U.S.NRC

United States Nuclear Regulatory Commission

Protecting People and the Environment

NUREG/CR-6962
BNL-NUREG-80141-2008

Traditional Probabilistic Risk Assessment Methods for Digital Systems

Office of Nuclear Regulatory Research

AVAILABILITY OF REFERENCE MATERIALS IN NRC PUBLICATIONS

NRC Reference Material

As of November 1999, you may electronically access NUREG-series publications and other NRC records at NRC's Public Electronic Reading Room at http://www.nrc.gov/reading-rm.html. Publicly released records include, to name a few, NUREG-series publications; *Federal Register* notices; applicant, licensee, and vendor documents and correspondence; NRC correspondence and internal memoranda; bulletins and information notices; inspection and investigative reports; licensee event reports; and Commission papers and their attachments.

NRC publications in the NUREG series, NRC regulations, and *Title 10, Energy*, in the Code of *Federal Regulations* may also be purchased from one of these two sources.

1. The Superintendent of Documents
   U.S. Government Printing Office
   Mail Stop SSOP
   Washington, DC 20402–0001
   Internet: bookstore.gpo.gov
   Telephone: 202-512-1800
   Fax: 202-512-2250
2. The National Technical Information Service
   Springfield, VA 22161–0002
   www.ntis.gov
   1–800–553–6847 or, locally, 703–605–6000

A single copy of each NRC draft report for comment is available free, to the extent of supply, upon written request as follows:

Address: U.S. Nuclear Regulatory Commission
Office of Administration
Mail, Distribution and Messenger Team
Washington, DC 20555-0001
E-mail: DISTRIBUTION@nrc.gov
Facsimile: 301–415–2289

Some publications in the NUREG series that are posted at NRC's Web site address http://www.nrc.gov/reading-rm/doc-collections/nuregs are updated periodically and may differ from the last printed version. Although references to material found on a Web site bear the date the material was accessed, the material available on the date cited may subsequently be removed from the site.

Non-NRC Reference Material

Documents available from public and special technical libraries include all open literature items, such as books, journal articles, and transactions, *Federal Register* notices, Federal and State legislation, and congressional reports. Such documents as theses, dissertations, foreign reports and translations, and non-NRC conference proceedings may be purchased from their sponsoring organization.

Copies of industry codes and standards used in a substantive manner in the NRC regulatory process are maintained at—

The NRC Technical Library
Two White Flint North
11545 Rockville Pike
Rockville, MD 20852–2738

These standards are available in the library for reference use by the public. Codes and standards are usually copyrighted and may be purchased from the originating organization or, if they are American National Standards, from—

American National Standards Institute
11 West 42nd Street
New York, NY 10036–8002
www.ansi.org
212–642–4900

Legally binding regulatory requirements are stated only in laws; NRC regulations; licenses, including technical specifications; or orders, not in NUREG-series publications. The views expressed in contractor-prepared publications in this series are not necessarily those of the NRC.

The NUREG series comprises (1) technical and administrative reports and books prepared by the staff (NUREG–XXXX) or agency contractors (NUREG/CR–XXXX), (2) proceedings of conferences (NUREG/CP–XXXX), (3) reports resulting from international agreements (NUREG/IA–XXXX), (4) brochures (NUREG/BR–XXXX), and (5) compilations of legal decisions and orders of the Commission and Atomic and Safety Licensing Boards and of Directors' decisions under Section 2.206 of NRC's regulations (NUREG–0750).

DISCLAIMER: This report was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any employee, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use, or the results of such use, of any information, apparatus, product, or process disclosed in this publication, or represents that its use by such third party would not infringe privately owned rights.

U.S.NRC

United States Nuclear Regulatory Commission

Protecting People and the Environment

NUREG/CR-6962
BNL-NUREG-80141-2008

Traditional Probabilistic Risk Assessment Methods for Digital Systems

Manuscript Completed: May 2008
Date Published: October 2008

Prepared by
T.L. Chu, G. Martinez-Guridi, M. Yue, J. Lehner, and P. Samanta

Brookhaven National Laboratory
P.O. Box 5000
Upton, NY 11973

A. Kuritzky, NRC Project Manager

NRC Job Code N6413

Office of Nuclear Regulatory Research

ABSTRACT

At present, there are no consensus methods for quantifying the reliability of digital systems. The U.S. Nuclear Regulatory Commission (NRC) currently is undertaking assessments of the reliability of digital instrumentation and control (I&C) systems, using traditional and non-traditional (dynamic) methods in parallel. The NRC tasked Brookhaven National Laboratory (BNL) with conducting the research on the traditional methods. In general, these are methods that are well-established but they differ from dynamic methods in that they do not explicitly model the interactions between the plant system being modeled and the plant physical processes, nor the timing of these interactions.

The principal objective of the current project is to determine the capabilities and limitations of using traditional reliability modeling methods to develop and quantify digital system reliability models, with the desired goal of supporting the development of regulatory guidance for assessing risk evaluations involving digital systems. To accomplish this objective, the following tasks will be performed:

1. Develop desirable characteristics for reliability models of digital systems that could provide input to the technical basis for risk evaluations related to current and new reactors.

2. Select two traditional reliability methods and apply them to two example digital systems to determine the capabilities and limitations of these methods.

3. Compare the resulting digital system reliability models to the desirable characteristics to identify areas where additional research will improve the capabilities of the methods.

4. Develop a method, if necessary, for integrating the digital system reliability models into a nuclear power plant probabilistic risk assessment (PRA).

This report specifically addresses the development of the desirable characteristics and lays out the process by which the first reliability study of an example digital system will be performed. This work indicates that the traditional methods of Event Tree/Fault Tree and Markov modeling appear to be useful for the PRA of digital I&C systems, but also reveals limitations in the state-of-the-art for modeling digital systems using traditional PRA methods and where additional research and development are needed. The report offers other insights and conclusions obtained during this work and proposes activities to be conducted when applying these methods to the first reliability study. Note, in keeping with the principal objective stated above, this project will generally not involve advancements in the state-of-the-art, such as the estimation of risk from software faults.

FOREWORD

Nuclear power plants have traditionally relied on analog systems for their instrumentation and control (I&C) functions. With a shift in technology to digital systems as the result of analog obsolescence and digital functional advantages, existing plants have begun to replace some current analog I&C systems, while new plant designs fully incorporate digital systems.

The current licensing process for digital systems is based on deterministic criteria. In its 1995 Probabilistic Risk Assessment (PRA) Policy Statement, the U.S. Nuclear Regulatory Commission (NRC) encouraged the use of PRA technology in all regulatory matters to the extent supported by the state of the art in PRA methods and data. Though many activities are carried out in the life cycle of digital systems to ensure a high-quality product, there are no consensus methods at present for quantifying the reliability of these systems. This has been an impediment to developing a risk-informed analysis process for digital systems.

To address this limitation, the NRC is currently researching the use of both traditional PRA methods and dynamic methods for modeling digital systems. The desired goal of this research is to develop regulatory guidance for the use of risk information in regulatory decisions for new and operating reactors. This research is consistent with the recommendations from the 1997 National Research Council report on digital I&C in nuclear power plants and with the Commission staff requirements memorandum (M061108), dated December 6, 2006, which directs the staff to address deployment of digital systems, including the area of risk-informed digital I&C.

This NUREG/CR report documents the initial research into the use of traditional PRA methods for modeling and quantifying the reliability of digital I&C systems. The objectives of this initial research are to (1) determine the capabilities and limitations of using traditional reliability methods to develop and quantify digital system reliability models, (2) develop desirable characteristics for this kind of model, and (3) identify any state-of-the-art advancements needed to enhance the use of risk information associated with digital systems in regulatory decisions.

Christiana H. Lui, Director
Division of Risk Analysis
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission

TABLE OF CONTENTS

Section **Page**

ABSTRACT iii
FOREWORD v
LIST OF FIGURES xi
LIST OF TABLES xii
ACKNOWLEDGEMENTS xiii
ACRONYMS xiv

1. INTRODUCTION 1-1
1.1 Background 1-1
1.2 Objectives 1-2
1.3 Project Scope 1-2
1.4 Structure of the Report 1-4

2. DESIRABLE CHARACTERISTICS FOR EVALUATING PROBABILISTIC MODELS OF DIGITAL SYSTEMS 2-1
2.1 Level of Detail of the Probabilistic Model 2-2
2.2 Identification of Failure Modes of the Components of a Digital System 2-4
2.3 Modeling of Software Failures 2-6
2.4 Modeling of Dependencies 2-7
2.5 Probabilistic Data 2-13
2.6 Uncertainty 2-15
2.7 Integration of the Digital System Model with a PRA Model 2-16
2.8 Human Errors 2-17
2.9 Documentation and Results 2-18
2.10 Summary 2-18

3. OVERALL APPROACH OF MODELING 3-1
3.1 Operational Aspects and Risk Insights of the DFWCS 3-1
3.2 Major Steps to Building Models 3-2
3.2.1 Definition and Scope of the Probabilistic Model 3-2
3.2.2 Evaluating the Frequency of an Initiating Event 3-2
3.2.2.1 Markov Method 3-4
3.2.2.2 Fault Tree Method 3-4
3.2.3 Overview of Modeling Process 3-5

4. DESCRIPTION OF A DIGITAL FEEDWATER CONTROL SYSTEM 4-1
4.1 System Level Description 4-3
4.1.1 Control Modes and Algorithms 4-3
4.1.2 Deviations and Failover Operation 4-11
4.2 Description of Azonix μMAC 7000 Controllers and Fischer & Porter 53MC5000 Controllers 4-13
4.2.1 Azonix μMAC 7000 Microprocessors 4-13
4.2.2 Fischer & Porter (F&P) 53MC5 Controllers 4-15
4.3 Digital Valve Controller and Speed Controller 4-17
4.3.1 Digital Valve Controller 4-17
4.3.2 Lovejoy Speed Controller for FWP 4-19

TABLE OF CONTENTS
(Continued)

Section		Page
4.4	Dependencies and Interfaces	4-19
4.4.1	Interfaces with Operators	4-19
4.4.2	Interfaces between Two Digital Feedwater Control Systems	4-19
4.4.3	Interfaces between Main and Backup Microprocessors of the DFWCS	4-19
4.4.4	Interfaces between Main/Backup Microprocessor and M/A Controllers	4-20
4.4.5	Interfaces Among the M/A Controllers	4-20
4.4.6	Interfaces with Sensors, Valves, and Pumps	4-20
4.4.7	Power Supply	4-21
4.5	Digital Features	4-21
4.5.1	Microlink Communication Issues	4-21
4.5.2	Watchdog Timers	4-25
4.5.3	Software	4-27
4.5.4	Missing Module Diagnostics	4-28
4.5.5	Cyclic Redundancy Check	4-28
5.	FMEA OF A DIGITAL FEEDWATER CONTROL SYSTEM	5-1
5.1	Introduction	5-1
5.2	Scope and Levels of Detail of FMEA	5-2
5.2.1	System Level FMEA	5-2
5.2.2	Module Level FMEA	5-2
5.2.3	Major-Component-of-Module Level FMEA	5-3
5.3	FMEA Approach	5-3
5.4	Summary of FMEAs at Different Levels	5-5
5.4.1	FMEA at Top-Level of DFWCS	5-5
5.4.2	FMEA at Level of DFWCS Modules	5-5
5.4.3	FMEA at Level of Major-Component-of-Module of DFWCS Main CPU Module	5-6
5.5	Insights Learned from the FMEA	5-8
5.6	General Issues Associated with FMEA of Digital Systems	5-10
5.7	Concluding Remarks	5-11
6.	DEVELOPMENT OF A MARKOV MODEL OF DIGITAL FEEDWATER CONTROL SYSTEM	6-1
6.1	Development of Module-Level Markov Model	6-2
6.1.1	A Markov Model of the Main CPU	6-2
6.1.2	Development of the Markov Model of Other Modules	6-4
6.2	Development of a System-Level Markov Model	6-5
6.3	Description of How Software Failure Rates Fit in the Model	6-8
6.4	A Simplified Method for Building and Solving the Model	6-8
7.	DEVELOPMENT OF A FAULT TREE MODEL OF THE DIGITAL FEEDWATER CONTROL SYSTEM	7-1
7.1	Fault Tree Construction	7-1
7.1.1	Modeling of Failure to Control MFRV	7-3
7.1.2	Modeling of Failure to Control Three Components	7-6
7.2	Fault Tree Evaluation	7-7

TABLE OF CONTENTS
(Continued)

Section		Page
8.	DEVELOPMENT OF RELIABILITY PARAMETERS FOR DIGITAL SYSTEM RELIABILITY MODEL	8-1
8.1	Categories of Potential Data Sources for Digital Systems and Components	8-2
8.2	Issues in Digital System Data Analyses	8-3
8.3	Reliability Parameters Based on Databases Using Reliability Prediction Methods (RPMs) and Other Methods	8-4
8.3.1	Military Handbook 217F	8-5
8.3.2	Telcordia	8-5
8.3.3	PRISM Database	8-6
8.3.4	IEEE Standard 500-1984	8-8
8.3.5	IEC Standard 61508	8-8
8.3.6	Reliability Data for Safety instrumented Systems, PDS Data Handbook, 2006 Edition	8-9
8.4	Reliability Data Collection and Analysis for Digital Systems and Components from Industrial Operational Experience	8-10
8.4.1	Digital Core Protection Calculators of Combustion Engineering Reactor Protection System	8-10
8.4.2	Eagle-21 Channels of Westinghouse Reactor Protection System	8-11
8.4.3	Operating Experience of Digital Core Protection Calculators of CE RPS	8-13
8.4.4	Failure Experience of Programmable Logic Controllers Used in Emergency Shutdown Systems of Natural Gas Compression Stations	8-15
8.4.5	Operational Failure Experience of Fault-Tolerant Digital Control Systems in Different Industries	8-19
8.4.6	Failure Rates for Programmable Logic Controllers Used in Chemical and Nuclear Plants	8-21
8.4.7	Savannah River Site (SRS) Generic Data Development Based on Data from Different Industries	8-23
8.4.8	Failure Parameters of Digital Trip Module (DTM) and Trip Logic Unit (TLU) in ESBWR Probabilistic Risk Assessment (PRA)	8-24
8.4.9	Reliability Study of Digital Engineered Safety Feature Actuation System of Korean Standard Nuclear Power Plant	8-27
8.4.10	Digital RPS and ESFAS of AP600 Reactors	8-28
8.4.11	Digital Systems of AP1000 Reactors	8-29
8.4.12	Reliability Modeling of Digital Instrumentation and Control Systems for Nuclear Reactor Probabilistic Risk Assessments	8-29
8.5	Data Sources in Nuclear Industry	8-29
8.5.1	Licensee Event Report (LER) Database	8-29
8.5.2	Equipment Performance and Information Exchange (EPIX) Database	8-31
8.6	Summary of Failure Data Review	8-32
8.6.1	Categorization of Data Collection Levels	8-32
8.6.2	Summary of Reliability Parameters of Microprocessor or Microprocessor-Related Systems	8-33
8.7	Generic Failure Rate Estimate Using a Hierarchical Bayesian Method	8-38
8.7.1	Hierarchical Bayesian Models for Failure Rates Determination	8-41
8.7.2	Failure Rates Estimate of Digital Components Using the HBM	8-42

TABLE OF CONTENTS
(Continued)

Section **Page**

8.7.3 Sensitivity Analysis 8-46
8.7.4 HBM Analysis of Other Digital Components 8-48

9. MODELING TO ADDRESS DESIRABLE CHARACTERISTICS 9-1

10. SUMMARY AND CONCLUSIONS 10-1
10.1 Selection of Traditional Methods 10-1
10.2 Development of Desirable Characteristics for Evaluating Reliability Models of Digital Systems 10-2
10.3 Performance of an FMEA of the DFWCS 10-3
10.4 Modeling Approach 10-4
10.4.1 Development of Markov Model 10-4
10.4.2 Development of Fault Tree Model 10-5
10.5 Development of Failure Parameter Database 10-5
10.6 Next Steps 10-6
10.7 Recommendations for Research 10-7

11. REFERENCES 11-1

APPENDIX A SUMMARY REPORT OF THE EXTERNAL REVIEW PANEL MEETING ON RELIABILITY MODELING OF DIGITAL SYSTEMS (MAY 23-24, 2007) A-1

APPENDIX B DETAILED FMEA OF THE DFWCS AT DIFFERENT LEVELS B-1

APPENDIX C OTHER METHODS FOR MODELING DIGITAL SYSTEMS C-1

LIST OF FIGURES

Figure		Page
4-1	A Simplified Diagram of the Feedwater System	4-2
4-2	One of the Reactor Coolant Loops with Its Associated DFWCS	4-3
4-3	S/G Level Deviation Logic	4-12
4-4	PC Section - A Standard ISA Architecture	4-14
4-5	S/G Level Connections	4-16
4-6	Data Exchange between Device Controllers	4-23
4-7	A Generic Design of Watchdog Timer	4-26
5-1	Connection of Digital Inputs and Outputs	5-5
6-1	Components of Main CPU Module	6-2
7-1	Loss of Control of the Loop Associated with a DFWCS	7-2
7-2	Loss of Automatic Control	7-3
7-3	Simplified Diagram of the Control of an MFRV	7-4
7-4	Failure to Control MFRV	7-4
7-5	Incorrect Signal from MFV	7-5
7-6	Incorrect Signal from the CPUs	7-6
7-7	Failure to Control 3 Components	7-6

LIST OF TABLES

Table		Page
2-1	Summary of Desirable Characteristics	2-20
4-1	Analog Inputs/Outputs of the Microprocessors	4-5
4-2	Digital Inputs and Outputs of the Microprocessors	4-7
4-3	Inputs and Outputs of PB4R Digital Signal Isolator	4-9
4-4	Inputs and Outputs of MFV M/A Controller	4-9
4-5	Inputs and Outputs of BFV M/A Controller	4-10
4-6	Inputs and Outputs of FWP M/A Controller	4-10
4-7	Inputs and Outputs of PDI Controller	4-11
4-8	Microlink Communication Summary	4-24
5-1	Failure Modes of the Main CPU Module	5-8
6-1	Postulated Failure Modes of Some Other Modules	6-4
8-1	SINTEF Recommended Input Data for Single Safety System	8-10
8-2	Failure Rate of Digital Core Protection Calculator from NUREG/CR-5500, Volume 10	8-12
8-3	Failure Rate of Eagle-21 Channel Processor of Westinghouse RPS from NUREG/CR-5500, Volume 2	8-14
8-4	Failure Rates for Different Types of CPCS and/or CEACS Failures from Bickel [2006]	8-16
8-5	Failure Rate of PLCs Used in Emergency Shutdown Systems from Mitchell [1993]	8-18
8-6	Failure Data of Fault-Tolerant Digital Control Systems from Paula [1993a]	8-20
8-7	Digital System Component Failure Rates from OREDA-84 and Humphreys & Daniels	8-21
8-8	Digital System Component Failure Rates of a TMR System from Triconex and Humphreys & Daniels	8-21
8-9	Summary of PLC Failure Data from a U.S. Phenol Plant from Paula [1993b]	8-22
8-10	Summary of PLC Failure Data from French NPPs from Paula [1993b]	8-23
8-11	Failure Rate and Coverage of PLCs in Chemical and Nuclear Power Plants from Paula [1993b]	8-23
8-12	Savannah River Site (SRS) Generic Data from Blanton [1993]	8-25
8-13	Generic Component Data for a Korean DESFAS Reliability Analysis from Varde [2003]	8-28
8-14	Example Failure Parameters of DFWCS Components from Aldemir [2007]	8-30
8-15	Data Collection Level and Failure Parameters for a Microprocessor or a Microprocessor-Related System from Different Sources	8-34
8-16	CCF Data from Different Sources	8-39
8-17	Failure Records of a Digital Component Extracted from PRISM RACdata Database	8-44
8-18	Characteristics of Population Variability Distribution of a Digital Component Data	8-46
8-19	Error Factors Based on a Hierarchical Bayes Analysis	8-49

ACKNOWLEDGEMENTS

The work presented in this report includes work carried out during several years under several U.S. Nuclear Regulatory Commission (NRC) projects. Many other people helped in one way or another with this work, and we are grateful to them.

In particular, we are indebted to the current NRC Project Manager, Alan Kuritzky, for his technical and managerial support. We are also especially grateful to the external peer reviewers who commented on some early parts of the work, as well as those reviewers from the NRC, the nuclear industry, other Department of Energy national laboratories, and academia who reviewed the final draft of this report.

We also express our appreciation to Avril Woodhead for her editorial review of several revisions of the report, and to Jean Frejka and Nicole Kelly who put several versions of the report together and helped with the logistical aspects of the project.

ACRONYMS

A/D	Analog/Digital
A/M	Auto/Manual
ABWR	Advanced Boiling Water Reactor
AIC	Airborne, Inhibited, Cargo
ALWR	Advanced Light Water Reactor
API	Application Program Interface
ASIC	Application Specific Integrated Circuit
ASME	American Society of Mechanical Engineers
BFRV	Bypass Feedwater Regulating Valve
BFV	Bypass Feedwater Valve
BNL	Brookhaven National Laboratory
BWR	Boiling Water Reactor
B/U	Backup
CCA	Circuit Card Assembly
CCF	Common Cause Failure
CDF	Core Damage Frequency
CE	Combustion Engineering
CEA	Control Element Assembly
CEAC	Control Element Assembly Calculator
CFR	Code of Federal Regulations
CIP	Communication Interface Processor
CMM	Capability Maturity Model
CPC	Core Protection Calculator
CPCS	Core Protection Calculator System
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSMA	Carrier Sense Multiple Access
CSMA/BA	Carrier Sense Multiple Access/Bitwise Arbitration
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access/Collision Detect
D/A	Digital/Analog
DCS	Digital Control System
DD	Dangerous Detected
DEMUX	Demultiplexer
DESFAS	Digital Engineered Safety Feature Actuation System
DFWCS	Digital Feedwater Control System
DOD	Department of Defense
DPPS	Digital Plant Protection System
DRPS	Digital Reactor Protection System
DTM	Digital Trip Module
DU	Dangerous Undetected

ACRONYMS
(Continued)

EMI	Electromagnetic Interference
ENF	Expected Number of System Failures
EPIX	Equipment Performance and Information Exchange
EPRD	Electronic Parts Reliability Date
EPRI	Electric Power Research Institute
ESBWR	Economic Simplified Boiling Water Reactor
ESD	Emergency Shutdown
ESF	Engineered Safety Feature
ESFAS	Engineered Safety Features Actuation System
ET	Event Tree
F&P	Fischer & Porter
F-T	Fault-Tolerant
FIX	Function Index
FMEA	Failure Modes and Effects Analysis
FRV	Feedwater Regulating Valve
FT	Fault Tree
FWP	Feedwater Pump
FWS	Feedwater System
HART	Highway Addressable Remote Technology
HAZOP	Hazard and Operability Study
HBM	Hierarchical Bayesian Method
HIFT	Hardware Implemented Fault Tolerant
HRA	Human Reliability Analysis
HSI	Human-System Interface
HVAC	Heating, Ventilation, and Air-Conditioning
I/I	Current/Current
I/O	Input/Output
I/P	Current-to-Pressure
I&C	Instrumentation and Control
IDE	Integrated Drive Electronics
IE	Initiating Event
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IIS	In-Core Instrumentation System
INL	Idaho National Laboratory
ISA	Industry Standard Architecture
ISR	Interrupt Service Routine
KSNPP	Korean Standard Nuclear Power Plant

ACRONYMS
(Continued)

LER	License Event Report
LVDT	Linear Variable Differential Transformer
M/A	Manual/Automatic
MAC	Media Access Control
MB	Mega-Byte
MCC	Motor Control Center
MCMC	Markov Chain Monte Carlo
MCS	Minimal Cutsets
MFRV	Main Feedwater Regulating Valve
MFV	Main Feedwater Valve
MFW	Main Feedwater
MMI	Man-Machine Interface
MTP	Maintenance and Test Panel
MTTF	Mean Time To Failure
MUX	Multiplexer
NAM	Nuclear Asset Management
NASA	National Aeronautics and Space Administration
NEA	Nuclear Energy Agency
NMSS	Office of Nuclear Material Safety and Safeguards
NONC	Non-Critical Failures
NPP	Nuclear Power Plant
NPRDS	Nuclear Plant Reliability Data System
NRC	Nuclear Regulatory Commission
NRO	Office of New Reactors
OREDA	Offshore Reliability Data
PC	Personal Computer
PCB	Printed Circuit Board
PDI	Pressure Differential Indication
PDU	Plasma Display Unit
PI	Performance Indicator
PLC	Programmable Logic Controller
PLS	Plant Control System
PMS	Protection and Plant Safety System
PRA	Probabilistic Risk Assessment
PROM	Programmable Read Only Memory
PVC	Population Variation Curve
PWB	Printed Wiring Board
PWM	Pulse Width Modulation
PWR	Pressurized Water Reactor

ACRONYMS
(Continued)

RAC	Reliability Analysis Center
RADS	Reliability and Availability Data System
RAM	Random Access Memory
RCP	Reactor Coolant Pump
RES	Office of Nuclear Reactor Research
RFI	Radio Frequency Interference
ROM	Read Only Memory
RPM	Reliability Prediction Method
RPS	Reactor Protection System
RTCA	Radio Technical Commission for Aeronautics
S/G	Steam Generator
SCSS	Sequence Coding and Search System
SFC	Single Failure of Channel
SIFT	Software Implemented Fault Tolerant
SIL	Safety Integrity Level
SINTEF	Scientific and Industrial Research at the Norwegian Institute of Technology
SLC	Software Life Cycle
SMS	Special Monitoring System
SPAR	Standard Plant Analysis Risk
SRS	Savannah River Site
SSPI	Safety System Performance Indicator
SSPS	Solid State Protection System
STD	Spurious Trip Detected
STU	Spurious Trip Undetected
TLU	Trip Logic Unit
TMR	Triple Modular Redundancy
UCN	Ulchin Nuclear Power Plant
UMD	University of Maryland
URD	Utility Requirements Document
US	United States
VGA	Video Graphics Array
Vref	Voltage Reference
WANO	World Association of Nuclear Operations
WDT	Watchdog Timer

1. INTRODUCTION

1.1 Background

Nuclear power plants (NPPs) have traditionally relied upon analog systems for their monitoring, control, and protection functions. With a shift in technology to digital systems due to analog obsolescence and digital functional advantages, existing plants have begun to replace current analog systems while new plant designs fully incorporate digital systems. Since digital instrumentation and control (I&C) systems are expected to play an increasingly important role in nuclear power plant safety, the U.S. Nuclear Regulatory Commission (NRC) established a digital system research plan [NRC 2006] that defines a coherent set of research programs to support its regulatory needs.

The current licensing process for digital systems is based on deterministic engineering criteria. In its 1995 Probabilistic Risk Assessment (PRA) policy statement [NRC 1995], the Commission encouraged the use of PRA technology in all regulatory matters to the extent supported by the state-of-the-art in PRA methods and data. Though many activities have been completed in the area of risk-informed regulation, the risk-informed analysis process for digital systems has not yet been satisfactorily developed. Since, at present, there are no consensus methods for quantifying the reliability of digital systems, one of the programs included in the NRC digital system research plan addresses risk assessment methods and data for digital systems.

The objective of the NRC digital system risk research is to identify and develop methods, analytical tools, and regulatory guidance to support (1) using information on the risks of digital systems in NPP regulatory decisions, and (2) including models of digital systems into NPP PRAs. The NRC currently is undertaking assessments of the reliability of digital I&C systems, using traditional and non-traditional (dynamic) methods in parallel. For the purposes of this research, dynamic methods are defined as those that explicitly attempt to model (1) the interactions between a plant system and the plant's physical processes, i.e., the values of process variables, and (2) the timing of these interactions, i.e., the timing of the progress of accident sequences. Traditional methods are defined here as those that are well-established but that do not explicitly model the interactions between the plant system being modeled and the plant physical processes, nor the exact timing of these interactions. An example of this type of method is the traditional Event Tree/Fault Tree (ET/FT) approach.

In the past few years, Brookhaven National Laboratory (BNL) has been working on NRC projects to investigate methods and tools for probabilistic modeling of digital systems. The work included reviewing literature on digital system modeling [Chu 2004, Chu 2007], reviewing and analyzing operating experience of digital systems [Chu 2006], developing failure rate estimates using a Hierarchical Bayesian analysis [Yue 2006], and performing Failure Modes and Effects Analyses (FMEAs) of digital systems. The results of these reviews show that failures of digital systems caused several events that resulted in either a reactor trip or equipment unavailability at U.S. NPPs, and at least one event at a foreign NPP that resulted in a small loss of coolant accident during refueling [NEA 1998], as well as many significant events in other industries. This experience indicates that digital system failures have the potential to be contributors to plant risk. The NRC has now tasked BNL with conducting research on the use of traditional reliability modeling methods for digital I&C systems, which is the subject of this report. Information on the NRC research on the

use of dynamic reliability modeling methods for digital I&C systems can be found in NUREG/CR-6901 [Aldemir 2006] and NUREG/CR-6942 [Aldemir 2007].

1.2 Objectives

The principal objective of the current project is to determine the existing capabilities and limitations of using traditional reliability modeling methods to develop and quantify digital system reliability models, with the desired goal of supporting the development of regulatory guidance for assessing risk evaluations involving digital systems. To accomplish this objective, the following tasks will be performed:

1. Develop a set of desirable characteristics for reliability models of digital systems that could provide input to the technical basis for risk evaluations related to current and new reactors.

2. Select two traditional reliability methods and apply them to two example digital systems to determine the capabilities and limitations of these methods.

3. Compare the resulting digital system reliability models to the set of desirable characteristics to identify areas where additional research might improve the capabilities of the methods.

4. Develop a method, if necessary, for integrating the digital system reliability models into a NPP PRA.

This report specifically addresses the development of the set of desirable characteristics and lays out the process by which the first reliability study of an example digital system will be performed. Note, in keeping with the principal objective stated above, this process will generally not involve advancements in the state-of-the-art, such as the estimation of risk from software faults.

1.3 Project Scope

The development of the set of desirable characteristics and the comparison of several existing digital system reliability models to these characteristics were documented in a BNL preliminary letter report to the NRC. To more fully involve the technical community in this task, an external review panel was set up to review the findings documented in the letter report. The panel was comprised of six members, all of whom have expertise in modeling and quantifying digital system reliability, and also in probabilistic risk assessment. The panel met at BNL on May 23 and 24, 2007. The updated information from the letter report, as well as a summary report of the external review panel meeting, are included as part of this NUREG/CR.

As mentioned in the previous section, this project includes the application of traditional reliability modeling methods to example digital systems to support the development of tools and methods for including this type of model into PRAs. In determining which traditional methods to select for trial application, two factors were considered. First, because the ultimate goal of this project is to support the NRC in developing regulatory guidance for using risk information related to digital systems in the licensing actions of current or future NPPs, heavy emphasis was placed on those methods likely to be used by the nuclear industry. Secondly, many dynamic methods (i.e., methods that attempt to explicitly model the interactions between a plant system and the plant's physical

processes, and the timing of these interactions) were not considered because they are the subject of a parallel NRC research project.

Considering the above factors, the two traditional reliability modeling methods selected for trial application as part of this project are the traditional ET/FT method and the Markov method. The traditional ET/FT method has been commonly used by the U.S. nuclear power industry and in other countries and industries. The Markov method can be a powerful tool for analyzing digital systems because it can explicitly model system configurations arising from the ability of some digital systems to detect failures and change their configuration during operation. The Markov method can also explicitly treat failure and repair times. Further, the Markov method was used previously to model NPP systems that are commonly included in PRAs, as well as digital systems.

A number of other methods that may be useful for developing and quantifying reliability models of digital systems are discussed in an appendix to this report. While it is not practical to further explore all of these methods as part of the current project, some of them may warrant further attention if other studies demonstrate their capability and practicality.

As part of this project, the traditional ET/FT and Markov methods will be applied to two example systems (referred to as "benchmark" test cases). The first benchmark test case involves a digital feedwater control system (DFWCS) of a two-loop pressurized water reactor (PWR); the second involves a Reactor Protection System (RPS). Both a control system (which is typically non-safety-related) and a safety-related protection system were selected because these two types of systems may entail different modeling issues.

During this phase of the project, detailed information was only available for the first benchmark system (i.e., the DFWCS). Therefore, the DFWCS is used in this report to illustrate how the traditional reliability modeling methods will be applied in the later tasks of the project (i.e., in the actual benchmark studies). In order to delineate how the PRA models of the first benchmark system will be analyzed, constructed, and quantified using each of the two methods selected in the first task, the following activities were undertaken:

1. The DFWCS was analyzed in detail, including its function, components, associated controllers, dependencies and interfaces, and digital features, in order to gain a full understanding of the way the DFWCS and each of its relevant components operate.

2. The failure modes of the DFWCS components and the impact of each of them on the system function were determined by performing an FMEA.

3. The relevant failure modes of the components and their impacts on the DFWCS were used in developing approaches for constructing and quantifying probabilistic models using the traditional ET/FT and Markov methods.

4. Probabilistic parameters needed for quantifying the probabilistic models were investigated for each digital component failure mode.

The actual detailed construction and quantification of the two PRA models for each of the benchmark systems, as well as the integration of the digital system models into an overall PRA of a NPP, will be the subject of later tasks. As stated previously, in keeping with the principal objective of this project (i.e., to determine the *existing* capabilities and limitations of traditional reliability

modeling methods when applied to digital systems), performance of the benchmark studies will generally not involve advancements in the state-of-the-art. For example, the estimation of risk from software faults is outside the scope of this project because the methods to accomplish this are not considered to be mature yet. Technical areas that require such advancements are identified in this report.

The objective of this report is to describe approaches for developing reliability models of the DFWCS using the two selected traditional methods to address the set of desirable characteristics as far as the current state-of-the-art of these methods allows. A comparison of the models against the characteristics will be carried out when the task of developing the models is complete, and results will be presented in a subsequent report.

The methods and approaches in this report are applied to attempt to develop as complete a probabilistic model of a digital system as possible, given the current limitations of the state of the art. This maximizes the insights that may be gained about aspects of digital system models, even if some of these aspects are ultimately determined to not be significant or necessary.

1.4 Structure of the Report

As mentioned previously, this report specifically addresses the development of a set of desirable characteristics for reliability models of digital systems and illustrates the process by which the benchmark studies will be performed. The set of characteristics is presented in Chapter 2, and these characteristics reflect feedback from the external review panel meeting (Appendix A documents the discussions that took place at the meeting)[1].

Chapters 3 to 9 illustrate the process by which the two benchmark studies will be performed, using the DFWCS as an example. Chapter 3 presents the overall approach to modeling the DFWCS, Chapter 4 describes this system, and Chapter 5 discusses its FMEA. This information is used in Chapters 6 and 7 to describe how the Markov and fault tree models, respectively, will be developed for the DFWCS. Chapter 8 presents the probabilistic data for digital components that is planned to be used for quantifying these models. Chapter 9 discusses the way these models will address the desirable characteristics described in Chapter 2. Finally, Chapters 10 and 11 contain the conclusions and references, respectively.

FMEA tables for the DFWCS are presented in Appendix B. Appendix C provides brief information on some other methods that may be useful for developing and quantifying reliability models of digital systems, though they were not explored further as part of this project.

[1] Appendix A refers to draft "evaluation criteria" for reliability models of digital systems. This nomenclature has since been changed to "desirable characteristics" of digital system reliability models.

2. DESIRABLE CHARACTERISTICS FOR PROBABILISTIC MODELS OF DIGITAL SYSTEMS

A probabilistic model of a system and its associated probabilistic data should adequately account for the design features of the system that could affect its reliability, and hence, contribute to plant risk. Thus, the goal of this chapter is to define those characteristics of a model of digital systems that reflect these features. To meet this goal, a draft set of desirable characteristics was generated for digital system reliability models that are based on general experience with probabilistic risk assessments (PRAs), and on the particular considerations for digital system models. The following sources were used for developing the characteristics:

1. A literature review on modeling methods and failure databases of digital systems, carried out under a previous task of this project, which identified reports and white papers, and is summarized in a conference paper [Chu 2004].

2. A review of software failure experience and hardware failure data, made under a previous activity of this project, which led to development of a model of software failures and a basis for modeling of software failures, along with a hardware failure database capturing the variability of different data sources (Chapter 8 summarizes the work on hardware failures). The concept of software failure is discussed in Section 2.3.

3. The development of Failure Modes and Effects Analyses (FMEAs) of the Triconex Tricon platform and the digital feedwater control system (DFWCS) of a Combustion Engineering nuclear power plant (NPP) performed under previous activities related to this project. Chapter 5 presents the FMEA of the DFWCS.

4. The knowledge and experience of the study team from developing and reviewing PRA models of NPPs.

The desirable characteristics are grouped into the following nine broad categories which cover the probabilistic model of a digital system and its documentation:

1. Level of detail of the probabilistic model,
2. Identification of the failure modes of the components of a digital system,
3. Modeling of software failures,
4. Modeling of dependencies,
5. Probabilistic data,
6. Uncertainty,
7. Ease of integration with a PRA model,
8. Human errors, and
9. Documentation and results.

For each category, background information is first provided, and then the related desirable characteristics are presented. The focus of the characteristics here is on the design features of digital systems. The PRA model would be expected to meet the general guidelines provided in documents such as the PRA Procedures Guide [Hickman 1983] and the American Society of Mechanical Engineers (ASME) standard for PRA for NPP applications [ASME 2005].

The desirable characteristics are potentially relevant to any kind of probabilistic model of a digital system. There are some characteristics for which methods and/or data may not be

currently available. The characteristics establish the important features of a probabilistic model of a digital system, but they do not specify how to achieve this goal (i.e., they do not specify methods for modeling these features). Furthermore, it is debatable whether some of the characteristics are relevant. The intent was to include all characteristics addressing design aspects that are potential contributors to system unreliability and plant risk. Some characteristics may be modified later using the findings from the benchmark studies.

2.1 Level of Detail of the Probabilistic Model

In its most basic form, a probabilistic model of a system (analog or digital) is a combination of a "logic model" and probabilistic data. The logic model describes the relationship of relevant failures of the components of the system leading to some undesired event, such as the failure of the system to respond adequately to a demand, while the data are some parameters of each failure, such as its failure rate.

In general, a system's logic model evolves by breaking down the failures of its major components, such as the channels of a system, into the individual failures of their constituent components. This refinement from failures of major components into failures of basic components is continued to a level of detail that the analyst considers appropriate. Rouvroye and Brombacher [1999] illustrated the difficulty in modeling digital systems. They employed different models from an early version of International Electrotechnical Commission (IEC) 61508 [IEC 61508] and other methods to determine the average probability of failure on demand of an example system. They demonstrated that the results obtained with different methods are significantly different, i.e., the failure probabilities can differ by more than an order-of-magnitude. The authors did not detail their analysis, e.g., how the methods were applied and whether or not consistent levels of detail and failure data were used. Nevertheless, these results are not surprising. Different methods model an example system at different levels of detail introducing different approximations/errors. Also, potential inconsistencies in the failure parameters used at different levels of modeling could introduce significant variations in the results.

The level of detail of the most basic failures of the model is driven by two considerations, i.e., the objective of the modeling and the availability of probabilistic data:

1. The objective of modeling the system. Modeling digital systems in a PRA is intended to support risk evaluations, particularly with regard to the digital systems themselves. Hence, the desirable characteristics in modeling, such as level of detail and quality, depend on the types of decisions to be made. In general, the decision could be about determining the acceptability of a system or component or making changes to it. For replacing an analog reactor protection system (RPS) with a digital one, it is desirable to show that the new system is no less reliable than the old one. This can be demonstrated by comparing their probabilities of failure on demand; that is, if realistic data reflecting the specific design and operating condition are available at the system level and the new system does not introduce any new dependencies, a simple analysis of these data may be adequate to demonstrate the objective.

   On the other hand, a decision may be required about an issue at a lower level of detail, e.g., eliminating the redundancy within the individual channels of a four channel system. Then, the level of modeling detail must be sensitive enough to explicitly address any difference. If a decision on selecting the protocol of a communication network is to be made, then a model able to differentiate between protocols has to be developed. Also,

different cyclic redundancy check (CRC) designs have different capabilities of detecting and correcting faults [Siewiorek 1992]. Therefore, to assess the impacts of the design difference, a model at the right level of detail must be formulated.

The level of detail of a reliability model of a system (analog or digital) should capture its design features that could affect reliability. This desirable characteristic is particularly difficult for digital systems because if a single bit is askew, the whole system may collapse, as on June 28, 1999, when a stuck-at-one fault on a data line of the Traffic Collision Avoidance System of a Korean Air Cargo flight contributed to a near-miss collision with British Air Flight 027. However, it may not be feasible to model individual bits. The appropriate lowest level of detail of a reliability model of a digital system may be modeling a microprocessor as a component because the execution of software is based on the processor and because the communications between the microprocessors and between the microprocessors and other components of the digital system can be modeled. Another possibility of capturing the details of the design of a digital system is using applicable failure data at a higher level, so that the design is implicitly included, as long as the data realistically reflects all the failure modes of the specific design, the model adequately supports the study's objective, and dependencies are accounted for.

2. The availability of probabilistic data. The process of refinement of a system in a typical PRA from failures of major components into failures of basic components is considered acceptable when it stops at the level of the basic components for which there are probabilistic data available. Accordingly, in general there is a close relationship between the level of detail of the logic model of a system (also called level of refinement or level of resolution) and its associated data. The same applies to digital systems. For example, a microcontroller consists of a microprocessor, its associated memory, and Input/Output (I/O) interfaces. If there are data at the microcontroller level, then the associated memory and I/O interfaces do not have to be explicitly modeled; that is, if the internal structure of the microcontroller is built in the data, then the microcontroller can be modeled as a black box, provided that its interactions with the other components can be modeled correctly.

   Due to the current scarcity of publicly available probabilistic data for digital system components, in order to capture all the design features of a digital system that could affect its reliability, it may be necessary to model to a level of detail for which there may not be data currently available. In this case, sensitivity studies may be warranted to address this shortcoming.

Desirable Characteristic

1.1 A reliability model of a digital system is developed to a level of detail that captures the design features affecting the system's reliability, provides the output needed for risk evaluations, and for which probabilistic data are available.

2.2 Identification of Failure Modes of the Components of a Digital System

In a probabilistic model, the effects of failure modes of components on the digital system and on the overall NPP are explicitly represented. To this end, it is first necessary to define the failure modes of the components of the digital system. Typical methods to identify failure modes for analog systems are the FMEA and the Hazard and Operability (HAZOP) analysis. Usually, the FMEA is carried out by successively analyzing the system at deeper levels. In other words, the system is first analyzed at a top-level, i.e., the entire system, and then failure modes of major components of the system, such as its channels, are postulated and evaluated. Subsequently, the failure modes of the components of each channel are analyzed. As discussed in Section 2.1 on "Level of Detail of the Probabilistic Model," this refinement is continued to the level of detail considered adequate for the objective of the model.

