READ(15,*) DESNUM(1,7), DESNUM(2,7), DESNUM(3,7),
     c        DESNUM(4,7), DESNUM(5,7), DESNUM(6,7)
              READ(15,*) DESNUM(1,8), DESNUM(2,8), DESNUM(3,8),
     c        DESNUM(4,8)
```

```
          DO 200 C=1,50
             DO 190 R=1,50
                IF (DESNUM(C,R).GT.NUMOFFADESIGNS) THEN
                   WRITE(*,*) 'THE FUEL ASSEMBLY DESIGN NUMBER ',
     c             'SPECIFIED FOR THE ASSEMBLY IN RELATIVE POSITION ',
     c             C,',',R,' IS LARGER THAN THE NUMBER OF FUEL ',
     c             'ASSEMBLY DESIGNS SPECIFIED.'
                   STOP
                ENDIF
  190        CONTINUE
  200     CONTINUE
*     Read in the number of different insertion rod assembly bank
designations and
*     bank insertion heights for the statepoint calculation.  The insertion
height
*     values should be the distances (cm) between the bottom of the absorber
material
*     in the insertion rods and the bottom of the active fuel region.
          READ(15,*) NUMOFBANKS
          DO 220 BANK=1,NUMOFBANKS
             READ(15,210) BANKID(BANK), BANKDES(BANK),
     c       BANKHEIGHT(BANK)
  210        FORMAT(T1,I2,1X,A5,1X,F7.3)
  220     CONTINUE
*     Read in the insertion rod assembly core layout.
          READ(15,*) BANKNUM(1,1), BANKNUM(2,1), BANKNUM(3,1),
     c    BANKNUM(4,1), BANKNUM(5,1), BANKNUM(6,1), BANKNUM(7,1),
     c    BANKNUM(8,1)
          READ(15,*) BANKNUM(1,2), BANKNUM(2,2), BANKNUM(3,2),
     c    BANKNUM(4,2), BANKNUM(5,2), BANKNUM(6,2), BANKNUM(7,2),
     c    BANKNUM(8,2)
          READ(15,*) BANKNUM(1,3), BANKNUM(2,3), BANKNUM(3,3),
     c    BANKNUM(4,3), BANKNUM(5,3), BANKNUM(6,3), BANKNUM(7,3),
     c    BANKNUM(8,3)
          READ(15,*) BANKNUM(1,4), BANKNUM(2,4), BANKNUM(3,4),
     c    BANKNUM(4,4), BANKNUM(5,4), BANKNUM(6,4),
     c    BANKNUM(7,4), BANKNUM(8,4)
          READ(15,*) BANKNUM(1,5), BANKNUM(2,5), BANKNUM(3,5),
     c    BANKNUM(4,5), BANKNUM(5,5), BANKNUM(6,5),
     c    BANKNUM(7,5)
          READ(15,*) BANKNUM(1,6), BANKNUM(2,6), BANKNUM(3,6),
     c    BANKNUM(4,6), BANKNUM(5,6), BANKNUM(6,6), BANKNUM(7,6)
          READ(15,*) BANKNUM(1,7), BANKNUM(2,7), BANKNUM(3,7),
     c    BANKNUM(4,7), BANKNUM(5,7), BANKNUM(6,7)
          READ(15,*) BANKNUM(1,8), BANKNUM(2,8), BANKNUM(3,8),
     c    BANKNUM(4,8)
*     Read in initial enrichments if it is a BOC or BOL case.
             READ(15,*) ENRICHMENT(1,1), ENRICHMENT(2,1),
     c       ENRICHMENT(3,1), ENRICHMENT(4,1), ENRICHMENT(5,1),
     c       ENRICHMENT(6,1), ENRICHMENT(7,1), ENRICHMENT(8,1)
             READ(15,*) ENRICHMENT(1,2), ENRICHMENT(2,2),
     c       ENRICHMENT(3,2), ENRICHMENT(4,2), ENRICHMENT(5,2),
     c       ENRICHMENT(6,2), ENRICHMENT(7,2), ENRICHMENT(8,2)
             READ(15,*) ENRICHMENT(1,3), ENRICHMENT(2,3),
```

```
     c           ENRICHMENT(3,3), ENRICHMENT(4,3), ENRICHMENT(5,3),
     c           ENRICHMENT(6,3), ENRICHMENT(7,3), ENRICHMENT(8,3)
                 READ(15,*) ENRICHMENT(1,4), ENRICHMENT(2,4),
     c           ENRICHMENT(3,4), ENRICHMENT(4,4), ENRICHMENT(5,4),
     c           ENRICHMENT(6,4), ENRICHMENT(7,4), ENRICHMENT(8,4)
                 READ(15,*) ENRICHMENT(1,5), ENRICHMENT(2,5),
     c           ENRICHMENT(3,5), ENRICHMENT(4,5), ENRICHMENT(5,5),
     c           ENRICHMENT(6,5), ENRICHMENT(7,5)
                 READ(15,*) ENRICHMENT(1,6), ENRICHMENT(2,6),
     c           ENRICHMENT(3,6), ENRICHMENT(4,6), ENRICHMENT(5,6),
     c           ENRICHMENT(6,6), ENRICHMENT(7,6)
                 READ(15,*) ENRICHMENT(1,7), ENRICHMENT(2,7),
     c           ENRICHMENT(3,7), ENRICHMENT(4,7), ENRICHMENT(5,7),
     c           ENRICHMENT(6,7)
                 READ(15,*) ENRICHMENT(1,8), ENRICHMENT(2,8),
     c           ENRICHMENT(3,8), ENRICHMENT(4,8)
*     Read in fuel status (fresh or burned).
            READ(15,221) STAT(1,1), STAT(2,1),
     c      STAT(3,1), STAT(4,1), STAT(5,1),
     c      STAT(6,1), STAT(7,1), STAT(8,1)
 221        FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,222) STAT(1,2), STAT(2,2),
     c      STAT(3,2), STAT(4,2), STAT(5,2),
     c      STAT(6,2), STAT(7,2), STAT(8,2)
 222        FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,223) STAT(1,3), STAT(2,3),
     c      STAT(3,3), STAT(4,3), STAT(5,3),
     c      STAT(6,3), STAT(7,3), STAT(8,3)
 223        FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,224) STAT(1,4), STAT(2,4),
     c      STAT(3,4), STAT(4,4), STAT(5,4),
     c      STAT(6,4), STAT(7,4), STAT(8,4)
 224        FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,225) STAT(1,5), STAT(2,5),
     c      STAT(3,5), STAT(4,5), STAT(5,5),
     c      STAT(6,5), STAT(7,5)
 225        FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,226) STAT(1,6), STAT(2,6),
     c      STAT(3,6), STAT(4,6), STAT(5,6),
     c      STAT(6,6), STAT(7,6)
 226        FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,227) STAT(1,7), STAT(2,7),
     c      STAT(3,7), STAT(4,7), STAT(5,7), STAT(6,7)
 227        FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,228) STAT(1,8), STAT(2,8),
     c      STAT(3,8), STAT(4,8)
 228        FORMAT(T1,A1,1X,A1,1X,A1,1X,A1)
        ELSEIF (FULL.EQ..TRUE.) THEN
*     Read in the fuel assembly archive identifiers for retrieval of
isotopics.
            READ(15,230) ASSYID(5,1), ASSYID(6,1), ASSYID(7,1),
     c      ASSYID(8,1), ASSYID(9,1), ASSYID(10,1), ASSYID(11,1)
 230        FORMAT(T25,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5)
            READ(15,240) ASSYID(3,2), ASSYID(4,2), ASSYID(5,2),
```

```
     c      ASSYID(6,2), ASSYID(7,2), ASSYID(8,2), ASSYID(9,2),
     c      ASSYID(10,2), ASSYID(11,2), ASSYID(12,2), ASSYID(13,2)
 240        FORMAT(T13,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c      A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5)
            READ(15,250) ASSYID(2,3), ASSYID(3,3), ASSYID(4,3),
     c      ASSYID(5,3), ASSYID(6,3), ASSYID(7,3), ASSYID(8,3),
     c      ASSYID(9,3), ASSYID(10,3), ASSYID(11,3), ASSYID(12,3),
     c      ASSYID(13,3), ASSYID(14,3)
 250        FORMAT(T7,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c      A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5)
            READ(15,260) ASSYID(2,4), ASSYID(3,4), ASSYID(4,4),
     c      ASSYID(5,4), ASSYID(6,4), ASSYID(7,4), ASSYID(8,4),
     c      ASSYID(9,4), ASSYID(10,4), ASSYID(11,4), ASSYID(12,4),
     c      ASSYID(13,4), ASSYID(14,4)
 260        FORMAT(T7,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c      A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5)
            READ(15,270) ASSYID(1,5), ASSYID(2,5), ASSYID(3,5),
     c      ASSYID(4,5), ASSYID(5,5), ASSYID(6,5), ASSYID(7,5),
     c      ASSYID(8,5), ASSYID(9,5), ASSYID(10,5), ASSYID(11,5),
     c      ASSYID(12,5), ASSYID(13,5), ASSYID(14,5), ASSYID(15,5)
 270        FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c      A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5)
            READ(15,280) ASSYID(1,6), ASSYID(2,6), ASSYID(3,6),
     c      ASSYID(4,6), ASSYID(5,6), ASSYID(6,6), ASSYID(7,6),
     c      ASSYID(8,6), ASSYID(9,6), ASSYID(10,6), ASSYID(11,6),
     c      ASSYID(12,6), ASSYID(13,6), ASSYID(14,6), ASSYID(15,6)
 280        FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c      A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5)
            READ(15,290) ASSYID(1,7), ASSYID(2,7), ASSYID(3,7),
     c      ASSYID(4,7), ASSYID(5,7), ASSYID(6,7), ASSYID(7,7),
     c      ASSYID(8,7), ASSYID(9,7), ASSYID(10,7), ASSYID(11,7),
     c      ASSYID(12,7), ASSYID(13,7), ASSYID(14,7), ASSYID(15,7)
 290        FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c      A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5)
            READ(15,300) ASSYID(1,8), ASSYID(2,8), ASSYID(3,8),
     c      ASSYID(4,8), ASSYID(5,8), ASSYID(6,8), ASSYID(7,8),
     c      ASSYID(8,8), ASSYID(9,8), ASSYID(10,8), ASSYID(11,8),
     c      ASSYID(12,8), ASSYID(13,8), ASSYID(14,8), ASSYID(15,8)
 300        FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c      A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5)
            READ(15,310) ASSYID(1,9), ASSYID(2,9), ASSYID(3,9),
     c      ASSYID(4,9), ASSYID(5,9), ASSYID(6,9), ASSYID(7,9),
     c      ASSYID(8,9), ASSYID(9,9), ASSYID(10,9), ASSYID(11,9),
     c      ASSYID(12,9), ASSYID(13,9), ASSYID(14,9), ASSYID(15,9)
 310        FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c      A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5)
            READ(15,320) ASSYID(1,10), ASSYID(2,10), ASSYID(3,10),
     c      ASSYID(4,10), ASSYID(5,10), ASSYID(6,10), ASSYID(7,10),
     c      ASSYID(8,10), ASSYID(9,10), ASSYID(10,10), ASSYID(11,10),
     c      ASSYID(12,10), ASSYID(13,10), ASSYID(14,10), ASSYID(15,10)
 320        FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c      A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5)
            READ(15,330) ASSYID(1,11), ASSYID(2,11), ASSYID(3,11),
     c      ASSYID(4,11), ASSYID(5,11), ASSYID(6,11), ASSYID(7,11),
```

```
     c       ASSYID(8,11), ASSYID(9,11), ASSYID(10,11), ASSYID(11,11),
     c       ASSYID(12,11), ASSYID(13,11), ASSYID(14,11), ASSYID(15,11)
 330         FORMAT(T1,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c       A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5)
             READ(15,340) ASSYID(2,12), ASSYID(3,12), ASSYID(4,12),
     c       ASSYID(5,12), ASSYID(6,12), ASSYID(7,12), ASSYID(8,12),
     c       ASSYID(9,12), ASSYID(10,12), ASSYID(11,12), ASSYID(12,12),
     c       ASSYID(13,12), ASSYID(14,12)
 340         FORMAT(T7,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c       A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5)
             READ(15,350) ASSYID(2,13), ASSYID(3,13), ASSYID(4,13),
     c       ASSYID(5,13), ASSYID(6,13), ASSYID(7,13), ASSYID(8,13),
     c       ASSYID(9,13), ASSYID(10,13), ASSYID(11,13), ASSYID(12,13),
     c       ASSYID(13,13), ASSYID(14,13)
 350         FORMAT(T7,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c       A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5)
             READ(15,360) ASSYID(3,14), ASSYID(4,14), ASSYID(5,14),
     c       ASSYID(6,14), ASSYID(7,14), ASSYID(8,14), ASSYID(9,14),
     c       ASSYID(10,14), ASSYID(11,14), ASSYID(12,14), ASSYID(13,14)
 360         FORMAT(T13,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,
     c       A5,1X,A5,1X,A5,1X,A5,1X,A5)
             READ(15,370) ASSYID(5,15), ASSYID(6,15), ASSYID(7,15),
     c       ASSYID(8,15), ASSYID(9,15), ASSYID(10,15), ASSYID(11,15)
 370         FORMAT(T25,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5,1X,A5)
*     Read in the number of different fuel assembly designs to be included
*     in the MCNP calculation.
             READ(15,*) NUMOFFADESIGNS
*     Read in the fuel assembly relative design designations.
             READ(15,*) DESNUM(5,1), DESNUM(6,1), DESNUM(7,1),
     c       DESNUM(8,1), DESNUM(9,1), DESNUM(10,1), DESNUM(11,1)
             READ(15,*) DESNUM(3,2), DESNUM(4,2), DESNUM(5,2),
     c       DESNUM(6,2), DESNUM(7,2), DESNUM(8,2), DESNUM(9,2),
     c       DESNUM(10,2), DESNUM(11,2), DESNUM(12,2), DESNUM(13,2)
             READ(15,*) DESNUM(2,3), DESNUM(3,3), DESNUM(4,3),
     c       DESNUM(5,3), DESNUM(6,3), DESNUM(7,3), DESNUM(8,3),
     c       DESNUM(9,3), DESNUM(10,3), DESNUM(11,3), DESNUM(12,3),
     c       DESNUM(13,3), DESNUM(14,3)
             READ(15,*) DESNUM(2,4), DESNUM(3,4), DESNUM(4,4),
     c       DESNUM(5,4), DESNUM(6,4), DESNUM(7,4), DESNUM(8,4),
     c       DESNUM(9,4), DESNUM(10,4), DESNUM(11,4), DESNUM(12,4),
     c       DESNUM(13,4), DESNUM(14,4)
             READ(15,*) DESNUM(1,5), DESNUM(2,5), DESNUM(3,5),
     c       DESNUM(4,5), DESNUM(5,5), DESNUM(6,5), DESNUM(7,5),
     c       DESNUM(8,5), DESNUM(9,5), DESNUM(10,5), DESNUM(11,5),
     c       DESNUM(12,5), DESNUM(13,5), DESNUM(14,5), DESNUM(15,5)
             READ(15,*) DESNUM(1,6), DESNUM(2,6), DESNUM(3,6),
     c       DESNUM(4,6), DESNUM(5,6), DESNUM(6,6), DESNUM(7,6),
     c       DESNUM(8,6), DESNUM(9,6), DESNUM(10,6), DESNUM(11,6),
     c       DESNUM(12,6), DESNUM(13,6), DESNUM(14,6), DESNUM(15,6)
             READ(15,*) DESNUM(1,7), DESNUM(2,7), DESNUM(3,7),
     c       DESNUM(4,7), DESNUM(5,7), DESNUM(6,7), DESNUM(7,7),
     c       DESNUM(8,7), DESNUM(9,7), DESNUM(10,7), DESNUM(11,7),
     c       DESNUM(12,7), DESNUM(13,7), DESNUM(14,7), DESNUM(15,7)
             READ(15,*) DESNUM(1,8), DESNUM(2,8), DESNUM(3,8),
```

```
     c        DESNUM(4,8), DESNUM(5,8), DESNUM(6,8), DESNUM(7,8),
     c        DESNUM(8,8), DESNUM(9,8), DESNUM(10,8), DESNUM(11,8),
     c        DESNUM(12,8), DESNUM(13,8), DESNUM(14,8), DESNUM(15,8)
              READ(15,*) DESNUM(1,9), DESNUM(2,9), DESNUM(3,9),
     c        DESNUM(4,9), DESNUM(5,9), DESNUM(6,9), DESNUM(7,9),
     c        DESNUM(8,9), DESNUM(9,9), DESNUM(10,9), DESNUM(11,9),
     c        DESNUM(12,9), DESNUM(13,9), DESNUM(14,9), DESNUM(15,9)
              READ(15,*) DESNUM(1,10), DESNUM(2,10), DESNUM(3,10),
     c        DESNUM(4,10), DESNUM(5,10), DESNUM(6,10), DESNUM(7,10),
     c        DESNUM(8,10), DESNUM(9,10), DESNUM(10,10), DESNUM(11,10),
     c        DESNUM(12,10), DESNUM(13,10), DESNUM(14,10), DESNUM(15,10)
              READ(15,*) DESNUM(1,11), DESNUM(2,11), DESNUM(3,11),
     c        DESNUM(4,11), DESNUM(5,11), DESNUM(6,11), DESNUM(7,11),
     c        DESNUM(8,11), DESNUM(9,11), DESNUM(10,11), DESNUM(11,11),
     c        DESNUM(12,11), DESNUM(13,11), DESNUM(14,11), DESNUM(15,11)
              READ(15,*) DESNUM(2,12), DESNUM(3,12), DESNUM(4,12),
     c        DESNUM(5,12), DESNUM(6,12), DESNUM(7,12), DESNUM(8,12),
     c        DESNUM(9,12), DESNUM(10,12), DESNUM(11,12), DESNUM(12,12),
     c        DESNUM(13,12), DESNUM(14,12)
              READ(15,*) DESNUM(2,13), DESNUM(3,13), DESNUM(4,13),
     c        DESNUM(5,13), DESNUM(6,13), DESNUM(7,13), DESNUM(8,13),
     c        DESNUM(9,13), DESNUM(10,13), DESNUM(11,13), DESNUM(12,13),
     c        DESNUM(13,13), DESNUM(14,13)
              READ(15,*) DESNUM(3,14), DESNUM(4,14), DESNUM(5,14),
     c        DESNUM(6,14), DESNUM(7,14), DESNUM(8,14), DESNUM(9,14),
     c        DESNUM(10,14), DESNUM(11,14), DESNUM(12,14), DESNUM(13,14)
              READ(15,*) DESNUM(5,15), DESNUM(6,15), DESNUM(7,15),
     c        DESNUM(8,15), DESNUM(9,15), DESNUM(10,15), DESNUM(11,15)
              DO 390 C=1,50
                 DO 380 R=1,50
                    IF (DESNUM(C,R).GT.NUMOFFADESIGNS) THEN
                       WRITE(*,*) 'THE FUEL ASSEMBLY DESIGN NUMBER ',
     c                 'SPECIFIED FOR THE ASSEMBLY IN POSITION ',
     c                 C,',',R,' IS LARGER THAN THE NUMBER OF FUEL ',
     c                 'ASSEMBLY DESIGNS SPECIFIED.'
                       STOP
                    ENDIF
  380            CONTINUE
  390         CONTINUE
*      Read in the number of different insertion rod assembly bank
designations and
*      bank insertion heights for the statepoint calculation.  The insertion
height
*      values should be the distances (cm) between the bottom of the absorber
material
*      in the insertion rods and the bottom of the active fuel region.
              READ(15,*) NUMOFBANKS
              DO 392 BANK=1,NUMOFBANKS
                 READ(15,391) BANKID(BANK), BANKDES(BANK),
     c           BANKHEIGHT(BANK)
  391            FORMAT(T1,I2,1X,A5,1X,F7.3)
  392         CONTINUE
*      Read in the insertion rod assembly core layout.
              READ(15,*) BANKNUM(5,1), BANKNUM(6,1), BANKNUM(7,1),
```

```
     c         BANKNUM(8,1), BANKNUM(9,1), BANKNUM(10,1), BANKNUM(11,1)
               READ(15,*) BANKNUM(3,2), BANKNUM(4,2), BANKNUM(5,2),
     c         BANKNUM(6,2), BANKNUM(7,2), BANKNUM(8,2), BANKNUM(9,2),
     c         BANKNUM(10,2), BANKNUM(11,2), BANKNUM(12,2), BANKNUM(13,2)
               READ(15,*) BANKNUM(2,3), BANKNUM(3,3), BANKNUM(4,3),
     c         BANKNUM(5,3), BANKNUM(6,3), BANKNUM(7,3), BANKNUM(8,3),
     c         BANKNUM(9,3), BANKNUM(10,3), BANKNUM(11,3), BANKNUM(12,3),
     c         BANKNUM(13,3), BANKNUM(14,3)
               READ(15,*) BANKNUM(2,4), BANKNUM(3,4), BANKNUM(4,4),
     c         BANKNUM(5,4), BANKNUM(6,4), BANKNUM(7,4), BANKNUM(8,4),
     c         BANKNUM(9,4), BANKNUM(10,4), BANKNUM(11,4), BANKNUM(12,4),
     c         BANKNUM(13,4), BANKNUM(14,4)
               READ(15,*) BANKNUM(1,5), BANKNUM(2,5), BANKNUM(3,5),
     c         BANKNUM(4,5), BANKNUM(5,5), BANKNUM(6,5), BANKNUM(7,5),
     c         BANKNUM(8,5), BANKNUM(9,5), BANKNUM(10,5), BANKNUM(11,5),
     c         BANKNUM(12,5), BANKNUM(13,5), BANKNUM(14,5), BANKNUM(15,5)
               READ(15,*) BANKNUM(1,6), BANKNUM(2,6), BANKNUM(3,6),
     c         BANKNUM(4,6), BANKNUM(5,6), BANKNUM(6,6), BANKNUM(7,6),
     c         BANKNUM(8,6), BANKNUM(9,6), BANKNUM(10,6), BANKNUM(11,6),
     c         BANKNUM(12,6), BANKNUM(13,6), BANKNUM(14,6), BANKNUM(15,6)
               READ(15,*) BANKNUM(1,7), BANKNUM(2,7), BANKNUM(3,7),
     c         BANKNUM(4,7), BANKNUM(5,7), BANKNUM(6,7), BANKNUM(7,7),
     c         BANKNUM(8,7), BANKNUM(9,7), BANKNUM(10,7), BANKNUM(11,7),
     c         BANKNUM(12,7), BANKNUM(13,7), BANKNUM(14,7), BANKNUM(15,7)
               READ(15,*) BANKNUM(1,8), BANKNUM(2,8), BANKNUM(3,8),
     c         BANKNUM(4,8), BANKNUM(5,8), BANKNUM(6,8), BANKNUM(7,8),
     c         BANKNUM(8,8), BANKNUM(9,8), BANKNUM(10,8), BANKNUM(11,8),
     c         BANKNUM(12,8), BANKNUM(13,8), BANKNUM(14,8), BANKNUM(15,8)
               READ(15,*) BANKNUM(1,9), BANKNUM(2,9), BANKNUM(3,9),
     c         BANKNUM(4,9), BANKNUM(5,9), BANKNUM(6,9), BANKNUM(7,9),
     c         BANKNUM(8,9), BANKNUM(9,9), BANKNUM(10,9), BANKNUM(11,9),
     c         BANKNUM(12,9), BANKNUM(13,9), BANKNUM(14,9), BANKNUM(15,9)
               READ(15,*) BANKNUM(1,10), BANKNUM(2,10), BANKNUM(3,10),
     c         BANKNUM(4,10), BANKNUM(5,10), BANKNUM(6,10), BANKNUM(7,10),
     c         BANKNUM(8,10), BANKNUM(9,10), BANKNUM(10,10),
     c         BANKNUM(11,10), BANKNUM(12,10), BANKNUM(13,10),
     c         BANKNUM(14,10), BANKNUM(15,10)
               READ(15,*) BANKNUM(1,11), BANKNUM(2,11), BANKNUM(3,11),
     c         BANKNUM(4,11), BANKNUM(5,11), BANKNUM(6,11), BANKNUM(7,11),
     c         BANKNUM(8,11), BANKNUM(9,11), BANKNUM(10,11),
     c         BANKNUM(11,11), BANKNUM(12,11), BANKNUM(13,11),
     c         BANKNUM(14,11), BANKNUM(15,11)
               READ(15,*) BANKNUM(2,12), BANKNUM(3,12), BANKNUM(4,12),
     c         BANKNUM(5,12), BANKNUM(6,12), BANKNUM(7,12), BANKNUM(8,12),
     c         BANKNUM(9,12), BANKNUM(10,12), BANKNUM(11,12),
     c         BANKNUM(12,12), BANKNUM(13,12), BANKNUM(14,12)
               READ(15,*) BANKNUM(2,13), BANKNUM(3,13), BANKNUM(4,13),
     c         BANKNUM(5,13), BANKNUM(6,13), BANKNUM(7,13), BANKNUM(8,13),
     c         BANKNUM(9,13), BANKNUM(10,13), BANKNUM(11,13),
     c         BANKNUM(12,13), BANKNUM(13,13), BANKNUM(14,13)
               READ(15,*) BANKNUM(3,14), BANKNUM(4,14), BANKNUM(5,14),
     c         BANKNUM(6,14), BANKNUM(7,14), BANKNUM(8,14), BANKNUM(9,14),
     c         BANKNUM(10,14), BANKNUM(11,14), BANKNUM(12,14),
     c         BANKNUM(13,14)
```

```
        READ(15,*) BANKNUM(5,15), BANKNUM(6,15), BANKNUM(7,15),
     c  BANKNUM(8,15), BANKNUM(9,15), BANKNUM(10,15),
     c  BANKNUM(11,15)
*   Read in initial enrichments if it is a BOC or BOL case.
           READ(15,*) ENRICHMENT(5,1), ENRICHMENT(6,1),
     c     ENRICHMENT(7,1), ENRICHMENT(8,1), ENRICHMENT(9,1),
     c     ENRICHMENT(10,1), ENRICHMENT(11,1)
           READ(15,*) ENRICHMENT(3,2), ENRICHMENT(4,2),
     c     ENRICHMENT(5,2), ENRICHMENT(6,2), ENRICHMENT(7,2),
     c     ENRICHMENT(8,2), ENRICHMENT(9,2), ENRICHMENT(10,2),
     c     ENRICHMENT(11,2), ENRICHMENT(12,2), ENRICHMENT(13,2)
           READ(15,*) ENRICHMENT(2,3), ENRICHMENT(3,3),
     c     ENRICHMENT(4,3), ENRICHMENT(5,3), ENRICHMENT(6,3),
     c     ENRICHMENT(7,3), ENRICHMENT(8,3), ENRICHMENT(9,3),
     c     ENRICHMENT(10,3), ENRICHMENT(11,3), ENRICHMENT(12,3),
     c     ENRICHMENT(13,3), ENRICHMENT(14,3)
           READ(15,*) ENRICHMENT(2,4),
     c     ENRICHMENT(3,4), ENRICHMENT(4,4), ENRICHMENT(5,4),
     c     ENRICHMENT(6,4), ENRICHMENT(7,4), ENRICHMENT(8,4),
     c     ENRICHMENT(9,4), ENRICHMENT(10,4), ENRICHMENT(11,4),
     c     ENRICHMENT(12,4), ENRICHMENT(13,4), ENRICHMENT(14,4)
           READ(15,*) ENRICHMENT(1,5), ENRICHMENT(2,5),
     c     ENRICHMENT(3,5),
     c     ENRICHMENT(4,5), ENRICHMENT(5,5), ENRICHMENT(6,5),
     c     ENRICHMENT(7,5), ENRICHMENT(8,5), ENRICHMENT(9,5),
     c     ENRICHMENT(10,5), ENRICHMENT(11,5), ENRICHMENT(12,5),
     c     ENRICHMENT(13,5), ENRICHMENT(14,5), ENRICHMENT(15,5)
           READ(15,*) ENRICHMENT(1,6), ENRICHMENT(2,6),
     c     ENRICHMENT(3,6), ENRICHMENT(4,6), ENRICHMENT(5,6),
     c     ENRICHMENT(6,6), ENRICHMENT(7,6), ENRICHMENT(8,6),
     c     ENRICHMENT(9,6), ENRICHMENT(10,6), ENRICHMENT(11,6),
     c     ENRICHMENT(12,6), ENRICHMENT(13,6), ENRICHMENT(14,6),
     c     ENRICHMENT(15,6)
           READ(15,*) ENRICHMENT(1,7), ENRICHMENT(2,7),
     c     ENRICHMENT(3,7), ENRICHMENT(4,7), ENRICHMENT(5,7),
     c     ENRICHMENT(6,7), ENRICHMENT(7,7), ENRICHMENT(8,7),
     c     ENRICHMENT(9,7), ENRICHMENT(10,7), ENRICHMENT(11,7),
     c     ENRICHMENT(12,7), ENRICHMENT(13,7), ENRICHMENT(14,7),
     c     ENRICHMENT(15,7)
           READ(15,*) ENRICHMENT(1,8), ENRICHMENT(2,8),
     c     ENRICHMENT(3,8), ENRICHMENT(4,8), ENRICHMENT(5,8),
     c     ENRICHMENT(6,8), ENRICHMENT(7,8), ENRICHMENT(8,8),
     c     ENRICHMENT(9,8), ENRICHMENT(10,8), ENRICHMENT(11,8),
     c     ENRICHMENT(12,8), ENRICHMENT(13,8), ENRICHMENT(14,8),
     c     ENRICHMENT(15,8)
           READ(15,*) ENRICHMENT(1,9), ENRICHMENT(2,9),
     c     ENRICHMENT(3,9), ENRICHMENT(4,9), ENRICHMENT(5,9),
     c     ENRICHMENT(6,9), ENRICHMENT(7,9), ENRICHMENT(8,9),
     c     ENRICHMENT(9,9), ENRICHMENT(10,9), ENRICHMENT(11,9),
     c     ENRICHMENT(12,9), ENRICHMENT(13,9), ENRICHMENT(14,9),
     c     ENRICHMENT(15,9)
           READ(15,*) ENRICHMENT(1,10), ENRICHMENT(2,10),
     c     ENRICHMENT(3,10), ENRICHMENT(4,10), ENRICHMENT(5,10),
     c     ENRICHMENT(6,10), ENRICHMENT(7,10), ENRICHMENT(8,10),
```

```
     c          ENRICHMENT(9,10), ENRICHMENT(10,10), ENRICHMENT(11,10),
     c          ENRICHMENT(12,10), ENRICHMENT(13,10), ENRICHMENT(14,10),
     c          ENRICHMENT(15,10)
                READ(15,*) ENRICHMENT(1,11), ENRICHMENT(2,11),
     c          ENRICHMENT(3,11),
     c          ENRICHMENT(4,11), ENRICHMENT(5,11), ENRICHMENT(6,11),
     c          ENRICHMENT(7,11), ENRICHMENT(8,11), ENRICHMENT(9,11),
     c          ENRICHMENT(10,11), ENRICHMENT(11,11), ENRICHMENT(12,11),
     c          ENRICHMENT(13,11), ENRICHMENT(14,11), ENRICHMENT(15,11)
                READ(15,*) ENRICHMENT(2,12), ENRICHMENT(3,12),
     c          ENRICHMENT(4,12), ENRICHMENT(5,12), ENRICHMENT(6,12),
     c          ENRICHMENT(7,12), ENRICHMENT(8,12), ENRICHMENT(9,12),
     c          ENRICHMENT(10,12), ENRICHMENT(11,12), ENRICHMENT(12,12),
     c          ENRICHMENT(13,12), ENRICHMENT(14,12)
                READ(15,*) ENRICHMENT(2,13), ENRICHMENT(3,13),
     c          ENRICHMENT(4,13),
     c          ENRICHMENT(5,13), ENRICHMENT(6,13), ENRICHMENT(7,13),
     c          ENRICHMENT(8,13), ENRICHMENT(9,13), ENRICHMENT(10,13),
     c          ENRICHMENT(11,13), ENRICHMENT(12,13),
     c          ENRICHMENT(13,13), ENRICHMENT(14,13)
                READ(15,*) ENRICHMENT(3,14), ENRICHMENT(4,14),
     c          ENRICHMENT(5,14),
     c          ENRICHMENT(6,14), ENRICHMENT(7,14), ENRICHMENT(8,14),
     c          ENRICHMENT(9,14), ENRICHMENT(10,14), ENRICHMENT(11,14),
     c          ENRICHMENT(12,14), ENRICHMENT(13,14)
                READ(15,*) ENRICHMENT(5,15), ENRICHMENT(6,15),
     c          ENRICHMENT(7,15), ENRICHMENT(8,15), ENRICHMENT(9,15),
     c          ENRICHMENT(10,15), ENRICHMENT(11,15)
*     Read in fuel status (fresh or burned).
            READ(15,393) STAT(5,1), STAT(6,1), STAT(7,1),
     c      STAT(8,1), STAT(9,1), STAT(10,1), STAT(11,1)
 393        FORMAT(T9,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,
     c      1X,A1)
            READ(15,394) STAT(3,2), STAT(4,2), STAT(5,2),
     c      STAT(6,2), STAT(7,2), STAT(8,2), STAT(9,2),
     c      STAT(10,2), STAT(11,2), STAT(12,2), STAT(13,2)
 394        FORMAT(T5,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,
     c      A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,395) STAT(2,3), STAT(3,3), STAT(4,3),
     c      STAT(5,3), STAT(6,3), STAT(7,3), STAT(8,3),
     c      STAT(9,3), STAT(10,3), STAT(11,3), STAT(12,3),
     c      STAT(13,3), STAT(14,3)
 395        FORMAT(T3,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,
     c      A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,396) STAT(2,4), STAT(3,4), STAT(4,4),
     c      STAT(5,4), STAT(6,4), STAT(7,4), STAT(8,4),
     c      STAT(9,4), STAT(10,4), STAT(11,4), STAT(12,4),
     c      STAT(13,4), STAT(14,4)
 396        FORMAT(T3,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,
     c      1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,397) STAT(1,5), STAT(2,5), STAT(3,5),
     c      STAT(4,5), STAT(5,5), STAT(6,5), STAT(7,5),
     c      STAT(8,5), STAT(9,5), STAT(10,5), STAT(11,5),
     c      STAT(12,5), STAT(13,5), STAT(14,5), STAT(15,5)
```

```
 397        FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,
     c      1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,398) STAT(1,6), STAT(2,6), STAT(3,6),
     c      STAT(4,6), STAT(5,6), STAT(6,6), STAT(7,6),
     c      STAT(8,6), STAT(9,6), STAT(10,6), STAT(11,6),
     c      STAT(12,6), STAT(13,6), STAT(14,6), STAT(15,6)
 398        FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,
     c      1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,399) STAT(1,7), STAT(2,7), STAT(3,7),
     c      STAT(4,7), STAT(5,7), STAT(6,7), STAT(7,7),
     c      STAT(8,7), STAT(9,7), STAT(10,7), STAT(11,7),
     c      STAT(12,7), STAT(13,7), STAT(14,7), STAT(15,7)
 399        FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,
     c      1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,400) STAT(1,8), STAT(2,8), STAT(3,8),
     c      STAT(4,8), STAT(5,8), STAT(6,8), STAT(7,8),
     c      STAT(8,8), STAT(9,8), STAT(10,8), STAT(11,8),
     c      STAT(12,8), STAT(13,8), STAT(14,8), STAT(15,8)
 400        FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,
     c      1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,401) STAT(1,9), STAT(2,9), STAT(3,9),
     c      STAT(4,9), STAT(5,9), STAT(6,9), STAT(7,9),
     c      STAT(8,9), STAT(9,9), STAT(10,9), STAT(11,9),
     c      STAT(12,9), STAT(13,9), STAT(14,9), STAT(15,9)
 401        FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,
     c      1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,402) STAT(1,10), STAT(2,10), STAT(3,10),
     c      STAT(4,10), STAT(5,10), STAT(6,10), STAT(7,10),
     c      STAT(8,10), STAT(9,10), STAT(10,10), STAT(11,10),
     c      STAT(12,10), STAT(13,10), STAT(14,10), STAT(15,10)
 402        FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,
     c      1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,403) STAT(1,11), STAT(2,11), STAT(3,11),
     c      STAT(4,11), STAT(5,11), STAT(6,11), STAT(7,11),
     c      STAT(8,11), STAT(9,11), STAT(10,11), STAT(11,11),
     c      STAT(12,11), STAT(13,11), STAT(14,11), STAT(15,11)
 403        FORMAT(T1,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,
     c      1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,404) STAT(2,12), STAT(3,12), STAT(4,12),
     c      STAT(5,12), STAT(6,12), STAT(7,12), STAT(8,12),
     c      STAT(9,12), STAT(10,12), STAT(11,12), STAT(12,12),
     c      STAT(13,12), STAT(14,12)
 404        FORMAT(T3,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,
     c      1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,405) STAT(2,13), STAT(3,13), STAT(4,13),
     c      STAT(5,13), STAT(6,13), STAT(7,13), STAT(8,13),
     c      STAT(9,13), STAT(10,13), STAT(11,13), STAT(12,13),
     c      STAT(13,13), STAT(14,13)
 405        FORMAT(T3,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,
     c      A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
            READ(15,406) STAT(3,14), STAT(4,14), STAT(5,14),
     c      STAT(6,14), STAT(7,14), STAT(8,14), STAT(9,14),
     c      STAT(10,14), STAT(11,14), STAT(12,14), STAT(13,14)
 406        FORMAT(T5,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,
```

```
     c        A1,1X,A1,1X,A1,1X,A1,1X,A1)
              READ(15,407) STAT(5,15), STAT(6,15), STAT(7,15),
     c        STAT(8,15), STAT(9,15), STAT(10,15), STAT(11,15)
 407          FORMAT(T9,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
          ENDIF
      RETURN
      END

*********************************************************************
*********************************************************************
*********************************************************************
*********************************************************************
      SUBROUTINE WESTCRA(SPACERDIST, ENDFITHEIGHT, SURFVALUESPEC,
     c CRADIM, CRABSMAT, CRABSWTS, CRUPLENMAT, CRUPLENWTS,
     c CRLPLENMAT, CRLPLENWTS, GTDATA, LEFMAT,
     c UEFMAT, MODDENSITY, REGABOVECRA, SPACERHEIGHT,
     c HOMOSPACERDEN, MN, LN, CRAUNIV, BANKNUM, CRCLADMAT,
     c CRCLADML, CRUPML, CRLPML, FRLEFML, FRUEFML, DESNUM,
     c GTMAT, GTML, BMODML, NUMREGABOVECRA, SYSTEMTOP,
     c FRUREGIONML, NUMOFSPACERS, HOMOSPACMLNUM, ASSYID,
     c SURFTYPESPEC, CRABSZAIDS, BANKDES, CRUPZS, CRLPLENZAIDS,
     c CRUNIQUE, HYBRID, SN, CRAXCLADML, GTAXML, NUMCRAXS,
     c GTAXMAT, NUMOFGTAXS, CRAXCLADMAT, GTUNIV,
     c CRABSAXWTS, CRAXDIM, GTAXDATA, CRABSAXMAT,
     c CRABSAXZAIDS, GTSPLIT, NODEBOTTOMSURF)
*
      INTEGER MN, LN, CRAUNIV(50,50), BANKNUM(50,50),
     c CRCLADMAT(20),
     c CRCLADML(50,50), CRUPML(50,50), CRLPML(50,50),
     c FRLEFML(50,50), FRUEFML(50,50), DESNUM(50,50),
     c GTMAT(20), GTML(50,50), BMODML, NUMREGABOVECRA,
     c SYSTEMTOP,
     c FRUREGIONML(50,50,20), NUMOFSPACERS(20),
     c HOMOSPACMLNUM(20,15),
     c ROW, COLUMN, CURRENTSURFLABEL, V, SN, UEFBOTTOMSURF,
     c UEFTOPSURF, CRABSML, CRABSSURF, CRABSTOPSURF,
     c CRABSBOTTOMSURF,
     c CRCLADIRSURF, CRCLADORSURF, CRCLADTOPSURF,
     c CRCLADBOTTOMSURF,
     c C, RO, CO, GTTOPSURF, GTBOTSURF, GTORSURF, GTIRSURF,
     c CRLEFTOPSURF, REGION, REGIONTOPSURF, REGIONBOTTOMSURF,
     c SPN, SPACERTOPSURF, SPACERBOTTOMSURF,
     c WATERREGIONTOPSURF,
     c WATERREGIONBOTTOMSURF, HYBRID, CRAXABSSURF(5),
     c CRAXABSTOPSURF(5), CRAXABSBOTTOMSURF(5),
     c CRAXCLADIRSURF(5), CRAXCLADORSURF(5),
     c CRAXCLADTOPSURF(5), CRAXCLADBOTTOMSURF(5),
     c GTSECTBOTSURF(5), GTSECTTOPSURF(5),
     c GTSECTORSURF(5), GTSECTIRSURF(5),
     c CRAXCLADML(50,50,5), GTAXML(50,50,5), NUMCRAXS(20),
     c GTAXMAT(20,5), NUMOFGTAXS(20), CRAXCLADMAT(20,20),
     c GTUNIV(50,50), GTSPLIT, NODEBOTTOMSURF
*
      REAL SPACERDIST(20,10), ENDFITHEIGHT(20,2),
```

```
     c SURFVALUESPEC(200),
     c CRADIM(20,7), CRABSMAT(20,2), CRABSWTS(20,35),
     c CRUPLENMAT(20,2),
     c CRUPLENWTS(20,35), CRLPLENMAT(20,2), CRLPLENWTS(20,35),
     c GTDATA(20,4), LEFMAT(20,2),
     c UEFMAT(20,2),
     c MODDENSITY, REGABOVECRA(20,3), SPACERHEIGHT(20,10),
     c HOMOSPACERDEN(20,15), CURRENTSURF, CLADRHO,
     c SPACHEIGHT, CRABSAXWTS(20,35,20), CRAXDIM(20,5,20),
     c GTAXDATA(20,4,5), CRABSAXMAT(20,2,20)
*
      CHARACTER ASSYID(50,50)*5, SURFTYPESPEC(200)*2,
     c CRABSZAIDS(20,35)*9, BANKDES(20)*5, CRUPZS(20,35)*9,
     c CRLPLENZAIDS(20,35)*9, CRABSAXZAIDS(20,35,20)*9
*
      LOGICAL CRUNIQUE(50,50), CLADMLUNIQUE, LEAVE,
     c CRUPMLUNIQUE, CRLPMLUNIQUE
*
*  Write the specifications for the CR universes that are
*  required to fill the assembly layout specifications previously defined.
      DO 3730 ROW=1,50
         DO 3720 COLUMN=1,50
*     Write the CR universe specification for the assembly if it
*     contains a unique CR material or unique CR position.
*
      IF (CRUNIQUE(COLUMN,ROW).EQ..TRUE.) THEN
*  Write the CR specification header.
         WRITE(30,10)
   10    FORMAT(T1,'C')
         WRITE(30,20) ASSYID(COLUMN,ROW)
   20    FORMAT(T1,
     c   'C   CONTROL ROD UNIVERSE SPECIFICATION FOR ASSEMBLY ',A5)
         WRITE(30,30)
   30    FORMAT(T1,'C')
*     Define the upper end-fitting bottom surface.
                CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c          ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
                CURRENTSURFLABEL=0
                DO 40 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
   40           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   UEFBOTTOMSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   UEFBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
```

```
*     Define the upper end-fitting top surface.
                 CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c           ENDFITHEIGHT(DESNUM(COLUMN,ROW),1)+
     c           ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
                 CURRENTSURFLABEL=0
                 DO 50 V=1,(SN-1)
                    IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                          CURRENTSURFLABEL=V
                          EXIT
                       ENDIF
                    ENDIF
 50              CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
                    UEFTOPSURF=SN
                    SURFTYPESPEC(SN)='PZ'
                    SURFVALUESPEC(SN)=CURRENTSURF
                    SN=SN+1
                 ELSE
                    UEFTOPSURF=CURRENTSURFLABEL
                 ENDIF
*     Define the lower end-fitting top surface.
                 CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
                 CURRENTSURFLABEL=0
                 DO 60 V=1,(SN-1)
                    IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                          CURRENTSURFLABEL=V
                          EXIT
                       ENDIF
                    ENDIF
 60              CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
                    CRLEFTOPSURF=SN
                    SURFTYPESPEC(SN)='PZ'
                    SURFVALUESPEC(SN)=CURRENTSURF
                    SN=SN+1
                 ELSE
                    CRLEFTOPSURF=CURRENTSURFLABEL
                 ENDIF
*     Loop through the regions above the CR (i.e. the appropriate upper core
regions)
*     Define the upper region lower surface.
                 DO 100 REGION=1,NUMREGABOVECRA
*     Determine the current upper region's lower surface specification.
                    IF (REGION.EQ.1) THEN
                       REGIONTOPSURF=SYSTEMTOP
                       CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
     c                 REGABOVECRA(REGION,1)
                    ENDIF
                    CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
     c              REGABOVECRA(REGION,1)
                    IF (REGION.EQ.NUMREGABOVECRA) THEN
                       REGIONBOTTOMSURF=UEFTOPSURF
```

```
                    ELSE
                    CURRENTSURFLABEL=0
                    DO 70 V=1,(SN-1)
                       IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                  IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                             CURRENTSURFLABEL=V
                             EXIT
                          ENDIF
                       ENDIF
   70               CONTINUE
                    IF (CURRENTSURFLABEL.EQ.0) THEN
                       REGIONBOTTOMSURF=SN
                       SURFTYPESPEC(SN)='PZ'
                       SURFVALUESPEC(SN)=CURRENTSURF
                       SN=SN+1
                    ELSE
                       REGIONBOTTOMSURF=CURRENTSURFLABEL
                    ENDIF
                    ENDIF
*     Write the cell specification for the CR universe upper region.
                 IF (REGION.EQ.1) THEN
                    WRITE(30,80) LN, FRUREGIONML(COLUMN,ROW,REGION),
    c               (-1*REGABOVECRA(REGION,2)),
    c               REGIONBOTTOMSURF, CRAUNIV(COLUMN,ROW), REGION
   80               FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
    c               ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                    LN=LN+1
                    REGIONTOPSURF=REGIONBOTTOMSURF
                 ELSE
                    WRITE(30,90) LN, FRUREGIONML(COLUMN,ROW,REGION),
    c               (-1*REGABOVECRA(REGION,2)), (-1*REGIONTOPSURF),
    c               REGIONBOTTOMSURF, CRAUNIV(COLUMN,ROW), REGION
   90               FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
    c               ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                    LN=LN+1
                    REGIONTOPSURF=REGIONBOTTOMSURF
                 ENDIF
  100            CONTINUE
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
```

```
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
*
*
*
*
*
*
*
*    SPECIFICATIONS FOR HYBRID=1 AND GTSPLIT=1
*
*
*
*
*
*
*
*
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
*     Define the CR and GT section surfaces
      IF ((HYBRID.EQ.1).AND.(GTSPLIT.EQ.1)) THEN
      DO 200 SECT=1,NUMCRAXS(BANKNUM(COLUMN,ROW))
*     Define the CR absorber radius.
               CURRENTSURF=CRAXDIM(BANKNUM(COLUMN,ROW),1,SECT)
               CURRENTSURFLABEL=0
               DO 110 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
```

```
                ENDIF
 110         CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                CRAXABSSURF(SECT)=SN
                SURFTYPESPEC(SN)='CZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                CRAXABSSURF(SECT)=CURRENTSURFLABEL
             ENDIF
*     Define the CR absorber top surface.
             CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c       CRAXDIM(BANKNUM(COLUMN,ROW),4,SECT)+
     c       CRAXDIM(BANKNUM(COLUMN,ROW),5,SECT)
             IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
             ENDIF
             CURRENTSURFLABEL=0
             DO 120 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
 120         CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                CRAXABSTOPSURF(SECT)=SN
                SURFTYPESPEC(SN)='PZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                CRAXABSTOPSURF(SECT)=CURRENTSURFLABEL
             ENDIF
*     Define the CR absorber bottom surface.
             CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c       CRAXDIM(BANKNUM(COLUMN,ROW),4,SECT)
             IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
             ENDIF
             CURRENTSURFLABEL=0
             DO 130 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
 130         CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                CRAXABSBOTTOMSURF(SECT)=SN
                SURFTYPESPEC(SN)='PZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
```

```
              ELSE
                 CRAXABSBOTTOMSURF(SECT)=CURRENTSURFLABEL
              ENDIF
*     Define the CR cladding inner radius.
              CURRENTSURF=CRAXDIM(BANKNUM(COLUMN,ROW),2,SECT)
              CURRENTSURFLABEL=0
              DO 140 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'CZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 140          CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 CRAXCLADIRSURF(SECT)=SN
                 SURFTYPESPEC(SN)='CZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 CRAXCLADIRSURF(SECT)=CURRENTSURFLABEL
              ENDIF
*     Define the CR cladding outer radius.
              CURRENTSURF=CRAXDIM(BANKNUM(COLUMN,ROW),3,SECT)
              CURRENTSURFLABEL=0
              DO 150 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'CZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 150          CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 CRAXCLADORSURF(SECT)=SN
                 SURFTYPESPEC(SN)='CZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 CRAXCLADORSURF(SECT)=CURRENTSURFLABEL
              ENDIF
           TOPSECT=1
           DO 160 C=2,NUMCRAXS(BANKNUM(COLUMN,ROW))
              IF (SURFVALUESPEC(CRAXCLADTOPSURF(C)).GT.
    c         SURFVALUESPEC(CRAXCLADTOPSURF(TOPSECT))) THEN
                 TOPSECT=C
              ENDIF
 160       CONTINUE
*     Define the CR cladding top surface.
           IF (SECT.EQ.TOPSECT) THEN
              CURRENTSURF=CRAXDIM(BANKNUM(COLUMN,ROW),4,SECT)+
    c         CRAXDIM(BANKNUM(COLUMN,ROW),5,SECT)+
    c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
    c         CRADIM(BANKNUM(COLUMN,ROW),7)
```