It is desirable to correctly model dependencies of digital systems. For example, synchronization, voting, and data communication are physical interactions between processors and redundant channels, and potentially can introduce dependent failures. Deterministic evaluations are desirable to help identify such dependencies. FMEA and HAZOP analysis can be used to identify the different failure modes of the components of the system and their effects, which can then be used to identify potential dependencies and decide how to model the system. Independence assessment, a requirement of the Institute of Electrical and Electronics Engineers, Inc. (IEEE) 7-4.3.2-2003, can be used to verify the assumptions used for modeling. The failure modes and dependencies identified in this way can be used as input when addressing the desirable characteristics on "Modeling of Dependencies."

Functional failure modes are those that involve loss or deterioration of the function of a component or system, such as loss of the ability of a data bus to transmit data. Physical failure modes are those that involve loss or deterioration of the physical characteristics of a component, such as a data bus that is physically broken. Only functional failure modes are included in the probabilistic model.

Experience in carrying out FMEAs of the Triconex Tricon Platform and the DFWCS of an NPP indicates the following:

1. The quality of an FMEA depends on how carefully it is carried out, the level of detail of the analysis, the availability of comprehensive information, the qualification and experience of the team of analysts, and the resource limitations. For example, detailed failure modes and their effects are needed to properly capture failures that might affect the system's reliability, such as those associated with data communication between redundant channels and synchronization and voting of redundant processors. These unique features of digital systems represent physical interactions between redundant channels and have the potential to introduce dependent failures. It is, therefore, desirable to undertake supporting analyses to verify the satisfaction of the independence requirement of IEEE 7-4.3.2-2003. Such analysis probably should be a part of the deterministic evaluation of digital systems, which might well confirm that no dependent failures are introduced. Otherwise, probabilistic modeling would have to be detailed enough to capture the dependencies. Depending on who is performing the FMEA, these supporting analyses, or the information needed to perform them, may not be readily available (e.g., proprietary).

2. The DFWCS has a very complex design. Each of its components, such as each central processing unit and controller, has embedded software and many inputs and outputs that are interconnected between the system's components. In general, when carrying out an FMEA of a digital system which has a complex design such as the DFWCS, it is difficult to predict the response and effects of every individual failure based solely on studying the reports documenting the system design. If several failures are analyzed simultaneously, the analysis becomes even more difficult. Accordingly, an important insight regarding FMEA of a digital system when carried out by just studying its associated reports is that it is an excellent tool for learning about and understanding system design and operation, and some possible safety weaknesses. On the other hand, undertaking an FMEA in this way is not a sufficient tool to determine how specific component-level failure modes affect a large complicated digital system. Hence, it is advisable to support it with more sophisticated approaches to explore the interactions between the components of a digital system and the effects of one or more failures. One possible approach is the use of a simulation tool that reflects the operation of the system, including the execution of the application software. Ideally, the FMEA and these supporting tools would be used in combination to reliably identify the vulnerabilities of the system.

Based on a review of software failure experience from different industries carried out in a previous activity related to this project, it was recognized that it is difficult to define software failure modes because they occur in many different ways depending on specific applications. In reviewing papers on software FMEA, it became apparent that different ways of defining failure modes, causes, and effects were proposed, and they typically suffer from shortcomings. For example, failure modes, causes, or effects frequently are mixed up or defined ambiguously, and sometimes they overlap or are even contradictory. In attempting to address these problems with the current software failure categorization methods, a categorization framework was developed that involves defining generic failure modes and failure causes. The generic failure modes can support software FMEAs by affording some examples of potential failure modes.

Operational experience revealed that many digital systems fail due to incorrect requirements. For example, a review of software failures in domestic NPPs from January 1, 1996 to December 31, 2005 shows that the most predominant cause of failure, accounting for 36% of them, is incorrect "Software requirements analysis". Incorrect design requirements may be due to vagueness or ambiguity in their description, incompleteness, and/or inconsistencies. Even if the design requirements are correct, there also may be a failure to correctly implement them into the software. In general, the issues related to design requirements also are applicable to hardware, i.e., to the entire digital system. For example, there may be inconsistencies in the requirements of the interactions between the hardware and the software. It is desirable that the probabilistic model of the system accounts for this important type of failure.

Desirable characteristics

2.1 *A method is applied for identifying failure modes of the basic components of the digital system and their impact on the system. This method provides a systematic way of carrying out this identification such that there is confidence that the failure modes obtained are as complete as possible.*

2.2 *Supporting analyses are carried out to determine how specific features of a design, such as communication, voting, and synchronization, could affect system operation. These*

analyses determine whether the specific design features could introduce dependent failures that should be modeled.

2.3 *Failure modes that have occurred in the operating experience are examined and their applicability to the digital system being studied is considered.*

2.4 *The probabilistic model of the digital system accounts for the possibility that the system may fail due to incorrect design requirements, or due to correct requirements that are not correctly implemented into the system.*

2.3 Modeling of Software Failures

The most unique characteristic of a digital system distinguishing it from an analog one is that it contains software. Software failures have caused digital system failure and have resulted in serious accidents involving airplanes. The review of software failures in the nuclear and non-nuclear industries confirms the significant impact that these failures can have on safety.

Software is developed in several stages that transform it from a concept into a code that is executed by a computer processor. During this development, sometimes called the software life cycle (SLC), faults may be (unintentionally) introduced. Software failure occurs from the combination of a fault and the specific set of conditions (i.e., a set of input data) that trigger it.

The failure mechanisms of hardware and software differ. Hardware fails due to factors such as wear out, while software fails because of the presence of a fault and the occurrence of a specific set of input data. Accordingly, software failures should not be considered to be included in hardware failures, or in the probabilistic data of hardware. Rather, it is desirable to explicitly include software failures in the logic model, such that their contribution to the reliability of the associated system and to the core damage frequency (CDF) (and other figures of merit) is properly accounted for.

The theoretical basis for explicitly modeling software failures was developed by considering triggering events; that is, the occurrence of software failure results from the occurrence of an input that triggers a fault in the software. The input to the software is data which changes randomly according to the plant's conditions. Since software failure occurs due to the combination of a fault in the software and a specific set of input data (which is random), it can be modeled in a probabilistic way in terms of failure rates and failure probabilities. The selection of a failure rate versus a failure probability should be consistent with the function of the system being modeled.

Ideally, a software reliability model also considers the context/boundary condition in which software is used. For example, it is desirable that the modeling of the actuation of an RPS accounts for the difference in the reactor trip's actuation logic between a reactivity accident and a loss of feedwater transient; and that the modeling of Engineered Safety Features Actuation System (ESFAS) actuation signals differs for different frontline systems, depending on the type of accidents being mitigated. However, if modeling at this level of precision is not practical, conservative simplifying assumptions may be needed.

Another relevant consideration for modeling software failures is that the software that performs the functions of the digital system usually is of two types: the "application software" that actually carries out these functions, and some type of "support software" for the application software.

The support software often is an operating system that provides some basic functions for the application software, such as the ability to communicate with the hardware of the system. Platform software, another kind of support software, is that developed by the hardware manufacturers. It is desirable that the scope of the probabilistic model of the software includes both types of software because either type may fail, and the effects may be serious, regardless of which type fails.

Desirable characteristics

3.1 Software failures are accounted for in the probabilistic model.

3.2 Modeling of software failures is consistent with the basis of how they occur, that is, software failures happen when triggering events occur.

3.3 Modeling of software failures accounts for the context/boundary condition in which a software is used.

3.4 The model of the software includes the "application software" and the "support software."

2.4 Modeling of Dependencies

An important requirement of a PRA model is that all types of dependencies are correctly modeled and included in the logic model. This is particularly important for digital systems, whose unique features can result in different types of dependencies. The dependencies associated with digital systems are binned into the following groups:

- Dependencies related to communication
- Dependencies related to support systems
- Dependencies related to sharing of hardware
- Modeling of fault-tolerance features
- Dependencies related to Type I and II interactions
- Dependencies related to common cause failures

The desirable characteristics of a reliability model for addressing each group of dependencies are discussed below.

<u>Dependencies Related to Communication</u>

Components of digital systems communicate through buses, hardwired connections, and networks. It is desirable that the propagation of failures through communication devices and their effects on the related components or systems are evaluated, and any effect considered relevant is included in the probabilistic model. Three cases of failure propagation are considered:

1. Inter-system failure propagation. A digital system may be receiving data from other systems, and also sending data to other systems in a NPP. Hence, failures may propagate from one system to another.

2. Inter-channel failure propagation. A digital system may contain several channels, each having redundant components. If redundant channels are connected through hardwiring or a communication network, a failure in one channel might propagate to another channel via the connection. It is advisable that such a possibility be analyzed, and if failure propagation cannot be ruled out, then this propagation is modeled.

3. Intra-channel failure propagation. When redundancy within a channel is used, it is desirable that the potential for transmitting incorrect signals between the subsystems inside it is considered in the same way as treating the communication of incorrect signals between channels.

Desirable characteristics

4.1.1 Inter-system failure propagation is addressed, and modeled as applicable.

4.1.2 Inter-channel failure propagation is addressed, and modeled as applicable.

4.1.3 Intra-channel failure propagation is addressed, and modeled as applicable.

Dependencies Related to Support Systems

Digital systems depend on AC or DC power, and may also depend on Heating, Ventilation, and Air Conditioning (HVAC) for room cooling. It is desirable that the probabilistic model encompasses all the relevant dependencies of a digital system on its support systems.

Desirable characteristics

4.2.1 Loss of power to safety-related digital systems is modeled. It is important to note that there may be cases where loss of power generates an actuation signal, i.e., the system or component fails safe. If this is the case, loss of electric power is not modeled as a cause of failure on demand of the system or component. Instead, it is modeled for the generation of a spurious signal.

4.2.2 If dependencies on HVAC are relevant, they are modeled.

4.2.3 Other potential dependencies on support systems are considered, and modeled as applicable.

Dependencies Related to Sharing of Hardware

Some hardware components may be shared by other components or systems; either within the system or across the boundaries of systems. For example, voters may receive signals from several channels within a system, and sensors may send signals to several systems.

If the same digital hardware is used for implementing several digital systems which perform different functions, such as those carried out by RPS and ESFAS as proposed by Oconee [Oconee 2003], a failure in the hardware or software of the digital platform may adversely affect all these functions. Should these functions be needed at the same time, they would be affected simultaneously. It is desirable that the probabilistic model explicitly includes this impact.

In many cases, a digital system receives input from analog sensors that sense and transmit data about relevant variables. For example, an RPS will receive information about variables indicating that a reactor trip is required. Failures of the sensors, such as a low signal or a loss of signal, should be considered.

For an RPS, the output signals of redundant channels may be sent to the trip breakers through logic devices, e.g., voters. The logic devices probably do not have as much redundancy as the channels and may become dominant causes of failures. The authors' review of the design of a programmable logic controller revealed a single point of failure of a system with a 2-out-of-3 redundancy. Therefore, it is important to correctly model the logic devices and the voters (voting using the input signals).

Digital components and systems share buses and computer networks. The same network or a different one may be used for the communication between the components of one digital system and the components of another. A network also may connect a digital system with the components controlled by the system. For new reactors, e.g., Advanced Boiling Water Reacotr (ABWR) and AP1000 [Westinghouse 2004], some systems share computer networks that allow communication throughout the plant. Failure of a communication network may cause failure of its attached components or systems. In other words, if a digital system shares a communication network with others, it is desirable that the effects on all systems due to failures of the network be modeled jointly. The impact of the failure can range from insignificant, i.e., it does not prevent the components from accomplishing their safety functions, to severe, i.e., it causes the associated systems to fail.

This dependency on communication devices is exemplified by an event that happened at the Browns Ferry Unit 3 on August 19, 2006 [NRC 2007a]. The unit was manually scrammed following a loss of both reactor recirculation pumps. A previous review of the design of a programmable logic controller determined that the root cause was the malfunction of the pumps' controllers because of excessive traffic on the plant's integrated computer system network. This event also demonstrates the value of considering operating experience when attempting to identify possible failure mechanisms.

Desirable characteristics

4.3.1 The digital systems of a plant are examined to determine if there are dependencies due to sharing digital hardware. Any relevant dependencies are modeled.

4.3.2 The effect of sensor failures on the digital system and on other components or systems of the plant are evaluated and included in the probabilistic model.

4.3.3 The failures of devices that process the output of redundant channels of a system are modeled.

4.3.4 Failure of a digital system may trigger an initiating event with possible additional failures of mitigation features. This dependency also is included in the model, as applicable.

Modeling of Fault-Tolerance Features

Fault-tolerance design features are intended to increase the availability and reliability of digital systems, so they are expected to have a positive effect on the system's reliability. However, these features may also have a negative impact on the reliability of digital systems if they are

not designed properly or fail to operate appropriately. The potentially negative impacts of these features should be included in the probabilistic model. The benefits of these features also may be included. Care should be taken to ensure that both the positive and negative impacts of these features are modeled correctly (e.g., ensuring that the beneficial impacts of these features are only credited for appropriate failure modes). The following paragraphs elaborate on these topics.

An important feature of many digital systems is their capability to diagnose failures and to automatically re-configure to reduce or eliminate the impact of failures. For example, a digital system may have a "watchdog timer" (WDT) that periodically receives a signal from a component of the system, such as a microprocessor. If the expected signal is not received, then the WDT sends a signal to the system that the microprocessor failed, and the system will cope by taking mitigating actions, such as using a backup microprocessor.

An issue with including a fault-tolerant feature of a digital system in a probabilistic model is that its design may be such that it can only detect, and hence fix, certain types of failures. In other words, the feature may not detect all the failure modes of the associated component, but just the selected ones that it was designed to repair. In the example above, the WDT can discover that the microprocessor failed when it stops sending a signal to the WDT. However, suppose that the microprocessor had some internal failure and is generating incorrect signals. Then, the WDT cannot determine that the microprocessor has failed because it still is receiving a signal from the microprocessor. Hence, an analysis of the digital system can deterministically identify those failure modes that the fault-tolerant features can detect and fix. Subsequently, the probabilistic model should only give credit to the ability of these features to automatically repair these specific failure modes; it should consider that all the remaining failure modes cannot be automatically tolerated.

As its name implies, a fault-tolerant feature allows a system to avoid a high-level failure, such as a system failure, given the occurrence of a low-level failure, such as that of a microprocessor. If the digital system is available for testing, the tolerance of the system to some failures might be assessed experimentally. By postulating several low-level failures and observing the impact of each on the system (or on selected components), the tolerance of the system (or selected components) can be estimated. A measure of this capability, termed "fault coverage," is the probability that a failure will be tolerated.

A very important characteristic of fault coverage is that it expresses the probability that a failure will be tolerated for the types of failures that were tested. Hence, fault coverage is a function of the failures that were used in testing. For example, if the digital system (or the selected component) is tested with failures for which a fault-tolerant feature was designed, then the probability that a failure will be tolerated (fault coverage) will be 1, unless there are additional failures. Conversely, if the digital system (or the selected component) is tested with failures for which a fault-tolerant feature was not designed, then the fault coverage will be 0. Therefore, it is essential to be aware of the types of failures that were used in testing to apply a value of fault coverage to a probabilistic model. Those failure modes that were not tested should not be considered to be included in the fault coverage, but should be included explicitly in the logic model.

If a digital system (or a selected component) is available for this kind of testing, a design-specific fault coverage would be obtained. On the other hand, fault coverage for other designs may be available in the literature. Again, to apply a published value of fault coverage to a probabilistic model, it is necessary to know the kind of testing that was carried out to arrive at

this value. This information would be stated in terms of the types of failures that were used in the testing.

A fault-tolerant feature of a digital system (or one of its components) can be explicitly included either in the logic model or in the probabilistic data of the components of the model. However, it should not be included in both because this would result in double counting the feature's contribution, that would, in turn, potentially generate an incorrect estimate (in the non-conservative direction) of the reliability of the digital system (or one of its components).

A related issue is when the probabilistic data from generic sources already was adjusted for the contribution of the fault coverage of a component. Hypothetically, assume that a generic database assigns a microprocessor a failure rate of 1×10^{-5}/year and that this rate already was modified to account for a fault-tolerant feature in the microprocessor. It is desirable that a specific datum from a generic database, such as this failure rate, be reviewed to assess whether it was adjusted for the contribution of fault coverage. If so, this failure rate may be used in a probabilistic model, but no additional factor for fault coverage should be applied to the datum for this microprocessor (and no explicit modeling of the fault tolerant feature should be included in the model), unless it is demonstrated that the two fault coverages are independent. Otherwise, applying the same or similar fault coverages would generate a non-conservative estimate of the microprocessor's failure rate.

The objective of a fault-tolerant feature is to have a positive impact on the risk metrics of a system, such as its reliability. On the other hand, a fault-tolerant feature may fail to detect and/or fix a failure mode that it was designed to catch. For example, a sensor failure may be detected by the microprocessor receiving the sensor's input, and the detection does not lie in the capability of the sensor itself, but depends on the features of the microprocessor. Therefore, such dependency must be correctly modeled by a combined model of the sensor and the microprocessor. In addition, if a failure is detected by a fault-tolerant feature, the system may fail to re-configure properly or may be set up into a configuration that is less reliable than the original one. Some or all of these kinds of failures may be relevant for a specific digital system, i.e., they may affect its reliability, so it is desirable that they are accounted for in the probabilistic model.

Summarizing, there is a relationship between the design of a fault-tolerant feature and its associated fault coverage. Hence, care should be exercised when building the probabilistic model, so appropriate credit is given to the fault-tolerant features.

Desirable characteristics

4.4.1 The deterministic analysis of the digital system identifies those failure modes of a component that the fault-tolerant features can detect and the system is able to reconfigure itself to cope with the failure. The probabilistic model only credits the ability of these features to automatically cope with these specific failure modes. It considers that all the remaining failure modes cannot be automatically tolerated.

4.4.2 When applying a value of "fault coverage" to the probabilistic data of a component, the types of failures that were employed in the testing used to derive this value are known. No credit for fault coverage is given to those failure modes that were not included in the testing. This also would apply when using a value of fault coverage from a generic database or the literature.

4.4.3 Information from a generic database about a specific probabilistic datum of a component, such as a failure rate, is reviewed to assess whether it was adjusted for the contribution of fault coverage. If so, this datum may be used in a probabilistic model, but no additional fault coverage is applied to this component, unless it can be shown that the two fault coverages are independent.

4.4.4 A fault-tolerant feature of a digital system (or one of its components) is explicitly included either in the logic model or in the probabilistic data of the relevant components, but not in both.

4.4.5 The probabilistic model accounts for the possibility that a fault-tolerant feature may fail to detect and/or fix a failure mode that it was designed to catch.

4.4.6 If the detection of a failure of a component depends on other components, e.g., a watchdog timer, then the dependency is modeled.

4.4.7 The probabilistic model accounts for the possibility that after a fault-tolerant feature detects a failure, the system may fail to re-configure properly, or may be set up into a configuration that is less reliable than the original one.

Dependencies Related to Type I and II Interactions

NUREG/CR-6901 [Aldemir et al. 2006] defines digital instrumentation and control systems as "...integrated hardware/software/firmware systems whose failure modes may be statistically interdependent due to coupling through the monitored/controlled process (Type I interactions) and/or due to communication between different components, multi-tasking and multiplexing (Type II interactions)." NUREG/CR-6942 [Aldemir et al. 2007] indicates that "...Type I interactions are interactions between digital systems such as the reactor protection system and control system and the controlled plant physical processes (e.g., heat up, pressurization) that would produce failure modes that may be statistically interdependent due to coupling through the monitored/controlled process. Type II interactions are hardware/software/firmware interactions within a digital system (e.g., communication between different components, multi-tasking, multiplexing, etc.) which can lead to failure modes that may originate from communication between different components, multi-tasking, and multiplexing."

Although the PRA technical community has not reached a consensus about the need for explicitly including these interactions in the PRA model, a tentative desirable characteristic is provided here.

Desirable characteristic

4.5 The probabilistic model addresses Type I and Type II interactions.

Dependencies Related to Common Cause Failures

The previous desirable characteristics in the category of "Modeling of dependencies" addressed dependencies that could be explicitly included in the probabilistic model. This section on Common Cause Failures (CCFs) is intended to address other potential dependencies that are not explicitly modeled.

In many cases, a digital system is implemented using several redundant channels. Furthermore, redundancy sometimes is used within a channel to enhance reliability, e.g., in the design of the ABWR. This high level of redundancy is typically used when a digital system is significant to the safety of a NPP, such as the RPS. Such redundancy at the channel level and within each channel usually employs identical components. Hence, CCFs may occur at each level. CCF events represent dependent failures that otherwise are not explicitly modeled, e.g., manufacturing defects or design errors.

Desirable characteristics

Within a digital system, the CCF of both redundant hardware and software should be included in the logic model.

4.6.1 Intra-system hardware CCF. Hardware CCF between similar components within a system is modeled.

4.6.2 Intra-system software CCF. If the channels or subsystems of a digital system (and/or the redundancy within a channel or subsystem) use similar software, software CCF is modeled.

If between different digital systems, the same hardware and software are used, then CCF should be considered.

4.6.3 Inter-system hardware CCF. Hardware CCF between different systems using the same hardware is modeled.

4.6.4 Inter-system software CCF. If similar software is used in different digital systems, software CCF is modeled.

2.5 Probabilistic Data

A digital system is comprised of hardware and software. As discussed throughout this chapter, the logic model of a digital system is expected to capture all relevant contributors (hardware, software, and human error) to system unreliability as well as to plant level risk metrics. To quantify the system unreliability and the plant CDF, it is necessary to have probabilistic data, e.g., a failure rate, for each hardware failure and software failure included in the system model. Probabilistic data for such failures are discussed separately next. The treatment of CCF and "fault coverage" is discussed in the subsection "Modeling of Dependencies." The same desirable characteristics listed below for hardware or software failure data are applicable to the data for CCF and "fault coverage."

<u>Probabilistic Data for Hardware</u>

It is desirable that hardware failure parameters for a particular failure mode of a component are as realistic as possible, so they reflect its reliability characteristics. Data of the same digital components, e.g., Intel Pentium microprocessors, collected from different sites could be used, i.e., component-specific data. If component-specific data are unavailable, generic data of the same generic component, e.g., generic category of microprocessors, may be used.

General guidelines on data analysis are described in several documents of the nuclear industry. In particular, the guidelines in Subsection 4.5.6, "Data analysis," of the ASME standard for PRA for NPP applications [ASME 2005] should be satisfied. Desirable characteristics for digital components are described below.

A digital system may be subject to some operating environments that could adversely affect its reliability. Examples of environmental variables are temperature, radiation, humidity, vibration, electromagnetic interference, radio frequency interference, and pressure. If data are obtained for similar components that are not subject to this type of environment, the data should be modified to account for the corresponding impact on the reliability of the components. In this way, the effects of the operating environment of the system are reflected in the reliability model.

As will be discussed in Chapter 8, hardware data for digital components are scarce or non-existent in the public domain. Hence, it may be difficult to obtain information addressing the desirable characteristics listed below. Additional research is required to generate and analyze raw data of digital components with the objective of obtaining realistic hardware reliability parameters to be used in a probabilistic model of a digital system.

Desirable characteristics

The following desirable characteristics apply to component-specific data, if available:

5.1.1 The data are obtained from the operating experience of the same component as that being evaluated, and preferably in the same or similar application and operating environment.

5.1.2 The sources of raw data are provided.

5.1.3 The method used in estimating the parameters is documented, so that the results can be reproduced.

If component-specific data are not available, generic data, i.e., from a generic database, may be used. The following desirable characteristics *apply to generic data:*

5.1.4 The data of the same generic type of component are used and wide uncertainty bounds are expected.

5.1.5 It is verified that the generic data were collected from components that were designed for applications similar to those in nuclear power plants.

5.1.6 The sources of the generic database are given.

The following desirable characteristics apply to both component-specific and generic data:

5.1.7 If the system being modeled is subject to an adverse environment and the data are obtained from systems that are not subject to a similarly adverse environment, then the data is modified to account for the corresponding impact of the specific environment on the reliability of the system components.

5.1.8 Characteristics 5.1.1 to 5.1.7 also apply to data for CCFs (applies to both component-specific and generic data, as appropriate).

5.1.9 Characteristics 5.1.1 to 5.1.7 also apply to data for "fault coverage" (applies to both component-specific and generic data, as appropriate).

5.1.10 Documentation of basic event calculations includes how the basic event probabilities are calculated in terms of failure rates, mission times, and test and maintenance frequencies.

Probabilistic Data for Software

The task of assessing relevant probabilistic parameters, such as the probability of software failure, for complex software is enormously troublesome. In fact, there is no consensus in the technical community on a method to estimate the reliability of this kind of software. For example, methods based on testing the software may be inadequate because the test environment is not identical to the operating environment, the software tests' results may not be used the same way as are the hardware test data, and exhaustive tests are impossible.

As also mentioned in Subsection 2.3, the software that performs the functions of the digital system usually is comprised of two types: the "application software" that actually carries out these functions, and some kind of "support software" for the former. It is desirable that the evaluation of the probabilistic parameters of the software includes both types.

Software is usually developed by a team of people who implement the software's design requirements. Hence, a specific software is tailored to these specific requirements, and thus, it is functionally and structurally different from any other software. Accordingly, if a technically sound method or process was employed to obtain a probabilistic parameter of a software, such as its probability of failure, in general this probability cannot be applied to any other software. Therefore, substantial technical justification must be given for assuming that a probabilistic parameter from one software can be used for a different software.

Since there is no consensus on a method to assess relevant probabilistic parameters of complex software, such as the probability of failure, an estimate of such parameters would have to be evaluated on a case-by-case basis. More research is needed to establish a method to assess these parameters and/or to validate published methods proposed for this purpose.

Desirable characteristic

5.2 A method for incorporating the contribution of software failures to digital system unreliability is used and documented.

2.6 Uncertainty

Reliability models used to predict the performance of digital systems are complex and consist of many elements. Since probabilities cannot be measured directly, there can be no direct verification of either the form or the results of such models. In addition, these models involve varying degrees of approximation. Therefore, the associated uncertainty in the predictions may be significant and must be addressed.

It is helpful and convenient to categorize uncertainties into those that are associated with the data used to quantify the models (parameter uncertainty), and those that are related to the

models employed (model uncertainty). It is also necessary to identify a third type of uncertainty; namely, uncertainty about the completeness of the model. This type of uncertainty, while it cannot be handled analytically, must be taken into account when making decisions using the results of a probabilistic model.

Parameter uncertainty relates to the uncertainty in the data used in the quantification of the PRA model, such as component failure probabilities. These uncertainties can be characterized by probability distributions. Accordingly, estimation of the reliability parameters used in the model should include an uncertainty analysis. It is desirable that the uncertainties associated with the parameters of the model, such as component failure rates, be propagated through the probabilistic model to estimate the probability distribution for the results of the PRA.

Model uncertainty relates to the uncertainty in the assumptions made in the analysis and the models used. The usual approach is to address model uncertainties by carrying out studies to determine the sensitivity of the results of the analysis if different assumptions were made or models used.

Completeness uncertainty relates to contributions to the reliability that are not included in the probabilistic model of the digital system. This could arise because of failure mechanisms or other factors which may have been left out of the analysis because their existence has not been recognized and therefore they remain unknown. Hence, there is a degree of uncertainty on what the true level of the risk would be and this needs to be recognized as a limitation of the PRA.

Desirable characteristics

6.1 Uncertainties associated with the probabilistic data for hardware and software are estimated.

6.2 Parameter uncertainty is propagated throughout the PRA model such that the uncertainty characteristics of the risk measures, such as CDF, can be determined.

6.3 Key assumptions of the model are identified, and a discussion of the associated model uncertainty provided, including the effects of alternative assumptions.

2.7 Integration of the Digital System Model with a PRA Model

As mentioned at the beginning of this chapter, the vast majority of the PRA models that were developed in the United States and that still are being developed today employ the event tree/fault tree (ET/FT) method. Therefore, it is desirable to ensure that any probabilistic model of a digital system can be integrated with this type of PRA model. This is particularly true for existing NPPs because each one already has a PRA model developed using the ET/FT method.

Regardless of whether a system is analog or digital, if the system executes a control function, such as the control of feed-water, its failure may lead to an initiating event, and so can be modeled in a PRA in terms of the frequency of the initiating event. If the system's function is to mitigate initiating events, it is considered as a protection system and is modeled in terms of the probability that it fails to perform its function. In addition, a spurious actuation of a protection system may lead to a reactor trip, with possible additional impacts due to the starting of the protection system.

A model of a digital system can be integrated with an existing PRA model in two ways:

1. By directly integrating the system model with the PRA model. Since the current PRAs use the ET/FT method, this approach can only be achieved by using a fault tree model of the digital system. Nevertheless, this is the most desirable way of integrating a system model with a PRA because it allows all dependencies of the digital system on other systems (such as its support systems) and vice versa to be explicitly modeled. Since all the dependencies are explicitly modeled in the logic model of the fault trees and event trees, both qualitative and quantitative results can be obtained directly from analysis of the PRA model. Examples of useful results are the minimal cuts sets, and the importance of the basic components of the digital system and of the overall system to the safety of the plant, as measured by a risk metric such as the plant CDF. It should be verified that the fault tree model of the digital system (including the treatment of software) is compatible with the rest of the PRA model.

2. By inputting the results from the model of a digital system into a PRA. This approach involves developing a model of a digital system using a technique other than ET/FT, such as the Markov method. The model then is evaluated, and some relevant measure of the overall system, such as its unavailability/unreliability, is obtained. At the highest level, the system's failure may be modeled in a PRA as a basic event in a fault tree, and this unavailability/unreliability would be used as the probability of failure of this event in the PRA. A Markov model possibly can be simplified to ease this integration with a PRA [Apostolakis 1980, Vesely 1981]. Use of this approach requires that the inter-system dependencies associated with the digital system, e.g., its support systems, be properly accounted for.

It is desirable that the process for integrating the model is relatively straightforward, so it can be mechanized using a software tool and easily verified.

Desirable characteristics

7.1 It is possible to integrate the digital system reliability model into the plant PRA model and the process for integration is verifiable.

7.2 If a model of a digital system has been integrated with a PRA model, all the dependencies related to the system are accounted for. They are the dependencies of the digital system on other systems (such as its support systems), and of other systems on the digital system.

2.8 Human Errors

In general, human errors related to digital systems can undergo the same treatment as those for analog systems. Discussing the probabilistic treatment of human errors is beyond the scope of this task; here, the focus is on some errors related to digital systems.

Once a digital system has been installed and is operational in a NPP, it may be upgraded to fix some identified problems, to enhance its functionality, or for another reason. An upgrade may introduce new errors into the system. For example, based on review of software failures in the domestic NPPs from January 1, 1996 to December 31, 2005, carried out in a previous activity

related to this project, it was found that the second-leading cause of software failure is "Operation and maintenance," accounting for 12 out of 45 events (i.e., about 27%)[2]. Most events related to this cause involve a failure introduced during modifications or upgrades of the software after it was developed, installed, and had operated for some time. In other words, software that was meeting its expected functions was modified, during which some fault was introduced. When upgrading a digital system, errors also may be introduced in its hardware. The people carrying out the upgrades may make mistakes, such as installing an incorrect version of hardware or software. This type of failure also may happen when upgrading an analog system. However, it may have a higher probability of occurring when upgrading digital systems due to their greater complexity and use of software.

The introduction of digital systems also brought new human-system interfaces (HSIs). If these interfaces are not well designed or implemented, they are likely to increase the probability of human error during use.

It is desirable that both types of human errors discussed above are accounted for in the probabilistic model and other types of human errors related to digital systems are analyzed, as applicable.

Desirable characteristics

8.1 Human errors during upgrade of hardware and software are included.

8.2 Human errors related to HSI are included.

2.9 Documentation and Results

These desirable characteristics consider documentation of key assumptions and results.

Desirable characteristics

9.1 Key assumptions made in developing the reliability model and probabilistic data are documented.

9.2 Assumptions made in developing the reliability model and probabilistic data are realistic, and the associated technical justifications are sound and documented.

9.3 The dominant failure modes of the reliability model are documented with a description of the sequence of events that need to take place and how the failures propagate to fail the system. The sequence of events realistically represents the system's behavior at the level of detail of the model.

2.10 Summary

The set of desirable characteristics takes into consideration the unique features of digital systems. The characteristics could eventually be used, in whole or in part, along with generic PRA quality requirements such as those of Regulatory Guide 1.200 [NRC 2007b]. The

[2]The most predominant cause is "Software requirements analysis," accounting for 16 out of the 45 events (i.e., about 36%).

characteristics are based on experience in analyzing digital systems and on a literature review of digital system reliability modeling.

Fifty-two (52) desirable characteristics were developed and grouped into nine broad categories covering the probabilistic model of a digital system and its documentation:

1. Level of Detail of the Probabilistic Model
2. Identification of Failure Modes of the Components of a Digital System
3. Modeling of Software Failures
4. Modeling of Dependencies
    - Dependencies Due to Communication
    - Dependencies Due to Support Systems
    - Dependencies Due to Sharing of Hardware
    - Modeling of Fault Tolerant Features
    - Dependencies Due to Type I and II Interactions
    - Dependencies Due to Common Cause Failures
5. Probabilistic Data
    - Probabilistic Data for Hardware
    - Probabilistic Data for Software
6. Uncertainty
7. Integration of the Digital System Model with a PRA Model
8. Human Errors
9. Documentation and Results

A list of the desirable characteristics is provided in Table 2-1.

Table 2-1 Summary of desirable characteristics.[3]

LEVEL OF DETAIL OF THE PROBABILISTIC MODEL	
1.1	A reliability model of a digital system is developed to a level of detail that captures the design features affecting the system's reliability, provides the output needed for risk evaluations, and for which probabilistic data are available.
IDENTIFICATION OF FAILURE MODES OF THE COMPONENTS OF A DIGITAL SYSTEM	
2.1	A method is applied for identifying failure modes of the basic components of the digital system and their impact on the system. This method provides a systematic way of carrying out this identification such that there is confidence that the failure modes obtained are as complete as possible.
2.2	Supporting analyses are carried out to determine how specific features of a design, such as communication, voting, and synchronization, could affect system operation. These analyses determine whether the specific design features could introduce dependent failures that should be modeled.
2.3	Failure modes that have occurred in the operating experience are examined and their applicability to the digital system being studied is considered.
2.4	The probabilistic model of the digital system accounts for the possibility that the system may fail due to incorrect design requirements, or due to correct requirements that are not correctly implemented into the system.
MODELING OF SOFTWARE FAILURES	
3.1	Software failures are accounted for in the probabilistic model.
3.2	Modeling of software failures is consistent with the basis of how they occur, that is, software failures happen when triggering events occur.
3.3	Modeling of software failures accounts for the context/boundary condition in which software is used.
3.4	The model of the software includes the "application software" and the "support software."
MODELING OF DEPENDENCIES	
Dependencies Due to Communication	
4.1.1	Inter-system failure propagation is addressed, and modeled as applicable.
4.1.2	Inter-channel failure propagation is addressed, and modeled as application.
4.1.3	Intra-channel failure propagation is addressed, and modeled as application.

[3]The reader is advised to refer to the background of each desirable characteristic in the previous sections of this chapter.

Table 2-1 Summary of desirable characteristics (cont'd).

Dependencies Due to Support Systems	
4.2.1	Loss of power to safety-related digital systems is modeled. It is important to note that there may be cases where loss of power generates an actuation signal, i.e., the system or component fails safe. If this is the case, loss of electric power is not modeled as a cause of failure on demand of the system or component. Instead, it is modeled for the generation of spurious signal.
4.2.2	If dependencies on HVAC are relevant, they are modeled.
4.2.3	Other potential dependencies on support systems are considered, and modeled as applicable.
Dependencies Due to Sharing of Hardware	
4.3.1	The digital systems of a plant are examined to determine if there are dependencies due to sharing digital hardware. Any relevant dependencies are modeled.
4.3.2	The effect of sensor failures on the digital system and on other components or systems of the plant are evaluated and included in the probabilistic model.
4.3.3	The failures of devices that process the output of redundant channels of a system are modeled.
4.3.4	Failure of a digital system may trigger an initiating event with possible additional failures of mitigation features. This dependency also is included in the model, as applicable.
Modeling of Fault Tolerant Features	
4.4.1	The deterministic analysis of the digital system identified those failure modes of a component that the fault-tolerant features can detect and the system is able to reconfigure itself to cope with the failure. The probabilistic model only credits the ability of these features to automatically cope with these specific failure modes. It considers that all the remaining failure modes cannot be automatically tolerated.
4.4.2	When applying a value of "fault coverage" to the probabilistic data of a component, the types of failures that were employed in the testing used to derive this value are known. No credit for fault coverage is given to those failure modes that were not included in the testing. This also would apply when using a value of fault coverage from a generic database or the literature.
4.4.3	Information from a generic database about a specific probabilistic datum of a component, such as a failure rate, is reviewed to assess whether it was adjusted for the contribution of fault coverage. If so, this datum may be used in a probabilistic model, but no additional fault coverage is applied to this component, unless it can be shown that the two fault coverage's are independent.
4.4.4	A fault-tolerant feature of a digital system (or one of its components) is explicitly included either in the logic model or in the probabilistic data of the relevant components, but not in both.
4.4.5	The probabilistic model accounts for the possibility that a fault-tolerant feature may fail to detect and/or fix a failure mode that it was designed to catch.

Table 2-1 Summary of desirable characteristics (cont'd).

4.4.6	If the detection of a failure of component depends on other components, e.g., a watchdog timer, then the dependency is modeled.
4.4.7	The probabilistic model accounts for the possibility that after a fault-tolerant feature detects a failure, the system may fail to re—configure properly, or may be set up into a configuration that is less reliable than the original one.
Dependencies Related to Type I and II Interactions	
4.5	The probabilistic model addresses Type I and Type II interactions.
Dependencies Related to Common Cause Failures	
4.6.1	Intra-system hardware CCF. Hardware CCF between similar components within a system is modeled.
4.6.2	Intra-system software CCF. If the channels or subsystems of a digital system (and/or the redundancy within a channel or subsystem) uses similar software, software CCF is modeled.
4.6.3	Inter-system hardware CCF. Hardware CCF between different systems using the same hardware is modeled.
4.6.4	Inter-system software CCF. If similar software is used in different digital systems, software CCF is modeled.
PROBABILISTIC DATA	
Probabilistic Data for Hardware	
5.1.1	The data are obtained form the operating experience of the same component as that being evaluated, and preferably in the same or similar application and operating environment.
5.1.2	The sources of raw data are provided.
5.1.3	The method used in estimating the parameters is documented, so that the results can be reproduced.
5.1.4	The data of the same generic type of component are used and wide uncertainty bounds are expected.
5.1.5	It is verified that the generic data were collected from components that were designed for applications similar to those in nuclear power plants.
5.1.6	The sources of the generic database are given.
5.1.7	If the system being modeled is subject to an adverse environment and the data re obtained from systems that are not subject to a similarly adverse environment, then the data is modified to account for the corresponding impact of the specific environment on the reliability of the system components.
5.1.8	Characteristics 5.1.1 to 5.1.7 also apply to data for CCFs (applies to both component-specific and generic data, as appropriate).

Table 2-1 Summary of desirable characteristics (cont'd).

5.1.9	Characteristics 5.1.1 to 5.1.7 also apply to data for "fault coverage" (applies to both component-specific and generic data, as appropriate).
5.1.10	Documentation of basic event calculations includes how the basic event probabilities are calculated in terms of failure rates, mission times, and test and maintenance frequencies.
Probabilistic Data for Software	
5.2	A method for incorporating the contribution of software failures to digital system unreliability is used and documented.
UNCERTAINTY	
6.1	Uncertainties associated with the probabilistic data for hardware and software are estimated.
6.2	Parameter uncertainty is propagated throughout the PRA model such that the uncertainty characteristics of the risk measures, such as CDF, can be determined.
6.3	Key assumptions of the model are identified, and a discussion of the associated model uncertainty provided, including the effects of alternative assumptions.
INTEGRATION OF THE DIGITAL SYSTEM MODEL WITH A PRA MODEL	
7.1	It is possible to integrate the digital system reliability model into the plant PRA model and the process for integration is verifiable.
7.2	If a model of a digital system has been integrated with a PRA model, all the dependencies related to the system are accounted for. They are the dependencies of the digital system on other systems (such as its support system), and of other systems on the digital system.
HUMAN ERRORS	
8.1	Human errors during upgrade of hardware and software are included.
8.2	Human errors related to HIS are included.
DOCUMENTATION AND RESULTS	
9.1	Key assumptions made in developing the reliability model and probabilistic data are documented.
9.2	Assumptions made in developing the reliability model and probabilistic data are realistic, and the associated technical justifications are sound and documented.
9.3	The dominant failure modes of the reliability model are documented with a description of the sequence of events that need to take place and how the failures propagate to fail the system. The sequence of events realistically represents the system's behavior at the level of detail of the mode.

3. OVERALL APPROACH OF MODELING

Due to the lack of peer reviewed models and analysis of digital systems for use in nuclear power plant (NPP) probabilistic risk assessments (PRAs), this project includes reliability models for two benchmark test cases to support the development of tools and methods for including these types of models in PRAs. The first test case involves a digital feedwater control system (DFWCS) of a two-loop pressurized water reactor; the second involves a Reactor Protection System (RPS). The main objective of this chapter is to delineate how the PRA models of the first benchmark system will be analyzed, constructed, and quantified using the traditional event tree/fault tree (ET/FT) and Markov methods. The detailed construction and quantification of these models, and the analysis of the second benchmark case, will be the subject of follow-up tasks of this project.

Section 3.1 discusses operational aspects and risk insights of the DFWCS that are relevant for the PRA modeling of this system, while Section 3.2 describes the overall approach to, and the major steps involved in, this modeling.

3.1 Operational Aspects and Risk Insights of the DFWCS

As mentioned above, the first benchmark system is based on a DFWCS of a two-loop pressurized water reactor. Each of the two reactor-coolant loops contains a reactor coolant pump and steam generator (S/G). The main feedwater system (FWS) consists of steam-turbine-driven feedwater pumps (FWPs), minimum flow control valves, a pump-seal water system, main feedwater regulating valves (MFRVs), bypass feedwater regulating valves (BFRVs), high-pressure feedwater heaters, and the associated piping and instrumentation. The feedwater of each secondary loop is controlled by a DFWCS, which is described in detail in Chapter 4.

During plant power operation, the function of the FWS is to remove heat from the primary system by providing feedwater to the S/Gs. Degradation that exceeds certain operational parameters or total loss of the FWS during this operation causes a reactor trip; accordingly, degradation or loss of the FWS contributes to plant risk because it causes an initiating event (IE). In general, several types of IEs that can occur are associated with such failures, such as excessive main feedwater, total loss of main feedwater, and partial loss of main feedwater. The contribution to plant risk of this degradation or total loss is considered significant because a reactor trip results in a transient that challenges the plant. Should some components or trains be unavailable at the time of the trip, the transient may evolve into a serious safety challenge. For example, the accident at Three Mile Island Unit 2 on March 28, 1979, started with a reactor trip with loss of feedwater. Furthermore, a reactor trip entails economic losses to the plant owner.

If a reactor is tripped due to causes unrelated to failures of FWS components, the FWS may be available after the trip, thus providing a means of decay heat removal. Accordingly, degradation or total loss of the FWS after the trip contributes to plant risk because this system would not be able to perform its intended mitigative function. In this case, the contribution to plant risk is not considered significant for two reasons: (1) the FWS is not available after some of the IEs that are important contributors to plant risk, such as loss of offsite power and (2) for those IEs after which the FWS may be available, the plant also may have available redundant and diverse means of removing decay heat, for example, the Auxiliary Feedwater System and feed-and-bleed cooling.

Hence, degradation or total loss of the FWS has two contributions to plant risk: (1) it may cause IEs and (2) it may fail to fulfill its mitigative function after a reactor trip. The first contribution is analyzed in this study because it is the one considered most significant to plant risk. Hence, a PRA model is developed for failures of the FWS that cause an IE.

3.2 Major Steps to Building Models

This section discusses the definition and scope of the probabilistic model of this type of IE and the major steps for building it.

3.2.1 Definition and Scope of the Probabilistic Model

The plant contains two secondary loops; the feedwater in each is controlled by an identical DFWCS, so that the analysis of a single DFWCS is applicable to the other. Hence, a probabilistic model will be developed only for one DFWCS. Potential dependencies between the two DFWCSs will be considered in the analysis of one DFWCS. As discussed below, failure of either DFWCS alone is sufficient to result in the undesired outcome (i.e., a reactor trip).

When the plant is in the power operation mode, a DFWCS automatically controls the feedwater in its associated secondary loop, unless the plant operators have set the DFWCS in the manual mode. Failures in one DFWCS can cause it to lose automatic control of its loop, requiring the operators to attempt to take manual control. The degree of difficulty in assuming manual control depends on two factors related to the specific failures that caused the loss of automatic control: (1) availability of indication, i.e., annunciation in the control room versus only at the plant computer and (2) availability of the hardware needed for manual control, which may be adversely affected by the failures. Hence, the loss of automatic control may or may not be recoverable by the operators. For the purpose of this study, failure of a DFWCS is defined as loss of automatic and manual control of its related loop. This loss is assumed to cause a reactor trip because it can result in undesired impacts; for example, a low level in the steam generators that triggers a reactor trip can occur if the speed of the FWS pumps falls substantially. Therefore, the IE modeled here is defined as "Loss of control of the loop associated with a DFWCS."

Two probabilistic models are delineated to account for failures of a DFWCS that cause this IE using the traditional ET/FT and Markov methods. The scope of the models is limited to failures of the components associated with the DFWCS; external events, such as fire and earthquakes, are not considered.

An important use of a probabilistic model of an IE is to estimate the frequency of occurrence of the IE. The following subsection describes an approach to assessing the IE frequency.

3.2.2 Evaluating the Frequency of an Initiating Event

In the PRA of a NPP, the frequency of an IE related to the loss or degradation of a system of the NPP usually is determined by either of two methods: (1) Statistical analysis of failure data collected during NPP operation or (2) building and evaluating a probabilistic model of the system. When applicable operational data are available, the first method is commonly employed. However, when a new digital system is installed in a NPP, these data are unavailable. Similarly, if an existing system is modified to include digital components, the existing data may not be

applicable. The use of the second method is indicated in these cases. This subsection discusses employing Markov and fault-tree models for assessing the frequency of an IE.

An initiating event frequency is the expected number of system failures (*ENF*) per unit time. As discussed above, failure of a DFWCS is defined as the loss of control of its associated loop. Typically, the frequency is expressed in units of "per year" for input into a NPP PRA. Thus, the period *T* over which the *ENF* must be estimated is one year. In other words, an initiating event frequency is the expected number of system failures per year.

The number of initiating events, i.e., number of failures of the system, is considered to follow a Poisson process. The conditional failure rate of the system, $\lambda(t)$, is defined as the probability that the system fails per unit time at time *t*, given that the system has been operable from time (t) zero up to time *t*. Assuming that the system is operable at $t = 0$, i.e., at the beginning of the period *T*, the expected number of occurrences of an IE is obtained by integrating this rate over the period *T*:

$$ENF = \int_0^T \lambda(t)dt \qquad (3\text{-}1)$$

The reliability of a system at time *t*, $R(t)$, is the probability that the system had no failure during the time interval (0,*t*], given that the system was operable at time 0. As pointed out by Barlow and Proschan [1975], the failure rate and the reliability of the system are related by

$$\lambda(t) = \frac{dR(t)}{R(t)dt} \qquad (3\text{-}2)$$

Inserting equation (3-2) into (3-1),

$$ENF = \int_0^T \frac{dR(t)}{R(t)} \qquad (3\text{-}3)$$

The right-hand side of equation (3-3) is equal to minus the natural (base e) logarithm of $R(T)$. Accordingly,

$$ENF = -\ln [R(T)] \qquad (3\text{-}4)$$

where "*ln*" means the natural logarithm, and *T* is one year. This derivation assumed that the system is initially operable (at time t = 0). In other words, its reliability at this time, $R(0)$, is 1.

The initiating event frequency, i.e., the expected number of system failures per year is obtained by dividing the *ENF* by the period of interest, *T*:

$$f = -\ln [R(T)] / T \qquad (3\text{-}5)$$

Equation (3-5) can be used to evaluate the initiating event frequency. The Markov and fault tree methods offer different approaches to calculate the reliability at time *T*, i.e., $R(T)$, used by this equation. They are described below.

3.2.2.1 Markov Method

Shooman [1968] indicates that "Any Markov model is defined by a set of probabilities p_{ij} which define the probability of transition from any state i to any state j...One of the most important features of any Markov model is that the transition probability p_{ij} depends only on states i and j and is completely independent of all past states except the last one, state i."

A Markov model can be expressed in terms of a set of linear differential equations modeling the transitions among system states.

$$d\underline{P}/dt = \underline{M}\,\underline{P} \qquad (3\text{-}6)$$

where $\underline{P}$ represents the probabilities of the system states, and $\underline{M}$ is the transition matrix containing the constant transition rates among the system states. A solution of the equation (3-6) provides the probabilistic information about the system. For example, the sum of the probabilities of success states is the reliability, and can be used to calculate the frequency of system failure.

In the case of the DFWCS, the system is assumed to be initially in an operable state (time = 0). Every time a component of the DFWCS fails, the DFWCS transits into another state. In general, the system experiences several "jumps" of states until it reaches the failed state, i.e., the state that causes an IE. Accordingly, the reliability at time T can be expressed as

$$R(T) = 1 - P_f(T) \qquad (3\text{-}7)$$

where $P_f(T)$ is the probability that the DFWCS is in the failed state at time T.

Since failure of a DFWCS causes an IE, the repair of this system is not relevant to the model of an IE. Accordingly, once the failure state of this system is reached, it is considered an absorbing state with no transition out of it.

The expression for the initiating event frequency (3-5) can be re-formulated by using equation (3-7):

$$f = -\ln\,[1 - P_f(T)] / T \qquad (3\text{-}8)$$

Thus, this frequency can be estimated by solving the Markov model to obtain $P_f(T)$. This model is described in Chapter 6.

3.2.2.2 Fault Tree Method

The traditional fault tree method does not have the capability to allow the assessment of the probability of being in the failed (absorbing) state as a function of time. Nevertheless, by building and solving a fault tree whose top event is the failure of the DFWCS within the period T, this method can be used to obtain the probability of this failure that then can be used in equation (3-8) to yield the initiating event frequency. The process of building this fault tree is delineated in Chapter 7.

To develop the Markov and fault tree models, some supporting information is necessary, such as identifying the failure modes of each relevant component of the DFWCS. Subsection 3.2.3 is an overview of the process used for developing these models.

3.2.3 Overview of Modeling Process

Chapters 4 to 8 present the process used for developing the Markov and fault tree models for the DFWCS. A full understanding of the way the DFWCS and each of its relevant components operate is necessary in developing any probabilistic model. Chapter 4 describes the DFWCS in detail, including its function, components, associated controllers, dependencies and interfaces, and digital features.

The failures of the components of any system (analog or digital) cause the failure of the system. In general, a component may fail in different ways or modes, called failure modes. For example, for an analog component such as a valve, depending on the required function of the valve, there are two typical failure modes: (1) The valve fails to open or (2) the valve fails to close. Similarly, a digital component can have several failure modes. For example, a DFWCS has two Central Processing Units (CPUs), main and backup; each has several failure modes, such as:

1. Continued operation with latent failures. This failure mode represents failures that do not immediately affect DFWCS operation. It is considered a failure mode because, combined with other failure(s), it could cause the DFWCS to fail.

2. Failures that cannot be detected by the DFWCS. These are failures of the CPU that cannot be detected, and hence fixed, by the failure-detection mechanisms implemented in the DFWCS. This failure mode leads to failure of the DFWCS.

Hence, in general, one or several failure modes of its components may be required for the DFWCS to fail. Accordingly, as discussed in Section 2.2, it is very important to identify the failure modes of the components of the system and the impact of each of them on the DFWCS. The technique of failure modes and effects analysis (FMEA) was used for this purpose and its results are discussed in Chapter 5; Appendix B presents the detailed FMEA.

After the relevant failure modes of the components and their impacts on the DFWCS are identified, they can be used to build probabilistic models using the traditional Markov and ET/FT methods. Using the Markov method as an example, those failure modes of components that directly cause system failure can be modeled as direct transitions from the normal state to the failed (absorbing) state; those that do not will lead to a transition to an intermediate state. For example, starting with the normal state, the occurrence of the failure mode "Continued operation with latent failures" of the Main CPU will cause the transition from the system's normal state to a state that includes this failure. Subsequent failures then cause the system to fail, i.e., to reach the failed state. Software failures also can be included in the model using the same process, i.e., if a software failure directly causes system failure, then it is modeled as a transition from the normal state to the failed state; otherwise, it is modeled as a transition to an intermediate state. Chapters 6 and 7 delineate the Markov and fault tree models for the DFWCS, respectively.

To quantify the two models, relevant probabilistic parameters for each failure mode are needed. For example, a transition in the Markov process occurs when one failure mode of a component happens. The transition rate associated with this transition is the failure rate of this failure mode.

Chapter 8 discusses the probabilistic data for hardware digital components that is planned to be used for this quantification. Due to weaknesses in the state-of-the-art in failure parameters of digital components, the data will only be used to demonstrate the reliability methods and exercise the models and are not appropriate for quantifying models that are to be used in support of decision-making.

Modeling using both traditional methods takes into consideration the following:

1. Since the models are developed to assess the frequency of an initiating event, the plant is assumed to be in the mode of power operation. In this mode of operation, it is expected that if some components of the system fail, they will not be repaired because this activity would likely cause or require a reactor trip. Hence, the plant's staff would wait until the reactor has been tripped for another reason to carry out any needed repair. For this reason, failures of components of the system, even if they are detected, are considered to be non-repairable.

2. The main focus of this task is to delineate probabilistic models of the hardware of the components of the system. The human errors that may contribute to the unreliability of the system are not studied in detail, and are considered out of the scope of this task. The discussions in the next chapters provide some information and insights that are relevant to a human reliability analysis, without modeling in detail nor evaluating the probability of human errors.

3. As also discussed in Section 2.3, software is a unique feature of digital systems, and its failure can significantly impact its associated digital system. Probabilistic parameters for this kind of failure, such as failure rates, also are necessary for quantifying the models. Hence, a method for assessing them is required. However, the technical community has not reached a consensus about a method to be used for this purpose. Accordingly, the current scope of this task does not include estimating these parameters. To address this shortcoming, a range of parameters will be used in the quantification. For example, a range of values for the failure rate of a specific software failure in the models will be employed to quantify the models and study the significance of this failure to each overall model.

4. DESCRIPTION OF A DIGITAL FEEDWATER CONTROL SYSTEM

This chapter provides a functional and physical overview of a digital feedwater control system (DFWCS) of an operating nuclear power plant (NPP). The information is based in large part on information provided by the NPP and is the basis for performing Failure Modes and Effects Analyses and creating reliability models of the DFWCS, as will be seen in later chapters.

The NPP has two units, each consisting of two reactor coolant loops. There are a reactor coolant pump and a steam generator (S/G) for each reactor coolant loop. The feedwater system (FWS) consists of steam-turbine-driven centrifugal S/G feedwater pumps (FWPs), minimum flow control valves, a pump seal water system, main feedwater regulating valves (MFRVs), bypass feedwater regulating valves (BFRVs), high pressure feedwater heaters, and associated piping and instrumentation. Figure 4-1 is a simplified diagram of the FWS (without mini-flow valves, seal water system, and high pressure feedwater heater) which shows the location of some of the sensors which provide input to the DFWCS. Note that the two trains of the FWS are headered together at the discharge as well as the suction of the FWPs. The sensors from the reactor coolant loops are shared by the DFWCSs of the two FWS trains.

The DFWCSs support both automatic and manual control of the FWS. The automatic control modes and algorithms of the two feedwater trains are identical and based on an assumption that the two reactor coolant loops are symmetrical. The two systems are not completely independent. For example, the S/G levels and steam flows are averaged before they are used in the calculation of the demand signals for the devices. In addition, the two DFWCSs exchange MFRV demand signals that are used in calculating the FWP demands. The pressurizer may introduce asymmetry in the loops. Use of the FWP speed bias potential meter allows manual starting or stopping of one of the FWPs.

A system level description of control modes is given in Section 4.1. Each DFWCS has two identical central processing units (CPUs): Main and Backup CPUs. Usually the Main CPU provides control demands. A failover to the Backup CPU may occur under certain circumstances, e.g., a large deviation between two feedwater level signals of the same S/G. The deviation logic that checks S/G level signals is discussed in Section 4.1. Section 4.2 presents details of the Main and Backup CPUs and device controllers. A digital valve controller (positioner) that directly positions feedwater regulating valves (FRVs) is described in Section 4.3.

Dependencies inside the DFWCS and between the DFWCS and other systems/components are discussed in Section 4.4. Section 4.5 presents digital features of the DFWCS, such as a Microlink communication network and watchdog timer, etc., that are important for digital system reliability modeling. Concluding remarks are provided in Section 4.6. Some of the details of the system design were not available for this study. However, since this is a proof-of-concept study, these details are not strictly necessary, and some assumptions were made, as discussed in the rest of the report. On the other hand, when the objective of a study requires an estimate of the reliability of the system (and of relevant risk metrics) that is realistic, it would be necessary to obtain all the relevant information on the system's design and operation.

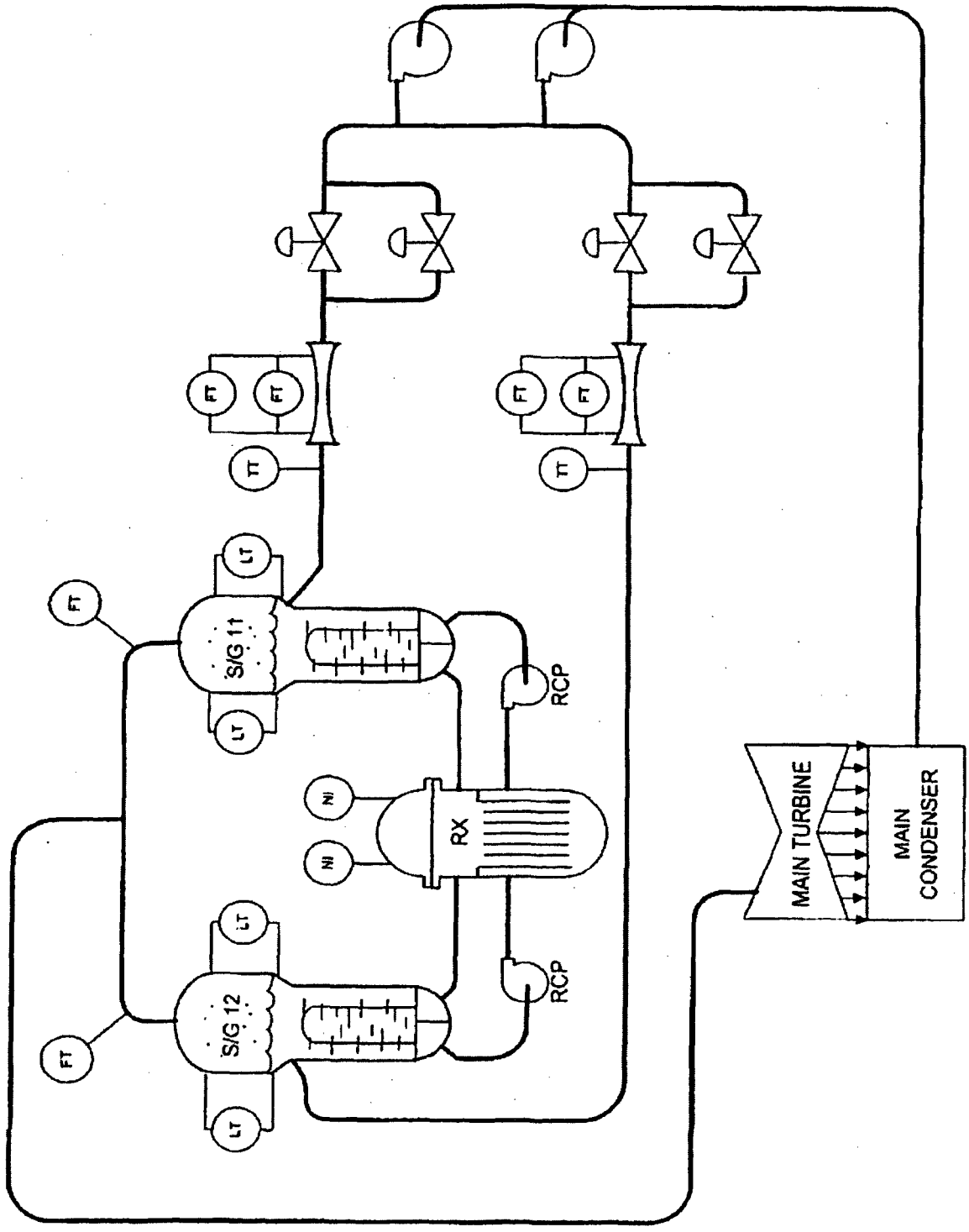


Figure 4-1 A simplified diagram of the feedwater system

4.1 System Level Description

4.1.1 Control Modes and Algorithms

Typically, the FWS is manually controlled below 2% power and automatically controlled by the DFWCS above 2%. It has two automatic modes of operation, low (2% to 15%) and high (above 15%) power, operating in three-element (S/G level, feedwater flow, and steam flow) and single-element (S/G level) controls, respectively.

Figure 4-2 is a simplified diagram which shows only one of the reactor coolant loops with its associated DFWCS. During low power operation, the FWPs are running at the minimum speed demand plus bias, the MFRVs are closed with a negative bias to guarantee closure, and the BFRVs are controlled to maintain S/G level. During high power operation, the BFRVs are normally closed and the DFWCS controls the MFRVs and FWPs.

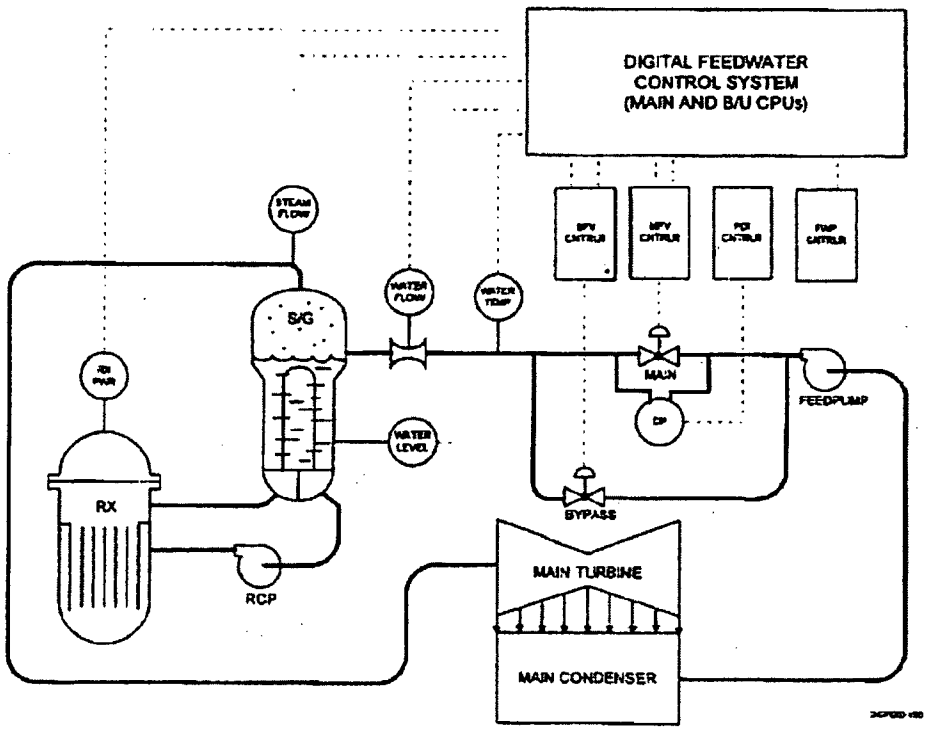


Figure 4-2 One of the reactor coolant loops with its associated DFWCS

The DFWCS of each reactor coolant loop consists of two identical microprocessors/CPUs, main and backup (Azonix model μMAC 7000) which run identical software to generate the control signals to the Manual/Automatic (M/A) controllers, i.e., FWP, main feedwater valve (MFV), and bypass feedwater valve (BFV) controllers (Fischer and Porter model 53MC5000). The M/A controllers serve as the man machine interface (MMI) which allows manual control of the devices. The MFV and BFV controllers normally pass the demand signals from the main

microprocessor to the MFRV and BFRV valve positioners (Fisher FIELDVUE Digital Valve Controller Type DVC5000 series [Fisher 2001]) and Lovejoy FWP turbine speed controller. If the main microprocessor is found failed, then the signals from the backup microprocessor are used. If both main and backup microprocessors are failed, then the M/A controllers allow manual control of the devices. A fourth M/A controller, the pressure differential indication (PDI) controller, is normally on standby and would automatically take over if the MFV controller fails to support manual control of the MFRV. It also can be used to take manual control of the BFRV by operating a hand switch.

The DFWCS also has automatic transfers from low power mode to high power mode and from high power mode to low power mode. Under certain conditions, automatic mode transfer is inhibited, e.g., when a MFV or BFV controller is in manual.

The DFWCS microprocessors also receive reactor trip and turbine trip signals, and upon the signals will adjust the demand signals to the M/A controllers which remain in auto mode. The MFRV will be ramped to shut bias immediately following a reactor trip. After a pre-determined time delay, the post trip positioning relay circuit will ensure that the MFV demand signal is reduced to zero. The BFRV response to the reactor trip is primarily controlled by hardware downstream of the BFV controller, i.e., BFV post trip controller, which moves the valve to a pre-set position that corresponds to about 3.5% of the full power feedwater flow rate. The DFWCS will act as a backup to the post trip controller by setting the BFV demand signal to the same preset value when the reactor trip is sensed. The DFWCS also sets the FWP speed demand signal to a minimum value plus negative bias which will provide sufficient pressure differential across the BFRV for post-trip feedwater flow requirement. Often, a turbine trip would also lead to a reactor trip, and the response of the DFWCS follows that of a reactor trip. When the plant is initially operating below the power threshold which will result in a turbine trip but not a reactor trip, the DFWCS will control feedwater flow to maintain S/G level at its normal setpoints.

The analog and digital inputs and outputs of the microprocessors are tabulated in Tables 4-1 and 4-2, respectively. They were based on the system requirement document of the DFWCS. The tables also include a brief description of how the signals are used for processing. Table 4-3 contains the signals associated with the optical isolator PB4R, e.g., watchdog timers. The inputs and outputs of the M/A controllers are listed in Tables 4-4 to 4-7.

As mentioned previously, the DFWCS can operate in high or low power control mode. In the high power mode, the DFWCS will perform dynamic compensation on the average of the two S/G level input signals and dynamic compensation on the flow error signal. The flow error signal will be the difference between the average of the two steam flow signals and the average of the two feedwater flow signals. The compensated level signal will be summed with the level setpoint to produce a level error signal. The flow error and level error signals will be combined and then processed by a proportional and integral controller network. The output of the proportional and integral controller will then be summed with a signal proportional to the BFRV position. The resulting flow demand signal will be indicative of the required feedwater flow through the MFRV and will be used to generate both the MFRV position demand signal and the FWP speed demand signal. When in this mode the BFRV will normally be closed.

Table 4-1 Analog inputs/outputs of the microprocessors.

No.	Description S/G #1 (S/G #2)	Input/Output	From/to	Processing
		Analog Board 1		
1	FW Pump A Demand plus Bias (Pump B)	Output	FWP controller	FWP speed control
2	Bypass Valve (BFV) Demand	Output	BFV controller	BFV control
3	Main Valve (MFV) Demand	Output	MFV controller	MFV control
4	FW Temperature #2	Input	FW temperature sensor #2	Input sensor
5	FW Temperature #1	Input	FW temperature sensor #1	Input sensor
6	FW Pump A Bias (Pump B)	Input	FWP controller	FWP demand calc.
7	Main Valve Tracking S/G #2 (S/G #1)	Input	MFV controller of S/G #2 through Microprocessors of S/G #2	Calculating FWP Demand.
8	FW Pump A Tracking (Pump B)	Input	FWP controller	Tracking and deviation determination
9	spare			
10	spare			
11	spare			
12	spare			
13	MFRV LVDT #2	Input	MFRV positioner	MFRV positioner failover
14	MFRV LVDT #1	Input	MFRV positioner	MFRV position failover
15	FRV differential pressure #2	Input	PDI controller	Gooseneck fill alarm
16	FRV differential pressure #1	Input	PDI controller	Gooseneck fill alarm

Table 4-1 Analog inputs/outputs of the microprocessors (cont'd).

No.	Description S/G #1 (S/G #2)	Input/Output	From/to	Processing
Analog Board 2				
1	(reserved for test point output)			
2	(reserved for test point output)			
3	spare			
4	spare			
5	spare			
6	S/G Level #1	Input		
7	S/G Level #2	Input	S/G Level #2	Sensor input
8	FW Flow #1	Input	FW Flow #1	Sensor input
9	FW Flow #2	Input	FW Flow #2	Sensor input
10	Steam Flow #1	Input	Steam Flow #1	Sensor input
11	Steam Flow #2	Input	Steam Flow #2	Sensor input
12	Neutron Flux #1	Input	Neutron Flux #1	Sensor input
13	Neutron Flux #2	Input	Neutron Flux #2	Sensor input
14	S/G Level Setpoint (adjustment to)	Input	MFV controller	Control calcs., deviation logic
15	Bypass Valve Tracking	Input	BFV controller	Tracking and deviation determination
16	Main Valve Tracking (shared with other S/G)	Input	MFV controller	Tracking and deviation determination

Table 4-2 Digital inputs and outputs of the microprocessors.

Digital inputs of microprocessors			
No.	**Description S/G #1 (SG #2)**	**From**	**Processing**
1	BFV M/A Controller Status	BFV controller	Automatic mode transfer logic, response to load change logic
2	MFV M/A Controller Status	MFV controller	Automatic mode transfer logic, response to load change logic
3	FW Pump A M/A Controller Status	FWP controller	Automatic mode transfer logic, response to load change logic
4	Reactor trip	Post reactor trip position relay	Reactor trip response
5	Main/Backup Microprocessor Status	Pre-selected	Identifies microprocessor as Main or Backup
6	Turbine Trip	Turbine trip relay	Turbine trip response
7	Main Microprocessor Failed	MFV controller	Deviation logic of other microprocessor
8	Backup Microprocessor Failed	MFV controller	Deviation logic of other microprocessor
9	Time Sync	external clock	Possible sync of main and backup microprocessors
10	Bypass Neutron Flux #1 (Keyswitch)	Keyswitch	Bypass neutron flux #1
11	Bypass Neutron Flux #2 (Keyswitch)	Keyswitch	Bypass neutron flux #2
12	Positioner Selected	MFRV positioner	Possibly failover logic
13	No Failures in Other Microprocessor	Other microprocessor	Possibly deviation/failover logic
14	No Deviations in Other Microprocessor	Other microprocessor	Possibly deviation/failover logic
15	Both Level Signals Valid in Other Microprocessor	Other microprocessor	Possibly S/G level deviation/failover logic
16	Both Steam Flow and Both FW Flow Signals Valid in Other Microprocessor	Other microprocessor	Possibly FW flow deviation/failover logic

Table 4-2 Digital inputs and outputs of the microprocessors (cont'd).

Digital outputs of microprocessors			
No.	**Description S/G #1 (S/G #2)**	**To**	**Processing**
1	Reserved for Watchdog Timer	Watchdog timer	Watchdog timer function
2	(unuseable)		
3	Power Failure or Microprocessor Not Controlling	MFV, BFV, FWP controllers	Failover logic
4	(unuseable)		
5	High Power Mode	Control room indicator	Indication
6	Transferring	Control room indicator	Indication
7	Low Power Mode	Control room indicator	Indication
8	Bypass Override Mode (BPO)	Control room indicator	Indication
9	Deviation to Plant Computer	Plant Computer	
10	Transfer Inhibit to Plant Computer	Plant Computer	
11	(Spare Output)		
12	Positioner Selected	Positioner	Positioner selection
13	No Failure in Microprocessor	Other Microprocessor	Deviation/failover logic
14	No Deviations	Other Microprocessor	Deviation/failover logic
15	Both Level Signal Valid	Other Microprocessor	S/G level deviation/failover logic
16	Both Steam Flow and Both FW Flow Signals Valid	Other Microprocessor	FW flow deviation/failover logic

Table 4-3 Inputs and outputs of PB4R digital signal Isolator.

No.	Description S/G #1 (S/G #2)	From	To	Processing
1	Watchdog Timer (Backup Microprocessor)	Backup Microprocessor	3 controllers	Pass through
2	Watchdog Timer (Main Microprocessor)	Main Microprocessor	3 controllers	Pass through
3	One Microprocessor Failed	BFV controller	Plant computer	Pass through
4	Both Microprocessors Failed	BFV controller	Annunciator	Pass through

Table 4-4 Inputs and outputs of MFV M/A controller.

Input	From	Output	To
CCI0- backup (B/U) PWR Fail/In-Test	B/U CPU	CCO1- A/M Status (B/U)	B/U CPU
CCI1- B/U Fail	B/U Watchdog timer	CCO2- B/U Fail	CPUs
CCI2- Main PWR/In-Test	Main CPU	CCO3- Main Fail	CPUs
CCI3- Main CPU Fail	Main Watchdog timer	CCO0- A/M Status (Main)	Main CPU
ANI1- Valve Demand (Main)	Main CPU	ANO0- Output to Valve	MFRV positioner, PDI controller, CPUs, other SG
ANI2- Valve Demand (B/U)	B/U CPU	ANO2- (S/G) Level Setpoint Output (from pushbutton)	CPUs, BFVcontroller (Display only)
ANI0- S/G Level (Display only)	Sensor	Through Microlink	
		FIX number	PDI controller
		Deviation alarm status and CPU failure status	BFV controller

Table 4-5 Inputs and outputs of BFV M/A controller.

Input	From	Output	To
CCI0- B/U PWR Fail/In-Test	B/U CPU	CCO1- A/M Status (B/U)	B/U CPU
CCI1- B/U CPU Fail	B/U watchdog timer	CCO2- Main and B/U Fail	Annunciator
CCI2- Main PWR Fail/In-Test	Main CPU	CCO3- Main or B/U Fail	Plant computer
CCI3- Main CPU Fail	Main Watchdog timer	CCO0- A/M Status (main)	Main CPU
ANI1- Valve Demand (main)	Main CPU	ANO0- Output to Valve	BFRV positioner, PDI controller, CPUs
ANI2- Level Setpoint (Display only)	MFV controller	Through Microlink	
ANI0- S/G Level (Display only)	Sensor	FIX number	PDI controller
ANI3- Valve Demand (B/U)	B/U CPU		

Table 4-6 Inputs and outputs of FWP M/A controller.

Input	From	Output	To
CCI0- B/U PWR Fail/In-Test	B/U CPU	CCO1- A/M Status (B/U)	B/U CPU
CCI1- B/U CPU Fail	B/U watchdog timer	ANO2- Bias Potential Excitation	Potential meter
CCI2- Main PWR Fail/In-Test	Main CPU	CCO0- A/M Status (Main)	Main CPU
CCI3- Main CPU Fail	Main watchdog timer	ANO0- Output to Pump	FWP controller, CPUs
ANI0- Pump Speed Demand (Main)	Main CPU	Through Microlink	
		Deviation alarm status and CPU failure status	BFV controller
ANI2- Bias Potential Meter Input	Potential meter (also output to CPUs)	Bias failure	BFV controller
ANI3- Pump Speed Demand (B/U)	B/U CPU	FIX number	PDI controller

Table 4-7 Inputs and outputs of PDI controller.

Input	From	Output	To
CCI0- Main Fail	MFV Failed position of HS-4516C	CCO3- Loss of Comm	Annunciator, plant computer
CCI1- Bypass Fail	BFV Failed position of HS-4516C	ANO0- Output to MFRV or BFRV	MFRV or BFRV
CCI2- Time Sync Input	external clock	Through Microlink	
ANI1- MFV Demand (circular buffer)	MFV controller	Date and time sync	3 controllers
ANI2- BFV Demand (circular buffer)	BFV controller		
ANI0- MFV dP	Sensor		

4.1.2 Deviations and Failover Operation

The DFWCS includes logic that monitors redundant input parameters for possible microprocessor input/output module failures or field transmitter failures and takes actions to notify the control room operator and minimize process perturbations. The logic consists primarily of deviation checks, out of range checks on redundant input parameters, and rate of change checks. The actions taken are based on the potential severity of the input failure modes and are tailored to the actual plant field transmitter configuration. A functional description of the basic failover process, as well as a detailed description of the S/G level deviation logic, are provided below. Deviation and failover logic of other input parameters, i.e., feedwater flow, steam flow, neutron flux, and feedwater temperature, are similar but *not identical* to those of the S/G level.

If an M/A controller is in automatic status, then the M/A controller will utilize the main microprocessor analog output signal. If the main microprocessor fails (CPU failure, power/software initiated failure, or test mode), then a bumpless and balanceless transfer to the backup microprocessor will occur. If an M/A controller is in automatic status and suddenly neither microprocessor is available, then the M/A controller will make a bumpless and balanceless transfer to the manual mode.

Figure 4-3 shows the S/G level deviation logic. The sensors' inputs are checked for out of range, bypass, and rate, and an input signal is considered valid if none of the conditions exists, and invalid otherwise. If both S/G level signals are valid and have a small/large deviation (difference), a deviation alarm is sent to the plant computer. If the deviation is small, the control continues. If there is a large deviation, after a delay the other microprocessor is checked for any error. If there is a large deviation in the level signals on the main processor and the backup microprocessor has no errors, then a failover (failing the main microprocessor and passing the status to the M/A controllers) is initiated. Otherwise, control is continued.

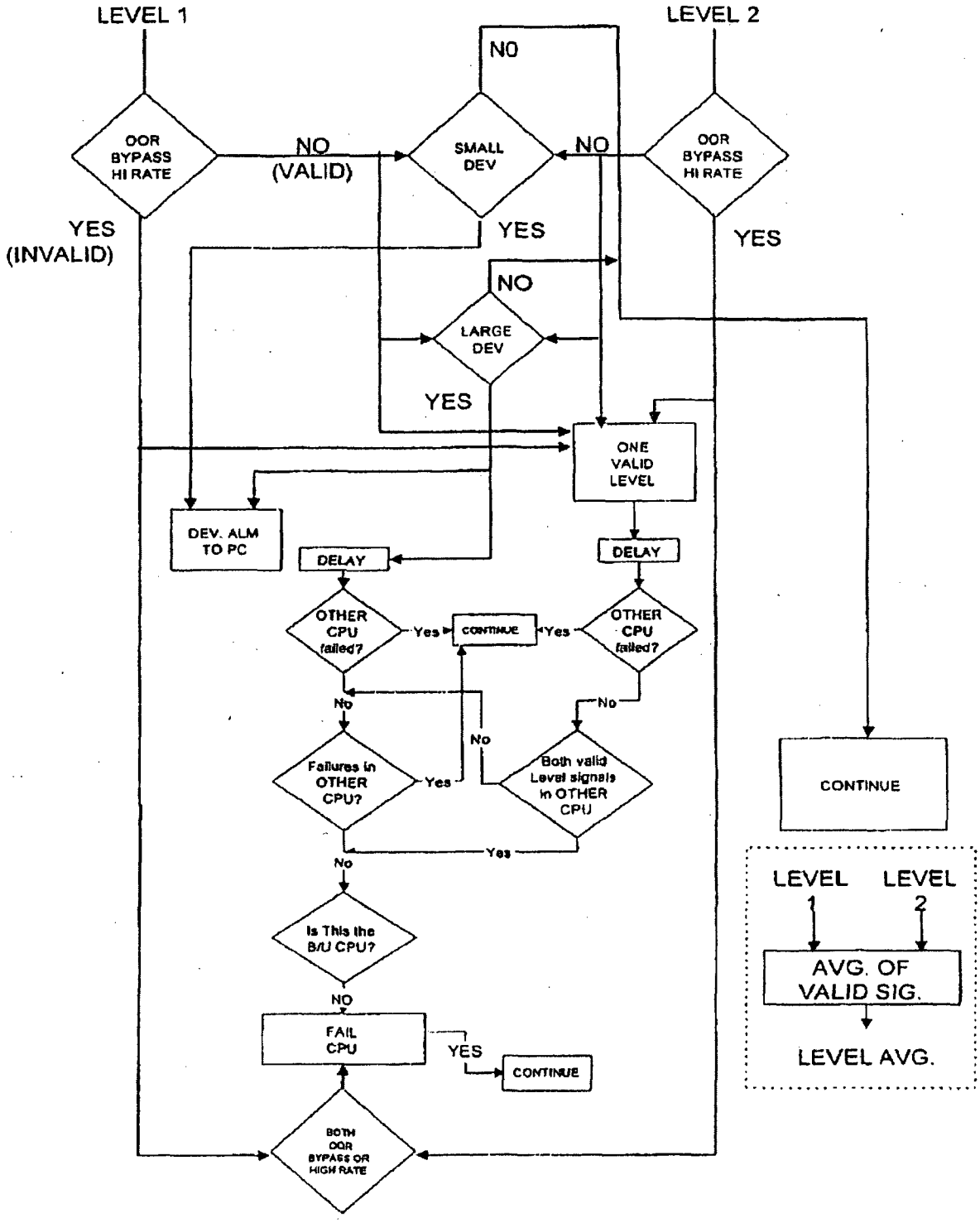


Figure 4-3 S/G Level deviation logic

If both level signals are invalid, a failover is initiated.

If one level signal is valid, after a delay the other microprocessor is checked for any error. If one level is invalid on the main microprocessor only, and the backup microprocessor has no errors or invalid level signals, then a failover is initiated. Otherwise, control is continued.

Signal rate errors will cause the signal to be automatically placed in bypass.

The above control logic allows detection of a transmitter failure by both microprocessors without affecting system operation. If an invalid signal is generated due to a failed analog/digital (A/D) converter, it will only be detected by one microprocessor and cause the microprocessor to be failed.

A delay is used to eliminate any race conditions in both microprocessors which could cause them to fail, causing the DFWCS to fail to manual unnecessarily.

4.2 Description of Azonix µMAC 7000 Controllers and Fischer & Porter 53MC5000 Controllers

4.2.1 Azonix µMAC 7000 Microprocessors

The µMAC 7000 is basically a standard 586 personal computer (PC) running Windows 3.1 16 bit applications. MSDOS 6.2, MS Windows 3.1, Azonix API DLL, and Azonix Link (for development work) are stored on a 20 mega-byte (MB) flash disk. Application programs can be developed with standard Windows tools, and ABB used MS Visual C++, Ver. 1.52C as the development environment.

The hardware consists of four main sections: PC, analog, digital 1, and digital 2. These sections are connected by an Industry Standard Architecture (ISA) bus. The analog section connects to one or more analog backplanes, the digital 2 section can connect up to 16 digital backplanes, and the digital 1 section can connect up to 15 digital backplanes, with the direct digital input/output (I/O) components in the section located on the internal µMAC 7000 board using an address slot equivalent to digital backplane number 16. The analog and digital backplanes are connected to the internal sections by three bus connectors located on the µMAC 7000 housing. The µMAC 7000 has limited diagnostic capabilities. Initialization diagnostics are the standard POST diagnostics of desktop PCs. One diagnostic function that was added is a missing module detection capability. This diagnostic function uses hardware circuitry and allows application programs to check for modules that have been configured by software but are not installed in an associated slot on the backplane. Azonix's intent for the function is to detect missing modules, and it will not necessarily detect an installed module that has failed.

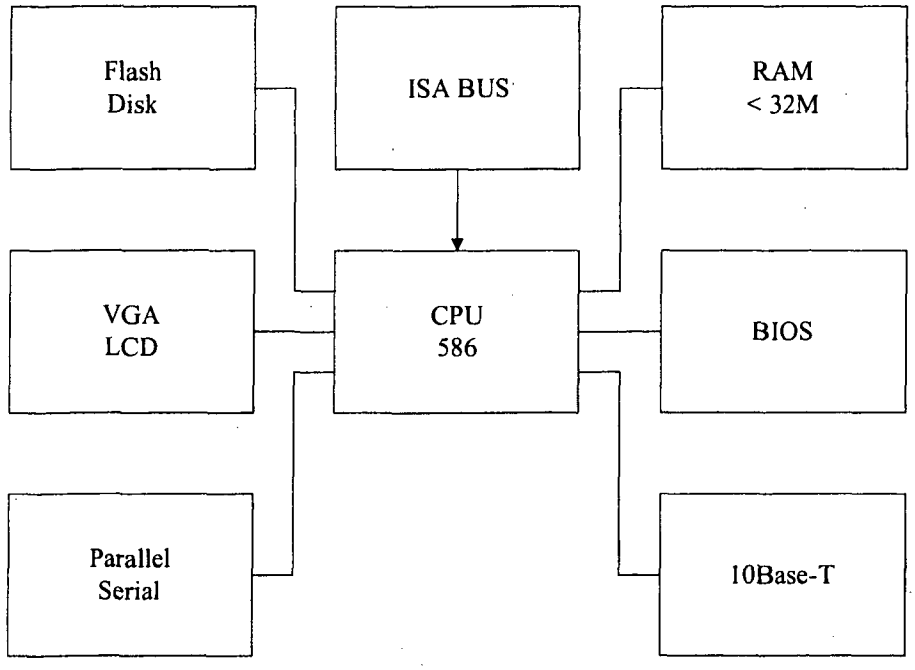


Figure 4-4 PC section – A standard ISA architecture

PC: The PC section as shown in Figure 4-4 is a standard ISA architecture. The CPU is an x586 part, and there is up to 32 MB of random access memory (RAM) on the board. Included on the board is a 10Base-T network adapter and a video graphics array (VGA) display adapter. A removable 20 MB flash disk is connected to the CPU via a standard integrated drive electronics interface. The µMAC 7000 analog and digital I/O sections are connected to the CPU over the ISA bus.