```
          ELSE
            CURRENTSURF=CRAXDIM(BANKNUM(COLUMN,ROW),4,SECT)+
     c      CRAXDIM(BANKNUM(COLUMN,ROW),5,SECT)+
     c      ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
          ENDIF
            IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
              CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
            ENDIF
            CURRENTSURFLABEL=0
            DO 170 V=1,(SN-1)
              IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
                ENDIF
              ENDIF
 170        CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
              CRAXCLADTOPSURF(SECT)=SN
              SURFTYPESPEC(SN)='PZ'
              SURFVALUESPEC(SN)=CURRENTSURF
              SN=SN+1
            ELSE
              CRAXCLADTOPSURF(SECT)=CURRENTSURFLABEL
            ENDIF
        BOTSECT=1
        DO 180 C=2,NUMCRAXS(BANKNUM(COLUMN,ROW))
          IF (SURFVALUESPEC(CRAXCLADBOTTOMSURF(C)).LT.
     c    SURFVALUESPEC(CRAXCLADBOTTOMSURF(BOTSECT))) THEN
            BOTSECT=C
          ENDIF
 180    CONTINUE
*     Define the CR cladding bottom surface.
          IF (SECT.EQ.BOTSECT) THEN
            CURRENTSURF=CRAXDIM(BANKNUM(COLUMN,ROW),4,SECT)+
     c      ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c      CRADIM(BANKNUM(COLUMN,ROW),6)
          ELSE
            CURRENTSURF=CRAXDIM(BANKNUM(COLUMN,ROW),4,SECT)+
     c      ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
          ENDIF
            IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
              CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
            ENDIF
            CURRENTSURFLABEL=0
            DO 190 V=1,(SN-1)
              IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
                ENDIF
              ENDIF
 190        CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
```

```
                 CRAXCLADBOTTOMSURF(SECT)=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 CRAXCLADBOTTOMSURF(SECT)=CURRENTSURFLABEL
              ENDIF
 200 CONTINUE
     DO 250 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
*    Define the GT section top surface.
              CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),3,SECT)
              IF (CURRENTSURF.GT.SURFVALUESPEC(UEFTOPSURF)) THEN
                 CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
              ENDIF
              CURRENTSURFLABEL=0
              DO 210 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 210          CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 GTSECTTOPSURF(SECT)=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 GTSECTTOPSURF(SECT)=CURRENTSURFLABEL
              ENDIF
*    Define the GT section bottom surface.
              CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),4,SECT)
              CURRENTSURFLABEL=0
              DO 220 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 220          CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 GTSECTBOTSURF(SECT)=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 GTSECTBOTSURF(SECT)=CURRENTSURFLABEL
              ENDIF
*    Define the GT section outer radius surface.
              CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),2,SECT)
              CURRENTSURFLABEL=0
              DO 230 V=1,(SN-1)
```

```
                 IF (SURFTYPESPEC(V).EQ.'CZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 230           CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  GTSECTORSURF(SECT)=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTSECTORSURF(SECT)=CURRENTSURFLABEL
               ENDIF
*      Define the GT section inner radius surface.
               CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),1,SECT)
               CURRENTSURFLABEL=0
               DO 240 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 240           CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  GTSECTIRSURF(SECT)=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTSECTIRSURF(SECT)=CURRENTSURFLABEL
               ENDIF
 250 CONTINUE
      DO 670 SECT=1,NUMCRAXS(BANKNUM(COLUMN,ROW))
         IF (SURFVALUESPEC(CRAXABSBOTTOMSURF(SECT)).LT.
   c     SURFVALUESPEC(UEFTOPSURF)) THEN
*  Check Control Rod Absorber Material
               CRABSML=MN
               DO 280 C=1,CRABSAXMAT(BANKNUM(COLUMN,ROW),2,SECT)
                  IF (C.EQ.1) THEN
                     WRITE(200,260) CRABSML,
   c                 CRABSAXZAIDS(BANKNUM(COLUMN,ROW),C,SECT),
   c                 (-1*CRABSAXWTS(BANKNUM(COLUMN,ROW),C,SECT)),
   c                 ASSYID(COLUMN,ROW)
 260                 FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
   c                 '    $ Control Rod Absorber Material in Assembly ',
   c                 A5)
                  ELSE
                     WRITE(200,270)
   c                 CRABSAXZAIDS(BANKNUM(COLUMN,ROW),C,SECT),
   c                 (-1*CRABSAXWTS(BANKNUM(COLUMN,ROW),C,SECT))
 270                 FORMAT(T9,A9,3X,G14.6)
```

```
                     ENDIF
  280              CONTINUE
                   MN=MN+1
*     Write the CR absorber cell in this CR universe.
                WRITE(30,290) LN, CRABSML,
     c          (-1*CRABSAXMAT(BANKNUM(COLUMN,ROW),1,SECT)),
     c          (-1*CRAXABSSURF(SECT)),
     c          (-1*CRAXABSTOPSURF(SECT)), CRAXABSBOTTOMSURF(SECT),
     c          CRAUNIV(COLUMN,ROW)
  290           FORMAT(T1,I4,T6,I4,T11,F10.6,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Control rod absorber material')
                LN=LN+1
*     Write the absorber-to-cladding gap cell in this CR universe.
                WRITE(30,300) LN, (-1*CRAXCLADIRSURF(SECT)),
     c          CRAXABSSURF(SECT), (-1*CRAXABSTOPSURF(SECT)),
     c          CRAXABSBOTTOMSURF(SECT), CRAUNIV(COLUMN,ROW)
  300           FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Absorber-to-cladding gap')
                LN=LN+1
            ENDIF
*     Write the CR cladding cell in this CR universe.
*     Determine if the CR cladding material specification has
*     previously been defined.  If it has been previously defined, determine
*     the cladding material specification label.
                CLADMLUNIQUE=.TRUE.
                LEAVE=.FALSE.
                IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                   DO 320 RO=1,(ROW-1)
                      DO 310 CO=1,50
                         IF (BANKNUM(CO,RO).NE.0) THEN
                         IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA  ') THEN
                            IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.
     c                      CRCLADMAT(BANKNUM(CO,RO))) THEN
                               CLADMLUNIQUE=.FALSE.
                               LEAVE=.TRUE.
                               CRAXCLADML(COLUMN,ROW,SECT)=CRCLADML(CO,RO)
                               EXIT
                            ENDIF
                            IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.
     c                      CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                               CLADMLUNIQUE=.FALSE.
                               LEAVE=.TRUE.
                               CRAXCLADML(COLUMN,ROW,SECT)=
     c                         CRAXCLADML(CO,RO,SECT)
                               EXIT
                            ENDIF
                         ENDIF
                         ENDIF
  310                 CONTINUE
                      IF (LEAVE.EQ..TRUE.) THEN
                         EXIT
                      ENDIF
  320              CONTINUE
                   IF (LEAVE.EQ..FALSE.) THEN
```

```
               DO 340 RO=ROW,ROW
                  DO 330 CO=1,(COLUMN-1)
                  IF (BANKNUM(CO,RO).NE.0) THEN
                  IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA   ') THEN
                     IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c               CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                        CLADMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        CRAXCLADML(COLUMN,ROW,SECT)=CRCLADML(CO,RO)
                        EXIT
                     ENDIF
                     IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.
     c               CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                        CLADMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        CRAXCLADML(COLUMN,ROW,SECT)=
     c                  CRAXCLADML(CO,RO,SECT)
                        EXIT
                     ENDIF
                  ENDIF
                  ENDIF
 330              CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 340           CONTINUE
            ENDIF
         ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
            DO 360 RO=1,(ROW-1)
               DO 350 CO=1,50
                  IF (BANKNUM(CO,RO).NE.0) THEN
                  IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA   ') THEN
                     IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c               CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                        CLADMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        CRAXCLADML(COLUMN,ROW,SECT)=CRCLADML(CO,RO)
                        EXIT
                     ENDIF
                     IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.
     c               CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                        CLADMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        CRAXCLADML(COLUMN,ROW,SECT)=
     c                  CRAXCLADML(CO,RO,SECT)
                        EXIT
                     ENDIF
                  ENDIF
                  ENDIF
 350           CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
 360        CONTINUE
```

```
          ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
             DO 380 RO=1,1
                DO 370 CO=1,(COLUMN-1)
                   IF (BANKNUM(CO,RO).NE.0) THEN
                   IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA   ') THEN
                      IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                         CLADMLUNIQUE=.FALSE.
                         LEAVE=.TRUE.
                         CRAXCLADML(COLUMN,ROW,SECT)=CRCLADML(CO,RO)
                         EXIT
                      ENDIF
                      IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.
     c                CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                         CLADMLUNIQUE=.FALSE.
                         LEAVE=.TRUE.
                         CRAXCLADML(COLUMN,ROW,SECT)=
     c                   CRAXCLADML(CO,RO,SECT)
                         EXIT
                      ENDIF
                   ENDIF
                   ENDIF
 370            CONTINUE
                IF (LEAVE.EQ..TRUE.) THEN
                   EXIT
                ENDIF
 380         CONTINUE
          ENDIF
       IF (SURFVALUESPEC(CRAXCLADBOTTOMSURF(SECT)).LT.
     c SURFVALUESPEC(UEFTOPSURF)) THEN
          IF (CLADMLUNIQUE.EQ..TRUE.) THEN
             CRAXCLADML(COLUMN,ROW,SECT)=MN
* Check Control Rod Cladding
                IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.1) THEN
                   DO 390 C=1,2
                      IF (C.EQ.1) THEN
                         WRITE(200,9300) CRAXCLADML(COLUMN,ROW,SECT)
                      ELSEIF (C.EQ.2) THEN
                         WRITE(200,9301)
                         WRITE(200,7000)
                         WRITE(200,7001)
                         WRITE(200,7002)
                         WRITE(200,9302)
                         WRITE(200,7003)
                         WRITE(200,7004)
                         WRITE(200,7005)
                         WRITE(200,9303)
                         WRITE(200,9304)
                      ENDIF
 390               CONTINUE
                ELSEIF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT)
     c          .EQ.2) THEN
                   DO 400 C=1,2
                      IF (C.EQ.1) THEN
```

```
                    WRITE(200,9305) CRAXCLADML(COLUMN,ROW,SECT)
                 ELSEIF (C.EQ.2) THEN
                    WRITE(200,9306)
                    WRITE(200,9307)
                    WRITE(200,9308)
                    WRITE(200,9309)
                    WRITE(200,9310)
                    WRITE(200,7006)
                    WRITE(200,7007)
                    WRITE(200,7008)
                    WRITE(200,9311)
                    WRITE(200,9312)
                    WRITE(200,7009)
                    WRITE(200,7010)
                    WRITE(200,7011)
                    WRITE(200,9313)
                    WRITE(200,7012)
                    WRITE(200,7013)
                    WRITE(200,7014)
                    WRITE(200,7015)
                 ENDIF
400           CONTINUE
           ELSEIF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT)
     c     .EQ.3) THEN
              DO 410 C=1,2
                 IF (C.EQ.1) THEN
                    WRITE(200,9314) CRAXCLADML(COLUMN,ROW,SECT)
                 ELSEIF (C.EQ.2) THEN
                    WRITE(200,9315)
                    WRITE(200,9316)
                    WRITE(200,9317)
                    WRITE(200,9318)
                    WRITE(200,7016)
                    WRITE(200,7017)
                    WRITE(200,7018)
                    WRITE(200,9319)
                    WRITE(200,9320)
                    WRITE(200,7019)
                    WRITE(200,7020)
                    WRITE(200,7021)
                    WRITE(200,9321)
                    WRITE(200,7022)
                    WRITE(200,7023)
                    WRITE(200,7024)
                    WRITE(200,7025)
                    WRITE(200,9322)
                    WRITE(200,9323)
                    WRITE(200,9324)
                    WRITE(200,9325)
                    WRITE(200,9326)
                    WRITE(200,9327)
                    WRITE(200,7026)
                    WRITE(200,9328)
                    WRITE(200,9329)
```

```
                      WRITE(200,9330)
                    ENDIF
 410              CONTINUE
                ENDIF
              MN=MN+1
            ENDIF
            IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.1) THEN
              CLADRHO=6.56
            ELSEIF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.2) THEN
              CLADRHO=7.90
            ELSEIF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.3) THEN
              CLADRHO=8.19
            ENDIF
            WRITE(30,420) LN, CRAXCLADML(COLUMN,ROW,SECT),
     c      (-1*CLADRHO), CRAXCLADIRSURF(SECT),
     c      (-1*CRAXCLADORSURF(SECT)), (-1*CRAXCLADTOPSURF(SECT)),
     c      CRAXCLADBOTTOMSURF(SECT), CRAUNIV(COLUMN,ROW)
 420        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Control rod cladding')
            LN=LN+1
          ENDIF
*     Write the CR upper plenum cell in this CR universe.
*     Determine if the CR upper plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the upper plenum material specification label.
            CRUPMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
              DO 440 RO=1,(ROW-1)
                DO 430 CO=1,50
                  IF (BANKNUM(CO,RO).NE.0) THEN
                    IF (BANKNUM(COLUMN,ROW).EQ.
     c                BANKNUM(CO,RO)) THEN
                      CRUPMLUNIQUE=.FALSE.
                      LEAVE=.TRUE.
                      CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                      EXIT
                    ENDIF
                  ENDIF
 430            CONTINUE
                IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
                ENDIF
 440          CONTINUE
              IF (LEAVE.EQ..FALSE.) THEN
                DO 460 RO=ROW,ROW
                  DO 450 CO=1,(COLUMN-1)
                    IF (BANKNUM(CO,RO).NE.0) THEN
                      IF (BANKNUM(COLUMN,ROW).EQ.
     c                BANKNUM(CO,RO)) THEN
                        CRUPMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                        EXIT
```

```
                        ENDIF
                     ENDIF
 450              CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 460           CONTINUE
            ENDIF
         ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
            DO 480 RO=1,(ROW-1)
               DO 470 CO=1,50
                  IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKNUM(COLUMN,ROW).EQ.
   c                    BANKNUM(CO,RO)) THEN
                        CRUPMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
 470           CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
 480        CONTINUE
         ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
            DO 500 RO=1,1
               DO 490 CO=1,(COLUMN-1)
                  IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKNUM(COLUMN,ROW).EQ.
   c                    BANKNUM(CO,RO)) THEN
                        CRUPMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
 490           CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
 500        CONTINUE
         ENDIF
      IF (SURFVALUESPEC(CRAXABSTOPSURF(TOPSECT)).LT.
   c  SURFVALUESPEC(UEFTOPSURF)) THEN
         IF (CRUPMLUNIQUE.EQ..TRUE.) THEN
            CRUPML(COLUMN,ROW)=MN
*  Check Control Rod Upper Plenum Regions
               DO 530 C=1,CRUPLENMAT(BANKNUM(COLUMN,ROW),2)
                  IF (C.EQ.1) THEN
                     WRITE(200,510) CRUPML(COLUMN,ROW),
   c                 CRUPZS(BANKNUM(COLUMN,ROW),C),
   c                 (-1*CRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 510                 FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
```

```
     c                    '   $ Control Rod Upper Plenum')
                       ELSE
                          WRITE(200,520)
     c                    CRUPZS(BANKNUM(COLUMN,ROW),C),
     c                    (-1*CRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 520                      FORMAT(T9,A9,3X,G14.6)
                       ENDIF
 530                CONTINUE
                 MN=MN+1
              ENDIF
              WRITE(30,540) LN, CRUPML(COLUMN,ROW),
     c        (-1*CRUPLENMAT(BANKNUM(COLUMN,ROW),1)),
     c        CRAXABSTOPSURF(TOPSECT),
     c        (-1*CRAXCLADTOPSURF(TOPSECT)),
     c        (-1*CRAXCLADIRSURF(TOPSECT)),
     c        CRAUNIV(COLUMN,ROW)
 540          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Control rod upper plenum')
              LN=LN+1
           ENDIF
*     Write the CR lower plenum cell in this CR universe.
*     Determine if the CR lower plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the lower plenum material specification label.
              CRLPMLUNIQUE=.TRUE.
              LEAVE=.FALSE.
              IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                 DO 560 RO=1,(ROW-1)
                    DO 550 CO=1,50
                       IF (BANKNUM(CO,RO).NE.0) THEN
                          IF (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO)) THEN
                             CRLPMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                             EXIT
                          ENDIF
                       ENDIF
 550                CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
 560             CONTINUE
                 IF (LEAVE.EQ..FALSE.) THEN
                    DO 580 RO=ROW,ROW
                       DO 570 CO=1,(COLUMN-1)
                          IF (BANKNUM(CO,RO).NE.0) THEN
                             IF (BANKNUM(COLUMN,ROW).EQ.
     c                       BANKNUM(CO,RO)) THEN
                                CRLPMLUNIQUE=.FALSE.
                                LEAVE=.TRUE.
                                CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                                EXIT
                             ENDIF
                          ENDIF
```

```
570               CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
580            CONTINUE
            ENDIF
         ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
            DO 600 RO=1,(ROW-1)
               DO 590 CO=1,50
                  IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKNUM(COLUMN,ROW).EQ.
   c                    BANKNUM(CO,RO)) THEN
                        CRLPMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
590            CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
600         CONTINUE
         ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
            DO 620 RO=1,1
               DO 610 CO=1,(COLUMN-1)
                  IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKNUM(COLUMN,ROW).EQ.
   c                    BANKNUM(CO,RO)) THEN
                        CRLPMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
610            CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
620         CONTINUE
         ENDIF
      IF (SURFVALUESPEC(CRAXCLADBOTTOMSURF(BOTSECT)).LT.
   c  SURFVALUESPEC(UEFTOPSURF)) THEN
         IF (CRLPMLUNIQUE.EQ..TRUE.) THEN
            CRLPML(COLUMN,ROW)=MN
*  Check Control Rod Lower Plenum Regions
               DO 650 C=1,CRLPLENMAT(BANKNUM(COLUMN,ROW),2)
                  IF (C.EQ.1) THEN
                     WRITE(200,630) CRLPML(COLUMN,ROW),
   c                 CRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
   c                 (-1*CRLPLENWTS(BANKNUM(COLUMN,ROW),C))
630                  FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
   c                 '    $ Control Rod Lower Plenum')
                  ELSE
```

```
                        WRITE(200,640)
     c                  CRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                  (-1*CRLPLENWTS(BANKNUM(COLUMN,ROW),C))
  640                   FORMAT(T9,A9,3X,G14.6)
                      ENDIF
  650               CONTINUE
                  MN=MN+1
                ENDIF
                WRITE(30,660) LN, CRLPML(COLUMN,ROW),
     c          (-1*CRLPLENMAT(BANKNUM(COLUMN,ROW),1)),
     c          CRAXCLADBOTTOMSURF(BOTSECT),
     c          (-1*CRAXABSBOTTOMSURF(BOTSECT)),
     c          (-1*CRAXCLADIRSURF(BOTSECT)),
     c          CRAUNIV(COLUMN,ROW)
  660           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Control rod lower plenum')
                LN=LN+1
              ENDIF
  670       CONTINUE
*     Write the GT material cell
          DO 800 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
*     Determine if the GT material specification has
*     previously been defined.  If it has been previously defined, determine
*     the material specification label.
            CLADMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
              DO 690 RO=1,(ROW-1)
                DO 680 CO=1,50
                  IF ((DESNUM(CO,RO).NE.0).AND.
     c            (BANKNUM(CO,RO).EQ.0)) THEN
                    IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c              GTMAT(DESNUM(CO,RO))) THEN
                      CLADMLUNIQUE=.FALSE.
                      LEAVE=.TRUE.
                      GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                      EXIT
                    ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c              GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                      CLADMLUNIQUE=.FALSE.
                      LEAVE=.TRUE.
                      GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                      EXIT
                    ENDIF
                  ENDIF
  680           CONTINUE
                IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
                ENDIF
  690         CONTINUE
              IF (LEAVE.EQ..FALSE.) THEN
                DO 710 RO=ROW,ROW
                  DO 700 CO=1,(COLUMN-1)
                    IF ((DESNUM(CO,RO).NE.0).AND.
```

```
     c                    (BANKNUM(CO,RO).EQ.0)) THEN
                            IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                      GTMAT(DESNUM(CO,RO))) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                              EXIT
                            ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                      GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                              EXIT
                            ENDIF
                          ENDIF
  700                   CONTINUE
                        IF (LEAVE.EQ..TRUE.) THEN
                          EXIT
                        ENDIF
  710                 CONTINUE
                    ENDIF
                  ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                    DO 730 RO=1,(ROW-1)
                      DO 720 CO=1,50
                        IF ((DESNUM(CO,RO).NE.0).AND.
     c                    (BANKNUM(CO,RO).EQ.0)) THEN
                          IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                    GTMAT(DESNUM(CO,RO))) THEN
                            CLADMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                            EXIT
                          ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                    GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                            CLADMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                            EXIT
                          ENDIF
                        ENDIF
  720                 CONTINUE
                      IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                      ENDIF
  730               CONTINUE
                  ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                    DO 750 RO=1,1
                      DO 740 CO=1,(COLUMN-1)
                        IF ((DESNUM(CO,RO).NE.0).AND.
     c                    (BANKNUM(CO,RO).EQ.0)) THEN
                          IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                    GTMAT(DESNUM(CO,RO))) THEN
                            CLADMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
```

```
                    GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                    EXIT
                 ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c           GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                    CLADMLUNIQUE=.FALSE.
                    LEAVE=.TRUE.
                    GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                    EXIT
                 ENDIF
              ENDIF
 740       CONTINUE
           IF (LEAVE.EQ..TRUE.) THEN
              EXIT
           ENDIF
 750    CONTINUE
     ENDIF
     IF (CLADMLUNIQUE.EQ..TRUE.) THEN
        GTAXML(COLUMN,ROW,SECT)=MN
* Check Guide Tube Material
           IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.1) THEN
              DO 760 C=1,2
                 IF (C.EQ.1) THEN
                    WRITE(200,9300) GTAXML(COLUMN,ROW,SECT)
                 ELSEIF (C.EQ.2) THEN
                    WRITE(200,9301)
                    WRITE(200,7000)
                    WRITE(200,7001)
                    WRITE(200,7002)
                    WRITE(200,9302)
                    WRITE(200,7003)
                    WRITE(200,7004)
                    WRITE(200,7005)
                    WRITE(200,9303)
                    WRITE(200,9304)
                 ENDIF
 760          CONTINUE
           ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT)
     c     .EQ.2) THEN
              DO 770 C=1,2
                 IF (C.EQ.1) THEN
                    WRITE(200,9305) GTAXML(COLUMN,ROW,SECT)
                 ELSEIF (C.EQ.2) THEN
                    WRITE(200,9306)
                    WRITE(200,9307)
                    WRITE(200,9308)
                    WRITE(200,9309)
                    WRITE(200,9310)
                    WRITE(200,7006)
                    WRITE(200,7007)
                    WRITE(200,7008)
                    WRITE(200,9311)
                    WRITE(200,9312)
                    WRITE(200,7009)
                    WRITE(200,7010)
```

```
                        WRITE(200,7011)
                        WRITE(200,9313)
                        WRITE(200,7012)
                        WRITE(200,7013)
                        WRITE(200,7014)
                        WRITE(200,7015)
                     ENDIF
 770              CONTINUE
               ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT)
    c          .EQ.3) THEN
                  DO 780 C=1,2
                     IF (C.EQ.1) THEN
                        WRITE(200,9314) GTAXML(COLUMN,ROW,SECT)
                     ELSEIF (C.EQ.2) THEN
                        WRITE(200,9315)
                        WRITE(200,9316)
                        WRITE(200,9317)
                        WRITE(200,9318)
                        WRITE(200,7016)
                        WRITE(200,7017)
                        WRITE(200,7018)
                        WRITE(200,9319)
                        WRITE(200,9320)
                        WRITE(200,7019)
                        WRITE(200,7020)
                        WRITE(200,7021)
                        WRITE(200,9321)
                        WRITE(200,7022)
                        WRITE(200,7023)
                        WRITE(200,7024)
                        WRITE(200,7025)
                        WRITE(200,9322)
                        WRITE(200,9323)
                        WRITE(200,9324)
                        WRITE(200,9325)
                        WRITE(200,9326)
                        WRITE(200,9327)
                        WRITE(200,7026)
                        WRITE(200,9328)
                        WRITE(200,9329)
                        WRITE(200,9330)
                     ENDIF
 780              CONTINUE
               ENDIF
            MN=MN+1
         ENDIF
         IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.1) THEN
            CLADRHO=6.56
         ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.2) THEN
            CLADRHO=7.90
         ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.3) THEN
            CLADRHO=8.19
         ENDIF
         WRITE(30,790) LN, GTAXML(COLUMN,ROW,SECT), (-1*CLADRHO),
```

```
     c      GTSECTIRSURF(SECT),
     c      (-1*GTSECTORSURF(SECT)), (-1*GTSECTTOPSURF(SECT)),
     c      GTSECTBOTSURF(SECT), CRAUNIV(COLUMN,ROW)
 790        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,'   $ Guide tube')
            LN=LN+1
 800      CONTINUE
*     Loop through the spacer and moderator regions along the axial
*     length of the GT (from top to bottom).
          SPACHEIGHT=0.0
          DO 810 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
             SPACHEIGHT=SPACHEIGHT+SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
 810      CONTINUE
          DO 1000 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
*     Define the homogenized spacer region bounding surfaces.
            IF (SPN.EQ.1) THEN
               SPACERTOPSURF=UEFBOTTOMSURF
               CURRENTSURF=SURFVALUESPEC(UEFBOTTOMSURF)-
     c         SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
               CURRENTSURFLABEL=0
               DO 820 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 820           CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  SPACERBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  SPACERBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
               WATERREGIONTOPSURF=SPACERBOTTOMSURF
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
               CURRENTSURFLABEL=0
               DO 830 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 830           CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  WATERREGIONBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
```

```
            ENDIF
         ELSEIF ((SPN.NE.1).AND.(SPN.NE.
     c   NUMOFSPACERS(DESNUM(COLUMN,ROW)))) THEN
            SPACERTOPSURF=WATERREGIONBOTTOMSURF
            CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c      SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
            CURRENTSURFLABEL=0
            DO 840 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 840        CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               SPACERBOTTOMSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               SPACERBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
            WATERREGIONTOPSURF=SPACERBOTTOMSURF
            CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
            CURRENTSURFLABEL=0
            DO 850 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 850        CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               WATERREGIONBOTTOMSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
         ELSEIF (SPN.EQ.NUMOFSPACERS(DESNUM(COLUMN,ROW))) THEN
            SPACERTOPSURF=WATERREGIONBOTTOMSURF
            CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c      SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
            CURRENTSURFLABEL=0
            DO 860 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
```

```
860         CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               SPACERBOTTOMSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               SPACERBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
            WATERREGIONTOPSURF=SPACERBOTTOMSURF
            WATERREGIONBOTTOMSURF=NODEBOTTOMSURF
         ENDIF
*     Write the current homogenized spacer region cell in this GT universe.
         DO 990 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
          IF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c    SURFVALUESPEC(SPACERTOPSURF)).AND.
     c    (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c    SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
          WRITE(30,870) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c    (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c    GTSECTORSURF(SECT),
     c    (-1*SPACERTOPSURF), SPACERBOTTOMSURF, CRAUNIV(COLUMN,ROW),
     c    SPN
870       FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c    ' IMP:N=1 U=',I4,
     c    '   $ Homogenized region for spacer ',I2)
          LN=LN+1
          ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c    SURFVALUESPEC(SPACERTOPSURF)).AND.
     c    (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c    SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
          WRITE(30,880) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c    (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c    GTSECTORSURF(SECT),
     c    (-1*SPACERTOPSURF), SPACERBOTTOMSURF, CRAUNIV(COLUMN,ROW),
     c    SPN
880       FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c    ' IMP:N=1 U=',I4,
     c    '   $ Homogenized region for spacer ',I2)
          LN=LN+1
          ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c    SURFVALUESPEC(SPACERTOPSURF)).AND.
     c    (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c    SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
          WRITE(30,890) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c    (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c    GTSECTORSURF(SECT),
     c    (-1*SPACERTOPSURF), SPACERBOTTOMSURF, CRAUNIV(COLUMN,ROW),
     c    SPN
890       FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c    ' IMP:N=1 U=',I4,
     c    '   $ Homogenized region for spacer ',I2)
          LN=LN+1
          ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
```

```
     c        SURFVALUESPEC(SPACERTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c        SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
              WRITE(30,900) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c        (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c        GTSECTORSURF(SECT),
     c        (-1*SPACERTOPSURF), SPACERBOTTOMSURF, CRAUNIV(COLUMN,ROW),
     c        SPN
  900         FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,
     c        '    $ Homogenized region for spacer ',I2)
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(SPACERTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).GT.
     c        SURFVALUESPEC(SPACERBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(SPACERTOPSURF))) THEN
              WRITE(30,910) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c        (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c        GTSECTORSURF(SECT),
     c        (-1*SPACERTOPSURF), GTSECTBOTSURF(SECT),
     c        CRAUNIV(COLUMN,ROW), SPN
  910         FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,
     c        '    $ Homogenized region for spacer ',I2)
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).LT.
     c        SURFVALUESPEC(SPACERTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(SPACERBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
              WRITE(30,920) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c        (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c        GTSECTORSURF(SECT),
     c        (-1*GTSECTTOPSURF(SECT)), SPACERBOTTOMSURF,
     c        CRAUNIV(COLUMN,ROW), SPN
  920         FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,
     c        '    $ Homogenized region for spacer ',I2)
              LN=LN+1
               ENDIF
*      Write the water region cell below the current homogenized spacer cell
in this GT universe.
               IF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,930) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        CRAUNIV(COLUMN,ROW)
  930         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
```

```
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,940) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        CRAUNIV(COLUMN,ROW)
 940          FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,950) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        CRAUNIV(COLUMN,ROW)
 950          FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,960) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        CRAUNIV(COLUMN,ROW)
 960          FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).GT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(WATERREGIONTOPSURF))) THEN
              WRITE(30,970) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), GTSECTBOTSURF(SECT),
     c        CRAUNIV(COLUMN,ROW)
 970          FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).LT.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
```

```
              WRITE(30,980) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*GTSECTTOPSURF(SECT)), WATERREGIONBOTTOMSURF,
     c        CRAUNIV(COLUMN,ROW)
 980          FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'   $ Borated moderator region')
              LN=LN+1
               ENDIF
 990          CONTINUE
1000      CONTINUE
*  Write the moderator inside of the GT in the CR universe
        DO 1060 CRSECT=1,NUMCRAXS(BANKNUM(COLUMN,ROW))
          DO 1050 GTSECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
              IF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c        SURFVALUESPEC(CRAXCLADTOPSURF(CRSECT))).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LE.
     c        SURFVALUESPEC(CRAXCLADBOTTOMSURF(CRSECT)))) THEN
*       Write the moderator cells within the GT in this CR universe.
                WRITE(30,1010) LN, BMODML, (-1*MODDENSITY),
     c          (-1*GTSECTIRSURF(GTSECT)),
     c          CRAXCLADORSURF(CRSECT), (-1*CRAXCLADTOPSURF(CRSECT)),
     c          CRAXCLADBOTTOMSURF(CRSECT),
     c          CRAUNIV(COLUMN,ROW)
1010            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '    $ Borated moderator inside guide tube')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c        SURFVALUESPEC(CRAXCLADTOPSURF(CRSECT))).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LT.
     c        SURFVALUESPEC(CRAXCLADTOPSURF(CRSECT))).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).GT.
     c        SURFVALUESPEC(CRAXCLADBOTTOMSURF(CRSECT)))) THEN
                WRITE(30,1020) LN, BMODML, (-1*MODDENSITY),
     c          (-1*GTSECTIRSURF(GTSECT)),
     c          CRAXCLADORSURF(CRSECT), (-1*CRAXCLADTOPSURF(CRSECT)),
     c          GTSECTBOTSURF(GTSECT),
     c          CRAUNIV(COLUMN,ROW)
1020            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '    $ Borated moderator inside guide tube')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).LT.
     c        SURFVALUESPEC(CRAXCLADTOPSURF(CRSECT))).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LE.
     c        SURFVALUESPEC(CRAXCLADBOTTOMSURF(CRSECT))).AND.
     c        (SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GT.
     c        SURFVALUESPEC(CRAXCLADBOTTOMSURF(CRSECT)))) THEN
                WRITE(30,1030) LN, BMODML, (-1*MODDENSITY),
     c          (-1*GTSECTIRSURF(GTSECT)),
     c          CRAXCLADORSURF(CRSECT), (-1*GTSECTTOPSURF(GTSECT)),
     c          CRAXCLADBOTTOMSURF(CRSECT),
     c          CRAUNIV(COLUMN,ROW)
1030            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
```

```
     c          ' IMP:N=1 U=',I3,
     c          '    $ Borated moderator inside guide tube')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).LT.
     c       SURFVALUESPEC(CRAXCLADTOPSURF(CRSECT))).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).GT.
     c       SURFVALUESPEC(CRAXCLADBOTTOMSURF(CRSECT)))) THEN
                WRITE(30,1040) LN, BMODML, (-1*MODDENSITY),
     c          (-1*GTSECTIRSURF(GTSECT)),
     c          CRAXCLADORSURF(CRSECT), (-1*GTSECTTOPSURF(GTSECT)),
     c          GTSECTBOTSURF(GTSECT),
     c          CRAUNIV(COLUMN,ROW)
 1040           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '    $ Borated moderator inside guide tube')
                LN=LN+1
             ENDIF
 1050     CONTINUE
 1060 CONTINUE
*   Determine the axial GT section which contains the lowest CR axial section
        DO 1070 GTSECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
          IF ((SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LT.
     c    SURFVALUESPEC(CRAXCLADBOTTOMSURF
     c    (NUMCRAXS(BANKNUM(COLUMN,ROW))))).AND.
     c    (SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c    SURFVALUESPEC(CRAXCLADBOTTOMSURF
     c    (NUMCRAXS(BANKNUM(COLUMN,ROW)))))) THEN
             BGT=GTSECT
             EXIT
          ENDIF
 1070 CONTINUE
        DO 1100 GTSECT=BGT,NUMOFGTAXS(DESNUM(COLUMN,ROW))
          IF (GTSECT.EQ.BGT) THEN
             WRITE(30,1080) LN, BMODML, (-1*MODDENSITY),
     c       (-1*GTSECTIRSURF(GTSECT)),
     c       (-1*CRAXCLADBOTTOMSURF(NUMCRAXS(BANKNUM(COLUMN,ROW)))),
     c       GTSECTBOTSURF(GTSECT),
     c       CRAUNIV(COLUMN,ROW)
 1080        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,
     c       '    $ Borated moderator inside guide tube')
             LN=LN+1
          ELSE
             WRITE(30,1090) LN, BMODML, (-1*MODDENSITY),
     c       (-1*GTSECTIRSURF(GTSECT)),
     c       (-1*GTSECTTOPSURF(GTSECT)),
     c       GTSECTBOTSURF(GTSECT),
     c       CRAUNIV(COLUMN,ROW)
 1090        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,
     c       '    $ Borated moderator inside guide tube')
             LN=LN+1
          ENDIF
 1100 CONTINUE
```

```
*     Write the lower end-fitting cell specification for this CR universe.
            GTBOTSURF=GTSECTBOTSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))
            IF (SURFVALUESPEC(GTBOTSURF).GE.
     c      ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)) THEN
               WRITE(30,1110) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*CRLEFTOPSURF),
     c         CRAUNIV(COLUMN,ROW)
 1110          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,' IMP:N=1 U=',I3,
     c         '   $ Lower end-fitting')
               LN=LN+1
            ELSE
               WRITE(30,1120) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*CRLEFTOPSURF),
     c         GTSECTORSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW))),
     c         CRAUNIV(COLUMN,ROW)
 1120          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Lower end-fitting')
               LN=LN+1
               WRITE(30,1130) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)),
     c         (-1*GTSECTBOTSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))),
     c         (-1*GTSECTORSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))),
     c         CRAUNIV(COLUMN,ROW)
 1130          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Lower end-fitting')
               LN=LN+1
            ENDIF
*     Write the upper end-fitting cell specification for this CR universe.
            GTTOPSURF=GTSECTTOPSURF(1)
            GTSECTORSURF(1)=GTSECTORSURF(1)
            GTSECTIRSURF(1)=GTSECTIRSURF(1)
            CRCLADTOPSURF=CRAXCLADTOPSURF(1)
            CRCLADIRSURF=CRAXCLADIRSURF(1)
            CRCLADORSURF=CRAXCLADORSURF(1)
            IF ((SURFVALUESPEC(GTTOPSURF).GE.
     c      SURFVALUESPEC(UEFTOPSURF)).AND.
     c      (SURFVALUESPEC(CRCLADTOPSURF).GE.
     c      SURFVALUESPEC(UEFTOPSURF))) THEN
               WRITE(30,1131) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c         (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 1131          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
               LN=LN+1
            ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c      SURFVALUESPEC(UEFTOPSURF)).AND.
     c      (SURFVALUESPEC(GTTOPSURF).GT.
     c      SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(CRCLADTOPSURF).GE.
     c      SURFVALUESPEC(UEFTOPSURF))) THEN
               WRITE(30,1132) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c         (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 1132          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
```

```
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
                WRITE(30,1133) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), CRCLADORSURF, (-1*GTSECTORSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
 1133           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTTOPSURF).LE.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).GE.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,1134) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), CRCLADORSURF, CRAUNIV(COLUMN,ROW)
 1134           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTTOPSURF).GE.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,1135) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 1135           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
                WRITE(30,1136) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
 1136           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).GT.
     c       SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,1137) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 1137           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
                WRITE(30,1138) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
```

```
     c          (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
 1138           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
                WRITE(30,1139) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), GTSECTIRSURF(1),
     c          (-1*GTSECTORSURF(1)), CRAUNIV(COLUMN,ROW)
 1139           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTTOPSURF).EQ.
     c       SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,1140) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 1140           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
                WRITE(30,1141) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
 1141           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c       SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,1142) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 1142           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
                WRITE(30,1143) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
 1143           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
                WRITE(30,1144) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
```

```
     c         (-1*CRCLADTOPSURF), CRCLADORSURF,
     c         (-1*GTSECTORSURF(1)), CRAUNIV(COLUMN,ROW)
 1144          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
               LN=LN+1
            ELSEIF ((SURFVALUESPEC(GTTOPSURF).LE.
     c      SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c      SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c      SURFVALUESPEC(UEFTOPSURF))) THEN
               WRITE(30,1145) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c         (-1*UEFTOPSURF), CRCLADORSURF, CRAUNIV(COLUMN,ROW)
 1145          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
               LN=LN+1
               WRITE(30,1146) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c         (-1*UEFTOPSURF), (-1*CRCLADORSURF),
     c         CRAUNIV(COLUMN,ROW)
 1146          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
               LN=LN+1
            ELSEIF ((SURFVALUESPEC(CRCLADTOPSURF).LE.
     c      SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(GTTOPSURF).GE.
     c      SURFVALUESPEC(UEFTOPSURF))) THEN
               WRITE(30,1147) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c         (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 1147          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
               LN=LN+1
               WRITE(30,1148) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c         (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c         CRAUNIV(COLUMN,ROW)
 1148          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
               LN=LN+1
            ELSEIF ((SURFVALUESPEC(CRCLADTOPSURF).LE.
     c      SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(GTTOPSURF).LT.
     c      SURFVALUESPEC(UEFTOPSURF)).AND.
     c      (SURFVALUESPEC(GTTOPSURF).GT.
     c      SURFVALUESPEC(UEFBOTTOMSURF))) THEN
               WRITE(30,1149) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c         (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 1149          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
               LN=LN+1
               WRITE(30,1150) LN, FRUEFML(COLUMN,ROW),
```

```
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
 1150           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
                WRITE(30,1151) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c          GTSECTIRSURF(1), CRAUNIV(COLUMN,ROW)
 1151           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
            ELSEIF ((SURFVALUESPEC(CRCLADTOPSURF).LE.
     c      SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(GTTOPSURF).LE.
     c      SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                WRITE(30,1152) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), CRAUNIV(COLUMN,ROW)
 1152           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
            ENDIF
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
****************************************************************************
*
*
*
*
*
*
*
*    SPECIFICATIONS FOR HYBRID=1 AND GTSPLIT<>1
*
```

```
*
*
*
*
*
*
*
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
      ELSEIF ((HYBRID.EQ.1).AND.(GTSPLIT.NE.1)) THEN
      DO 1280 SECT=1,NUMCRAXS(BANKNUM(COLUMN,ROW))
*     Define the CR absorber radius.
               CURRENTSURF=CRAXDIM(BANKNUM(COLUMN,ROW),1,SECT)
               CURRENTSURFLABEL=0
               DO 1190 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
1190           CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  CRAXABSSURF(SECT)=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  CRAXABSSURF(SECT)=CURRENTSURFLABEL
               ENDIF
*     Define the CR absorber top surface.
               CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c         CRAXDIM(BANKNUM(COLUMN,ROW),4,SECT)+
     c         CRAXDIM(BANKNUM(COLUMN,ROW),5,SECT)
               IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
```

```
                    CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                 ENDIF
                 CURRENTSURFLABEL=0
                 DO 1200 V=1,(SN-1)
                    IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                          CURRENTSURFLABEL=V
                          EXIT
                       ENDIF
                    ENDIF
 1200            CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
                    CRAXABSTOPSURF(SECT)=SN
                    SURFTYPESPEC(SN)='PZ'
                    SURFVALUESPEC(SN)=CURRENTSURF
                    SN=SN+1
                 ELSE
                    CRAXABSTOPSURF(SECT)=CURRENTSURFLABEL
                 ENDIF
*     Define the CR absorber bottom surface.
                 CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c           CRAXDIM(BANKNUM(COLUMN,ROW),4,SECT)
                 IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                    CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                 ENDIF
                 CURRENTSURFLABEL=0
                 DO 1210 V=1,(SN-1)
                    IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                          CURRENTSURFLABEL=V
                          EXIT
                       ENDIF
                    ENDIF
 1210            CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
                    CRAXABSBOTTOMSURF(SECT)=SN
                    SURFTYPESPEC(SN)='PZ'
                    SURFVALUESPEC(SN)=CURRENTSURF
                    SN=SN+1
                 ELSE
                    CRAXABSBOTTOMSURF(SECT)=CURRENTSURFLABEL
                 ENDIF
*     Define the CR cladding inner radius.
                 CURRENTSURF=CRAXDIM(BANKNUM(COLUMN,ROW),2,SECT)
                 CURRENTSURFLABEL=0
                 DO 1220 V=1,(SN-1)
                    IF (SURFTYPESPEC(V).EQ.'CZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                          CURRENTSURFLABEL=V
                          EXIT
                       ENDIF
                    ENDIF
 1220            CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
```

```
                CRAXCLADIRSURF(SECT)=SN
                SURFTYPESPEC(SN)='CZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                CRAXCLADIRSURF(SECT)=CURRENTSURFLABEL
             ENDIF
*     Define the CR cladding outer radius.
             CURRENTSURF=CRAXDIM(BANKNUM(COLUMN,ROW),3,SECT)
             CURRENTSURFLABEL=0
             DO 1230 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'CZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
 1230        CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                CRAXCLADORSURF(SECT)=SN
                SURFTYPESPEC(SN)='CZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                CRAXCLADORSURF(SECT)=CURRENTSURFLABEL
             ENDIF
          TOPSECT=1
          DO 1240 C=2,NUMCRAXS(BANKNUM(COLUMN,ROW))
             IF (SURFVALUESPEC(CRAXCLADTOPSURF(C)).GT.
     c       SURFVALUESPEC(CRAXCLADTOPSURF(TOPSECT))) THEN
                TOPSECT=C
             ENDIF
 1240     CONTINUE
*     Define the CR cladding top surface.
           IF (SECT.EQ.TOPSECT) THEN
             CURRENTSURF=CRAXDIM(BANKNUM(COLUMN,ROW),4,SECT)+
     c       CRAXDIM(BANKNUM(COLUMN,ROW),5,SECT)+
     c       ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c       CRADIM(BANKNUM(COLUMN,ROW),7)
           ELSE
             CURRENTSURF=CRAXDIM(BANKNUM(COLUMN,ROW),4,SECT)+
     c       CRAXDIM(BANKNUM(COLUMN,ROW),5,SECT)+
     c       ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
           ENDIF
             IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
             ENDIF
             CURRENTSURFLABEL=0
             DO 1250 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
```

```
            ENDIF
 1250    CONTINUE
         IF (CURRENTSURFLABEL.EQ.0) THEN
            CRAXCLADTOPSURF(SECT)=SN
            SURFTYPESPEC(SN)='PZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
         ELSE
            CRAXCLADTOPSURF(SECT)=CURRENTSURFLABEL
         ENDIF
      BOTSECT=1
      DO 1260 C=2,NUMCRAXS(BANKNUM(COLUMN,ROW))
         IF (SURFVALUESPEC(CRAXCLADBOTTOMSURF(C)).LT.
    c    SURFVALUESPEC(CRAXCLADBOTTOMSURF(BOTSECT))) THEN
            BOTSECT=C
         ENDIF
 1260 CONTINUE
*     Define the CR cladding bottom surface.
         IF (SECT.EQ.BOTSECT) THEN
            CURRENTSURF=CRAXDIM(BANKNUM(COLUMN,ROW),4,SECT)+
    c       ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
    c       CRADIM(BANKNUM(COLUMN,ROW),6)
         ELSE
            CURRENTSURF=CRAXDIM(BANKNUM(COLUMN,ROW),4,SECT)+
    c       ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
         ENDIF
            IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
               CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
            ENDIF
            CURRENTSURFLABEL=0
            DO 1270 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 1270       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               CRAXCLADBOTTOMSURF(SECT)=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               CRAXCLADBOTTOMSURF(SECT)=CURRENTSURFLABEL
            ENDIF
 1280 CONTINUE
*     Define the GT top surface.
            CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),3)
            IF (CURRENTSURF.GT.SURFVALUESPEC(UEFTOPSURF)) THEN
               CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
            ENDIF
            CURRENTSURFLABEL=0
            DO 1290 V=1,(SN-1)
```

```
            IF (SURFTYPESPEC(V).EQ.'PZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
1290     CONTINUE
         IF (CURRENTSURFLABEL.EQ.0) THEN
            GTTOPSURF=SN
            SURFTYPESPEC(SN)='PZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
         ELSE
            GTTOPSURF=CURRENTSURFLABEL
         ENDIF
*     Define the GT bottom surface.
         CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),4)
         CURRENTSURFLABEL=0
         DO 1300 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'PZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
1300     CONTINUE
         IF (CURRENTSURFLABEL.EQ.0) THEN
            GTBOTSURF=SN
            SURFTYPESPEC(SN)='PZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
         ELSE
            GTBOTSURF=CURRENTSURFLABEL
         ENDIF
*     Define the GT outer radius surface.
         CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),2)
         CURRENTSURFLABEL=0
         DO 1310 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'CZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
1310     CONTINUE
         IF (CURRENTSURFLABEL.EQ.0) THEN
            GTORSURF=SN
            SURFTYPESPEC(SN)='CZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
         ELSE
            GTORSURF=CURRENTSURFLABEL
         ENDIF
*     Define the GT inner radius surface.
```

```
                 CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),1)
                 CURRENTSURFLABEL=0
                 DO 1320 V=1,(SN-1)
                    IF (SURFTYPESPEC(V).EQ.'CZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                     ENDIF
                  ENDIF
 1320            CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
                    GTIRSURF=SN
                    SURFTYPESPEC(SN)='CZ'
                    SURFVALUESPEC(SN)=CURRENTSURF
                    SN=SN+1
                 ELSE
                    GTIRSURF=CURRENTSURFLABEL
                 ENDIF
*     Write the GT material cell in this GT universe.
*     Determine if the GT material specification has
*     previously been defined.  If it has been previously defined, determine
*     the material specification label.
              CLADMLUNIQUE=.TRUE.
              LEAVE=.FALSE.
              IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                 DO 1340 RO=1,(ROW-1)
                    DO 1330 CO=1,50
                       IF ((DESNUM(CO,RO).NE.0).AND.
     c                 (BANKNUM(CO,RO).EQ.0)) THEN
                          IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                    GTMAT(DESNUM(CO,RO))) THEN
                             CLADMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             GTML(COLUMN,ROW)=GTML(CO,RO)
                             EXIT
                          ENDIF
                       ENDIF
 1330               CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
 1340            CONTINUE
                 IF (LEAVE.EQ..FALSE.) THEN
                    DO 1360 RO=ROW,ROW
                       DO 1350 CO=1,(COLUMN-1)
                          IF ((DESNUM(CO,RO).NE.0).AND.
     c                    (BANKNUM(CO,RO).EQ.0)) THEN
                             IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                       GTMAT(DESNUM(CO,RO))) THEN
                                CLADMLUNIQUE=.FALSE.
                                LEAVE=.TRUE.
                                GTML(COLUMN,ROW)=GTML(CO,RO)
                                EXIT
                             ENDIF
```

```
                    ENDIF
 1350             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                  ENDIF
 1360           CONTINUE
              ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
              DO 1380 RO=1,(ROW-1)
                DO 1370 CO=1,50
                  IF ((DESNUM(CO,RO).NE.0).AND.
     c              (BANKNUM(CO,RO).EQ.0)) THEN
                    IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c              GTMAT(DESNUM(CO,RO))) THEN
                      CLADMLUNIQUE=.FALSE.
                      LEAVE=.TRUE.
                      GTML(COLUMN,ROW)=GTML(CO,RO)
                      EXIT
                    ENDIF
                  ENDIF
 1370           CONTINUE
                IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
                ENDIF
 1380         CONTINUE
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
              DO 1400 RO=1,1
                DO 1390 CO=1,(COLUMN-1)
                  IF ((DESNUM(CO,RO).NE.0).AND.
     c              (BANKNUM(CO,RO).EQ.0)) THEN
                    IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c              GTMAT(DESNUM(CO,RO))) THEN
                      CLADMLUNIQUE=.FALSE.
                      LEAVE=.TRUE.
                      GTML(COLUMN,ROW)=GTML(CO,RO)
                      EXIT
                    ENDIF
                  ENDIF
 1390           CONTINUE
                IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
                ENDIF
 1400         CONTINUE
            ENDIF
            IF (CLADMLUNIQUE.EQ..TRUE.) THEN
              GTML(COLUMN,ROW)=MN
*  Check Guide Tube Material
                IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.1) THEN
                  DO 1410 C=1,2
                    IF (C.EQ.1) THEN
                      WRITE(200,9300) GTML(COLUMN,ROW)
                    ELSEIF (C.EQ.2) THEN
                      WRITE(200,9301)
                      WRITE(200,7000)
```