Communication with the µMAC 7000 can be over the network connection or may be accomplished using a serial port. Communication between µMAC 7000 and a Plasma Display Unit (PDU) is through a serial port of µMAC 7000. There are no plans to connect the µMAC 7000 to a local network for normal operation. Also, the VGA display and keyboard are not planned to be connected for normal operation but may be used during maintenance procedures.

Analog section: The µMAC 7000 analog section uses the ISA I/O bus to communicate to the PC section. The main analog parts are an A/D converter, a digital/analog (D/A) converter, a voltage reference (Vref), and Address logic. The A/D converter is a Burr Brown ADS 7805 that operates at 100,000 16 bit conversions per second. The D/A converter is a Burr Brown DAC 712 16 bit converter. The "Vref" is an Analog Devices (AD) 586 voltage reference.

Input signals to the system are connected from the analog backplanes, or Panels, to a multiplexer which switches the appropriate signal to the A/D. A D/A converter, a 5V source, and Vref source are connected to a demultiplexer (DEMUX). The Vref source is used during initiation to correct for voltage offsets in the input signal path. The 5V source and D/A signals can be used for limited diagnostics.

The D/A outputs are switched through the DEMUX to the panels, and control of the panels is handled by the analog address logic system.

Digital sections: The digital sections handle digital I/O to digital backplanes. There are two digital sections: digital 1 uses one address slot for internal Pulse Width Modulation/Pulse generators and can connect to 15 external digital backplanes and provide 448 digital I/O signals, while digital 2 can connect to 16 external backplanes and provide 512 I/O lines.

4.2.2 Fischer & Porter (F&P) 53MC5 Controllers

The controller is an 8051 processor on board an application-specific integrated circuit (ASIC) chip that performs a variety of functions. The application software of the controllers of the DFWCS was developed using F-TRAN, a proprietary reverse Polish interpretive language.

The Microlink interface, using RS485 hardware interface and Carrier Sense Multiple Access (CSMA) protocol, connects all four controllers. The MFV, BFV, and FWP controllers are date and time synchronized with the PDI controller within +/- 1 second on a one minute frequency, and communicate their function index (FIX) numbers (database point B00) to the PDI controller on a 1-second frequency. The MFV controller also sends its deviation alarm status and its microprocessor failure status to the BFV controller on a 5-second frequency. The PDI controller normally displays the differential pressure across the MFRV and has a buffer for holding the MFV and BFV controller outputs until the PDI controller can be manually switched into the control loop. If the MFV controller fails (the analog demand output goes to zero), the PDI controller, which is connected in parallel to the MFV controller, will automatically take over to support manual control of MFRV. If the BFV controller fails (the analog demand output goes to zero), the PDI controller which is connected in parallel to the BFV controller, can take over via a handswitch. The FWP controller communicates its deviation alarm status and its microprocessor failure status to the BFV controller on a 5-second frequency.

MFV controller: The MFV controller receives analog and contact inputs from the digital feedwater system microprocessors, performs some signal processing, and provides analog and contact outputs. It is the MMI for the digital feedwater system and is located on the main control board. It is also used to manually increase or decrease the S/G level setpoint for the entire DFWCS. See Figure 4-5 for the connection.

In the automatic mode, the MFV controller receives the valve demand signals from the main and backup microprocessors and forwards one of these signals to the MFRV positioner based upon the current microprocessor status. It performs deviation evaluation of the signals from the CPU, by checking if they differ by more than a setpoint, and sends the deviation status to the BFV controller via Microlink. When either of the two main microprocessor status digital input signals indicate that there is a problem with the main microprocessor, the main microprocessor availability digital output changes its state indicating that the main microprocessor is not available, and the demand signal from the main microprocessor is blocked and cannot be forwarded to the output. If the backup microprocessor is available, then the backup demand signal is sent to the output. When either of the two backup microprocessor status input signals indicate that there is a problem with the backup microprocessor, the backup microprocessor availability digital output changes state indicating that the backup microprocessor is not

available. If the main microprocessor is already unavailable, then the controller switches to manual mode.

The MFV controller also sends the CPU status to the CPUs.

In the manual mode, the operator uses push buttons to increase or decrease the output. This mode can also be entered by the auto/manual (A/M) pushbutton.

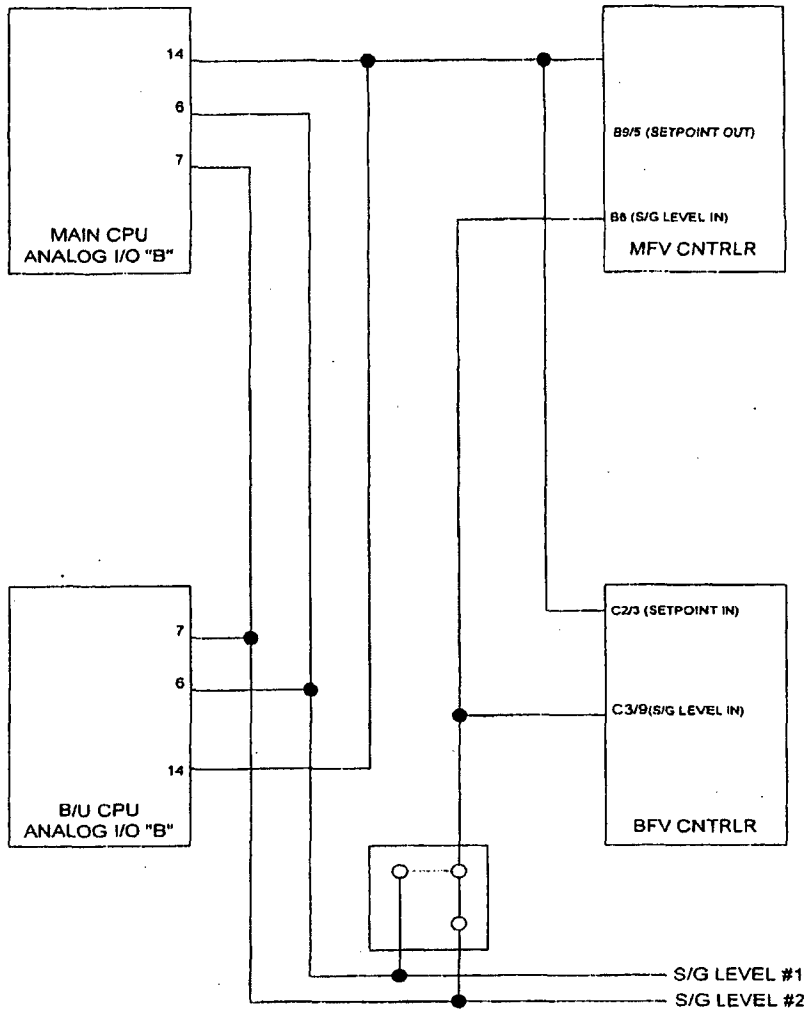


Figure 4-5 S/G level connections

BFV controller: The BFV controller processes the BFV demand signal in exactly the same way the MFV controller processes the MFV demand signal. An additional function it performs is providing alarms to the plant annunciator system and the plant computer based on deviation alarm status and microprocessor failure status received from MFV and FWP controllers through Microlink.

FWP controller: The FWP controller processes the FWP demand signal in exactly the same way the MFV controller processes the MFV demand signal. In addition, it has an analog input from the FWP speed bias potential meter mounted on the main control board. The bias is added to the feedwater pump demand by the microprocessor. This control will be typically used in adjusting the fraction of the feedwater flow through each of the two pumps, such as when starting or securing the second main feedwater pump or when matching the microprocessor output with the manual feedwater (FW) pump M/A controller output prior to switching the M/A controller from manual to automatic. The FWP controller monitors the rate of change of the bias signal. If the rate of change exceeds a preset limit, the FWP controller switches to manual mode, and a bias failure signal is sent to the BFV controller via the Microlink connection.

4.3 Digital Valve Controller and Speed Controller

4.3.1 Digital Valve Controller

Digital valve controllers have been used to replace analog current-to-pressure (I/P) positioners to control feedwater regulating valves (MFRV or BFRV).

A MFRV or BFRV was originally controlled by an analog input signal (4-20 mA dc) via an analog I/P positioner acting through a conventional I/P transducer and pneumatic output relay. Main feedwater valves used Fisher Controls model 3570 electropneumatic valve positioners with control valve assemblies to provide a valve stem position that is proportional to the input signal received from a control device, e.g., the MFV controller. The input signal range can be 0.2 to 0.1 bar (3 to 15 psig), 0.4 to 2.0 bar (6 to 30 psig), or another pneumatic input signal range [Fisher 2006].

The feedwater regulating bypass valves use Masoneilan Model 8012 electropneumatic valve positioners. These are force balance, electropneumatic devices which provide a means of obtaining a valve stem position directly proportional to a DC input signal. In addition, the positioner provides a means of split ranging the controller output signal. It may have either direct or reverse action on either direct or reverse actuators [Masoneilan 1997].

Regulating valves are currently controlled by Fisher Controls FIELDVUE Digital Valve Controller DVC5000 Series [Fisher 2001]. "Controller" and "positioner" will be used interchangeably here since the "controller" performs the same functions as an analog I/P positioner does. Although the digital valve controller contains firmware that implements the A/D and D/A conversions and control logic based on its input signal and the valve position feedback signal, it does not perform any complex automatic control function in this application. Its function is limited to that of positioning the regulating valve and maintaining the valve position, as directed by the DFWCS flow controller, and providing valve position indication.

The digital valve controller is a communicating, microprocessor-based current to pneumatic device. It converts the input current signal from the DFWCS to a pressure signal. In addition, the controller gives access to information on valve and actuator performance using the optional Highway Addressed RemoteTechnology (HART), communication protocol. When integrated with a HART communication based system, the digital valve controller user-configurable alerts provide real-time notification of current and potential valve and instrument problems. This can be done through FIELDVUE ValveLink software or the Rosemount Model 275 HART Communicator using an optional handheld communicator at the valve or at a field junction box, or by using a personal computer or a system console in a remote location, such as the full-range DFWCS panel or the control room [Fisher 2001].

The digital valve controller has one analog input signal between 4 and 20 mA DC. The positioner receives its input signal and power through a single twisted pair of wires brought into the terminal box. The input current signal is routed to the printed wiring board (PWB) assembly submodule where it is digitized and processed. In this application, the digitized signal can have many parameters applied, such as characterization, limits, etc., which are a function of previously established user defined parameters (programmed). The I/P converter is a conventional force-balance electropneumatic device, which coupled with a single conventional pneumatic relay, supplies the motive force to the valve actuator. The I/P converter transforms the input signal to a pressure signal. The pressure signal is sent to the pneumatic relay, where it is amplified and delivered as the output signal up to 95% of supply pressure to the actuator. The valve's position will be maintained in direct proportion to the analog control signal level.

The output signal to the actuator is also sensed by a pressure sensor located on the PWB and used for valve/actuator diagnostics. Stem position of the valve and actuator is an input to the PWB and used as a control feedback for the positioner. Mechanical gauges are provided to give visual indication of supply pressure and output pressure.

If the input current signal of the positioner falls below 3.5 mA DC or if the voltage level of the signal drops below 11.5V DC, the positioner will cease functioning. Once the signal current and voltage levels are restored to minimum values, the positioner will automatically restart (reboot) in less than one second.

As discussed above, the 4-20 mA DC input signal is used to power the positioner using the existing cable. A current/current (I/I) isolator (Devar Series 18-119, not a software based device) is installed between the DFWCS flow controller and the positioner to ensure adequate 4-20 mA drive current to operate positioners. Therefore, the loss of power to the positioner would only result from a loss of output signal from the DFWCS flow controller or the I/I isolator. Power for the I/I isolator is provided by an auctioneered 24V DC supply. As the signal level decreases (towards 3.5 mA DC or lower), the feedwater regulating valve will close and remain closed as the I/P positioner becomes non-functional, i.e., the FRV fails closed on a loss of control signal. When the input signal level is restored, if the DFWCS flow controller requires the feedwater regulating valve to be partially or fully open, it will send a signal in excess of 4 mA DC and the I/P positioner will respond accordingly.

It is known that the MFRV has two positioners and the controlling CPU will select one of them to control the valve. The available information of positioners does not tell how the CPU and the positioner interact with each other and this needs to be further investigated.

4.3.2 Lovejoy Speed Controller for FWP

The type of speed controller being used to control the feedwater pump is currently not known to the study team. Some information describing different Lovejoy speed controllers is available at http://www.lovejoycontrols.com.

4.4 Dependencies and Interfaces

4.4.1 Interfaces with Operators

Microprocessors: The PDU is the direct interface between the microprocessors and the operator. It displays system status, deviation status, event log, sensor values, and key system parameters. It also allows operators to change control setpoints and select MFRV positioners. The microprocessors also send DFWCS operating mode status to control room indicators, and deviation and transfer inhibit signals to the plant computer. The communication between the microprocessor and the PDU is implemented using a serial communication port of the microprocessor.

M/A controllers: From the M/A controllers, an operator can take manual control of the devices. In automatic mode, the operator can change S/G level setpoint through MFV controllers. The potential meter on the main control board allows the operator to specify the FWP bias used in calculating FWP demand signal. The bias is used in facilitating starting and tripping of a pump.

The M/A controllers also annunciate problems of main and backup microprocessors through the BFV controller, in addition to local alarms and indications, e.g., the MFV controller generates a deviation alarm when the main CPU demand signal differs from the B/U CPU demand signal by greater than a settable predetermined amount, after a time delay, and the BFV controller displays S/G level and S/G level setpoint without further processing them.

The PDI controller provides time and date synchronization and its own clock is manually set by the operator.

The M/A controllers have a watchdog function which causes the entire display to flash when analog outputs are not updated within a time limit, indicating a problem of the main printed circuit board.

4.4.2 Interfaces between Two Digital Feedwater Control Systems

As discussed in Section 4.1, the two DFWCSs share sensor inputs and exchange MFV demand signals that are used in calculating FWP demands.

4.4.3 Interfaces between Main and Backup Microprocessors of the DFWCS

The two microprocessors of the same DFWCS exchange digital information on microprocessor failure, deviations, S/G level signal validity, and steam flow and FW flow signal validity. No written description is currently available to the study team regarding how this status information is used; it is probably used in deviation and failover logic.

4.4.4 Interfaces between Main/Backup Microprocessor and M/A Controllers

The interfaces between the microprocessors and the M/A controllers are summarized in Tables 4-1 to 4-7.

Microprocessor: The microprocessors send demand signals to the MFV, BFV, and FWP controllers. When there is a power failure, software detected failure, or the microprocessor is in test, a signal is sent to the M/A controllers. In addition, an independent watchdog timer sends a signal to the M/A controllers when a time limit is exceeded without the watchdog timer being refreshed. It is capable of recognizing halting of the input scan routine and the application software.

M/A Controllers: The M/A controllers send their M/A status and output demand signals to the microprocessors. The M/A status is used by the microprocessors in automatic mode transfer logic and response to load change logic. The output demand signals are used by the microprocessors for tracking and deviation determination. Tracking is done by the microprocessor that is not in control, by setting the microprocessor output to the output of the M/A controller. The purpose is to facilitate a smooth transition when the tracking microprocessor is taking over control. A deviation between the output demand signal of an M/A controller and the demand calculated by the microprocessor in control indicates that there is a potential failure of the microprocessor, and the microprocessor should be failed by the software via de-energizing a relay.

In particular, the MFV controller sends the status of the microprocessors to the microprocessors. The status information of one microprocessor is used by the other microprocessor's deviation logic. It also sends to the CPUs the manually entered changes to the S/G level.

4.4.5 Interfaces Among the M/A Controllers

Through I/O connections-

MFV controller sends its valve demand to the PDI controller and the MFRV positioner. The BFV controller sends its valve demand to the PDI controller and the BFRV positioner.

Through Microlink-

The Microlink connection is described in detail in Section 4.5.1.

4.4.6 Interfaces with Sensors, Valves, and Pumps

The DFWCS receives analog sensor input signals and reactor trip and turbine trip signals, and sends analog demand signals to the MFRV and BFRV positioners and the Lovejoy turbine speed controller of the FWP. The valve positioners and turbine speed controller are also digital systems. They do not send signals directly to the DFWCS, except possibly for the MFRV positioners. The MFRV has two positioners, and normally only one positioner is actively running. Upon a large accumulated deviation between the valve demand and valve position

(linear variable differential transformer signals form the valve positioners), a failover logic of the microprocessor in control would automatically put the standby positioner in service. The microprocessors of the DFWCS receive a digital signal on the positioner selected, and allow manual selection of the MFRV positioners through the PDU interface. A digital output on the positioner selected is sent from the microprocessors to the MFRV positioners. Due to inadequate available design information associated with the MFRV positioner, the above description is based on an interpretation of available information.

4.4.7 Power Supply

Device controllers can be operated with either a +24V DC input or with AC inputs of 110/120V or 220/240V power supply [MicroMod 2004]. The power supply assembly to the device controllers of this DFWCS accepts two 120V AC instrument power buses. In the NPP where the DFWCS is installed, there are two non-safety related instrument 208/120V AC buses that supply power to the controllers. The two instrument buses are each powered by a separate 480V AC motor control center (MCC). Each instrument power bus enters a +24V DC power supply. The outputs from the two 24V DC power supplies are then diode-auctioneered to supply one 24V DC output to be used by the controllers. Lights on the assembly indicate which power supply or supplies are in use.

For the microprocessors, there are two 5V DC power supplies. One of the power supplies will provide power to the main CPUs of both S/Gs, and another one will provide power to the backup CPUs of both S/Gs. Therefore, it seems that there is no redundant power supply for the main CPUs or the backup CPUs, i.e., if the power supply to the main CPUs fails, the main CPUs will fail (but the power supply to the backup CPUs may still be working properly). Twelve (12) V DC power is required for operation of the internal fans.

4.5 Digital Features

4.5.1 Microlink Communication Issues

Generally speaking, Microlink is a Carrier Sense Multiple Access (CSMA) network permitting direct communication between any two nodes (peer-to-peer) or devices. CSMA is a probabilistic Media Access Control (MAC) protocol. "Carrier Sense" indicates that a node tries to detect the carrier wave from another node before attempting to send data. If a carrier wave is sensed, the node waits for the transmission in progress to finish before initiating its own transmission. "Multiple Access" means that multiple nodes may send and receive data in the network.

From the description of the communication of the DFWCS, the basic CSMA protocol is used, where a node (a device controller) only attempts to detect the carrier sense to avoid collisions. In the basic CSMA protocol, a collision may happen when two nodes try to send a frame at nearly the same time. In this case, neither node detects a carrier so both begin sending. The sending nodes do not detect collisions and will transmit the entire frame (thus wasting the bandwidth used). Since receiving nodes cannot distinguish between collisions and other sources of frame errors, the collision recovery relies on the ability of the communicating nodes

to detect frame errors and invoke an error recovery procedure. For example, the receiver may not send a required ACK (acknowledgment) signal, causing transmitters to time out and retry. This is implemented in the Microlink using a COMMAND (command packet from sending node)-RESPONSE (response message from receiving node) sequence. The Microlink COMMAND-RESPONSE sequence is called a transaction. In case of a collision, each sending node determines that the collision has occurred when a response is not received within a preset time. When the sending node detects the collision, it attempts a retry.

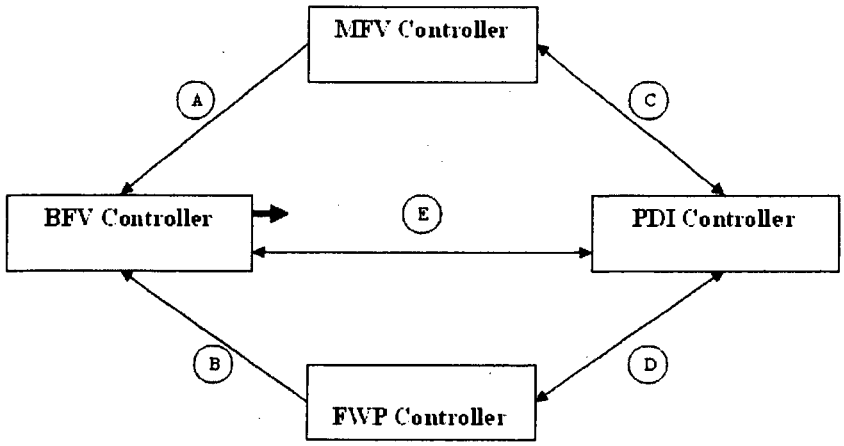
Microlink uses a serial RS-485 physical interface and can connect up to 32 nodes. RS-485 uses balanced line drivers (transmitting circuitry used to transmit data) and receivers (receiving circuitry used to receive data) to produce differential signals, i.e., the voltage produced by the driver appears across a pair of signal lines that transmit only one signal (RS-232 uses unbalanced line drivers and receivers). RS-485 permits the transmission line to be shared in a party line or multi-drop mode [B&B Electronics 1997]. As many as 32 driver/receiver pairs can share a multi-drop network. RS-485 specification does not define the network configuration and it is defined by the designers. RS-485 can be configured in a way such that the driver and the receiver can each use two wires or the driver and the receiver share two wires to send and receive data.

RS-485 is a hardware specification, and software protocol is not defined in the specification. The protocol to be adopted is also decided by the designers. The communication protocol used by the Microlink is briefly discussed here. Each message packet begins with a leading delimiter, and all initial characters are ignored by the interrupt service routine (ISR) until the delimiter is encountered. The second byte is an address byte that contains a 5-bit node address and a 3-bit command code. The third byte is the number of bytes to be read or written. The fourth and fifth bytes contain a 16-bit RAM address. After these bytes are the data. Finally, there is a checksum byte. The sending node will send an acknowledgment within a specified time interval after receiving the response from the receiving node.

The Microlink interface used in the DFWCS connects all four device controllers together. Regarding the exact structure of the Microlink communication network of the four device controllers, currently available information does not give completely consistent descriptions. Master/slave units are mentioned in some plant documentation. In addition, other plant documentation states that a query of FIX numbers from MFV, BFV, and FWP controllers is performed by the PDI controller. This seems to suggest that one of the device controllers is configured as a master node that has control of the RS485 transmit circuit, i.e., the PDI controller has a monitoring role by acting as the master node while the other three controllers act as slave nodes. However, an inspection of I/O diagrams of device controllers shows that the pair of transmitting and receiving wires (as labeled T+, T-, R+, R- in all diagrams of device controllers) are connected in a way that T+, T-, R+, and R- are connected together, respectively. Note that Microlink is based on a RS-485 physical interface and the communication module for each controller has a driver and a receiver. Each driver or receiver uses two wires. This suggests a peer-to-peer communication of the four device controllers using Microlink, i.e., any two controllers can communicate with each other. This also agrees with the way of exchanging information between device controllers. For the peer-to-peer communication, more collisions are anticipated than the master-slave communication. This is because in the peer-to-peer mode, any node is able to initiate communication, and in the master-slave mode, usually the communication between slave nodes is via the master node in the master-slave communication

configuration, especially for four-wired master-slave systems based on an RS-485 physical interface.

The information exchanged via the Microlink between device controllers is shown in Figure 4-6. The information will be communicated at regular intervals, which are also described in Figure 4-6. The general setting of the Microlink communication between device controllers is 9600 Baud and 0.25 Mbps.



→ : Alarms connected to the plant annunciator system and to the plant computer. Status will be changed according to the transmitted data from MFV and FWP controllers via Microlink interface;

(A): Deviation alarm status and Main and Backup CPU failure status (every 5 seconds);

(B) : Bias pot rate failure alarm, deviation alarm, and Main and Backup CPU failure status (every 5 seconds);

(C), (D), and (E) : Date and time synchronization, and FIX number (every minute);

Figure 4-6 Data exchange between device controllers

The MFV, BFV, and FWP controllers should be date and time synchronized with the PDI controller to within +/- 1 second on a minute frequency, and communicate their FIX number (database point B00) to the PDI controller on a 1-second frequency.

The MFV controller and the FWP controller should send their deviation alarm status and their CPU failure status (Main CPU Fail, B/U CPU Fail, and Main and B/U CPU Fail) to the BFV controller on a 5-second frequency. In addition, the FWP controller should communicate its bias failure alarm status to the BFV controller on a 5-second frequency. The impacts of these data will be presented by the BFV controller using its CCO2 and CCO3. CCO2 will change state after receiving the signal of Main and B/U CPU Fail from either MFV or FWP controller. CCO3 will change state after receiving (1) deviation alarm or Main CPU Fail or B/U CPU Fail from the MFV controller; or (2) deviation alarm or Main CPU Fail or B/U CPU Fail or FWP Bias Rate Fail from the FWP controller via Microlink. The above description is summarized in Table 4-8.

Table 4-8 Microlink communication summary.

Source controller	Destination controller	Microlink data	Communication frequency	Effects of received data
PDI	MFV	Date and time synchronization	1 Minute	MFV, BFV, and FWP controllers should adjust their own clock to and display the time passed by the PDI controller.
	BFV		0	
	FWP		0	
MFV	PDI	FIX numbers	1 Second	Unknown
BFV			0	
FWP			0	
MFV	BFV	Main CPU and B/U CPU fail	5 Seconds	Affects status of BFV CCO2 (connected to plant annunciator)
		Deviation alarm or Main CPU fails or B/U CPU fails	5 Seconds	Affects status of BFV CCO3 (connected to plant computer)
FWP	BFV	Main CPU and B/U CPU fail	5 Seconds	Affects status of BFV CCO2 (connected to plant annunciator)
		Deviation alarm or Main CPU fails or B/U CPU fails or FWP bias rate fails	5 Seconds	Affects status of BFV CCO3 (connected to plant computer)

Loss of communication is sensed via a communication status logic built into the PDI controller and via periodic (once a second) communication with three other controllers, where the alternate controller (the PDI controller) queries the FIX number of each of those controllers. Upon the loss of communication, CCO3 changes state actuating a DFWCS trouble alarm and CCO3 will be reset after communication is restored. For the DFWCS, the synchronized time on all device controllers is only used for display. Thus, a loss of the Microlink communication network affects alarm and time synchronization only, and does not affect control since CPUs and device controllers are asynchronously running.

Time delay may be caused by collisions and possible recovery actions, as discussed above. Therefore, real-time communication cannot be guaranteed using the basic CSMA protocol. In the DFWCS application, control related data are mainly sent and received via analog and/or digital I/Os. In spite of the unpredicted delay in the Microlink communication network, it is still possible that real-time data can be communicated if different protocols or different versions of CSMA protocols are used. A brief discussion is provided here.

There are other versions of CSMA, such as CSMA/CD, CSMA/CA, and CSMA/BA, which might be able to improve the communication performance. Carrier Sense Multiple Access/Collision Detect (CSMA/CD) is the protocol specified in the IEEE 802.3 standard for carrier transmission

access in Ethernet. On Ethernet, any device is allowed to send a data frame at any time. Each device senses whether the line is available to be used. If it is, the device begins to transmit its first frame. If another device has tried to send at the same time, a collision then occurs. The collision will be detected immediately and the transmission will consequently stop and the frames are discarded. Each device then waits a random amount of time and retries until successful in getting its transmission sent.

CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) is a protocol for carrier transmission in IEEE 802.11 networks. Unlike CSMA/CD, which deals with transmissions after a collision has occurred; CSMA/CA attempts to prevent collisions before they happen. In CSMA/CA, before the node tries to send a frame, it checks to be sure that there is no undergoing transmission and the channel is clear. If the channel is not clear, the node waits for a randomly chosen period of time, and then checks again to see if the channel is clear. This period of time is called the back-off factor, and is counted down by a back-off counter. If the channel is clear when the back-off counter reaches zero, the node transmits the packet. If the channel is not clear when the back-off counter reaches zero, the back-off factor is set again, and the process is repeated.

In CSMA/BA (Carrier Sense Multiple Access with Bitwise Arbitration), all of the nodes on the bus are assigned an identification number or priority code. When a collision occurs, one of the nodes that is attempting to send at the same time will be given priority to transmit according to its identification number or priority code. Therefore, waiting a random amount of time and then retransmitting will not happen, as in CSMA/CD.

Other general issues regarding communication include error detection and recovery mechanism. An example is sending an error frame from the node that detects the error first. However, one of the deficiencies is that once a node itself is at fault, it may cause the error frames to be sent all the time from other nodes. This effectively will block all the communication. The solution to this problem is to assign two error counters to each node. One is the transmit error counter, and the other is the receive error counter. Transmission failures and successes will increase or decrease the transmit error counter. The node can be in an error active mode, an error passive mode, or a bus off mode based on the values of the error counters. Therefore, the node at fault that constantly causes transmission errors will be finally taken off the network after a certain number of transmission failures. The faulted node will not block the communication between other nodes.

4.5.2 Watchdog Timers

A properly designed watchdog timer can be used to automatically detect anomalies and reset the processor or take other actions according to design. Generally speaking, a watchdog timer is based on a counter that counts down from a certain initial value to zero. The counter's initial value can be pre-selected and periodically restarted. If the counter ever reaches zero before the software or the CPU restarts it, the software or the CPU is presumed to be malfunctioning and the processor's reset signal or some other actions should be taken.

Usually, each processor contains a built-in watchdog timer while an external watchdog can also be designed to perform a similar function. A typical external watchdog timer design is shown in Figure 4-7. Note that the clock does not have to be shared by the processor and the watchdog.

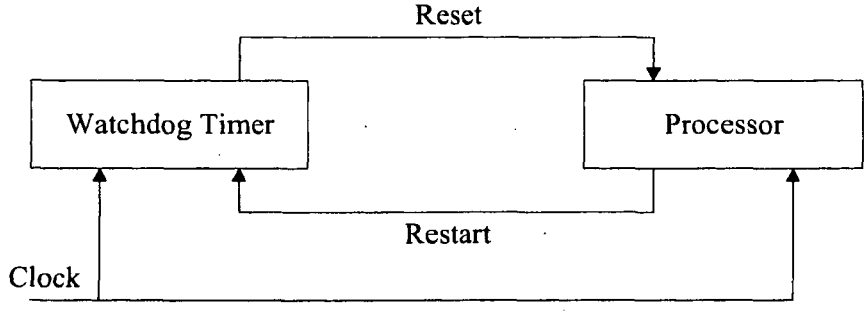


Figure 4-7 A generic design of watchdog timer

A common design is that output from the watchdog timer is tied directly to the processor's reset signal such that once the watchdog detects the malfunction of the processor, the processor will be rebooted.

The most important application of a watchdog timer is to get a system out of dangerous situations. A number of reasons may cause the system to hang, e.g., a logical fallacy resulting in the execution of an infinite loop. If this condition occurs to the CPU, none of the other software (except ISRs, if interrupts are still enabled) is able to run. Another reason is that an unusual number of interrupts arrives during one pass of the loop. Any extra time spent in ISRs is time not spent executing the main loop. Obviously, the delay is dangerous for a real-time control system. Furthermore, when multitasking kernels are used, deadlocks can occur. For example, a group of tasks might get stuck waiting on each other and some external signal that one of them needs, leaving the whole set of tasks hung indefinitely. The solution to all these problems can be provided by a watchdog timer.

In the DFWCS, two types of micro-processors are used. The device controllers use an Intel 8051, which is an ASIC, i.e., an Application Specific Integrated Circuit that aims at a particular use, rather than a general purpose. The other type is an Intel 80586 used in the Azonix µMAC 7000. For the device controller, the circuitry of a watchdog timer is also implemented using ASIC. The watchdog timer is reset by the power-on signal, which is asserted by the power supply after supply voltages have stabilized. The ASIC's design is partitioned so that if something in the core blocks fails causing the watchdog not to be reset, then the watchdog timer circuit still functions causing a flashing display.

The watchdog timer for the Intel 8051-based device controller is actually an external timer. If the watchdog timeout occurs, the display of the controllers will blink and the processors halt. The control task and the display task stop updating. The contact outputs go to "open" status and the analog outputs will become 0 mA. More details about the watchdog timer of the device controller are not available to the study team, but it is anticipated that this watchdog timer at least performs the function as a built-in watchdog of the Intel 8051 processor, if not more.

Azonix µMAC 7000 is basically a standard 586 PC running Windows 3.1 16-bit application. Azonix µMAC 7000 has a built-in watchdog timer, which uses a hardware timer on the interface

board. The watchdog timer will be activated by the application task and updated by the poll task (also referred to as the scan routine) at preset intervals, i.e., the watchdog will only be updated by the poll task. The application task mainly calculates the control values for the system and the poll task handles data input and output. The problem with this is that the application task may be hung, which is an undesirable situation since the application no longer controls the system, and the watchdog will not take any actions as long as the poll task is running. A solution to this is to adopt an external hardware timer to ensure that both the application task and the poll task are monitored by the watchdog.

The external watchdog timer is connected to a digital output on the Azonix µMAC 7000 interface board. Thus, the external watchdog will be updated by the output of the application task via the poll task that sends the output to hardware. Obviously, failure of either the application or the poll task will cause the watchdog not to be updated. In the case that the application task hangs, the watchdog timer will not receive the application output and will indicate the CPU failure after a certain time delay. Also, the poll task is monitored by the watchdog since it updates the watchdog timer hardware using the output of the application task. The application output will be sent to update the watchdog timer at the next 50 ms interrupt since it is performed by the poll task interrupt routine. The procedures to update the external watchdog timer are: (1) the application task outputs its data using an Azonix µMAC 7000 application program interface (API) function call; (2) the API function call updates the Driver Table (the application task and the poll task are linked in software through a Memory Area Driver Table) and sets a write flag in the table for the updated output variable; and (3) the poll task sends the updated outputs to the watchdog timer hardware at the next 50 ms MM_TIMER event.

The application task output used to update the watchdog is generated by the output-processing software, which uses the calculated analog and digital output values and writes them out to the MAC interface board via the poll task. For each cycle, the value of the output will be toggled. This watchdog output is connected to an optical isolator (PB4R) which performs conversions between electrical signals and optical signals and isolates the electrical coupling between the digital backplane bus and the external watchdog. The logic to determine whether the CPU fails is performed in the external watchdog timer, and the watchdog output will be changed accordingly. The watchdog output that indicates the CPU status will be sent to the device controllers (through the FWP controller). Furthermore, the CPU status will also be sent to the plant computer from the BFV device controller via the same optical isolator PB4R.

The external watchdog timer interval (within which there is no output of the application task received) should be less than 750 ms, which is based on engineering judgment and knowing that system stability (related to the fastest transient) is acceptable up to this value.

4.5.3 Software

The microprocessors run identical application software. Due to the designation of the Main and Backup status of the microprocessors, the two microprocessors actually run different parts of the software. It is when a failover takes place from the main microprocessor to the backup microprocessor that the backup microprocessor takes over control and runs the part of the software for the microprocessor in control.

The application software of the microprocessors interfaces with Windows services through API calls defined in the Azonix µMAC 7000 User Manual. Specific Windows API calls are employed, such as the timer or file I/O for saving and retrieving setpoints, when using particular operating system services.

The Azonix data acquisition system is the Azonix code which reads hardware inputs and writes corresponding values to a memory area driver table which is read by the application software of the microprocessors. The Azonix data acquisition system and the application software run asynchronously; that is, they use different timers, and have different cycle time of 50 ms and 110 ms, respectively.

The M/A controllers each perform its own control function, and their application software are different. The application software calls the real-time kernel and math library designed and tested by F&P.

4.5.4 Missing Module Diagnostics

The Azonix µMAC 7000 has limited diagnostic capabilities. Initialization diagnostics are the standard POST diagnostics of desktop PCs. One diagnostic that was added is a missing module detection capability. This diagnostic function uses hardware circuitry and allows application programs to check for modules that have been configured by software but are not installed in an associated slot on the backplane. Azonix's intent for the function is to detect missing modules, and it will not necessarily detect an installed module that has failed.

4.5.5 Cyclic Redundancy Check

Low level failures can potentially be detected and possibly corrected by cyclic redundancy check. The read only memory checksum of the F&P controllers and the checksum of the MicroLink communication are examples of this fault-tolerant feature.

5. FMEA OF A DIGITAL FEEDWATER CONTROL SYSTEM

This chapter summarizes the findings from the Failure Modes and Effects Analysis (FMEA) of the digital feedwater control system (DFWCS) performed at different levels. The study team used the FMEA to familiarize themselves with details of the system design; this formed the knowledge base for developing the reliability models of the system. Specifically, a detailed FMEA of the Main central processing unit (CPU) module was used to demonstrate how it could be performed in developing a Markov model of the DFWCS. Detailed FMEAs for the other DFWCS modules will be performed as part of the next task of this project. It is anticipated that the components of the other modules, such as the controllers, can be similarly identified and analyzed.

Section 5.1 provides a brief introduction to the work. The scope and level of detail of the FMEA are defined in Section 5.2. The approach adopted in this study is presented in Section 5.3. Section 5.4 summarizes major results from the FMEAs performed at different levels, and the insights gained are shown in Section 5.5. Issues with current FMEAs are discussed in Section 5.6, and Section 5.7 presents the conclusions from this study. Appendix B contains detailed FMEA tables.

5.1 Introduction

FMEA is a well-known method used to identify the failure modes of a system and their effects or consequences upon it. In this approach, failure modes can be categorized according to how serious their consequences are, how frequently they occur, and how easily they can be detected. Ideally, an FMEA begins during the earliest conceptual stages of design, and continues throughout the life of the product or service.

With regard to instrumentation and control systems, FMEA has been used by the nuclear industry and others, such as the defense-, automobile-, and chemical-industries, mainly for analog systems, i.e., ones that do not contain digital components, such as microprocessors. Some guidance for undertaking an FMEA is available, i.e., Institute of Electrical and Electronics Engineers (IEEE) Standard 352 [IEEE 1987], Military Standard 1629A [DOD 1984], Military Handbook 338b [DOD 1998], and the British Standard Institute 5760-5 [BSI 1991]. Typically, an FMEA is done using a top-down approach to a level of detail that is consistent with the objective of the study, subject to the limitations of design information and resources. For example, the British Standard describes the process of decomposition, and states that "...The usual requirement and purpose of an FMEA is to identify the effect of all failure modes of all constituent items at the lowest level in the system."

Since digital-based components and systems are being installed at nuclear power plants (NPPs), the nuclear industry has made efforts to extend the current methods to such systems. For example, the Electric Power Research Institute (EPRI) published the report "Generic Requirements Specification for Qualifying a Commercially Available PLC for Safety-Related Applications in Nuclear Power Plants," EPRI TR-107330 [EPRI 1996], requiring that an FMEA shall be provided of components in the Programmable Logic Controller (PLC) modules on the PLC performance, and referring to IEEE 352 [IEEE 1987] for guidance on how to carry it out. However, the standard does not have specific guidance for digital components.

For the DFWCS analyzed in this study, the hazard analyses performed by the plant were extensively used. The hazard analyses essentially are FMEAs. They include hazard analyses of the system, and those supporting the replacement of the main feedwater regulatory valve (MFRV) and bypass feedwater regulatory valve positioners. In the FMEA performed for this study, many other plant documents were also used. The FMEA for this study (documented in Appendix B) differs in a few areas from the hazard analyses undertaken by the plant, mainly due to a different understanding of how the system works.

5.2 Scope and Levels of Detail of FMEA

From the system description, when the plant is operating at power, the DFWCS can operate at either of two power modes, high or low power. During plant shutdown, the DFWCS operates at the low-power mode. For the demonstration purposes of this project, the FMEA was conducted only for the case where the plant is operating at power and the main feedwater is in the high-power mode. The scope of the FMEA encompasses the internal failures of the system, but excludes external events, such as fire or seismic events.

The three different levels of detail of the FMEA that were studied for this case are defined below. The lowest (third) level was chosen because it is the level at which most probabilistic data were available from publicly available sources, as discussed in Chapter 8. This level is more detailed than what has typically been used in other probabilistic models of digital systems in the literature. This level of detail is considered appropriate for analyzing the DFWCS because it is more capable of capturing the design features that potentially affect system reliability.

As discussed in Chapter 1, the estimation of risk from software faults is out of the scope of this study. Accordingly, the FMEA and reliability modeling in subsequent chapters mainly focus on hardware failures. In the FMEA, software failures are considered as a possible failure mode of a component that contains software. In other words, a component is "failed" when its software fails. Common cause failure (CCF) of software is considered in the same way. Possibly there are interactions between hardware and software failures, i.e., some hardware failures may lead to software failures and vice versa. For the purpose of this analysis, hardware and software were assumed to fail separately; more detailed research is needed to study these interactions.

5.2.1 System Level FMEA

For the system level (top-level) FMEA, the scope of analysis included the whole DFWCS system.

5.2.2 Module Level FMEA

The next level of the FMEA included the modules of the DFWCS, with the major ones being the Main CPU, Backup CPU, Main Feedwater Valve (MFV) Controller, Bypass Feedwater Valve (BFV) Controller, Feedwater Pump (FWP) Controller, Pressure Differential Indication (PDI) Controller, and the optical isolator that is related to the watchdog timer (WDT) signal. Failures of individual input and output signals of the major modules and their impacts on the behaviors of the modules were analyzed. The input and output signals directly reflect the failure modes of these major modules.

5.2.3 Major-Component-of-Module Level FMEA

The lowest level FMEA analyzed the components inside the modules of the DFWCS. For example, the major components of the module of the Main CPU (and thus the Backup CPU) include the analog and digital backplanes, multiplexer and demultiplexer, analog/digital (A/D) and digital/analog (D/A) converters, current loop devices, digital input modules, buffer, digital output modules, address logics, random access memory, BIOS, flash disk, serial port, and the central processing unit. The Main CPU model is used as an example of how the lowest level FMEA can be performed. The controllers are Application Specific Integrated Circuit-based devices, but they are expected to have similar major components and can be analyzed in a comparable way.

Failure parameters for these major components are required to assess reliability of the digital system once the reliability models have been created. The development of reliability parameters of digital systems or components is discussed in Chapter 8, which provides details about how to obtain them. Another important parameter is the distribution of the failure modes of these components, since different modes may have different effects on the modules. The failure mode distributions described in Meeldijk [1996] and RAC [1997b] were used. Similar information is available in DOD [1998]. The failure parameters and failure mode distributions are used to estimate the rates of failure for different component failure modes of the Main CPU module.

The lowest level FMEA was performed only for the Main CPU module of the DFWCS, but the same approach can be used to analyze its other modules. The findings support the construction of the Markov and event tree/fault tree (ET/FT) models described in Chapters 6 and 7, respectively.

5.3 FMEA Approach

The FMEA of a system can be conducted at different levels of detail, that is, from a top-level description to more detailed one, which might describe how the microprocessor functions in terms of its hardware architecture, its software, and also the ways in which the microprocessor achieves these functions via its hardware and software. As discussed above, the DFWCS's FMEA consists of "decomposing" the system into three levels, and carrying out an FMEA at each level. The first level analyzes the entire system at a coarse level, while each successive level involves more detailed resolution. The FMEA at the previous level can be used in performing the next level FMEA since the failure mode of a specific level represents the effects of failure at its immediate lower level. In this study, the failure modes at the module level are the causes of the failure modes at the system level.

Decomposition continues until the information available cannot support a more detailed analysis, or the purpose of the FMEA does not require a more detailed analysis. In general, the more detailed the analysis, the more can be learned about possible failures of the system, but the more costly the analysis becomes. Selecting each level of analysis in the decomposition is somewhat arbitrary but generally depends upon the purpose of the FMEA. The level selected should match the system's major architectural blocks or components for which information is available. In this study, the lowest level FMEA, i.e., the major-component-of-module level, was used to support the development of the module-level Markov and ET/FT models of the Main CPU.

In addition to the complexity of digital systems, the flexibility and interactions between hardware and software are difficult to capture with high-level FMEA. For the FMEA of the DFWCS, first a top-level analysis was conducted, that is, at the level of the entire system, followed by a lower level analysis of the inputs and outputs of the major digital modules of the DFWCS, such as the Main CPU and MFV controller. Since some components of digital systems, such as microprocessors, make decisions, and their hardware and software usually implement relatively complex algorithms, a detailed analysis of them is required. Hence, these digital modules are further decomposed into digital components or parts, and the FMEA obtained at this level is used for Markov- and fault tree-modeling, as illustrated in Chapters 6 and 7, respectively. More detailed analyses at deeper levels can be carried out, but they were not necessary for the purpose of this "proof-of-concept" study.

There is no universally agreed-upon definition of the failure modes of digital systems/components failure modes. One way to do so is based on their functionality. In this FMEA, the identification of the failure modes of a digital module or component is based on its input and output signals. This is because the status of these signals directly reflects whether the desired functions can be accomplished correctly.

This FMEA and the plant hazard analyses typically analyze one failure at a time. In many cases, the DFWCS can tolerate one failure without serious consequences. However, after considering two or more failures in some cases it was found that the consequences could be significant. It is advisable to analyze the system (using FMEA and/or other methods) to study the impact of combinations of failures. In particular, it is important to assess the impacts of CCFs.

Consistent with the plant hazard analyses, this analysis assumed that a postulated failure would not propagate through any physical connections between components. This supposition is equivalent to considering that the local failure is physically isolated from the input to other controllers. It is a fundamental assumption for this analysis. To evaluate and verify that it is correct, the physical circuits must be examined and detailed information about them obtained. An example of a digital input-and-output connection and the related issue is described here.

Two or more input modules can be connected to a single output module; Figure 5-1 illustrates the possible configuration wherein the input pins are directly connected to the output pins. A problem arises when a short-circuit analysis of the Digital Contact Input 1 is performed, i.e., Digital Contact Input Fail Closed (the dashed line inside Digital Contact Input 1 in Figure 5-1 represents a short). In this FMEA, this failure is assumed to be isolated to Digital Contact Input 1 and does not affect the status of Digital Contact Input 2. However, a close inspection of the figure reveals that this might not be true because the pins of Digital Contact Input 2 are also connected to those of Digital Contact Input 1. Thus, the short inside Digital Contact Input 1 may also fail Digital Contact Input 2.

The scenario postulated here also is applicable to a cascaded connection of analog signals. However, the postulated scenario might not occur if alternate designs are selected, e.g., two sets of switches controlled by the same relay can be used to connect the two input modules. However, the scenario was postulated due to the lack of detailed information about the physical circuits. This underscores the fact that a careful examination of the design is necessary to generate a valid FMEA.

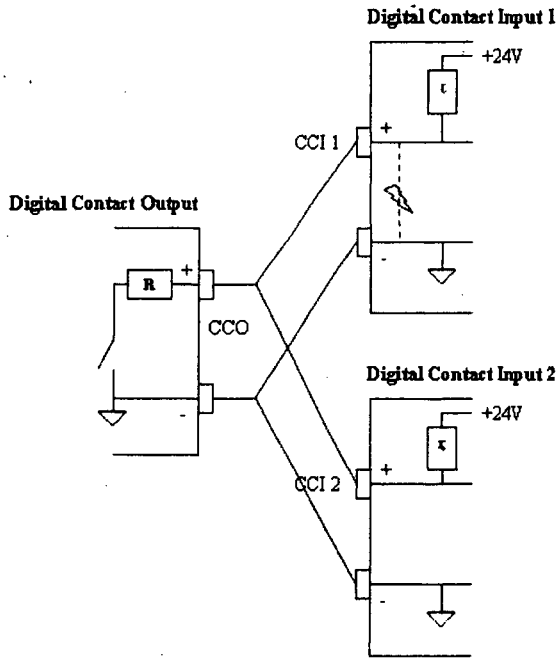


Figure 5-1 Connection of digital inputs and outputs

5.4 Summary of FMEAs at Different Levels

5.4.1 FMEA at Top-Level of DFWCS

The failure modes of the DFWCS and their impacts on the entire main feedwater system are analyzed at this level. All failure modes that were considered feasible were analyzed; it was not verified that there is a mechanism or cause of failure for each failure mode. Control room indications of the failures, and potential failure effects, e.g., loss of feedwater and reactor trip, were identified.

5.4.2 FMEA at Level of DFWCS Modules

An FMEA was also conducted at the level of the digital modules of the DFWCS. This analysis was carried out for each input and output of the Main CPU, MFV Controller, BFV Controller, FWP Controller, PDI Controller, and the Optical isolator that is related to the external watchdog timer. All the inputs and outputs of the Main CPU are through the Analog Backplane and Digital Backplane. Therefore, the FMEA of the analog and digital backplanes is actually considered the FMEA of the Main CPU.

In carrying out the low-level FMEA, the following were considered:

1. The term CPU or microprocessor refers to a CPU of the DFWCS, not the on-board processor inside one of its controllers.

2. There is one DFWCS for each of the plant's secondary loops. It is assumed that initially the DFWCS is in automatic high-power mode with all system components normally running, and that the Main CPU is controlling the DFWCS.

3. The MFRV has two positioners, A and B. The Main CPU controls one of them, called the active positioner. There are two "Diagnostic Transfer" modes in the CPU related to the MFV controller; normal (enabled) and lockout. When the Diagnostic Transfer is in lockout mode, no auto transfer can occur. The initial conditions for this analysis were that the active positioner is A, and that the Diagnostic Transfer is in the normal (enabled) mode.

4. The failure analyzed is the only one presumed to have taken place. That is, the effects of the failure were determined by assuming that everything else works as designed. Some extra cases thought to be of special interest were analyzed, e.g., when an additional failure posed a greater challenge to safety. In these cases, the additional failures were postulated and the associated response of the system was evaluated.

5. For this analysis, the identification of the failure modes of a digital system module or component is based only on its input and output signals. Failure of the entire digital module or component is not considered, though the effects of this type of failure are represented by some of the failure modes included in the FMEA, e.g., failures of CPU outputs represent the effects of the failure of the CPU.

6. It is also assumed that signals can be isolated such that one postulated failure mode of a signal does not physically cause other signals to fail. For example, the Main and Backup (B/U) CPUs share sensor inputs. When modeling the failure of a sensor input to the Main CPU, it is assumed that the sensor input to the B/U CPU is not failed.

A difficulty encountered in the FMEA process was that in a few cases the detailed control logic could only be determined by review of the software. Due to the level of effort that would be required, in the majority of these few cases the logic of the software was not traced, and assumptions were made.

Tables B.2-1 to B.2-8 of Appendix B.2 show the detailed FMEAs at the level of the DFWCS modules.

5.4.3 FMEA at Level of Major-Component-of-Module of DFWCS Main CPU Module

The results of the FMEA are used not only to better understand the DFWCS, but also to eventually provide information for the reliability modeling of the DFWCS using the ET/FT and Markov approaches.

An obvious issue of creating a reliability model based on the FMEA information given in Sections 5.4.1 and 5.4.2 is the difficulty in acquiring the associated reliability parameters. In the FMEA at the level of digital modules, the failures of input/output signals cannot be quantified

because they can be due to many factors. Without analyzing how the signals fail, the associated parameter of a specific failure mode (e.g., a specific signal of the modules) cannot be determined.

Information on the structures and components of a digital module in the DFWCS suggests that a deeper-level FMEA be necessary for such reliability modeling. An FMEA at the level of major-component-of-module of the DFWCS was performed in the present study for the Main CPU module only. Similarly detailed FMEAs for the other DFWCS modules will be performed as part of a subsequent task of this project. It is anticipated that the components of other modules, such as the controllers, can be similarly identified and analyzed. However, less information is currently available to the study team on the controllers as compared to the CPUs; therefore, there may be a different level of completeness in the FMEA for the controllers.

In the FMEA of the Main CPU module, the module was broken down into its individual digital components, the major ones of which were identified in Section 5.2.3. FMEA of each component then was conducted to determine the impacts of failure on the component, the detectability of the failure, and the associated effects on the Main CPU module, i.e., the failure modes of the Main CPU module. Then, the failure rates of the components were estimated using the generic data from Chapter 8. These estimations mostly were based upon a Hierarchical Bayesian method with raw data extracted from the PRISM database [RAC PRISM]. Because of the lack of validation and the large uncertainties, the failure rates obtained from the analysis are not appropriate for use in quantifying reliability models for purposes of making decisions. They are used in this study only to demonstrate the usefulness of the model developed for the DFWCS. The failure rates of different component failure modes were calculated using the failure mode distributions from two sources [Meeldijk 1996] and [RAC 1997b]. Military Handbook 338B [DOD 1998] is another source of failure mode distributions. It is noted that the failure mode distributions are simply tabulated in these references with no information on how they were estimated. Many of the distributions contain failure modes that are failure causes and failure mechanisms, and can not be used. Those failure distributions that contain failure modes fitting the failure modes defined in this study were used. Table B.3-1 of Appendix B.3 summarizes the FMEA.

The component failure modes were further grouped, based on their failure effects on the Main CPU module, into the failure modes of the Main CPU module listed in Table 5-1. These failure modes form the Markov model of the Main CPU, which is further discussed in Chapter 6. Each failure mode in Table 5-1 is defined as a state of the Main CPU Markov model. The failure rate of each of the module failure modes in the table can be estimated by summing the failure rates of the component failure modes that were grouped into the module failure mode.

In this study the main purpose of the FMEA at the component level of the digital modules is to create Markov and ET/FT reliability models. Only the failure effects on the Main CPU module were considered. The effects of the failure upon the whole system depend on the combination of the states of different system modules.

Table 5-1 Failure modes of the main CPU module.

Failure mode
WDT detectable failures
Software detectable failures
Continued operation with latent failures
Main CPU Tracking
Undetectable failures

5.5 Insights Learned from the FMEA

The design, operation, and response to the postulated failures of the digital control of the MFW system were studied. FMEAs of this system were carried out at three different levels, i.e., at the level of the entire system, at the level of the inputs and outputs of the system's digital modules, and at the level of the components of each digital module.

This analysis of the DFWCS revealed the considerable complexity of the design of this system. Each of its components, such as each CPU and controller, has embedded software and many inputs and outputs all of which are interconnected. The general insight from undertaking these FMEAs is that, due to the great complexity of the DFWCS, it is very difficult to reliably predict the response to, and effects of, an individual failure. If several failures are analyzed concurrently, the analysis becomes even more difficult. Accordingly, an important insight is that this process is an excellent tool for learning about and understanding the design, the operation, and some possible safety weaknesses of the system. On the other hand, another insight is that FMEA, by itself, is not a sufficient tool to determine how specific component-level failure modes affect a complex digital system. Hence, it is advisable to employ/develop other more sophisticated tools, e.g., an integrated simulation model that simulates the operation of the DFWCS, including execution of the software, to analyze the interactions between the components of a digital system and the effects of one or more failures. Ideally, the FMEA and these tools would be used in combination to identify the vulnerabilities of the system in a more reliable way than when using the FMEA alone.

While performing the FMEA, the information available from the plant was used, especially the data contained in the plant's hazard analysis. In several cases, the analysis of the effect of a failure mode differed from the plant's hazard analysis. These differences will be addressed in a subsequent task of this project.

Some insights about the design and operating features of the DFWCS also were obtained from the exercise of performing the FMEA. The controllers of the DFWCS share information about the status of some components of the system, such as the failure status of the CPUs. Apparently, this information is only used by the BFV controller to send signals of trouble alarms to the plant's annunciator system and computer. If this is true, it is suggested that this information be used to cross-check the information that the controllers receive from other sources, such as the status of the CPUs, which the controllers receive from the CPUs and the watchdog timers.

The digital control system has a microprocessor that controls the main feedwater of one secondary loop of the plant. The microprocessor receives digital input signals through a "digital backplane." One of the input signals is a "reactor trip" signal, i.e., a signal indicating that a reactor trip has occurred. The microprocessor receives this signal through a digital contact associated with channel 19 of the digital backplane. If this contact fails open, the microprocessor will receive a signal that a reactor trip occurred, even though a trip did not happen. Hence, it is assumed that if this contact fails open, a reactor trip will occur. This trip is undesirable because it is a challenge to the safety of the plant and an economic loss for the plant owner. A different design could possibly avoid this situation.

Similar to the previous point, one of the digital input signals to the controlling microprocessor is a "turbine trip" signal. The microprocessor receives this signal through a digital contact associated with channel 21 of the digital backplane. If this contact fails open, the microprocessor will receive a signal that a turbine trip occurred, even though it did not actually occur. Therefore, it is assumed that if this contact fails open, the microprocessor will send a signal to the main feedwater regulating valve to ramp shut that, in turn, will cause a reactor trip. As discussed above, this trip is undesirable and may be avoidable with a different design.

The FMEA of PB4R (an optical isolator) is included in this study because the WDT signal that passes through PB4R is used by the WDT to determine the status of the microprocessors (CPUs). The WDT signal from the CPU toggles every cycle and passes through the PB4R as an input to the external WDT. According to plant information, if the WDT receives the low signal (contact closes) from PB4R within a preset period, the watchdog will be reset and there will be no watchdog timeout, i.e., the CPU is considered to be working properly. Otherwise (high signal because contact becomes open), the WDT will timeout and signal the three device controllers that the CPU has failed. Therefore, if the controlling CPU WDT signal fails closed or low at PB4R, and the controlling CPU truly fails, the WDT will be unable to notify the device controllers of the failure of the controlling CPU. This is an undesirable situation because the system is controlled by the failed CPU, and the operator will not receive any alarms about its failure until the loss of control is noticeable.

The communication between the manual/automatic (M/A) controllers uses the basic Carrier Sense Multiple Access (CSMA) protocol that has the potential for an unlimited delay in communication due to a faulty node on the network. If there is unlimited delay, there would be a permanent loss of communication between the controllers. Pinho [2000] suggests a solution that would remove the faulty node from the network after a certain number of transmission failures. The faulty node will not permanently block the communication between other nodes. More advanced CSMA protocols that are more able to avoid collisions are described in Chapter 4.

The PDI controller serves as a backup to the MFV controller. It receives the MFV controller's output signal, and upon failure of the signal, will take over control and become a manual controller of the MFRV by sending the pre-failure MFV controller signal to the MFRV. Assuming that the analog input signal to the MFV controller fails to zero, the MFV controller will forward the failed signal to its output, and the signal will be received by the PDI controller and the CPUs. With this failure, plant information states that the main CPU would detect the signal's deviation and initiate a failover. The PDI controller has a scan time of not exceeding 100 milliseconds, while the CPU failover has a 1-second delay. Therefore, the PDI take over will take place first and prevent the CPU failover. The issue also applies when the analog output of the main CPU is assumed to have failed.

The FMEA of components inside a digital module shows that failure of a specific component, e.g., A/D converter, will result in a loss of all analog signals to the Main CPU. This finding prompts analysis of multiple failures, although as discussed, the undertaking might be difficult.

In the FMEA, the indications and alarms available to alert the operator after the postulated failure modes are tabulated. The plant uses a graded approach on such indications and alarms. That is, depending on the severity of the failure, different indications and alarms are used, i.e., local alarm at the controller, alarm to the plant computer, and annunciator in the control room. When the FMEA at a detailed level was carried out, it was noticed that the indications and alarms that may result often are not directly indicative of the postulated failure mode; instead, they indicate the subsequent effects of the postulated failure.

While it is outside the scope of this project, the estimation of risk from software faults is a major issue in the safety evaluation of digital systems. The CPUs use the same software, and each controller uses a different software. Failure of software may fail its associated component, such as a controller, and may cause a failure or degradation of the entire system. Since the CPUs use the same software, a failure of the software may fail or degrade both CPUs. A detailed study and evaluation of the system's different software would be useful in identifying the applicable software-related failure modes.

5.6 General Issues Associated with FMEA of Digital Systems

A few general issues associated with performing an FMEA of digital systems are summarized below.

Specific guidance about how to perform FMEA of digital systems appears to be lacking (at least in the public domain). IEEE 352 and other publications describe the method of FMEA and it is not repeated here. They offer generic guidance on FMEA, but no specific guidance on FMEA of digital systems. An even bigger issue is that there is no generic or standard list of failure modes of digital systems/components. As discussed in this report, an FMEA was performed at different levels and different failure modes were identified at each level. However, it is possible that an FMEA of this same system by other analysts might result in a different set of failure modes.

For a given failure mode, it takes intensive efforts to postulate and determine the failure effects. It is difficult for the traditional FMEA method to handle the complexity of digital systems. Currently, in the FMEA, a failure mode is first postulated. Sometimes its immediate impact can be easily identified, but generally this is difficult due to interactions inside the digital systems and/or interaction between the digital system and the plant. A simple example is the existence of a feedback signal in a control system. Without considering the controlled process, FMEA can not evaluate the transient responses. For the FMEA of components inside modules, extra problems are posed in analyses because sometimes knowledge of both application software and system software is required. For example, the serial port is the device that enables the plasma display unit and the Main CPU to communicate. The serial port is accessed by the application software of the CPU that calls the Windows subroutines. It is difficult to draw conclusions about effects of failure of different failure modes of the serial port without knowledge of both application and support software.

In addition, a single component failure might not affect the operation of the system at all. It might also be undetectable until one or more additional failures occur. At that point, the combination of the two or more failures may be detected or may cause the system to fail. The analysis of multiple failures greatly complicates the FMEA.

Complete identification of the failure modes for the components of a digital system requires a detailed understanding of system design and operation. Due to the complexity of digital systems, this can be very resource intensive. In addition, some detailed information on a digital system may be difficult to obtain, or different sources of information may provide contradictory information. In particular, to completely understand some aspects of the system control logic, it may be necessary to review the source code.

5.7 Concluding Remarks

FMEAs of the DFWCS were performed at different levels to better understand its design and operation for creating reliability models of the system. Failure modes at different levels were defined and analyzed based on available information using the approach described here. It appears that a better design might be achieved using the FMEA as an analysis tool, as discussed in the insights learned from the FMEA. Chapters 6 and 7 describe how the FMEAs discussed in this chapter and Appendix B support the reliability modeling of digital systems using the traditional Markov and ET/FT methods, respectively.

6. DEVELOPMENT OF A MARKOV MODEL OF THE DIGITAL FEEDWATER CONTROL SYSTEM

Chapter 4 describes the digital feedwater control system (DFWCS) that is discussed in this chapter. The system consists of two identical control systems, one for each steam generator. Here, only one such system is considered, assuming that the loss of control of one system is an initiating event (as discussed in Chapter 3). The interactions between the two systems will be explored, e.g., sharing of the main feedwater valve (MFV) Tracking signals. The system model includes sensors, central processing units (CPUs), watchdog timers, controllers, valve positioners, and pump-speed controllers. The approach for developing a Markov model is described in the following sections.

In Chapter 5, the entire DFWCS system was decomposed into a level of major modules and a deeper level of major components of the modules, and Failure Modes and Effects Analyses (FMEAs) were performed at these three different levels. The major component level FMEA is needed because that is the level at which generic component failure data are available. Developing a Markov model at the lowest level would be too complicated. Instead, the failure modes of the major components of a module can be grouped into module-level failure modes, based on their effects on the module, and the grouped failure modes define failure states of the module. The failure rate of a failure mode at the module level is simply the sum of the failure rates of the component failure modes that were grouped. In Chapter 5, the main CPU module was used as an example to demonstrate the above approach.

Markov states at the system level can be defined in terms of combinations of the states of the modules. The system level Markov model defines transitions among system states including "failed" states in which automatic control of the feedwater system is lost. With an initial condition that the components of the system are operating normally, the Markov model can be solved to obtain the probabilistic behavior of the system including the probabilities of the failed states. The system's failure states can be further divided into those in which manual control subsequently is possible, and those in which it is not. Manual control can then be modeled accordingly.

As discussed in this chapter, a Markov model of a system can be complicated and difficult to solve. It is believed that a detailed Markov model can capture the relevant characteristics that contribute to the reliability of the system. Once developed, such a model can potentially be used to develop simpler models of the digital system that are easier to work with and solve than the complete model. This chapter describes how a Markov model of the DFWCS should be developed and solved in detail.

Section 6.1 describes how module-level Markov models can be developed using the example of a CPU and extensively employing the FMEA conducted on the CPU (see Chapter 5 and Section B.2.1 of Appendix B for a description of this FMEA).

Section 6.2 then describes how a system-level Markov model can be developed based on module-level Markov models, taking into account the interactions/connections between the modules and the control logic implemented in the software. Examples of system-level transitions are discussed.

Section 6.3 details the modeling of software failures in the Markov model of the DFWCS.

Section 6.4 offers an approximate way of building and solving the Markov model by identifying and quantifying single failures first, double failures next, and so on, until the results converge.

6.1 Development of Module-Level Markov Model

This section discusses the development of a Markov model of the Main CPU module in detail. The approach can similarly be applied to other modules of the DFWCS, as will be done in a follow-up task of this project. The module-level Markov models form the foundation of the system-level Markov model discussed in Section 6.2.

6.1.1 A Markov Model of the Main CPU

Figure 6-1 shows the "internal" components of the Main CPU module of and modeled in the Markov model as internal parts of the Main CPU. In the diagram, Analog Backplanes and Digital Backplanes interface all inputs and outputs of the Main CPU module. A standard ISA bus is used for the CPU (central processing unit of the Main CPU module) to interact with components of backplanes. "C. L." represents a current loop device that produces a 0 - 20 mA current output. It is assumed that each analog output uses one current loop. "DI" indicates digital input module and "DO" indicates digital output module. "VREF" is an Analog Devices 586 voltage reference and is only used to correct for voltage offsets in the input signal path when the system is initialized. Other components in Figure 6-1 are all standard in digital systems. The A/D is a Burr Brown ADS 7805 that operates at 100,000 16-bit conversions per second. The D/A is a Burr Brown DAC 712 16-bit converter. Arrows represent signal flows between different components. Note that two analog backplanes are simply represented by one.

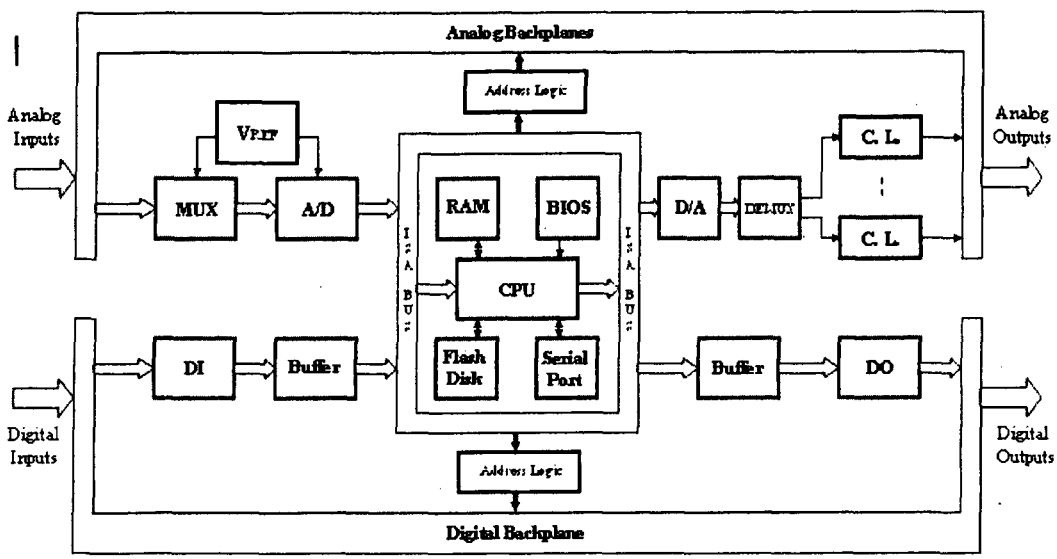


Figure 6-1 Components of the main CPU module

The Markov model of the Main CPU module was constructed using the FMEA of the components of the module, and its failure modes were quantified using the failure rates of the components and the distributions of the failure modes. Appendix B.3 describes the FMEA conducted on the Main CPU module including all its internal components, and provides detailed information of the failure modes and effects that were used in building the Markov model. This information includes estimated failure rates of the detailed failure modes, whether or not the failure modes can be detected by the software and watchdog timer, and the effects on the operation of the Main CPU. The detailed failure modes of the Main CPU were grouped according to their effects on its operation, and the failure rates of the detailed failure modes were summed to obtain the failure rate of each group. The failure rates of the groups were subsequently used as the transition rates in the Markov model.

Table 6-1 summarizes the failure modes of the Markov model. The Main CPU has five different failure modes defining five failed states, and its Markov model consists of six states, including a success state. The states were defined by considering their interactions with the Markov models of other modules of the system, e.g., watchdog timer and controllers. A CPU cannot automatically recover from its failure modes. Therefore, no recovery transitions were needed. The failure rates of the failure modes were estimated using the failure rates of the component failure modes listed in Appendix B.3, and have been listed in Table 5-1. The considerations for modeling the failure modes are given below:

Watchdog timer (WDT) detectable failures. The watchdog timer should be able to detect this failure mode and initiate fail-over; that is, the controllers will instead use the demand signals from the Backup CPU, provided that the backup CPU is in good condition. If the WDT fails to detect this failure mode, then it becomes an undetected failure.

Continued operation with latent failures. This failure mode represents failures of the internal components of the Main CPU that do not affect the DFWCS's operation. It is considered a failure mode because, combined with other failure/failures, it could affect the DFWCS's operation. For example, a localized failure of the S/G 12 MFV Tracking signal at the analog backplane does not affect the DFWCS's operation, but its combination with the failure of the signal of the other S/G would cause a failure in calculating the feedwater pump (FWP) demand. Approximately 64 of the failure modes of the internal components were grouped into this failure mode. A more refined grouping may be required to properly account for the combination of failures.

Software detectable failures. Given this failure mode, a software-initiated fail-over will take place if the Backup CPU is in good condition.

Main CPU tracking. In this mode, the Main CPU is switched to Tracking mode, but the rest of the DFWCS does not recognize this change; therefore, the system is failed. Potentially, this failure mode can be merged with undetectable failures.

Failures that cannot be detected by the DFWCS (Undetectable failures) This failure mode leads to loss of automatic control by the DFWCS.

6.1.2 Development of the Markov Model of Other Modules

Following the example in Section 6.1.1, models of the Backup CPU, controllers, valve positioners, and FWP controllers can be developed. In addition, standard Markov models of sensors, watchdog timers, and 120V AC buses can be developed.

Table 6-1 lists the possible failure modes of other modules of the system. They were derived by expert judgment rather than the thorough consideration that was done for the Main CPU (Section 6.1.1). Hence, these failure modes are not intended to be exhaustive, but are used here to demonstrate how a system-level Markov model could be generated from the module-level Markov models.

Table 6-1 Postulated failure modes of some other modules.

Modules	Total number of failure modes	Failure modes
Main CPU	5	WDT detectable failures
		Software detectable failures
		Continued operation with latent failures
		Main CPU Tracking
		Undetectable failures
Backup CPU	4	WDT detectable failures
		Software detectable failures
		Continued operation with latent failures
		Undetectable failures
Watchdog timer - Main (2)	2	Loss of ability to detect detectable failures of the Main CPU
		Spurious trips
Watchdog timer - Backup	2	Loss of ability to detect detectable failures of the Backup CPU
		Spurious trips
MFV controller	3	High or arbitrary signal (not detectable by the PDI controller)
		Low signal and loss of power (detectable by the PDI controller)
		Continued operation with latent failures
FWP controller	3	High or arbitrary signal (detectable by the Lovejoy controller)
		Low signal and loss of power (detectable by the Lovejoy controller)
		Continued operation with latent failures
		Low signal and loss of power (detectable by the PDI controller)
		Continued operation with latent failures

Table 6-1 Postulated failure modes of some other modules (cont'd).

Modules	Total number of failure modes	Failure modes
PDI controller	3	Inadvertent takeover of MFRV control with excessive MFRV demand
		Other undetected failure
		Continued operation with latent failures
S/G Level sensor #1	2	Signal failed Hi Out Of Range (OOR)
		Excess drift or step change
S/G Level sensor #2	2	Signal failed Hi OOR
		Excess drift or step change
MFRV positioner A (assumed initially operating)	1	Loss of function
MFRV positioner B (assumed initially on standby)	1	Loss of function

6.2 Development of a System-Level Markov Model

System-level Markov states can be defined in terms of the module-level Markov states. In general, a very large number of system states can be defined, and transitions among them have to be determined; that is, the total number of system-level states could be as large as the product of the numbers of possible module states. Using Table 6-1 as an example, the possible number of system-level states is 6*5*3*3*4*4*4*4*3*3*2*2=2,488,320 given the 12 modules of the DFWCS system. Since the modules are interconnected and affect each other's operation, not every combination of module states is possible, nor is every transition between the system states. A few system-specific considerations potentially can significantly reduce the number of system-level states and the size of the associated transition matrix. The DFWCS system does not have multiple redundancies. For example, many of the failure modes of the controllers are single failures of the system. Once a system-level failure state is reached, it is modeled as an absorbing state with no transition out of it.

The following describes the steps to follow in developing the system-level transition matrix. Each successive step involves postulating one additional failure mode/transition for each possible system-level state, and then determining whether or not system failure occurs. The process continues until all transition paths to the system failure state are identified. The formulation of the transition matrix essentially is a manual process requiring an extensive knowledge of how the system works, including the software. Because the software is complex, it is not possible to develop a model that follows it exactly; hence, the Markov model is an approximation of the actual failure behavior of the software. Here, each failure is represented using a character and a digit. The character "F" indicates the first failure and "S" indicates the second failure, and the digit just distinguishes between the failures.

The failures described in the following steps result in loss of automatic control by the DFWCS. Typically, for those failures that cause loss of automatic control, manual control of the system may still be possible. The degree of difficulty in assuming manual control depends on the specific failures, e.g., whether or not the failure(s) is (are) annunciated in the control room or only indicated at the plant computer, and whether or not the hardware needed for manual control is adversely affected by the failure(s). The detailed human reliability analysis (HRA) that would be necessary to accurately evaluate the likelihood of successful manual control is beyond the scope of this proof-of-concept study, and therefore only a very simplified treatment of HRA will be undertaken. For some regulatory applications, a more detailed HRA may be necessary.

Step 1: Consideration of first failures

The development of a transition matrix starts with the initial system state in which everything is working correctly, and then successively postulates failure modes of the modules one at a time. For each postulated failure mode of a module, the interactions with other modules must be accounted for to determine how the failure mode of the module affects the system. Using the example failure modes in Table 6-1, thirty-one module-level failure modes need to be considered. The following examples demonstrate the possible transitions of the transition matrix.

- F1 If the steam generator (S/G) level sensor #1 failed High Out Of Range, according to the deviation logic described in Subsection 4.1.2, the invalid signal will be detected by both the Main and Backup CPUs, a fail-over will take place, and the Backup CPU will take over control. Therefore, the module-level failure leads to a system-level transition into a system state wherein the S/G level sensor has failed, the main CPU is failed as a result, and every other module is operating. Additional failures need to be considered to result in system failure.

- F2 If the Main CPU has a failure that is detectable by its watchdog timer, a fail-over will take place since the WDT is available; therefore, further failures will need to be considered to result in system failure.

- F3 If the Main CPU has a failure that is detected by its software, then a fail-over will take place. Additional failures need to be considered to result in system failure.

- F4 If the Main CPU has a latent failure and continues to operate normally, there must be additional failures inside or outside the Main CPU to cause the system to fail. In general, the additional failures that need to be taken into account depend on the specific latent failures. This Main CPU failure mode requires further refinement.

- F5 If the Main CPU is inadvertently switched to the Tracking mode and the remainder of the DFWCS system does not recognize this change, then the system has lost automatic control.

- F6 If the Main CPU has an undetectable failure, then incorrect demands will be sent to the controllers. The controllers may detect the failure by comparing the demands with the signals from the Backup CPU. However, such deviations will only be alarmed and will not change the control. Therefore, the system loses the desired automatic control, and the failure brings about an absorbing state; no additional failures need to be considered.

- F7 If the watchdog timer of the Main CPU generates a spurious trip signal, a spurious fail-over to the Back-up CPU will take place. Additional failures need to be considered to result in system failure.

- F8 If the MFV controller fails and generates a low signal or loses its signal, the PDI controller will detect the failure, take over control from the MFV, send the last good signal to the main feedwater regulating valve (MFRV) positioner, and allow manual control of the MFRV. In this situation, automatic control is lost and the system is considered failed.

- F9 If the MFRV positioner A loses its function, the Main CPU will detect the deviation of the valve's position from its demanded position, and select positioner B instead.

As evident from these examples of first failures, some are single failures that directly lead to loss of automatic control of DFWCS, such as F5, F6, and F8. For those failure modes that do not entail loss of automatic control of the system, their indications may or may not be obvious. It is assumed that online repair is not possible. Additional failures are considered in the next step of developing the transition matrix.

Step 2: Consideration of second failures

For first failures that do not cause the system to fail, additional failures have to be considered. The process is similar to that of first failures; the failure modes of the modules are postulated one at a time, given that a first failure has occurred. This procedure is more difficult because combinations of first and second failures will have to be considered. Given the 31 module failure modes in Table 6-1, a maximum of 31*31 = 961 combinations may have to be covered. The following are example second failures for those first ones considered in Step 1 that did not cause the system to fail.

- S1 Given F1, the Backup CPU is in control with only one available S/G level sensor, i.e., S/G level sensor #2. Therefore, a failure of either the remaining sensor or the Backup CPU would lead to a system failure (i.e., the system would enter a failed state).

  In general, the failure modes of all other modules also have to be combined with this first failure (F1). Many of these combinations of first and second failures may not lead to system failure, in which case additional failures would need to be considered in successive steps.

- S2 Given F2, F3, or F7, the Backup CPU is in control, and the consideration of second failures is similar to that of S1.

- S3 Given F4, the Main CPU has a latent failure, but continues to operate. For the example of a latent failure given in Section 6.1.1, i.e., internal failure of Main CPU due to loss of S/G 12 MFV Tracking signal, a loss of the S/G 11 MFV Tracking signal due to either an

internal or external failure would lead to incorrect calculation of the FWP pump's demand and a system failure. To capture these possible combinations of failures, it is necessary to consider either a second failure internal to the Main CPU or a failure of the MFV controller.

- S4 Given F9, MFRV positioner B is put into service replacing the failed positioner A. An additional failure of positioner B would lead to system failure. As discussed in S1, failure modes of all other modules need also be considered. Some of them may lead to system failure. If not, then additional failures would need to be considered in successive steps.

The above process is continued for third failures, fourth failures, etc., until all transitions result in a system failure.

6.3 Description of How Software Failure Rates Fit in the Model

The DFWCS is a control system. Therefore, software failure rates (versus failure probabilities) should be used to quantify the software failures.

It is proposed to include software failure rates in the Markov model of each module containing software. For the DFWCS, this includes the CPUs, controllers, MFRV and bypass feedwater regulating valve positioners, and FWP speed controllers. For the Main CPU, as indicated in Table B.3-1, a software failure may or may not be detectable by the WDT. These two types of software failures contribute to the failure rates representing the WDT detectable failures and undetectable failures of Table 6-1. The same failure rates will be used for the Backup CPU, assuming complete dependence between the Main and Backup CPUs; that is, in the transition matrix, when a software failure occurs in the Main CPU, the Backup CPU also is failed with the same failure mode.

Software failure rates need to be determined for both application software and support software. These rates will be quantified after a quantitative software-reliability method is developed. The failure rates of the two types of software should be modeled wherever the software are used. For the controllers, each controller has its own application software, but they all use the same support software. Since each controller performs its own function, i.e., they are not redundant to each other, the common cause failures (CCFs) of software among the controllers do not need to be modeled. However, the CCF of the two MFRV positioners should be modeled because they use identical software and are redundant to each other.

6.4 A Simplified Method for Building and Solving the Model

The number of states in a Markov model can grow extremely large. Since the DFWCS has several modules each of which has several failure modes, the total number of system-level states correspondingly is very large. For example, Section 6.2 calculated 2,488,320 states using some defined modules of the DFWCS; when all modules are considered, the total number of states can be substantially larger. As discussed in Section 6.2, not all these states are possible, but, even so, the expected number of states is great. Thus, building the transition matrix and solving the resulting Markov model with them is a very difficult technical challenge. Accordingly, this section proposes a simplified approach for building and solving the Markov model of the DFWCS.

This approach takes advantage of two observations:

1. As described in Section 6.2, in many cases the occurrence of a few module failures causes the DFWCS to fail. In other words, in many instances it is not necessary for many modules to fail for the DWFCS to do so. This observation is specific to the DFWCS.

2. In some cases, it may be that more than a few failures are required for the DFWCS to fail. However, the probability of occurrence of a path with multiple failures is usually lower than that of a path requiring few failures. In fact, it is expected that every failure included in the path from the normal state to a failed state reduces the probability of the path by several orders-of-magnitude. This reduction results because every failure is expected to have a probability of occurrence that is several orders-of-magnitude lower than 1. An exception would be if there is some level of dependence between the multiple failures. However, such dependencies would typically be accounted for through the treatment of CCFs in the system reliability model.

These observations indicate that accounting for the paths involving a few module failures in a Markov model will yield a good approximation to the Markov model with a full transition matrix and, hence, to the correct assessment of the probability of failure of the DFWCS. Accordingly, ignoring the paths involving more than a few failures should not significantly affect the estimate of this probability because their contribution is expected to be negligible.

While only those paths involving a few module failures are expected to be required for the Markov model, there is no process for establishing in advance the maximum number of failures in a path that must be considered to produce a good approximation of the total failure probability. Hence, this maximum number is determined using an iterative process that progressively gives a closer estimate to the total failure probability.

Accordingly, the process consists of the following major steps:

1. Identify single failures causing system failure, and assess the corresponding DFWCS failure probability. The failure modes in Chapter 5 and the associated Appendix B are reviewed to identify those modes that directly cause system failure (referred to here as singles). Since each of these singles moves the system state from normal to failed, a transition matrix is built from them. The resulting Markov model effectively has two states, normal and failed.

2. Identify double failures causing system failure, include them in the transition matrix, and assess the corresponding DFWCS' failure probability. Those failure modes that do not directly cause system failure are paired with each other to determine those combinations of two failures (referred to here as doubles) that cause system failure. The paths involving doubles then are added to the transition matrix, and the resulting Markov model is solved to estimate a probability that the DFWCS is in the failed state that is more accurate than the one obtained in the previous step.

3. Iterate by including paths with one more failure at each step until the probabilities converge. This process is continued iteratively, building paths that contain one additional failure at each step, adding those paths that cause system failure to the transition matrix, evaluating the probability that the DFWCS is in the failed state, and comparing the

resulting probability with that of the previous step. The process can be stopped when the probability obtained in the current step does not increase significantly from the one obtained in the previous step. In other words, the probabilities are considered to have converged to the total probability that the DFWCS is in the failed state.

Each step includes the paths causing the DFWCS to be in the failed state with one more failure than those in the previous step. Using this process, an incrementally more accurate estimate of the likelihood that the DFWCS is failed is obtained in each step. A decision of when convergence has been achieved is subjective. For example, a difference between two probabilities that are of the same order of magnitude may be considered insignificant, given the uncertainties in the models and data.

The application of this process can be illustrated as follows. The probability that the DFWCS is in the failed state obtained using singles and doubles (from Step 2) is compared with the probability obtained using only singles (from Step 1). If the two contributions are similar, e.g., they are of the same order of magnitude, then this estimate using singles and doubles can be considered as a good approximation of the total system failure probability. If they are not, then an additional single that does not directly cause system failure is added to the path of those doubles that do not cause system failure either. In this way, paths containing three failures (triples) that cause system failure are obtained, and the likelihood of system failure using singles, doubles and triples is assessed, and compared with the probability obtained using only singles and doubles (from Step 2).

This iterative process of building paths with singles, then doubles, then triples, and so on, is essentially the same as the steps needed to develop a full transition matrix as discussed in Section 6.2, except in each step a simplified Markov model is developed and quantified to obtain an estimate of system failure probability. This process is named here the "simplified process" to distinguish it from the procedure described in Section 6.2. Since the latter systematically searches for all paths, it guarantees that all paths causing system failure are included in the model, i.e., it encompasses paths that may have a large number of failures. On the other hand, the goal of the simplified process is not to include all paths since it neglects those with a large number of failures due to their expected low probability of occurrence.

Hence, while the advantage of the procedure of Section 6.2 is that it guarantees that all paths causing the DFWCS to fail are included in the model, its disadvantages are that:

1. It is a laborious, time-consuming manual process.

2. The resulting transition matrix can be extremely large. In fact, it can be so large that it may not be practical to build it.

3. If the matrix can be built, it may be difficult to solve the associated Markov model to obtain the probability that the DFWCS is in the failed state.

The advantage of the simplified process is that the resulting Markov model is simpler to build and solve than the one that would be obtained from applying the Section 6.2 procedure. The simplified process appears feasible and its estimate of the probability that the DFWCS is in the failed state should be a good approximation. Its disadvantage is that it neglects some paths (i.e., those with multiple failures) that cause system failure that may be relevant contributors to the likelihood of this failure.

Based on the above considerations, if the Section 6.2 procedure for building the transition matrix becomes too difficult or impractical to implement, or if the resulting Markov model is too large to solve, the simplified process will be attempted.

7. DEVELOPMENT OF A FAULT TREE MODEL OF THE DIGITAL FEEDWATER CONTROL SYSTEM

This chapter delineates the application of the traditional event tree/fault tree method to construct and solve a probabilistic model of the digital feedwater control system (DFWCS). Chapter 3 concluded that a model will be developed for the initiating event (IE) "Loss of control of the loop associated with a DFWCS."

A model of an IE is typically developed using a fault tree, and that is the approach used here; accordingly, the capabilities of "event tree analysis" are not used. Having established the model of the IE, it then can be linked with its associated accident scenarios in an event tree.

As derived in Chapter 3, the frequency of the IE can be obtained using the following equation:

$$f = - \ln[1 - P_f(T)] / T \qquad (7\text{-}1)$$

wherein

ln means natural logarithm,
T is the period of interest, i.e., one year, and
$P_f(T)$ is the probability that the DFWCS fails within the period T.

This chapter describes the construction and solution of a fault tree for estimating the probability $P_f(T)$, thus enabling the calculation of the frequency of the IE using equation (7-1). Fault tree analysis is a well-established and commonly used method for qualitatively and quantitatively assessing system unreliability. Hence, it is not discussed here (the method is described in detail in references such as the Fault Tree Handbook [Vesely 1981]. Sections 7.1 and 7.2, below, describe the construction and evaluation of the fault tree, respectively.