```
                    WRITE(200,7001)
                    WRITE(200,7002)
                    WRITE(200,9302)
                    WRITE(200,7003)
                    WRITE(200,7004)
                    WRITE(200,7005)
                    WRITE(200,9303)
                    WRITE(200,9304)
                 ENDIF
1410          CONTINUE
           ELSEIF (GTMAT(DESNUM(COLUMN,ROW))
     c     .EQ.2) THEN
              DO 1420 C=1,2
                 IF (C.EQ.1) THEN
                    WRITE(200,9305) GTML(COLUMN,ROW)
                 ELSEIF (C.EQ.2) THEN
                    WRITE(200,9306)
                    WRITE(200,9307)
                    WRITE(200,9308)
                    WRITE(200,9309)
                    WRITE(200,9310)
                    WRITE(200,7006)
                    WRITE(200,7007)
                    WRITE(200,7008)
                    WRITE(200,9311)
                    WRITE(200,9312)
                    WRITE(200,7009)
                    WRITE(200,7010)
                    WRITE(200,7011)
                    WRITE(200,9313)
                    WRITE(200,7012)
                    WRITE(200,7013)
                    WRITE(200,7014)
                    WRITE(200,7015)
                 ENDIF
1420          CONTINUE
           ELSEIF (GTMAT(DESNUM(COLUMN,ROW))
     c     .EQ.3) THEN
              DO 1430 C=1,2
                 IF (C.EQ.1) THEN
                    WRITE(200,9314) GTML(COLUMN,ROW)
                 ELSEIF (C.EQ.2) THEN
                    WRITE(200,9315)
                    WRITE(200,9316)
                    WRITE(200,9317)
                    WRITE(200,9318)
                    WRITE(200,7016)
                    WRITE(200,7017)
                    WRITE(200,7018)
                    WRITE(200,9319)
                    WRITE(200,9320)
                    WRITE(200,7019)
                    WRITE(200,7020)
                    WRITE(200,7021)
```

```
                                       WRITE(200,9321)
                                       WRITE(200,7022)
                                       WRITE(200,7023)
                                       WRITE(200,7024)
                                       WRITE(200,7025)
                                       WRITE(200,9322)
                                       WRITE(200,9323)
                                       WRITE(200,9324)
                                       WRITE(200,9325)
                                       WRITE(200,9326)
                                       WRITE(200,9327)
                                       WRITE(200,7026)
                                       WRITE(200,9328)
                                       WRITE(200,9329)
                                       WRITE(200,9330)
                                    ENDIF
 1430                            CONTINUE
                              ENDIF
                           MN=MN+1
                        ENDIF
                        IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.1) THEN
                           CLADRHO=6.56
                        ELSEIF (GTMAT(DESNUM(COLUMN,ROW)).EQ.2) THEN
                           CLADRHO=7.90
                        ELSEIF (GTMAT(DESNUM(COLUMN,ROW)).EQ.3) THEN
                           CLADRHO=8.19
                        ENDIF
                        WRITE(30,1440) LN, GTML(COLUMN,ROW), (-1*CLADRHO),
     c                  GTIRSURF,
     c                  (-1*GTORSURF), (-1*GTTOPSURF), GTBOTSURF,
     c                  GTUNIV(COLUMN,ROW)
 1440                   FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c                  ' IMP:N=1 U=',I4,'   $ Guide tube')
                        LN=LN+1
                         SPACHEIGHT=0.0
*      Loop through the spacer and moderator regions along the axial
*      length of the GT (from top to bottom).
                      DO 1450 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
                         SPACHEIGHT=SPACHEIGHT+SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
 1450                 CONTINUE
                      DO 1530 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
*      Define the homogenized spacer region bounding surfaces.
                        IF (SPN.EQ.1) THEN
                           SPACERTOPSURF=UEFBOTTOMSURF
                           CURRENTSURF=SURFVALUESPEC(UEFBOTTOMSURF)-
     c                     SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                           CURRENTSURFLABEL=0
                           DO 1460 V=1,(SN-1)
                              IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                                    CURRENTSURFLABEL=V
                                    EXIT
                                 ENDIF
                              ENDIF
```

```
 1460         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 SPACERBOTTOMSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 SPACERBOTTOMSURF=CURRENTSURFLABEL
              ENDIF
              WATERREGIONTOPSURF=SPACERBOTTOMSURF
              CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
              CURRENTSURFLABEL=0
              DO 1470 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 1470         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 WATERREGIONBOTTOMSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
              ENDIF
           ELSEIF ((SPN.NE.1).AND.(SPN.NE.
     c     NUMOFSPACERS(DESNUM(COLUMN,ROW)))) THEN
              SPACERTOPSURF=WATERREGIONBOTTOMSURF
              CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c        SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
              CURRENTSURFLABEL=0
              DO 1480 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 1480         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 SPACERBOTTOMSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 SPACERBOTTOMSURF=CURRENTSURFLABEL
              ENDIF
              WATERREGIONTOPSURF=SPACERBOTTOMSURF
              CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
              CURRENTSURFLABEL=0
              DO 1490 V=1,(SN-1)
```

```
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 1490           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   WATERREGIONBOTTOMSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
             ELSEIF (SPN.EQ.NUMOFSPACERS(DESNUM(COLUMN,ROW))) THEN
                SPACERTOPSURF=WATERREGIONBOTTOMSURF
                CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c          SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                CURRENTSURFLABEL=0
                DO 1500 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 1500           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   SPACERBOTTOMSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   SPACERBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
                WATERREGIONTOPSURF=SPACERBOTTOMSURF
                WATERREGIONBOTTOMSURF=NODEBOTTOMSURF
             ENDIF
*     Write the current homogenized spacer region cell in this GT universe.
             WRITE(30,1510) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c       (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)), GTORSURF,
     c       (-1*SPACERTOPSURF), SPACERBOTTOMSURF, GTUNIV(COLUMN,ROW),
     c       SPN
 1510        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,
     c       '    $ Homogenized region for spacer ',I2)
             LN=LN+1
*     Write the water region cell below the current homogenized spacer cell
in this GT universe.
             WRITE(30,1520) LN, BMODML, (-1*MODDENSITY), GTORSURF,
     c       (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c       GTUNIV(COLUMN,ROW)
 1520        FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
```

```
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
 1530      CONTINUE
        DO 1950 SECT=1,NUMCRAXS(BANKNUM(COLUMN,ROW))
           IF (SURFVALUESPEC(CRAXABSBOTTOMSURF(SECT)).LT.
     c     SURFVALUESPEC(UEFTOPSURF)) THEN
*  Check Control Rod Absorber Material
                 CRABSML=MN
                 DO 1560 C=1,CRABSAXMAT(BANKNUM(COLUMN,ROW),2,SECT)
                    IF (C.EQ.1) THEN
                       WRITE(200,1540) CRABSML,
     c                 CRABSAXZAIDS(BANKNUM(COLUMN,ROW),C,SECT),
     c                 (-1*CRABSAXWTS(BANKNUM(COLUMN,ROW),C,SECT)),
     c                 ASSYID(COLUMN,ROW)
 1540                  FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                 '    $ Control Rod Absorber Material in Assembly ',
     c                 A5)
                    ELSE
                       WRITE(200,1550)
     c                 CRABSAXZAIDS(BANKNUM(COLUMN,ROW),C,SECT),
     c                 (-1*CRABSAXWTS(BANKNUM(COLUMN,ROW),C,SECT))
 1550                  FORMAT(T9,A9,3X,G14.6)
                    ENDIF
 1560            CONTINUE
                 MN=MN+1
*     Write the CR absorber cell in this CR universe.
              WRITE(30,1570) LN, CRABSML,
     c        (-1*CRABSAXMAT(BANKNUM(COLUMN,ROW),1,SECT)),
     c        (-1*CRAXABSSURF(SECT)),
     c        (-1*CRAXABSTOPSURF(SECT)), CRAXABSBOTTOMSURF(SECT),
     c        CRAUNIV(COLUMN,ROW)
 1570         FORMAT(T1,I4,T6,I4,T11,F10.6,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'    $ Control rod absorber material')
              LN=LN+1
*     Write the absorber-to-cladding gap cell in this CR universe.
              WRITE(30,1580) LN, (-1*CRAXCLADIRSURF(SECT)),
     c        CRAXABSSURF(SECT), (-1*CRAXABSTOPSURF(SECT)),
     c        CRAXABSBOTTOMSURF(SECT), CRAUNIV(COLUMN,ROW)
 1580         FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'    $ Absorber-to-cladding gap')
              LN=LN+1
           ENDIF
*     Write the CR cladding cell in this CR universe.
*     Determine if the CR cladding material specification has
*     previously been defined.  If it has been previously defined, determine
*     the cladding material specification label.
              CLADMLUNIQUE=.TRUE.
              LEAVE=.FALSE.
              IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                 DO 1600 RO=1,(ROW-1)
                    DO 1590 CO=1,50
                       IF (BANKNUM(CO,RO).NE.0) THEN
                       IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA  ') THEN
                          IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.
```

```
     c                     CRCLADMAT(BANKNUM(CO,RO))) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              CRAXCLADML(COLUMN,ROW,SECT)=CRCLADML(CO,RO)
                              EXIT
                           ENDIF
                           IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.
     c                     CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              CRAXCLADML(COLUMN,ROW,SECT)=
     c                        CRAXCLADML(CO,RO,SECT)
                              EXIT
                           ENDIF
                        ENDIF
                        ENDIF
 1590                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 1600             CONTINUE
                  IF (LEAVE.EQ..FALSE.) THEN
                     DO 1620 RO=ROW,ROW
                        DO 1610 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA  ') THEN
                           IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                     CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              CRAXCLADML(COLUMN,ROW,SECT)=CRCLADML(CO,RO)
                              EXIT
                           ENDIF
                           IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.
     c                     CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              CRAXCLADML(COLUMN,ROW,SECT)=
     c                        CRAXCLADML(CO,RO,SECT)
                              EXIT
                           ENDIF
                        ENDIF
                        ENDIF
 1610                   CONTINUE
                        IF (LEAVE.EQ..TRUE.) THEN
                           EXIT
                        ENDIF
 1620                CONTINUE
                  ENDIF
               ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                  DO 1640 RO=1,(ROW-1)
                     DO 1630 CO=1,50
                        IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA  ') THEN
```

```
                        IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRAXCLADML(COLUMN,ROW,SECT)=CRCLADML(CO,RO)
                           EXIT
                        ENDIF
                        IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.
     c                  CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRAXCLADML(COLUMN,ROW,SECT)=
     c                     CRAXCLADML(CO,RO,SECT)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
 1630             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 1640          CONTINUE
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
               DO 1660 RO=1,1
                  DO 1650 CO=1,(COLUMN-1)
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA   ') THEN
                        IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRAXCLADML(COLUMN,ROW,SECT)=CRCLADML(CO,RO)
                           EXIT
                        ENDIF
                        IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.
     c                  CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRAXCLADML(COLUMN,ROW,SECT)=
     c                     CRAXCLADML(CO,RO,SECT)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
 1650             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 1660          CONTINUE
            ENDIF
         IF (SURFVALUESPEC(CRAXCLADBOTTOMSURF(SECT)).LT.
     c   SURFVALUESPEC(UEFTOPSURF)) THEN
            IF (CLADMLUNIQUE.EQ..TRUE.) THEN
               CRAXCLADML(COLUMN,ROW,SECT)=MN
```

```
*    Check Control Rod Cladding
               IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.1) THEN
                 DO 1670 C=1,2
                   IF (C.EQ.1) THEN
                     WRITE(200,9300) CRAXCLADML(COLUMN,ROW,SECT)
                   ELSEIF (C.EQ.2) THEN
                     WRITE(200,9301)
                     WRITE(200,7000)
                     WRITE(200,7001)
                     WRITE(200,7002)
                     WRITE(200,9302)
                     WRITE(200,7003)
                     WRITE(200,7004)
                     WRITE(200,7005)
                     WRITE(200,9303)
                     WRITE(200,9304)
                   ENDIF
 1670            CONTINUE
               ELSEIF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT)
     c         .EQ.2) THEN
                 DO 1680 C=1,2
                   IF (C.EQ.1) THEN
                     WRITE(200,9305) CRAXCLADML(COLUMN,ROW,SECT)
                   ELSEIF (C.EQ.2) THEN
                     WRITE(200,9306)
                     WRITE(200,9307)
                     WRITE(200,9308)
                     WRITE(200,9309)
                     WRITE(200,9310)
                     WRITE(200,7006)
                     WRITE(200,7007)
                     WRITE(200,7008)
                     WRITE(200,9311)
                     WRITE(200,9312)
                     WRITE(200,7009)
                     WRITE(200,7010)
                     WRITE(200,7011)
                     WRITE(200,9313)
                     WRITE(200,7012)
                     WRITE(200,7013)
                     WRITE(200,7014)
                     WRITE(200,7015)
                   ENDIF
 1680            CONTINUE
               ELSEIF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT)
     c         .EQ.3) THEN
                 DO 1690 C=1,2
                   IF (C.EQ.1) THEN
                     WRITE(200,9314) CRAXCLADML(COLUMN,ROW,SECT)
                   ELSEIF (C.EQ.2) THEN
                     WRITE(200,9315)
                     WRITE(200,9316)
                     WRITE(200,9317)
                     WRITE(200,9318)
```

```
                                    WRITE(200,7016)
                                    WRITE(200,7017)
                                    WRITE(200,7018)
                                    WRITE(200,9319)
                                    WRITE(200,9320)
                                    WRITE(200,7019)
                                    WRITE(200,7020)
                                    WRITE(200,7021)
                                    WRITE(200,9321)
                                    WRITE(200,7022)
                                    WRITE(200,7023)
                                    WRITE(200,7024)
                                    WRITE(200,7025)
                                    WRITE(200,9322)
                                    WRITE(200,9323)
                                    WRITE(200,9324)
                                    WRITE(200,9325)
                                    WRITE(200,9326)
                                    WRITE(200,9327)
                                    WRITE(200,7026)
                                    WRITE(200,9328)
                                    WRITE(200,9329)
                                    WRITE(200,9330)
                                 ENDIF
1690                          CONTINUE
                           ENDIF
                        MN=MN+1
                     ENDIF
                     IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.1) THEN
                        CLADRHO=6.56
                     ELSEIF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.2) THEN
                        CLADRHO=7.90
                     ELSEIF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.3) THEN
                        CLADRHO=8.19
                     ENDIF
                     WRITE(30,1700) LN, CRAXCLADML(COLUMN,ROW,SECT),
     c               (-1*CLADRHO), CRAXCLADIRSURF(SECT),
     c               (-1*CRAXCLADORSURF(SECT)), (-1*CRAXCLADTOPSURF(SECT)),
     c               CRAXCLADBOTTOMSURF(SECT), CRAUNIV(COLUMN,ROW)
1700                 FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c               ' IMP:N=1 U=',I3,'   $ Control rod cladding')
                     LN=LN+1
                  ENDIF
*     Write the CR upper plenum cell in this CR universe.
*     Determine if the CR upper plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the upper plenum material specification label.
                  CRUPMLUNIQUE=.TRUE.
                  LEAVE=.FALSE.
                  IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                     DO 1720 RO=1,(ROW-1)
                        DO 1710 CO=1,50
                           IF (BANKNUM(CO,RO).NE.0) THEN
                              IF (BANKNUM(COLUMN,ROW).EQ.
```

```
     c                      BANKNUM(CO,RO)) THEN
                            CRUPMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                            EXIT
                         ENDIF
                      ENDIF
 1710              CONTINUE
                   IF (LEAVE.EQ..TRUE.) THEN
                      EXIT
                   ENDIF
 1720           CONTINUE
                IF (LEAVE.EQ..FALSE.) THEN
                   DO 1740 RO=ROW,ROW
                      DO 1730 CO=1,(COLUMN-1)
                         IF (BANKNUM(CO,RO).NE.0) THEN
                            IF (BANKNUM(COLUMN,ROW).EQ.
     c                      BANKNUM(CO,RO)) THEN
                               CRUPMLUNIQUE=.FALSE.
                               LEAVE=.TRUE.
                               CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                               EXIT
                            ENDIF
                         ENDIF
 1730                 CONTINUE
                      IF (LEAVE.EQ..TRUE.) THEN
                         EXIT
                      ENDIF
 1740              CONTINUE
                ENDIF
             ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                DO 1760 RO=1,(ROW-1)
                   DO 1750 CO=1,50
                      IF (BANKNUM(CO,RO).NE.0) THEN
                         IF (BANKNUM(COLUMN,ROW).EQ.
     c                   BANKNUM(CO,RO)) THEN
                            CRUPMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                            EXIT
                         ENDIF
                      ENDIF
 1750              CONTINUE
                   IF (LEAVE.EQ..TRUE.) THEN
                      EXIT
                   ENDIF
 1760           CONTINUE
             ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                DO 1780 RO=1,1
                   DO 1770 CO=1,(COLUMN-1)
                      IF (BANKNUM(CO,RO).NE.0) THEN
                         IF (BANKNUM(COLUMN,ROW).EQ.
     c                   BANKNUM(CO,RO)) THEN
                            CRUPMLUNIQUE=.FALSE.
```

```
                        LEAVE=.TRUE.
                        CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
1770           CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
1780        CONTINUE
         ENDIF
      IF (SURFVALUESPEC(CRAXABSTOPSURF(TOPSECT)).LT.
     c SURFVALUESPEC(UEFTOPSURF)) THEN
         IF (CRUPMLUNIQUE.EQ..TRUE.) THEN
            CRUPML(COLUMN,ROW)=MN
*  Check Control Rod Upper Plenum Regions
               DO 1810 C=1,CRUPLENMAT(BANKNUM(COLUMN,ROW),2)
                  IF (C.EQ.1) THEN
                     WRITE(200,1790) CRUPML(COLUMN,ROW),
     c               CRUPZS(BANKNUM(COLUMN,ROW),C),
     c               (-1*CRUPLENWTS(BANKNUM(COLUMN,ROW),C))
1790                 FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c               '     $ Control Rod Upper Plenum')
                  ELSE
                     WRITE(200,1800)
     c               CRUPZS(BANKNUM(COLUMN,ROW),C),
     c               (-1*CRUPLENWTS(BANKNUM(COLUMN,ROW),C))
1800                 FORMAT(T9,A9,3X,G14.6)
                  ENDIF
1810           CONTINUE
            MN=MN+1
         ENDIF
         WRITE(30,1820) LN, CRUPML(COLUMN,ROW),
     c   (-1*CRUPLENMAT(BANKNUM(COLUMN,ROW),1)),
     c   CRAXABSTOPSURF(TOPSECT),
     c   (-1*CRAXCLADTOPSURF(TOPSECT)),
     c   (-1*CRAXCLADIRSURF(TOPSECT)),
     c   CRAUNIV(COLUMN,ROW)
1820     FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c   ' IMP:N=1 U=',I3,'    $ Control rod upper plenum')
         LN=LN+1
      ENDIF
*     Write the CR lower plenum cell in this CR universe.
*     Determine if the CR lower plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the lower plenum material specification label.
         CRLPMLUNIQUE=.TRUE.
         LEAVE=.FALSE.
         IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
            DO 1840 RO=1,(ROW-1)
               DO 1830 CO=1,50
                  IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO)) THEN
                        CRLPMLUNIQUE=.FALSE.
```

```
                    LEAVE=.TRUE.
                    CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                    EXIT
                 ENDIF
              ENDIF
 1830      CONTINUE
           IF (LEAVE.EQ..TRUE.) THEN
              EXIT
           ENDIF
 1840   CONTINUE
        IF (LEAVE.EQ..FALSE.) THEN
           DO 1860 RO=ROW,ROW
              DO 1850 CO=1,(COLUMN-1)
                 IF (BANKNUM(CO,RO).NE.0) THEN
                    IF (BANKNUM(COLUMN,ROW).EQ.
     c              BANKNUM(CO,RO)) THEN
                       CRLPMLUNIQUE=.FALSE.
                       LEAVE=.TRUE.
                       CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                       EXIT
                    ENDIF
                 ENDIF
 1850         CONTINUE
              IF (LEAVE.EQ..TRUE.) THEN
                 EXIT
              ENDIF
 1860      CONTINUE
        ENDIF
     ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
        DO 1880 RO=1,(ROW-1)
           DO 1870 CO=1,50
              IF (BANKNUM(CO,RO).NE.0) THEN
                 IF (BANKNUM(COLUMN,ROW).EQ.
     c           BANKNUM(CO,RO)) THEN
                    CRLPMLUNIQUE=.FALSE.
                    LEAVE=.TRUE.
                    CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                    EXIT
                 ENDIF
              ENDIF
 1870      CONTINUE
           IF (LEAVE.EQ..TRUE.) THEN
              EXIT
           ENDIF
 1880   CONTINUE
     ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
        DO 1900 RO=1,1
           DO 1890 CO=1,(COLUMN-1)
              IF (BANKNUM(CO,RO).NE.0) THEN
                 IF (BANKNUM(COLUMN,ROW).EQ.
     c           BANKNUM(CO,RO)) THEN
                    CRLPMLUNIQUE=.FALSE.
                    LEAVE=.TRUE.
                    CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
```

```
                        EXIT
                     ENDIF
                  ENDIF
 1890          CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
 1900       CONTINUE
         ENDIF
      IF (SURFVALUESPEC(CRAXCLADBOTTOMSURF(BOTSECT)).LT.
   c  SURFVALUESPEC(UEFTOPSURF)) THEN
         IF (CRLPMLUNIQUE.EQ..TRUE.) THEN
            CRLPML(COLUMN,ROW)=MN
*  Check Control Rod Lower Plenum Regions
               DO 1930 C=1,CRLPLENMAT(BANKNUM(COLUMN,ROW),2)
                  IF (C.EQ.1) THEN
                     WRITE(200,1910) CRLPML(COLUMN,ROW),
   c                 CRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
   c                 (-1*CRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 1910                FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
   c                 '    $ Control Rod Lower Plenum')
                  ELSE
                     WRITE(200,1920)
   c                 CRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
   c                 (-1*CRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 1920                FORMAT(T9,A9,3X,G14.6)
                  ENDIF
 1930          CONTINUE
            MN=MN+1
         ENDIF
         WRITE(30,1940) LN, CRLPML(COLUMN,ROW),
   c     (-1*CRLPLENMAT(BANKNUM(COLUMN,ROW),1)),
   c     CRAXCLADBOTTOMSURF(BOTSECT),
   c     (-1*CRAXABSBOTTOMSURF(BOTSECT)),
   c     (-1*CRAXCLADIRSURF(BOTSECT)),
   c     CRAUNIV(COLUMN,ROW)
 1940    FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
   c     ' IMP:N=1 U=',I3,'    $ Control rod lower plenum')
         LN=LN+1
      ENDIF
 1950 CONTINUE
*  Write the moderator inside of the GT in the CR universe
      DO 2000 CRSECT=1,NUMCRAXS(BANKNUM(COLUMN,ROW))
         IF ((SURFVALUESPEC(GTTOPSURF).GE.
   c     SURFVALUESPEC(CRAXCLADTOPSURF(CRSECT))).AND.
   c     (SURFVALUESPEC(GTBOTSURF).LE.
   c     SURFVALUESPEC(CRAXCLADBOTTOMSURF(CRSECT)))) THEN
*     Write the moderator cells within the GT in this CR universe.
            WRITE(30,1960) LN, BMODML, (-1*MODDENSITY),
   c        (-1*GTIRSURF),
   c        CRAXCLADORSURF(CRSECT), (-1*CRAXCLADTOPSURF(CRSECT)),
   c        CRAXCLADBOTTOMSURF(CRSECT),
   c        CRAUNIV(COLUMN,ROW)
 1960       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
```

```
     c           ' IMP:N=1 U=',I3,
     c           '    $ Borated moderator inside guide tube')
                 LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTTOPSURF).GE.
     c        SURFVALUESPEC(CRAXCLADTOPSURF(CRSECT))).AND.
     c        (SURFVALUESPEC(GTBOTSURF).LT.
     c        SURFVALUESPEC(CRAXCLADTOPSURF(CRSECT))).AND.
     c        (SURFVALUESPEC(GTBOTSURF).GT.
     c        SURFVALUESPEC(CRAXCLADBOTTOMSURF(CRSECT)))) THEN
                 WRITE(30,1970) LN, BMODML, (-1*MODDENSITY),
     c           (-1*GTIRSURF),
     c           CRAXCLADORSURF(CRSECT), (-1*CRAXCLADTOPSURF(CRSECT)),
     c           GTBOTSURF,
     c           CRAUNIV(COLUMN,ROW)
 1970            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,
     c           '    $ Borated moderator inside guide tube')
                 LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c        SURFVALUESPEC(CRAXCLADTOPSURF(CRSECT))).AND.
     c        (SURFVALUESPEC(GTBOTSURF).LE.
     c        SURFVALUESPEC(CRAXCLADBOTTOMSURF(CRSECT))).AND.
     c        (SURFVALUESPEC(GTTOPSURF).GT.
     c        SURFVALUESPEC(CRAXCLADBOTTOMSURF(CRSECT)))) THEN
                 WRITE(30,1980) LN, BMODML, (-1*MODDENSITY),
     c           (-1*GTIRSURF),
     c           CRAXCLADORSURF(CRSECT), (-1*GTTOPSURF),
     c           CRAXCLADBOTTOMSURF(CRSECT),
     c           CRAUNIV(COLUMN,ROW)
 1980            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,
     c           '    $ Borated moderator inside guide tube')
                 LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c        SURFVALUESPEC(CRAXCLADTOPSURF(CRSECT))).AND.
     c        (SURFVALUESPEC(GTBOTSURF).GT.
     c        SURFVALUESPEC(CRAXCLADBOTTOMSURF(CRSECT)))) THEN
                 WRITE(30,1990) LN, BMODML, (-1*MODDENSITY),
     c           (-1*GTIRSURF),
     c           CRAXCLADORSURF(CRSECT), (-1*GTTOPSURF),
     c           GTBOTSURF,
     c           CRAUNIV(COLUMN,ROW)
 1990            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,
     c           '    $ Borated moderator inside guide tube')
                 LN=LN+1
              ENDIF
 2000 CONTINUE
              WRITE(30,2010) LN, BMODML, (-1*MODDENSITY),
     c        (-1*GTIRSURF),
     c        (-1*CRAXCLADBOTTOMSURF(NUMCRAXS(BANKNUM(COLUMN,ROW)))),
     c        GTBOTSURF,
     c        CRAUNIV(COLUMN,ROW)
 2010         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
```

```
     c      ' IMP:N=1 U=',I3,
     c      '   $ Borated moderator inside guide tube')
            LN=LN+1
*    Write the lower end-fitting cell specification for this CR universe.
            IF (SURFVALUESPEC(GTBOTSURF).GE.
     c      ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)) THEN
               WRITE(30,2020) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*CRLEFTOPSURF),
     c         CRAUNIV(COLUMN,ROW)
 2020          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,' IMP:N=1 U=',I3,
     c         '   $ Lower end-fitting')
               LN=LN+1
            ELSE
               WRITE(30,2030) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*CRLEFTOPSURF),
     c         GTORSURF,
     c         CRAUNIV(COLUMN,ROW)
 2030          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Lower end-fitting')
               LN=LN+1
               WRITE(30,2040) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)),
     c         (-1*GTBOTSURF),
     c         (-1*GTORSURF),
     c         CRAUNIV(COLUMN,ROW)
 2040          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Lower end-fitting')
               LN=LN+1
            ENDIF
*    Write the upper end-fitting cell specification for this CR universe.
            GTTOPSURF=GTTOPSURF
            GTSECTORSURF(1)=GTORSURF
            GTSECTIRSURF(1)=GTIRSURF
            CRCLADTOPSURF=CRAXCLADTOPSURF(1)
            CRCLADIRSURF=CRAXCLADIRSURF(1)
            CRCLADORSURF=CRAXCLADORSURF(1)
            IF ((SURFVALUESPEC(GTTOPSURF).GE.
     c      SURFVALUESPEC(UEFTOPSURF)).AND.
     c      (SURFVALUESPEC(CRCLADTOPSURF).GE.
     c      SURFVALUESPEC(UEFTOPSURF))) THEN
               WRITE(30,2050) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c         (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 2050          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
               LN=LN+1
            ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c      SURFVALUESPEC(UEFTOPSURF)).AND.
     c      (SURFVALUESPEC(GTTOPSURF).GT.
     c      SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(CRCLADTOPSURF).GE.
     c      SURFVALUESPEC(UEFTOPSURF))) THEN
               WRITE(30,2051) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
```

```
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 2051           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
                WRITE(30,2052) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), CRCLADORSURF, (-1*GTSECTORSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
 2052           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTTOPSURF).LE.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).GE.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,2053) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), CRCLADORSURF, CRAUNIV(COLUMN,ROW)
 2053           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTTOPSURF).GE.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,2054) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 2054           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
                WRITE(30,2055) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
 2055           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).GT.
     c       SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,2056) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 2056           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
```

```
              WRITE(30,2057) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c        (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c        CRAUNIV(COLUMN,ROW)
 2057         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
              LN=LN+1
              WRITE(30,2058) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c        (-1*UEFTOPSURF), GTSECTIRSURF(1),
     c        (-1*GTSECTORSURF(1)), CRAUNIV(COLUMN,ROW)
 2058         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
              LN=LN+1
           ELSEIF ((SURFVALUESPEC(GTTOPSURF).EQ.
     c     SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c     (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c     SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c     (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c     SURFVALUESPEC(UEFTOPSURF))) THEN
              WRITE(30,2059) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c        (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 2059         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
              LN=LN+1
              WRITE(30,2060) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c        (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c        CRAUNIV(COLUMN,ROW)
 2060         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
              LN=LN+1
           ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c     SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c     (SURFVALUESPEC(GTTOPSURF).GT.
     c     SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c     (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c     SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c     (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c     SURFVALUESPEC(UEFTOPSURF))) THEN
              WRITE(30,2061) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c        (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 2061         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
              LN=LN+1
              WRITE(30,2062) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c        (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c        CRAUNIV(COLUMN,ROW)
 2062         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
              LN=LN+1
```

```
            WRITE(30,2063) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c      (-1*CRCLADTOPSURF), CRCLADORSURF,
     c      (-1*GTSECTORSURF(1)), CRAUNIV(COLUMN,ROW)
 2063       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
            LN=LN+1
         ELSEIF ((SURFVALUESPEC(GTTOPSURF).LE.
     c   SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c   (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c   SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c   (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c   SURFVALUESPEC(UEFTOPSURF))) THEN
            WRITE(30,2064) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c      (-1*UEFTOPSURF), CRCLADORSURF, CRAUNIV(COLUMN,ROW)
 2064       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
            LN=LN+1
            WRITE(30,2065) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c      (-1*UEFTOPSURF), (-1*CRCLADORSURF),
     c      CRAUNIV(COLUMN,ROW)
 2065       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
            LN=LN+1
         ELSEIF ((SURFVALUESPEC(CRCLADTOPSURF).LE.
     c   SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c   (SURFVALUESPEC(GTTOPSURF).GE.
     c   SURFVALUESPEC(UEFTOPSURF))) THEN
            WRITE(30,2066) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c      (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 2066       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
            LN=LN+1
            WRITE(30,2067) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c      (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c      CRAUNIV(COLUMN,ROW)
 2067       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
            LN=LN+1
         ELSEIF ((SURFVALUESPEC(CRCLADTOPSURF).LE.
     c   SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c   (SURFVALUESPEC(GTTOPSURF).LT.
     c   SURFVALUESPEC(UEFTOPSURF)).AND.
     c   (SURFVALUESPEC(GTTOPSURF).GT.
     c   SURFVALUESPEC(UEFBOTTOMSURF))) THEN
            WRITE(30,2068) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c      (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 2068       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
```

```
            LN=LN+1
            WRITE(30,2069) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c      (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c      CRAUNIV(COLUMN,ROW)
 2069       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
            LN=LN+1
            WRITE(30,2070) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c      (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c      GTSECTIRSURF(1), CRAUNIV(COLUMN,ROW)
 2070       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
            LN=LN+1
         ELSEIF ((SURFVALUESPEC(CRCLADTOPSURF).LE.
     c   SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c   (SURFVALUESPEC(GTTOPSURF).LE.
     c   SURFVALUESPEC(UEFBOTTOMSURF))) THEN
            WRITE(30,2071) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c      (-1*UEFTOPSURF), CRAUNIV(COLUMN,ROW)
 2071       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
            LN=LN+1
         ENDIF
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
*
*
*
*
*
*
*
```

```
*    SPECIFICATIONS FOR HYBRID<>1 AND GTSPLIT=1
*
*
*
*
*
*
*
*
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
      ELSEIF ((HYBRID.NE.1).AND.(GTSPLIT.EQ.1)) THEN
*     Define the CR absorber radius.
               CURRENTSURF=CRADIM(BANKNUM(COLUMN,ROW),1)
               CURRENTSURFLABEL=0
               DO 2100 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
2100           CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  CRABSSURF=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  CRABSSURF=CURRENTSURFLABEL
               ENDIF
*     Define the CR absorber top surface.
               CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
    c          CRADIM(BANKNUM(COLUMN,ROW),4)+
    c          CRADIM(BANKNUM(COLUMN,ROW),5)
```

```
            IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
               CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
            ENDIF
            CURRENTSURFLABEL=0
            DO 2110 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
2110        CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               CRABSTOPSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               CRABSTOPSURF=CURRENTSURFLABEL
            ENDIF
*     Define the CR absorber bottom surface.
            CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c      CRADIM(BANKNUM(COLUMN,ROW),4)
            IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
               CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
            ENDIF
            CURRENTSURFLABEL=0
            DO 2120 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
2120        CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               CRABSBOTTOMSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               CRABSBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
*     Define the CR cladding inner radius.
            CURRENTSURF=CRADIM(BANKNUM(COLUMN,ROW),2)
            CURRENTSURFLABEL=0
            DO 2130 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'CZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
2130        CONTINUE
```

```
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 CRCLADIRSURF=SN
                 SURFTYPESPEC(SN)='CZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 CRCLADIRSURF=CURRENTSURFLABEL
              ENDIF
*     Define the CR cladding outer radius.
              CURRENTSURF=CRADIM(BANKNUM(COLUMN,ROW),3)
              CURRENTSURFLABEL=0
              DO 2140 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 2140         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 CRCLADORSURF=SN
                 SURFTYPESPEC(SN)='CZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 CRCLADORSURF=CURRENTSURFLABEL
              ENDIF
*     Define the CR cladding top surface.
              CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c        CRADIM(BANKNUM(COLUMN,ROW),4)+
     c        CRADIM(BANKNUM(COLUMN,ROW),5)+
     c        CRADIM(BANKNUM(COLUMN,ROW),7)
              IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                 CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
              ENDIF
              CURRENTSURFLABEL=0
              DO 2150 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 2150         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 CRCLADTOPSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 CRCLADTOPSURF=CURRENTSURFLABEL
              ENDIF
*     Define the CR cladding bottom surface.
              CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
```

```
     c          CRADIM(BANKNUM(COLUMN,ROW),4)-
     c          CRADIM(BANKNUM(COLUMN,ROW),6)
                IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                   CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                ENDIF
                CURRENTSURFLABEL=0
                DO 2160 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 2160           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   CRCLADBOTTOMSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   CRCLADBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
           IF (SURFVALUESPEC(CRABSBOTTOMSURF).LT.
     c     SURFVALUESPEC(UEFTOPSURF)) THEN
*  Check Control Rod Absorber Material
                CRABSML=MN
                DO 2190 C=1,CRABSMAT(BANKNUM(COLUMN,ROW),2)
                   IF (C.EQ.1) THEN
                      WRITE(200,2170) CRABSML,
     c                CRABSZAIDS(BANKNUM(COLUMN,ROW),C),
     c                (-1*CRABSWTS(BANKNUM(COLUMN,ROW),C)),
     c                ASSYID(COLUMN,ROW)
 2170                 FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                '   $ Control Rod Absorber Material in Assembly ',
     c                A5)
                   ELSE
                      WRITE(200,2180) CRABSZAIDS(BANKNUM(COLUMN,ROW),C),
     c                (-1*CRABSWTS(BANKNUM(COLUMN,ROW),C))
 2180                 FORMAT(T9,A9,3X,G14.6)
                   ENDIF
 2190           CONTINUE
                MN=MN+1
*     Write the CR absorber cell in this CR universe.
           WRITE(30,2200) LN, CRABSML,
     c     (-1*CRABSMAT(BANKNUM(COLUMN,ROW),1)), (-1*CRABSSURF),
     c     (-1*CRABSTOPSURF), CRABSBOTTOMSURF,
     c     CRAUNIV(COLUMN,ROW)
 2200      FORMAT(T1,I4,T6,I4,T11,F10.6,T25,I4,1X,I4,1X,I4,
     c     ' IMP:N=1 U=',I3,'   $ Control rod absorber material')
           LN=LN+1
*     Write the absorber-to-cladding gap cell in this CR universe.
           WRITE(30,2210) LN, (-1*CRCLADIRSURF), CRABSSURF,
     c     (-1*CRABSTOPSURF),
     c     CRABSBOTTOMSURF, CRAUNIV(COLUMN,ROW)
```

```
 2210       FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Absorber-to-cladding gap')
            LN=LN+1
         ENDIF
*     Write the CR cladding cell in this CR universe.
*     Determine if the CR cladding material specification has
*     previously been defined.  If it has been previously defined, determine
*     the cladding material specification label.
            CLADMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 2230 RO=1,(ROW-1)
                  DO 2220 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA  ') THEN
                        IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  CRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRCLADML(COLUMN,ROW)=CRCLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
 2220             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 2230          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 2250 RO=ROW,ROW
                     DO 2240 CO=1,(COLUMN-1)
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA  ') THEN
                        IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  CRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRCLADML(COLUMN,ROW)=CRCLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
 2240                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 2250             CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 2270 RO=1,(ROW-1)
                  DO 2260 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA  ') THEN
```

```
                        IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  CRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRCLADML(COLUMN,ROW)=CRCLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
 2260             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 2270          CONTINUE
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
               DO 2290 RO=1,1
                  DO 2280 CO=1,(COLUMN-1)
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA  ') THEN
                        IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  CRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRCLADML(COLUMN,ROW)=CRCLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
 2280             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 2290          CONTINUE
            ENDIF
         IF (SURFVALUESPEC(CRCLADBOTTOMSURF).LT.
     c   SURFVALUESPEC(UEFTOPSURF)) THEN
            IF (CLADMLUNIQUE.EQ..TRUE.) THEN
               CRCLADML(COLUMN,ROW)=MN
*  Check Control Rod Cladding
                  IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
                     DO 2300 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9300) CRCLADML(COLUMN,ROW)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9301)
                           WRITE(200,7000)
                           WRITE(200,7001)
                           WRITE(200,7002)
                           WRITE(200,9302)
                           WRITE(200,7003)
                           WRITE(200,7004)
                           WRITE(200,7005)
                           WRITE(200,9303)
                           WRITE(200,9304)
```

```
                ENDIF
 2300         CONTINUE
            ELSEIF (CRCLADMAT(BANKNUM(COLUMN,ROW))
     c      .EQ.2) THEN
              DO 2310 C=1,2
                IF (C.EQ.1) THEN
                  WRITE(200,9305) CRCLADML(COLUMN,ROW)
                ELSEIF (C.EQ.2) THEN
                  WRITE(200,9306)
                  WRITE(200,9307)
                  WRITE(200,9308)
                  WRITE(200,9309)
                  WRITE(200,9310)
                  WRITE(200,7006)
                  WRITE(200,7007)
                  WRITE(200,7008)
                  WRITE(200,9311)
                  WRITE(200,9312)
                  WRITE(200,7009)
                  WRITE(200,7010)
                  WRITE(200,7011)
                  WRITE(200,9313)
                  WRITE(200,7012)
                  WRITE(200,7013)
                  WRITE(200,7014)
                  WRITE(200,7015)
                ENDIF
 2310         CONTINUE
            ELSEIF (CRCLADMAT(BANKNUM(COLUMN,ROW))
     c      .EQ.3) THEN
              DO 2320 C=1,2
                IF (C.EQ.1) THEN
                  WRITE(200,9314) CRCLADML(COLUMN,ROW)
                ELSEIF (C.EQ.2) THEN
                  WRITE(200,9315)
                  WRITE(200,9316)
                  WRITE(200,9317)
                  WRITE(200,9318)
                  WRITE(200,7016)
                  WRITE(200,7017)
                  WRITE(200,7018)
                  WRITE(200,9319)
                  WRITE(200,9320)
                  WRITE(200,7019)
                  WRITE(200,7020)
                  WRITE(200,7021)
                  WRITE(200,9321)
                  WRITE(200,7022)
                  WRITE(200,7023)
                  WRITE(200,7024)
                  WRITE(200,7025)
                  WRITE(200,9322)
                  WRITE(200,9323)
                  WRITE(200,9324)
```

```
                        WRITE(200,9325)
                        WRITE(200,9326)
                        WRITE(200,9327)
                        WRITE(200,7026)
                        WRITE(200,9328)
                        WRITE(200,9329)
                        WRITE(200,9330)
                     ENDIF
 2320             CONTINUE
               ENDIF
               MN=MN+1
            ENDIF
            IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
               CLADRHO=6.56
            ELSEIF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.2) THEN
               CLADRHO=7.90
            ELSEIF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.3) THEN
               CLADRHO=8.19
            ENDIF
            WRITE(30,2330) LN, CRCLADML(COLUMN,ROW), (-1*CLADRHO),
     c      CRCLADIRSURF,
     c      (-1*CRCLADORSURF), (-1*CRCLADTOPSURF), CRCLADBOTTOMSURF,
     c      CRAUNIV(COLUMN,ROW)
 2330       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'    $ Control rod cladding')
            LN=LN+1
         ENDIF
*     Write the CR upper plenum cell in this CR universe.
*     Determine if the CR upper plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the upper plenum material specification label.
            CRUPMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 2350 RO=1,(ROW-1)
                  DO 2340 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                           CRUPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 2340             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 2350          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 2370 RO=ROW,ROW
                     DO 2360 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
```

```
                          IF (BANKNUM(COLUMN,ROW).EQ.
     c                    BANKNUM(CO,RO)) THEN
                             CRUPMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                             EXIT
                          ENDIF
                       ENDIF
 2360               CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
 2370            CONTINUE
              ENDIF
           ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
              DO 2390 RO=1,(ROW-1)
                 DO 2380 CO=1,50
                    IF (BANKNUM(CO,RO).NE.0) THEN
                       IF (BANKNUM(COLUMN,ROW).EQ.
     c                    BANKNUM(CO,RO)) THEN
                          CRUPMLUNIQUE=.FALSE.
                          LEAVE=.TRUE.
                          CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                          EXIT
                       ENDIF
                    ENDIF
 2380            CONTINUE
                 IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                 ENDIF
 2390         CONTINUE
           ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
              DO 2410 RO=1,1
                 DO 2400 CO=1,(COLUMN-1)
                    IF (BANKNUM(CO,RO).NE.0) THEN
                       IF (BANKNUM(COLUMN,ROW).EQ.
     c                    BANKNUM(CO,RO)) THEN
                          CRUPMLUNIQUE=.FALSE.
                          LEAVE=.TRUE.
                          CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                          EXIT
                       ENDIF
                    ENDIF
 2400            CONTINUE
                 IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                 ENDIF
 2410         CONTINUE
           ENDIF
        IF (SURFVALUESPEC(CRABSTOPSURF).LT.
     c  SURFVALUESPEC(UEFTOPSURF)) THEN
           IF (CRUPMLUNIQUE.EQ..TRUE.) THEN
              CRUPML(COLUMN,ROW)=MN
*  Check Control Rod Upper Plenum Regions
```

```
                  DO 2440 C=1,CRUPLENMAT(BANKNUM(COLUMN,ROW),2)
                     IF (C.EQ.1) THEN
                        WRITE(200,2420) CRUPML(COLUMN,ROW),
     c                  CRUPZS(BANKNUM(COLUMN,ROW),C),
     c                  (-1*CRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 2420                   FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                  '    $ Control Rod Upper Plenum')
                     ELSE
                        WRITE(200,2430)
     c                  CRUPZS(BANKNUM(COLUMN,ROW),C),
     c                  (-1*CRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 2430                   FORMAT(T9,A9,3X,G14.6)
                     ENDIF
 2440             CONTINUE
                MN=MN+1
             ENDIF
             WRITE(30,2450) LN, CRUPML(COLUMN,ROW),
     c       (-1*CRUPLENMAT(BANKNUM(COLUMN,ROW),1)),
     c       CRABSTOPSURF,
     c       (-1*CRCLADTOPSURF), (-1*CRCLADIRSURF),
     c       CRAUNIV(COLUMN,ROW)
 2450        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'   $ Control rod upper plenum')
             LN=LN+1
          ENDIF
*     Write the CR lower plenum cell in this CR universe.
*     Determine if the CR lower plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the lower plenum material specification label.
             CRLPMLUNIQUE=.TRUE.
             LEAVE=.FALSE.
             IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                DO 2470 RO=1,(ROW-1)
                   DO 2460 CO=1,50
                      IF (BANKNUM(CO,RO).NE.0) THEN
                         IF (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO)) THEN
                            CRLPMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                            EXIT
                         ENDIF
                      ENDIF
 2460              CONTINUE
                   IF (LEAVE.EQ..TRUE.) THEN
                      EXIT
                   ENDIF
 2470           CONTINUE
                IF (LEAVE.EQ..FALSE.) THEN
                   DO 2490 RO=ROW,ROW
                      DO 2480 CO=1,(COLUMN-1)
                         IF (BANKNUM(CO,RO).NE.0) THEN
                            IF (BANKNUM(COLUMN,ROW).EQ.
     c                      BANKNUM(CO,RO)) THEN
                               CRLPMLUNIQUE=.FALSE.
```

```
                           LEAVE=.TRUE.
                           CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 2480             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 2490          CONTINUE
            ENDIF
         ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
            DO 2510 RO=1,(ROW-1)
               DO 2500 CO=1,50
                  IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKNUM(COLUMN,ROW).EQ.
     c                  BANKNUM(CO,RO)) THEN
                        CRLPMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
 2500          CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
 2510       CONTINUE
         ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
            DO 2530 RO=1,1
               DO 2520 CO=1,(COLUMN-1)
                  IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKNUM(COLUMN,ROW).EQ.
     c                  BANKNUM(CO,RO)) THEN
                        CRLPMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
 2520          CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
 2530       CONTINUE
         ENDIF
      IF (SURFVALUESPEC(CRCLADBOTTOMSURF).LT.
     c SURFVALUESPEC(UEFTOPSURF)) THEN
         IF (CRLPMLUNIQUE.EQ..TRUE.) THEN
            CRLPML(COLUMN,ROW)=MN
*  Check Control Rod Lower Plenum Regions
               DO 2560 C=1,CRLPLENMAT(BANKNUM(COLUMN,ROW),2)
                  IF (C.EQ.1) THEN
                     WRITE(200,2540) CRLPML(COLUMN,ROW),
```

```
     c                   CRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                   (-1*CRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 2540                    FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                   '    $ Control Rod Lower Plenum')
                       ELSE
                         WRITE(200,2550)
     c                   CRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                   (-1*CRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 2550                    FORMAT(T9,A9,3X,G14.6)
                       ENDIF
 2560                CONTINUE
                   MN=MN+1
                 ENDIF
                 WRITE(30,2570) LN, CRLPML(COLUMN,ROW),
     c           (-1*CRLPLENMAT(BANKNUM(COLUMN,ROW),1)), CRCLADBOTTOMSURF,
     c           (-1*CRABSBOTTOMSURF), (-1*CRCLADIRSURF),
     c           CRAUNIV(COLUMN,ROW)
 2570            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'   $ Control rod lower plenum')
                 LN=LN+1
               ENDIF
             DO 2620 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
*            Define the GT section top surface.
                   CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),3,SECT)
                   IF (CURRENTSURF.GT.SURFVALUESPEC(UEFTOPSURF)) THEN
                     CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                   ENDIF
                   CURRENTSURFLABEL=0
                   DO 2580 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                       ENDIF
                     ENDIF
 2580              CONTINUE
                   IF (CURRENTSURFLABEL.EQ.0) THEN
                     GTSECTTOPSURF(SECT)=SN
                     SURFTYPESPEC(SN)='PZ'
                     SURFVALUESPEC(SN)=CURRENTSURF
                     SN=SN+1
                   ELSE
                     GTSECTTOPSURF(SECT)=CURRENTSURFLABEL
                   ENDIF
*            Define the GT section bottom surface.
                   CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),4,SECT)
                   CURRENTSURFLABEL=0
                   DO 2590 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                       ENDIF
                     ENDIF
```

```
 2590          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  GTSECTBOTSURF(SECT)=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTSECTBOTSURF(SECT)=CURRENTSURFLABEL
               ENDIF
*     Define the GT section outer radius surface.
               CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),2,SECT)
               CURRENTSURFLABEL=0
               DO 2600 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 2600          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  GTSECTORSURF(SECT)=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTSECTORSURF(SECT)=CURRENTSURFLABEL
               ENDIF
*     Define the GT section inner radius surface.
               CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),1,SECT)
               CURRENTSURFLABEL=0
               DO 2610 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 2610          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  GTSECTIRSURF(SECT)=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTSECTIRSURF(SECT)=CURRENTSURFLABEL
               ENDIF
 2620 CONTINUE
*     Write the GT material cell
          DO 2750 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
*     Determine if the GT material specification has
*     previously been defined.  If it has been previously defined, determine
*     the material specification label.
            CLADMLUNIQUE=.TRUE.
```

```
              LEAVE=.FALSE.
              IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                 DO 2640 RO=1,(ROW-1)
                    DO 2630 CO=1,50
                       IF ((DESNUM(CO,RO).NE.0).AND.
     c                 (BANKNUM(CO,RO).EQ.0)) THEN
                          IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                    GTMAT(DESNUM(CO,RO))) THEN
                             CLADMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                             EXIT
                          ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                    GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                             CLADMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                             EXIT
                          ENDIF
                       ENDIF
 2630               CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
 2640            CONTINUE
                 IF (LEAVE.EQ..FALSE.) THEN
                    DO 2660 RO=ROW,ROW
                       DO 2650 CO=1,(COLUMN-1)
                          IF ((DESNUM(CO,RO).NE.0).AND.
     c                    (BANKNUM(CO,RO).EQ.0)) THEN
                             IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                       GTMAT(DESNUM(CO,RO))) THEN
                                CLADMLUNIQUE=.FALSE.
                                LEAVE=.TRUE.
                                GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                                EXIT
                             ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                       GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                                CLADMLUNIQUE=.FALSE.
                                LEAVE=.TRUE.
                                GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                                EXIT
                             ENDIF
                          ENDIF
 2650                  CONTINUE
                       IF (LEAVE.EQ..TRUE.) THEN
                          EXIT
                       ENDIF
 2660               CONTINUE
                 ENDIF
              ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                 DO 2680 RO=1,(ROW-1)
                    DO 2670 CO=1,50
                       IF ((DESNUM(CO,RO).NE.0).AND.
```