7.1 Fault Tree Construction

The first step in building the fault tree is to precisely define the undesired event, called the top event, associated with a DFWCS. As discussed in Chapter 3, for this study, failure of a DFWCS is defined as loss of automatic and manual control of its related loop. Hence, the top event is defined as "Loss of control of the loop associated with a DFWCS." The fault tree is developed to model this loss over a period of one year. In this way, the probability of loss during one year obtained from the fault tree can be used in equation (7-1) to yield the frequency per year of the IE.

A fault tree is built via a deductive approach. Once the top event is defined, the possible failures that can cause this event are systematically deduced, then, their causes are determined, and so on. As described in Chapter 2, it is important to include in the fault tree all relevant failures of the components of a system contributing to system failure, such as the failure modes of these components, dependent failures (including common-cause failures (CCFs)), and related human errors.

When the plant is in power operation, a DFWCS automatically controls the feedwater in its associated secondary loop, unless the plant operators set the DFWCS to the manual mode.

Failures in one DFWCS can cause it to lose automatic control of its loop. Thereafter, the operators may be able to take manual control. Hence, the top-level of the fault tree can be depicted as shown in Figure 7-1.

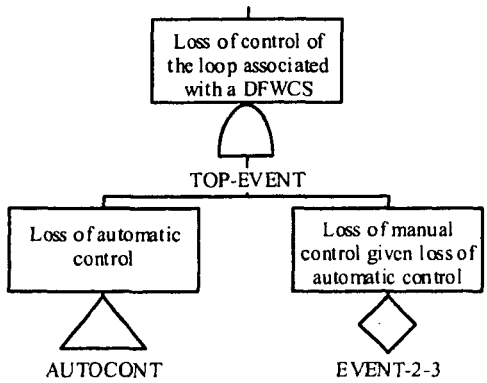


Figure 7-1 Loss of control of the loop associated with a DFWCS

The degree of difficulty in assuming manual control depends on several factors related to the specific failures that caused the loss of automatic control, such as (1) the availability of indication, i.e., annunciation in the control room or only at the plant's computer, (2) the availability of hardware needed for manual control, which will depend on whether that hardware was adversely affected by the failures, and (3) the time available for the operators to respond to the loss of automatic control. Hence, the loss of automatic control may or may not be recoverable by the operators. Since manual recovery is contingent upon the specific failure(s) that caused the loss of automatic control, a specific recovery action (basic event) should be modeled together with each specific failure(s). However, treating human errors is not the objective of this task, so for the purpose of illustration they are simply modeled with a single basic event at the top of the fault tree, as shown in Figure 7-1, and not developed further at this time.

A DFWCS controls a steam-turbine-driven feedwater pump (FWP), a main feedwater-regulating valve (MFRV), and a bypass feedwater-regulating valve (BFRV). In general, if the DFWCS fails to control any one of them, automatic control is lost. Further, some failure modes of the modules of the DFWCS cause a direct failure to control all of the modules. Direct failure means that a single failure causes the system to fail. Hence, loss of automatic control can be modeled as shown in Figure 7-2.

Each failure in Figure 7-2 must be developed. To illustrate this process, the main considerations in modeling "Failure to control MFRV" and "Direct failure to control 3 components" are delineated in the next two subsections. The fault trees of this chapter use the typical symbols, that is, the events represented with diamonds are 'undeveloped" events that are not created at this time; they are candidates for development in the follow-up task of this project. The triangles represent transfers to branches of the tree that are developed in this chapter.

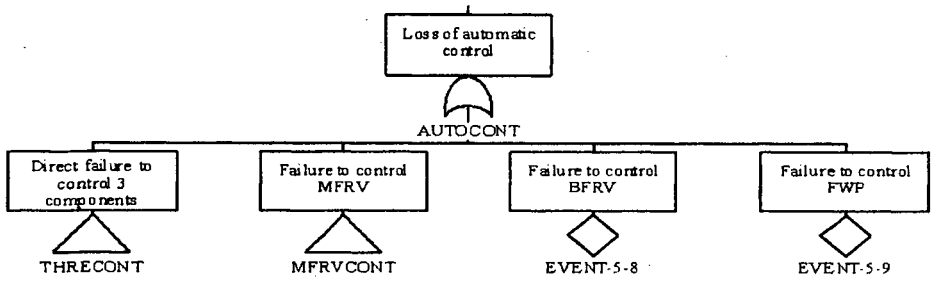


Figure 7-2 Loss of automatic control

7.1.1 Modeling of Failure to Control MFRV

Figure 7-3 is a simplified diagram of the control of an MFRV. During normal operation, the MFRV is commanded by its controller, main feedwater valve (MFV), i.e., it receives a demand from the MFV specifying its opening position. The pressure differential indication (PDI) monitors the output from the MFV to the MFRV; if the PDI detects a failure of the command from the MFV to the MFRV, the DFWCS is set to the manual mode, and the operators can take control using the PDI controller. The MFV controller normally receives the MFRV demand from the main central processing unit (CPU), while the backup CPU is in tracking mode. Both processors receive input from several sensors, calculate demand values for each component (in this case the MFRV) using this input, send these demands to the controllers (here, the MFV), and exchange data between each other. For simplicity, the figure shows only the water level sensors of one steam generator. The sensors measure process variables and transmit them to the processors.

Knowing this "chain of command" between the modules of the DFWCS, the fault tree can be developed following a deductive process and the "Immediate Cause" concept [Vesely 1981]. In other words, the immediate causes of the loss of control of the MFRV are determined, which are failures associated with the MFV and PDI. For example, loss of control of the MFRV is due to an incorrect demand from the MFV and control cannot be re-established using the PDI. These failures are included in the fault tree under the gate "Failure to control MFRV." Then, the immediate causes of the failures related to the MFV and PDI are identified, which are failures of these controllers themselves, or failures coming from the processors. These failures then are included in the fault tree under the appropriate gates. This process is continued until the "origin" of the signal is reached, i.e., the sensors. The failure modes of each module of the DFWCS and the impact of each mode on the system, defined and discussed in Chapter 5 and Appendix B, are used to establish the immediate cause of each failure in the fault tree. Thus, the basic events of the fault tree correspond to the failure modes of the DFWCS modules.

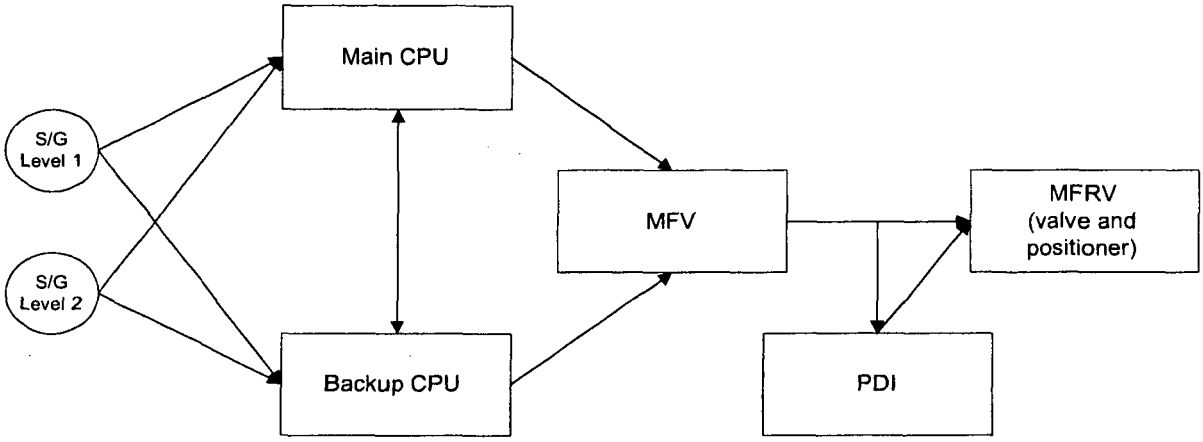


Figure 7-3 Simplified diagram of the control of an MFRV

Hardware and software failure modes of the DFWCS modules are incorporated into the fault tree in this way. In addition, if CCF of the hardware or software of some modules might be possible, that type of failure also is included. Figure 7-4 presents the branch of the tree "Failure to Control MFRV," developed using this deductive process.

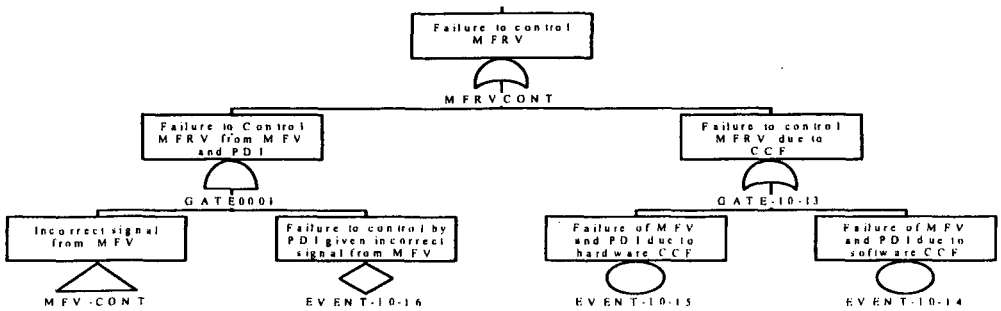


Figure 7-4 Failure to control MFRV

To continue illustrating the process of building the fault tree, next the "Incorrect signal from MFV" is refined. According to the "Immediate Cause" concept, this failure is due to failures of the MFV or incorrect signal from the CPUs. In addition, modeling the failures of support systems can be introduced at this stage of development. The MFV has its electrical supply from two 24V DC power supplies that are diode-auctioneered. Hence, both supplies must fail for the MFV to lose its power supply. Accordingly, failures of the MFV can be due to its internal failure modes, or failure of its electrical supply. Figure 7-5 shows this modeling.

The failure of the 24V DC power supplies can be modeled as typically is done in probabilistic risk assessments (PRAs), i.e., in terms of the electrical buses providing power to these supplies. Similarly, the internal failure mode of the MFV can be developed in terms of its hardware and software failure modes. Accordingly, these failures in the fault tree are not further developed at this time.

Continuing the process of building the fault tree, the "Incorrect signal from the CPUs" is then developed. The main CPU normally controls the MFRV by sending a demand to the MFV, which, in turn, forwards it to the MFRV; the backup CPU is in the "tracking" mode. If a failure of the main CPU is discovered by the detection mechanisms of the DFWCS, such as the watchdog timer of this CPU, a "fail-over" to the backup CPU occurs. In other words, the backup CPU takes over as the controlling CPU. Figure 7-6 depicts this modeling, wherein the CCFs of the hardware and of the software of the CPUs are postulated as possible failure modes that would fail both of the CPUs. The independent failure of each CPU would then be developed in terms of its failure modes. Subsequently, failures of the sensors can be included in the tree.

This process of refinement using the "Immediate Cause" concept is continued until reaching the level of detail considered appropriate for capturing the relevant contributors to the failure of the system, that is, the level of the failure modes defined in Chapter 5. In general, these failure modes become basic events in the tree.

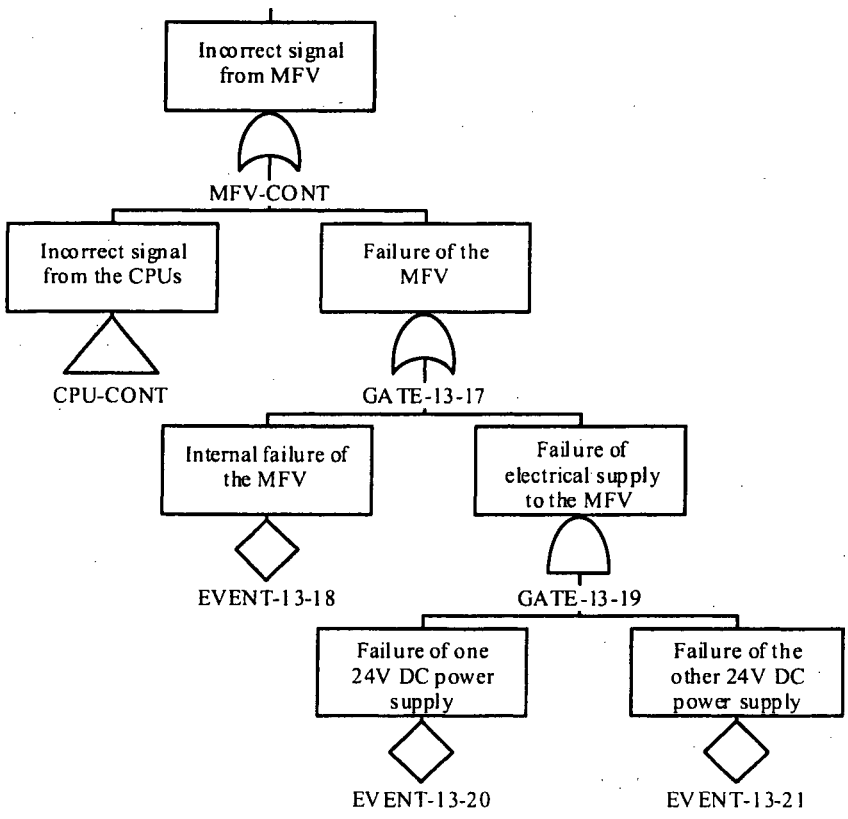


Figure 7-5 Incorrect signal from MFV

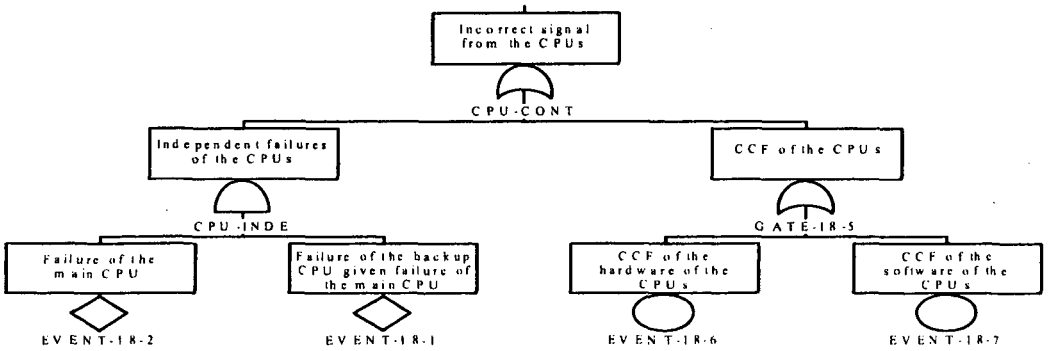


Figure 7-6 Incorrect signal from the CPUs

7.1.2 Modeling of Failure to Control Three Components

So far in developing the fault tree, failures of the DFWCS modules are combined according to the logic gates of the fault tree, thus leading to the failure of the system. In addition, two types of failure are considered to lead directly to DFWCS failure: some CCFs and some failure modes of individual modules. Two examples of the first type are the CCF of the hardware and of the software of the controllers of the MFRV, BFRV, and FWP. An example of the second type is the failure mode of the main CPU in which the failure is not discovered by the DFWCS's detection mechanisms, such as the watchdog timer of this CPU. As stated in Chapters 5 and 6, this failure mode causes the DFWCS to fail directly.

Figure 7-7 presents these two types of failure for the purpose of illustration; other failures of these types, such as the CCF of two of the controllers, may be applicable but are not included to void unnecessary complexity at this time.

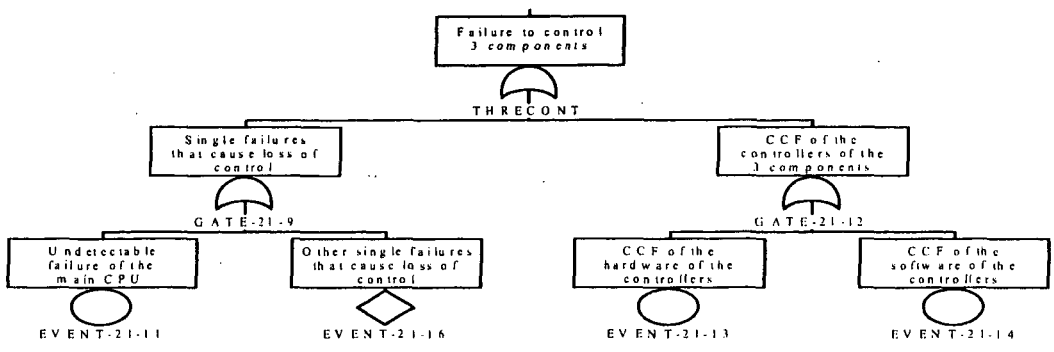


Figure 7-7 Failure to control 3 components

7.2 Fault Tree Evaluation

A fault tree can be solved qualitatively and quantitatively. The qualitative solution provides the combinations of failures leading to the top event. In other words, the failure modes of each DFWCS module included in the fault tree are combined according to the fault tree logic to obtain the unique ways in which the system can fail, i.e., the minimal cut sets (MCSs). The quantitative solution yields the probability of the top event, i.e., the probability of failure of the DFWCS within one year. Important considerations for obtaining these solutions for the DFWCS fault tree are discussed next.

When several large or complex fault trees are combined, as in an overall PRA, the number of MCSs can be extremely large. Therefore, a "cut-off" probability is commonly used to obtain those MCSs with a probability larger than, or equal to, the cut-off. Except for this limitation, all the unique failure modes of the DFWCS can be obtained by solving its associated fault tree. Incidentally, since there are an extremely large number of unique ways in which a system can fail, this limitation appears to be common to all methods available for identifying these failure combinations.

As discussed at the beginning of this chapter, the top event is defined as "Loss of control of the loop associated with a DFWCS." To obtain the probability of this event, the probability of occurrence of each basic event of the fault tree must be assessed. An approach to estimating the latter probability is described next.

Since the DFWCS is running while the plant is operating at power, the modules of this system also are usually running during this time. Hence, a basic event of the fault tree usually represents the failure of a DFWCS module to run over one year. The exponential distribution can be employed to calculate the probability of this failure because it is the cumulative probability that the failure occurred by one year:

$$P(T) = 1 - \exp^{-\lambda T} \qquad (7\text{-}2)$$

where P(T) is the probability of the occurrence of a failure mode of a module within the period T, i.e., one year, and λ is the failure rate of this failure mode. In other words, P(T) is the probability that this failure mode occurs before T. The failure rates of the modules of the DFWCS are discussed in Chapter 8.

Using the MCSs and the probability of each basic event, the quantitative solution of the fault tree provides the probability of failure of the DFWCS within one year. This probability, in turn, can be used in equation (7-1) to estimate the frequency of the initiating event "Loss of control of the loop associated with a DFWCS." Computer codes that are commonly employed in current PRAs can be used for building, and qualitatively and quantitatively solving the fault tree.

8. DEVELOPMENT OF RELIABILITY PARAMETERS FOR DIGITAL SYSTEM RELIABILITY MODEL

Quantification of digital system reliability models requires failure parameters, e.g., failure rates for a Markov model and failure probabilities and/or rates for event tree/fault tree methods. In this chapter, the data sources and the methods for estimating these parameters are discussed.

This study reviewed available databases and performed a Bayesian analysis that attempts to account for variability of different raw data sources. In the review, potential weaknesses and limitations of the available databases are identified and discussed, and no attempt was made to validate or invalidate the available databases. The limitations in the publicly available failure parameters of digital components identified in this study indicate that additional research and development is needed in this area. The data documented in this chapter are not appropriate for quantifying models that are to be used in support of decision-making (e.g., regulatory decisions or design changes). The data will only be used in this project to demonstrate the reliability methods and exercise the reliability models.

To define the reliability parameters for digital system modeling, currently available analyses of digital systems and components were reviewed. The available data on digital system and component failures were also studied to evaluate the methods for the estimation of parameters. The activities conducted for development of reliability parameters can be defined as follows:

1. Review of currently available analysis methods for estimating reliability parameters, including estimated parameters and failure data,

2. Review of publicly available sources of raw data and applicability of the raw data for digital components in nuclear power plants (NPPs),

3. Defining an approach for estimating failure parameters using available data for digital components.

Development of reliability parameters for digital systems and a database that can potentially be used for the estimation of the parameters essentially consists of the following aspects:

1. Identifying available analyses that contain estimated parameters for digital systems/components used in NPPs,

2. Identifying raw data in the available analyses and in different data sources for NPPs,

3. Defining a method for estimating parameters that addresses issues in digital system data collection,

4. Defining a method for evaluating dependent failures, and

5. Identifying factors that contribute to uncertainties in the estimation of parameters and defining methods for treatment of the uncertainties.

Based on the reviews conducted of the available databases and data sources, the use of the available data and methods are defined. Section 8.1 discusses different data sources and Section 8.2 presents the issues in digital system and component data collection and analyses.

Sections 8.3, 8.4, and 8.5 present reviews of three types of databases. A summary of the failure data review for digital systems is presented in Section 8.6. A method for estimating reliability parameters for digital components is presented in Section 8.7. As mentioned previously, the results of the analysis along with the estimates of the PRISM database [RAC PRISM] will be used in later tasks of this project only to demonstrate the methods for modeling digital systems and the usefulness of the models developed. Dependent failures are discussed as part of the review of the available analyses of digital systems (Section 8.6.2). Treatment of uncertainties is discussed in various parts of this chapter, as well as in Chapter 2.

8.1 Categories of Potential Data Sources for Digital Systems and Components

To identify the available data on digital systems and components, a systematic search of different potential data sources was conducted. The data sources included commercially available databases for different applications, analyses conducted for digital systems in nuclear and other industries, and failure databases commonly used in the nuclear industry. For discussion in this report, these different data sources are organized into three groups, as presented below. Detailed discussion of these databases is presented in Sections 8.3, 8.4, and 8.5.

- Commercially available databases that obtain reliability parameters of digital systems and components using Reliability Prediction Methods (RPMs) or other methods

These databases use RPMs to provide failure parameters of digital systems and components directly. The failure parameters of specific digital systems or components can be calculated or acquired directly if related information such as component types, operating environment, etc., is specified. The following databases are included in this group: Military Handbook 217F [DOD 1995], Telcordia [2001], PRISM User's Manual [RAC PRISM], Institute of Electrical and Electronics Engineers, Inc. (IEEE) Standard 500 [1986], International Eletrotechnical Commission (IEC) Standard 61508 [1998], and a PDS method [Hauge 2006b]. The IEEE standard was found not to contain any digital component data, and is included in this report only because it was part of the search for data for digital components. A new RPM 217+ [RIAC 2006] supersedes the Military Handbook 217F [DOD 1995] but was not used in this study.

- Analyses used in different industries to obtain reliability parameters of specific digital components for specific applications

These databases are based on operating experience of digital systems in various industries. Digital systems have been used in many industries for years. Analyses conducted by users of digital systems and components to obtain failure parameters are considered in this group of databases. In many cases, raw data are also available for specific components. The details regarding the data collection and analyses are different among various databases discussed here. Example databases included in this group are: Digital Core Protection Calculators in Combustion Engineering Reactor Protection Systems (RPS) [Wierman 2002], Eagle-21 Channels in Westinghouse RPS [Eide 1999], and Programmable Logic Controllers Used in Emergency Shutdown Systems of Natural Gas Compression Stations [Mitchell 1993]. In these databases, relatively detailed analyses of data are performed.

- Databases for identifying digital system and components failures in NPP operation

These databases contain failure data for systems and components in NPPs and are not specific to digital systems and components. These databases can be used to obtain raw data on digital system and component failures. Two databases, Licensee Event Reports (LERs) and the Equipment Performance and Information Exchange (EPIX) Database, are included in this group. In this report, these two databases are studied to analyze their capability to provide digital system and equipment failure data.

8.2 Issues in Digital System Data Analyses

Level of data

A digital system reliability model can be developed at different levels as long as the reliability model captures the design features that affect the system reliability. To quantify the reliability model, failure parameters at the corresponding level are required. Effectively, failure data for calculating the failure parameters have to be collected at the same or a lower level (more refined level). Note that it is possible to calculate high level failure parameters using lower level component data, if data are available for all of the lower level components that contribute to failure of the higher level component, and the complete set of lower level component failure combinations that lead to failure of the higher level component are known.

Digital systems can be decomposed into different levels. A digital system may have several independent channels; failure data may be collected for the channels. The functions performed by a digital system or a single channel can be divided into input, output, and processing. Failure data can be collected for input, output, and processing. An individual circuit board or card is present in all digital systems; failure data can be collected for each of the circuit boards and cards. Care is needed in using the failure data because the available data are collected at different levels. The categorization of collection levels for digital data will be further discussed in Section 8.6.1.

Failure modes of the digital systems and components

Failure modes of digital systems and components are not yet clearly defined and can be an impediment to using the available data. Because of the lack of clearly defined failure modes, failure data may be defined ambiguously requiring detailed analyses for data categorization. Frequently, in the digital system failure data, failure modes are implied instead of being defined explicitly. The implied failure mode in many data items is that the digital system fails to perform its designated function.

Component-specific data

For quantifying digital system models in NPP PRAs, component-specific reliability parameters are desirable. Component-specific data for digital systems and components in NPPs may not be available. Very often, data at a level lower than the component may have to be used to estimate component-specific parameters. Also, because of the lack of sufficient operating experience with these equipment, data from similar components operating in different environments may have to be used. Methods used for the estimation of the reliability parameters should have the capability to incorporate different types of data in estimating component-specific reliability parameters.

Sparsity of data

With a few exceptions, digital systems are only relatively recently being used in NPPs. The operating experience of these equipment is relatively short and these systems are generally reliable resulting in sparse failure data. The sparsity of failure data for digital systems and components in NPP applications necessitates using data from different industries. Combining data from different applications becomes a necessity for digital systems.

Impact of operating environment

Digital equipment reliability is impacted by factors relating to its operating environment and its specific features designed to cope with the operating conditions. Failures of these equipment may be caused in part by electro-magnetic interference, temperature, humidity, and the time period in a specific environment. Data collected for digital equipment may need to be sorted considering the differences in operating conditions. In many cases, the operating conditions associated with these failures may not be clearly known, resulting in an inability to accurately sort them and, in turn, an increased uncertainty in the estimated parameters.

Software failures

A unique feature of digital systems is the use of software. It is known that software can fail resulting in failure of the digital component it supports. Software failure may be separately modeled in the system reliability model and quantified based on available data. For example, Teleperm XS modules have reported 5260 module years of operating experience [Niedzballa 2004], which is useful in estimating the failure rate of the platform software. The estimation of the software reliability parameters should be consistent with the model being used. It is possible that in some databases, some software-induced hardware failures may not be attributed to failure of the software. Without knowing how software failure is treated in the data, digital system reliability modeling may be difficult.

Need to combine data from different sources

Discussion of different issues presented above point to the need for combining data from different industries. Experience with digital equipment in military, chemical, aviation, telecommunications, non-nuclear power plants, and other industries is sometimes more than that in commercial NPPs. There is a need to learn from the failure experiences in other industries and, also, to use available failure data in other industries in the estimation of failure parameters for digital components in the nuclear industry. Of course, quality and relevancy of the data for the digital component and its operating environment will need to be addressed.

8.3 Reliability Parameters Based on Databases Using Reliability Prediction Methods (RPMs) and Other Methods

This section discusses generic databases using RPMs or other methods for digital systems and components. Military Handbook 217F [DOD 1995], Telcordia [2001], and PRISM consider impacts of different stresses such as temperature and humidity to adjust the failure rates using empirical methods. IEEE Standard 500 adopts a so-called "Delphi Method" to estimate failure rates of systems and components. IEC 61508 and PDS Data Handbook [Hauge 2006a] focus more on reliability method development instead of failure parameter development.

Among these databases, IEEE Standard 500 provides uncertainties of failure rates, i.e., lower and upper confidence bounds in addition to failure rates. IEC Standard 61508 provides guidance on the assignment of safety integrity levels (SILs) without providing an associated database. The rest of the RPMs give failure rate estimates only.

Failure parameters of basic digital components from sources of most prediction methods are generic. These parameters usually do not reflect design and integration features of digital systems. Therefore, these basic components may be considered the lowest level components at which the RPMs can provide failure parameters.

8.3.1 Military Handbook 217F [DOD 1995]

This handbook is the most frequently quoted database. It is a source of component level digital hardware data, e.g., microprocessor, digital gate/logic array, and memory. The failure parameter is in the form of failure rates, and no raw failure data are provided. Updating of the database has been discontinued by the Department of Defense (DOD) since 1991.

The Military Handbook 217F [DOD 1995] contains two methods for estimating failure rates of boards/systems, the part count method and the part stress method. The part count method adds the failure rates of the components of the system to obtain a system failure rate. The part stress method further makes adjustments to the base component failure rates by considering part quality and use environment.

One difficulty in reviewing the military handbook is that its underlying models are not available. Supposedly, the failure rates and different multiplicative factors (i.e., the C and π factors) listed in numerous tables of the handbook were estimated using these underlying models. The military handbook method has been criticized [Pecht 1994 and Sinnadurai 1998] for being inaccurate by orders of magnitude. The basic problem with the method is that empirical formulae are approximations to the laws of physics, which are much more complicated and difficult to model, and have to be validated with data.

Due to a lack of knowledge regarding the raw data and the underlying models used in estimating the failure rates in the military handbook, it is difficult to determine how the failure rates should be used in modeling a digital system.

It appears that software failures are not included in the database, and some fault-tolerant features are implicitly included in the failure rate estimates. For example, it appears that a fault that is automatically detected and corrected will not be included in the database. Therefore, crediting such fault-tolerant features in a reliability model would be double-crediting the feature. These issues need to be clarified prior to using this database.

8.3.2 Telcordia

Telcordia [2001] provides guidance on estimating failure rates of electronic components, units consisting of components (e.g., a customer-replaceable assembly of components), and systems consisting of units in series. It is intended to be used by the telecommunication industry in planning maintenance actions, life cycle cost analysis, making decisions on competing products, design trade-off studies, and comparison with performance standards. It is not required by any government regulation or industry standard. However, according to Telcordia, it is the de facto

standard for hardware reliability predictions in the telecommunications industry. No publicly available papers or reports on applications of the method are available.

The method used by Telcordia is similar to that of Military Handbook 217F [DOD 1995]. Therefore, many of the comments regarding the military handbook that were discussed above also apply to Telcordia. At a component level, generic failure rates are modified by π factors representing the effects of quality, electrical stress, and operating temperature. Tables of the π factors are provided without indicating how they were derived. If test data or field data are available, then Bayesian analysis is performed. Only the results of the Bayesian derivations are given in equations, without the associated assumptions. Burn-in factors are used to account for the higher failure rates during burn-in periods. At the unit level, unit failure rates are simply the sums of the failure rates of the components in the units, modified by the environmental factors representing different operating environments such as a ground-based, fixed, and controlled environment. The treatment of test and field data of a unit is the same as that for a component, and the unit level first-year factor is the weighted average of the factors of the components. At the system level consisting of series units, the failure rate is simply the sum of the failure rates of the units in the system, and the system level first-year multiplier is simply the weighted average of the unit level first-year multipliers. For redundant systems, SR-332 only provides a few references that provide guidance on system modeling, e.g., Markov modeling. These references only provide general guidance on Markov modeling and no specific guidance on modeling digital systems.

Telcordia provides lists of generic component failure rates of electronic components. The failure rate estimates represent the 90th percentile of the distribution of the component failure rates. The failure rate estimates are specified for different complexities (e.g., number of gates and power ratings), specific temperatures, and electrical stresses. The only available information regarding the sources of the failure rates is that the failure rates were derived from data provided by several suppliers. The method used in deriving the failure rates is not provided. The use of 90th percentile is intended to ensure that the results are conservative.

8.3.3 PRISM Database

PRISM is a software tool developed by the Reliability Analysis Center (RAC) for assessing system reliability. It includes a failure rate database for both electronic [RAC 1997a] and non-electronic components [RAC 1995]. A separate database [RAC 1997b] contains the failure mode and mechanism distributions, which allows partitioning of failure rates in Electronic Parts Reliability Data [RAC 1997a] into failure modes and mechanisms.

The PRISM database can be considered an update of Military Handbook 217F [DOD 1995] with more recent data up to the year 2000 and improvements in the reliability prediction method [Dylis 2001]. Similar to the Military Handbook, PRISM allows a user to make predictions about the failure rates of series systems. In addition to providing guidance on the use of different factors that modify the base failure rate of a component according to different stresses (e.g., operational, environmental, power and thermal cycling, electrical, and solder joint), PRISM also provides guidance on the use of process grading factors to account for design and manufacturing variability at the system level. Seventy-eight percent (78%) of the system failures in the database, of which software failures contributed 9%, are not caused by component failures, but by system-level failures, such as problems during the design and manufacturing processes. Hence, this 78% of system failures may not have been explicitly addressed in earlier prediction methods.

PRISM provides a method for determining software failure rates at the system level using the capability maturity model (CMM) of the Software Engineering Institute of Carnegie Mellon University. Basically, the CMM level or other measures (e.g., RTCA safety level [RTCA 1992] and ISO 9000 [ISO 2000] certification) is converted into the number of faults per thousand lines of code, which in turn is converted into mean time to failure using a reliability growth model. The PRISM method is described in more detail in the PRISM User's Manual [RAC PRISM].

PRISM contains two methods for estimating component failure rates, the new component level model described above, designated as the RACRates model, and the more traditional method based on failure records collected for different components, designated as RACdata [RAC 1995, 1997a]. PRISM also allows user-specified component failure rates. The sources of data in the database include published reports and papers, data collected from government-sponsored studies, military maintenance data collection systems (e.g., Air Force REMIS system), commercial warranty repair systems, commercial/industrial maintenance databases, and data submitted to RAC from military or commercial organizations that maintain failure databases.

A few applications of the new PRISM method (RACRates model) were reported with the results compared with failure experience [Priore 2002, Brown 2003, Smith 2004]. Priore analyzed a small electronic subsystem having both hardware and software components of TRW Automotive, and found that the predictions based on the PRISM method are approximately a factor of two higher than the failure rates calculated using warranty data. Smith and Womack [2004] analyzed three electronic units used in military aircrafts and found that the PRISM method underestimated the failure rates. Brown analyzed three different circuit card assemblies (CCAs) used in a military airborne environment using the part stress method of Military Handbook 217 as well as the PRISM method, and found that the PRISM method underestimated the failure rate of two CCAs and overestimated the failure rate of the third CCA, while the part stress method underestimated all three CCAs. Overall, the case studies seem to indicate that a factor-of-two error in the estimation is possible. A way to address this is taken into consideration in estimating uncertainties in the method.

The RACRates model of PRISM contains only three models for hermetic/ceramic integrated circuits and three models for non-hermetic/plastic integrated circuits (i.e., digital, linear, and memory/processor). The components of the same model have exactly the same failure rate estimates. The components of the same model include a wide collection of components, and PRISM is assuming they have the same failure rate. The environment parameters (e.g., temperature, humidity, vibration) and operating profile parameters (e.g., automotive vs. military airborne, duty cycle, cycle rate, and year of manufacture) could change the predictions, but the same change applies to all components. This represents a lack of discriminative ability of the PRISM RACRate model.

The RACdata database contains failure data records in the form of the number of failures in a number of operating/calendar hours for components (part types) listed in Appendix N of the PRISM User's Manual. Failure data records for specific components are grouped to represent a generic component. The variability among the estimated failure rates of the specific components is an indication of the variability of the estimated failure rate of the generic component. It will be discussed in Section 8.7 that a hierarchical Bayesian analysis [Atwood 2003] can be used with RACdata to account for the uncertainty associated with variability of data sources.

In Section 8.7, the raw failure data extracted from the PRISM software will be used in a hierarchical Bayesian analysis to estimate the generic failure rates of digital components, accounting for variability of the failure data from different sources.

8.3.4 IEEE Standard 500-1984

IEEE Standard 500-1984 [1986] contains reliability parameters for electrical, electronic, sensing components, and mechanical equipment for the purpose of being utilized by NPP reliability calculations and predictions. In addition to upper and lower confidence bounds (10th and 90th percentiles) of failures in terms of 10^6 hours or 10^6 cycles, recommended failure rates are provided to determine the best estimates for equipment. Environmental factors caused by temperature, humidity, radiation, etc., are provided to factor in the effects caused by environmental stresses. Repair time in terms of hours and various failure modes are defined for items or equipment. The data is collected from nuclear facilities, fossil-fired generating stations, and other industries (i.e., chemical industry, transmission grids, individuals familiar with the operating and failure histories of specific generic types of devices, and other published sources).

Nuclear power industry related equipment are categorized in 17 different types. The only category that may contain digital systems and components is "Instruments, Controls, and Sensors." In this standard, there are no indications whether the items from this category are analog or digital, even for regulators or controllers. Considering the year this standard was edited, it is very likely that no digital systems were included within the time period that the data were collected. Therefore, the failure parameters provided by this standard are not further considered and it is reviewed for the purpose of completeness only.

8.3.5 IEC Standard 61508

IEC Standard 61508 [IEC 61508] specifies requirements of safety-related systems and provides guidance on assignment of SILs. This standard consists of seven parts. Part 1 of the standard specifies target failure probabilities for failure on demand and target failure rates for different SILs. It describes the steps for determining the SIL of a system based on the SILs of the functions that the system performs. The SIL of a function is determined by using a concept of "necessary risk reduction" either quantitatively or qualitatively. Section 7.4.3.1 in Part 2 of the standard provides a method for determining the hardware SIL of a system with redundant channels based on system architecture, fault tolerance, and safe failure fraction. Section 7.4.3.2 provides guidance on estimating probability of failure of safety functions due to random hardware failures. Quantitative methods for evaluating the probability of failure of safety functions include cause consequence analysis, fault tree analysis, Markov analysis, and reliability block diagram. The standard states that component failure data from a recognized industry source should be used. The guidance specifies that common cause failures (CCFs), diagnostics, test interval, safe and dangerous failure, and mean time to repair be taken into consideration. This effectively requires that a Markov type of model be used. Section A.2 of Part 6 of the standard specifies the functional steps in applying the analysis of Part 2, and requires that the predicted reliability be compared to the target measures with changes made to the analysis and hardware if necessary.

Annex C of Part 2 defines diagnostic coverage and safe failure fraction, and describes a procedure for calculating them for a subsystem, consisting of components in series, in terms of the same parameters of the components in the subsystem. Diagnostic coverage is the fraction of failures that will be detected by diagnostic tests, and safe failure fraction is the fraction of failures that will not lead to a loss of the safety function. These are some of the parameters needed in a Markov model described in Annex B of Part 6. Safe failure fraction is used in IEC 61508 to determine the SIL of a subsystem. The procedure starts with an FMEA without considering diagnostic tests to identify safe and dangerous failure modes of the components. Table A.1 of Part 2 lists potential failure modes of different components. The determination of safe or dangerous failure is dependent on the specific application/function of the subsystem. Next, the failure rates of the components are estimated using data from a recognized industry source, and application-specific data is preferred. The diagnostic coverage for dangerous failures of components is then estimated by considering the effectiveness of diagnostic tests, and safe failure fraction can be calculated as the fraction of total failures that is either a safe failure or a detected dangerous failure. A difficulty of this method is a lack of needed data. No known database exists that contains safe and unsafe failure rates. These parameters may be application specific. Diagnostic coverage is another parameter which is in general not available and design specific.

8.3.6 Reliability Data for Safety Instrumented Systems, PDS Data Handbook, 2006 Edition

The PDS method [Hauge 2006b] was developed by Scientific and Industrial Research at the Norwegian Institute of Technology (SINTEF) to quantify the reliability and the safety and Life Cycle Cost of computer-based systems. The PDS method uses the approach provided by the IEC 61508 standard and accounts for major related factors that affect reliability during system operations, which include all failure categories/causes, CCFs, coverage due to automatic self-tests or incidental operation of personnel, systematic failures, complete safety function, and redundancies and voting logic, etc. These factors are considered to calculate reliability parameters based on input data and failure classification. For example, to model CCFs, the PDS method modifies the beta factor method by considering voting configuration. The PDS method is intended to be a tool for non-experts in reliability that can enhance the use of reliability analysis in engineering applications [Hauge 2006b].

The PDS method presents a detailed failure classification scheme. Generally, the failure causes are random hardware failure and systematic failure consisting of stress failure, design failure, and interaction failures. It classifies the failure modes into the following categories: dangerous detected failures (DD), dangerous undetected failures (DU), spurious trip detected failures (STD), spurious trip undetected failures (STU), and non-critical failures (NONC). The detection means include both automatic self-testing of digital components and maintenance.

An important source of input data to the PDS method is Offshore Reliability Data (OREDA) [1984]. Recommended input data for a single safety system in [Hauge 2006a] are summarized in Table 8-1. A safety system is generally defined to include input/output cards, central processing unit (CPU) that includes memory and watchdog, controllers (i.e., internal bus, communication, etc.), system bus, and power supply [Hauge 2006a].

In Table 8-1, the rate of a critical failure indicates the rate of failures that will cause loss of one of the following system and component functions: (1) shut down when the production is unsafe, i.e., dangerous failures and (2) maintain production when it is safe, i.e., spurious trip failure. Dangerous failures may consist of both undetected and detected failures. The fault coverage is

used to quantify the performance of self-testing and operator testing. For the logic units, the coverage mainly includes failures detected by self-testing. It is assumed that casual observation of control logic failures is very unlikely.

A set of parameters that measures the safety and reliability performance can be calculated using the PDS method.

Table 8-1 SINTEF recommended input data for single safety system.

Control logic units	Critical failure rate	Dangerous failure rate	Coverage of dangerous failures	Coverage of spurious trip failures	Failure Rate of undetected dangerous failures	Failure rate of undetected spurious failure	SFF (Safe Failure Fraction)
Standard Industrial PLC: Single System	$3x10^{-05}$ per hour	$1.5x10^{-05}$ per hour	67%	20%	$5.0x10^{-06}$ per hour	$1.2x10^{-05}$ per hour	83%
Programmable Safety System: Single System	$2x10^{-05}$ per hour	$1x10^{-05}$ per hour	90%	20%	$1.0x10^{-06}$ per hour	$8.0x10^{-06}$ per hour	95%
Hardwired Safety System: Single System	$2x10^{-06}$ per hour	$1.0x10^{-06}$ per hour	90%	0%	$1.0x10^{-07}$ per hour	$1.0x10^{-06}$ per hour	95%

8.4 Reliability Data Collection and Analysis for Digital Systems and Components from Industrial Operational Experience

Digital systems and components investigated in this section are very diversified in terms of level, complexity, system architectures, and industrial application. The details regarding the data collection and analysis are very different from one database to another. The applicability of each set of data should be defined before a specific use. In each subsection here, background information of collected data such as system and component descriptions and individual applications is provided.

8.4.1 Digital Core Protection Calculators of Combustion Engineering Reactor Protection System

NUREG/CR-5500, Vol. 10 [Wierman 2002], reports an unavailability study of the Combustion Engineering Reactor Protection System (CE RPS) that contains digital core protection calculators (CPCs). The study documents an analysis of operational experience data of the CE RPS from 1984 through 1998, and compares the results with models used in PRA and individual plant examination analysis.