```
     c                   (BANKNUM(CO,RO).EQ.0)) THEN
                         IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                   GTMAT(DESNUM(CO,RO))) THEN
                            CLADMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                            EXIT
                         ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                   GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                            CLADMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                            EXIT
                         ENDIF
                      ENDIF
 2670              CONTINUE
                   IF (LEAVE.EQ..TRUE.) THEN
                      EXIT
                   ENDIF
 2680           CONTINUE
             ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                DO 2700 RO=1,1
                   DO 2690 CO=1,(COLUMN-1)
                      IF ((DESNUM(CO,RO).NE.0).AND.
     c                   (BANKNUM(CO,RO).EQ.0)) THEN
                         IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                   GTMAT(DESNUM(CO,RO))) THEN
                            CLADMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                            EXIT
                         ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                   GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                            CLADMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                            EXIT
                         ENDIF
                      ENDIF.
 2690              CONTINUE
                   IF (LEAVE.EQ..TRUE.) THEN
                      EXIT
                   ENDIF
 2700           CONTINUE
             ENDIF
             IF (CLADMLUNIQUE.EQ..TRUE.)  THEN
                GTAXML(COLUMN,ROW,SECT)=MN
*  Check Guide Tube Material
                   IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.1) THEN
                      DO 2710 C=1,2
                         IF (C.EQ.1) THEN
                            WRITE(200,9300) GTAXML(COLUMN,ROW,SECT)
                         ELSEIF (C.EQ.2) THEN
                            WRITE(200,9301)
```

```
                      WRITE(200,7000)
                      WRITE(200,7001)
                      WRITE(200,7002)
                      WRITE(200,9302)
                      WRITE(200,7003)
                      WRITE(200,7004)
                      WRITE(200,7005)
                      WRITE(200,9303)
                      WRITE(200,9304)
                    ENDIF
 2710             CONTINUE
                ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT)
     c          .EQ.2) THEN
                  DO 2720 C=1,2
                    IF (C.EQ.1) THEN
                      WRITE(200,9305) GTAXML(COLUMN,ROW,SECT)
                    ELSEIF (C.EQ.2) THEN
                      WRITE(200,9306)
                      WRITE(200,9307)
                      WRITE(200,9308)
                      WRITE(200,9309)
                      WRITE(200,9310)
                      WRITE(200,7006)
                      WRITE(200,7007)
                      WRITE(200,7008)
                      WRITE(200,9311)
                      WRITE(200,9312)
                      WRITE(200,7009)
                      WRITE(200,7010)
                      WRITE(200,7011)
                      WRITE(200,9313)
                      WRITE(200,7012)
                      WRITE(200,7013)
                      WRITE(200,7014)
                      WRITE(200,7015)
                    ENDIF
 2720             CONTINUE
                ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT)
     c          .EQ.3) THEN
                  DO 2730 C=1,2
                    IF (C.EQ.1) THEN
                      WRITE(200,9314) GTAXML(COLUMN,ROW,SECT)
                    ELSEIF (C.EQ.2) THEN
                      WRITE(200,9315)
                      WRITE(200,9316)
                      WRITE(200,9317)
                      WRITE(200,9318)
                      WRITE(200,7016)
                      WRITE(200,7017)
                      WRITE(200,7018)
                      WRITE(200,9319)
                      WRITE(200,9320)
                      WRITE(200,7019)
                      WRITE(200,7020)
```

```
                        WRITE(200,7021)
                        WRITE(200,9321)
                        WRITE(200,7022)
                        WRITE(200,7023)
                        WRITE(200,7024)
                        WRITE(200,7025)
                        WRITE(200,9322)
                        WRITE(200,9323)
                        WRITE(200,9324)
                        WRITE(200,9325)
                        WRITE(200,9326)
                        WRITE(200,9327)
                        WRITE(200,7026)
                        WRITE(200,9328)
                        WRITE(200,9329)
                        WRITE(200,9330)
                      ENDIF
 2730               CONTINUE
                  ENDIF
                MN=MN+1
              ENDIF
              IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.1) THEN
                CLADRHO=6.56
              ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.2) THEN
                CLADRHO=7.90
              ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.3) THEN
                CLADRHO=8.19
              ENDIF
              WRITE(30,2740) LN, GTAXML(COLUMN,ROW,SECT), (-1*CLADRHO),
     c        GTSECTIRSURF(SECT),
     c        (-1*GTSECTORSURF(SECT)), (-1*GTSECTTOPSURF(SECT)),
     c        GTSECTBOTSURF(SECT), CRAUNIV(COLUMN,ROW)
 2740         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'   $ Guide tube')
              LN=LN+1
 2750       CONTINUE
*     Loop through the spacer and moderator regions along the axial
*     length of the GT (from top to bottom).
            SPACHEIGHT=0.0
            DO 2760 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
              SPACHEIGHT=SPACHEIGHT+SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
 2760       CONTINUE
            DO 2950 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
*     Define the homogenized spacer region bounding surfaces.
              IF (SPN.EQ.1) THEN
                SPACERTOPSURF=UEFBOTTOMSURF
                CURRENTSURF=SURFVALUESPEC(UEFBOTTOMSURF)-
     c          SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                CURRENTSURFLABEL=0
                DO 2770 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
```

```
                  ENDIF
               ENDIF
 2770       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               SPACERBOTTOMSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               SPACERBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
            WATERREGIONTOPSURF=SPACERBOTTOMSURF
            CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
            CURRENTSURFLABEL=0
            DO 2780 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 2780       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               WATERREGIONBOTTOMSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
         ELSEIF ((SPN.NE.1).AND.(SPN.NE.
     c   NUMOFSPACERS(DESNUM(COLUMN,ROW)))) THEN
            SPACERTOPSURF=WATERREGIONBOTTOMSURF
            CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c      SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
            CURRENTSURFLABEL=0
            DO 2790 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 2790       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               SPACERBOTTOMSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               SPACERBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
            WATERREGIONTOPSURF=SPACERBOTTOMSURF
            CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
```

```
               CURRENTSURFLABEL=0
               DO 2800 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 2800          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  WATERREGIONBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
            ELSEIF (SPN.EQ.NUMOFSPACERS(DESNUM(COLUMN,ROW))) THEN
               SPACERTOPSURF=WATERREGIONBOTTOMSURF
               CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c         SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
               CURRENTSURFLABEL=0
               DO 2810 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 2810          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  SPACERBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  SPACERBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
               WATERREGIONTOPSURF=SPACERBOTTOMSURF
               WATERREGIONBOTTOMSURF=NODEBOTTOMSURF
            ENDIF
*     Write the current homogenized spacer region cell in this GT universe.
          DO 2940 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
             IF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c      SURFVALUESPEC(SPACERTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c      SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
            WRITE(30,2820) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c      (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c      GTSECTORSURF(SECT),
     c      (-1*SPACERTOPSURF), SPACERBOTTOMSURF, CRAUNIV(COLUMN,ROW),
     c      SPN
 2820       FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,
```

```
     c        '   $ Homogenized region for spacer ',I2)
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c        SURFVALUESPEC(SPACERTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
              WRITE(30,2830) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c        (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c        GTSECTORSURF(SECT),
     c        (-1*SPACERTOPSURF), SPACERBOTTOMSURF, CRAUNIV(COLUMN,ROW),
     c        SPN
 2830         FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,
     c        '   $ Homogenized region for spacer ',I2)
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c        SURFVALUESPEC(SPACERTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c        SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
              WRITE(30,2840) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c        (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c        GTSECTORSURF(SECT),
     c        (-1*SPACERTOPSURF), SPACERBOTTOMSURF, CRAUNIV(COLUMN,ROW),
     c        SPN
 2840         FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,
     c        '   $ Homogenized region for spacer ',I2)
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(SPACERTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c        SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
              WRITE(30,2850) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c        (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c        GTSECTORSURF(SECT),
     c        (-1*SPACERTOPSURF), SPACERBOTTOMSURF, CRAUNIV(COLUMN,ROW),
     c        SPN
 2850         FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,
     c        '   $ Homogenized region for spacer ',I2)
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(SPACERTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).GT.
     c        SURFVALUESPEC(SPACERBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(SPACERTOPSURF))) THEN
              WRITE(30,2860) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c        (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c        GTSECTORSURF(SECT),
     c        (-1*SPACERTOPSURF), GTSECTBOTSURF(SECT),
     c        CRAUNIV(COLUMN,ROW), SPN
 2860         FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,
```

```
     c        '   $ Homogenized region for spacer ',I2)
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).LT.
     c        SURFVALUESPEC(SPACERTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(SPACERBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
              WRITE(30,2870) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c        (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c        GTSECTORSURF(SECT),
     c        (-1*GTSECTTOPSURF(SECT)), SPACERBOTTOMSURF,
     c        CRAUNIV(COLUMN,ROW), SPN
 2870         FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,
     c        '   $ Homogenized region for spacer ',I2)
              LN=LN+1
               ENDIF
*    Write the water region cell below the current homogenized spacer cell
in this GT universe.
               IF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,2880) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        CRAUNIV(COLUMN,ROW)
 2880         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'   $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,2890) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        CRAUNIV(COLUMN,ROW)
 2890         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'   $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,2900) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        CRAUNIV(COLUMN,ROW)
 2900         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'   $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
```

```
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,2910) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        CRAUNIV(COLUMN,ROW)
 2910         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).GT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(WATERREGIONTOPSURF))) THEN
              WRITE(30,2920) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), GTSECTBOTSURF(SECT),
     c        CRAUNIV(COLUMN,ROW)
 2920         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).LT.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,2930) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*GTSECTTOPSURF(SECT)), WATERREGIONBOTTOMSURF,
     c        CRAUNIV(COLUMN,ROW)
 2930         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
               ENDIF
 2940         CONTINUE
 2950     CONTINUE
*  Write the moderator inside of the GT in the CR universe
          DO 3000 GTSECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
              IF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c        SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LE.
     c        SURFVALUESPEC(CRCLADBOTTOMSURF))) THEN
*     Write the moderator cells within the GT in this CR universe.
                 WRITE(30,2960) LN, BMODML, (-1*MODDENSITY),
     c           (-1*GTSECTIRSURF(GTSECT)),
     c           CRCLADORSURF, (-1*CRCLADTOPSURF),
     c           CRCLADBOTTOMSURF,
     c           CRAUNIV(COLUMN,ROW)
 2960            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,
     c           '    $ Borated moderator inside guide tube')
```

```
               LN=LN+1
            ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c      SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LT.
     c      SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).GT.
     c      SURFVALUESPEC(CRCLADBOTTOMSURF))) THEN
               WRITE(30,2970) LN, BMODML, (-1*MODDENSITY),
     c         (-1*GTSECTIRSURF(GTSECT)),
     c         CRCLADORSURF, (-1*CRCLADTOPSURF),
     c         GTSECTBOTSURF(GTSECT),
     c         CRAUNIV(COLUMN,ROW)
 2970          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,
     c         '    $ Borated moderator inside guide tube')
               LN=LN+1
            ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).LT.
     c      SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LE.
     c      SURFVALUESPEC(CRCLADBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GT.
     c      SURFVALUESPEC(CRCLADBOTTOMSURF))) THEN
               WRITE(30,2980) LN, BMODML, (-1*MODDENSITY),
     c         (-1*GTSECTIRSURF(GTSECT)),
     c         CRCLADORSURF, (-1*GTSECTTOPSURF(GTSECT)),
     c         CRCLADBOTTOMSURF,
     c         CRAUNIV(COLUMN,ROW)
 2980          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,
     c         '    $ Borated moderator inside guide tube')
               LN=LN+1
            ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).LT.
     c      SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).GT.
     c      SURFVALUESPEC(CRCLADBOTTOMSURF))) THEN
               WRITE(30,2990) LN, BMODML, (-1*MODDENSITY),
     c         (-1*GTSECTIRSURF(GTSECT)),
     c         CRCLADORSURF, (-1*GTSECTTOPSURF(GTSECT)),
     c         GTSECTBOTSURF(GTSECT),
     c         CRAUNIV(COLUMN,ROW)
 2990          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,
     c         '    $ Borated moderator inside guide tube')
               LN=LN+1
            ENDIF
 3000    CONTINUE
*    Determine the axial GT section which contains the lowest CR axial section
        DO 3010 GTSECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
           IF ((SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LT.
     c     SURFVALUESPEC(CRCLADBOTTOMSURF)).AND.
     c     (SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c     SURFVALUESPEC(CRCLADBOTTOMSURF))) THEN
              BGT=GTSECT
              EXIT
```

```
            ENDIF
 3010  CONTINUE
       DO 3040 GTSECT=BGT,NUMOFGTAXS(DESNUM(COLUMN,ROW))
          IF (GTSECT.EQ.BGT) THEN
             WRITE(30,3020) LN, BMODML, (-1*MODDENSITY),
     c       (-1*GTSECTIRSURF(GTSECT)),
     c       (-1*CRCLADBOTTOMSURF),
     c       GTSECTBOTSURF(GTSECT),
     c       CRAUNIV(COLUMN,ROW)
 3020        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,
     c       '    $ Borated moderator inside guide tube')
             LN=LN+1
          ELSE
             WRITE(30,3030) LN, BMODML, (-1*MODDENSITY),
     c       (-1*GTSECTIRSURF(GTSECT)),
     c       (-1*GTSECTTOPSURF(GTSECT)),
     c       GTSECTBOTSURF(GTSECT),
     c       CRAUNIV(COLUMN,ROW)
 3030        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,
     c       '    $ Borated moderator inside guide tube')
             LN=LN+1
          ENDIF
 3040  CONTINUE
*      Write the lower end-fitting cell specification for this CR universe.
             GTBOTSURF=GTSECTBOTSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))
             IF (SURFVALUESPEC(GTBOTSURF).GE.
     c       ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)) THEN
                WRITE(30,3050) LN, FRLEFML(COLUMN,ROW),
     c          (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*CRLEFTOPSURF),
     c          CRAUNIV(COLUMN,ROW)
 3050           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,' IMP:N=1 U=',I3,
     c          '    $ Lower end-fitting')
                LN=LN+1
             ELSE
                WRITE(30,3060) LN, FRLEFML(COLUMN,ROW),
     c          (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*CRLEFTOPSURF),
     c          GTSECTORSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW))),
     c          CRAUNIV(COLUMN,ROW)
 3060           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Lower end-fitting')
                LN=LN+1
                WRITE(30,3070) LN, FRLEFML(COLUMN,ROW),
     c          (-1*LEFMAT(DESNUM(COLUMN,ROW),1)),
     c          (-1*GTSECTBOTSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))),
     c          (-1*GTSECTORSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))),
     c          CRAUNIV(COLUMN,ROW)
 3070           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Lower end-fitting')
                LN=LN+1
             ENDIF
*      Write the upper end-fitting cell specification for this CR universe.
             GTTOPSURF=GTSECTTOPSURF(1)
```

```
          IF ((SURFVALUESPEC(GTTOPSURF).GE.
     c    SURFVALUESPEC(UEFTOPSURF)).AND.
     c    (SURFVALUESPEC(CRCLADTOPSURF).GE.
     c    SURFVALUESPEC(UEFTOPSURF))) THEN
             WRITE(30,3080) LN, FRUEFML(COLUMN,ROW),
     c       (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c       (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
3080         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
             LN=LN+1
          ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c    SURFVALUESPEC(UEFTOPSURF)).AND.
     c    (SURFVALUESPEC(GTTOPSURF).GT.
     c    SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c    (SURFVALUESPEC(CRCLADTOPSURF).GE.
     c    SURFVALUESPEC(UEFTOPSURF))) THEN
             WRITE(30,3090) LN, FRUEFML(COLUMN,ROW),
     c       (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c       (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
3090         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
             LN=LN+1
             WRITE(30,3100) LN, FRUEFML(COLUMN,ROW),
     c       (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c       (-1*UEFTOPSURF), CRCLADORSURF, (-1*GTSECTORSURF(1)),
     c       CRAUNIV(COLUMN,ROW)
3100         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
             LN=LN+1
          ELSEIF ((SURFVALUESPEC(GTTOPSURF).LE.
     c    SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c    (SURFVALUESPEC(CRCLADTOPSURF).GE.
     c    SURFVALUESPEC(UEFTOPSURF))) THEN
             WRITE(30,3110) LN, FRUEFML(COLUMN,ROW),
     c       (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c       (-1*UEFTOPSURF), CRCLADORSURF, CRAUNIV(COLUMN,ROW)
3110         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
             LN=LN+1
          ELSEIF ((SURFVALUESPEC(GTTOPSURF).GE.
     c    SURFVALUESPEC(UEFTOPSURF)).AND.
     c    (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c    SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c    (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c    SURFVALUESPEC(UEFTOPSURF))) THEN
             WRITE(30,3120) LN, FRUEFML(COLUMN,ROW),
     c       (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c       (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
3120         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
             LN=LN+1
             WRITE(30,3130) LN, FRUEFML(COLUMN,ROW),
     c       (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c       (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
```

```
     c          CRAUNIV(COLUMN,ROW)
3130            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).GT.
     c       SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,3140) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
3140            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
                WRITE(30,3150) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
3150            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
                WRITE(30,3160) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), GTSECTIRSURF(1),
     c          (-1*GTSECTORSURF(1)), CRAUNIV(COLUMN,ROW)
3160            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTTOPSURF).EQ.
     c       SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,3170) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
3170            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
                WRITE(30,3180) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
3180            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c       SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).GT.
```

```
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c        SURFVALUESPEC(UEFTOPSURF))) THEN
                 WRITE(30,3190) LN, FRUEFML(COLUMN,ROW),
     c           (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c           (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 3190            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                 LN=LN+1
                 WRITE(30,3200) LN, FRUEFML(COLUMN,ROW),
     c           (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c           (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c           CRAUNIV(COLUMN,ROW)
 3200            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                 LN=LN+1
                 WRITE(30,3210) LN, FRUEFML(COLUMN,ROW),
     c           (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c           (-1*CRCLADTOPSURF), CRCLADORSURF,
     c           (-1*GTSECTORSURF(1)), CRAUNIV(COLUMN,ROW)
 3210            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                 LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTTOPSURF).LE.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c        SURFVALUESPEC(UEFTOPSURF))) THEN
                 WRITE(30,3220) LN, FRUEFML(COLUMN,ROW),
     c           (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c           (-1*UEFTOPSURF), CRCLADORSURF, CRAUNIV(COLUMN,ROW)
 3220            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                 LN=LN+1
                 WRITE(30,3230) LN, FRUEFML(COLUMN,ROW),
     c           (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c           (-1*UEFTOPSURF), (-1*CRCLADORSURF),
     c           CRAUNIV(COLUMN,ROW)
 3230            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                 LN=LN+1
              ELSEIF ((SURFVALUESPEC(CRCLADTOPSURF).LE.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTTOPSURF).GE.
     c        SURFVALUESPEC(UEFTOPSURF))) THEN
                 WRITE(30,3240) LN, FRUEFML(COLUMN,ROW),
     c           (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c           (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 3240            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                 LN=LN+1
```

```
                WRITE(30,3250) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
 3250           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(CRCLADTOPSURF).LE.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                WRITE(30,3260) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 3260           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
                WRITE(30,3270) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
 3270           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
                WRITE(30,3275) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c          GTSECTIRSURF(1), CRAUNIV(COLUMN,ROW)
 3275           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(CRCLADTOPSURF).LE.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).LE.
     c       SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                WRITE(30,3280) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), CRAUNIV(COLUMN,ROW)
 3280           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
             ENDIF
       ENDIF
 9300                 FORMAT(T1,'M',I4,T9,' 8016.50c      -0.120',
     c                '                 $ Zirc-4 Cladding')
 9301                     FORMAT(T9,'24050.60c      -0.004')
 7000                     FORMAT(T9,'24052.60c      -0.084')
 7001                     FORMAT(T9,'24053.60c      -0.010')
 7002                     FORMAT(T9,'24054.60c      -0.002')
 9302                     FORMAT(T9,'26054.60c      -0.011')
 7003                     FORMAT(T9,'26056.60c      -0.184')
 7004                     FORMAT(T9,'26057.60c      -0.004')
```

```
 7005            FORMAT(T9,'26058.60c      -0.001')
 9303            FORMAT(T9,'40000.60c      -98.180')
 9304            FORMAT(T9,'50000.35c      -1.400')
 9305       FORMAT(T1,'M',I4,T9,'6000.50c        -0.080',
     c           '             $ SS304 Cladding')
 9306            FORMAT(T9,'7014.50c       -0.100')
 9307            FORMAT(T9,'14000.50c      -0.750')
 9308            FORMAT(T9,'15031.50c      -0.045')
 9309            FORMAT(T9,'16032.50c      -0.030')
 9310            FORMAT(T9,'24050.60c      -0.793')
 7006            FORMAT(T9,'24052.60c      -15.903')
 7007            FORMAT(T9,'24053.60c      -1.838')
 7008            FORMAT(T9,'24054.60c      -0.466')
 9311            FORMAT(T9,'25055.50c      -2.000')
 9312            FORMAT(T9,'26054.60c      -3.918')
 7009            FORMAT(T9,'26056.60c      -63.156')
 7010            FORMAT(T9,'26057.60c      -1.472')
 7011            FORMAT(T9,'26058.60c      -0.200')
 9313            FORMAT(T9,'28058.60c      -6.234')
 7012            FORMAT(T9,'28060.60c      -2.465')
 7013            FORMAT(T9,'28061.60c      -0.109')
 7014            FORMAT(T9,'28062.60c      -0.350')
 7015            FORMAT(T9,'28064.60c      -0.092')
 9314       FORMAT(T1,'M',I4,T9,'6000.50c        -0.080',
     c           '             $ Inconel Cladding')
 9315            FORMAT(T9,'14000.50c      -0.350')
 9316            FORMAT(T9,'15031.50c      -0.015')
 9317            FORMAT(T9,'16032.50c      -0.015')
 9318            FORMAT(T9,'24050.60c      -0.793')
 7016            FORMAT(T9,'24052.60c      -15.903')
 7017            FORMAT(T9,'24053.60c      -1.838')
 7018            FORMAT(T9,'24054.60c      -0.466')
 9319            FORMAT(T9,'25055.50c      -0.350')
 9320            FORMAT(T9,'26054.60c      -0.958')
 7019            FORMAT(T9,'26056.60c      -15.442')
 7020            FORMAT(T9,'26057.60c      -0.360')
 7021            FORMAT(T9,'26058.60c      -0.049')
 9321            FORMAT(T9,'28058.60c      -35.382')
 7022            FORMAT(T9,'28060.60c      -13.993')
 7023            FORMAT(T9,'28061.60c      -0.616')
 7024            FORMAT(T9,'28062.60c      -1.989')
 7025            FORMAT(T9,'28064.60c      -0.520')
 9322            FORMAT(T9,'5010.50c       -1.078E-3')
 9323            FORMAT(T9,'5011.56c       -4.925E-3')
 9324            FORMAT(T9,'13027.50c      -0.500')
 9325            FORMAT(T9,'22000.50c      -0.900')
 9326            FORMAT(T9,'27059.50c      -1.000')
 9327            FORMAT(T9,'29063.60c      -0.205')
 7026            FORMAT(T9,'29065.60c      -0.095')
 9328            FORMAT(T9,'41093.50c      -2.563')
 9329            FORMAT(T9,'42000.50c      -3.050')
 9330            FORMAT(T9,'73181.50c      -2.563')
      ENDIF
 3720    CONTINUE
```

```
 3730 CONTINUE

      RETURN
      END

************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
      SUBROUTINE WESTBPR(BANKNUM, BMODML, BPCLADML, BPNODEML,
     c BPRAUNIV, BPRCLADMAT, BPRLPML, BPRUPML, COLUMN, DESNUM,
     c FRLEFML, FRUEFML, FRUREGIONML, GTAXMAT, GTAXML, GTDATA,
     c GTMAT, GTML, GTSPLIT, HOMOSPACMLNUM, LN, MN,
     c NUMOFBPRANODES, NUMOFGTAXS, NUMOFSPACERS,
     c NUMREGABOVEBPRA, ROW, SN, SYSTEMTOP, WBPRA,
     c WBPRATYPE, AL2O3B4CDENSITY, AL2O3DENSITY,
     c BOTBPNODEHEIGHT, BPDENTOGO, BPNONABSMAT,
     c BPRAXDIM, BPRDIM, BPRLPLENMAT, BPRLPLENWTS,
     c BPRPLEN, BPRUPLENMAT, BPRUPLENWTS, ENDFITHEIGHT,
     c GTAXDATA, HOMOSPACERDEN, LEFMAT, MCNPBPRAHEIGHT,
     c MODDENSITY, NONBPMATDATA, REGABOVEBPRA,
     c SPACERDIST, SPACERHEIGHT, SURFVALUESPEC, UEFMAT,
     c BANKDES, BPRABSNODE, BPRLPLENZAIDS,
     c BPRUPLENZAIDS, CURRENTSURFLABEL, SURFTYPESPEC,
     c NODEBOTTOMSURF)
*
      INTEGER BANKNUM(50,50), BGT, BMODML, BPCLADBOTTOMSURF,
     c BPCLADIRSURF, BPCLADML(50,50), BPCLADORSURF, BPCLADTOPSURF,
     c BPICIRSURF, BPICORSURF, BPIRSURF, BPLEFTOPSURF,
     c BPNODEBOTTOMSURF, BPNODEML, BPNODETOPSURF, BPNONABSMAT(20),
     c BPOCIRSURF,
     c BPOCORSURF, BPORSURF, BPRADIUS, BPRAUNIV(50,50),
     c BPRCLADMAT(20), BPRLPML(50,50), BPRUPML(50,50), C, CO,
     c COLUMN, CURRENTSURFLABEL,
     c DESNUM(50,50), FRLEFML(50,50), FRUEFML(50,50),
     c FRUREGIONML(50,50,20), GTAXMAT(20,5), GTAXML(50,50,5),
     c GTBOTSURF, GTIRSURF, GTMAT(20), GTML(50,50), GTORSURF,
     c GTSECT, GTSECTBOTSURF(5), GTSECTIRSURF(5), GTSECTORSURF(5),
     c GTSECTTOPSURF(5), GTSPLIT, GTTOPSURF, HOMOSPACMLNUM(20,15),
     c LN, MCNPNODE, MN, NODEBOTTOMSURF, NUMOFBPRANODES(20),
     c NUMOFGTAXS(20), NUMOFSPACERS(20), NUMREGABOVEBPRA, REGION,
     c REGIONBOTTOMSURF, REGIONTOPSURF, RO, ROW, SECT, SN,
     c SPACERBOTTOMSURF, SPACERTOPSURF, SPN, SYSTEMTOP,
     c TOPBPNODETOPSURF, UEFBOTTOMSURF, UEFTOPSURF, V,
     c WATERREGIONBOTTOMSURF, WATERREGIONTOPSURF, WBPRA(20),
     c WBPRATYPE(20), Z
```

```
*
      REAL AL2O3B4CDENSITY(20), AL2O3DENSITY(20),
     c BOTBPNODEHEIGHT(20), BPDENTOGO(50,50,50),
     c BPRAXDIM(20,6), BPRDIM(20,3), BPRLPLENMAT(20,2),
     c BPRLPLENWTS(20,35), BPRPLEN(20,2), BPRUPLENMAT(20,2),
     c BPRUPLENWTS(20,35), CLADRHO, CURRENTSURF,
     c ENDFITHEIGHT(20,2), GTAXDATA(20,4,5), GTDATA(20,4),
     c HOMOSPACERDEN(20,15),
     c LEFMAT(20,2), MCNPBPRAHEIGHT(20,50), MODDENSITY,
     c NONBPMATDATA(20,2), REGABOVEBPRA(20,3), SPACERDIST(20,10),
     c SPACERHEIGHT(20,10), SPACHEIGHT, SURFVALUESPEC(200),
     c TOTBPHEIGHT, UEFMAT(20,2)
*
      CHARACTER BANKDES(20)*5, BPRABSNODE(20,50)*1,
     c BPRLPLENZAIDS(20,35)*9, BPRUPLENZAIDS(20,35)*9,
     c SURFTYPESPEC(200)*2
*
      LOGICAL BPRLPMLUNIQUE, BPRUPMLUNIQUE, CLADMLUNIQUE, LEAVE
*
      IF (GTSPLIT.EQ.1) THEN
         IF (WBPRA(BANKNUM(COLUMN,ROW)).EQ.1) THEN
         DO 110 MCNPNODE=1,NUMOFBPRANODES(BANKNUM(COLUMN,ROW))
*     Define the upper end-fitting bottom surface.
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 10 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 10            CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  UEFBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  UEFBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
*     Define the upper end-fitting top surface.
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),1)+
     c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 20 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
```

```
20          CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               UEFTOPSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               UEFTOPSURF=CURRENTSURFLABEL
            ENDIF
*     Define the BP node bounding surfaces.
         IF (MCNPNODE.EQ.1) THEN
            TOTBPHEIGHT=0.0
            DO 30 Z=1,NUMOFBPRANODES(BANKNUM(COLUMN,ROW))
               TOTBPHEIGHT=TOTBPHEIGHT+
     c         MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),Z)
30          CONTINUE
            CURRENTSURF=BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))+
     c      TOTBPHEIGHT
            IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
               CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
            ENDIF
            CURRENTSURFLABEL=0
            DO 40 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
40          CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               TOPBPNODETOPSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               TOPBPNODETOPSURF=CURRENTSURFLABEL
            ENDIF
            BPNODETOPSURF=TOPBPNODETOPSURF
            CURRENTSURF=SURFVALUESPEC(BPNODETOPSURF)-
     c      MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),MCNPNODE)
            IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
               CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
            ENDIF
            CURRENTSURFLABEL=0
            DO 50 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
50          CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
```

```
               BPNODEBOTTOMSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               BPNODEBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
            CURRENTSURF=BPRDIM(BANKNUM(COLUMN,ROW),1)
            CURRENTSURFLABEL=0
            DO 60 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'CZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                 ENDIF
               ENDIF
 60         CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               BPRADIUS=SN
               SURFTYPESPEC(SN)='CZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               BPRADIUS=CURRENTSURFLABEL
            ENDIF
         ELSEIF (MCNPNODE.NE.1) THEN
            BPNODETOPSURF=BPNODEBOTTOMSURF
            CURRENTSURF=SURFVALUESPEC(BPNODETOPSURF)-
     c      MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),MCNPNODE)
            IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
               CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
            ENDIF
            CURRENTSURFLABEL=0
            DO 70 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                 ENDIF
               ENDIF
 70         CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               BPNODEBOTTOMSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               BPNODEBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
         ENDIF
      IF (SURFVALUESPEC(BPNODEBOTTOMSURF).LT.
     c SURFVALUESPEC(UEFTOPSURF)) THEN
*    Write the BP node cells in this BPR universe.
      IF ((BPRABSNODE(BANKNUM(COLUMN,ROW),MCNPNODE).EQ.'Y').AND.
```

```
     c       (BPNONABSMAT(BANKNUM(COLUMN,ROW)).EQ.1)) THEN
             WRITE(30,80) LN, BPNODEML,
     c       (-1*AL2O3DENSITY(BANKNUM(COLUMN,ROW))), (-1*BPRADIUS),
     c       (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c       BPRAUNIV(COLUMN,ROW), MCNPNODE
  80         FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'    $ Burnable poison node ',I2)
             LN=LN+1
             BPNODEML=BPNODEML+1
          ELSEIF ((BPRABSNODE(BANKNUM(COLUMN,ROW),MCNPNODE).EQ.'Y').AND.
     c       (BPNONABSMAT(BANKNUM(COLUMN,ROW)).NE.1)) THEN
             WRITE(30,90) LN, BPNODEML,
     c       (-1*NONBPMATDATA(BANKNUM(COLUMN,ROW),1)), (-1*BPRADIUS),
     c       (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c       BPRAUNIV(COLUMN,ROW), MCNPNODE
  90         FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'    $ Burnable poison node ',I2)
             LN=LN+1
             BPNODEML=BPNODEML+1
          ELSE
             WRITE(30,100) LN, BPNODEML,
     c       (-1*BPDENTOGO(COLUMN,ROW,MCNPNODE)), (-1*BPRADIUS),
     c       (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c       BPRAUNIV(COLUMN,ROW), MCNPNODE
 100         FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'    $ Burnable poison node ',I2)
             LN=LN+1
             BPNODEML=BPNODEML+1
          ENDIF
          ENDIF
 110       CONTINUE
*    Define the BPR cladding inner radius.
               CURRENTSURF=BPRDIM(BANKNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 120 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 120           CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPCLADIRSURF=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPCLADIRSURF=CURRENTSURFLABEL
               ENDIF
*    Define the BPR cladding outer radius.
               CURRENTSURF=BPRDIM(BANKNUM(COLUMN,ROW),3)
               CURRENTSURFLABEL=0
               DO 130 V=1,(SN-1)
```

```
                IF (SURFTYPESPEC(V).EQ.'CZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
 130         CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                BPCLADORSURF=SN
                SURFTYPESPEC(SN)='CZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                BPCLADORSURF=CURRENTSURFLABEL
             ENDIF
*     Define the BPR cladding top surface.
             CURRENTSURF=TOTBPHEIGHT+
     c       BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))+
     c       BPRPLEN(BANKNUM(COLUMN,ROW),1)
             IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
             ENDIF
             CURRENTSURFLABEL=0
             DO 140 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
 140         CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                BPCLADTOPSURF=SN
                SURFTYPESPEC(SN)='PZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                BPCLADTOPSURF=CURRENTSURFLABEL
             ENDIF
*     Define the BPR cladding bottom surface.
             CURRENTSURF=BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))-
     c       BPRPLEN(BANKNUM(COLUMN,ROW),2)
             CURRENTSURFLABEL=0
             DO 150 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
 150         CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                BPCLADBOTTOMSURF=SN
                SURFTYPESPEC(SN)='PZ'
```

```
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPCLADBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
*     Write the BP-to-cladding gap cell in this BPR universe.
            WRITE(30,160) LN, (-1*BPCLADIRSURF), BPRADIUS,
     c      (-1*TOPBPNODETOPSURF),
     c      BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 160        FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,
     c      '    $ Burnable poison-to-cladding gap')
            LN=LN+1
*     Write the BPR cladding cell in this BPR universe.
*     Determine if the BPR cladding material specification has
*     previously been defined.  If it has been previously defined, determine
*     the cladding material specification label.
            CLADMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 180 RO=1,(ROW-1)
                  DO 170 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                        IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  BPRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
 170              CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 180           CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 200 RO=ROW,ROW
                     DO 190 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                           IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                     BPRCLADMAT(BANKNUM(CO,RO))) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
                        ENDIF
 190                 CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
```

```
                  EXIT
               ENDIF
 200        CONTINUE
         ENDIF
      ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
         DO 220 RO=1,(ROW-1)
            DO 210 CO=1,50
               IF (BANKNUM(CO,RO).NE.0) THEN
               IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                  IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
   c              BPRCLADMAT(BANKNUM(CO,RO))) THEN
                     CLADMLUNIQUE=.FALSE.
                     LEAVE=.TRUE.
                     BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                     EXIT
                  ENDIF
               ENDIF
               ENDIF
 210        CONTINUE
            IF (LEAVE.EQ..TRUE.) THEN
               EXIT
            ENDIF
 220     CONTINUE
      ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
         DO 240 RO=1,1
            DO 230 CO=1,(COLUMN-1)
               IF (BANKNUM(CO,RO).NE.0) THEN
               IF (BANKDES(BANKNUM(CO,RO)).NE.'BPRA ') THEN
                  IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
   c              BPRCLADMAT(BANKNUM(CO,RO))) THEN
                     CLADMLUNIQUE=.FALSE.
                     LEAVE=.TRUE.
                     BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                     EXIT
                  ENDIF
               ENDIF
               ENDIF
 230        CONTINUE
            IF (LEAVE.EQ..TRUE.) THEN
               EXIT
            ENDIF
 240     CONTINUE
      ENDIF
      IF (CLADMLUNIQUE.EQ..TRUE.) THEN
         BPCLADML(COLUMN,ROW)=MN
*  Check BPR Cladding Material
            IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
               DO 250 C=1,2
                  IF (C.EQ.1) THEN
                     WRITE(200,9300) BPCLADML(COLUMN,ROW)
                  ELSEIF (C.EQ.2) THEN
                     WRITE(200,9301)
                     WRITE(200,7000)
                     WRITE(200,7001)
```

```
                     WRITE(200,7002)
                     WRITE(200,9302)
                     WRITE(200,7003)
                     WRITE(200,7004)
                     WRITE(200,7005)
                     WRITE(200,9303)
                     WRITE(200,9304)
                  ENDIF
250            CONTINUE
            ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW))
     c      .EQ.2) THEN
               DO 260 C=1,2
                  IF (C.EQ.1) THEN
                     WRITE(200,9305) BPCLADML(COLUMN,ROW)
                  ELSEIF (C.EQ.2) THEN
                     WRITE(200,9306)
                     WRITE(200,9307)
                     WRITE(200,9308)
                     WRITE(200,9309)
                     WRITE(200,9310)
                     WRITE(200,7006)
                     WRITE(200,7007)
                     WRITE(200,7008)
                     WRITE(200,9311)
                     WRITE(200,9312)
                     WRITE(200,7009)
                     WRITE(200,7010)
                     WRITE(200,7011)
                     WRITE(200,9313)
                     WRITE(200,7012)
                     WRITE(200,7013)
                     WRITE(200,7014)
                     WRITE(200,7015)
                  ENDIF
260            CONTINUE
            ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW))
     c      .EQ.3) THEN
               DO 270 C=1,2
                  IF (C.EQ.1) THEN
                     WRITE(200,9314) BPCLADML(COLUMN,ROW)
                  ELSEIF (C.EQ.2) THEN
                     WRITE(200,9315)
                     WRITE(200,9316)
                     WRITE(200,9317)
                     WRITE(200,9318)
                     WRITE(200,7016)
                     WRITE(200,7017)
                     WRITE(200,7018)
                     WRITE(200,9319)
                     WRITE(200,9320)
                     WRITE(200,7019)
                     WRITE(200,7020)
                     WRITE(200,7021)
                     WRITE(200,9321)
```

```
                                WRITE(200,7022)
                                WRITE(200,7023)
                                WRITE(200,7024)
                                WRITE(200,7025)
                                WRITE(200,9322)
                                WRITE(200,9323)
                                WRITE(200,9324)
                                WRITE(200,9325)
                                WRITE(200,9326)
                                WRITE(200,9327)
                                WRITE(200,7026)
                                WRITE(200,9328)
                                WRITE(200,9329)
                                WRITE(200,9330)
                             ENDIF
 270                      CONTINUE
                       ENDIF
                    MN=MN+1
                 ENDIF
                 IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
                    CLADRHO=6.56
                 ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.2) THEN
                    CLADRHO=7.90
                 ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.3) THEN
                    CLADRHO=8.19
                 ENDIF
                 WRITE(30,280) LN, BPCLADML(COLUMN,ROW), (-1*CLADRHO),
     c           BPCLADIRSURF,
     c           (-1*BPCLADORSURF), (-1*BPCLADTOPSURF), BPCLADBOTTOMSURF,
     c           BPRAUNIV(COLUMN,ROW)
 280             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'    $ BPR cladding')
                 LN=LN+1
*      Write the BPR upper plenum cell in this BPR universe.
*      Determine if the BPR upper plenum material specification has
*      previously been defined.  If it has been previously defined, determine
*      the upper plenum material specification label.
                 BPRUPMLUNIQUE=.TRUE.
                 LEAVE=.FALSE.
                 IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                    DO 300 RO=1,(ROW-1)
                       DO 290 CO=1,50
                          IF (BANKNUM(CO,RO).NE.0) THEN
                             IF (BANKNUM(COLUMN,ROW).EQ.
     c                          BANKNUM(CO,RO)) THEN
                                BPRUPMLUNIQUE=.FALSE.
                                LEAVE=.TRUE.
                                BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                                EXIT
                             ENDIF
                          ENDIF
 290                   CONTINUE
                       IF (LEAVE.EQ..TRUE.) THEN
                          EXIT
```

```
                  ENDIF
 300           CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 320 RO=ROW,ROW
                     DO 310 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
                           IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                              BPRUPMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 310                 CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 320              CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 340 RO=1,(ROW-1)
                  DO 330 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                  BANKNUM(CO,RO)) THEN
                           BPRUPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 330              CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 340           CONTINUE
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
               DO 360 RO=1,1
                  DO 350 CO=1,(COLUMN-1)
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                  BANKNUM(CO,RO)) THEN
                           BPRUPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 350              CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 360           CONTINUE
```

```
          ENDIF
       IF (SURFVALUESPEC(TOPBPNODETOPSURF).LT.
    c  SURFVALUESPEC(UEFTOPSURF)) THEN
          IF (BPRUPMLUNIQUE.EQ..TRUE.) THEN
             BPRUPML(COLUMN,ROW)=MN
* Check Burnable Poison Rod Upper Plenum Regions
                DO 390 C=1,BPRUPLENMAT(BANKNUM(COLUMN,ROW),2)
                   IF (C.EQ.1) THEN
                      WRITE(200,370) BPRUPML(COLUMN,ROW),
    c                 BPRUPLENZAIDS(BANKNUM(COLUMN,ROW),C),
    c                 (-1*BPRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 370                  FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
    c                 '    $ Burnable Poison Rod Upper Plenum')
                   ELSE
                      WRITE(200,380)
    c                 BPRUPLENZAIDS(BANKNUM(COLUMN,ROW),C),
    c                 (-1*BPRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 380                  FORMAT(T9,A9,3X,G14.6)
                   ENDIF
 390            CONTINUE
             MN=MN+1
          ENDIF
          WRITE(30,400) LN, BPRUPML(COLUMN,ROW),
    c     (-1*BPRUPLENMAT(BANKNUM(COLUMN,ROW),1)),
    c     TOPBPNODETOPSURF,
    c     (-1*BPCLADTOPSURF), (-1*BPCLADIRSURF),
    c     BPRAUNIV(COLUMN,ROW)
 400      FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
    c     ' IMP:N=1 U=',I3,'    $ BPR upper plenum region')
          LN=LN+1
       ENDIF
*   Write the BPR lower plenum cell (lower end plug) in this BPR universe.
*   Determine if the BPR lower plenum material specification has
*   previously been defined.  If it has been previously defined, determine
*   the lower plenum material specification label.
          BPRLPMLUNIQUE=.TRUE.
          LEAVE=.FALSE.
          IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
             DO 420 RO=1,(ROW-1)
                DO 410 CO=1,50
                   IF (BANKNUM(CO,RO).NE.0) THEN
                      IF (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO)) THEN
                         BPRLPMLUNIQUE=.FALSE.
                         LEAVE=.TRUE.
                         BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                         EXIT
                      ENDIF
                   ENDIF
 410            CONTINUE
                IF (LEAVE.EQ..TRUE.) THEN
                   EXIT
                ENDIF
 420         CONTINUE
             IF (LEAVE.EQ..FALSE.) THEN
```

```
                  DO 440 RO=ROW,ROW
                     DO 430 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
                           IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                              BPRLPMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 430                 CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 440              CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 460 RO=1,(ROW-1)
                  DO 450 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                  BANKNUM(CO,RO)) THEN
                           BPRLPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 450              CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 460           CONTINUE
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
               DO 480 RO=1,1
                  DO 470 CO=1,(COLUMN-1)
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                  BANKNUM(CO,RO)) THEN
                           BPRLPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 470              CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 480           CONTINUE
            ENDIF
            IF (BPRLPMLUNIQUE.EQ..TRUE.) THEN
               BPRLPML(COLUMN,ROW)=MN
```

```
*  Check Burnable Poison Rod Lower Plenum Regions
                  DO 510 C=1,BPRLPLENMAT(BANKNUM(COLUMN,ROW),2)
                     IF (C.EQ.1) THEN
                        WRITE(200,490) BPRLPML(COLUMN,ROW),
     c                  BPRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                  (-1*BPRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 490                    FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                  '    $ Burnable Poison Rod Lower Plenum')
                     ELSE
                        WRITE(200,500)
     c                  BPRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                  (-1*BPRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 500                    FORMAT(T9,A9,3X,G14.6)
                     ENDIF
 510              CONTINUE
               MN=MN+1
            ENDIF
            WRITE(30,520) LN, BPRLPML(COLUMN,ROW),
     c      (-1*BPRLPLENMAT(BANKNUM(COLUMN,ROW),1)), BPCLADBOTTOMSURF,
     c      (-1*BPNODEBOTTOMSURF), (-1*BPCLADIRSURF),
     c      BPRAUNIV(COLUMN,ROW)
 520        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'    $ BPR lower plenum region')
            LN=LN+1
*    Loop through the regions above the BPR (i.e. the appropriate upper core
regions)
*    Define the upper region lower surface.
               DO 560 REGION=1,NUMREGABOVEBPRA
*    Determine the current upper region's lower surface specification.
                  IF (REGION.EQ.1) THEN
                     REGIONTOPSURF=SYSTEMTOP
                     CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
     c               REGABOVEBPRA(REGION,1)
                  ENDIF
                  CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
     c            REGABOVEBPRA(REGION,1)
                  IF (REGION.EQ.NUMREGABOVEBPRA) THEN
                     REGIONBOTTOMSURF=UEFTOPSURF
                  ELSE
                  CURRENTSURFLABEL=0
                  DO 530 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                           CURRENTSURFLABEL=V
                           EXIT
                        ENDIF
                     ENDIF
 530              CONTINUE
                  IF (CURRENTSURFLABEL.EQ.0) THEN
                     REGIONBOTTOMSURF=SN
                     SURFTYPESPEC(SN)='PZ'
                     SURFVALUESPEC(SN)=CURRENTSURF
                     SN=SN+1
                  ELSE
```

```
                    REGIONBOTTOMSURF=CURRENTSURFLABEL
                 ENDIF
                 ENDIF
*     Write the cell specification for the BPR universe upper region.
               IF (REGION.EQ.1) THEN
                 WRITE(30,540) LN, FRUREGIONML(COLUMN,ROW,REGION),
    c            (-1*REGABOVEBPRA(REGION,2)),
    c            REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
  540            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
    c            ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                 LN=LN+1
                 REGIONTOPSURF=REGIONBOTTOMSURF
               ELSE
                 WRITE(30,550) LN, FRUREGIONML(COLUMN,ROW,REGION),
    c            (-1*REGABOVEBPRA(REGION,2)), (-1*REGIONTOPSURF),
    c            REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
  550            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
    c            ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                 LN=LN+1
                 REGIONTOPSURF=REGIONBOTTOMSURF
               ENDIF
  560          CONTINUE
      DO 610 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
*     Define the GT section top surface.
               CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),3,SECT)
               IF (CURRENTSURF.GT.SURFVALUESPEC(UEFTOPSURF)) THEN
                 CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
               ENDIF
               CURRENTSURFLABEL=0
               DO 570 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                   ENDIF
                 ENDIF
  570          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                 GTSECTTOPSURF(SECT)=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
               ELSE
                 GTSECTTOPSURF(SECT)=CURRENTSURFLABEL
               ENDIF
*     Define the GT section bottom surface.
               CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),4,SECT)
               CURRENTSURFLABEL=0
               DO 580 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                   ENDIF
```

```
               ENDIF
 580         CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                GTSECTBOTSURF(SECT)=SN
                SURFTYPESPEC(SN)='PZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                GTSECTBOTSURF(SECT)=CURRENTSURFLABEL
             ENDIF
*     Define the GT section outer radius surface.
             CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),2,SECT)
             CURRENTSURFLABEL=0
             DO 590 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'CZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
 590         CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                GTSECTORSURF(SECT)=SN
                SURFTYPESPEC(SN)='CZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                GTSECTORSURF(SECT)=CURRENTSURFLABEL
             ENDIF
*     Define the GT section inner radius surface.
             CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),1,SECT)
             CURRENTSURFLABEL=0
             DO 600 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'CZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
 600         CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                GTSECTIRSURF(SECT)=SN
                SURFTYPESPEC(SN)='CZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                GTSECTIRSURF(SECT)=CURRENTSURFLABEL
             ENDIF
 610 CONTINUE
*     Write the GT material cell
         DO 740 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
*     Determine if the GT material specification has
*     previously been defined.  If it has been previously defined, determine
*     the material specification label.
```

```
            CLADMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 630 RO=1,(ROW-1)
                  DO 620 CO=1,50
                     IF ((DESNUM(CO,RO).NE.0).AND.
     c               (BANKNUM(CO,RO).EQ.0)) THEN
                        IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                  GTMAT(DESNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                           EXIT
                        ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                  GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                           EXIT
                        ENDIF
                     ENDIF
 620              CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 630           CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 650 RO=ROW,ROW
                     DO 640 CO=1,(COLUMN-1)
                        IF ((DESNUM(CO,RO).NE.0).AND.
     c                  (BANKNUM(CO,RO).EQ.0)) THEN
                           IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                     GTMAT(DESNUM(CO,RO))) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                              EXIT
                           ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                     GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                              EXIT
                           ENDIF
                        ENDIF
 640                 CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 650              CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 670 RO=1,(ROW-1)
                  DO 660 CO=1,50
```

```
                    IF ((DESNUM(CO,RO).NE.0).AND.
     c                 (BANKNUM(CO,RO).EQ.0)) THEN
                       IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                 GTMAT(DESNUM(CO,RO))) THEN
                          CLADMLUNIQUE=.FALSE.
                          LEAVE=.TRUE.
                          GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                          EXIT
                       ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                 GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                          CLADMLUNIQUE=.FALSE.
                          LEAVE=.TRUE.
                          GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                          EXIT
                       ENDIF
                    ENDIF
 660             CONTINUE
                 IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                 ENDIF
 670          CONTINUE
           ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
              DO 690 RO=1,1
                 DO 680 CO=1,(COLUMN-1)
                    IF ((DESNUM(CO,RO).NE.0).AND.
     c                 (BANKNUM(CO,RO).EQ.0)) THEN
                       IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                 GTMAT(DESNUM(CO,RO))) THEN
                          CLADMLUNIQUE=.FALSE.
                          LEAVE=.TRUE.
                          GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                          EXIT
                       ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                 GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                          CLADMLUNIQUE=.FALSE.
                          LEAVE=.TRUE.
                          GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                          EXIT
                       ENDIF
                    ENDIF
 680             CONTINUE
                 IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                 ENDIF
 690          CONTINUE
           ENDIF
           IF (CLADMLUNIQUE.EQ..TRUE.) THEN
              GTAXML(COLUMN,ROW,SECT)=MN
*  Check Guide Tube Material
                 IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.1) THEN
                    DO 700 C=1,2
                       IF (C.EQ.1) THEN
                          WRITE(200,9300) GTAXML(COLUMN,ROW,SECT)
                       ELSEIF (C.EQ.2) THEN
```