CE RPS systems have four different configurations. One of the most important differences between these configurations is whether digital CPCs are used. Digital CPCs take inputs from Hot Leg and Cold Leg Temperature Sensor/Transmitters and decide whether the reactor should be tripped. A diagram of a digital core protection calculator system (CPCS) channel (Channel A) can be found in Bickel [2006], and each channel of the CPCS is a digital CPC. A digital CPC consists of computer boards, memory boards, and a multiplexer, etc. Different channels (e.g., channels A, B, C, or D) might include different numbers of boards. In this study,

all the components (or circuit boards) of a single channel of the digital CPCS are lumped together as a basic component in the RPS. From the period 1984 through 1998, there were no RPS failures in 612 demands (unplanned reactor trips). To estimate a realistic RPS unavailability, a RPS fault tree was built with digital CPCs modeled as basic components.

The major task in the CE RPS study was to collect and manipulate the basic component data. This task included failure data collection and characterization, demand data collection, and data analysis. The RPS data were first collected from LERs. Since LERs normally do not report RPS independent component failures, the LER search was supplemented by an NPRDS (Nuclear Plant Reliability Data System) data search. The SCSS (Sequence Coding and Search System) database was also searched for all RPS failures over the same period. In addition, the NRC PI (Performance Indicator) Database and the 1987-1998 database used for initiating events study, NUREG/CR-5750 [Poloski 1999], were compared to obtain a list of unplanned RPS demands (reactor trips).

The failure data were further classified into three categories of safety function impacts (non-fail-safe, unknown, and fail-safe) and three degrees of failure completeness (complete failure, unknown failure, and no failure). This resulted in a three-by-three matrix with different bins into which an event could be placed.

An Alpha factor method [Marshall 1998] was used to quantify the CCF events collected from the database. The digital CPCS failure data from the study are summarized in Table 8-2 after an update of the data using other PWR RPS data and a screening process which is based on the safety function significance of the failure and the failure completeness (degradation) value. Component codes in Table 8-2 were defined in NUREG/CR-5500, Vol. 10 [Wierman 2002]. "N/A" in Table 8-2 means "Not Applicable". The unavailability (failure probability), 5% bound, and 95% bound are also shown in Table 8-2. The unavailabilities are obtained using a Jeffreys noninformative prior and a Bayesian update technique. Note that the updated Alpha vectors for CE3-CPD-CF-T2OF3TM, CE3-CPD-CF-T3OF4-TM, CE4-CPD-CF-T2OF3-TM, and CE4-CPD-CF-T3OF4TM are all [9.6×10^{-01} 2.9×10^{-02} 1.5×10^{-03} 8.1×10^{-04}].

From Table 8-2, the failure rate of a single digital CPC channel is 1/548 demands=0.0018 per demand. Model variation indicates the type of data grouping used to determine the uncertainty bounds. For example, for the plant-to-plant variation, data are organized by plants to obtain component failure probabilities per plant. Then, the plant failure probabilities are combined to obtain the mean and variance for the component uncertainty distribution.

In Table 8-2, the modeled variation indicates the type of data grouping used to determine the uncertainty bounds. More details regarding Table 8-2 can be found in Appendix A of NUREG/CR-5500, Volume 10.

8.4.2 Eagle-21 Channels of Westinghouse Reactor Protection System

NUREG/CR-5500, Volume 2 [Eide 1999], presents a reliability study of the Westinghouse RPS systems based on operational experience of Westinghouse commercial reactors during the period of 1984 through 1995.

Table 8-2 Failure rate of digital core protection calculator from NUREG/CR-5500, Volume 10.

Component code	Component type	Fault tree basic event	Number of failures (Weighted average number of failure events)	Number of demands	Modeled variation	Distribution	Unavailability of Bayes update: 5%, mean, 95%	Basic event description
CPD	Digital Core Protection Calculator	CE3-CPD-FF-TA, B, C, D; CE4-CPD-FF-TA, B, C, D	1 (1.0)	548	Sampling	Lognormal	$6.5x10^{-04}$ $2.7x10^{-03}$ $6.8x10^{-03}$	A channel (A, B, C, or D) of the digital CPS fails to send a signal to the trip unit
		CE3-CPD-CF-T2OF3TM; CE4-CPD-CF-T2OF3TM;	9	N/A	N/A	Lognormal	$2.3x10^{-05}$ $1.4x10^{-04}$ $3.8x10^{-04}$	CCF of 2 of 3 digital core protection calculators
		CE3-CPD-CF-T3OF4TM; CE4-CPD-CF-T3OF4TM;	9	N/A	N/A	Lognormal	$6.3x10^{-06}$ $5.7x10^{-05}$ $1.8x10^{-04}$	CCF of 3 of 4 digital core protection calculators

The Westinghouse RPS is equipped with either Analog Series 7300 or Eagle-21 channels in an instrumentation rack depending on individual design and solid state protection system (SSPS) trains. The Eagle-21 upgrade to the RPS replaces the channel analog process logic modules with a digital Eagle-21 module. This upgrade effectively increases the online monitoring and diagnostics capabilities and efficiency of testing. The increased on-line monitoring capability results from most failures being detected almost instantaneously, rather than during quarterly testing.

Westinghouse RPS systems have different configurations, i.e., with different numbers of loops. The Westinghouse RPS study is based on a four-loop design with either an Eagle-21 or Analog Series 7300 sensor processing system and an SSPS for the logic cabinet. An RPS with Eagle-21 has four digital process logic modules. The Westinghouse RPS operational data were collected from LERs as reported in the SCSS and the NPRDS from 1984 to 1995 and categorized. Characterized data, including both independent component failures and CCFs, were obtained after collected data were evaluated. The risk-based analysis of the RPS operational data focused on obtaining failure probabilities for component independent failure and CCF events in the RPS fault tree.

The basic event data of a single loop of the digital Eagle-21 presented in NUREG/CR-5500 Volume 2 [Eide 1999] are summarized in Table 8-3 below, where "N/A" means "Not Applicable". Again, the unavailability (failure probability) and bounds were obtained using a Jeffreys noninformative prior and a Bayesian update technique. CCF data were also modeled using the alpha factor method described in Marshall [1998]. A failure detection and repair duration of eight hours was assumed. The failures were annunciated in the control room. The Alpha vectors of basic events WES-C21-CF-2OF3 and WES-C21-CF-3OF4 are each [9.4×10^{-01} 4.4×10^{-02} 1.1×10^{-02} 3.3×10^{-03}].

From Table 8-3, the failure rate of a single Eagle-21 channel processor is 11/972577=1.1×10^{-05} failures per hour.

8.4.3 Operating Experience of Digital Core Protection Calculators of CE RPS

Bickel [2006] reviews and summarizes operating experience to classify the observed failures of the CE digital CPCS into a limited number of categories based on the failure modes and data collected from LERs during the period between January 1984 and September 2006. This effort is intended to be a starting point of risk-informed evaluation of digital RPS systems.

The digital CE CPCS can be decomposed into redundant channels and major channel components that are consistent with the level of detail being reported in the LERs. Digital CPCS Channels A and D each contain a computer board, a memory board, a multiplexer unit, an internal and external watchdog timer, and a set of digital and process input channels. Each of the B and C channels contains an additional computer board and memory board, but does not have an external watchdog timer.

There are seven CE-designed NPPs that use the digital CPCS. Components contained in the digital CPCS at each plant are:

1. 6 computer boards;
2. 6 memory boards;
3. 4 multiplexers;
4. 6 watchdog timers;
5. 8 cold leg temperature channels;
6. 8 hot leg temperature channels;
7. 4 pressurizer pressure channels;
8. 4 upper core level neutron flux channels;
9. 4 middle core level neutron flux channels;
10. 4 lower core level neutron flux channels;
11. 4 Reactor Coolant Pump digital pump speed channels.

Table 8-3 Failure rate of Eagle-21 channel processor of Westinghouse RPS from NUREG/CR-5500, Volume 2.

Component code	Component type	Fault tree basic event	Number of failures (Weighted average number of failure events)	Operating time (Hours)	Modeled variation	Distribution	Unavailability of Bayes update 5%, mean, 95%	Basic event description
C21	Eagle-21 Channel Processor	WES-C21-FF-E21A, B, C, D	11 (10.6)	972577 hours	Plant	Lognormal	7.4×10^{-06} 6.5×10^{-05} 2.1×10^{-04}	A Eagle-21 channel processor (A, B, C, or D) fails to process reactor trip signals and send appropriate outputs to channel bistables (8.2×10^{-06}/h*8h repair time)
		WES-C21-CF-E2OF3	2	N/A	N/A	Lognormal	3.0×10^{-08} 5.1×10^{-07} 1.8×10^{-06}	CCF of 2 or more of 3 Eagle-21 channel modules (1 channel bypassed)
		WES-C21-CF-E3OF4	2	N/A	N/A	Lognormal	3.6×10^{-09} 1.5×10^{-07} 5.8×10^{-07}	CCF of 3 or more of 4 Eagle-21 channel modules

The total operating experience pool was estimated based on the total calendar time from initial criticality, plus 3 months to account for pre-operational testing. Thus, a total minimum exposure time of 145.5 years or $1.3\times10^{+06}$ hours was obtained. The run time of CPCS and its subsystem components was estimated as:

Total CPCS system run time:	$1.3\times10^{+06}$ hours
CPCS channel run time:	$5.1\times10^{+06}$ hours
CEAC (control element assembly calculator) channel run time:	$2.6\times10^{+06}$ hours
Multiplexer run time:	$5.1\times10^{+06}$ hours
Watchdog timer run time:	$5.1\times10^{+06}$ hours

The operating data include operating events requiring the digital CPCS actions, CCFs and single channel failures of the digital CPCS. Each CPCS channel effectively monitors one quarter of the reactor core and makes a projection of core conditions assuming a uniform positioning of all CEAs (Control Element Assemblies), which are monitored by a set of digital CEACs. Table 8-4 summarizes the CCF and single failure data of the digital core protection systems based on Tables 2 and 3 in Bickel [2006]. The descriptions of CCF or single failures are more like failure causes rather than failure modes. The failure data are collected for circuit boards. The failure probabilities and 5% and 95% bounds on failure rates in the table were estimated

using a Jeffreys noninformative prior and a Bayesian update. From the description of CCFs in Bickel [2006], observed CCFs did not cause any reactor trip. In fact, impacts of most of the CCFs are delaying the generation of a trip rather than preventing the trip altogether. Note the fault probabilities in the last column were calculated by multiplying the failure rates by the time it takes to detect the failure. In Table 8-4, fault probabilities are calculated for CCF1 - CCF5 only because the rest of the observed CCFs were of minor safety significance and were neglected. For single failures, fault duration times are not available in Bickel [2006] and thus fault probabilities are not calculated, as marked as N/A (Not Available) in Table 8-4. It should also be pointed out that some acronyms (such as COLSS, RCS, RTD, and OOT) in Table 8-4 are unavailable because they were not provided in Bickel [2006].

8.4.4 Failure Experience of Programmable Logic Controllers Used in Emergency Shutdown Systems of Natural Gas Compression Stations

Mitchell [1993] presents failure data of programmable logic controllers (PLCs) used in emergency shutdown (ESD) systems for natural gas compressor stations. The purpose of ESD systems is to monitor the stations, detect unsafe operating conditions, and mitigate the consequence of unsafe operating conditions, if any, by isolating the station and venting residual gas to the atmosphere.

In order to obtain the plant-specific data, telephone survey results were collected from 16 natural gas compressor station facilities that use PLC-based ESD systems. The results of the telephone survey of operating station personnel, conducted in 1991, are presented. It should be noted that a typical PLC evaluated in this study consists of three input boards with about 40 input signals and two output boards with approximately 20 actuation signals. Therefore, the PLC here is actually a PLC-based digital system that is part of the ESD. The results of the PLC failure data are shown in Table 8-5 (see Table 1 of Mitchell [1993]). In Table 8-5, only survey results from 13 stations are reported because available information from the other three stations is not sufficient to report, according to Mitchell [1993].

Two types of failures are included in Table 8-5. The number of all failures (Column 4) includes all contributors to failure including the PLC failures. It might be affected by specific design and maintenance, e.g., power supply failures and human errors. ESD PLC failure data in Column 5 are limited to PLC failures only. Column 6 indicates the number of unsafe failures of PLCs due to specific design of ESD systems. No unsafe failure manner has been reported because the hardware and configuration of the PLC system are designed to fail safe for a given PLC failure mode. The ESD PLC failure rate column shows the failure rate, i.e., the number of ESD PLC failures (Column 5) divided by the product of years of experience (Column 2) and number of PLCs (Column 3). If the number of ESD PLC failures is zero, then the table lists "No Failure Rate" in Column 7.

If only PLC failures are considered, the failure rate is 13 failures/180 PLC years=0.072 failures/PLC year, which is equivalent to 8.2×10^{-06} failures per hour for a PLC. If PLC failures of all causes are considered, the failure rate becomes 58 failures/180 PLC years=0.32 failures/PLC year.

Table 8-4 Failure rates for different types of CPCS and/or CEAC failures from Bickel [2006].

CCF or Single failure of channel (SFC)	Number observed	Event caused reactor trip	Failure rate (faults/h)	5% Bound on failure rate (faults/h)	95% Bound on failure rate (faults/h)	Fault probability
CCF1: inaccurate cross calibration of ex-core neutron data sets (cross channel, COLSS, etc)	7	No	5.9×10^{-06}	2.9×10^{-06}	9.8×10^{-06}	9.8×10^{-04}
CCF2: computer technicians input wrong data sets to all 4 channels	3	No	2.8×10^{-06}	8.5×10^{-07}	5.5×10^{-06}	1.0×10^{-04}
CCF3: reactor vendor supplies erroneous data sets or software updates input to all channels	2	No	2.0×10^{-06}	4.5×10^{-07}	4.3×10^{-06}	2.2×10^{-02}
CCF4: reactor vendor supplies software update containing latent software design error input to all 4 channels	1	No	1.2×10^{-06}	1.4×10^{-07}	3.1×10^{-06}	5.0×10^{-05}
CCF5: high log power bypass removal set points (for > 1×10^{-4} power) incorrect	3	No	2.8×10^{-06}	8.5×10^{-07}	5.5×10^{-06}	Neglected
CCF6: operators fail to confirm ASI (Axial Shape Index) in all 4 channels when reactor power is greater 20%	2	No	2.0×10^{-06}	4.5×10^{-07}	4.3×10^{-06}	Neglected
CCF7: inaccurate cross calibration of RCS flow data sets (cross channel, COLSS, etc)	2	No	2.0×10^{-06}	4.5×10^{-07}	4.3×10^{-06}	Neglected
CCF8: operators fail to perform 12 hours auto-restart surveillance on all 4 channels	1	No	1.2×10^{-06}	1.4×10^{-07}	3.1×10^{-06}	Neglected
CCF9: operators fail to perform refueling interval surveillance on all 4 channels	1	No	1.2×10^{-06}	1.4×10^{-07}	3.1×10^{-06}	Neglected
CCF10: communication data link failure to plant computer results in missed surveillance on 2 of 2 CEAC channels	1	No	1.2×10^{-06}	1.4×10^{-07}	3.1×10^{-06}	Neglected
CCF11: 2 of 2 CEAC channels inoperable	1	No	1.2×10^{-06}	1.4×10^{-07}	3.1×10^{-06}	Neglected

Table 8-4 Failure rates for different types of CPCS and/or CEAC failures from Bickel [2006].

CCF or Single failure of channel (SFC)	Number observed	Event caused reactor trip	Failure rate (faults/h)	5% Bound on failure rate (faults/h)	95% Bound on failure rate (faults/h)	Fault probability
CCF12: 3 of 4 ex-core neutron flux cross channel calibrations out of tolerance	1	No	$1.2x10^{-06}$	$1.4x10^{-07}$	$3.1x10^{-06}$	Neglected
CCF13: incorrect acceptance criteria used for ex-core neutron data set calibration checks are greater than 80%.	1	No	$1.2x10^{-06}$	$1.4x10^{-07}$	$3.1x10^{-06}$	Neglected
SFC1: CEAC computer board failure	5	Yes	$2.2x10^{-06}$	$9.0x10^{-07}$	$3.9x10^{-06}$	N/A
SFC2: CPC computer board failure	0	No	$2.2x10^{-06}$	$9.0x10^{-07}$	$3.9x10^{-06}$	N/A
SFC3: CEA position transmitter or cable failure	4	Yes	$2.8x10^{-09}$	$1.1x10^{-09}$	$5.4x10^{-09}$	N/A
SFC4: single CPCS channel ex-core data set invalid	4	No	$8.8x10^{-07}$	$3.3x10^{-07}$	$1.7x10^{-06}$	N/A
SFC5: loss of 120V inverter causes CPCS channel trip and CEAC channel trip	3	Yes	$1.4x10^{-06}$	$4.3x10^{-07}$	$2.8x10^{-06}$	N/A
SFC6: single CPCS channel RTD time constants OOT	3	No	$6.9x10^{-07}$	$2.1x10^{-07}$	$1.4x10^{-06}$	N/A
SFC7: CEAC memory board failure	2	Yes	$9.8x10^{-07}$	$2.2x10^{-07}$	$2.2x10^{-06}$	N/A
SFC8: CPCS memory board failure	0	No	$9.8x10^{-07}$	$2.2x10^{-07}$	$2.2x10^{-06}$	N/A
SFC9: single CPCS channel data link failure to plant computer (inability to auto confirm CEA positions)	2	No	$4.9x10^{-07}$	$1.1x10^{-07}$	$1.1x10^{-06}$	N/A
SFC10: DC power supply failure causes CPCS channel failure and CPCS watchdog timer failure	2	No	$4.9x10^{-07}$	$1.1x10^{-07}$	$1.1x10^{-06}$	N/A
SFC11: electrical fault causes CEA data set error– single CPCS channel and single CEAC channel trips	1	Yes	$5.9x10^{-07}$	$6.9x10^{-08}$	$1.5x10^{-06}$	N/A
SFC12: MUX failure causes CEA data set error – single CPCS channel and single CEAC channel trips	1	Yes	$2.9x10^{-07}$	$3.5x10^{-08}$	$7.7x10^{-07}$	N/A

Table 8-4 Failure rates for different types of CPCS and/or CEAC failures from Bickel [2006].

CCF or Single failure of channel (SFC)	Number observed	Event caused reactor trip	Failure rate (faults/h)	5% Bound on failure rate (faults/h)	95% Bound on failure rate (faults/h)	Fault probability
SFC13: single CPCS channel DNBR (Deviation from Nucleate Boiling Ratio) /LPD (Linear Power Density) time constant data set invalid	1	No	2.9×10^{-07}	3.5×10^{-08}	7.7×10^{-07}	N/A
SFC14: operators fail to perform monthly CEAC surveillance	1	No	5.9×10^{-07}	6.9×10^{-08}	6.9×10^{-08}	N/A

Table 8-5 Failure rate of PLCs used in emergency shutdown systems from Mitchell [1993].

System identification number	Years of experience	Number of PLCs	All failures	ESD PLC failures	Non-fail-safe failures	ESD PLC failure rate (per PLC year)
1	5	8	15	5	0	0.13
2	2	4	3	2	0	0.25
3	2	3	0	0	0	No Failure Rate
4	3	5	1	0	0	No Failure Rate
5	2	5	2	0	0	No Failure Rate
6	2	4	2	2	0	0.25
7	2	3	1	0	0	No Failure Rate
8	4	6	5	3	0	0.13
9	1	1	4	0	0	No Failure Rate
10	3	2	8	0	0	No Failure Rate
11	2	6	12	1	0	0.083
12	2	1	0	0	0	No Failure Rate
13	6	7	5	0	0	No Failure Rate
Combined Experience (180 PLC years)			58	13	0	

8.4.5 Operational Failure Experience of Fault-Tolerant Digital Control Systems in Different Industries

Paula [1993a] tries to identify major contributors to fault-tolerant digital control system (F-T DCS) unavailability in addition to presenting actual failure experience from 20 different digital system installations. The advantage of a F-T DCS is that it will continue to function for most single hardware failures.

The study did not compare equipment from different manufacturers but it did compare the digital control system with different architectures, i.e., the dual and the TMR (triple modular redundancy). In the study, digital control systems were defined to include main processors, input/output modules, and input sensors and instrumentation. Actuators were not considered part of the digital control systems. Dual redundancy and triple redundancy could also be adopted in input/output modules for some critical applications. The voting logic was either implemented using hardware or software (HIFT: hardware implemented fault tolerant and SIFT: software implemented fault tolerant).

For input modules, two types of failure are considered: failures that affect signals to all processors (e.g., failure of multiplexing devices) and failures that affect signals to one of the processors only (e.g., failure of transceivers to one of the processors). Similarly, for the output modules, two types of failures considered in the study are those that affect signals from all processors (these cause a system failure if the system does not have redundant output modules) and those that affect signals from one of the processors only (e.g., failure of a transceiver). Failure data collected from different industries presented in several tables of Paula [1993a] are summarized in Table 8-6. Note, all the failures here indicate the failure of the full F-T DCS. Also, the details of computer systems 1 - 10 (System ID 1 - 10 in Table 8-6) are not clear, and most of the desired functions and complexities of these computer systems are unknown.

In Table 8-6, "NA" indicates that the data are not available. The number in parentheses indicates the fraction of the total number of known system failures attributed to the corresponding type of system failure. System 5 has a hard-wired backup computer, and thus does not have a software CCF issue. Types of failures for systems 11, 12, and 13 are not differentiated in the analysis. Systems 17 and 18 are used to control circuit breakers and their control functions are less complex compared to other systems in Table 8-6. From Table 8-6, the total operating time in terms of system years is 174 and the total operating time in terms of channel years is 346.5. Therefore, the failure rate of the digital systems is 40/174 system years = 2.6×10^{-05} per hour for a digital system, and the failure rate of a single channel of digital systems is 279/346.5 channel years = 9.2×10^{-05} per hour.

CCF rates can also be calculated. The hardware CCF rate is 1/174 = 6.6×10^{-07} per hour and the software CCF rate is 9/194 = 5.9×10^{-06} per hour. Note that failures due to loss of power supply are not included in the hardware CCF estimate.

Tables 8-7 and 8-8 summarize data predicated in Paula [1993a] using an earlier version of Military Handbook 217 (217D) for digital components from OREDA [1984], Humphreys & Daniels [1982], and Triconex [Triconex]. Since the failure indicates the components or modules only, redundancy is not considered in these calculations.

Table 8-6 Failure data of fault-tolerant digital control systems from Paula [1993a].

System ID	Redundancy	Total operating time (System Years)	*Number of failure events attributable to each type of failure*								
			Spurious	Inadvertent	Hardware CCF	Software CCF	External physical damage	Electric power supply	Unidentified	System failure (All Types)	Single channel failure
1	1-out-of-2	3	0	1	0	2	0	0	3	6	11
2	1-out-of-2	6	0	1	0	4	0	0	6	11	23
3	1-out-of-2	9	0	0	0	0	0	1	0	1	55
4	2-out-of-4	3	0	0	1	0	0	0	0	1	7
5	1-out-of-2 diverse computer	7	0	0	0	0	0	1	0	1	120
6	1-out-of-2	40	0	0	0	0	0	4	0	4	NA
7	1-out-of-2	10	0	0	0	0	0	0	0	0	NA
8	1-out-of-2	1.3	0	3	0	0	0	0	0	3	0
9	1-out-of-2	0.5	0	0	0	3	1	0	2	6	33
10	1-out-of-2	10	2	0	0	0	0	0	0	2	30
11	TMR	1.9	NA	NA	NA	NA	NA	NA	NA	NA	NA
12	TMR	1.4	NA	NA	NA	NA	NA	NA	NA	NA	NA
13	TMR	4.9	NA	NA	NA	NA	NA	NA	NA	NA	NA
14	Dual	2.8	0	0	0	0	0	0	0	0	NA
15	Dual	4.5	1	0	0	0	0	0	0	1	NA
16	No	3.8	0	0	0	0	0	0	0	0	NA
17	No	4.9	1	0	0	0	0	0	0	0	NA
18	No	1.0	2	0	0	0	0	0	0	0	NA
19	Dual	45	0	4	0	0	0	0	0	4	NA
20	Dual	14	0	0	0	0	0	0	0	0	NA
All Systems Excluding 5, 11 - 13,17, & 18			3(0.11)	9 (0.32)	1(0.04)	9(0.32)	1(0.04)	5(0.17)	11	39	159
All Systems			6	9	1	9	1	6	11	40	279

Table 8-7 Digital system component failure rates from OREDA-84 and Humphreys & Daniels.

Digital control system component	Failure rate estimate	Sources
Processor (CPU/memory)	0.45 per year (5.1×10^{-05} per hour)	OREDA
	1.8 per year (2.1×10^{-04} per hour)	Humphreys & Daniels
Input/Output Modules	0.071 per year (8.1×10^{-06} per hour)	Humphreys & Daniels
Input/Output Modules: Digital Input Cards (typically 16-32 channels)	0.0073 per year (8.3×10^{-07} per hour)	OREDA
Input/Output Modules: Digital Output Cards (typically 16-32 channels)	0.072 per year (8.2×10^{-06} per hour)	OREDA
Power Supply (Single Source)	0.25 per year (2.9×10^{-05} per hour)	OREDA
	0.14 per year (1.6×10^{-05} per hour)	Humphreys & Daniels

Table 8-8 Digital system component failure rates of a TMR system from Triconex and Humphreys & Daniels.

Digital control system component	Failure rate estimate	Sources
Processor (CPU/memory)	0.33 per year (3.8×10^{-05} per hour)	Triconex
	1.3 per year (1.5×10^{-04} per hour)	Humphreys & Daniels
Input/Output Modules	0.096 per year (1.1×10^{-05} per hour)	Humphreys & Daniels
Digital Input Cards	0.086 per year (9.8×10^{-06} per hour)	Triconex
Digital Output Cards	0.040 per year (4.6×10^{-06} per hour)	Triconex
Analog Input Cards	0.11 per year (1.3×10^{-05} per hour)	Triconex
Analog Output Cards	0.15 per year (1.7×10^{-05} per hour)	Triconex
Power Supply (Single Source)	0.14 per year (1.6×10^{-05} per hour)	Triconex
	0.52 per year (5.9×10^{-05} per hour)	Humphreys & Daniels

The failure rates of processors and single source power supplies in Tables 8-7 and 8-8 are within a range of 5 and 4, respectively. The failure rates of digital input cards present a large discrepancy (a factor of about 12), as shown in Tables 8-7 and 8-8.

8.4.6 Failure Rates for Programmable Logic Controllers Used in Chemical and Nuclear Plants

Failure data of PLCs obtained from chemical and nuclear plants were used to estimate PLC failure rates in Paula [1993b]. The three different types of plants include a U.S. phenol plant, French NPPs, and a Canadian NPP. The failure data were obtained from the PRA performed by JBF Associates, Inc., and personnel responsible for PLC operation and maintenance in these

plants. It should be pointed out that although these PLCs might include a number of circuit boards or modules, the failure data record processor failures only.

For a PLC with redundant processors, i.e., the PLC can recover from a single processor failure by successfully switching the control functions to other processors, coverage values can also be calculated from the collected data. The failure data of about 35 fault-tolerant dual-redundancy processors used in the phenol plant [Paula 1993b] are summarized in Table 8-9. These applications are mainly about a variety of control and shutdown functions. Only six PLCs perform critical functions, and these PLCs have redundant input/output modules.

Table 8-9 Summary of PLC failure data from a U.S. phenol plant from Paula [1993b].

Chemical plant	Number of complete PLC failures (Both primary and secondary processors)	Total PLC years of operation	PLC failure rate (per PLC Year)
Phenol	4	7*35=245	0.016

Note that two of the reported failures occurred during the first few months of PLC operation. If this time period is not included, the PLC failure rate becomes about 0.0082/PLC year, which is equivalent to 9.361×10^{-06} failures per hour for a PLC.

All PLC systems in the phenol plant have dual redundancy. Plant personnel reported a failure rate per processor of two failures per year. Coverage can be defined as the fraction of failures that a PLC can recognize and isolate the signal of the failed processor to prevent process upsets. In order to calculate the coverage value, it is assumed that two of the four complete failures occurred due to a lack of coverage and only failures of the primary processor of the PLC challenge the PLC coverage. Therefore, the total number of primary processor failures is 2 failures per PLC year * 7 years * 35 PLCs = 490 based on the failure rate of the primary processor, and the coverage value = (490-2)/490 = 0.996.

Failure data of PLCs from French NPPs presented in Paula [1993b] are summarized in Table 8-10. All of these PLCs have fault-tolerant redundant architecture. The functionality of these PLCs is to perform control interlocks in these power plants. From Table 8-10, the PLC failure rate becomes 1.3×10^{-06} failure per hour if power supply failures are excluded.

There were about 500-600 PLCs in a Canadian NPP for the distributed control. These PLCs had been in operation for about 2 years when the failure data were collected. Each PLC typically had 8 boards or cards. The failure rate calculated from the actual failure data (the actual failure data were not presented in this study) was 0.025/PLC year and the original theoretical failure rate predicted for this application was 0.3/PLC year.

Coverage values for PLCs in NPPs are not available. A summary of the failure data from three sources is shown in Table 8-11, where "NA" indicates "Not Available". It is not clear whether PLCs in the Canadian nuclear power plant have redundancy, though from the failure rates in Table 8-11, it seems PLCs in the Canadian plant might have redundant processors.

Table 8-10 Summary of PLC failure data from French NPPs from Paula [1993b].

Year	Population of PLCs	Cumulative operation time (PLC Years)	Number of total failures (both processors)		PLC failure rate (per PLC year)	
			Excluding power supply	Including power supply	Excluding power supply	Including power supply
1	630	630	10	18	0.016	0.029
2	873	873	7	11	0.008	0.013
3	1201	1201	12	29	0.01	0.024
All Years		2704	29	58	0.011	0.021

Table 8-11 Failure rate and coverage of PLCs in chemical and nuclear power plants from Paula [1993b].

PLC application	Single processor failure rate (per PLC Year)	Total PLC failure rate (per PLC Year)	Coverage
USA Phenol Plant	2	0.082 (excluding failures in the first few months)	0.9918 - 0.9959 (Excluding failures in the first few months)
		0.016 (Including failures in the first few months)	0.9837 - 0.9918 (Including failures in the first few months)
French-design Nuclear Power Plants	NA	0.011 (excluding power supply failures)	NA
		0.021 (Including power supply failures)	NA
A Canadian Nuclear Power Plant	NA	0.025	NA

The failure rates of PLCs in Table 8-11 are very consistent although the applications are very different. Also, the PLC applications in the study are relatively simple. The failure rates are anticipated to increase with the complexity and size of the systems. It is claimed in the study that the failure rates of fault-tolerant digital control systems are about 15 - 50 times higher than the PLC failure rates in Table 8-11.

8.4.7 Savannah River Site (SRS) Generic Data Development Based on Data from Different Industries

The SRS generic data project intended to develop a generic database that can be used in the nuclear industry by collecting data from different industrial applications. The data collection was not limited to digital systems and components. Failure data of a set of components and the associated failure modes are shown in SRS [Blanton 1993]. There are three categories of data depending on the source of the data. The Category 1 data are from sources with actual failure

data obtained from a detailed review of failure events and a detailed review of component populations and exposure durations (or demands). Such data include the plant-specific component failure data collected from PRAs or reliability studies.

Category 2 data are from sources with actual failure data, but which have an added uncertainty in the data compared to Category 1 sources. This added uncertainty can result from a less comprehensive search for actual failures, a more approximate method for determining component populations or exposure durations (or demands), or a less clear breakdown of failures into the failure modes of concern.

Category 3 data are from sources that list only the failure rate estimates.

Two different routines were used to aggregate the collected data.

Aggregation Routine 1 for data in Categories 1 and 2:

1. Compute point estimates of source data;
2. Match moments to obtain underlying normal distribution;
3. Determine mean and error factor of lognormal distribution.

Aggregation Routine 2 for data in Category 3:

1. Determine variance for each source;
2. Determine natural logarithm of median for each source;
3. Determine average of source variances;
4. Determine average of natural logarithms of medians;
5. Determine variance of natural logarithms of medians;
6. Determine mean and error factor of lognormal distribution.

Details of the above aggregation routines can be found in Blanton [1993]. Resultant failure rates of different systems or components from Blanton [1993] are shown in Table 8-12. Reliability data of various sensors are also shown in Table 8-12 because they are necessary input data to evaluate reliability of digital systems. Regarding the data of microprocessors, Table 8-12 only lists a failure rate for generic PLC devices, and the error factor is 10.

8.4.8 Failure Parameters of Digital Trip Module (DTM) and Trip Logic Unit (TLU) in Economic Simplified Boiling Water Reactor (ESBWR) Probabilistic Risk Assessment (PRA)

Many systems in an ESBWR design [GE 2006] contain digital parts. Digital systems and components were explicitly modeled and integrated into the ESBWR PRA. Failure modes of digital system components were defined at a very high level, e.g., TLU fails to trip or DTM fails to trip in RPS system. Both DTM and TLU are microprocessor-based modules that consist of an unknown number of circuit boards or cards.

Generic reliability data for the ESBWR PRA are based on data from the Advanced Light Water Reactor Utility Requirements Document. The reliability data of microprocessor-based components and discrete logic components are from GE. The failure rate for DTMs and TLUs is selected to be 5.0×10^{-06}/h based on a required MTBF of 200,000 hours. To obtain the failure probability on demand, it is assumed that 95% of the component failures would be detected by self-testing. Both DTMs and TLUs are self-tested every 30 minutes ($T_{self\text{-}test}$). The remaining 5% would be detected only during surveillance tests performed quarterly (test period is 2190 hours,

Table 8-12 Savannah river site (SRS) generic data from Blanton [1993].

System/Component failure Mode	SRS aggregated and recommended failure rates: Mean (error factor)			
	Category 1	Category 2	Category 3	Recommended by [Blanton 1993]
Alarm/Annunciator Failure to Alarm		$3.6x10^{-05}$/h (10)	$1.5x10^{-06}$/h (15)	$3.0x10^{-05}$/h (10)
Alarm/Annunciator Spurious Failure		$4.7x10^{-06}$/h (10)	$8.7x10^{-07}$/h (15)	$5.0x10^{-06}$/h (10)
Sensor/Transmitter/ Transducer/Process Switch Temperature Failure	$2.8x10^{-04}$/d (3.5) (Spurious: $1.1x10^{-06}$/h (3.3)	$8.6x10^{-07}$/h (7.9)	$8.1x10^{-06}$/h (14)	$1.0x10^{-06}$/h (3)
Sensor/Transmitter/ Transducer/Process Switch Pressure Spurious Failure	$2.8x10^{-04}$/d (3.5) (Spurious: $1.1x10^{-06}$/h (3.3)	$8.3x10^{-07}$/h (3.1)	$6.8x10^{-06}$/h (17)	$1.0x10^{-06}$/h (3)
Sensor/Transmitter/ Transducer/Process Switch Differential Pressure Failure		$2.7x10^{-06}$/h (10)	$1.2x10^{-04}$/h (16)	$3.0x10^{-06}$/h (10)
Sensor/Transmitter/ Transducer/Process Switch Flow Failure		$2.9x10^{-06}$/h (2.0)	$3.2x10^{-05}$/h (15)	$3.0x10^{-06}$/h (3)
Sensor/Transmitter/ Transducer/Process Switch Level Failure		$5.3x10^{-07}$/h (3.7)	$6.4x10^{-06}$/h (7.7)	$5.0x10^{-07}$/h (3)
Sensor/Transmitter/ Transducer/Process Switch Humidity Failure		$1.2x10^{-05}$/h (10)	$4.2x10^{-06}$/h (3.9)	$1.0x10^{-05}$/h (10)
Sensor/Transmitter/ Transducer/Process Switch pH Failure			$5.8x10^{-07}$/h (5.0)	$5.0x10^{-07}$/h (5)
Sensor/Transmitter/ Transducer/Process Switch Oxygen Concentration Failure		$9.5x10^{-06}$/h (10)		$1.0x10^{-05}$/h (10)
Sensor/Transmitter/ Transducer/Process Switch Hydrogen Concentration Failure			$9.8x10^{-05}$/h (10)	$1.0x10^{-04}$/h (10)
Sensor/Transmitter/ Transducer/Process Switch Nitrogen Concentration Failure				$1.0x10^{-05}$/h (3)

Table 8-12 Savannah river site (SRS) generic data from Blanton [1993].

System/Component failure Mode	SRS aggregated and recommended failure rates: Mean (error factor)			
	Category 1	Category 2	Category 3	Recommended by [Blanton 1993]
Sensor/Transmitter/ Transducer/Process Switch Hydrocarbon Concentration Failure		7.9×10^{-06}/h (2.2)		1.0×10^{-05}/h (3)
Sensor/Transmitter/ Transducer/Process Switch Helium Concentration Failure				1.0×10^{-05}/h (3)
Sensor/Transmitter/ Transducer/Process Switch Speed Failure			1.7×10^{-06}/h (8.1)	1.0×10^{-06}/h (10)
Sensor/Transmitter/ Transducer/Process Switch Seismic Failure			1.6×10^{-06}/h (4.0)	1.0×10^{-06}/h (5)
Sensor/Transmitter/ Transducer/Process Switch Radiation Failure			6.3×10^{-06}/h (4.7)	5.0×10^{-06}/h (5)
Indicator Failure		1.5×10^{-05}/h (10)	1.1×10^{-04}/h (22)	1.0×10^{-05}/h (10)
Amplifier Failure				5.0×10^{-06}/h (10)
Modifier/Signal Conditioner Failure			3.3×10^{-07}/h (2.1)	3.0×10^{-07}/h (3)
Logic Module Failure			3.7×10^{-06}/h (7.7)	3.0×10^{-06}/h (5)
Recorder Failure			3.8×10^{-05}/h (22)	3.0×10^{-05}/h (30)
Sampler Failure		1.2×10^{-05}/h (10)		1.0×10^{-05}/h (10)
Analyzer Failure		6.0×10^{-06}/h (10)	1.3×10^{-03}/h (26)	5.0×10^{-06}/h (10)
Timer Failure				5.0×10^{-06}/h (10)
Gas Chromatography Failure		7.3×10^{-05}/h (10)		5.0×10^{-05}/h (10)
Voltage Regulator Failure		3.2×10^{-06}/h (10)		3.0×10^{-06}/h (10)
Transmitter Failure	2.2×10^{-06}/h (1.4)		1.0×10^{-05}/h (30)	3.0×10^{-06}/h (10)
Transducer Failure		9.2×10^{-07}/h (10)	6.3×10^{-05}/h (15)	1.0×10^{-06}/h (10)
PLC		3.2×10^{-05}/h (10)		3.0×10^{-05}/h (10)

T_{test}). A MTTR of five hours is assumed for both cases. The expression used for the probability calculation is:

$$P=\lambda[0.95(\frac{T_{self-test}}{2}+MTTR)+0.05(\frac{T_{test}}{2}+MTTR)]$$

where λ is the failure rate. For the digital I&C components, all the error factors are 10, which is considered to represent large uncertainty.

Software failure is modeled to contribute to the CCF of DTMs by adding a value of 1.0×10^{-05} to the CCF. Thus, software-introduced common cause failures are accounted for. The alpha-factor method is used to model hardware CCF. The CCF factors (f_{cc}) for DTMs and TLUs are both 3.0×10^{-03}.

8.4.9 Reliability Study of Digital Engineered Safety Feature Actuation System of Korean Standard Nuclear Power Plant

The Ulchin Nuclear Power Plant Units 5 and 6 (UCN 5&6) are the first Korean Standard Nuclear Power Plants (KSNPPs) that have installed a Digital Plant Protection System (DPPS), which includes a Digital Reactor Protection System and a Digital Engineered Safety Features Actuation System (DESFAS). The KSNPP further developed some Combustion Engineering System 80 features and incorporated many of the US Advanced Light Water Reactor design requirements from 1984. The DESFAS actuation signals are received from the DPPS channels, and then processed by the DESFAS pump/valve coincidence logic processor according to the selective 2-out-of-4 coincidence logic. The output signals are transmitted to the plant control system that controls the actuated equipment of the ESFAS system.

The DESFAS Auxiliary Cabinet consists of two redundant trains, i.e., Train A and Train B. Each train contains a cabinet divided into three bays, pump processors, valve processors, communication interface processor, maintenance and test panel, PLC internal network, cooling and ventilation system, and power supply system.

The reliability analysis was performed using data from generic sources. The main source of the data is the Westinghouse document on "Unavailability Analysis for the Digital Plant Protection System" [Westinghouse 2001]. The beta-factor model was used to model CCFs in KAERI/TR-2468 [Varde 2003] and an MGL model is used in KAERI/TR-2467 [Sudarno 2003].

A beta factor of 0.03 was used to represent the CCF of digital components in KAERI/TR-2468 [Varde 2003]. The reliability analysis assumed that contribution from software to the total failure probability of processors used in the digital system is 10% of the hardware failure probabilities. However, the software CCF data are not available in the report.

The component failure rates are shown in Table 8-13. The components are not limited to digital components. Components here actually represent modules of digital systems. The error factors are relatively small. Failure modes are defined loosely, i.e., a component (module) fails to perform its intended function.

Table 8-13 Generic component data for a Korean DESFAS reliability analysis from Varde [2003].

Components	Failure Mode	Failure Rate	Error Factor	Distribution
Processor Module (Advent 645C and primary rack)	Fail to generate trip output	3.2×10^{-06} per hour	3	Lognormal
Digital Input Module (Advent DI620)	Fail to generate trip output	9.0×10^{-07} per hour	3	Lognormal
Analog Input Module (Advent AI620)	Fail to generate trip output	2.0×10^{-06} per hour	3	Lognormal
Digital Output Module (Advent DO630)	Fail to generate trip output	8.2×10^{-07} per hour	3	Lognormal
Fiber Optic Transmitter	Fail to actuate	4.4×10^{-06} per hour	3	Lognormal
Watchdog Timer	Fail to open	8.2×10^{-08} per demand	3	Lognormal
Instrument Power Supply	Fail to supply	1.6×10^{-03} per demand	3	Lognormal
Pressure Transmitter	Fail to provide	4.4×10^{-06} per hour	3	Lognormal

8.4.10 Digital RPS and ESFAS of AP600 Reactors

AP600 [Westinghouse 1996] is an advanced nuclear plant design that has been reviewed and approved by the US NRC. It has an integrated digital I&C architecture. The RPS and ESFAS functions are performed by the protection and safety monitoring system (PMS). In addition, the plant control system (PLS) that controls non-safety related functions, e.g., chemical and volume control system pumps and the feedwater system, was modeled. The following is a description of the integrated protection cabinet part of the PMS and the associated fault tree modeling.

The integrated protection cabinet contains the necessary equipment to actuate reactor trip and engineered safety features (ESFs). It consists of four divisions, each division consisting of subsystems including two reactor trip groups, two ESF groups, a global trip subsystem, a trip enable subsystem, a communication subsystem, and an automatic tester subsystem. Each subsystem includes: functional processor, bus monitor card, data link processor, data highway control, parallel input/output (I/O) card, isolated parallel I/O card, analog input processor, universal site memory expansion card, digital/analog conversion card, and test bus controller.