```
                  WRITE(200,9301)
                  WRITE(200,7000)
                  WRITE(200,7001)
                  WRITE(200,7002)
                  WRITE(200,9302)
                  WRITE(200,7003)
                  WRITE(200,7004)
                  WRITE(200,7005)
                  WRITE(200,9303)
                  WRITE(200,9304)
               ENDIF
700         CONTINUE
         ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT)
     c   .EQ.2) THEN
            DO 710 C=1,2
               IF (C.EQ.1) THEN
                  WRITE(200,9305) GTAXML(COLUMN,ROW,SECT)
               ELSEIF (C.EQ.2) THEN
                  WRITE(200,9306)
                  WRITE(200,9307)
                  WRITE(200,9308)
                  WRITE(200,9309)
                  WRITE(200,9310)
                  WRITE(200,7006)
                  WRITE(200,7007)
                  WRITE(200,7008)
                  WRITE(200,9311)
                  WRITE(200,9312)
                  WRITE(200,7009)
                  WRITE(200,7010)
                  WRITE(200,7011)
                  WRITE(200,9313)
                  WRITE(200,7012)
                  WRITE(200,7013)
                  WRITE(200,7014)
                  WRITE(200,7015)
               ENDIF
710         CONTINUE
         ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT)
     c   .EQ.3) THEN
            DO 720 C=1,2
               IF (C.EQ.1) THEN
                  WRITE(200,9314) GTAXML(COLUMN,ROW,SECT)
               ELSEIF (C.EQ.2) THEN
                  WRITE(200,9315)
                  WRITE(200,9316)
                  WRITE(200,9317)
                  WRITE(200,9318)
                  WRITE(200,7016)
                  WRITE(200,7017)
                  WRITE(200,7018)
                  WRITE(200,9319)
                  WRITE(200,9320)
                  WRITE(200,7019)
```

```
                        WRITE(200,7020)
                        WRITE(200,7021)
                        WRITE(200,9321)
                        WRITE(200,7022)
                        WRITE(200,7023)
                        WRITE(200,7024)
                        WRITE(200,7025)
                        WRITE(200,9322)
                        WRITE(200,9323)
                        WRITE(200,9324)
                        WRITE(200,9325)
                        WRITE(200,9326)
                        WRITE(200,9327)
                        WRITE(200,7026)
                        WRITE(200,9328)
                        WRITE(200,9329)
                        WRITE(200,9330)
                     ENDIF
 720              CONTINUE
               ENDIF
            MN=MN+1
         ENDIF
         IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.1) THEN
            CLADRHO=6.56
         ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.2) THEN
            CLADRHO=7.90
         ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.3) THEN
            CLADRHO=8.19
         ENDIF
         WRITE(30,730) LN, GTAXML(COLUMN,ROW,SECT), (-1*CLADRHO),
   c     GTSECTIRSURF(SECT),
   c     (-1*GTSECTORSURF(SECT)), (-1*GTSECTTOPSURF(SECT)),
   c     GTSECTBOTSURF(SECT), BPRAUNIV(COLUMN,ROW)
 730     FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
   c     ' IMP:N=1 U=',I4,'   $ Guide tube')
         LN=LN+1
 740   CONTINUE
*    Loop through the spacer and moderator regions along the axial
*    length of the GT (from top to bottom).
       SPACHEIGHT=0.0
       DO 750 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
          SPACHEIGHT=SPACHEIGHT+SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
 750   CONTINUE
       DO 940 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
*    Define the homogenized spacer region bounding surfaces.
         IF (SPN.EQ.1) THEN
            SPACERTOPSURF=UEFBOTTOMSURF
            CURRENTSURF=SURFVALUESPEC(UEFBOTTOMSURF)-
   c        SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
            CURRENTSURFLABEL=0
            DO 760 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
```

```
                EXIT
              ENDIF
            ENDIF
  760     CONTINUE
          IF (CURRENTSURFLABEL.EQ.0) THEN
            SPACERBOTTOMSURF=SN
            SURFTYPESPEC(SN)='PZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
          ELSE
            SPACERBOTTOMSURF=CURRENTSURFLABEL
          ENDIF
          WATERREGIONTOPSURF=SPACERBOTTOMSURF
          CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
          CURRENTSURFLABEL=0
          DO 770 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'PZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                CURRENTSURFLABEL=V
                EXIT
              ENDIF
            ENDIF
  770     CONTINUE
          IF (CURRENTSURFLABEL.EQ.0) THEN
            WATERREGIONBOTTOMSURF=SN
            SURFTYPESPEC(SN)='PZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
          ELSE
            WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
          ENDIF
        ELSEIF ((SPN.NE.1).AND.(SPN.NE.
     c  NUMOFSPACERS(DESNUM(COLUMN,ROW)))) THEN
          SPACERTOPSURF=WATERREGIONBOTTOMSURF
          CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c    SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
          CURRENTSURFLABEL=0
          DO 780 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'PZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                CURRENTSURFLABEL=V
                EXIT
              ENDIF
            ENDIF
  780     CONTINUE
          IF (CURRENTSURFLABEL.EQ.0) THEN
            SPACERBOTTOMSURF=SN
            SURFTYPESPEC(SN)='PZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
          ELSE
            SPACERBOTTOMSURF=CURRENTSURFLABEL
          ENDIF
          WATERREGIONTOPSURF=SPACERBOTTOMSURF
```

```
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
               CURRENTSURFLABEL=0
               DO 790 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 790           CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  WATERREGIONBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
            ELSEIF (SPN.EQ.NUMOFSPACERS(DESNUM(COLUMN,ROW))) THEN
               SPACERTOPSURF=WATERREGIONBOTTOMSURF
               CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
    c          SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
               CURRENTSURFLABEL=0
               DO 800 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 800           CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  SPACERBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  SPACERBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
               WATERREGIONTOPSURF=SPACERBOTTOMSURF
               WATERREGIONBOTTOMSURF=NODEBOTTOMSURF
            ENDIF
*     Write the current homogenized spacer region cell in this GT universe.
           DO 930 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
              IF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
    c         SURFVALUESPEC(SPACERTOPSURF)).AND.
    c         (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
    c         SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
              WRITE(30,810) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
    c         (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
    c         GTSECTORSURF(SECT),
    c         (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
    c         SPN
 810          FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
```

```
     c         ' IMP:N=1 U=',I4,
     c         '    $ Homogenized region for spacer ',I2)
               LN=LN+1
                ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c         SURFVALUESPEC(SPACERTOPSURF)).AND.
     c         (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c         SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
               WRITE(30,820) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c         (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c         GTSECTORSURF(SECT),
     c         (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c         SPN
 820           FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I4,
     c         '    $ Homogenized region for spacer ',I2)
               LN=LN+1
                ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c         SURFVALUESPEC(SPACERTOPSURF)).AND.
     c         (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c         SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
               WRITE(30,830) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c         (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c         GTSECTORSURF(SECT),
     c         (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c         SPN
 830           FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I4,
     c         '    $ Homogenized region for spacer ',I2)
               LN=LN+1
                ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c         SURFVALUESPEC(SPACERTOPSURF)).AND.
     c         (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c         SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
               WRITE(30,840) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c         (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c         GTSECTORSURF(SECT),
     c         (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c         SPN
 840           FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I4,
     c         '    $ Homogenized region for spacer ',I2)
               LN=LN+1
                ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c         SURFVALUESPEC(SPACERTOPSURF)).AND.
     c         (SURFVALUESPEC(GTSECTBOTSURF(SECT)).GT.
     c         SURFVALUESPEC(SPACERBOTTOMSURF)).AND.
     c         (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c         SURFVALUESPEC(SPACERTOPSURF))) THEN
               WRITE(30,850) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c         (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c         GTSECTORSURF(SECT),
     c         (-1*SPACERTOPSURF), GTSECTBOTSURF(SECT),
     c         BPRAUNIV(COLUMN,ROW), SPN
 850           FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
```

```
     c       ' IMP:N=1 U=',I4,
     c       '    $ Homogenized region for spacer ',I2)
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).LT.
     c       SURFVALUESPEC(SPACERTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(SPACERBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
             WRITE(30,860) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c       (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c       GTSECTORSURF(SECT),
     c       (-1*GTSECTTOPSURF(SECT)), SPACERBOTTOMSURF,
     c       BPRAUNIV(COLUMN,ROW), SPN
 860         FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,
     c       '    $ Homogenized region for spacer ',I2)
             LN=LN+1
              ENDIF
*     Write the water region cell below the current homogenized spacer cell
in this GT universe.
              IF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
             WRITE(30,870) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c       BPRAUNIV(COLUMN,ROW)
 870         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'    $ Borated moderator region')
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
             WRITE(30,880) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c       BPRAUNIV(COLUMN,ROW)
 880         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'    $ Borated moderator region')
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
             WRITE(30,890) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c       BPRAUNIV(COLUMN,ROW)
 890         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'    $ Borated moderator region')
             LN=LN+1
```

```
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
             WRITE(30,900) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c       BPRAUNIV(COLUMN,ROW)
 900         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'    $ Borated moderator region')
             LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).GT.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(WATERREGIONTOPSURF))) THEN
             WRITE(30,910) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*WATERREGIONTOPSURF), GTSECTBOTSURF(SECT),
     c       BPRAUNIV(COLUMN,ROW)
 910         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'    $ Borated moderator region')
             LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).LT.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
             WRITE(30,920) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*GTSECTTOPSURF(SECT)), WATERREGIONBOTTOMSURF,
     c       BPRAUNIV(COLUMN,ROW)
 920         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'    $ Borated moderator region')
             LN=LN+1
             ENDIF
 930         CONTINUE
 940      CONTINUE
*  Write the moderator inside of the GT in the BPR universe
          DO 990 GTSECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
             IF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c       SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LE.
     c       SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
*     Write the moderator cells within the GT in this BPR universe.
                WRITE(30,950) LN, BMODML, (-1*MODDENSITY),
     c          (-1*GTSECTIRSURF(GTSECT)),
     c          BPCLADORSURF, (-1*BPCLADTOPSURF),
     c          BPCLADBOTTOMSURF,
     c          BPRAUNIV(COLUMN,ROW)
 950            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
```

```
     c          '   $ Borated moderator inside guide tube')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c       SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LT.
     c       SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).GT.
     c       SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
                WRITE(30,960) LN, BMODML, (-1*MODDENSITY),
     c          (-1*GTSECTIRSURF(GTSECT)),
     c          BPCLADORSURF, (-1*BPCLADTOPSURF),
     c          GTSECTBOTSURF(GTSECT),
     c          BPRAUNIV(COLUMN,ROW)
 960            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '   $ Borated moderator inside guide tube')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).LT.
     c       SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LE.
     c       SURFVALUESPEC(BPCLADBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GT.
     c       SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
                WRITE(30,970) LN, BMODML, (-1*MODDENSITY),
     c          (-1*GTSECTIRSURF(GTSECT));
     c          BPCLADORSURF, (-1*GTSECTTOPSURF(GTSECT)),
     c          BPCLADBOTTOMSURF,
     c          BPRAUNIV(COLUMN,ROW)
 970            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '   $ Borated moderator inside guide tube')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).LT.
     c       SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).GT.
     c       SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
                WRITE(30,980) LN, BMODML, (-1*MODDENSITY),
     c          (-1*GTSECTIRSURF(GTSECT)),
     c          BPCLADORSURF, (-1*GTSECTTOPSURF(GTSECT)),
     c          GTSECTBOTSURF(GTSECT),
     c          BPRAUNIV(COLUMN,ROW)
 980            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '   $ Borated moderator inside guide tube')
                LN=LN+1
             ENDIF
 990     CONTINUE
*   Determine the axial GT section which contains the lowest BPR axial
section
       DO 1000 GTSECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
         IF ((SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LT.
     c   SURFVALUESPEC(BPCLADBOTTOMSURF)).AND.
     c   (SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c   SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
```

```
            BGT=GTSECT
            EXIT
         ENDIF
 1000 CONTINUE
      DO 1030 GTSECT=BGT,NUMOFGTAXS(DESNUM(COLUMN,ROW))
         IF (GTSECT.EQ.BGT) THEN
            WRITE(30,1010) LN, BMODML, (-1*MODDENSITY),
     c      (-1*GTSECTIRSURF(GTSECT)),
     c      (-1*BPCLADBOTTOMSURF),
     c      GTSECTBOTSURF(GTSECT),
     c      BPRAUNIV(COLUMN,ROW)
 1010       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,
     c      '    $ Borated moderator inside guide tube')
            LN=LN+1
         ELSE
            WRITE(30,1020) LN, BMODML, (-1*MODDENSITY),
     c      (-1*GTSECTIRSURF(GTSECT)),
     c      (-1*GTSECTTOPSURF(GTSECT)),
     c      GTSECTBOTSURF(GTSECT),
     c      BPRAUNIV(COLUMN,ROW)
 1020       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,
     c      '    $ Borated moderator inside guide tube')
            LN=LN+1
         ENDIF
 1030 CONTINUE
*     Define the lower end-fitting top surface.
               CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 1040 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 1040          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPLEFTOPSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPLEFTOPSURF=CURRENTSURFLABEL
               ENDIF
*     Write the lower end-fitting cell specification for this BPR universe.
            GTBOTSURF=GTSECTBOTSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))
            IF (SURFVALUESPEC(GTBOTSURF).GE.
     c      ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)) THEN
               WRITE(30,1050) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*BPLEFTOPSURF),
     c         BPRAUNIV(COLUMN,ROW)
 1050          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,' IMP:N=1 U=',I3,
```

```
     c         '   $ Lower end-fitting')
               LN=LN+1
            ELSE
               WRITE(30,1060) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*BPLEFTOPSURF),
     c         GTSECTORSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW))),
     c         BPRAUNIV(COLUMN,ROW)
 1060          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'    $ Lower end-fitting')
               LN=LN+1
               WRITE(30,1070) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)),
     c         (-1*GTSECTBOTSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))),
     c         (-1*GTSECTORSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))),
     c         BPRAUNIV(COLUMN,ROW)
 1070          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'    $ Lower end-fitting')
               LN=LN+1
            ENDIF
*     Write the upper end-fitting cell specification for this BPR universe.
            IF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c      SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(GTSECTTOPSURF(1)).LE.
     c      SURFVALUESPEC(UEFBOTTOMSURF))) THEN
               WRITE(30,1080) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c         (-1*UEFTOPSURF), BPRAUNIV(COLUMN,ROW)
 1080          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
               LN=LN+1
            ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c      SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c      SURFVALUESPEC(UEFTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTTOPSURF(1)).LE.
     c      SURFVALUESPEC(UEFBOTTOMSURF))) THEN
               WRITE(30,1090) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c         (-1*UEFTOPSURF), BPCLADORSURF, BPRAUNIV(COLUMN,ROW)
 1090          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
               LN=LN+1
               WRITE(30,1100) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), BPCLADTOPSURF,
     c         (-1*UEFTOPSURF), (-1*BPCLADORSURF),
     c         BPRAUNIV(COLUMN,ROW)
 1100          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
               LN=LN+1
            ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c      SURFVALUESPEC(GTSECTTOPSURF(1))).AND.
     c      (SURFVALUESPEC(GTSECTTOPSURF(1)).LT.
     c      SURFVALUESPEC(UEFTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTTOPSURF(1)).GT.
```

```
     c       SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                WRITE(30,1110) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), BPRAUNIV(COLUMN,ROW)
 1110           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,1120) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTSECTTOPSURF(1),
     c          (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c          BPRAUNIV(COLUMN,ROW)
 1120           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,1130) LN, BMODML,
     c          (-1*MODDENSITY), BPCLADTOPSURF,
     c          (-1*GTSECTTOPSURF(1)), (-1*GTSECTIRSURF(1)),
     c          BPRAUNIV(COLUMN,ROW)
 1130           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c       SURFVALUESPEC(GTSECTTOPSURF(1))).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(1)).EQ.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,1140) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1),
     c          BPRAUNIV(COLUMN,ROW)
 1140           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
             IF (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c       SURFVALUESPEC(GTSECTTOPSURF(1))) THEN
                WRITE(30,1150) LN, BMODML,
     c          (-1*MODDENSITY), BPCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c          BPRAUNIV(COLUMN,ROW)
 1150           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
             ENDIF
             ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(1)).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(1)).LT.
     c       SURFVALUESPEC(BPCLADTOPSURF))) THEN
                WRITE(30,1160) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1),
     c          BPRAUNIV(COLUMN,ROW)
```

```
1160          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
    c         ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
              LN=LN+1
              WRITE(30,1170) LN, FRUEFML(COLUMN,ROW),
    c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)),
    c         GTSECTTOPSURF(1),
    c         (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
    c         BPCLADORSURF, BPRAUNIV(COLUMN,ROW)
1170          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
    c         1X,I4,' IMP:N=1 U=',I3,
    c         '    $ Assembly upper end-fitting')
              LN=LN+1
              WRITE(30,1180) LN, FRUEFML(COLUMN,ROW),
    c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), BPCLADTOPSURF,
    c         (-1*UEFTOPSURF), (-1*BPCLADORSURF),
    c         BPRAUNIV(COLUMN,ROW)
1180          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
    c         ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
              LN=LN+1
           ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).EQ.
    c      SURFVALUESPEC(UEFTOPSURF)).AND.
    c      (SURFVALUESPEC(GTSECTTOPSURF(1)).GT.
    c      SURFVALUESPEC(UEFBOTTOMSURF)).AND.
    c      (SURFVALUESPEC(GTSECTTOPSURF(1)).LT.
    c      SURFVALUESPEC(BPCLADTOPSURF))) THEN
              WRITE(30,1190) LN, FRUEFML(COLUMN,ROW),
    c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
    c         (-1*UEFTOPSURF), GTSECTORSURF(1),
    c         BPRAUNIV(COLUMN,ROW)
1190          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
    c         ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
              LN=LN+1
              WRITE(30,1200) LN, FRUEFML(COLUMN,ROW),
    c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)),
    c         GTSECTTOPSURF(1),
    c         (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
    c         BPCLADORSURF, BPRAUNIV(COLUMN,ROW)
1200          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
    c         1X,I4,' IMP:N=1 U=',I3,
    c         '    $ Assembly upper end-fitting')
              LN=LN+1
           ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
    c      SURFVALUESPEC(UEFBOTTOMSURF)).AND.
    c      (SURFVALUESPEC(BPCLADTOPSURF).LT.
    c      SURFVALUESPEC(UEFTOPSURF)).AND.
    c      (SURFVALUESPEC(GTSECTTOPSURF(1)).EQ.
    c      SURFVALUESPEC(BPCLADTOPSURF))) THEN
              WRITE(30,1210) LN, FRUEFML(COLUMN,ROW),
    c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
    c         (-1*UEFTOPSURF), GTSECTORSURF(1),
    c         BPRAUNIV(COLUMN,ROW)
1210          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
    c         ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
              LN=LN+1
```

```
          WRITE(30,1220) LN, FRUEFML(COLUMN,ROW),
     c    (-1*UEFMAT(DESNUM(COLUMN,ROW),1)),
     c    GTSECTTOPSURF(1),
     c    (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c    BPCLADORSURF, BPRAUNIV(COLUMN,ROW)
 1220     FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c    1X,I4,' IMP:N=1 U=',I3,
     c    '   $ Assembly upper end-fitting')
          LN=LN+1
        ENDIF
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
      ELSEIF (WBPRA(BANKNUM(COLUMN,ROW)).EQ.2) THEN
      DO 1390 MCNPNODE=1,NUMOFBPRANODES(BANKNUM(COLUMN,ROW))
*    Define the upper end-fitting bottom surface.
          CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c    ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
          CURRENTSURFLABEL=0
          DO 1230 V=1,(SN-1)
             IF (SURFTYPESPEC(V).EQ.'PZ') THEN
        IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                CURRENTSURFLABEL=V
                EXIT
             ENDIF
           ENDIF
 1230     CONTINUE
          IF (CURRENTSURFLABEL.EQ.0) THEN
             UEFBOTTOMSURF=SN
             SURFTYPESPEC(SN)='PZ'
             SURFVALUESPEC(SN)=CURRENTSURF
             SN=SN+1
          ELSE
             UEFBOTTOMSURF=CURRENTSURFLABEL
          ENDIF
*    Define the upper end-fitting top surface.
```

```
            CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c      ENDFITHEIGHT(DESNUM(COLUMN,ROW),1)+
     c      ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
            CURRENTSURFLABEL=0
            DO 1240 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 1240       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               UEFTOPSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               UEFTOPSURF=CURRENTSURFLABEL
            ENDIF
*     Define the inner BPR cladding inner radius.
            CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),1)
            CURRENTSURFLABEL=0
            DO 1250 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'CZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 1250       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               BPICIRSURF=SN
               SURFTYPESPEC(SN)='CZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               BPICIRSURF=CURRENTSURFLABEL
            ENDIF
*     Define the inner BPR cladding outer radius.
            CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),2)
            CURRENTSURFLABEL=0
            DO 1260 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'CZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 1260       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               BPICORSURF=SN
               SURFTYPESPEC(SN)='CZ'
               SURFVALUESPEC(SN)=CURRENTSURF
```

```
            SN=SN+1
         ELSE
            BPICORSURF=CURRENTSURFLABEL
         ENDIF
*     Define the BP absorber inner radius.
         CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),3)
         CURRENTSURFLABEL=0
         DO 1270 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'CZ') THEN
       IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
 1270    CONTINUE
         IF (CURRENTSURFLABEL.EQ.0) THEN
            BPIRSURF=SN
            SURFTYPESPEC(SN)='CZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
         ELSE
            BPIRSURF=CURRENTSURFLABEL
         ENDIF
*     Define the BP absorber outer radius.
         CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),4)
         CURRENTSURFLABEL=0
         DO 1280 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'CZ') THEN
       IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
 1280    CONTINUE
         IF (CURRENTSURFLABEL.EQ.0) THEN
            BPORSURF=SN
            SURFTYPESPEC(SN)='CZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
         ELSE
            BPORSURF=CURRENTSURFLABEL
         ENDIF
*     Define the outer BPR cladding inner radius.
         CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),5)
         CURRENTSURFLABEL=0
         DO 1290 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'CZ') THEN
       IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
 1290    CONTINUE
         IF (CURRENTSURFLABEL.EQ.0) THEN
```

```
                BPOCIRSURF=SN
                SURFTYPESPEC(SN)='CZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                BPOCIRSURF=CURRENTSURFLABEL
             ENDIF
*     Define the outer BPR cladding outer radius.
             CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),6)
             CURRENTSURFLABEL=0
             DO 1300 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'CZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
 1300        CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                BPOCORSURF=SN
                SURFTYPESPEC(SN)='CZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                BPOCORSURF=CURRENTSURFLABEL
             ENDIF
*     Define the BP node bounding surfaces.
          IF (MCNPNODE.EQ.1) THEN
             TOTBPHEIGHT=0.0
             DO 1310 Z=1,NUMOFBPRANODES(BANKNUM(COLUMN,ROW))
                TOTBPHEIGHT=TOTBPHEIGHT+
     c          MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),Z)
 1310        CONTINUE
             CURRENTSURF=BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))+
     c       TOTBPHEIGHT
             IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
             ENDIF
             CURRENTSURFLABEL=0
             DO 1320 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
 1320        CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                TOPBPNODETOPSURF=SN
                SURFTYPESPEC(SN)='PZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                TOPBPNODETOPSURF=CURRENTSURFLABEL
```

```
              ENDIF
              BPNODETOPSURF=TOPBPNODETOPSURF
              CURRENTSURF=SURFVALUESPEC(BPNODETOPSURF)-
     c        MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),MCNPNODE)
              IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                 CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
              ENDIF
              CURRENTSURFLABEL=0
              DO 1330 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                   ENDIF
                 ENDIF
 1330         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 BPNODEBOTTOMSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 BPNODEBOTTOMSURF=CURRENTSURFLABEL
              ENDIF
           ELSEIF (MCNPNODE.NE.1) THEN
              BPNODETOPSURF=BPNODEBOTTOMSURF
              CURRENTSURF=SURFVALUESPEC(BPNODETOPSURF)-
     c        MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),MCNPNODE)
              IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                 CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
              ENDIF
              CURRENTSURFLABEL=0
              DO 1350 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                   ENDIF
                 ENDIF
 1350         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 BPNODEBOTTOMSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 BPNODEBOTTOMSURF=CURRENTSURFLABEL
              ENDIF
           ENDIF
        IF (SURFVALUESPEC(BPNODEBOTTOMSURF).LT.
     c  SURFVALUESPEC(UEFTOPSURF)) THEN
*    Write the BP node cells in this BPR universe.
        IF ((BPRABSNODE(BANKNUM(COLUMN,ROW),MCNPNODE).EQ.'Y').AND.
     c     (BPNONABSMAT(BANKNUM(COLUMN,ROW)).EQ.1)) THEN
```

```
            WRITE(30,1360) LN, BPNODEML,
     c      (-1*AL2O3DENSITY(BANKNUM(COLUMN,ROW))), (-1*BPORSURF),
     c      BPIRSURF, (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW), MCNPNODE
 1360       FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c      1X,I4,' IMP:N=1 U=',I3,'    $ Burnable poison node ',I2)
            LN=LN+1
            BPNODEML=BPNODEML+1
         ELSEIF ((BPRABSNODE(BANKNUM(COLUMN,ROW),MCNPNODE).EQ.'Y').AND.
     c      (BPNONABSMAT(BANKNUM(COLUMN,ROW)).NE.1)) THEN
            WRITE(30,1370) LN, BPNODEML,
     c      (-1*NONBPMATDATA(BANKNUM(COLUMN,ROW),1)), (-1*BPORSURF),
     c      BPIRSURF, (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW), MCNPNODE
 1370       FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c      1X,I4,' IMP:N=1 U=',I3,'    $ Burnable poison node ',I2)
            LN=LN+1
            BPNODEML=BPNODEML+1
         ELSE
            WRITE(30,1380) LN, BPNODEML,
     c      (-1*BPDENTOGO(COLUMN,ROW,MCNPNODE)), (-1*BPORSURF),
     c      BPIRSURF, (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW), MCNPNODE
 1380       FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c      1X,I4,' IMP:N=1 U=',I3,'    $ Burnable poison node ',I2)
            LN=LN+1
            BPNODEML=BPNODEML+1
         ENDIF
         ENDIF
 1390     CONTINUE
*    Define the BPR cladding top surface.
               CURRENTSURF=TOTBPHEIGHT+
     c         BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))+
     c         BPRPLEN(BANKNUM(COLUMN,ROW),1)
               IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                  CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
               ENDIF
               CURRENTSURFLABEL=0
               DO 1400 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 1400          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPCLADTOPSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPCLADTOPSURF=CURRENTSURFLABEL
               ENDIF
```

```
*     Define the BPR cladding bottom surface.
               CURRENTSURF=BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))-
     c         BPRPLEN(BANKNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 1410 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 1410          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPCLADBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPCLADBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
*     Write the inner BP-to-cladding gap cell in this BPR universe.
            WRITE(30,1420) LN, (-1*BPIRSURF), BPICORSURF,
     c      (-1*TOPBPNODETOPSURF),
     c      BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 1420       FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,
     c      '   $ Burnable poison-to-cladding gap')
            LN=LN+1
*     Write the outer BP-to-cladding gap cell in this BPR universe.
            WRITE(30,1430) LN, (-1*BPOCIRSURF), BPORSURF,
     c      (-1*TOPBPNODETOPSURF),
     c      BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 1430       FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,
     c      '   $ Burnable poison-to-cladding gap')
            LN=LN+1
*     Write the annular gap cell in this BPR universe.
            WRITE(30,1440) LN, (-1*BPICIRSURF),
     c      (-1*TOPBPNODETOPSURF),
     c      BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 1440       FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,
     c      '   $ Burnable poison-to-cladding gap')
            LN=LN+1
*     Write the BPR cladding cell in this BPR universe.
*     Determine if the BPR cladding material specification has
*     previously been defined.  If it has been previously defined, determine
*     the cladding material specification label.
            CLADMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 1460 RO=1,(ROW-1)
                  DO 1450 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
```

```
                  IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                     IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c               BPRCLADMAT(BANKNUM(CO,RO))) THEN
                        CLADMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
                  ENDIF
 1450          CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
 1460       CONTINUE
            IF (LEAVE.EQ..FALSE.) THEN
               DO 1480 RO=ROW,ROW
                  DO 1470 CO=1,(COLUMN-1)
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                        IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  BPRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
 1470             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 1480          CONTINUE
            ENDIF
         ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
            DO 1500 RO=1,(ROW-1)
               DO 1490 CO=1,50
                  IF (BANKNUM(CO,RO).NE.0) THEN
                  IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                     IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c               BPRCLADMAT(BANKNUM(CO,RO))) THEN
                        CLADMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
                  ENDIF
 1490          CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
 1500       CONTINUE
```

```
                    WRITE(200,9311)
                    WRITE(200,9312)
                    WRITE(200,7009)
                    WRITE(200,7010)
                    WRITE(200,7011)
                    WRITE(200,9313)
                    WRITE(200,7012)
                    WRITE(200,7013)
                    WRITE(200,7014)
                    WRITE(200,7015)
                 ENDIF
 1540         CONTINUE
           ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW))
     c     .EQ.3) THEN
              DO 1550 C=1,2
                 IF (C.EQ.1) THEN
                    WRITE(200,9314) BPCLADML(COLUMN,ROW)
                 ELSEIF (C.EQ.2) THEN
                    WRITE(200,9315)
                    WRITE(200,9316)
                    WRITE(200,9317)
                    WRITE(200,9318)
                    WRITE(200,7016)
                    WRITE(200,7017)
                    WRITE(200,7018)
                    WRITE(200,9319)
                    WRITE(200,9320)
                    WRITE(200,7019)
                    WRITE(200,7020)
                    WRITE(200,7021)
                    WRITE(200,9321)
                    WRITE(200,7022)
                    WRITE(200,7023)
                    WRITE(200,7024)
                    WRITE(200,7025)
                    WRITE(200,9322)
                    WRITE(200,9323)
                    WRITE(200,9324)
                    WRITE(200,9325)
                    WRITE(200,9326)
                    WRITE(200,9327)
                    WRITE(200,7026)
                    WRITE(200,9328)
                    WRITE(200,9329)
                    WRITE(200,9330)
                 ENDIF
 1550         CONTINUE
           ENDIF
        MN=MN+1
     ENDIF
     IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
        CLADRHO=6.56
     ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.2) THEN
        CLADRHO=7.90
```

```
            ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.3) THEN
               CLADRHO=8.19
            ENDIF
            WRITE(30,1560) LN, BPCLADML(COLUMN,ROW), (-1*CLADRHO),
     c      BPOCIRSURF,
     c      (-1*BPOCORSURF), (-1*BPCLADTOPSURF), BPCLADBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW)
 1560       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ BPR cladding')
            LN=LN+1
            WRITE(30,1570) LN, BPCLADML(COLUMN,ROW), (-1*CLADRHO),
     c      BPICIRSURF,
     c      (-1*BPICORSURF), (-1*TOPBPNODETOPSURF), BPNODEBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW)
 1570       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ BPR cladding')
            LN=LN+1
*     Write the BPR upper plenum cell in this BPR universe.
*     Determine if the BPR upper plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the upper plenum material specification label.
            BPRUPMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 1590 RO=1,(ROW-1)
                  DO 1580 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                           BPRUPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 1580             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 1590          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 1610 RO=ROW,ROW
                     DO 1600 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
                           IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                              BPRUPMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 1600                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
```

```
                       EXIT
                    ENDIF
 1610            CONTINUE
              ENDIF
           ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
              DO 1630 RO=1,(ROW-1)
                 DO 1620 CO=1,50
                    IF (BANKNUM(CO,RO).NE.0) THEN
                       IF (BANKNUM(COLUMN,ROW).EQ.
     c                    BANKNUM(CO,RO)) THEN
                          BPRUPMLUNIQUE=.FALSE.
                          LEAVE=.TRUE.
                          BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                          EXIT
                       ENDIF
                    ENDIF
 1620            CONTINUE
                 IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                 ENDIF
 1630         CONTINUE
           ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
              DO 1650 RO=1,1
                 DO 1640 CO=1,(COLUMN-1)
                    IF (BANKNUM(CO,RO).NE.0) THEN
                       IF (BANKNUM(COLUMN,ROW).EQ.
     c                    BANKNUM(CO,RO)) THEN
                          BPRUPMLUNIQUE=.FALSE.
                          LEAVE=.TRUE.
                          BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                          EXIT
                       ENDIF
                    ENDIF
 1640            CONTINUE
                 IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                 ENDIF
 1650         CONTINUE
           ENDIF
        IF (SURFVALUESPEC(TOPBPNODETOPSURF).LT.
     c  SURFVALUESPEC(UEFTOPSURF)) THEN
           IF (BPRUPMLUNIQUE.EQ..TRUE.) THEN
              BPRUPML(COLUMN,ROW)=MN
*  Check Burnable Poison Rod Upper Plenum Regions
                 DO 1680 C=1,BPRUPLENMAT(BANKNUM(COLUMN,ROW),2)
                    IF (C.EQ.1) THEN
                       WRITE(200,1660) BPRUPML(COLUMN,ROW),
     c                 BPRUPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                 (-1*BPRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 1660                  FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                 '     $ Burnable Poison Rod Upper Plenum')
                    ELSE
                       WRITE(200,1670)
     c                 BPRUPLENZAIDS(BANKNUM(COLUMN,ROW),C),
```

```
     c                    (-1*BPRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 1670                     FORMAT(T9,A9,3X,G14.6)
                       ENDIF
 1680               CONTINUE
                 MN=MN+1
              ENDIF
              WRITE(30,1690) LN, BPRUPML(COLUMN,ROW),
     c        (-1*BPRUPLENMAT(BANKNUM(COLUMN,ROW),1)),
     c        TOPBPNODETOPSURF,
     c        (-1*BPCLADTOPSURF), (-1*BPOCIRSURF),
     c        BPRAUNIV(COLUMN,ROW)
 1690         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ BPR upper plenum region')
              LN=LN+1
           ENDIF
*    Write the BPR lower plenum cell (lower end plug) in this BPR universe.
*    Determine if the BPR lower plenum material specification has
*    previously been defined.  If it has been previously defined, determine
*    the lower plenum material specification label.
              BPRLPMLUNIQUE=.TRUE.
              LEAVE=.FALSE.
              IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                 DO 1710 RO=1,(ROW-1)
                    DO 1700 CO=1,50
                       IF (BANKNUM(CO,RO).NE.0) THEN
                          IF (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO)) THEN
                             BPRLPMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                             EXIT
                          ENDIF
                       ENDIF
 1700               CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
 1710            CONTINUE
                 IF (LEAVE.EQ..FALSE.) THEN
                    DO 1730 RO=ROW,ROW
                       DO 1720 CO=1,(COLUMN-1)
                          IF (BANKNUM(CO,RO).NE.0) THEN
                             IF (BANKNUM(COLUMN,ROW).EQ.
     c                       BANKNUM(CO,RO)) THEN
                                BPRLPMLUNIQUE=.FALSE.
                                LEAVE=.TRUE.
                                BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                                EXIT
                             ENDIF
                          ENDIF
 1720                  CONTINUE
                       IF (LEAVE.EQ..TRUE.) THEN
                          EXIT
                       ENDIF
 1730               CONTINUE
```

```
                  ENDIF
               ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                  DO 1750 RO=1,(ROW-1)
                     DO 1740 CO=1,50
                        IF (BANKNUM(CO,RO).NE.0) THEN
                           IF (BANKNUM(COLUMN,ROW).EQ.
     c                        BANKNUM(CO,RO)) THEN
                              BPRLPMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 1740                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 1750             CONTINUE
               ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                  DO 1770 RO=1,1
                     DO 1760 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
                           IF (BANKNUM(COLUMN,ROW).EQ.
     c                        BANKNUM(CO,RO)) THEN
                              BPRLPMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 1760                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 1770             CONTINUE
               ENDIF
               IF (BPRLPMLUNIQUE.EQ..TRUE.) THEN
                  BPRLPML(COLUMN,ROW)=MN
*  Check Burnable Poison Rod Lower Plenum Regions
                     DO 1800 C=1,BPRLPLENMAT(BANKNUM(COLUMN,ROW),2)
                        IF (C.EQ.1) THEN
                           WRITE(200,1780) BPRLPML(COLUMN,ROW),
     c                     BPRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                     (-1*BPRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 1780                      FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                     '    $ Burnable Poison Rod Lower Plenum')
                        ELSE
                           WRITE(200,1790)
     c                     BPRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                     (-1*BPRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 1790                      FORMAT(T9,A9,3X,G14.6)
                        ENDIF
 1800                CONTINUE
                  MN=MN+1
```

```
          ENDIF
          WRITE(30,1810) LN, BPRLPML(COLUMN,ROW),
    c     (-1*BPRLPLENMAT(BANKNUM(COLUMN,ROW),1)), BPCLADBOTTOMSURF,
    c     (-1*BPNODEBOTTOMSURF), (-1*BPOCIRSURF),
    c     BPRAUNIV(COLUMN,ROW)
 1810     FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
    c     ' IMP:N=1 U=',I3,'   $ BPR lower plenum region')
          LN=LN+1
*    Loop through the regions above the BPR (i.e. the appropriate upper core
regions)
*    Define the upper region lower surface.
            DO 1850 REGION=1,NUMREGABOVEBPRA
*    Determine the current upper region's lower surface specification.
               IF (REGION.EQ.1) THEN
                  REGIONTOPSURF=SYSTEMTOP
                  CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
    c             REGABOVEBPRA(REGION,1)
               ENDIF
               CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
    c          REGABOVEBPRA(REGION,1)
               IF (REGION.EQ.NUMREGABOVEBPRA) THEN
                  REGIONBOTTOMSURF=UEFTOPSURF
               ELSE
               CURRENTSURFLABEL=0
               DO 1820 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 1820          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  REGIONBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  REGIONBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
               ENDIF
*    Write the cell specification for the BPR universe upper region.
            IF (REGION.EQ.1) THEN
               WRITE(30,1830) LN, FRUREGIONML(COLUMN,ROW,REGION),
    c          (-1*REGABOVEBPRA(REGION,2)),
    c          REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 1830          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
    c          ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
               LN=LN+1
               REGIONTOPSURF=REGIONBOTTOMSURF
            ELSE
               WRITE(30,1840) LN, FRUREGIONML(COLUMN,ROW,REGION),
    c          (-1*REGABOVEBPRA(REGION,2)), (-1*REGIONTOPSURF),
    c          REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
```

```
1840           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
    c          ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
               LN=LN+1
               REGIONTOPSURF=REGIONBOTTOMSURF
            ENDIF
1850        CONTINUE
      DO 1900 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
*     Define the GT section top surface.
            CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),3,SECT)
            IF (CURRENTSURF.GT.SURFVALUESPEC(UEFTOPSURF)) THEN
               CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
            ENDIF
            CURRENTSURFLABEL=0
            DO 1860 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
1860        CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               GTSECTTOPSURF(SECT)=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               GTSECTTOPSURF(SECT)=CURRENTSURFLABEL
            ENDIF
*     Define the GT section bottom surface.
            CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),4,SECT)
            CURRENTSURFLABEL=0
            DO 1870 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
1870        CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               GTSECTBOTSURF(SECT)=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               GTSECTBOTSURF(SECT)=CURRENTSURFLABEL
            ENDIF
*     Define the GT section outer radius surface.
            CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),2,SECT)
            CURRENTSURFLABEL=0
            DO 1880 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'CZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
```

```
                    CURRENTSURFLABEL=V
                    EXIT
                 ENDIF
              ENDIF
 1880      CONTINUE
           IF (CURRENTSURFLABEL.EQ.0) THEN
              GTSECTORSURF(SECT)=SN
              SURFTYPESPEC(SN)='CZ'
              SURFVALUESPEC(SN)=CURRENTSURF
              SN=SN+1
           ELSE
              GTSECTORSURF(SECT)=CURRENTSURFLABEL
           ENDIF
*     Define the GT section inner radius surface.
           CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),1,SECT)
           CURRENTSURFLABEL=0
           DO 1890 V=1,(SN-1)
              IF (SURFTYPESPEC(V).EQ.'CZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                 ENDIF
              ENDIF
 1890      CONTINUE
           IF (CURRENTSURFLABEL.EQ.0) THEN
              GTSECTIRSURF(SECT)=SN
              SURFTYPESPEC(SN)='CZ'
              SURFVALUESPEC(SN)=CURRENTSURF
              SN=SN+1
           ELSE
              GTSECTIRSURF(SECT)=CURRENTSURFLABEL
           ENDIF
 1900 CONTINUE
*     Write the GT material cell
         DO 2030 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
*     Determine if the GT material specification has
*     previously been defined.  If it has been previously defined, determine
*     the material specification label.
        CLADMLUNIQUE=.TRUE.
        LEAVE=.FALSE.
        IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
           DO 1920 RO=1,(ROW-1)
              DO 1910 CO=1,50
                 IF ((DESNUM(CO,RO).NE.0).AND.
     c           (BANKNUM(CO,RO).EQ.0)) THEN
                    IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c              GTMAT(DESNUM(CO,RO))) THEN
                       CLADMLUNIQUE=.FALSE.
                       LEAVE=.TRUE.
                       GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                       EXIT
                    ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c              GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                       CLADMLUNIQUE=.FALSE.
```

```
                          LEAVE=.TRUE.
                          GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                          EXIT
                       ENDIF
                    ENDIF
 1910            CONTINUE
                 IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                 ENDIF
 1920         CONTINUE
              IF (LEAVE.EQ..FALSE.) THEN
                 DO 1940 RO=ROW,ROW
                    DO 1930 CO=1,(COLUMN-1)
                       IF ((DESNUM(CO,RO).NE.0).AND.
     c                 (BANKNUM(CO,RO).EQ.0)) THEN
                          IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                    GTMAT(DESNUM(CO,RO))) THEN
                             CLADMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                             EXIT
                          ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                    GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                             CLADMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                             EXIT
                          ENDIF
                       ENDIF
 1930               CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
 1940            CONTINUE
              ENDIF
           ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
              DO 1960 RO=1,(ROW-1)
                 DO 1950 CO=1,50
                    IF ((DESNUM(CO,RO).NE.0).AND.
     c              (BANKNUM(CO,RO).EQ.0)) THEN
                       IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                 GTMAT(DESNUM(CO,RO))) THEN
                          CLADMLUNIQUE=.FALSE.
                          LEAVE=.TRUE.
                          GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                          EXIT
                       ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                 GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                          CLADMLUNIQUE=.FALSE.
                          LEAVE=.TRUE.
                          GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                          EXIT
                       ENDIF
                    ENDIF
```

```
 1950             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 1960          CONTINUE
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
               DO 1980 RO=1,1
                  DO 1970 CO=1,(COLUMN-1)
                     IF ((DESNUM(CO,RO).NE.0).AND.
     c                  (BANKNUM(CO,RO).EQ.0)) THEN
                        IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                  GTMAT(DESNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                           EXIT
                        ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                  GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                           EXIT
                        ENDIF
                     ENDIF
 1970             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 1980          CONTINUE
            ENDIF
            IF (CLADMLUNIQUE.EQ..TRUE.) THEN
               GTAXML(COLUMN,ROW,SECT)=MN
*  Check Guide Tube Material
                  IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.1) THEN
                     DO 1990 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9300) GTAXML(COLUMN,ROW,SECT)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9301)
                           WRITE(200,7000)
                           WRITE(200,7001)
                           WRITE(200,7002)
                           WRITE(200,9302)
                           WRITE(200,7003)
                           WRITE(200,7004)
                           WRITE(200,7005)
                           WRITE(200,9303)
                           WRITE(200,9304)
                        ENDIF
 1990                CONTINUE
                  ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT)
     c            .EQ.2) THEN
                     DO 2000 C=1,2
                        IF (C.EQ.1) THEN
```

```
                  WRITE(200,9305) GTAXML(COLUMN,ROW,SECT)
               ELSEIF (C.EQ.2) THEN
                  WRITE(200,9306)
                  WRITE(200,9307)
                  WRITE(200,9308)
                  WRITE(200,9309)
                  WRITE(200,9310)
                  WRITE(200,7006)
                  WRITE(200,7007)
                  WRITE(200,7008)
                  WRITE(200,9311)
                  WRITE(200,9312)
                  WRITE(200,7009)
                  WRITE(200,7010)
                  WRITE(200,7011)
                  WRITE(200,9313)
                  WRITE(200,7012)
                  WRITE(200,7013)
                  WRITE(200,7014)
                  WRITE(200,7015)
               ENDIF
2000        CONTINUE
         ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT)
     c   .EQ.3) THEN
            DO 2010 C=1,2
               IF (C.EQ.1) THEN
                  WRITE(200,9314) GTAXML(COLUMN,ROW,SECT)
               ELSEIF (C.EQ.2) THEN
                  WRITE(200,9315)
                  WRITE(200,9316)
                  WRITE(200,9317)
                  WRITE(200,9318)
                  WRITE(200,7016)
                  WRITE(200,7017)
                  WRITE(200,7018)
                  WRITE(200,9319)
                  WRITE(200,9320)
                  WRITE(200,7019)
                  WRITE(200,7020)
                  WRITE(200,7021)
                  WRITE(200,9321)
                  WRITE(200,7022)
                  WRITE(200,7023)
                  WRITE(200,7024)
                  WRITE(200,7025)
                  WRITE(200,9322)
                  WRITE(200,9323)
                  WRITE(200,9324)
                  WRITE(200,9325)
                  WRITE(200,9326)
                  WRITE(200,9327)
                  WRITE(200,7026)
                  WRITE(200,9328)
                  WRITE(200,9329)
```

```
                              WRITE(200,9330)
                           ENDIF
 2010                   CONTINUE
                     ENDIF
                  MN=MN+1
               ENDIF
               IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.1) THEN
                  CLADRHO=6.56
               ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.2) THEN
                  CLADRHO=7.90
               ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.3) THEN
                  CLADRHO=8.19
               ENDIF
               WRITE(30,2020) LN, GTAXML(COLUMN,ROW,SECT), (-1*CLADRHO),
     c         GTSECTIRSURF(SECT),
     c         (-1*GTSECTORSURF(SECT)), (-1*GTSECTTOPSURF(SECT)),
     c         GTSECTBOTSURF(SECT), BPRAUNIV(COLUMN,ROW)
 2020          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I4,'   $ Guide tube')
               LN=LN+1
 2030       CONTINUE
*     Loop through the spacer and moderator regions along the axial
*     length of the GT (from top to bottom).
            SPACHEIGHT=0.0
            DO 2040 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
               SPACHEIGHT=SPACHEIGHT+SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
 2040       CONTINUE
            DO 2230 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
*     Define the homogenized spacer region bounding surfaces.
               IF (SPN.EQ.1) THEN
                  SPACERTOPSURF=UEFBOTTOMSURF
                  CURRENTSURF=SURFVALUESPEC(UEFBOTTOMSURF)-
     c            SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                  CURRENTSURFLABEL=0
                  DO 2050 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                           CURRENTSURFLABEL=V
                           EXIT
                        ENDIF
                     ENDIF
 2050             CONTINUE
                  IF (CURRENTSURFLABEL.EQ.0) THEN
                     SPACERBOTTOMSURF=SN
                     SURFTYPESPEC(SN)='PZ'
                     SURFVALUESPEC(SN)=CURRENTSURF
                     SN=SN+1
                  ELSE
                     SPACERBOTTOMSURF=CURRENTSURFLABEL
                  ENDIF
                  WATERREGIONTOPSURF=SPACERBOTTOMSURF
                  CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
                  CURRENTSURFLABEL=0
                  DO 2060 V=1,(SN-1)
```

```
            IF (SURFTYPESPEC(V).EQ.'PZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
2060     CONTINUE
         IF (CURRENTSURFLABEL.EQ.0) THEN
            WATERREGIONBOTTOMSURF=SN
            SURFTYPESPEC(SN)='PZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
         ELSE
            WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
         ENDIF
      ELSEIF ((SPN.NE.1).AND.(SPN.NE.
     c NUMOFSPACERS(DESNUM(COLUMN,ROW)))) THEN
         SPACERTOPSURF=WATERREGIONBOTTOMSURF
         CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c   SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
         CURRENTSURFLABEL=0
         DO 2070 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'PZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
2070     CONTINUE
         IF (CURRENTSURFLABEL.EQ.0) THEN
            SPACERBOTTOMSURF=SN
            SURFTYPESPEC(SN)='PZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
         ELSE
            SPACERBOTTOMSURF=CURRENTSURFLABEL
         ENDIF
         WATERREGIONTOPSURF=SPACERBOTTOMSURF
         CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
         CURRENTSURFLABEL=0
         DO 2080 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'PZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
2080     CONTINUE
         IF (CURRENTSURFLABEL.EQ.0) THEN
            WATERREGIONBOTTOMSURF=SN
            SURFTYPESPEC(SN)='PZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
         ELSE
```

```
                WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
             ENDIF
          ELSEIF (SPN.EQ.NUMOFSPACERS(DESNUM(COLUMN,ROW))) THEN
             SPACERTOPSURF=WATERREGIONBOTTOMSURF
             CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c       SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
             CURRENTSURFLABEL=0
             DO 2090 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
 2090        CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                SPACERBOTTOMSURF=SN
                SURFTYPESPEC(SN)='PZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                SPACERBOTTOMSURF=CURRENTSURFLABEL
             ENDIF
             WATERREGIONTOPSURF=SPACERBOTTOMSURF
             WATERREGIONBOTTOMSURF=NODEBOTTOMSURF
          ENDIF
*    Write the current homogenized spacer region cell in this GT universe.
        DO 2220 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
          IF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c    SURFVALUESPEC(SPACERTOPSURF)).AND.
     c    (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c    SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
          WRITE(30,2100) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c    (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c    GTSECTORSURF(SECT),
     c    (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c    SPN
 2100     FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c    ' IMP:N=1 U=',I4,
     c    '   $ Homogenized region for spacer ',I2)
          LN=LN+1
          ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c    SURFVALUESPEC(SPACERTOPSURF)).AND.
     c    (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c    SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
          WRITE(30,2110) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c    (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c    GTSECTORSURF(SECT),
     c    (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c    SPN
 2110     FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c    ' IMP:N=1 U=',I4,
     c    '   $ Homogenized region for spacer ',I2)
          LN=LN+1
```

```
         ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c   SURFVALUESPEC(SPACERTOPSURF)).AND.
     c   (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c   SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
         WRITE(30,2120) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c   (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c   GTSECTORSURF(SECT),
     c   (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c   SPN
2120     FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c   ' IMP:N=1 U=',I4,
     c   '    $ Homogenized region for spacer ',I2)
         LN=LN+1
         ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c   SURFVALUESPEC(SPACERTOPSURF)).AND.
     c   (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c   SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
         WRITE(30,2130) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c   (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c   GTSECTORSURF(SECT),
     c   (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c   SPN
2130     FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c   ' IMP:N=1 U=',I4,
     c   '    $ Homogenized region for spacer ',I2)
         LN=LN+1
         ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c   SURFVALUESPEC(SPACERTOPSURF)).AND.
     c   (SURFVALUESPEC(GTSECTBOTSURF(SECT)).GT.
     c   SURFVALUESPEC(SPACERBOTTOMSURF)).AND.
     c   (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c   SURFVALUESPEC(SPACERTOPSURF))) THEN
         WRITE(30,2140) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c   (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c   GTSECTORSURF(SECT),
     c   (-1*SPACERTOPSURF), GTSECTBOTSURF(SECT),
     c   BPRAUNIV(COLUMN,ROW), SPN
2140     FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c   ' IMP:N=1 U=',I4,
     c   '    $ Homogenized region for spacer ',I2)
         LN=LN+1
         ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).LT.
     c   SURFVALUESPEC(SPACERTOPSURF)).AND.
     c   (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c   SURFVALUESPEC(SPACERBOTTOMSURF)).AND.
     c   (SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c   SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
         WRITE(30,2150) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c   (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c   GTSECTORSURF(SECT),
     c   (-1*GTSECTTOPSURF(SECT)), SPACERBOTTOMSURF,
     c   BPRAUNIV(COLUMN,ROW), SPN
2150     FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c   ' IMP:N=1 U=',I4,
```

```
     c        '   $ Homogenized region for spacer ',I2)
              LN=LN+1
               ENDIF
*     Write the water region cell below the current homogenized spacer cell
in this GT universe.
               IF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,2160) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        BPRAUNIV(COLUMN,ROW)
 2160         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,2170) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        BPRAUNIV(COLUMN,ROW)
 2170         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,2180) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        BPRAUNIV(COLUMN,ROW)
 2180         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,2190) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        BPRAUNIV(COLUMN,ROW)
 2190         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).GT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
```

```
     c        SURFVALUESPEC(WATERREGIONTOPSURF))) THEN
              WRITE(30,2200) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), GTSECTBOTSURF(SECT),
     c        BPRAUNIV(COLUMN,ROW)
 2200         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).LT.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,2210) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*GTSECTTOPSURF(SECT)), WATERREGIONBOTTOMSURF,
     c        BPRAUNIV(COLUMN,ROW)
 2210         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
               ENDIF
 2220         CONTINUE
 2230      CONTINUE
*  Write the moderator inside of the GT in the BPR universe
           DO 2280 GTSECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
              IF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c        SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LE.
     c        SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
*     Write the moderator cells within the GT in this BPR universe.
                 WRITE(30,2240) LN, BMODML, (-1*MODDENSITY),
     c           (-1*GTSECTIRSURF(GTSECT)),
     c           BPOCORSURF, (-1*BPCLADTOPSURF),
     c           BPCLADBOTTOMSURF,
     c           BPRAUNIV(COLUMN,ROW)
 2240            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,
     c           '    $ Borated moderator inside guide tube')
                 LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c        SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LT.
     c        SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).GT.
     c        SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
                 WRITE(30,2250) LN, BMODML, (-1*MODDENSITY),
     c           (-1*GTSECTIRSURF(GTSECT)),
     c           BPOCORSURF, (-1*BPCLADTOPSURF),
     c           GTSECTBOTSURF(GTSECT),
     c           BPRAUNIV(COLUMN,ROW)
 2250            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,
     c           '    $ Borated moderator inside guide tube')
```

```
            LN=LN+1
          ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).LT.
     c    SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c    (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LE.
     c    SURFVALUESPEC(BPCLADBOTTOMSURF)).AND.
     c    (SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GT.
     c    SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
            WRITE(30,2260) LN, BMODML, (-1*MODDENSITY),
     c      (-1*GTSECTIRSURF(GTSECT)),
     c      BPOCORSURF, (-1*GTSECTTOPSURF(GTSECT)),
     c      BPCLADBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW)
 2260       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,
     c      '    $ Borated moderator inside guide tube')
            LN=LN+1
          ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).LT.
     c    SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c    (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).GT.
     c    SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
            WRITE(30,2270) LN, BMODML, (-1*MODDENSITY),
     c      (-1*GTSECTIRSURF(GTSECT)),
     c      BPOCORSURF, (-1*GTSECTTOPSURF(GTSECT)),
     c      GTSECTBOTSURF(GTSECT),
     c      BPRAUNIV(COLUMN,ROW)
 2270       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,
     c      '    $ Borated moderator inside guide tube')
            LN=LN+1
          ENDIF
 2280   CONTINUE
*   Determine the axial GT section which contains the lowest BPR axial
section
      DO 2290 GTSECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
        IF ((SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LT.
     c  SURFVALUESPEC(BPCLADBOTTOMSURF)).AND.
     c  (SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c  SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
          BGT=GTSECT
          EXIT
        ENDIF
 2290 CONTINUE
      DO 2320 GTSECT=BGT,NUMOFGTAXS(DESNUM(COLUMN,ROW))
        IF (GTSECT.EQ.BGT) THEN
          WRITE(30,2300) LN, BMODML, (-1*MODDENSITY),
     c    (-1*GTSECTIRSURF(GTSECT)),
     c    (-1*BPCLADBOTTOMSURF),
     c    GTSECTBOTSURF(GTSECT),
     c    BPRAUNIV(COLUMN,ROW)
 2300     FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c    ' IMP:N=1 U=',I3,
     c    '    $ Borated moderator inside guide tube')
          LN=LN+1
        ELSE
```

```
            WRITE(30,2310) LN, BMODML, (-1*MODDENSITY),
     c      (-1*GTSECTIRSURF(GTSECT)),
     c      (-1*GTSECTTOPSURF(GTSECT)),
     c      GTSECTBOTSURF(GTSECT),
     c      BPRAUNIV(COLUMN,ROW)
 2310       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,
     c      '    $ Borated moderator inside guide tube')
            LN=LN+1
         ENDIF
 2320 CONTINUE
*     Define the lower end-fitting top surface.
               CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 2330 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 2330          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPLEFTOPSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPLEFTOPSURF=CURRENTSURFLABEL
               ENDIF
*     Write the lower end-fitting cell specification for this BPR universe.
            GTBOTSURF=GTSECTBOTSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))
            IF (SURFVALUESPEC(GTBOTSURF).GE.
     c      ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)) THEN
               WRITE(30,2340) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*BPLEFTOPSURF),
     c         BPRAUNIV(COLUMN,ROW)
 2340          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,' IMP:N=1 U=',I3,
     c         '   $ Lower end-fitting')
               LN=LN+1
            ELSE
               WRITE(30,2350) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*BPLEFTOPSURF),
     c         GTSECTORSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW))),
     c         BPRAUNIV(COLUMN,ROW)
 2350          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'    $ Lower end-fitting')
               LN=LN+1
               WRITE(30,2360) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)),
     c         (-1*GTSECTBOTSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))),
     c         (-1*GTSECTORSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))),
     c         BPRAUNIV(COLUMN,ROW)
 2360          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
```

```
     c          ' IMP:N=1 U=',I3,'    $ Lower end-fitting')
                LN=LN+1
             ENDIF
*     Write the upper end-fitting cell specification for this BPR universe.
             IF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(1)).LE.
     c       SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                WRITE(30,2370) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), BPRAUNIV(COLUMN,ROW)
 2370           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(1)).LE.
     c       SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                WRITE(30,2380) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), BPOCORSURF, BPRAUNIV(COLUMN,ROW)
 2380           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,2390) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), BPCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*BPOCORSURF),
     c          BPRAUNIV(COLUMN,ROW)
 2390           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c       SURFVALUESPEC(GTSECTTOPSURF(1))).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(1)).LT.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(1)).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                WRITE(30,2400) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), BPRAUNIV(COLUMN,ROW)
 2400           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,2410) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTSECTTOPSURF(1),
     c          (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c          BPRAUNIV(COLUMN,ROW)
 2410           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,2420) LN, BMODML,
     c          (-1*MODDENSITY), BPCLADTOPSURF,
```

```
     c          (-1*GTSECTTOPSURF(1)), (-1*GTSECTIRSURF(1)),
     c          BPRAUNIV(COLUMN,ROW)
 2420           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c       SURFVALUESPEC(GTSECTTOPSURF(1))).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(1)).EQ.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,2430) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1),
     c          BPRAUNIV(COLUMN,ROW)
 2430           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
             IF (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c       SURFVALUESPEC(GTSECTTOPSURF(1))) THEN
                WRITE(30,2440) LN, BMODML,
     c          (-1*MODDENSITY), BPCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c          BPRAUNIV(COLUMN,ROW)
 2440           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
             ENDIF
             ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(1)).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(1)).LT.
     c       SURFVALUESPEC(BPCLADTOPSURF))) THEN
                WRITE(30,2450) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1),
     c          BPRAUNIV(COLUMN,ROW)
 2450           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,2460) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)),
     c          GTSECTTOPSURF(1),
     c          (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c          BPOCORSURF, BPRAUNIV(COLUMN,ROW)
 2460           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          1X,I4,' IMP:N=1 U=',I3,
     c          '   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,2470) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), BPCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*BPOCORSURF),
     c          BPRAUNIV(COLUMN,ROW)
```

```
2470         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
    c        ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
             LN=LN+1
          ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).EQ.
    c     SURFVALUESPEC(UEFTOPSURF)).AND.
    c     (SURFVALUESPEC(GTSECTTOPSURF(1)).GT.
    c     SURFVALUESPEC(UEFBOTTOMSURF)).AND.
    c     (SURFVALUESPEC(GTSECTTOPSURF(1)).LT.
    c     SURFVALUESPEC(BPCLADTOPSURF))) THEN
             WRITE(30,2480) LN, FRUEFML(COLUMN,ROW),
    c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
    c        (-1*UEFTOPSURF), GTSECTORSURF(1),
    c        BPRAUNIV(COLUMN,ROW)
2480         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
    c        ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
             LN=LN+1
             WRITE(30,2490) LN, FRUEFML(COLUMN,ROW),
    c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)),
    c        GTSECTTOPSURF(1),
    c        (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
    c        BPOCORSURF, BPRAUNIV(COLUMN,ROW)
2490         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
    c        1X,I4,' IMP:N=1 U=',I3,
    c        '   $ Assembly upper end-fitting')
             LN=LN+1
          ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
    c     SURFVALUESPEC(UEFBOTTOMSURF)).AND.
    c     (SURFVALUESPEC(BPCLADTOPSURF).LT.
    c     SURFVALUESPEC(UEFTOPSURF)).AND.
    c     (SURFVALUESPEC(GTSECTTOPSURF(1)).EQ.
    c     SURFVALUESPEC(BPCLADTOPSURF))) THEN
             WRITE(30,2500) LN, FRUEFML(COLUMN,ROW),
    c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
    c        (-1*UEFTOPSURF), GTSECTORSURF(1),
    c        BPRAUNIV(COLUMN,ROW)
2500         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
    c        ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
             LN=LN+1
             WRITE(30,2510) LN, FRUEFML(COLUMN,ROW),
    c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)),
    c        GTSECTTOPSURF(1),
    c        (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
    c        BPOCORSURF, BPRAUNIV(COLUMN,ROW)
2510         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
    c        1X,I4,' IMP:N=1 U=',I3,
    c        '   $ Assembly upper end-fitting')
             LN=LN+1
          ENDIF
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
```

```
********************************************************************
********************************************************************
********************************************************************
********************************************************************
********************************************************************
********************************************************************
********************************************************************
********************************************************************
********************************************************************
********************************************************************
********************************************************************
********************************************************************
********************************************************************
********************************************************************
********************************************************************
         ELSEIF (WBPRA(BANKNUM(COLUMN,ROW)).EQ.3) THEN
         DO 2680 MCNPNODE=1,NUMOFBPRANODES(BANKNUM(COLUMN,ROW))
*     Define the upper end-fitting bottom surface.
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 2520 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
2520           CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  UEFBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  UEFBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
*     Define the upper end-fitting top surface.
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),1)+
     c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 2530 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
2530           CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  UEFTOPSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
```

```
            SN=SN+1
         ELSE
            UEFTOPSURF=CURRENTSURFLABEL
         ENDIF
*     Define the inner BPR cladding inner radius.
         CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),1)
         CURRENTSURFLABEL=0
         DO 2540 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'CZ') THEN
       IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
 2540    CONTINUE
         IF (CURRENTSURFLABEL.EQ.0) THEN
            BPICIRSURF=SN
            SURFTYPESPEC(SN)='CZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
         ELSE
            BPICIRSURF=CURRENTSURFLABEL
         ENDIF
*     Define the inner BPR cladding outer radius.
         CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),2)
         CURRENTSURFLABEL=0
         DO 2550 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'CZ') THEN
       IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
 2550    CONTINUE
         IF (CURRENTSURFLABEL.EQ.0) THEN
            BPICORSURF=SN
            SURFTYPESPEC(SN)='CZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
         ELSE
            BPICORSURF=CURRENTSURFLABEL
         ENDIF
*     Define the BP absorber inner radius.
         CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),3)
         CURRENTSURFLABEL=0
         DO 2560 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'CZ') THEN
       IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
 2560    CONTINUE
         IF (CURRENTSURFLABEL.EQ.0) THEN
```

```
                 BPIRSURF=SN
                 SURFTYPESPEC(SN)='CZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 BPIRSURF=CURRENTSURFLABEL
              ENDIF
*     Define the BP absorber outer radius.
              CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),4)
              CURRENTSURFLABEL=0
              DO 2570 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 2570         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 BPORSURF=SN
                 SURFTYPESPEC(SN)='CZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 BPORSURF=CURRENTSURFLABEL
              ENDIF
*     Define the outer BPR cladding inner radius.
              CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),5)
              CURRENTSURFLABEL=0
              DO 2580 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 2580         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 BPOCIRSURF=SN
                 SURFTYPESPEC(SN)='CZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 BPOCIRSURF=CURRENTSURFLABEL
              ENDIF
*     Define the outer BPR cladding outer radius.
              CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),6)
              CURRENTSURFLABEL=0
              DO 2590 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
```

```
                 ENDIF
 2590         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 BPOCORSURF=SN
                 SURFTYPESPEC(SN)='CZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 BPOCORSURF=CURRENTSURFLABEL
              ENDIF
*     Define the BP node bounding surfaces.
           IF (MCNPNODE.EQ.1) THEN
              TOTBPHEIGHT=0.0
              DO 2600 Z=1,NUMOFBPRANODES(BANKNUM(COLUMN,ROW))
                 TOTBPHEIGHT=TOTBPHEIGHT+
     c           MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),Z)
 2600         CONTINUE
              CURRENTSURF=BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))+
     c        TOTBPHEIGHT
              IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                 CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
              ENDIF
              CURRENTSURFLABEL=0
              DO 2610 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 2610         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 TOPBPNODETOPSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 TOPBPNODETOPSURF=CURRENTSURFLABEL
              ENDIF
              BPNODETOPSURF=TOPBPNODETOPSURF
              CURRENTSURF=SURFVALUESPEC(BPNODETOPSURF)-
     c        MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),MCNPNODE)
              IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                 CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
              ENDIF
              CURRENTSURFLABEL=0
              DO 2620 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 2620         CONTINUE
```

```
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 BPNODEBOTTOMSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 BPNODEBOTTOMSURF=CURRENTSURFLABEL
              ENDIF
           ELSEIF (MCNPNODE.NE.1) THEN
              BPNODETOPSURF=BPNODEBOTTOMSURF
              CURRENTSURF=SURFVALUESPEC(BPNODETOPSURF)-
     c        MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),MCNPNODE)
              IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                 CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
              ENDIF
              CURRENTSURFLABEL=0
              DO 2640 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 2640         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 BPNODEBOTTOMSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 BPNODEBOTTOMSURF=CURRENTSURFLABEL
              ENDIF
           ENDIF
        IF (SURFVALUESPEC(BPNODEBOTTOMSURF).LT.
     c  SURFVALUESPEC(UEFTOPSURF)) THEN
*    Write the BP node cells in this BPR universe.
        IF ((BPRABSNODE(BANKNUM(COLUMN,ROW),MCNPNODE).EQ.'Y').AND.
     c      (BPNONABSMAT(BANKNUM(COLUMN,ROW)).EQ.1)) THEN
           WRITE(30,2650) LN, BPNODEML,
     c     (-1*AL2O3DENSITY(BANKNUM(COLUMN,ROW))), (-1*BPORSURF),
     c     BPIRSURF, (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c     BPRAUNIV(COLUMN,ROW), MCNPNODE
 2650      FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c     1X,I4,' IMP:N=1 U=',I3,'   $ Burnable poison node ',I2)
           LN=LN+1
           BPNODEML=BPNODEML+1
        ELSEIF ((BPRABSNODE(BANKNUM(COLUMN,ROW),MCNPNODE).EQ.'Y').AND.
     c      (BPNONABSMAT(BANKNUM(COLUMN,ROW)).NE.1)) THEN
           WRITE(30,2660) LN, BPNODEML,
     c     (-1*NONBPMATDATA(BANKNUM(COLUMN,ROW),1)), (-1*BPORSURF),
     c     BPIRSURF, (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c     BPRAUNIV(COLUMN,ROW), MCNPNODE
 2660      FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c     1X,I4,' IMP:N=1 U=',I3,'   $ Burnable poison node ',I2)
```

```
           LN=LN+1
           BPNODEML=BPNODEML+1
        ELSE
           WRITE(30,2670) LN, BPNODEML,
     c     (-1*BPDENTOGO(COLUMN,ROW,MCNPNODE)), (-1*BPORSURF),
     c     BPIRSURF, (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c     BPRAUNIV(COLUMN,ROW), MCNPNODE
 2670      FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c     1X,I4,' IMP:N=1 U=',I3,'   $ Burnable poison node ',I2)
           LN=LN+1
           BPNODEML=BPNODEML+1
        ENDIF
        ENDIF
 2680    CONTINUE
*    Define the BPR cladding top surface.
              CURRENTSURF=TOTBPHEIGHT+
     c        BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))+
     c        BPRPLEN(BANKNUM(COLUMN,ROW),1)
              IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                 CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
              ENDIF
              CURRENTSURFLABEL=0
              DO 2690 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                   ENDIF
                 ENDIF
 2690         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 BPCLADTOPSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 BPCLADTOPSURF=CURRENTSURFLABEL
              ENDIF
*    Define the BPR cladding bottom surface.
              CURRENTSURF=BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))-
     c        BPRPLEN(BANKNUM(COLUMN,ROW),2)
              CURRENTSURFLABEL=0
              DO 2700 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                   ENDIF
                 ENDIF
 2700         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 BPCLADBOTTOMSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
```

```
                 SN=SN+1
              ELSE
                 BPCLADBOTTOMSURF=CURRENTSURFLABEL
              ENDIF
*     Write the inner BP-to-cladding gap cell in this BPR universe.
           WRITE(30,2710) LN, (-1*BPIRSURF), BPICORSURF,
     c     (-1*TOPBPNODETOPSURF),
     c     BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 2710      FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c     ' IMP:N=1 U=',I3,
     c     '   $ Burnable poison-to-cladding gap')
           LN=LN+1
*     Write the outer BP-to-cladding gap cell in this BPR universe.
           WRITE(30,2720) LN, (-1*BPOCIRSURF), BPORSURF,
     c     (-1*TOPBPNODETOPSURF),
     c     BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 2720      FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c     ' IMP:N=1 U=',I3,
     c     '   $ Burnable poison-to-cladding gap')
           LN=LN+1
*     Write the annular water cell in this BPR universe.
           WRITE(30,2730) LN, BMODML,
     c     (-1*MODDENSITY), (-1*BPICIRSURF),
     c     (-1*TOPBPNODETOPSURF),
     c     BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 2730      FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c     ' IMP:N=1 U=',I3,
     c     '   $ Burnable poison-to-cladding gap')
           LN=LN+1
*     Write the BPR cladding cell in this BPR universe.
*     Determine if the BPR cladding material specification has
*     previously been defined.  If it has been previously defined, determine
*     the cladding material specification label.
           CLADMLUNIQUE=.TRUE.
           LEAVE=.FALSE.
           IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
              DO 2750 RO=1,(ROW-1)
                 DO 2740 CO=1,50
                    IF (BANKNUM(CO,RO).NE.0) THEN
                    IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                       IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                 BPRCLADMAT(BANKNUM(CO,RO))) THEN
                          CLADMLUNIQUE=.FALSE.
                          LEAVE=.TRUE.
                          BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                          EXIT
                       ENDIF
                    ENDIF
                    ENDIF
 2740            CONTINUE
                 IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                 ENDIF
 2750         CONTINUE
```

```
                IF (LEAVE.EQ..FALSE.) THEN
                   DO 2770 RO=ROW,ROW
                      DO 2760 CO=1,(COLUMN-1)
                         IF (BANKNUM(CO,RO).NE.0) THEN
                         IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                            IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                      BPRCLADMAT(BANKNUM(CO,RO))) THEN
                               CLADMLUNIQUE=.FALSE.
                               LEAVE=.TRUE.
                               BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                               EXIT
                            ENDIF
                         ENDIF
                         ENDIF
 2760                 CONTINUE
                      IF (LEAVE.EQ..TRUE.) THEN
                         EXIT
                      ENDIF
 2770              CONTINUE
                ENDIF
             ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                DO 2790 RO=1,(ROW-1)
                   DO 2780 CO=1,50
                      IF (BANKNUM(CO,RO).NE.0) THEN
                      IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                         IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                   BPRCLADMAT(BANKNUM(CO,RO))) THEN
                            CLADMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                            EXIT
                         ENDIF
                      ENDIF
                      ENDIF
 2780              CONTINUE
                   IF (LEAVE.EQ..TRUE.) THEN
                      EXIT
                   ENDIF
 2790           CONTINUE
             ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                DO 2810 RO=1,1
                   DO 2800 CO=1,(COLUMN-1)
                      IF (BANKNUM(CO,RO).NE.0) THEN
                      IF (BANKDES(BANKNUM(CO,RO)).NE.'BPRA ') THEN
                         IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                   BPRCLADMAT(BANKNUM(CO,RO))) THEN
                            CLADMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                            EXIT
                         ENDIF
                      ENDIF
                      ENDIF
 2800              CONTINUE
```

```
                IF (LEAVE.EQ..TRUE.) THEN
                   EXIT
                ENDIF
 2810         CONTINUE
            ENDIF
            IF (CLADMLUNIQUE.EQ..TRUE.) THEN
               BPCLADML(COLUMN,ROW)=MN
*  Check BPR Cladding Material
                IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
                   DO 2820 C=1,2
                      IF (C.EQ.1) THEN
                         WRITE(200,9300) BPCLADML(COLUMN,ROW)
                      ELSEIF (C.EQ.2) THEN
                         WRITE(200,9301)
                         WRITE(200,7000)
                         WRITE(200,7001)
                         WRITE(200,7002)
                         WRITE(200,9302)
                         WRITE(200,7003)
                         WRITE(200,7004)
                         WRITE(200,7005)
                         WRITE(200,9303)
                         WRITE(200,9304)
                      ENDIF
 2820              CONTINUE
                ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW))
     c          .EQ.2) THEN
                   DO 2830 C=1,2
                      IF (C.EQ.1) THEN
                         WRITE(200,9305) BPCLADML(COLUMN,ROW)
                      ELSEIF (C.EQ.2) THEN
                         WRITE(200,9306)
                         WRITE(200,9307)
                         WRITE(200,9308)
                         WRITE(200,9309)
                         WRITE(200,9310)
                         WRITE(200,7006)
                         WRITE(200,7007)
                         WRITE(200,7008)
                         WRITE(200,9311)
                         WRITE(200,9312)
                         WRITE(200,7009)
                         WRITE(200,7010)
                         WRITE(200,7011)
                         WRITE(200,9313)
                         WRITE(200,7012)
                         WRITE(200,7013)
                         WRITE(200,7014)
                         WRITE(200,7015)
                      ENDIF
 2830              CONTINUE
                ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW))
     c          .EQ.3) THEN
                   DO 2840 C=1,2
```

```
                  IF (C.EQ.1) THEN
                     WRITE(200,9314) BPCLADML(COLUMN,ROW)
                  ELSEIF (C.EQ.2) THEN
                     WRITE(200,9315)
                     WRITE(200,9316)
                     WRITE(200,9317)
                     WRITE(200,9318)
                     WRITE(200,7016)
                     WRITE(200,7017)
                     WRITE(200,7018)
                     WRITE(200,9319)
                     WRITE(200,9320)
                     WRITE(200,7019)
                     WRITE(200,7020)
                     WRITE(200,7021)
                     WRITE(200,9321)
                     WRITE(200,7022)
                     WRITE(200,7023)
                     WRITE(200,7024)
                     WRITE(200,7025)
                     WRITE(200,9322)
                     WRITE(200,9323)
                     WRITE(200,9324)
                     WRITE(200,9325)
                     WRITE(200,9326)
                     WRITE(200,9327)
                     WRITE(200,7026)
                     WRITE(200,9328)
                     WRITE(200,9329)
                     WRITE(200,9330)
                  ENDIF
2840           CONTINUE
             ENDIF
          MN=MN+1
       ENDIF
       IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
          CLADRHO=6.56
       ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.2) THEN
          CLADRHO=7.90
       ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.3) THEN
          CLADRHO=8.19
       ENDIF
       WRITE(30,2850) LN, BPCLADML(COLUMN,ROW), (-1*CLADRHO),
    c  BPOCIRSURF,
    c  (-1*BPOCORSURF), (-1*BPCLADTOPSURF), BPCLADBOTTOMSURF,
    c  BPRAUNIV(COLUMN,ROW)
2850   FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
    c  ' IMP:N=1 U=',I3,'    $ BPR cladding')
       LN=LN+1
       WRITE(30,2860) LN, BPCLADML(COLUMN,ROW), (-1*CLADRHO),
    c  BPICIRSURF,
    c  (-1*BPICORSURF), (-1*TOPBPNODETOPSURF),
    c  BPNODEBOTTOMSURF,
    c  BPRAUNIV(COLUMN,ROW)
```

```
 2860       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ BPR cladding')
            LN=LN+1
*     Write the BPR upper plenum cell in this BPR universe.
*     Determine if the BPR upper plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the upper plenum material specification label.
            BPRUPMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 2880 RO=1,(ROW-1)
                  DO 2870 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                           BPRUPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 2870             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 2880          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 2900 RO=ROW,ROW
                     DO 2890 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
                           IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                              BPRUPMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 2890                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 2900             CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 2920 RO=1,(ROW-1)
                  DO 2910 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                           BPRUPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                           EXIT
```

```
                    ENDIF
                 ENDIF
2910          CONTINUE
              IF (LEAVE.EQ..TRUE.) THEN
                 EXIT
              ENDIF
2920       CONTINUE
        ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
           DO 2940 RO=1,1
              DO 2930 CO=1,(COLUMN-1)
                 IF (BANKNUM(CO,RO).NE.0) THEN
                    IF (BANKNUM(COLUMN,ROW).EQ.
     c                 BANKNUM(CO,RO)) THEN
                       BPRUPMLUNIQUE=.FALSE.
                       LEAVE=.TRUE.
                       BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                       EXIT
                    ENDIF
                 ENDIF
2930          CONTINUE
              IF (LEAVE.EQ..TRUE.) THEN
                 EXIT
              ENDIF
2940       CONTINUE
        ENDIF
     IF (SURFVALUESPEC(TOPBPNODETOPSURF).LT.
     c SURFVALUESPEC(UEFTOPSURF)) THEN
        IF (BPRUPMLUNIQUE.EQ..TRUE.) THEN
           BPRUPML(COLUMN,ROW)=MN
* Check Burnable Poison Rod Upper Plenum Regions
              DO 2970 C=1,BPRUPLENMAT(BANKNUM(COLUMN,ROW),2)
                 IF (C.EQ.1) THEN
                    WRITE(200,2950) BPRUPML(COLUMN,ROW),
     c              BPRUPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c              (-1*BPRUPLENWTS(BANKNUM(COLUMN,ROW),C))
2950                FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c              '    $ Burnable Poison Rod Upper Plenum')
                 ELSE
                    WRITE(200,2960)
     c              BPRUPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c              (-1*BPRUPLENWTS(BANKNUM(COLUMN,ROW),C))
2960                FORMAT(T9,A9,3X,G14.6)
                 ENDIF
2970          CONTINUE
           MN=MN+1
        ENDIF
        WRITE(30,2980) LN, BPRUPML(COLUMN,ROW),
     c  (-1*BPRUPLENMAT(BANKNUM(COLUMN,ROW),1)),
     c  TOPBPNODETOPSURF,
     c  (-1*BPCLADTOPSURF), (-1*BPOCIRSURF),
     c  BPRAUNIV(COLUMN,ROW)
2980    FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c  ' IMP:N=1 U=',I3,'   $ BPR upper plenum region')
        LN=LN+1
```

```
         ENDIF
*       Write the BPR lower plenum cell (lower end plug) in this BPR universe.
*       Determine if the BPR lower plenum material specification has
*       previously been defined.  If it has been previously defined, determine
*       the lower plenum material specification label.
            BPRLPMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 3000 RO=1,(ROW-1)
                  DO 2990 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO)) THEN
                           BPRLPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 2990             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 3000          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 3020 RO=ROW,ROW
                     DO 3010 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
                           IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                              BPRLPMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 3010                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 3020             CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 3040 RO=1,(ROW-1)
                  DO 3030 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                  BANKNUM(CO,RO)) THEN
                           BPRLPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 3030             CONTINUE
```

```
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 3040          CONTINUE
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
               DO 3060 RO=1,1
                  DO 3050 CO=1,(COLUMN-1)
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                           BPRLPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 3050             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 3060          CONTINUE
            ENDIF
            IF (BPRLPMLUNIQUE.EQ..TRUE.) THEN
               BPRLPML(COLUMN,ROW)=MN
*  Check Burnable Poison Rod Lower Plenum Regions
                  DO 3090 C=1,BPRLPLENMAT(BANKNUM(COLUMN,ROW),2)
                     IF (C.EQ.1) THEN
                        WRITE(200,3070) BPRLPML(COLUMN,ROW),
     c                  BPRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                  (-1*BPRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 3070                   FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                  '    $ Burnable Poison Rod Lower Plenum')
                     ELSE
                        WRITE(200,3080)
     c                  BPRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                  (-1*BPRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 3080                   FORMAT(T9,A9,3X,G14.6)
                     ENDIF
 3090             CONTINUE
               MN=MN+1
            ENDIF
            WRITE(30,3100) LN, BPRLPML(COLUMN,ROW),
     c      (-1*BPRLPLENMAT(BANKNUM(COLUMN,ROW),1)), BPCLADBOTTOMSURF,
     c      (-1*BPNODEBOTTOMSURF), (-1*BPOCIRSURF),
     c      BPRAUNIV(COLUMN,ROW)
 3100       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'    $ BPR lower plenum region')
            LN=LN+1
*     Loop through the regions above the BPR (i.e. the appropriate upper core
regions)
*     Define the upper region lower surface.
               DO 3140 REGION=1,NUMREGABOVEBPRA
*     Determine the current upper region's lower surface specification.
                  IF (REGION.EQ.1) THEN
```

```
                  REGIONTOPSURF=SYSTEMTOP
                  CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
     c            REGABOVEBPRA(REGION,1)
               ENDIF
               CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
     c         REGABOVEBPRA(REGION,1)
               IF (REGION.EQ.NUMREGABOVEBPRA) THEN
                  REGIONBOTTOMSURF=UEFTOPSURF
               ELSE
               CURRENTSURFLABEL=0
               DO 3110 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 3110          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  REGIONBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  REGIONBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
               ENDIF
*     Write the cell specification for the BPR universe upper region.
            IF (REGION.EQ.1) THEN
               WRITE(30,3120) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c         (-1*REGABOVEBPRA(REGION,2)),
     c         REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 3120          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
     c         ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
               LN=LN+1
               REGIONTOPSURF=REGIONBOTTOMSURF
            ELSE
               WRITE(30,3130) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c         (-1*REGABOVEBPRA(REGION,2)), (-1*REGIONTOPSURF),
     c         REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 3130          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
               LN=LN+1
               REGIONTOPSURF=REGIONBOTTOMSURF
            ENDIF
 3140       CONTINUE
      DO 3190 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
*     Define the GT section top surface.
            CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),3,SECT)
            IF (CURRENTSURF.GT.SURFVALUESPEC(UEFTOPSURF)) THEN
               CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
            ENDIF
            CURRENTSURFLABEL=0
            DO 3150 V=1,(SN-1)
```

```
            IF (SURFTYPESPEC(V).EQ.'PZ') THEN
        IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
3150      CONTINUE
          IF (CURRENTSURFLABEL.EQ.0) THEN
             GTSECTTOPSURF(SECT)=SN
             SURFTYPESPEC(SN)='PZ'
             SURFVALUESPEC(SN)=CURRENTSURF
             SN=SN+1
          ELSE
             GTSECTTOPSURF(SECT)=CURRENTSURFLABEL
          ENDIF
*     Define the GT section bottom surface.
          CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),4,SECT)
          CURRENTSURFLABEL=0
          DO 3160 V=1,(SN-1)
             IF (SURFTYPESPEC(V).EQ.'PZ') THEN
        IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
3160      CONTINUE
          IF (CURRENTSURFLABEL.EQ.0) THEN
             GTSECTBOTSURF(SECT)=SN
             SURFTYPESPEC(SN)='PZ'
             SURFVALUESPEC(SN)=CURRENTSURF
             SN=SN+1
          ELSE
             GTSECTBOTSURF(SECT)=CURRENTSURFLABEL
          ENDIF
*     Define the GT section outer radius surface.
          CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),2,SECT)
          CURRENTSURFLABEL=0
          DO 3170 V=1,(SN-1)
             IF (SURFTYPESPEC(V).EQ.'CZ') THEN
        IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
3170      CONTINUE
          IF (CURRENTSURFLABEL.EQ.0) THEN
             GTSECTORSURF(SECT)=SN
             SURFTYPESPEC(SN)='CZ'
             SURFVALUESPEC(SN)=CURRENTSURF
             SN=SN+1
          ELSE
             GTSECTORSURF(SECT)=CURRENTSURFLABEL
          ENDIF
*     Define the GT section inner radius surface.
```

```
                 CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),1,SECT)
                 CURRENTSURFLABEL=0
                 DO 3180 V=1,(SN-1)
                    IF (SURFTYPESPEC(V).EQ.'CZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                          CURRENTSURFLABEL=V
                          EXIT
                       ENDIF
                    ENDIF
 3180            CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
                    GTSECTIRSURF(SECT)=SN
                    SURFTYPESPEC(SN)='CZ'
                    SURFVALUESPEC(SN)=CURRENTSURF
                    SN=SN+1
                 ELSE
                    GTSECTIRSURF(SECT)=CURRENTSURFLABEL
                 ENDIF
 3190 CONTINUE
*     Write the GT material cell
          DO 3320 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
*     Determine if the GT material specification has
*     previously been defined.  If it has been previously defined, determine
*     the material specification label.
            CLADMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 3210 RO=1,(ROW-1)
                  DO 3200 CO=1,50
                     IF ((DESNUM(CO,RO).NE.0).AND.
     c               (BANKNUM(CO,RO).EQ.0)) THEN
                        IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                  GTMAT(DESNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                           EXIT
                        ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                  GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                           EXIT
                        ENDIF
                     ENDIF
 3200             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 3210          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 3230 RO=ROW,ROW
                     DO 3220 CO=1,(COLUMN-1)
                        IF ((DESNUM(CO,RO).NE.0).AND.
```

```
     c                    (BANKNUM(CO,RO).EQ.0)) THEN
                             IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                       GTMAT(DESNUM(CO,RO))) THEN
                                CLADMLUNIQUE=.FALSE.
                                LEAVE=.TRUE.
                                GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                                EXIT
                             ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                       GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                                CLADMLUNIQUE=.FALSE.
                                LEAVE=.TRUE.
                                GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                                EXIT
                             ENDIF
                          ENDIF
 3220                  CONTINUE
                       IF (LEAVE.EQ..TRUE.) THEN
                          EXIT
                       ENDIF
 3230               CONTINUE
                 ENDIF
              ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                 DO 3250 RO=1,(ROW-1)
                    DO 3240 CO=1,50
                       IF ((DESNUM(CO,RO).NE.0).AND.
     c                    (BANKNUM(CO,RO).EQ.0)) THEN
                          IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                    GTMAT(DESNUM(CO,RO))) THEN
                             CLADMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                             EXIT
                          ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                    GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                             CLADMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                             EXIT
                          ENDIF
                       ENDIF
 3240               CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
 3250            CONTINUE
              ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                 DO 3270 RO=1,1
                    DO 3260 CO=1,(COLUMN-1)
                       IF ((DESNUM(CO,RO).NE.0).AND.
     c                    (BANKNUM(CO,RO).EQ.0)) THEN
                          IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                    GTMAT(DESNUM(CO,RO))) THEN
                             CLADMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
```

```
                        GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                        EXIT
                     ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c               GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                        CLADMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                        EXIT
                     ENDIF
                  ENDIF
 3260          CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
 3270       CONTINUE
         ENDIF
         IF (CLADMLUNIQUE.EQ..TRUE.) THEN
            GTAXML(COLUMN,ROW,SECT)=MN
*  Check Guide Tube Material
               IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.1) THEN
                  DO 3280 C=1,2
                     IF (C.EQ.1) THEN
                        WRITE(200,9300) GTAXML(COLUMN,ROW,SECT)
                     ELSEIF (C.EQ.2) THEN
                        WRITE(200,9301)
                        WRITE(200,7000)
                        WRITE(200,7001)
                        WRITE(200,7002)
                        WRITE(200,9302)
                        WRITE(200,7003)
                        WRITE(200,7004)
                        WRITE(200,7005)
                        WRITE(200,9303)
                        WRITE(200,9304)
                     ENDIF
 3280             CONTINUE
               ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT)
     c         .EQ.2) THEN
                  DO 3290 C=1,2
                     IF (C.EQ.1) THEN
                        WRITE(200,9305) GTAXML(COLUMN,ROW,SECT)
                     ELSEIF (C.EQ.2) THEN
                        WRITE(200,9306)
                        WRITE(200,9307)
                        WRITE(200,9308)
                        WRITE(200,9309)
                        WRITE(200,9310)
                        WRITE(200,7006)
                        WRITE(200,7007)
                        WRITE(200,7008)
                        WRITE(200,9311)
                        WRITE(200,9312)
                        WRITE(200,7009)
                        WRITE(200,7010)
```

```
                        WRITE(200,7011)
                        WRITE(200,9313)
                        WRITE(200,7012)
                        WRITE(200,7013)
                        WRITE(200,7014)
                        WRITE(200,7015)
                     ENDIF
3290              CONTINUE
               ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT)
     c         .EQ.3) THEN
                  DO 3300 C=1,2
                     IF (C.EQ.1) THEN
                        WRITE(200,9314) GTAXML(COLUMN,ROW,SECT)
                     ELSEIF (C.EQ.2) THEN
                        WRITE(200,9315)
                        WRITE(200,9316)
                        WRITE(200,9317)
                        WRITE(200,9318)
                        WRITE(200,7016)
                        WRITE(200,7017)
                        WRITE(200,7018)
                        WRITE(200,9319)
                        WRITE(200,9320)
                        WRITE(200,7019)
                        WRITE(200,7020)
                        WRITE(200,7021)
                        WRITE(200,9321)
                        WRITE(200,7022)
                        WRITE(200,7023)
                        WRITE(200,7024)
                        WRITE(200,7025)
                        WRITE(200,9322)
                        WRITE(200,9323)
                        WRITE(200,9324)
                        WRITE(200,9325)
                        WRITE(200,9326)
                        WRITE(200,9327)
                        WRITE(200,7026)
                        WRITE(200,9328)
                        WRITE(200,9329)
                        WRITE(200,9330)
                     ENDIF
3300              CONTINUE
               ENDIF
            MN=MN+1
         ENDIF
         IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.1) THEN
            CLADRHO=6.56
         ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.2) THEN
            CLADRHO=7.90
         ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.3) THEN
            CLADRHO=8.19
         ENDIF
         WRITE(30,3310) LN, GTAXML(COLUMN,ROW,SECT), (-1*CLADRHO),
```

```
     c       GTSECTIRSURF(SECT),
     c       (-1*GTSECTORSURF(SECT)), (-1*GTSECTTOPSURF(SECT)),
     c       GTSECTBOTSURF(SECT), BPRAUNIV(COLUMN,ROW)
 3310        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'   $ Guide tube')
             LN=LN+1
 3320     CONTINUE
*     Loop through the spacer and moderator regions along the axial
*     length of the GT (from top to bottom).
          SPACHEIGHT=0.0
          DO 3330 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
             SPACHEIGHT=SPACHEIGHT+SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
 3330     CONTINUE
          DO 3520 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
*     Define the homogenized spacer region bounding surfaces.
            IF (SPN.EQ.1) THEN
               SPACERTOPSURF=UEFBOTTOMSURF
               CURRENTSURF=SURFVALUESPEC(UEFBOTTOMSURF)-
     c         SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
               CURRENTSURFLABEL=0
               DO 3340 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                    ENDIF
                  ENDIF
 3340          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  SPACERBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  SPACERBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
               WATERREGIONTOPSURF=SPACERBOTTOMSURF
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
               CURRENTSURFLABEL=0
               DO 3350 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                    ENDIF
                  ENDIF
 3350          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  WATERREGIONBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
```

```
                 ENDIF
              ELSEIF ((SPN.NE.1).AND.(SPN.NE.
     c        NUMOFSPACERS(DESNUM(COLUMN,ROW)))) THEN
                 SPACERTOPSURF=WATERREGIONBOTTOMSURF
                 CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c           SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                 CURRENTSURFLABEL=0
                 DO 3360 V=1,(SN-1)
                    IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                          CURRENTSURFLABEL=V
                          EXIT
                       ENDIF
                    ENDIF
 3360            CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
                    SPACERBOTTOMSURF=SN
                    SURFTYPESPEC(SN)='PZ'
                    SURFVALUESPEC(SN)=CURRENTSURF
                    SN=SN+1
                 ELSE
                    SPACERBOTTOMSURF=CURRENTSURFLABEL
                 ENDIF
                 WATERREGIONTOPSURF=SPACERBOTTOMSURF
                 CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
                 CURRENTSURFLABEL=0
                 DO 3370 V=1,(SN-1)
                    IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                          CURRENTSURFLABEL=V
                          EXIT
                       ENDIF
                    ENDIF
 3370            CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
                    WATERREGIONBOTTOMSURF=SN
                    SURFTYPESPEC(SN)='PZ'
                    SURFVALUESPEC(SN)=CURRENTSURF
                    SN=SN+1
                 ELSE
                    WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
                 ENDIF
              ELSEIF (SPN.EQ.NUMOFSPACERS(DESNUM(COLUMN,ROW))) THEN
                 SPACERTOPSURF=WATERREGIONBOTTOMSURF
                 CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c           SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                 CURRENTSURFLABEL=0
                 DO 3380 V=1,(SN-1)
                    IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                          CURRENTSURFLABEL=V
                          EXIT
                       ENDIF
                    ENDIF
```

```
 3380           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   SPACERBOTTOMSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   SPACERBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
                WATERREGIONTOPSURF=SPACERBOTTOMSURF
                WATERREGIONBOTTOMSURF=NODEBOTTOMSURF
             ENDIF
*     Write the current homogenized spacer region cell in this GT universe.
           DO 3510 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
              IF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(SPACERTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
             WRITE(30,3390) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c       (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c       GTSECTORSURF(SECT),
     c       (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c       SPN
 3390        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,
     c       '   $ Homogenized region for spacer ',I2)
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c       SURFVALUESPEC(SPACERTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
             WRITE(30,3400) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c       (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c       GTSECTORSURF(SECT),
     c       (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c       SPN
 3400        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,
     c       '   $ Homogenized region for spacer ',I2)
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c       SURFVALUESPEC(SPACERTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c       SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
             WRITE(30,3410) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c       (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c       GTSECTORSURF(SECT),
     c       (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c       SPN
 3410        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,
     c       '   $ Homogenized region for spacer ',I2)
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
```

```
     c        SURFVALUESPEC(SPACERTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c        SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
              WRITE(30,3420) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c        (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c        GTSECTORSURF(SECT),
     c        (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c        SPN
 3420         FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,
     c        '    $ Homogenized region for spacer ',I2)
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(SPACERTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).GT.
     c        SURFVALUESPEC(SPACERBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(SPACERTOPSURF))) THEN
              WRITE(30,3430) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c        (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c        GTSECTORSURF(SECT),
     c        (-1*SPACERTOPSURF), GTSECTBOTSURF(SECT),
     c        BPRAUNIV(COLUMN,ROW), SPN
 3430         FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,
     c        '    $ Homogenized region for spacer ',I2)
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).LT.
     c        SURFVALUESPEC(SPACERTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(SPACERBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
              WRITE(30,3440) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c        (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c        GTSECTORSURF(SECT),
     c        (-1*GTSECTTOPSURF(SECT)), SPACERBOTTOMSURF,
     c        BPRAUNIV(COLUMN,ROW), SPN
 3440         FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,
     c        '    $ Homogenized region for spacer ',I2)
              LN=LN+1
               ENDIF
*    Write the water region cell below the current homogenized spacer cell
in this GT universe.
               IF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,3450) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        BPRAUNIV(COLUMN,ROW)
 3450         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
```

```
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,3460) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        BPRAUNIV(COLUMN,ROW)
 3460         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,3470) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        BPRAUNIV(COLUMN,ROW)
 3470         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
              WRITE(30,3480) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c        BPRAUNIV(COLUMN,ROW)
 3480         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).GT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(WATERREGIONTOPSURF))) THEN
              WRITE(30,3490) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*WATERREGIONTOPSURF), GTSECTBOTSURF(SECT),
     c        BPRAUNIV(COLUMN,ROW)
 3490         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'    $ Borated moderator region')
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).LT.
     c        SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
```

```
              WRITE(30,3500) LN, BMODML, (-1*MODDENSITY),
     c        GTSECTORSURF(SECT),
     c        (-1*GTSECTTOPSURF(SECT)), WATERREGIONBOTTOMSURF,
     c        BPRAUNIV(COLUMN,ROW)
 3500         FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'   $ Borated moderator region')
              LN=LN+1
               ENDIF
 3510         CONTINUE
 3520     CONTINUE
*  Write the moderator inside of the GT in the BPR universe
          DO 3570 GTSECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
              IF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c        SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LE.
     c        SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
*     Write the moderator cells within the GT in this BPR universe.
                 WRITE(30,3530) LN, BMODML, (-1*MODDENSITY),
     c           (-1*GTSECTIRSURF(GTSECT)),
     c           BPOCORSURF, (-1*BPCLADTOPSURF),
     c           BPCLADBOTTOMSURF,
     c           BPRAUNIV(COLUMN,ROW)
 3530            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,
     c           '    $ Borated moderator inside guide tube')
                 LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c        SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LT.
     c        SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).GT.
     c        SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
                 WRITE(30,3540) LN, BMODML, (-1*MODDENSITY),
     c           (-1*GTSECTIRSURF(GTSECT)),
     c           BPOCORSURF, (-1*BPCLADTOPSURF),
     c           GTSECTBOTSURF(GTSECT),
     c           BPRAUNIV(COLUMN,ROW)
 3540            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,
     c           '    $ Borated moderator inside guide tube')
                 LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).LT.
     c        SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LE.
     c        SURFVALUESPEC(BPCLADBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GT.
     c        SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
                 WRITE(30,3550) LN, BMODML, (-1*MODDENSITY),
     c           (-1*GTSECTIRSURF(GTSECT)),
     c           BPOCORSURF, (-1*GTSECTTOPSURF(GTSECT)),
     c           BPCLADBOTTOMSURF,
     c           BPRAUNIV(COLUMN,ROW)
 3550            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,
```

```
     c          '   $ Borated moderator inside guide tube')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).LT.
     c       SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).GT.
     c       SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
                WRITE(30,3560) LN, BMODML, (-1*MODDENSITY),
     c          (-1*GTSECTIRSURF(GTSECT)),
     c          BPOCORSURF, (-1*GTSECTTOPSURF(GTSECT)),
     c          GTSECTBOTSURF(GTSECT),
     c          BPRAUNIV(COLUMN,ROW)
 3560           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '   $ Borated moderator inside guide tube')
                LN=LN+1
             ENDIF
 3570     CONTINUE
*   Determine the axial GT section which contains the lowest BPR axial
section
        DO 3580 GTSECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
          IF ((SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LT.
     c    SURFVALUESPEC(BPCLADBOTTOMSURF)).AND.
     c    (SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c    SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
             BGT=GTSECT
             EXIT
          ENDIF
 3580 CONTINUE
        DO 3610 GTSECT=BGT,NUMOFGTAXS(DESNUM(COLUMN,ROW))
          IF (GTSECT.EQ.BGT) THEN
             WRITE(30,3590) LN, BMODML, (-1*MODDENSITY),
     c       (-1*GTSECTIRSURF(GTSECT)),
     c       (-1*BPCLADBOTTOMSURF),
     c       GTSECTBOTSURF(GTSECT),
     c       BPRAUNIV(COLUMN,ROW)
 3590        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,
     c       '   $ Borated moderator inside guide tube')
             LN=LN+1
          ELSE
             WRITE(30,3600) LN, BMODML, (-1*MODDENSITY),
     c       (-1*GTSECTIRSURF(GTSECT)),
     c       (-1*GTSECTTOPSURF(GTSECT)),
     c       GTSECTBOTSURF(GTSECT),
     c       BPRAUNIV(COLUMN,ROW)
 3600        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,
     c       '   $ Borated moderator inside guide tube')
             LN=LN+1
          ENDIF
 3610 CONTINUE
*     Define the lower end-fitting top surface.
               CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
```

```
            DO 3620 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
3620        CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               BPLEFTOPSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               BPLEFTOPSURF=CURRENTSURFLABEL
            ENDIF
*     Write the lower end-fitting cell specification for this BPR universe.
         GTBOTSURF=GTSECTBOTSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))
         IF (SURFVALUESPEC(GTBOTSURF).GE.
     c   ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)) THEN
            WRITE(30,3630) LN, FRLEFML(COLUMN,ROW),
     c      (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*BPLEFTOPSURF),
     c      BPRAUNIV(COLUMN,ROW)
3630        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,' IMP:N=1 U=',I3,
     c      '    $ Lower end-fitting')
            LN=LN+1
         ELSE
            WRITE(30,3640) LN, FRLEFML(COLUMN,ROW),
     c      (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*BPLEFTOPSURF),
     c      GTSECTORSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW))),
     c      BPRAUNIV(COLUMN,ROW)
3640        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'    $ Lower end-fitting')
            LN=LN+1
            WRITE(30,3650) LN, FRLEFML(COLUMN,ROW),
     c      (-1*LEFMAT(DESNUM(COLUMN,ROW),1)),
     c      (-1*GTSECTBOTSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))),
     c      (-1*GTSECTORSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))),
     c      BPRAUNIV(COLUMN,ROW)
3650        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'    $ Lower end-fitting')
            LN=LN+1
         ENDIF
*     Write the upper end-fitting cell specification for this BPR universe.
         IF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c   SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c   (SURFVALUESPEC(GTSECTTOPSURF(1)).LE.
     c   SURFVALUESPEC(UEFBOTTOMSURF))) THEN
            WRITE(30,3660) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c      (-1*UEFTOPSURF), BPRAUNIV(COLUMN,ROW)
3660        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
            LN=LN+1
```

```
          ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c    SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c    (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c    SURFVALUESPEC(UEFTOPSURF)).AND.
     c    (SURFVALUESPEC(GTSECTTOPSURF(1)).LE.
     c    SURFVALUESPEC(UEFBOTTOMSURF))) THEN
            WRITE(30,3670) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c      (-1*UEFTOPSURF), BPOCORSURF, BPRAUNIV(COLUMN,ROW)
 3670       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
            LN=LN+1
            WRITE(30,3680) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), BPCLADTOPSURF,
     c      (-1*UEFTOPSURF), (-1*BPOCORSURF),
     c      BPRAUNIV(COLUMN,ROW)
 3680       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
            LN=LN+1
          ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c    SURFVALUESPEC(GTSECTTOPSURF(1))).AND.
     c    (SURFVALUESPEC(GTSECTTOPSURF(1)).LT.
     c    SURFVALUESPEC(UEFTOPSURF)).AND.
     c    (SURFVALUESPEC(GTSECTTOPSURF(1)).GT.
     c    SURFVALUESPEC(UEFBOTTOMSURF))) THEN
            WRITE(30,3690) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c      (-1*UEFTOPSURF), GTSECTORSURF(1), BPRAUNIV(COLUMN,ROW)
 3690       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
            LN=LN+1
            WRITE(30,3700) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTSECTTOPSURF(1),
     c      (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c      BPRAUNIV(COLUMN,ROW)
 3700       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
            LN=LN+1
            WRITE(30,3710) LN, BMODML,
     c      (-1*MODDENSITY), BPCLADTOPSURF,
     c      (-1*GTSECTTOPSURF(1)), (-1*GTSECTIRSURF(1)),
     c      BPRAUNIV(COLUMN,ROW)
 3710       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
            LN=LN+1
          ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c    SURFVALUESPEC(GTSECTTOPSURF(1))).AND.
     c    (SURFVALUESPEC(GTSECTTOPSURF(1)).EQ.
     c    SURFVALUESPEC(UEFTOPSURF))) THEN
            WRITE(30,3720) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c      (-1*UEFTOPSURF), GTSECTORSURF(1),
     c      BPRAUNIV(COLUMN,ROW)
 3720       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
```

```
     c        ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
              LN=LN+1
           IF (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c     SURFVALUESPEC(GTSECTTOPSURF(1))) THEN
              WRITE(30,3730) LN, BMODML,
     c        (-1*MODDENSITY), BPCLADTOPSURF,
     c        (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c        BPRAUNIV(COLUMN,ROW)
 3730         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
              LN=LN+1
           ENDIF
           ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c     SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c     (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c     SURFVALUESPEC(UEFTOPSURF)).AND.
     c     (SURFVALUESPEC(GTSECTTOPSURF(1)).GT.
     c     SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c     (SURFVALUESPEC(GTSECTTOPSURF(1)).LT.
     c     SURFVALUESPEC(BPCLADTOPSURF))) THEN
              WRITE(30,3740) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c        (-1*UEFTOPSURF), GTSECTORSURF(1),
     c        BPRAUNIV(COLUMN,ROW)
 3740         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
              LN=LN+1
              WRITE(30,3750) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)),
     c        GTSECTTOPSURF(1),
     c        (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c        BPOCORSURF, BPRAUNIV(COLUMN,ROW)
 3750         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        1X,I4,' IMP:N=1 U=',I3,
     c        '    $ Assembly upper end-fitting')
              LN=LN+1
              WRITE(30,3760) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), BPCLADTOPSURF,
     c        (-1*UEFTOPSURF), (-1*BPOCORSURF),
     c        BPRAUNIV(COLUMN,ROW)
 3760         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
              LN=LN+1
           ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).EQ.
     c     SURFVALUESPEC(UEFTOPSURF)).AND.
     c     (SURFVALUESPEC(GTSECTTOPSURF(1)).GT.
     c     SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c     (SURFVALUESPEC(GTSECTTOPSURF(1)).LT.
     c     SURFVALUESPEC(BPCLADTOPSURF))) THEN
              WRITE(30,3770) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c        (-1*UEFTOPSURF), GTSECTORSURF(1),
     c        BPRAUNIV(COLUMN,ROW)
 3770         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
```

```
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,3780) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)),
     c          GTSECTTOPSURF(1),
     c          (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c          BPOCORSURF, BPRAUNIV(COLUMN,ROW)
 3780           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          1X,I4,' IMP:N=1 U=',I3,
     c          '    $ Assembly upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(1)).EQ.
     c       SURFVALUESPEC(BPCLADTOPSURF))) THEN
                WRITE(30,3790) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1),
     c          BPRAUNIV(COLUMN,ROW)
 3790           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,3800) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)),
     c          GTSECTTOPSURF(1),
     c          (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c          BPOCORSURF, BPRAUNIV(COLUMN,ROW)
 3800           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          1X,I4,' IMP:N=1 U=',I3,
     c          '    $ Assembly upper end-fitting')
                LN=LN+1
             ENDIF
          ENDIF
       ELSE
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
```

```
*********************************************************************
*********************************************************************
*********************************************************************
         IF (WBPRA(BANKNUM(COLUMN,ROW)).EQ.2) THEN
         DO 3970 MCNPNODE=1,NUMOFBPRANODES(BANKNUM(COLUMN,ROW))
*     Define the upper end-fitting bottom surface.
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 3810 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 3810          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  UEFBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  UEFBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
*     Define the upper end-fitting top surface.
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),1)+
     c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 3820 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 3820          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  UEFTOPSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  UEFTOPSURF=CURRENTSURFLABEL
               ENDIF
*     Define the inner BPR cladding inner radius.
               CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),1)
               CURRENTSURFLABEL=0
               DO 3830 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
```

```
                     ENDIF
                  ENDIF
 3830          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPICIRSURF=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPICIRSURF=CURRENTSURFLABEL
               ENDIF
*     Define the inner BPR cladding outer radius.
               CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 3840 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 3840          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPICORSURF=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPICORSURF=CURRENTSURFLABEL
               ENDIF
*     Define the BP absorber inner radius.
               CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),3)
               CURRENTSURFLABEL=0
               DO 3850 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 3850          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPIRSURF=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPIRSURF=CURRENTSURFLABEL
               ENDIF
*     Define the BP absorber outer radius.
               CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),4)
               CURRENTSURFLABEL=0
               DO 3860 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
```

```
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 3860          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPORSURF=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPORSURF=CURRENTSURFLABEL
               ENDIF
*     Define the outer BPR cladding inner radius.
               CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),5)
               CURRENTSURFLABEL=0
               DO 3870 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 3870          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPOCIRSURF=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPOCIRSURF=CURRENTSURFLABEL
               ENDIF
*     Define the outer BPR cladding outer radius.
               CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),6)
               CURRENTSURFLABEL=0
               DO 3880 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 3880          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPOCORSURF=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPOCORSURF=CURRENTSURFLABEL
               ENDIF
*     Define the BP node bounding surfaces.
            IF (MCNPNODE.EQ.1) THEN
```

```
            TOTBPHEIGHT=0.0
            DO 3890 Z=1,NUMOFBPRANODES(BANKNUM(COLUMN,ROW))
               TOTBPHEIGHT=TOTBPHEIGHT+
     c         MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),Z)
 3890       CONTINUE
            CURRENTSURF=BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))+
     c      TOTBPHEIGHT
            IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
               CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
            ENDIF
            CURRENTSURFLABEL=0
            DO 3900 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 3900       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               TOPBPNODETOPSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               TOPBPNODETOPSURF=CURRENTSURFLABEL
            ENDIF
            BPNODETOPSURF=TOPBPNODETOPSURF
            CURRENTSURF=SURFVALUESPEC(BPNODETOPSURF)-
     c      MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),MCNPNODE)
            IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
               CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
            ENDIF
            CURRENTSURFLABEL=0
            DO 3910 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 3910       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               BPNODEBOTTOMSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               BPNODEBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
         ELSEIF (MCNPNODE.NE.1) THEN
            BPNODETOPSURF=BPNODEBOTTOMSURF
            CURRENTSURF=SURFVALUESPEC(BPNODETOPSURF)-
     c      MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),MCNPNODE)
```

```
               IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                  CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
               ENDIF
               CURRENTSURFLABEL=0
               DO 3930 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 3930          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPNODEBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPNODEBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
            ENDIF
         IF (SURFVALUESPEC(BPNODEBOTTOMSURF).LT.
     c   SURFVALUESPEC(UEFTOPSURF)) THEN
*     Write the BP node cells in this BPR universe.
         IF ((BPRABSNODE(BANKNUM(COLUMN,ROW),MCNPNODE).EQ.'Y').AND.
     c      (BPNONABSMAT(BANKNUM(COLUMN,ROW)).EQ.1)) THEN
            WRITE(30,3940) LN, BPNODEML,
     c      (-1*AL2O3DENSITY(BANKNUM(COLUMN,ROW))), (-1*BPORSURF),
     c      BPIRSURF, (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW), MCNPNODE
 3940       FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c      1X,I4,' IMP:N=1 U=',I3,'   $ Burnable poison node ',I2)
            LN=LN+1
            BPNODEML=BPNODEML+1
         ELSEIF ((BPRABSNODE(BANKNUM(COLUMN,ROW),MCNPNODE).EQ.'Y').AND.
     c      (BPNONABSMAT(BANKNUM(COLUMN,ROW)).NE.1)) THEN
            WRITE(30,3950) LN, BPNODEML,
     c      (-1*NONBPMATDATA(BANKNUM(COLUMN,ROW),1)), (-1*BPORSURF),
     c      BPIRSURF, (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW), MCNPNODE
 3950       FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c      1X,I4,' IMP:N=1 U=',I3,'   $ Burnable poison node ',I2)
            LN=LN+1
            BPNODEML=BPNODEML+1
         ELSE
            WRITE(30,3960) LN, BPNODEML,
     c      (-1*BPDENTOGO(COLUMN,ROW,MCNPNODE)), (-1*BPORSURF),
     c      BPIRSURF, (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW), MCNPNODE
 3960       FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c      1X,I4,' IMP:N=1 U=',I3,'   $ Burnable poison node ',I2)
            LN=LN+1
            BPNODEML=BPNODEML+1
         ENDIF
```

```
          ENDIF
 3970      CONTINUE
*      Define the BPR cladding top surface.
                CURRENTSURF=TOTBPHEIGHT+
     c          BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))+
     c          BPRPLEN(BANKNUM(COLUMN,ROW),1)
                IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                   CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                ENDIF
                CURRENTSURFLABEL=0
                DO 3980 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 3980           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   BPCLADTOPSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   BPCLADTOPSURF=CURRENTSURFLABEL
                ENDIF
*      Define the BPR cladding bottom surface.
                CURRENTSURF=BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))-
     c          BPRPLEN(BANKNUM(COLUMN,ROW),2)
                CURRENTSURFLABEL=0
                DO 3990 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 3990           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   BPCLADBOTTOMSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   BPCLADBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
*      Write the inner BP-to-cladding gap cell in this BPR universe.
          WRITE(30,4000) LN, (-1*BPIRSURF), BPICORSURF,
     c    (-1*TOPBPNODETOPSURF),
     c    BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 4000     FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c    ' IMP:N=1 U=',I3,
     c    '    $ Burnable poison-to-cladding gap')
          LN=LN+1
```

```
*     Write the outer BP-to-cladding gap cell in this BPR universe.
            WRITE(30,4010) LN, (-1*BPOCIRSURF), BPORSURF,
     c      (-1*TOPBPNODETOPSURF),
     c      BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 4010       FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,
     c      '   $ Burnable poison-to-cladding gap')
            LN=LN+1
*     Write the annular gap cell in this BPR universe.
            WRITE(30,4020) LN, (-1*BPICIRSURF),
     c      (-1*TOPBPNODETOPSURF),
     c      BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 4020       FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,
     c      '   $ Burnable poison-to-cladding gap')
            LN=LN+1
*     Write the BPR cladding cell in this BPR universe.
*     Determine if the BPR cladding material specification has
*     previously been defined.  If it has been previously defined, determine
*     the cladding material specification label.
            CLADMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 4040 RO=1,(ROW-1)
                  DO 4030 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                        IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  BPRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
 4030             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 4040          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 4060 RO=ROW,ROW
                     DO 4050 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                           IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                     BPRCLADMAT(BANKNUM(CO,RO))) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
```

```
                    ENDIF
 4050            CONTINUE
                 IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                 ENDIF
 4060         CONTINUE
           ENDIF
        ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
           DO 4080 RO=1,(ROW-1)
              DO 4070 CO=1,50
                 IF (BANKNUM(CO,RO).NE.0) THEN
                 IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                    IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c              BPRCLADMAT(BANKNUM(CO,RO))) THEN
                       CLADMLUNIQUE=.FALSE.
                       LEAVE=.TRUE.
                       BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                       EXIT
                    ENDIF
                 ENDIF
                 ENDIF
 4070         CONTINUE
              IF (LEAVE.EQ..TRUE.) THEN
                 EXIT
              ENDIF
 4080      CONTINUE
        ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
           DO 4100 RO=1,1
              DO 4090 CO=1,(COLUMN-1)
                 IF (BANKNUM(CO,RO).NE.0) THEN
                 IF (BANKDES(BANKNUM(CO,RO)).NE.'BPRA ') THEN
                    IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c              BPRCLADMAT(BANKNUM(CO,RO))) THEN
                       CLADMLUNIQUE=.FALSE.
                       LEAVE=.TRUE.
                       BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                       EXIT
                    ENDIF
                 ENDIF
                 ENDIF
 4090         CONTINUE
              IF (LEAVE.EQ..TRUE.) THEN
                 EXIT
              ENDIF
 4100      CONTINUE
        ENDIF
        IF (CLADMLUNIQUE.EQ..TRUE.) THEN
           BPCLADML(COLUMN,ROW)=MN
*  Check BPR Cladding Material
              IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
                 DO 4110 C=1,2
                    IF (C.EQ.1) THEN
                       WRITE(200,9300) BPCLADML(COLUMN,ROW)
                    ELSEIF (C.EQ.2) THEN
```

```
                  WRITE(200,9301)
                  WRITE(200,7000)
                  WRITE(200,7001)
                  WRITE(200,7002)
                  WRITE(200,9302)
                  WRITE(200,7003)
                  WRITE(200,7004)
                  WRITE(200,7005)
                  WRITE(200,9303)
                  WRITE(200,9304)
               ENDIF
4110        CONTINUE
         ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW))
     c   .EQ.2) THEN
            DO 4120 C=1,2
               IF (C.EQ.1) THEN
                  WRITE(200,9305) BPCLADML(COLUMN,ROW)
               ELSEIF (C.EQ.2) THEN
                  WRITE(200,9306)
                  WRITE(200,9307)
                  WRITE(200,9308)
                  WRITE(200,9309)
                  WRITE(200,9310)
                  WRITE(200,7006)
                  WRITE(200,7007)
                  WRITE(200,7008)
                  WRITE(200,9311)
                  WRITE(200,9312)
                  WRITE(200,7009)
                  WRITE(200,7010)
                  WRITE(200,7011)
                  WRITE(200,9313)
                  WRITE(200,7012)
                  WRITE(200,7013)
                  WRITE(200,7014)
                  WRITE(200,7015)
               ENDIF
4120        CONTINUE
         ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW))
     c   .EQ.3) THEN
            DO 4130 C=1,2
               IF (C.EQ.1) THEN
                  WRITE(200,9314) BPCLADML(COLUMN,ROW)
               ELSEIF (C.EQ.2) THEN
                  WRITE(200,9315)
                  WRITE(200,9316)
                  WRITE(200,9317)
                  WRITE(200,9318)
                  WRITE(200,7016)
                  WRITE(200,7017)
                  WRITE(200,7018)
                  WRITE(200,9319)
                  WRITE(200,9320)
                  WRITE(200,7019)
```

```
                        WRITE(200,7020)
                        WRITE(200,7021)
                        WRITE(200,9321)
                        WRITE(200,7022)
                        WRITE(200,7023)
                        WRITE(200,7024)
                        WRITE(200,7025)
                        WRITE(200,9322)
                        WRITE(200,9323)
                        WRITE(200,9324)
                        WRITE(200,9325)
                        WRITE(200,9326)
                        WRITE(200,9327)
                        WRITE(200,7026)
                        WRITE(200,9328)
                        WRITE(200,9329)
                        WRITE(200,9330)
                     ENDIF
4130              CONTINUE
               ENDIF
            MN=MN+1
         ENDIF
         IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
            CLADRHO=6.56
         ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.2) THEN
            CLADRHO=7.90
         ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.3) THEN
            CLADRHO=8.19
         ENDIF
         WRITE(30,4140) LN, BPCLADML(COLUMN,ROW), (-1*CLADRHO),
    c    BPOCIRSURF,
    c    (-1*BPOCORSURF), (-1*BPCLADTOPSURF), BPCLADBOTTOMSURF,
    c    BPRAUNIV(COLUMN,ROW)
4140     FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
    c    ' IMP:N=1 U=',I3,'   $ BPR cladding')
         LN=LN+1
         WRITE(30,4150) LN, BPCLADML(COLUMN,ROW), (-1*CLADRHO),
    c    BPICIRSURF,
    c    (-1*BPICORSURF), (-1*TOPBPNODETOPSURF), BPNODEBOTTOMSURF,
    c    BPRAUNIV(COLUMN,ROW)
4150     FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
    c    ' IMP:N=1 U=',I3,'   $ BPR cladding')
         LN=LN+1
*     Write the BPR upper plenum cell in this BPR universe.
*     Determine if the BPR upper plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the upper plenum material specification label.
         BPRUPMLUNIQUE=.TRUE.
         LEAVE=.FALSE.
         IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
            DO 4170 RO=1,(ROW-1)
               DO 4160 CO=1,50
                  IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKNUM(COLUMN,ROW).EQ.
```

```
     c                         BANKNUM(CO,RO)) THEN
                               BPRUPMLUNIQUE=.FALSE.
                               LEAVE=.TRUE.
                               BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                               EXIT
                            ENDIF
                         ENDIF
 4160                 CONTINUE
                      IF (LEAVE.EQ..TRUE.) THEN
                         EXIT
                      ENDIF
 4170              CONTINUE
                   IF (LEAVE.EQ..FALSE.) THEN
                      DO 4190 RO=ROW,ROW
                         DO 4180 CO=1,(COLUMN-1)
                            IF (BANKNUM(CO,RO).NE.0) THEN
                               IF (BANKNUM(COLUMN,ROW).EQ.
     c                         BANKNUM(CO,RO)) THEN
                                  BPRUPMLUNIQUE=.FALSE.
                                  LEAVE=.TRUE.
                                  BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                                  EXIT
                               ENDIF
                            ENDIF
 4180                    CONTINUE
                         IF (LEAVE.EQ..TRUE.) THEN
                            EXIT
                         ENDIF
 4190                 CONTINUE
                   ENDIF
                ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                   DO 4210 RO=1,(ROW-1)
                      DO 4200 CO=1,50
                         IF (BANKNUM(CO,RO).NE.0) THEN
                            IF (BANKNUM(COLUMN,ROW).EQ.
     c                         BANKNUM(CO,RO)) THEN
                               BPRUPMLUNIQUE=.FALSE.
                               LEAVE=.TRUE.
                               BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                               EXIT
                            ENDIF
                         ENDIF
 4200                 CONTINUE
                      IF (LEAVE.EQ..TRUE.) THEN
                         EXIT
                      ENDIF
 4210              CONTINUE
                ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                   DO 4230 RO=1,1
                      DO 4220 CO=1,(COLUMN-1)
                         IF (BANKNUM(CO,RO).NE.0) THEN
                            IF (BANKNUM(COLUMN,ROW).EQ.
     c                         BANKNUM(CO,RO)) THEN
                               BPRUPMLUNIQUE=.FALSE.
```

```
                        LEAVE=.TRUE.
                        BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
 4220          CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
 4230       CONTINUE
         ENDIF
      IF (SURFVALUESPEC(TOPBPNODETOPSURF).LT.
    c SURFVALUESPEC(UEFTOPSURF)) THEN
         IF (BPRUPMLUNIQUE.EQ..TRUE.) THEN
            BPRUPML(COLUMN,ROW)=MN
*  Check Burnable Poison Rod Upper Plenum Regions
               DO 4260 C=1,BPRUPLENMAT(BANKNUM(COLUMN,ROW),2)
                  IF (C.EQ.1) THEN
                     WRITE(200,4240) BPRUPML(COLUMN,ROW),
    c                BPRUPLENZAIDS(BANKNUM(COLUMN,ROW),C),
    c                (-1*BPRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 4240                FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
    c                '   $ Burnable Poison Rod Upper Plenum')
                  ELSE
                     WRITE(200,4250)
    c                BPRUPLENZAIDS(BANKNUM(COLUMN,ROW),C),
    c                (-1*BPRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 4250                FORMAT(T9,A9,3X,G14.6)
                  ENDIF
 4260          CONTINUE
            MN=MN+1
         ENDIF
         WRITE(30,4270) LN, BPRUPML(COLUMN,ROW),
    c    (-1*BPRUPLENMAT(BANKNUM(COLUMN,ROW),1)),
    c    TOPBPNODETOPSURF,
    c    (-1*BPCLADTOPSURF), (-1*BPOCIRSURF),
    c    BPRAUNIV(COLUMN,ROW)
 4270    FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
    c    ' IMP:N=1 U=',I3,'   $ BPR upper plenum region')
         LN=LN+1
      ENDIF
*     Write the BPR lower plenum cell (lower end plug) in this BPR universe.
*     Determine if the BPR lower plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the lower plenum material specification label.
         BPRLPMLUNIQUE=.TRUE.
         LEAVE=.FALSE.
         IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
            DO 4290 RO=1,(ROW-1)
               DO 4280 CO=1,50
                  IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO)) THEN
                        BPRLPMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
```

```
                    BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                    EXIT
                  ENDIF
                ENDIF
4280          CONTINUE
              IF (LEAVE.EQ..TRUE.) THEN
                EXIT
              ENDIF
4290        CONTINUE
            IF (LEAVE.EQ..FALSE.) THEN
              DO 4310 RO=ROW,ROW
                DO 4300 CO=1,(COLUMN-1)
                  IF (BANKNUM(CO,RO).NE.0) THEN
                    IF (BANKNUM(COLUMN,ROW).EQ.
     c              BANKNUM(CO,RO)) THEN
                      BPRLPMLUNIQUE=.FALSE.
                      LEAVE=.TRUE.
                      BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                      EXIT
                    ENDIF
                  ENDIF
4300            CONTINUE
                IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
                ENDIF
4310          CONTINUE
            ENDIF
          ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
            DO 4330 RO=1,(ROW-1)
              DO 4320 CO=1,50
                IF (BANKNUM(CO,RO).NE.0) THEN
                  IF (BANKNUM(COLUMN,ROW).EQ.
     c            BANKNUM(CO,RO)) THEN
                    BPRLPMLUNIQUE=.FALSE.
                    LEAVE=.TRUE.
                    BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                    EXIT
                  ENDIF
                ENDIF
4320          CONTINUE
              IF (LEAVE.EQ..TRUE.) THEN
                EXIT
              ENDIF
4330        CONTINUE
          ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
            DO 4350 RO=1,1
              DO 4340 CO=1,(COLUMN-1)
                IF (BANKNUM(CO,RO).NE.0) THEN
                  IF (BANKNUM(COLUMN,ROW).EQ.
     c            BANKNUM(CO,RO)) THEN
                    BPRLPMLUNIQUE=.FALSE.
                    LEAVE=.TRUE.
                    BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                    EXIT
```

```
                        ENDIF
                     ENDIF
 4340             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 4350          CONTINUE
            ENDIF
            IF (BPRLPMLUNIQUE.EQ..TRUE.) THEN
               BPRLPML(COLUMN,ROW)=MN
*  Check Burnable Poison Rod Lower Plenum Regions
                  DO 4380 C=1,BPRLPLENMAT(BANKNUM(COLUMN,ROW),2)
                     IF (C.EQ.1) THEN
                        WRITE(200,4360) BPRLPML(COLUMN,ROW),
    c                   BPRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
    c                   (-1*BPRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 4360                   FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
    c                   '    $ Burnable Poison Rod Lower Plenum')
                     ELSE
                        WRITE(200,4370)
    c                   BPRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
    c                   (-1*BPRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 4370                   FORMAT(T9,A9,3X,G14.6)
                     ENDIF
 4380             CONTINUE
               MN=MN+1
            ENDIF
            WRITE(30,4390) LN, BPRLPML(COLUMN,ROW),
    c       (-1*BPRLPLENMAT(BANKNUM(COLUMN,ROW),1)), BPCLADBOTTOMSURF,
    c       (-1*BPNODEBOTTOMSURF), (-1*BPOCIRSURF),
    c       BPRAUNIV(COLUMN,ROW)
 4390       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
    c       ' IMP:N=1 U=',I3,'    $ BPR lower plenum region')
            LN=LN+1
*     Loop through the regions above the BPR (i.e. the appropriate upper core
regions)
*     Define the upper region lower surface.
               DO 4430 REGION=1,NUMREGABOVEBPRA
*     Determine the current upper region's lower surface specification.
                  IF (REGION.EQ.1) THEN
                     REGIONTOPSURF=SYSTEMTOP
                     CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
    c                REGABOVEBPRA(REGION,1)
                  ENDIF
                  CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
    c             REGABOVEBPRA(REGION,1)
                  IF (REGION.EQ.NUMREGABOVEBPRA) THEN
                     REGIONBOTTOMSURF=UEFTOPSURF
                  ELSE
                  CURRENTSURFLABEL=0
                  DO 4400 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                           CURRENTSURFLABEL=V
```

```
                          EXIT
                       ENDIF
                    ENDIF
 4400            CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
                    REGIONBOTTOMSURF=SN
                    SURFTYPESPEC(SN)='PZ'
                    SURFVALUESPEC(SN)=CURRENTSURF
                    SN=SN+1
                 ELSE
                    REGIONBOTTOMSURF=CURRENTSURFLABEL
                 ENDIF
                 ENDIF
*     Write the cell specification for the BPR universe upper region.
              IF (REGION.EQ.1) THEN
                 WRITE(30,4410) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c           (-1*REGABOVEBPRA(REGION,2)),
     c           REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 4410            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
     c           ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                 LN=LN+1
                 REGIONTOPSURF=REGIONBOTTOMSURF
              ELSE
                 WRITE(30,4420) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c           (-1*REGABOVEBPRA(REGION,2)), (-1*REGIONTOPSURF),
     c           REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 4420            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                 LN=LN+1
                 REGIONTOPSURF=REGIONBOTTOMSURF
              ENDIF
 4430         CONTINUE
*     Define the GT top surface.
              CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),3)
              IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                 CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
              ENDIF
              CURRENTSURFLABEL=0
              DO 4440 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 4440         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 GTTOPSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 GTTOPSURF=CURRENTSURFLABEL
              ENDIF
```

```
*     Define the GT bottom surface.
            CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),4)
            CURRENTSURFLABEL=0
            DO 4450 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 4450       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               GTBOTSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               GTBOTSURF=CURRENTSURFLABEL
            ENDIF
*     Define the GT outer radius surface.
            CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),2)
            CURRENTSURFLABEL=0
            DO 4460 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'CZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 4460       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               GTORSURF=SN
               SURFTYPESPEC(SN)='CZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               GTORSURF=CURRENTSURFLABEL
            ENDIF
*     Define the GT inner radius surface.
            CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),1)
            CURRENTSURFLABEL=0
            DO 4470 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'CZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 4470       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               GTIRSURF=SN
               SURFTYPESPEC(SN)='CZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
```

```
            ELSE
               GTIRSURF=CURRENTSURFLABEL
            ENDIF
*     Define the lower end-fitting top surface.
            CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
            CURRENTSURFLABEL=0
            DO 4480 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 4480       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               BPLEFTOPSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               BPLEFTOPSURF=CURRENTSURFLABEL
            ENDIF
*     Write the lower end-fitting cell specification for this BPR universe.
         IF (SURFVALUESPEC(GTBOTSURF).GE.
     c   ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)) THEN
            WRITE(30,4490) LN, FRLEFML(COLUMN,ROW),
     c      (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*BPLEFTOPSURF),
     c      BPRAUNIV(COLUMN,ROW)
 4490       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,' IMP:N=1 U=',I3,
     c      '    $ Assembly lower end-fitting')
            LN=LN+1
         ELSE
            WRITE(30,4500) LN, FRLEFML(COLUMN,ROW),
     c      (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*BPLEFTOPSURF),
     c      GTORSURF, BPRAUNIV(COLUMN,ROW)
 4500       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'    $ Assembly lower end-fitting')
            LN=LN+1
            WRITE(30,4510) LN, FRLEFML(COLUMN,ROW),
     c      (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*GTBOTSURF),
     c      (-1*GTORSURF), BPRAUNIV(COLUMN,ROW)
 4510       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'    $ Assembly lower end-fitting')
            LN=LN+1
         ENDIF
*     Write the upper end-fitting cell specification for this BPR universe.
         IF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c   SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c   (SURFVALUESPEC(GTTOPSURF).LE.
     c   SURFVALUESPEC(UEFBOTTOMSURF))) THEN
            WRITE(30,4520) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c      (-1*UEFTOPSURF), BPRAUNIV(COLUMN,ROW)
 4520       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
```

```
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c        SURFVALUESPEC(UEFTOPSURF)).AND.
     c        (SURFVALUESPEC(GTTOPSURF).LE.
     c        SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                WRITE(30,4530) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), BPOCORSURF, BPRAUNIV(COLUMN,ROW)
 4530           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,4540) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), BPCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*BPOCORSURF),
     c          BPRAUNIV(COLUMN,ROW)
 4540           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c        SURFVALUESPEC(GTTOPSURF)).AND.
     c        (SURFVALUESPEC(GTTOPSURF).LT.
     c        SURFVALUESPEC(UEFTOPSURF)).AND.
     c        (SURFVALUESPEC(GTTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                WRITE(30,4550) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
 4550           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,4560) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTORSURF),
     c          BPRAUNIV(COLUMN,ROW)
 4560           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,4570) LN, BMODML,
     c          (-1*MODDENSITY), BPCLADTOPSURF,
     c          (-1*GTTOPSURF), (-1*GTIRSURF),
     c          BPRAUNIV(COLUMN,ROW)
 4570           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c        SURFVALUESPEC(GTTOPSURF)).AND.
     c        (SURFVALUESPEC(GTTOPSURF).EQ.
     c        SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,4580) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
```

```
4580        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
            LN=LN+1
          IF (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c    SURFVALUESPEC(GTTOPSURF)) THEN
            WRITE(30,4590) LN, BMODML,
     c       (-1*MODDENSITY), BPCLADTOPSURF,
     c       (-1*UEFTOPSURF), (-1*GTIRSURF),
     c       BPRAUNIV(COLUMN,ROW)
4590        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
            LN=LN+1
          ENDIF
          ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c    SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c    (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c    SURFVALUESPEC(UEFTOPSURF)).AND.
     c    (SURFVALUESPEC(GTTOPSURF).GT.
     c    SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c    (SURFVALUESPEC(GTTOPSURF).LT.
     c    SURFVALUESPEC(BPCLADTOPSURF))) THEN
            WRITE(30,4600) LN, FRUEFML(COLUMN,ROW),
     c       (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c       (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
4600        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
            LN=LN+1
            WRITE(30,4610) LN, FRUEFML(COLUMN,ROW),
     c       (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c       (-1*UEFTOPSURF), (-1*GTORSURF),
     c       BPOCORSURF, BPRAUNIV(COLUMN,ROW)
4610        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       1X,I4,' IMP:N=1 U=',I3,
     c       '   $ Assembly upper end-fitting')
            LN=LN+1
            WRITE(30,4620) LN, FRUEFML(COLUMN,ROW),
     c       (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), BPCLADTOPSURF,
     c       (-1*UEFTOPSURF), (-1*BPOCORSURF),
     c       BPRAUNIV(COLUMN,ROW)
4620        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
            LN=LN+1
          ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).EQ.
     c    SURFVALUESPEC(UEFTOPSURF)).AND.
     c    (SURFVALUESPEC(GTTOPSURF).GT.
     c    SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c    (SURFVALUESPEC(GTTOPSURF).LT.
     c    SURFVALUESPEC(BPCLADTOPSURF))) THEN
            WRITE(30,4630) LN, FRUEFML(COLUMN,ROW),
     c       (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c       (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
4630        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
            LN=LN+1
```

```
            WRITE(30,4640) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c      (-1*UEFTOPSURF), (-1*GTORSURF),
     c      BPOCORSURF, BPRAUNIV(COLUMN,ROW)
 4640       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      1X,I4,' IMP:N=1 U=',I3,
     c      '   $ Assembly upper end-fitting')
            LN=LN+1
         ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c   SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c   (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c   SURFVALUESPEC(UEFTOPSURF)).AND.
     c   (SURFVALUESPEC(GTTOPSURF).EQ.
     c   SURFVALUESPEC(BPCLADTOPSURF))) THEN
            WRITE(30,4650) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c      (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
 4650       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
            LN=LN+1
            WRITE(30,4660) LN, FRUEFML(COLUMN,ROW),
     c      (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c      (-1*UEFTOPSURF), (-1*GTORSURF),
     c      BPOCORSURF, BPRAUNIV(COLUMN,ROW)
 4660       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      1X,I4,' IMP:N=1 U=',I3,
     c      '   $ Assembly upper end-fitting')
            LN=LN+1
         ENDIF
*     Write the GT material cell in this BPR universe.
*     Determine if the GT material specification has
*     previously been defined.  If it has been previously defined, determine
*     the material specification label.
         CLADMLUNIQUE=.TRUE.
         LEAVE=.FALSE.
         IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
            DO 4680 RO=1,(ROW-1)
               DO 4670 CO=1,50
                  IF (DESNUM(CO,RO).NE.0) THEN
                     IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c               GTMAT(DESNUM(CO,RO))) THEN
                        CLADMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        GTML(COLUMN,ROW)=GTML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
 4670          CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
 4680       CONTINUE
            IF (LEAVE.EQ..FALSE.) THEN
               DO 4700 RO=ROW,ROW
```

```
                        DO 4690 CO=1,(COLUMN-1)
                           IF (DESNUM(CO,RO).NE.0) THEN
                              IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                        GTMAT(DESNUM(CO,RO))) THEN
                                 CLADMLUNIQUE=.FALSE.
                                 LEAVE=.TRUE.
                                 GTML(COLUMN,ROW)=GTML(CO,RO)
                                 EXIT
                              ENDIF
                           ENDIF
 4690                   CONTINUE
                        IF (LEAVE.EQ..TRUE.) THEN
                           EXIT
                        ENDIF
 4700                CONTINUE
                  ENDIF
               ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                  DO 4720 RO=1,(ROW-1)
                     DO 4710 CO=1,50
                        IF (DESNUM(CO,RO).NE.0) THEN
                           IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                     GTMAT(DESNUM(CO,RO))) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              GTML(COLUMN,ROW)=GTML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 4710                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 4720             CONTINUE
               ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                  DO 4740 RO=1,1
                     DO 4730 CO=1,(COLUMN-1)
                        IF (DESNUM(CO,RO).NE.0) THEN
                           IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                     GTMAT(DESNUM(CO,RO))) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              GTML(COLUMN,ROW)=GTML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 4730                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 4740             CONTINUE
               ENDIF
               IF (CLADMLUNIQUE.EQ..TRUE.) THEN
                  GTML(COLUMN,ROW)=MN
*  Check Guide Tube Material
```

```
      IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.1) THEN
         DO 4750 C=1,2
            IF (C.EQ.1) THEN
               WRITE(200,9300) GTML(COLUMN,ROW)
            ELSEIF (C.EQ.2) THEN
               WRITE(200,9301)
               WRITE(200,7000)
               WRITE(200,7001)
               WRITE(200,7002)
               WRITE(200,9302)
               WRITE(200,7003)
               WRITE(200,7004)
               WRITE(200,7005)
               WRITE(200,9303)
               WRITE(200,9304)
            ENDIF
4750     CONTINUE
      ELSEIF (GTMAT(DESNUM(COLUMN,ROW))
     c .EQ.2) THEN
         DO 4760 C=1,2
            IF (C.EQ.1) THEN
               WRITE(200,9305) GTML(COLUMN,ROW)
            ELSEIF (C.EQ.2) THEN
               WRITE(200,9306)
               WRITE(200,9307)
               WRITE(200,9308)
               WRITE(200,9309)
               WRITE(200,9310)
               WRITE(200,7006)
               WRITE(200,7007)
               WRITE(200,7008)
               WRITE(200,9311)
               WRITE(200,9312)
               WRITE(200,7009)
               WRITE(200,7010)
               WRITE(200,7011)
               WRITE(200,9313)
               WRITE(200,7012)
               WRITE(200,7013)
               WRITE(200,7014)
               WRITE(200,7015)
            ENDIF
4760     CONTINUE
      ELSEIF (GTMAT(DESNUM(COLUMN,ROW))
     c .EQ.3) THEN
         DO 4770 C=1,2
            IF (C.EQ.1) THEN
               WRITE(200,9314) GTML(COLUMN,ROW)
            ELSEIF (C.EQ.2) THEN
               WRITE(200,9315)
               WRITE(200,9316)
               WRITE(200,9317)
               WRITE(200,9318)
               WRITE(200,7016)
```

```
                        WRITE(200,7017)
                        WRITE(200,7018)
                        WRITE(200,9319)
                        WRITE(200,9320)
                        WRITE(200,7019)
                        WRITE(200,7020)
                        WRITE(200,7021)
                        WRITE(200,9321)
                        WRITE(200,7022)
                        WRITE(200,7023)
                        WRITE(200,7024)
                        WRITE(200,7025)
                        WRITE(200,9322)
                        WRITE(200,9323)
                        WRITE(200,9324)
                        WRITE(200,9325)
                        WRITE(200,9326)
                        WRITE(200,9327)
                        WRITE(200,7026)
                        WRITE(200,9328)
                        WRITE(200,9329)
                        WRITE(200,9330)
                     ENDIF
 4770             CONTINUE
               ENDIF
            MN=MN+1
          ENDIF
          IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.1) THEN
             CLADRHO=6.56
          ELSEIF (GTMAT(DESNUM(COLUMN,ROW)).EQ.2) THEN
             CLADRHO=7.90
          ELSEIF (GTMAT(DESNUM(COLUMN,ROW)).EQ.3) THEN
             CLADRHO=8.19
          ENDIF
          WRITE(30,4780) LN, GTML(COLUMN,ROW), (-1*CLADRHO),
     c    GTIRSURF,
     c    (-1*GTORSURF), (-1*GTTOPSURF), GTBOTSURF,
     c    BPRAUNIV(COLUMN,ROW)
 4780     FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c    ' IMP:N=1 U=',I3,'   $ Guide tube region')
          LN=LN+1
*     Write the moderator cells within the GT in this BPR universe.
          IF (SURFVALUESPEC(BPCLADTOPSURF).GE.
     c    SURFVALUESPEC(GTTOPSURF)) THEN
             WRITE(30,4790) LN, BMODML, (-1*MODDENSITY),
     c       (-1*GTIRSURF),
     c       BPOCORSURF, (-1*GTTOPSURF), GTBOTSURF,
     c       BPRAUNIV(COLUMN,ROW)
 4790        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,
     c       '   $ Borated moderator inside guide tube')
             LN=LN+1
          ELSEIF (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c    SURFVALUESPEC(GTTOPSURF)) THEN
```

```
                WRITE(30,4800) LN, BMODML, (-1*MODDENSITY),
     c          (-1*GTIRSURF),
     c          BPOCORSURF, (-1*BPCLADTOPSURF), GTBOTSURF,
     c          BPRAUNIV(COLUMN,ROW)
 4800           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '    $ Borated moderator inside guide tube')
                LN=LN+1
              ENDIF
              WRITE(30,4810) LN, BMODML, (-1*MODDENSITY),
     c        (-1*BPOCORSURF), (-1*BPCLADBOTTOMSURF), GTBOTSURF,
     c        BPRAUNIV(COLUMN,ROW)
 4810         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Borated moderator inside guide tube')
              LN=LN+1
*     Loop through the regions above the BPR (i.e. the appropriate upper core
regions)
*     Define the upper region lower surface.
                DO 4850 REGION=1,NUMREGABOVEBPRA
*     Determine the current upper region's lower surface specification.
                  IF (REGION.EQ.1) THEN
                     REGIONTOPSURF=SYSTEMTOP
                     CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
     c               REGABOVEBPRA(REGION,1)
                  ENDIF
                  CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
     c            REGABOVEBPRA(REGION,1)
                  IF (REGION.EQ.NUMREGABOVEBPRA) THEN
                     REGIONBOTTOMSURF=UEFTOPSURF
                  ELSE
                  CURRENTSURFLABEL=0
                  DO 4820 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                           CURRENTSURFLABEL=V
                           EXIT
                        ENDIF
                     ENDIF
 4820             CONTINUE
                  IF (CURRENTSURFLABEL.EQ.0) THEN
                     REGIONBOTTOMSURF=SN
                     SURFTYPESPEC(SN)='PZ'
                     SURFVALUESPEC(SN)=CURRENTSURF
                     SN=SN+1
                  ELSE
                     REGIONBOTTOMSURF=CURRENTSURFLABEL
                  ENDIF
                  ENDIF
*     Write the cell specification for the BPR universe upper region.
                IF (REGION.EQ.1) THEN
                  WRITE(30,4830) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c            (-1*REGABOVEBPRA(REGION,2)),
     c            REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 4830             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
```

```
     c              ' IMP:N=1 U=',I3,'    $ Upper core region ',I2)
                    LN=LN+1
                    REGIONTOPSURF=REGIONBOTTOMSURF
                 ELSE
                    WRITE(30,4840) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c              (-1*REGABOVEBPRA(REGION,2)), (-1*REGIONTOPSURF),
     c              REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 4840               FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c              ' IMP:N=1 U=',I3,'    $ Upper core region ',I2)
                    LN=LN+1
                    REGIONTOPSURF=REGIONBOTTOMSURF
                 ENDIF
 4850            CONTINUE
            SPACHEIGHT=0.0
*     Loop through the spacer and moderator regions along the axial
*     length of the BPR (from top to bottom).
            DO 4860 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
               SPACHEIGHT=SPACHEIGHT+SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
 4860       CONTINUE
            DO 4940 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
*     Define the homogenized spacer region bounding surfaces.
               IF (SPN.EQ.1) THEN
                  SPACERTOPSURF=UEFBOTTOMSURF
                  CURRENTSURF=SURFVALUESPEC(UEFBOTTOMSURF)-
     c            SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                  CURRENTSURFLABEL=0
                  DO 4870 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                           CURRENTSURFLABEL=V
                           EXIT
                        ENDIF
                     ENDIF
 4870             CONTINUE
                  IF (CURRENTSURFLABEL.EQ.0) THEN
                     SPACERBOTTOMSURF=SN
                     SURFTYPESPEC(SN)='PZ'
                     SURFVALUESPEC(SN)=CURRENTSURF
                     SN=SN+1
                  ELSE
                     SPACERBOTTOMSURF=CURRENTSURFLABEL
                  ENDIF
                  WATERREGIONTOPSURF=SPACERBOTTOMSURF
                  CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
                  CURRENTSURFLABEL=0
                  DO 4880 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                           CURRENTSURFLABEL=V
                           EXIT
                        ENDIF
                     ENDIF
 4880             CONTINUE
                  IF (CURRENTSURFLABEL.EQ.0) THEN
```

```
            WATERREGIONBOTTOMSURF=SN
            SURFTYPESPEC(SN)='PZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
         ELSE
            WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
         ENDIF
      ELSEIF ((SPN.NE.1).AND.(SPN.NE.
c     NUMOFSPACERS(DESNUM(COLUMN,ROW)))) THEN
         SPACERTOPSURF=WATERREGIONBOTTOMSURF
         CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
c        SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
         CURRENTSURFLABEL=0
         DO 4890 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'PZ') THEN
       IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
4890     CONTINUE
         IF (CURRENTSURFLABEL.EQ.0) THEN
            SPACERBOTTOMSURF=SN
            SURFTYPESPEC(SN)='PZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
         ELSE
            SPACERBOTTOMSURF=CURRENTSURFLABEL
         ENDIF
         WATERREGIONTOPSURF=SPACERBOTTOMSURF
         CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
         CURRENTSURFLABEL=0
         DO 4900 V=1,(SN-1)
            IF (SURFTYPESPEC(V).EQ.'PZ') THEN
       IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
4900     CONTINUE
         IF (CURRENTSURFLABEL.EQ.0) THEN
            WATERREGIONBOTTOMSURF=SN
            SURFTYPESPEC(SN)='PZ'
            SURFVALUESPEC(SN)=CURRENTSURF
            SN=SN+1
         ELSE
            WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
         ENDIF
      ELSEIF (SPN.EQ.NUMOFSPACERS(DESNUM(COLUMN,ROW))) THEN
         SPACERTOPSURF=WATERREGIONBOTTOMSURF
         CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
c        SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
         CURRENTSURFLABEL=0
         DO 4910 V=1,(SN-1)
```

```
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 4910          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  SPACERBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  SPACERBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
               WATERREGIONTOPSURF=SPACERBOTTOMSURF
               WATERREGIONBOTTOMSURF=NODEBOTTOMSURF
            ENDIF
*     Write the current homogenized spacer region cell in this BPR universe.
            WRITE(30,4920) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c      (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)), GTORSURF,
     c      (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c      SPN
 4920       FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Homogenized region for spacer ',I2)
            LN=LN+1
*     Write the water region cell below the current homogenized spacer cell
in this BPR universe.
            WRITE(30,4930) LN, BMODML, (-1*MODDENSITY), GTORSURF,
     c      (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW)
 4930       FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Borated moderator')
            LN=LN+1
 4940    CONTINUE
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
```

```
*******************************************************************
*******************************************************************
*******************************************************************
*******************************************************************
*******************************************************************
        ELSEIF (WBPRA(BANKNUM(COLUMN,ROW)).EQ.3) THEN
        DO 5110 MCNPNODE=1,NUMOFBPRANODES(BANKNUM(COLUMN,ROW))
*     Define the upper end-fitting bottom surface.
              CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c        ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
              CURRENTSURFLABEL=0
              DO 4950 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
4950          CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 UEFBOTTOMSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 UEFBOTTOMSURF=CURRENTSURFLABEL
              ENDIF
*     Define the upper end-fitting top surface.
              CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c        ENDFITHEIGHT(DESNUM(COLUMN,ROW),1)+
     c        ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
              CURRENTSURFLABEL=0
              DO 4960 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
4960          CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 UEFTOPSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 UEFTOPSURF=CURRENTSURFLABEL
              ENDIF
*     Define the inner BPR cladding inner radius.
              CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),1)
              CURRENTSURFLABEL=0
              DO 4970 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
```

```
                 CURRENTSURFLABEL=V
                 EXIT
              ENDIF
           ENDIF
 4970   CONTINUE
        IF (CURRENTSURFLABEL.EQ.0) THEN
           BPICIRSURF=SN
           SURFTYPESPEC(SN)='CZ'
           SURFVALUESPEC(SN)=CURRENTSURF
           SN=SN+1
        ELSE
           BPICIRSURF=CURRENTSURFLABEL
        ENDIF
*     Define the inner BPR cladding outer radius.
        CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),2)
        CURRENTSURFLABEL=0
        DO 4980 V=1,(SN-1)
           IF (SURFTYPESPEC(V).EQ.'CZ') THEN
      IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                 CURRENTSURFLABEL=V
                 EXIT
              ENDIF
           ENDIF
 4980   CONTINUE
        IF (CURRENTSURFLABEL.EQ.0) THEN
           BPICORSURF=SN
           SURFTYPESPEC(SN)='CZ'
           SURFVALUESPEC(SN)=CURRENTSURF
           SN=SN+1
        ELSE
           BPICORSURF=CURRENTSURFLABEL
        ENDIF
*     Define the BP absorber inner radius.
        CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),3)
        CURRENTSURFLABEL=0
        DO 4990 V=1,(SN-1)
           IF (SURFTYPESPEC(V).EQ.'CZ') THEN
      IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                 CURRENTSURFLABEL=V
                 EXIT
              ENDIF
           ENDIF
 4990   CONTINUE
        IF (CURRENTSURFLABEL.EQ.0) THEN
           BPIRSURF=SN
           SURFTYPESPEC(SN)='CZ'
           SURFVALUESPEC(SN)=CURRENTSURF
           SN=SN+1
        ELSE
           BPIRSURF=CURRENTSURFLABEL
        ENDIF
*     Define the BP absorber outer radius.
        CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),4)
        CURRENTSURFLABEL=0
```

```
            DO 5000 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'CZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 5000       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               BPORSURF=SN
               SURFTYPESPEC(SN)='CZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               BPORSURF=CURRENTSURFLABEL
            ENDIF
*     Define the outer BPR cladding inner radius.
            CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),5)
            CURRENTSURFLABEL=0
            DO 5010 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'CZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 5010       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               BPOCIRSURF=SN
               SURFTYPESPEC(SN)='CZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               BPOCIRSURF=CURRENTSURFLABEL
            ENDIF
*     Define the outer BPR cladding outer radius.
            CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),6)
            CURRENTSURFLABEL=0
            DO 5020 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'CZ') THEN
         IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 5020       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               BPOCORSURF=SN
               SURFTYPESPEC(SN)='CZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               BPOCORSURF=CURRENTSURFLABEL
            ENDIF
```

```
*       Define the BP node bounding surfaces.
           IF (MCNPNODE.EQ.1) THEN
              TOTBPHEIGHT=0.0
              DO 5030 Z=1,NUMOFBPRANODES(BANKNUM(COLUMN,ROW))
                 TOTBPHEIGHT=TOTBPHEIGHT+
     c           MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),Z)
 5030         CONTINUE
              CURRENTSURF=BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))+
     c        TOTBPHEIGHT
              IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                 CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
              ENDIF
              CURRENTSURFLABEL=0
              DO 5040 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 5040         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 TOPBPNODETOPSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 TOPBPNODETOPSURF=CURRENTSURFLABEL
              ENDIF
              BPNODETOPSURF=TOPBPNODETOPSURF
              CURRENTSURF=SURFVALUESPEC(BPNODETOPSURF)-
     c        MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),MCNPNODE)
              IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                 CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
              ENDIF
              CURRENTSURFLABEL=0
              DO 5050 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 5050         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 BPNODEBOTTOMSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 BPNODEBOTTOMSURF=CURRENTSURFLABEL
              ENDIF
           ELSEIF (MCNPNODE.NE.1) THEN
              BPNODETOPSURF=BPNODEBOTTOMSURF
```

```
                 CURRENTSURF=SURFVALUESPEC(BPNODETOPSURF)-
     c           MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),MCNPNODE)
                 IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                    CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                 ENDIF
                 CURRENTSURFLABEL=0
                 DO 5070 V=1,(SN-1)
                    IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                          CURRENTSURFLABEL=V
                          EXIT
                       ENDIF
                    ENDIF
 5070            CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
                    BPNODEBOTTOMSURF=SN
                    SURFTYPESPEC(SN)='PZ'
                    SURFVALUESPEC(SN)=CURRENTSURF
                    SN=SN+1
                 ELSE
                    BPNODEBOTTOMSURF=CURRENTSURFLABEL
                 ENDIF
              ENDIF
           IF (SURFVALUESPEC(BPNODEBOTTOMSURF).LT.
     c     SURFVALUESPEC(UEFTOPSURF)) THEN
*         Write the BP node cells in this BPR universe.
           IF ((BPRABSNODE(BANKNUM(COLUMN,ROW),MCNPNODE).EQ.'Y').AND.
     c        (BPNONABSMAT(BANKNUM(COLUMN,ROW)).EQ.1)) THEN
              WRITE(30,5080) LN, BPNODEML,
     c        (-1*AL2O3DENSITY(BANKNUM(COLUMN,ROW))), (-1*BPORSURF),
     c        BPIRSURF, (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c        BPRAUNIV(COLUMN,ROW), MCNPNODE
 5080         FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c        1X,I4,' IMP:N=1 U=',I3,'    $ Burnable poison node ',I2)
              LN=LN+1
              BPNODEML=BPNODEML+1
           ELSEIF ((BPRABSNODE(BANKNUM(COLUMN,ROW),MCNPNODE).EQ.'Y').AND.
     c        (BPNONABSMAT(BANKNUM(COLUMN,ROW)).NE.1)) THEN
              WRITE(30,5090) LN, BPNODEML,
     c        (-1*NONBPMATDATA(BANKNUM(COLUMN,ROW),1)), (-1*BPORSURF),
     c        BPIRSURF, (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c        BPRAUNIV(COLUMN,ROW), MCNPNODE
 5090         FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c        1X,I4,' IMP:N=1 U=',I3,'    $ Burnable poison node ',I2)
              LN=LN+1
              BPNODEML=BPNODEML+1
           ELSE
              WRITE(30,5100) LN, BPNODEML,
     c        (-1*BPDENTOGO(COLUMN,ROW,MCNPNODE)), (-1*BPORSURF),
     c        BPIRSURF, (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c        BPRAUNIV(COLUMN,ROW), MCNPNODE
 5100         FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c        1X,I4,' IMP:N=1 U=',I3,'    $ Burnable poison node ',I2)
              LN=LN+1
```

```
              BPNODEML=BPNODEML+1
           ENDIF
           ENDIF
 5110       CONTINUE
*     Define the BPR cladding top surface.
                  CURRENTSURF=TOTBPHEIGHT+
     c            BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))+
     c            BPRPLEN(BANKNUM(COLUMN,ROW),1)
                  IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                     CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                  ENDIF
                  CURRENTSURFLABEL=0
                  DO 5120 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                       ENDIF
                     ENDIF
 5120             CONTINUE
                  IF (CURRENTSURFLABEL.EQ.0) THEN
                     BPCLADTOPSURF=SN
                     SURFTYPESPEC(SN)='PZ'
                     SURFVALUESPEC(SN)=CURRENTSURF
                     SN=SN+1
                  ELSE
                     BPCLADTOPSURF=CURRENTSURFLABEL
                  ENDIF
*     Define the BPR cladding bottom surface.
                  CURRENTSURF=BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))-
     c            BPRPLEN(BANKNUM(COLUMN,ROW),2)
                  CURRENTSURFLABEL=0
                  DO 5130 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                       ENDIF
                     ENDIF
 5130             CONTINUE
                  IF (CURRENTSURFLABEL.EQ.0) THEN
                     BPCLADBOTTOMSURF=SN
                     SURFTYPESPEC(SN)='PZ'
                     SURFVALUESPEC(SN)=CURRENTSURF
                     SN=SN+1
                  ELSE
                     BPCLADBOTTOMSURF=CURRENTSURFLABEL
                  ENDIF
*     Write the inner BP-to-cladding gap cell in this BPR universe.
              WRITE(30,5140) LN, (-1*BPIRSURF), BPICORSURF,
     c        (-1*TOPBPNODETOPSURF),
     c        BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 5140         FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,
```

```
     c      '   $ Burnable poison-to-cladding gap')
            LN=LN+1
*     Write the outer BP-to-cladding gap cell in this BPR universe.
            WRITE(30,5150) LN, (-1*BPOCIRSURF), BPORSURF,
     c      (-1*TOPBPNODETOPSURF),
     c      BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 5150       FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,
     c      '   $ Burnable poison-to-cladding gap')
            LN=LN+1
*     Write the annular water cell in this BPR universe.
            WRITE(30,5160) LN, BMODML,
     c      (-1*MODDENSITY), (-1*BPICIRSURF),
     c      (-1*TOPBPNODETOPSURF),
     c      BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 5160       FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,
     c      '   $ Burnable poison-to-cladding gap')
            LN=LN+1
*     Write the BPR cladding cell in this BPR universe.
*     Determine if the BPR cladding material specification has
*     previously been defined.  If it has been previously defined, determine
*     the cladding material specification label.
            CLADMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 5180 RO=1,(ROW-1)
                  DO 5170 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                        IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  BPRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
 5170             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 5180          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 5200 RO=ROW,ROW
                     DO 5190 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                           IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                     BPRCLADMAT(BANKNUM(CO,RO))) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
```

```
                        EXIT
                     ENDIF
                  ENDIF
                  ENDIF
 5190          CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
 5200       CONTINUE
         ENDIF
      ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
         DO 5220 RO=1,(ROW-1)
            DO 5210 CO=1,50
               IF (BANKNUM(CO,RO).NE.0) THEN
               IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                  IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c            BPRCLADMAT(BANKNUM(CO,RO))) THEN
                     CLADMLUNIQUE=.FALSE.
                     LEAVE=.TRUE.
                     BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                     EXIT
                  ENDIF
               ENDIF
               ENDIF
 5210       CONTINUE
            IF (LEAVE.EQ..TRUE.) THEN
               EXIT
            ENDIF
 5220    CONTINUE
      ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
         DO 5240 RO=1,1
            DO 5230 CO=1,(COLUMN-1)
               IF (BANKNUM(CO,RO).NE.0) THEN
               IF (BANKDES(BANKNUM(CO,RO)).NE.'BPRA ') THEN
                  IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c            BPRCLADMAT(BANKNUM(CO,RO))) THEN
                     CLADMLUNIQUE=.FALSE.
                     LEAVE=.TRUE.
                     BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                     EXIT
                  ENDIF
               ENDIF
               ENDIF
 5230       CONTINUE
            IF (LEAVE.EQ..TRUE.) THEN
               EXIT
            ENDIF
 5240    CONTINUE
      ENDIF
      IF (CLADMLUNIQUE.EQ..TRUE.) THEN
         BPCLADML(COLUMN,ROW)=MN
*  Check BPR Cladding Material
            IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
               DO 5250 C=1,2
```

```
               IF (C.EQ.1) THEN
                  WRITE(200,9300) BPCLADML(COLUMN,ROW)
               ELSEIF (C.EQ.2) THEN
                  WRITE(200,9301)
                  WRITE(200,7000)
                  WRITE(200,7001)
                  WRITE(200,7002)
                  WRITE(200,9302)
                  WRITE(200,7003)
                  WRITE(200,7004)
                  WRITE(200,7005)
                  WRITE(200,9303)
                  WRITE(200,9304)
               ENDIF
5250        CONTINUE
         ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW))
     c   .EQ.2) THEN
            DO 5260 C=1,2
               IF (C.EQ.1) THEN
                  WRITE(200,9305) BPCLADML(COLUMN,ROW)
               ELSEIF (C.EQ.2) THEN
                  WRITE(200,9306)
                  WRITE(200,9307)
                  WRITE(200,9308)
                  WRITE(200,9309)
                  WRITE(200,9310)
                  WRITE(200,7006)
                  WRITE(200,7007)
                  WRITE(200,7008)
                  WRITE(200,9311)
                  WRITE(200,9312)
                  WRITE(200,7009)
                  WRITE(200,7010)
                  WRITE(200,7011)
                  WRITE(200,9313)
                  WRITE(200,7012)
                  WRITE(200,7013)
                  WRITE(200,7014)
                  WRITE(200,7015)
               ENDIF
5260        CONTINUE
         ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW))
     c   .EQ.3) THEN
            DO 5270 C=1,2
               IF (C.EQ.1) THEN
                  WRITE(200,9314) BPCLADML(COLUMN,ROW)
               ELSEIF (C.EQ.2) THEN
                  WRITE(200,9315)
                  WRITE(200,9316)
                  WRITE(200,9317)
                  WRITE(200,9318)
                  WRITE(200,7016)
                  WRITE(200,7017)
                  WRITE(200,7018)
```

```
                          WRITE(200,9319)
                          WRITE(200,9320)
                          WRITE(200,7019)
                          WRITE(200,7020)
                          WRITE(200,7021)
                          WRITE(200,9321)
                          WRITE(200,7022)
                          WRITE(200,7023)
                          WRITE(200,7024)
                          WRITE(200,7025)
                          WRITE(200,9322)
                          WRITE(200,9323)
                          WRITE(200,9324)
                          WRITE(200,9325)
                          WRITE(200,9326)
                          WRITE(200,9327)
                          WRITE(200,7026)
                          WRITE(200,9328)
                          WRITE(200,9329)
                          WRITE(200,9330)
                        ENDIF
5270                  CONTINUE
                    ENDIF
                  MN=MN+1
                ENDIF
                IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
                  CLADRHO=6.56
                ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.2) THEN
                  CLADRHO=7.90
                ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.3) THEN
                  CLADRHO=8.19
                ENDIF
                WRITE(30,5280) LN, BPCLADML(COLUMN,ROW), (-1*CLADRHO),
    c           BPOCIRSURF,
    c           (-1*BPOCORSURF), (-1*BPCLADTOPSURF), BPCLADBOTTOMSURF,
    c           BPRAUNIV(COLUMN,ROW)
5280            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
    c           ' IMP:N=1 U=',I3,'   $ BPR cladding')
                LN=LN+1
                WRITE(30,5290) LN, BPCLADML(COLUMN,ROW), (-1*CLADRHO),
    c           BPICIRSURF,
    c           (-1*BPICORSURF), (-1*BPCLADTOPSURF), BPCLADBOTTOMSURF,
    c           BPRAUNIV(COLUMN,ROW)
5290            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
    c           ' IMP:N=1 U=',I3,'   $ BPR cladding')
                LN=LN+1
*     Write the BPR upper plenum cell in this BPR universe.
*     Determine if the BPR upper plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the upper plenum material specification label.
               BPRUPMLUNIQUE=.TRUE.
               LEAVE=.FALSE.
               IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                 DO 5310 RO=1,(ROW-1)
```

```
                    DO 5300 CO=1,50
                       IF (BANKNUM(CO,RO).NE.0) THEN
                          IF (BANKNUM(COLUMN,ROW).EQ.
     c                       BANKNUM(CO,RO)) THEN
                             BPRUPMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                             EXIT
                          ENDIF
                       ENDIF
 5300               CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
 5310            CONTINUE
                 IF (LEAVE.EQ..FALSE.) THEN
                    DO 5330 RO=ROW,ROW
                       DO 5320 CO=1,(COLUMN-1)
                          IF (BANKNUM(CO,RO).NE.0) THEN
                             IF (BANKNUM(COLUMN,ROW).EQ.
     c                          BANKNUM(CO,RO)) THEN
                                BPRUPMLUNIQUE=.FALSE.
                                LEAVE=.TRUE.
                                BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                                EXIT
                             ENDIF
                          ENDIF
 5320                  CONTINUE
                       IF (LEAVE.EQ..TRUE.) THEN
                          EXIT
                       ENDIF
 5330               CONTINUE
                 ENDIF
              ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                 DO 5350 RO=1,(ROW-1)
                    DO 5340 CO=1,50
                       IF (BANKNUM(CO,RO).NE.0) THEN
                          IF (BANKNUM(COLUMN,ROW).EQ.
     c                       BANKNUM(CO,RO)) THEN
                             BPRUPMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                             EXIT
                          ENDIF
                       ENDIF
 5340               CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
 5350            CONTINUE
              ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                 DO 5370 RO=1,1
                    DO 5360 CO=1,(COLUMN-1)
                       IF (BANKNUM(CO,RO).NE.0) THEN
```

```
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                           BPRUPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 5360             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 5370          CONTINUE
            ENDIF
         IF (SURFVALUESPEC(TOPBPNODETOPSURF).LT.
     c   SURFVALUESPEC(UEFTOPSURF)) THEN
            IF (BPRUPMLUNIQUE.EQ..TRUE.) THEN
               BPRUPML(COLUMN,ROW)=MN
*  Check Burnable Poison Rod Upper Plenum Regions
                  DO 5400 C=1,BPRUPLENMAT(BANKNUM(COLUMN,ROW),2)
                     IF (C.EQ.1) THEN
                        WRITE(200,5380) BPRUPML(COLUMN,ROW),
     c                  BPRUPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                  (-1*BPRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 5380                   FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                  '    $ Burnable Poison Rod Upper Plenum')
                     ELSE
                        WRITE(200,5390)
     c                  BPRUPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                  (-1*BPRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 5390                   FORMAT(T9,A9,3X,G14.6)
                     ENDIF
 5400             CONTINUE
               MN=MN+1
            ENDIF
            WRITE(30,5410) LN, BPRUPML(COLUMN,ROW),
     c      (-1*BPRUPLENMAT(BANKNUM(COLUMN,ROW),1)),
     c      TOPBPNODETOPSURF,
     c      (-1*BPCLADTOPSURF), (-1*BPCLADIRSURF),
     c      BPRAUNIV(COLUMN,ROW)
 5410       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ BPR upper plenum region')
            LN=LN+1
         ENDIF
*     Write the BPR lower plenum cell (lower end plug) in this BPR universe.
*     Determine if the BPR lower plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the lower plenum material specification label.
            BPRLPMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 5430 RO=1,(ROW-1)
                  DO 5420 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
```

```
                    IF (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO)) THEN
                       BPRLPMLUNIQUE=.FALSE.
                       LEAVE=.TRUE.
                       BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                       EXIT
                    ENDIF
                 ENDIF
 5420         CONTINUE
              IF (LEAVE.EQ..TRUE.) THEN
                 EXIT
              ENDIF
 5430      CONTINUE
           IF (LEAVE.EQ..FALSE.) THEN
              DO 5450 RO=ROW,ROW
                 DO 5440 CO=1,(COLUMN-1)
                    IF (BANKNUM(CO,RO).NE.0) THEN
                       IF (BANKNUM(COLUMN,ROW).EQ.
     c                 BANKNUM(CO,RO)) THEN
                          BPRLPMLUNIQUE=.FALSE.
                          LEAVE=.TRUE.
                          BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                          EXIT
                       ENDIF
                    ENDIF
 5440            CONTINUE
                 IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                 ENDIF
 5450         CONTINUE
           ENDIF
        ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
           DO 5470 RO=1,(ROW-1)
              DO 5460 CO=1,50
                 IF (BANKNUM(CO,RO).NE.0) THEN
                    IF (BANKNUM(COLUMN,ROW).EQ.
     c                 BANKNUM(CO,RO)) THEN
                       BPRLPMLUNIQUE=.FALSE.
                       LEAVE=.TRUE.
                       BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                       EXIT
                    ENDIF
                 ENDIF
 5460         CONTINUE
              IF (LEAVE.EQ..TRUE.) THEN
                 EXIT
              ENDIF
 5470      CONTINUE
        ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
           DO 5490 RO=1,1
              DO 5480 CO=1,(COLUMN-1)
                 IF (BANKNUM(CO,RO).NE.0) THEN
                    IF (BANKNUM(COLUMN,ROW).EQ.
     c                 BANKNUM(CO,RO)) THEN
                       BPRLPMLUNIQUE=.FALSE.
```

```
                        LEAVE=.TRUE.
                        BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
 5480          CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
 5490        CONTINUE
           ENDIF
           IF (BPRLPMLUNIQUE.EQ..TRUE.) THEN
             BPRLPML(COLUMN,ROW)=MN
*  Check Burnable Poison Rod Lower Plenum Regions
               DO 5520 C=1,BPRLPLENMAT(BANKNUM(COLUMN,ROW),2)
                  IF (C.EQ.1) THEN
                     WRITE(200,5500) BPRLPML(COLUMN,ROW),
    c                BPRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
    c                (-1*BPRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 5500                FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
    c                '   $ Burnable Poison Rod Lower Plenum')
                  ELSE
                     WRITE(200,5510)
    c                BPRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
    c                (-1*BPRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 5510                FORMAT(T9,A9,3X,G14.6)
                  ENDIF
 5520          CONTINUE
             MN=MN+1
           ENDIF
           WRITE(30,5530) LN, BPRLPML(COLUMN,ROW),
    c      (-1*BPRLPLENMAT(BANKNUM(COLUMN,ROW),1)), BPCLADBOTTOMSURF,
    c      (-1*BPNODEBOTTOMSURF), (-1*BPCLADIRSURF),
    c      BPRAUNIV(COLUMN,ROW)
 5530      FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
    c      ' IMP:N=1 U=',I3,'   $ BPR lower plenum region')
           LN=LN+1
*     Loop through the regions above the BPR (i.e. the appropriate upper core
regions)
*     Define the upper region lower surface.
             DO 5570 REGION=1,NUMREGABOVEBPRA
*     Determine the current upper region's lower surface specification.
               IF (REGION.EQ.1) THEN
                  REGIONTOPSURF=SYSTEMTOP
                  CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
    c             REGABOVEBPRA(REGION,1)
               ENDIF
               CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
    c          REGABOVEBPRA(REGION,1)
               IF (REGION.EQ.NUMREGABOVEBPRA) THEN
                  REGIONBOTTOMSURF=UEFTOPSURF
               ELSE
               CURRENTSURFLABEL=0
               DO 5540 V=1,(SN-1)
```

```
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                    ENDIF
                  ENDIF
 5540           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                  REGIONBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
                ELSE
                  REGIONBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
                ENDIF
*     Write the cell specification for the BPR universe upper region.
              IF (REGION.EQ.1) THEN
                WRITE(30,5550) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c          (-1*REGABOVEBPRA(REGION,2)),
     c          REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 5550           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                LN=LN+1
                REGIONTOPSURF=REGIONBOTTOMSURF
              ELSE
                WRITE(30,5560) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c          (-1*REGABOVEBPRA(REGION,2)), (-1*REGIONTOPSURF),
     c          REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 5560           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                LN=LN+1
                REGIONTOPSURF=REGIONBOTTOMSURF
              ENDIF
 5570         CONTINUE
*     Define the GT top surface.
              CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),3)
              IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
              ENDIF
              CURRENTSURFLABEL=0
              DO 5580 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                  ENDIF
                ENDIF
 5580         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                GTTOPSURF=SN
                SURFTYPESPEC(SN)='PZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
```

```
              ELSE
                 GTTOPSURF=CURRENTSURFLABEL
              ENDIF
*     Define the GT bottom surface.
              CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),4)
              CURRENTSURFLABEL=0
              DO 5590 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 5590         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 GTBOTSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 GTBOTSURF=CURRENTSURFLABEL
              ENDIF
*     Define the GT outer radius surface.
              CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),2)
              CURRENTSURFLABEL=0
              DO 5600 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'CZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 5600         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 GTORSURF=SN
                 SURFTYPESPEC(SN)='CZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 GTORSURF=CURRENTSURFLABEL
              ENDIF
*     Define the GT inner radius surface.
              CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),1)
              CURRENTSURFLABEL=0
              DO 5610 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'CZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 5610         CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 GTIRSURF=SN
```

```
                SURFTYPESPEC(SN)='CZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                GTIRSURF=CURRENTSURFLABEL
             ENDIF
*     Define the lower end-fitting top surface.
             CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
             CURRENTSURFLABEL=0
             DO 5620 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
 5620        CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                BPLEFTOPSURF=SN
                SURFTYPESPEC(SN)='PZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                BPLEFTOPSURF=CURRENTSURFLABEL
             ENDIF
*     Write the lower end-fitting cell specification for this BPR universe.
          IF (SURFVALUESPEC(GTBOTSURF).GE.
     c    ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)) THEN
             WRITE(30,5630) LN, FRLEFML(COLUMN,ROW),
     c       (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*BPLEFTOPSURF),
     c       BPRAUNIV(COLUMN,ROW)
 5630        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,' IMP:N=1 U=',I3,
     c       '    $ Assembly lower end-fitting')
             LN=LN+1
          ELSE
             WRITE(30,5640) LN, FRLEFML(COLUMN,ROW),
     c       (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*BPLEFTOPSURF),
     c       GTORSURF, BPRAUNIV(COLUMN,ROW)
 5640        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'    $ Assembly lower end-fitting')
             LN=LN+1
             WRITE(30,5650) LN, FRLEFML(COLUMN,ROW),
     c       (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*GTBOTSURF),
     c       (-1*GTORSURF), BPRAUNIV(COLUMN,ROW)
 5650        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'    $ Assembly lower end-fitting')
             LN=LN+1
          ENDIF
*     Write the upper end-fitting cell specification for this BPR universe.
          IF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c    SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c    (SURFVALUESPEC(GTTOPSURF).LE.
     c    SURFVALUESPEC(UEFBOTTOMSURF))) THEN
             WRITE(30,5660) LN, FRUEFML(COLUMN,ROW),
```

```
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), BPRAUNIV(COLUMN,ROW)
 5660           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).LE.
     c       SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                WRITE(30,5670) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), BPCLADORSURF, BPRAUNIV(COLUMN,ROW)
 5670           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,5680) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), BPCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*BPCLADORSURF),
     c          BPRAUNIV(COLUMN,ROW)
 5680           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c       SURFVALUESPEC(GTTOPSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                WRITE(30,5690) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
 5690           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,5700) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTORSURF),
     c          BPRAUNIV(COLUMN,ROW)
 5700           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,5710) LN, BMODML,
     c          (-1*MODDENSITY), BPCLADTOPSURF,
     c          (-1*GTTOPSURF), (-1*GTIRSURF),
     c          BPRAUNIV(COLUMN,ROW)
 5710           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c       SURFVALUESPEC(GTTOPSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).EQ.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
```

```
             WRITE(30,5720) LN, FRUEFML(COLUMN,ROW),
     c       (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c       (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
 5720        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
             LN=LN+1
          IF (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c    SURFVALUESPEC(GTTOPSURF)) THEN
             WRITE(30,5730) LN, BMODML,
     c       (-1*MODDENSITY), BPCLADTOPSURF,
     c       (-1*UEFTOPSURF), (-1*GTIRSURF),
     c       BPRAUNIV(COLUMN,ROW)
 5730        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
             LN=LN+1
          ENDIF
          ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c    SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c    (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c    SURFVALUESPEC(UEFTOPSURF)).AND.
     c    (SURFVALUESPEC(GTTOPSURF).GT.
     c    SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c    (SURFVALUESPEC(GTTOPSURF).LT.
     c    SURFVALUESPEC(BPCLADTOPSURF))) THEN
             WRITE(30,5740) LN, FRUEFML(COLUMN,ROW),
     c       (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c       (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
 5740        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
             LN=LN+1
             WRITE(30,5750) LN, FRUEFML(COLUMN,ROW),
     c       (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c       (-1*UEFTOPSURF), (-1*GTORSURF),
     c       BPCLADORSURF, BPRAUNIV(COLUMN,ROW)
 5750        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       1X,I4,' IMP:N=1 U=',I3,
     c       '    $ Assembly upper end-fitting')
             LN=LN+1
             WRITE(30,5760) LN, FRUEFML(COLUMN,ROW),
     c       (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), BPCLADTOPSURF,
     c       (-1*UEFTOPSURF), (-1*BPCLADORSURF),
     c       BPRAUNIV(COLUMN,ROW)
 5760        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
             LN=LN+1
          ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).EQ.
     c    SURFVALUESPEC(UEFTOPSURF)).AND.
     c    (SURFVALUESPEC(GTTOPSURF).GT.
     c    SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c    (SURFVALUESPEC(GTTOPSURF).LT.
     c    SURFVALUESPEC(BPCLADTOPSURF))) THEN
             WRITE(30,5770) LN, FRUEFML(COLUMN,ROW),
     c       (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c       (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
```

```
 5770        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
    c        ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
             LN=LN+1
             WRITE(30,5780) LN, FRUEFML(COLUMN,ROW),
    c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
    c        (-1*UEFTOPSURF), (-1*GTORSURF),
    c        BPCLADORSURF, BPRAUNIV(COLUMN,ROW)
 5780        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
    c        1X,I4,' IMP:N=1 U=',I3,
    c        '   $ Assembly upper end-fitting')
             LN=LN+1
          ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
    c     SURFVALUESPEC(UEFBOTTOMSURF)).AND.
    c     (SURFVALUESPEC(BPCLADTOPSURF).LT.
    c     SURFVALUESPEC(UEFTOPSURF)).AND.
    c     (SURFVALUESPEC(GTTOPSURF).EQ.
    c     SURFVALUESPEC(BPCLADTOPSURF))) THEN
             WRITE(30,5790) LN, FRUEFML(COLUMN,ROW),
    c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
    c        (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
 5790        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
    c        ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
             LN=LN+1
             WRITE(30,5800) LN, FRUEFML(COLUMN,ROW),
    c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
    c        (-1*UEFTOPSURF), (-1*GTORSURF),
    c        BPCLADORSURF, BPRAUNIV(COLUMN,ROW)
 5800        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
    c        1X,I4,' IMP:N=1 U=',I3,
    c        '   $ Assembly upper end-fitting')
             LN=LN+1
          ENDIF
*     Write the GT material cell in this BPR universe.
*     Determine if the GT material specification has
*     previously been defined.  If it has been previously defined, determine
*     the material specification label.
          CLADMLUNIQUE=.TRUE.
          LEAVE=.FALSE.
          IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
             DO 5820 RO=1,(ROW-1)
                DO 5810 CO=1,50
                   IF (DESNUM(CO,RO).NE.0) THEN
                      IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
    c                 GTMAT(DESNUM(CO,RO))) THEN
                         CLADMLUNIQUE=.FALSE.
                         LEAVE=.TRUE.
                         GTML(COLUMN,ROW)=GTML(CO,RO)
                         EXIT
                      ENDIF
                   ENDIF
 5810           CONTINUE
                IF (LEAVE.EQ..TRUE.) THEN
                   EXIT
                ENDIF
```

```
 5820        CONTINUE
             IF (LEAVE.EQ..FALSE.) THEN
                DO 5840 RO=ROW,ROW
                   DO 5830 CO=1,(COLUMN-1)
                      IF (DESNUM(CO,RO).NE.0) THEN
                         IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
    c                    GTMAT(DESNUM(CO,RO))) THEN
                            CLADMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            GTML(COLUMN,ROW)=GTML(CO,RO)
                            EXIT
                         ENDIF
                      ENDIF
 5830              CONTINUE
                   IF (LEAVE.EQ..TRUE.) THEN
                      EXIT
                   ENDIF
 5840           CONTINUE
             ENDIF
          ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
             DO 5860 RO=1,(ROW-1)
                DO 5850 CO=1,50
                   IF (DESNUM(CO,RO).NE.0) THEN
                      IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
    c                 GTMAT(DESNUM(CO,RO))) THEN
                         CLADMLUNIQUE=.FALSE.
                         LEAVE=.TRUE.
                         GTML(COLUMN,ROW)=GTML(CO,RO)
                         EXIT
                      ENDIF
                   ENDIF
 5850           CONTINUE
                IF (LEAVE.EQ..TRUE.) THEN
                   EXIT
                ENDIF
 5860        CONTINUE
          ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
             DO 5880 RO=1,1
                DO 5870 CO=1,(COLUMN-1)
                   IF (DESNUM(CO,RO).NE.0) THEN
                      IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
    c                 GTMAT(DESNUM(CO,RO))) THEN
                         CLADMLUNIQUE=.FALSE.
                         LEAVE=.TRUE.
                         GTML(COLUMN,ROW)=GTML(CO,RO)
                         EXIT
                      ENDIF
                   ENDIF
 5870           CONTINUE
                IF (LEAVE.EQ..TRUE.) THEN
                   EXIT
                ENDIF
 5880        CONTINUE
          ENDIF
```

```
            IF (CLADMLUNIQUE.EQ..TRUE.) THEN
               GTML(COLUMN,ROW)=MN
*  Check Guide Tube Material
                  IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.1) THEN
                     DO 5890 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9300) GTML(COLUMN,ROW)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9301)
                           WRITE(200,7000)
                           WRITE(200,7001)
                           WRITE(200,7002)
                           WRITE(200,9302)
                           WRITE(200,7003)
                           WRITE(200,7004)
                           WRITE(200,7005)
                           WRITE(200,9303)
                           WRITE(200,9304)
                        ENDIF
 5890                CONTINUE
                  ELSEIF (GTMAT(DESNUM(COLUMN,ROW))
     c            .EQ.2) THEN
                     DO 5900 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9305) GTML(COLUMN,ROW)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9306)
                           WRITE(200,9307)
                           WRITE(200,9308)
                           WRITE(200,9309)
                           WRITE(200,9310)
                           WRITE(200,7006)
                           WRITE(200,7007)
                           WRITE(200,7008)
                           WRITE(200,9311)
                           WRITE(200,9312)
                           WRITE(200,7009)
                           WRITE(200,7010)
                           WRITE(200,7011)
                           WRITE(200,9313)
                           WRITE(200,7012)
                           WRITE(200,7013)
                           WRITE(200,7014)
                           WRITE(200,7015)
                        ENDIF
 5900                CONTINUE
                  ELSEIF (GTMAT(DESNUM(COLUMN,ROW))
     c            .EQ.3) THEN
                     DO 5910 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9314) GTML(COLUMN,ROW)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9315)
                           WRITE(200,9316)
```

```
                                    WRITE(200,9317)
                                    WRITE(200,9318)
                                    WRITE(200,7016)
                                    WRITE(200,7017)
                                    WRITE(200,7018)
                                    WRITE(200,9319)
                                    WRITE(200,9320)
                                    WRITE(200,7019)
                                    WRITE(200,7020)
                                    WRITE(200,7021)
                                    WRITE(200,9321)
                                    WRITE(200,7022)
                                    WRITE(200,7023)
                                    WRITE(200,7024)
                                    WRITE(200,7025)
                                    WRITE(200,9322)
                                    WRITE(200,9323)
                                    WRITE(200,9324)
                                    WRITE(200,9325)
                                    WRITE(200,9326)
                                    WRITE(200,9327)
                                    WRITE(200,7026)
                                    WRITE(200,9328)
                                    WRITE(200,9329)
                                    WRITE(200,9330)
                                 ENDIF
5910                          CONTINUE
                           ENDIF
                        MN=MN+1
                     ENDIF
                     IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.1) THEN
                        CLADRHO=6.56
                     ELSEIF (GTMAT(DESNUM(COLUMN,ROW)).EQ.2) THEN
                        CLADRHO=7.90
                     ELSEIF (GTMAT(DESNUM(COLUMN,ROW)).EQ.3) THEN
                        CLADRHO=8.19
                     ENDIF
                     WRITE(30,5920) LN, GTML(COLUMN,ROW), (-1*CLADRHO),
     c               GTIRSURF,
     c               (-1*GTORSURF), (-1*GTTOPSURF), GTBOTSURF,
     c               BPRAUNIV(COLUMN,ROW)
5920                 FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c               ' IMP:N=1 U=',I3,'   $ Guide tube region')
                     LN=LN+1
*     Write the moderator cells within the GT in this BPR universe.
                     IF (SURFVALUESPEC(BPCLADTOPSURF).GE.
     c               SURFVALUESPEC(GTTOPSURF)) THEN
                        WRITE(30,5930) LN, BMODML, (-1*MODDENSITY),
     c                  (-1*GTIRSURF),
     c                  BPOCORSURF, (-1*GTTOPSURF), GTBOTSURF,
     c                  BPRAUNIV(COLUMN,ROW)
5930                    FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c                  ' IMP:N=1 U=',I3,
     c                  '   $ Borated moderator inside guide tube')
```

```
                LN=LN+1
             ELSEIF (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c       SURFVALUESPEC(GTTOPSURF)) THEN
                WRITE(30,5940) LN, BMODML, (-1*MODDENSITY),
     c          (-1*GTIRSURF),
     c          BPOCORSURF, (-1*BPCLADTOPSURF), GTBOTSURF,
     c          BPRAUNIV(COLUMN,ROW)
 5940           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '    $ Borated moderator inside guide tube')
                LN=LN+1
             ENDIF
             WRITE(30,5950) LN, BMODML, (-1*MODDENSITY),
     c       (-1*BPOCORSURF), (-1*BPCLADBOTTOMSURF), GTBOTSURF,
     c       BPRAUNIV(COLUMN,ROW)
 5950        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'    $ Borated moderator inside guide tube')
             LN=LN+1
*     Loop through the regions above the BPR (i.e. the appropriate upper core
regions)
*     Define the upper region lower surface.
               DO 5990 REGION=1,NUMREGABOVEBPRA
*     Determine the current upper region's lower surface specification.
                  IF (REGION.EQ.1) THEN
                     REGIONTOPSURF=SYSTEMTOP
                     CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
     c               REGABOVEBPRA(REGION,1)
                  ENDIF
                  CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
     c            REGABOVEBPRA(REGION,1)
                  IF (REGION.EQ.NUMREGABOVEBPRA) THEN
                     REGIONBOTTOMSURF=UEFTOPSURF
                  ELSE
                  CURRENTSURFLABEL=0
                  DO 5960 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                           CURRENTSURFLABEL=V
                           EXIT
                        ENDIF
                     ENDIF
 5960             CONTINUE
                  IF (CURRENTSURFLABEL.EQ.0) THEN
                     REGIONBOTTOMSURF=SN
                     SURFTYPESPEC(SN)='PZ'
                     SURFVALUESPEC(SN)=CURRENTSURF
                     SN=SN+1
                  ELSE
                     REGIONBOTTOMSURF=CURRENTSURFLABEL
                  ENDIF
                  ENDIF
*     Write the cell specification for the BPR universe upper region.
               IF (REGION.EQ.1) THEN
                  WRITE(30,5970) LN, FRUREGIONML(COLUMN,ROW,REGION),
```

```
     c            (-1*REGABOVEBPRA(REGION,2)),
     c            REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 5970             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
     c            ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                  LN=LN+1
                  REGIONTOPSURF=REGIONBOTTOMSURF
               ELSE
                  WRITE(30,5980) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c            (-1*REGABOVEBPRA(REGION,2)), (-1*REGIONTOPSURF),
     c            REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 5980             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c            ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                  LN=LN+1
                  REGIONTOPSURF=REGIONBOTTOMSURF
               ENDIF
 5990          CONTINUE
            SPACHEIGHT=0.0
*     Loop through the spacer and moderator regions along the axial
*     length of the BPR (from top to bottom).
            DO 6000 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
               SPACHEIGHT=SPACHEIGHT+SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
 6000       CONTINUE
            DO 6080 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
*     Define the homogenized spacer region bounding surfaces.
               IF (SPN.EQ.1) THEN
                  SPACERTOPSURF=UEFBOTTOMSURF
                  CURRENTSURF=SURFVALUESPEC(UEFBOTTOMSURF)-
     c            SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                  CURRENTSURFLABEL=0
                  DO 6010 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                           CURRENTSURFLABEL=V
                           EXIT
                        ENDIF
                     ENDIF
 6010             CONTINUE
                  IF (CURRENTSURFLABEL.EQ.0) THEN
                     SPACERBOTTOMSURF=SN
                     SURFTYPESPEC(SN)='PZ'
                     SURFVALUESPEC(SN)=CURRENTSURF
                     SN=SN+1
                  ELSE
                     SPACERBOTTOMSURF=CURRENTSURFLABEL
                  ENDIF
                  WATERREGIONTOPSURF=SPACERBOTTOMSURF
                  CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
                  CURRENTSURFLABEL=0
                  DO 6020 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                           CURRENTSURFLABEL=V
                           EXIT
                        ENDIF
```

```
              ENDIF
 6020       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
              WATERREGIONBOTTOMSURF=SN
              SURFTYPESPEC(SN)='PZ'
              SURFVALUESPEC(SN)=CURRENTSURF
              SN=SN+1
            ELSE
              WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
          ELSEIF ((SPN.NE.1).AND.(SPN.NE.
     c    NUMOFSPACERS(DESNUM(COLUMN,ROW)))) THEN
            SPACERTOPSURF=WATERREGIONBOTTOMSURF
            CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c      SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
            CURRENTSURFLABEL=0
            DO 6030 V=1,(SN-1)
              IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
                ENDIF
              ENDIF
 6030       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
              SPACERBOTTOMSURF=SN
              SURFTYPESPEC(SN)='PZ'
              SURFVALUESPEC(SN)=CURRENTSURF
              SN=SN+1
            ELSE
              SPACERBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
            WATERREGIONTOPSURF=SPACERBOTTOMSURF
            CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
            CURRENTSURFLABEL=0
            DO 6040 V=1,(SN-1)
              IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
                ENDIF
              ENDIF
 6040       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
              WATERREGIONBOTTOMSURF=SN
              SURFTYPESPEC(SN)='PZ'
              SURFVALUESPEC(SN)=CURRENTSURF
              SN=SN+1
            ELSE
              WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
          ELSEIF (SPN.EQ.NUMOFSPACERS(DESNUM(COLUMN,ROW))) THEN
            SPACERTOPSURF=WATERREGIONBOTTOMSURF
            CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
```

```
     c          SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                CURRENTSURFLABEL=0
                DO 6050 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 6050           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   SPACERBOTTOMSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   SPACERBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
                WATERREGIONTOPSURF=SPACERBOTTOMSURF
                WATERREGIONBOTTOMSURF=NODEBOTTOMSURF
             ENDIF
*     Write the current homogenized spacer region cell in this BPR universe.
             WRITE(30,6060) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
    c        (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)), GTORSURF,
    c        (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
    c        SPN
 6060        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
    c        ' IMP:N=1 U=',I3,'   $ Homogenized region for spacer ',I2)
             LN=LN+1
*     Write the water region cell below the current homogenized spacer cell
in this BPR universe.
             WRITE(30,6070) LN, BMODML, (-1*MODDENSITY), GTORSURF,
    c        (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
    c        BPRAUNIV(COLUMN,ROW)
 6070        FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
    c        ' IMP:N=1 U=',I3,'   $ Borated moderator')
             LN=LN+1
 6080     CONTINUE
        ENDIF
     ENDIF
 9300               FORMAT(T1,'M',I4,T9,' 8016.50c     -0.120',
    c                  '               $ Zirc-4 Cladding')
 9301                  FORMAT(T9,'24050.60c     -0.004')
 7000                  FORMAT(T9,'24052.60c     -0.084')
 7001                  FORMAT(T9,'24053.60c     -0.010')
 7002                  FORMAT(T9,'24054.60c     -0.002')
 9302                  FORMAT(T9,'26054.60c     -0.011')
 7003                  FORMAT(T9,'26056.60c     -0.184')
 7004                  FORMAT(T9,'26057.60c     -0.004')
 7005                  FORMAT(T9,'26058.60c     -0.001')
 9303                  FORMAT(T9,'40000.60c     -98.180')
 9304                  FORMAT(T9,'50000.35c     -1.400')
 9305               FORMAT(T1,'M',I4,T9,'6000.50c      -0.080',
    c                  '               $ SS304 Cladding')
```

```
9306          FORMAT(T9,'7014.50c        -0.100')
9307          FORMAT(T9,'14000.50c       -0.750')
9308          FORMAT(T9,'15031.50c       -0.045')
9309          FORMAT(T9,'16032.50c       -0.030')
9310          FORMAT(T9,'24050.60c       -0.793')
7006          FORMAT(T9,'24052.60c       -15.903')
7007          FORMAT(T9,'24053.60c       -1.838')
7008          FORMAT(T9,'24054.60c       -0.466')
9311          FORMAT(T9,'25055.50c       -2.000')
9312          FORMAT(T9,'26054.60c       -3.918')
7009          FORMAT(T9,'26056.60c       -63.156')
7010          FORMAT(T9,'26057.60c       -1.472')
7011          FORMAT(T9,'26058.60c       -0.200')
9313          FORMAT(T9,'28058.60c       -6.234')
7012          FORMAT(T9,'28060.60c       -2.465')
7013          FORMAT(T9,'28061.60c       -0.109')
7014          FORMAT(T9,'28062.60c       -0.350')
7015          FORMAT(T9,'28064.60c       -0.092')
9314      FORMAT(T1,'M',I4,T9,'6000.50c        -0.080',
     c        '                 $ Inconel Cladding')
9315          FORMAT(T9,'14000.50c       -0.350')
9316          FORMAT(T9,'15031.50c       -0.015')
9317          FORMAT(T9,'16032.50c       -0.015')
9318          FORMAT(T9,'24050.60c       -0.793')
7016          FORMAT(T9,'24052.60c       -15.903')
7017          FORMAT(T9,'24053.60c       -1.838')
7018          FORMAT(T9,'24054.60c       -0.466')
9319          FORMAT(T9,'25055.50c       -0.350')
9320          FORMAT(T9,'26054.60c       -0.958')
7019          FORMAT(T9,'26056.60c       -15.442')
7020          FORMAT(T9,'26057.60c       -0.360')
7021          FORMAT(T9,'26058.60c       -0.049')
9321          FORMAT(T9,'28058.60c       -35.382')
7022          FORMAT(T9,'28060.60c       -13.993')
7023          FORMAT(T9,'28061.60c       -0.616')
7024          FORMAT(T9,'28062.60c       -1.989')
7025          FORMAT(T9,'28064.60c       -0.520')
9322          FORMAT(T9,'5010.50c        -1.078E-3')
9323          FORMAT(T9,'5011.56c        -4.925E-3')
9324          FORMAT(T9,'13027.50c       -0.500')
9325          FORMAT(T9,'22000.50c       -0.900')
9326          FORMAT(T9,'27059.50c       -1.000')
9327          FORMAT(T9,'29063.60c       -0.205')
7026          FORMAT(T9,'29065.60c       -0.095')
9328          FORMAT(T9,'41093.50c       -2.563')
9329          FORMAT(T9,'42000.50c       -3.050')
9330          FORMAT(T9,'73181.50c       -2.563')

      RETURN
      END
```

This attachment contains the MACE input decks for the reactivity calculations for Sequoyah Unit 2. The input decks are contained on an attachment tape of this calculation file (the attachment tape has been moved to Reference 7.13). The information contained in this hard-copy representation of Attachment II is a listing of the various MACE input deck files and their attributes. The file sizes listed in the following table are the file sizes as they appear on the Hewlett Packard (HP) Series 9000 workstation. The HP file sizes differ from the file sizes on the attachment tape due to the difference in the block sizes between the HP and the personal computer. The tape containing Attachment II was written using the Colorado Model T1000e External Parallel Port Backup System for personal computers.

Filename	File Type	File Size (Bytes)	Date File Copied to Tape
seqi1a.txt	ASCII	54,675	5/20/98
seqi2a.txt	ASCII	79,255	5/20/98
seqi2b.txt	ASCII	79,255	5/20/98
seqi2c.txt	ASCII	79,255	5/20/98
seqi2d.txt	ASCII	79,255	5/20/98
seqi3a.txt	ASCII	79,256	5/20/98
seqi3b.txt	ASCII	79,256	5/20/98
seqi3c.txt	ASCII	79,256	5/20/98
seqi3d.txt	ASCII	79,256	5/20/98

This attachment contains the MCNP input decks for the reactivity calculations for Sequoyah Unit 2 that were generated by MACE. The input decks are contained on an attachment tape of this calculation file (the attachment tape has been moved to reference 7.13). The information contained in this hard-copy representation of Attachment III is a listing of the various MCNP input deck files and their attributes. The file sizes listed in the following table are the file sizes as they appear on the Hewlett Packard (HP) Series 9000 workstation. The HP file sizes differ from the file sizes on the attachment tape due to the difference in the block sizes between the HP and the personal computer. The tape containing Attachment III was written using the Colorado Model T1000e External Parallel Port Backup System for personal computers.

Filename	File Type	File Size (Bytes)	Date File Copied to Tape
seqi1a	ASCII	401,390	5/20/98
seqi2a	ASCII	1,472,183	5/20/98
seqi2b	ASCII	989,847	5/20/98
seqi2c	ASCII	814,199	5/20/98
seqi2d	ASCII	767,736	5/20/98
seqi3a	ASCII	1,458,355	5/20/98
seqi3b	ASCII	989,847	5/20/98
seqi3c	ASCII	814,199	5/20/98
seqi3d	ASCII	767,735	5/20/98

This attachment contains the MCNP output files for the reactivity calculations for Sequoyah Unit 2. The output files are contained on an attachment tape of this calculation file (the attachment tape has been moved to Reference 7.13). The information contained in this hard-copy representation of Attachment IV is a listing of the various MCNP output files and their attributes. The file sizes listed in the following table are the file sizes as they appear on the Hewlett Packard (HP) Series 9000 workstation. The HP file sizes differ from the file sizes on the attachment tape due to the difference in the block sizes between the HP and the personal computer. The tape containing Attachment IV was written using the Colorado Model T1000e External Parallel Port Backup System for personal computers.

Filename	File Type	File Size (Bytes)	Date File Copied to Tape
seqi1a.O	ASCII	6,290,005	5/20/98
seqi2a.O	ASCII	13,000,922	5/20/98
seqi2b.O	ASCII	10,041,780	5/20/98
seqi2c.O	ASCII	8,752,372	5/20/98
seqi2d.O	ASCII	8,690,897	5/20/98
seqi3a.O	ASCII	12,993,251	5/20/98
seqi3b.O	ASCII	10,157,479	5/20/98
seqi3c.O	ASCII	8,929,202	5/20/98
seqi3d.O	ASCII	8,699,472	5/20/98

This attachment contains the revised MACE input files, MACE generated MCNP input files, and MCNP output files for the reactivity calculations for Sequoyah Unit 2. The files are contained on an attachment tape of this calculation file (the attachment tape has been moved to Reference 7.13). The information contained in this hard-copy representation of Attachment V is a listing of the revised MACE input files, MACE generated MCNP input files, MCNP output files, and their attributes. The file sizes listed in the following table are the file sizes as they appear on the Hewlett Packard (HP) Series 9000 workstation. The HP file sizes differ from the file sizes on the attachment tape due to the difference in the block sizes between the HP and the personal computer. The tape containing Attachment V was written using the Colorado Model T1000e External Parallel Port Backup System for personal computers.The files are contained on an attachment tape of this calculation file. The information contained in this hard-copy representation of Attachment V is a listing of the various files and their attributes.

Filename	File Type	File Size (Bytes)	Date File Copied to Tape
seqi3a.txt	ASCII	79,260	5/20/98
seqi3b.txt	ASCII	79,260	5/20/98
seqi3c.txt	ASCII	79,260	5/20/98
seqi3d.txt	ASCII	79,260	5/20/98
seqi3a	ASCII	1,811,741	5/20/98
seqi3b	ASCII	1,136,693	5/20/98
seqi3c	ASCII	873,221	5/20/98
seqi3d	ASCII	803,526	5/20/98
seqi3a.O	ASCII	15,100,854	5/20/98
seqi3b.O	ASCII	10,779,440	5/20/98
seqi3c.O	ASCII	9,075,992	5/20/98
seqi3d.O	ASCII	8,628,306	5/20/98

This attachment contains the revised MCNP input files and MCNP output files for the Principle Isotope fuel material reactivity calculations for Sequoyah Unit 2. The files are contained on an attachment tape of this calculation file (the attachment tape has been moved to Reference 7.13). The information contained in this hard-copy representation of Attachment VI is a listing of the revised MCNP input files, MCNP output files, and their attributes. The file sizes listed in the following table are the file sizes as they appear on the Hewlett Packard (HP) Series 9000 workstation. The HP file sizes differ from the file sizes on the attachment tape due to the difference in the block sizes between the HP and the personal computer. The tape containing Attachment VI was written using the Colorado Model T1000e External Parallel Port Backup System for personal computers. The files are contained on an attachment tape of this calculation file. The information contained in this hard-copy representation of Attachment VI is a listing of the various files and their attributes.

Filename	File Type	File Size (Bytes)	Date File Copied to Tape
seqi2b	ASCII	989,847	5/20/98
seqi3b	ASCII	1,136,693	5/20/98
seqi2b.O	ASCII	10,101,506	5/20/98
seqi3b.O	ASCII	10,780,844	5/20/98