The fault tree model has basic events that model power interface boards, analog input boards, logic groups, multiplexer transmitters, analog output boards, and sensors. Hardware failure data for microprocessor-based components were derived from Westinghouse data. In the generic data table in the AP600 safety analysis report (Table 32-1) the only relevant data for microprocessor-based components is a failure rate of 5.0×10^{-6} per hour for logic cards. CCFs were modeled for most digital components. The CCF data source is not provided. Software CCFs are modeled with probabilities from 1.0×10^{-5} to 1.0×10^{-6}, which are considered the goals of the design.

8.4.11 Digital Systems of AP1000 Reactors

There are a number of digital systems and/or components in an AP1000 reactor design. The PRA document of AP1000 [Westinghouse 2004] describes the reliability model containing digital systems and/or components of the AP1000 reactor. Plant protection, control, and monitoring functions of the AP1000 are performed by the PMS, the PLS, the special monitoring system, and the in-core instrumentation system. The system that is most important to safety is the PMS.

All of the logic and instrumentation failures are modeled at the level of circuit board or line replaceable unit. The AP1000 PRA did not present a detailed failure mode analysis for digital systems. The failure data for microprocessor-based components were obtained from Westinghouse. The failure rate of a single logic card (microprocessor-based) is 5.0×10^{-06} per hour with an error factor of 10 for all failure modes.

The CCF due to software is modeled in the PRA analysis. A generic software failure model is considered in the fault trees. This model produces a software CCF probability of 1.2×10^{-06} failures per demand for software failures that would manifest themselves across all types of software modules derived from the same basic design program in all applications.

8.4.12 Reliability Modeling of Digital Instrumentation and Control Systems for Nuclear Reactor Probabilistic Risk Assessments

A Markov approach is used to assess reliability of digital feedwater control system in Aldemir [2007]. Failure parameters from Table 2.4.1 of [Aldemir 2007] are presented in Table 8-14.

According to Aldemir [2007], reliability data in this analysis are obtained from actual failure data of a digital feedwater control system (DFWCS), which was first installed in 1993 in an operating PWR NPP. Overall, the components in the system are very reliable. Controllers of the DFWCS, had not failed at the time of the study. The commercial database of component failures, PRISM, was used. Interviews with key vendors of digital systems were conducted in order to acquire "off-the-record" failure data information. All failures are permanent failures and do not include transients in the reports and failure logs. The failure rates obtained are considered conservative. An uncertainty factor of 10 is recommended by vendors for the failure data. However, an uncertainty analysis on the failure rates of the components shown in Table 8-14 was not conducted in Aldemir [2007]. Another important reliability parameter is fault coverage value for individual DFWCS components. How to obtain fault coverage values were discussed in detail but the values were not provided in Aldemir [2007].

8.5 Data Sources in Nuclear Industry

8.5.1 Licensee Event Report (LER) Database

LERs are required by Title 10 of the Code of Federal Regulations (CFR) Part 50.73 [CFR 2007] and the reporting criteria are specified in NUREG-1022, Rev. 2 [NRC 2000b]. Based on these criteria, many non-safety-related digital equipment failures may not be reported. This imposes a limit on the use of LERs to estimate component failure rates. It should be valid to use LERs to estimate the contribution of digital components to initiating events, because every initiating event should be reported. Idaho National Laboratory (INL) currently maintains an LER search system [INL] that is available to NRC staff and contractors.

Table 8-14 Example failure parameters of DFWCS components from Aldemir [2007].

Component	Example failure rates
Power Level Sensor	4.5×10^{-07} per hour
Steam flow sensor	1.9×10^{-06} per hour
Water Flow Sensor	1.5×10^{-06} per hour
Water Temperature Sensor	1.0×10^{-06} per hour
Water Level Sensor	1.0×10^{-06} per hour
Main Controller	3.7×10^{-05} per hour
Backup Controller	3.7×10^{-05} per hour
MFV Controller	1.0×10^{-06} per hour
BFV Controller	1.0×10^{-06} per hour
PDI Controller	1.0×10^{-06} per hour
FWP Controller	1.0×10^{-06} per hour
Main Flow Valve	1.2×10^{-06} per hour
Bypass Flow Valve	1.2×10^{-06} per hour
Feedwater Pump	3.1×10^{-06} per hour

An LER search was performed to identify failures associated with microprocessors. The LERs were reviewed to determine the causes of the failures and their effects. It is often difficult to determine if a failure is associated with the microprocessor itself or the circuit board/module containing the microprocessor. It is suspected that the reports do not always clearly make the distinction. This could affect the level of detail at which failure parameters can be estimated using the LERs. In general, the LERs do not contain information about how long the failed components have been operating, and how many of the same or similar components are operating at the specific plant and other plants. This information is required to estimate realistic component failure rates.

A search of the title and abstract fields of the LER database was performed by specifying "microprocessor" as the key word. A total of 45 LERs were identified and reviewed to determine the causes of the failures and their effects. These reports are categorized by their failure causes which are identified below:

1. Microprocessor (8 events)

   These events represent failures that may be due to hardware and/or software failures. The most notable event occurred at Pilgrim on April 1, 1997, during which a software CCF caused 3 voltage regulating transformers to trip and a loss of power at 3 safety related 120 volt AC buses occurred. The software was programmed to shutdown the 480/120 volt transformers when the 480 volt power supply became undervoltage or overvoltage. It is not obvious why the uninterruptible buses which are backed up by

batteries became unavailable simply because the transformers were shutdown. It is also not clear if the transformers would be adversely affected by the power transient if the software were not programmed to shutdown the transformers.

2. Supporting components/circuitry (20 events)

   These events represent failures of the components and circuitry on the circuit board/module containing the microprocessor. The specific components that had failed include power supplies, PROM, hold down clip, terminal connections, and communication.

3. External disturbance (8 events)

   These events represent failures of the microprocessors due to electrical noise or spike.

4. Human factor (5 events)

   These events represent human errors in leaving a control system in an inoperable or undesirable state, or human error induced power spikes which caused system failures.

5. Non-microprocessor related failure cause (4 events)

   These events represent failures due to external causes, i.e., external power supply problem, and failures not directly associated with a microprocessor.

As mentioned earlier, since operating times of individual systems are not available, failure rates cannot be estimated using this data.

8.5.2 Equipment Performance and Information Exchange (EPIX) Database

The EPIX System is a web-based database system. It includes equipment failure records voluntarily provided by the utilities, and replaces the Nuclear Plant Reliability Data System (NPRDS) as the reliability database of the nuclear industry in the United States. Its failure record scope includes:

- key components in the World Association of Nuclear Operators (WANO) safety system performance indicator (SSPI) program
- key components in the Maintenance rule scope [NRC 2000a, NEI 2002]
- other components that caused generation losses.

For information on total demands and operating hours, its scope includes only key components of WANO SSPI systems and key components that perform risk-significant functions in non-WANO SSPI systems.

It seems that instrumentation and control (I&C) components may only be within the scope of EPIX for those plants that include the RPS and ESFAS within their Maintenance Rule risk-significant scope. This limitation of EPIX is probably not as important as the voluntary nature of the failure reporting, because RPS and ESFAS are important systems.

The NRC has developed the Reliability and Availability Data System (RADS) to provide the reliability and availability data needed by the NRC to perform generic and plant-specific assessments and to support PRA and risk-informed regulatory applications. The NRC is incorporating data from EPIX and the WANO's SSPI system along with information from other data sources (e.g., LERs and monthly operating reports) into RADS. Data are available for the major components in the most risk important systems in both boiling water reactors and pressurized water reactors (PWRs).

RADS was developed to support the generic failure database needed in NRC's Standard Plant Analysis Risk models (which use traditional PRA modeling). EPIX was also developed, among other objectives, to support PRA modeling. Traditional PRAs do not model instrumentation and control systems in detail, and therefore, the information in RADS can only be used to estimate initiating event (IE) frequencies and the contributions of I&C components to IE frequencies, and is not adequate to support failure parameter estimation of digital components.

For this study, a search was conducted of RADS to identify failure records of digital components in instrumentation and control systems after December 31, 1999. It was performed by identifying failure records of I&C components using component type "ICNTRL," and reviewing the failure descriptions. Only 18 failure records of digital I&C component failures were identified in this review of failure records of RADS. In most of these cases, failure was traced to a circuit board which was replaced. Approximately 2/3 of the failures are associated with valve controls in feedwater and electro-hydraulic control systems. Two of the failures involved a failure of a redundant processor which was alarmed, representing cases in which redundancy helped prevent system level failures. Only 7 failure records identified specific component failures (i.e., 2 connection failures, 2 power supply failures, 1 resistor failure, 1 relay failure, and 1 processor failure). Often, if a specific component failure was not identified by the plant, the failed board was sent to the vendor for more detailed failure analysis which is not included in EPIX.

In some cases, previous similar failure experiences were discussed. However, the information is not adequate for estimating failure rates of the circuit boards or the specific failed components. For a processor module of an IA Series controller made by Foxboro and used in the feedwater control system at a boiling water reactor plant, EPIX recorded plant specific experience of 6 failures in 40 years. The estimated failure rate is 6 failures divided by 40 years to give 0.15 per year. The processor failure at this plant is the only digital failure event where RADS (EPIX) contains the in-service duration time for the failed processor.

In general, RADS (EPIX) does not contain adequate information on how long the failed component had operated, and how many of the same components are operating at the specific plant and other plants without any failures. This additional information would be required in order to use the failure records to estimate component level failure rates. The required component level failure information/data may be obtainable from manufacturers.

8.6 Summary of Failure Data Review

8.6.1 Categorization of Data Collection Levels

From the above discussion and data sources reviewed, the level of detail at which the failure data are collected can be categorized into:

Level I: This is defined as the highest level at which a digital system is studied. At this level, the whole digital system itself is treated as an entity. The determination of the level depends on the definition of the boundary of the digital system. Sometimes it depends on the purpose of the study.

Level II: Very often a digital system may have several independent channels, especially for a safety related system, e.g., a digital RPS might have four independent channels that perform the same task. Sometimes, the reliability data of a single channel of the digital system are available. If the digital system has only one channel, then the data at Level II are equivalent to the data at Level I.

Level III: The type of functions that a digital system or a single channel of a digital system possesses can be divided into input, output, and processing. If there is more than one processor in a digital system or a single channel of the digital system, each processor again has input, output, and processing. Some systems might require several circuit boards to perform each of these functions. Sometimes, the data of failure to perform input, output, or processing are collected. The failure data of input, output, or processing can be obtained by lumping multiple circuit boards or cards that perform the corresponding functions.

Level IV: All digital systems are implemented using individual circuit board or card and bus connections. Failure data can be collected for each of the circuit boards or cards although they might have very different purposes. If each of the input, output, and processing functions is implemented using only one circuit board, then the failure data at Level IV is equivalent to the data at Level III.

Level V: A circuit board or card consists of a number of components that are physically wired. Failure data can also be collected for individual I&C circuits/chips on a circuit board or card.

In this classification, Level I means that the data are collected at the highest level for a digital system and Level V means the lowest level. However, the boundaries between these levels sometimes are not defined rigorously.

It should be pointed out that in Chapter 5 the failure modes and effects analyses (FMEAs) were performed at different levels where the top-level, module level and major-component-of-module level of FMEAs are actually Level I, III, and V in the proposed categorization scheme, respectively. In order to quantify the reliability model developed at a specific level, the failure data at the same level or a lower level have to be used to calculate the failure parameters.

8.6.2 Summary of Reliability Parameters of Microprocessor or Microprocessor-Related Systems

To illustrate the variability in failure rate of digital components in different applications, as obtained from analyses in different databases, failure rates of an Intel 80486 processor are estimated using different databases. Failure parameters from Telcordia, Military Handbook 217F, and PRISM, along with databases in Sections 8.4 and 8.5 are presented in Table 8-15.

Table 8-15 Data collection level and failure parameters for a microprocessor or a microprocessor-related system from different sources.

Data sources		Failure rates					Comments
		Level I	Level II	Level III	Level IV	Level V	
Section 8.3.1: Military Handbook 217F	Intel 80486 Processor					1.0×10^{-08} per hour	
Section 8.3.2: Telcordia SR-332	Intel 80486 Processor					1.8×10^{-08} per hour	
Section 8.3.3: PRISM Database	Intel 80486 Processor					6.3×10^{-09} per hour	
	Micro-processor					3.3×10^{-08} per hour	
	Micro-controller					5.5×10^{-08} per hour	
Section 8.3.4: IEC Standard 500	Regulators or controllers					N/A	It is unknown whether regulators or controllers are analog or digital from IEEE Standard-500, 1984
Section 8.3.5: IEC Standard 61508							N/A
Section 8.3.6: PDS Method	Table 8-1: Standard industrial PLC: single system	3.0×10^{-05} per hour					Single system only (PDS Data Handbook). According to PDS method, a safety system is generally defined to include input/output cards, a CPU that includes memory and watchdog, controllers (internal bus, communication, etc), system busy, and power supply. Therefore, a single system here is actually a complete digital system.
	Table 8-1: Programmable safety system: single system	2.0×10^{-05} per hour					
	Table 8-1: Hardwired safety system: single system	2.0×10^{-06} per hour					

Table 8-15 Data collection level and failure parameters for a microprocessor or a microprocessor-related system from different sources.

Data sources		Failure rates					Comments
		Level I	Level II	Level III	Level IV	Level V	
Section 8.4.1: Digital Core Protection Calculator of CE RPS	Table 8-2: A single channel of digital core protection calculator of CE RPS		0.0018 per demand				The Digital CPCS consists of four channels. The failure data are collected from each channel. (1984-1998: LERs, NPRDS, SCSS, PI, & NUREG/CR-5750).
Section 8.4.2: Eagle-21 of Westinghouse RPS	Table 8-3: A single Eagle-21 channel processor of Westing-house RPS		1.1×10^{05} per hour				The Digital RPS consists of four channels (1984-1995: LERs, NPRDS, and SSCS)
Section 8.4.3: Digital Core Protection Calculator of CE RPS	Table 8-4: A single computer board of digital CPCS				2.2×10^{-06} per hour		The Digital CPCS consists of four channels but the data are collected only for a single circuit board (1984-2006: LERs)
Section 8.4.4: PLC used in emergency shutdown system of natural gas industry	Table 8-5: A PLC based digital system that is part of the emergency shutdown system (ESD)	8.2×10^{-06} per hour					The ESD system does not have redundant PLCs (Up to 1991: Natural Gas Compressor Stations). The PLC studied here typically consists of three input boards and two output boards and is thus, a complete digital system contained in the ESD.
Section 8.4.5: Fault tolerant digital control systems	Table 8-6: Fault-tolerant digital control systems	2.6×10^{-05} per hour					Failure data of both the system and a single channel are collected (Wright's survey, Gas Turbine Control System, U.S. and Canadian Nuclear Power Plants)
	Table 8-6: A single channel of fault-tolerant digital control systems		9.2×10^{-05} per hour				

Table 8-15 Data collection level and failure parameters for a microprocessor or a microprocessor-related system from different sources.

Data sources		Failure rates					Comments
		Level I	Level II	Level III	Level IV	Level V	
	Table 8-7: Processor (CPU/ Memory)					5.1×10^{-05} per hour	The data for CPU/memory are from OREDA
	Table 8-7: Processor (CPU/ Memory)					2.1×10^{-04} per hour	The data for CPU/memory may be collected from [Humphreys & Daniels]
	Table 8-8: Processor (CPU/ Memory)					3.8×10^{-05} per hour	The data for CPU/memory are predicated data using the Military Handbook for a TRICONEX TMR digital system
	Table 8-8: Processor (CPU/ Memory)					1.5×10^{-04} per hour	The data for CPU/memory are predicated data using the Military Handbook for a digital system in [Humphreys & Daniels]
Section 8.4.6: PLC used in some chemical and nuclear plants	Table 8-9: PLC of digital systems for control and shutdown			9.4×10^{-06} per hour			All PLCs have redundancy (primary and secondary processor). (U.S. Phenol Plant)
	Table 8-10: PLC of control interlock systems			1.3×10^{-06} per hour			All PLCs have redundancy with two processors (French-designed Nuclear Power Plant)
	Table 8-11: PLC of distributed control systems			2.9×10^{-06} per hour			It is not clear whether these PLCs have redundancy but it seems they do (Canadian Nuclear Power Plant)
Section 8.4.7: Savannah River Site generic data development	Table 8-12: Individual PLCs			3.2×10^{-05} per hour			Data were collected from: Nuclear Power Plants, OREDA, Chemical Processing Plants, Tritium Handling Facility at LANL, Liquid Natural Gas Facilities, and Other Industries

Table 8-15 Data collection level and failure parameters for a microprocessor or a microprocessor-related system from different sources.

Data sources		Failure rates					Comments
		Level I	Level II	Level III	Level IV	Level V	
Section 8.4.8: Digital systems of ESBWR	Digital Trip Module(DTM) and Trip Logic Unit (TLU)		5.0×10^{-06} per hour				The Digital RPS consists of four channels and each of them has microprocessor-based DTM and TLU. It is expected that each module contains multiple circuit boards. The failure data are collected for each DTM and TLU (GE data according to [GE 2006])
Section 8.4.9: Digital systems of Korean standard nuclear power plants	Table 8-13: Digital Engineered Safety Feature Actuation Systems			3.2×10^{-06} per hour			A processor module failure s considered. It is likely that a processor module contains at least one circuit board (Westinghouse data)
Section 8.4.10 and 11: Digital systems of AP600 and AP1000 reactors	Circuit boards of digital systems in AP600 and AP1000				5.0×10^{-06} per hour		Failure rate of each individual circuit board is considered the same [Westinghouse 2004]
Section 8.4.12: DFWCS reliability performed by OSU	Intel 586 CPU					3.75×10^{-05} per hour	It is very likely that the failure data were collected at Level V from nuclear power plants and RAC PRISM database.
	8051 CPU of 53MC5000 Controller					1.0×10^{-06} per hour	
Section 8.5.1: Licensee Event Reports	Micro-processor					N/A	Failure rates can not be estimated due to unknown operating time periods from LER.
Section 8.5.2: Equipment performance and information exchange database (EPIX)	A single circuit board				1.7×10^{-05} per hour		

The specification of the Intel 80486 processor is as follows: 32-bit, MOS technology, quality level comparable to military or space mission grades, case temperature of 35℃, operating temperature of 30℃, thermal resistance of 1.5 C/W, rated voltage of 5.0V, active thermal design maximum current of 1145mA, junction temperature of 44℃, PGA non-Hermetic packing with 180 pins, ground based with well-controlled environment, and more than two years in production.

All CCF data and/or parameters shown in the above sources are presented in Table 8-16. CCF data are not available in many analyses; this table includes data for analyses that provided CCF parameters. If a model has been used for the CCF calculation, the types of models and parameters are also specified in Table 8-16, if available. Some CCF data do not have values of lower and upper (e.g., 5% and 95%) bounds.

8.7 Generic Failure Rate Estimate Using a Hierarchical Bayesian Method

The review of digital system failure data indicates that there is a need to combine data from different sources to estimate reliability parameters. Failure data of a certain type of component can be collected from a wide group of sources that may involve different manufacturers, designs, quality levels, and environment. The estimated reliability parameters from the collected data should provide the probability distribution of the failure rate representing the population variability. The population variability captures the variation among different data sources and can be considered to provide generic reliability parameters of the specified component. In general, application specific data should be collected and used in a Bayesian analysis using the population variability curves as prior distributions to obtain failure distributions that are applicable to the specific components.

This section discusses how to estimate the generic failure parameter of a specific component using a Hierarchical Bayesian Method (HBM) by categorizing and grouping raw data collected from different sources. As indicated in Chapter 5, the Markov model of the DFWCS will be developed at a level of major-component-of-a-module. Because this level of component data can be found in the PRISM database, these data are extracted and used in the HBM analysis presented.

Application of the HBM method in this study is complicated by lack of information about the raw data and obtained population variability curves with very large uncertainties. The curves are to be used in the benchmark studies to exercise the models and should not be used in quantifying models developed to support decision-making.

Table 8-16 CCF data from different sources.

Data sources		Failure rates (5% & 95%) or failure probability of CCF	Description of CCF	Comments
Section 8.4.1: Digital Core Protection Calculator of CE RPS	Table 8-2: Digital CPCS	$1.4x10^{-04}$ per hour ($2.3x10^{-05}$, $3.8x10^{-04}$)	CCF of 2 out of 3 digital CPCS channels	1. Alpha vectors for CE3-CPD-CF-T2OF3TM, CE3-CPD-CF-T3OF4-TM, CE4-CPD-CF-T2OF3-TM, and CE4-CPD-CF-T3OF4TM are all [$9.6x10^{-01}$ $2.9x10^{-02}$ $1.5x10^{-03}$ $8.1x10^{-04}$]. 2. Digital CPCS consists of four channels; The Alpha factor model is used.(1984-1998: LERs, NPRDS, SCSS, PI, and NUREG/CR-5750)
		$5.7x10^{-05}$ per hour ($6.3x10^{-06}$, $1.8x10^{-04}$)	CCF of 3 out of 4 digital CPCS channels	
Section 8.4.2: Eagle-21 of Westinghouse RPS	Table 8-3: Eagle-21 Digital RPS	$5.1x10^{-07}$ per hour ($3.0x10^{-08}$, $1.8x10^{-06}$)	CCF of 2 out of 3 Eagle-21 channels	1. The Alpha vectors of basic events WES-C21-CF-2OF3 and WES-C21-CF-3OF4 are both [$9.4x10^{-01}$ $4.4x10^{-02}$ $1.1x10^{-02}$ $3.3x10^{-03}$]. 2. Digital RPS consists of four channels;The Alpha factor model is used. (Data from LERs, NPRDS, and SSCS)
		$1.5x10^{-07}$ per hour ($3.6x10^{-09}$, $5.8x10^{-07}$)	CCF of 3 out of 4 Eagle-21 channels	
Section 8.4.3: Digital Core Protection Calculator of CE RPS	Table 8-4: Digital CPCS	$5.9x10^{-06}$ per hour ($2.9x10^{-06}$, $9.8x10^{-06}$)	Inaccurate cross calibration of ex-core neutron data sets (cross channel, COLSS, etc)	The Digital CPCS consists of four channels; Other CCFs are neglected but can be found in Table 8.4; Experience data from LERs from 1984 to 2006.
		$2.8x10^{-06}$ per hour ($8.5x10^{-07}$, $5.5x10^{-06}$)	Computer technicians input wrong data sets to all 4 channels	
		$2.0x10^{-06}$ per hour ($4.5x10^{-07}$, $4.3x10^{-06}$)	Reactor vendor supplies erroneous data sets or software updates input to all channels	
		$1.2x10^{-06}$ per hour ($1.4x10^{-07}$, $3.1x10^{-06}$)	Reactor vendor supplies software update containing latent software design error input to all 4 channels	

Table 8-16 CCF data from different sources.

Data sources		Failure rates (5% & 95%) or failure probability of CCF	Description of CCF	Comments
Section 8.4.5: Fault tolerant digital control systems	Table 8-7: Fault-tolerant Digital Control Systems	6.6×10^{-07} per hour	Hardware CCF	Experience data: Wright's survey, Gas Turbine Control System, U.S. and Canadian Nuclear Power Plants.
		5.9×10^{-06} per hour	Software CCF	
Section 8.4.8: Digital systems of ESBWR	Total 4 DTMs	1.2×10^{-05} (failure probability)	C-74-DTM-CF-ALL (CCF 3/4)	1. The failure data were collected for each DTM and TLU; 2. The Alpha factor model is used. ([GE 2006]). CCF factors for C-74-DTM-CF-ALL (CCF 3/4) and C74-SLU-CF-TLU (CCF 3/4) are both 3.0×10^{-03}.
	Total 4 TLUs	2.7×10^{-06} (failure probability)	C74-SLU-CF-TLU (CCF 3/4)	
	Each of channels of digital RPS of ESBWR has microprocessor-based DTM and TLU	1×10^{-05} (failure probability)	Software CCF	
Section 8.4.9: Digital systems of Korean standard nuclear power plants	Digital Protection System of Korean Nuclear Power Plants	Unavailable	CCF of electronic components	The Beta model is used and Beta=0.03. (Westinghouse)
		Unavailable	CCF of software	
Section 8.4.10: Digital systems of AP600 reactors	AP600 Digital systems	$1.0 \times 10^{-05} \sim 1 \times 10^{-06}$ (failure probability)	CCF of software	From AP600 PRA
Section 8.4.11: Digital systems of AP1000 reactors	AP1000 Digital Systems	1.1×10^{-05} failures per demand	CCF of any particular software module	From AP1000 PRA
		1.2×10^{-06} failures per demand	Software CCF among all boards	

8.7.1 Hierarchical Bayesian Models for Failure Rate Determination

In a simple Bayesian analysis, Bayes' Theorem is applied to obtain a posterior distribution by updating the prior distribution. Often, it is assumed that the data were collected from a single source. However, sometimes the data was collected from different sources, as is the case of the data of PRISM. The assumption of a single data source leads to a narrow posterior distribution because the source-to-source variation (population variation) is ignored. The two-stage Bayesian method [Kaplan 1984] has been used to take into consideration the source–to-source variability [Siu 1998, Porn 1996, and Bunea 2005]. In this study, this variability is addressed by using the HBM [Atwood 2003] which is a more general approach. In the HBM, the prior distribution is developed in multiple stages of a hierarchical structure, i.e., the parameters of the prior distribution are also considered uncertain and can be modeled as a probability distribution function with, again, uncertain parameters. This process can be repeated until the last stage, where the prior distribution is called hyperprior with corresponding constant hyperparameters. It can be demonstrated that an HBM model with two stages is the same as the two-stage Bayesian model. In two-stage analysis [Kaplan 1984], a discretized probability distribution method is often used to solve the Bayesian equations. The HBM provides its own way of solving the model [Atwood 2003].

The population variation curve (PVC) is denoted as $g(\lambda)$, where λ is the parameter of interest, e.g., the failure rate. Usually it is assumed to be lognormal or gamma distributed with $\underline{\theta}$ representing the parameter vector. Data is collected from m different sources/plants whose failure rates λ_i are random samples from $g(\lambda)$. Obviously, $\underline{\theta}$ might consist of different variables depending on the assumption of population variability distribution. Different prior distributions can be selected for each element of parameter vector $\underline{\theta}$, e.g., $\underline{\theta}=[\alpha,\beta]$ for a gamma distribution. The prior distributions of the parameters are called hyperprior distributions and denoted as $\pi_0(\underline{\theta})$.

The posterior distribution of the uncertain parameter vector $\underline{\theta}$, i.e., the hyperposterior distribution, is required for PVC and can be calculated by applying Bayes' Theorem in the multiple-dimensional form [Siu 1998]:

$$\pi_1(\underline{\theta}|E) = \frac{L(E|\underline{\theta})\pi_0(\underline{\theta})}{\int_0^\infty \cdots \int_0^\infty L(E|\underline{\theta})\pi_0(\underline{\theta})d\theta} \tag{8-1}$$

where $L(E|\underline{\theta})$ is the likelihood of the collected data. The likelihood function for a specific source/plant is given as $L(E_i|\underline{\theta}) = \int_0^\infty P(x_i|t_i,\lambda_i)g(\lambda_i|\underline{\theta})d\lambda_i$, where λ_i is the failure rate of source/plant i, x_i is the number of failures that took place in time period t_i, and $P(x_i|t_i,\lambda_i) =$

$\frac{(\lambda_i t_i)^{x_i} e^{-\lambda_i t_i}}{x_i!}$. The likelihood function for the entire set of the evidence is the product of likelihood functions for the individual sources

$$L(E|\underline{\theta}) = \prod_{i=1}^{m} L(E_i|\underline{\theta}) = \int \cdots \int \prod_{i=1}^{m} P(x_i|t_i, \lambda_i) g(\lambda_i|\underline{\theta}) d\lambda_1 \cdots d\lambda_m \tag{8-2}$$

The expected PVC can be calculated using the hyperposterior distribution of $\underline{\theta}$:

$$\begin{aligned} g(\lambda|E) &= \int_0^\infty \cdots \int_0^\infty g(\lambda, \underline{\theta}|E) d\underline{\theta} \\ &= \int_0^\infty \cdots \int_0^\infty g(\lambda|\underline{\theta}) \pi_1(\underline{\theta}|E) d\underline{\theta} \\ &= \int_0^\infty \cdots \int_0^\infty g(\lambda|\underline{\theta}) \frac{\prod_{i=1}^{m} L(E_i|\underline{\theta}) \pi_0(\theta)}{\int_0^\infty \cdots \int_0^\infty \prod_{i=1}^{m} L(E_i|\underline{\theta}) \pi_0(\underline{\theta}) d\underline{\theta}} d\underline{\theta} \end{aligned} \tag{8-3}$$

which can be used as a generic informative prior distribution for a Bayesian analysis of the data collected for the same component from a specific source/plant. Due to the unclear identification of data sources of the PRISM, it is not likely that the specific source/plant can be associated with the data sources of the PRISM.

Usually, it is impossible to evaluate the equation (8-3) analytically. The solution using Markov Chain Monte Carlo (MCMC) simulation considers posterior distribution of all the parameters of interest, i.e., $\underline{\theta}$ and λ_i's, and generates samples from a joint posterior distribution by constructing a Markov chain that has the parameters of interest as its state space and taking samples from the conditional distributions of the parameters. More specifically, Gibbs sampling or the Metreopolis-Hastings algorithm can be used in MCMC implementation [Atwood 2003].

8.7.2 Failure Rate Estimates of Digital Components Using the HBM

The RAC database denoted as RACdata in PRISM contains failure data records in the form of the number of failures in a number of operating/calendar hours. The RAC database sources are not completely specified and only identified in a format such as "warranty repair data from a manufacturer." In addition, little information is available on how the raw data of PRISM was collected, e.g., no information is provided on the boundary defining a component, how failure is defined, or how failed components were identified. The failure records of a specific type of component, e.g., memory, are further categorized according to sub-level component types, e.g., random access memory (RAM) or programmable read only memory (PROM); quality, e.g., commercial-grade or military-grade; environment, e.g., ground or airborne; hermeticity, e.g., plastic or ceramic; and time period within which the data are collected, etc. In this study, the failure data of various digital components were extracted from the RAC database. It was decided, for each sub-level component, to group the failure records of different qualities, environments, hermeticities, and time periods.

Table 8-17 lists the grouped failure records of an example digital component. The definitions of the quality and environment can be found in [DOD 1995], [RAC 1998], and PRISM manual [RAC PRISM]. The failure records were used in estimating the population variability curve of this type of digital component. The last column of the table lists the point estimates of failure rates of those failure records with at least one failure. The point estimate is simply the number of failures divided by the number of hours. It provides information on the possible range of the population variability curve. The point estimate information was used in estimation of hyperprior parameters. The wide variation in the point estimates in the table leads to a very wide population variability curve. It stems from the decision to include as many data records as available. This wide variation indicates that these data may not be appropriate for quantifying reliability models that are to be used in support of decision-making. As mentioned earlier, these data are only used for illustrating the methods proposed in this report.

Some of the symbols of environment (Column 2 in Table 8-17) are described below:

AIF: Airborne, Inhabited, Fighter. Same as Airborne, Inhabited, Cargo (AIC) but installed on high performance aircraft such as fighters and interceptors.

GB: Ground Benign. Non-mobile, temperature and humidity controlled environments readily accessible to maintenance; includes laboratory instruments and test equipment, medical electronic and test equipment, medical electronic equipment in ground silos.

GF: Ground, Fixed. Moderately controlled environments such as installation in permanent racks with adequate cooling air and possible installation in unheated buildings; includes permanent installation of air traffic control radar and communications facilities.

GM: Ground, Mobile. Equipment installed on wheeled or tracked vehicles and equipment manually transported; includes tactical missile ground support equipment, mobile communication equipment, tactical fire direction systems, handheld communications equipment, laser designations and range finders.

NS: Naval, Sheltered. Includes sheltered or below deck conditions on surface ships and equipment installed in submarines.

A Chi-square (χ^2) test was performed for the data of each sub-level component type to determine whether the failure records can be pooled. If they cannot be pooled, the population variability should be used to model the failure rates of the components. A Chi-square test was performed on the digital component failure data in Table 8-17 and a χ^2 value of 14481 was obtained. This indicates that for the data, the confidence is high that the failure records can not be pooled, i.e., the failure records are samples from different failure rate sources and a population variability distribution should be used to model the variability in the failure rates.

Table 8-17 Failure records of a digital component extracted from PRISM RACdata database.

Quality	Environment	Number of failures	Number of hours (*1.0x10^{06})	Data source reference	Point estimate failure rate (per million hours)
Commercial	GB	12	633.8929	13567	1.9x10^{-02}
Unknown	GB	0	0.2600	13567	
Unknown	GB	0	0.0625	18216	
Commercial	GB	16	2597.365	13567	6.2x10^{-03}
Commercial	GM	4	701.1615		5.7x10^{-03}
Commercial	N/R	2	509.1335		3.9x10^{-03}
Commercial	GB	28	22751.18	13567	1.2x10^{-03}
Commercial	GB	0	1105.13	13597	
Unknown	GB	80	444.0000	15293	1.8x10^{-01}
Unknown	GB	44	307.8874	17941	1.4x10^{-01}
Unknown	GB	0	6.5937	18216	
Commercial	GB	0	19.3613	13567	
Commercial	GB	188	20069.9345	13567	9.4x10^{-03}
Commercial	GM	1	692.6390		1.4x10^{-03}
Military	N/R	1	149.2384		6.7x10^{-03}
Military	AIF	0	0.0253	17867	
Military	AIF	0	1.8755	18138	
Military	AIF	0	11.3706	18139	
Military	GB	0	0.7367	13567	
Military	GF	0	53.6832	17191	
Military	NS	0	29.2752	12449	
Unknown	AIU	0	0.2376	12569	
Unknown	AUF	0	1.5206	12569	
Unknown	AUT	0	1.3585	15312	
Unknown	GB	0	90.4280	13567	
Unknown	GB	0	1.8878	13569	
Unknown	GB	54	205.2583	17941	2.6x10^{-01}
Unknown	GB	2	1.4060	18216	1.42
Unknown	GF	0	2.0275	13999	
Unknown	GF	2	553.6315	14434	3.6x10^{-03}
Unknown	GF	332	590.3949	17191	5.6x10^{-01}
Unknown	GF	0	0.0080	17571	
Unknown	GF	0	2.1948	17604	
Unknown	NS	0	2.0799	15280	
Unknown	NSB	0	0.0121	16562	

In the hierarchical Bayesian analysis, different distribution types can be assumed for the failure rates and the hyperpriors. The parameters for hyperpriors were chosen based on the range of the point estimates of failure rates in the data set and properties of the type of the distribution. The criterion used here for selecting the mean values of the prior parameters is that the maximum and minimum values of the point estimate failure rates lie within the 95th and 5th percentile of the distribution defined by the selected mean values. A software tool, WinBUGS [Spiegelhalter 2003], was used to create the model and to calculate the population variability distributions. The details of WinBUGS can be found in WinBUGS and are not discussed here.

There are no general rules about how to select the types of the priors and hyperpriors. Sensitivity calculations performed by comparing different distributions were documented. In the base case calculation described here, it was assumed that the population variability distribution is lognormally distributed with parameters μ and σ. The last column of Table 8-17 shows that the point estimate failure rates are approximately in the range of $1.0\text{x}10^{-03}$ and 1.4. Assuming that the range of $1.0\text{x}10^{-03}$ and 2.0 is the 90% confidence interval of a lognormal distribution, the mean values of μ and σ can be calculated using formula:

$$\bar{\sigma} = \frac{\ln(b/a)}{3.29}, \bar{\mu} = \ln b - 1.645\sigma \qquad (8\text{-}4)$$

where a and b are the lower bound and the upper bound of the point estimate, respectively, i.e., a = $1\text{x}10^{-03}$ and b = 2.0. Therefore, $\bar{\sigma}$=2.31 and $\bar{\mu}$=−3.1073. According to the HBM, the parameters μ and σ are also associated with uncertainties. In the absence of any information concerning parameters μ and σ, the uncertainties can be addressed by further assuming that μ and σ are uniformly distributed with lower and upper bounds equal to -7 and -0.1, and 1 and 3.5, respectively.

A WinBUGS analysis of the data of the digital component in Table 8-17 resulting in posterior distributions of μ and σ that are within the bounds of the uniform hyperprior distributions are reasonable. WinBUGS does not directly produce an output of the population variability distributions and only provides the characteristics of the posterior distributions of μ and σ separately. It may not be accurate to simply use the calculated mean values of μ and σ to define a population variability distribution because μ and σ are correlated. Instead, a trick was used to generate information on the population variability distribution, by adding an artificial failure record with no failures and very small operating hours in the data set. Such a failure record is not expected to introduce any significant bias in the results of WinBUGS, and its posterior distribution is effectively the population variability distribution. The estimated population variability distribution has a mean value of 0.33, and 95% confidence interval of $8.8\text{x}10^{-05}$ and 0.51. This is the base case shown in Table 8-18 wherein the error factor is calculated as the square root of the ratio of the 95th and 5th percentiles.

Table 8-18 Characteristics of population variability distribution of a digital component data.

Case	Mean	5th	Median	95th	Error factor
Base	0.33	8.8×10^{-5}	7.2×10^{-03}	0.51	76
LNL1-100,000 samples	0.30	9.5×10^{-5}	7.5×10^{-03}	0.46	69
LNL2-300,000 samples	0.34	9.5×10^{-5}	7.4×10^{-03}	0.44	68
LNG	0.32	8.9×10^{-05}	7.8×10^{-03}	0.47	73
LUG	0.31	2.1×10^{-04}	1.1×10^{-02}	0.53	50
GEG-100,000 samples	0.09	3.4×10^{-7}	1.3×10^{-02}	0.51	1100
GEL	0.11	1.1×10^{-7}	1.3×10^{-02}	0.52	2100
GUU	0.15	2.0×10^{-8}	1.3×10^{-02}	0.77	2000

8.7.3 Sensitivity Analysis

A few sensitivity calculations were performed using different distribution types and different hyperprior distributions. The results of the sensitivity calculations are shown in Table 8-18 using different models described below, where L represents lognormal distribution, N the normal distribution, G the gamma distribution, E the exponential distribution, and U the uniform distribution. The first capital letter indicates the type of the distribution of the population variability curve, and the second and the last letter indicate the distributions of its parameters.

LNL1: In this sensitivity calculation, the failure rate is assumed to be lognormally distributed with its parameters μ and σ distributed normally, and lognormally, respectively, that is, $\mu \sim Normal\ (\mu_\mu, \sigma_\mu)$, and $\sigma \sim Lognormal\ (\mu_\sigma, \sigma_\sigma)$. The prior mean values of μ and σ were calculated such that the lognormal distribution based on the mean values has a confidence interval of 1×10^{-03} and 2.0. Using equation (8-4) we again have $\bar{\mu} = -2.1073$ and $\bar{\sigma} = 2.31$, i.e., $\bar{\mu} = \mu_\mu - 2.1073$, and $\bar{\sigma} = \sigma_\mu = 2.31$. The standard deviation of μ, i.e., σ_μ, was selected to be 15. According to $\bar{\sigma} = \exp(\mu_\sigma + \frac{\sigma_\sigma^2}{2})$, the parameters of σ, μ_σ and σ_σ, were selected to be -3.66 and 3, respectively. The confidence intervals of the posterior distributions of μ and σ are well within the confidence intervals of the hyperpriors. The characteristics of the population variability distribution are very similar to those obtained with uniform hyperprior distributions. The mean values of the LNL1 model changed significantly with different sample size in WinBUGS. The mean value is 0.374 for 10,000 samples, 0.3025 for 100,000 samples, and 0.2946 for 1,000,000 samples. Thus, the calculation converged for 100,000 samples. Note that

WinBUGS does not have a tool to tell whether a convergence has been achieved, and the only way to assure the convergence is to compare the results using different sample sizes.

LNL2: This sensitivity calculation is the same as LNL1 except that the prior distribution of σ was changed to a narrower distribution (smaller variance) which still covers the confidence interval of the posterior distribution of σ obtained in LNL1. The resulting confidence interval of σ is practically the same as that obtained from LNL1. The characteristics of the population variability distribution are close to those of the previous cases, except the mean value which deviates from that of the base case by a larger factor. Sensitivity calculations were performed using this model by changing the number of samples. The results show that the mean value varies significantly, i.e., from 0.49 with 10,000 samples, to 0.22 with 100,000 samples (0.4 with 1,000,000 samples), while other characteristics do not change much. It is easy to conclude that it has not converged. Using more samples is necessary for the simulation to converge. The mean value becomes 0.3361 for 3,000,000 samples and 0.3384 for more than 4,000,000 samples. It is shown in Table 8-18 that the mean value, median, and 5% and 95% percentiles of the population variability are very close to each other using LNL1 and LNL2 models once the convergence is achieved.

LNG: This sensitivity calculation is the same as LNL1 except that σ is assumed to be gamma distributed with a mean equal to 2.31. The two parameters μ_σ and σ_σ are assumed to be 2.31 and 1, respectively. The resulting population variability distribution is close to those of other cases.

LUG: It is assumed that the failure rate is lognormally distributed with parameters μ and σ. The parameters μ and σ are uniformly and gamma distributed, respectively. That is, $\mu \sim Unif\ (a_1,b_1)$ and $\sigma \sim Gamma\ (\alpha,\beta)$. The prior mean values of μ and σ are selected as -3.1073 and 2.3103 such that the lognormal distribution based on the mean values has a confidence interval of $1\text{x}10^{-03}$ and 2.0. We choose $a_1 = -7$ and $b_1 = -0.1$. The standard deviation of μ, i.e., σ_μ, is selected to be 15. The parameters of σ are $\alpha = 0.023103$ and $\beta = 0.01$. The calculation results are also close to previous results, as shown in Table 8-18.

GEG: This sensitivity calculation assumes that the failure rate is gamma distributed. The mean values of its hyperpriors are selected as 0.44 and 0.87, such that the prior distribution of the failure rate has a confidence interval approximately between $1.0\text{x}10^{-03}$ and 2 (between 0.001 and 2.0337). The parameter α is assumed to be exponentially distributed with mean of 0.44. The parameter β is assumed to be gamma distributed with parameters α_β and β_β equal to 0.01 and 0.0115 (such that the mean of β is 0.01/0.0115=0.87), respectively. With this choice of hyperpriors, the posterior distributions of the hyper parameters are covered by the hyperpriors. However, the population variability distribution is significantly different from the previous models.

GEL: It is assumed that the failure rate is gamma distributed with parameters of α and β, which are of exponential and lognormal distribution, respectively. The mean values of its hyperpriors are still 0.44 and 0.87. Lognormal distribution parameters are -2.1393 and 0.25 for β such that the mean of β is around 0.87. The mean value of the PVC is only slightly different from that of the GEG model but significantly different from those of other models.

GUU: It is again assumed that the failure rate is of gamma distribution with parameters of α and β. The parameters of α and β are both uniformly distributed, that is, $\alpha \sim Unif(a_1,b_1)$ and $\beta \sim Unif(a_2,b_2)$. The values $a_1 = 0.1$, $b_1 = 1$, $a_2 = 0.1$, and $b_2 = 2$ are chosen. The calculation shows that the mean value of PVC is slightly larger than those of the GEG and GEL models but much smaller than those of other models.

An inspection of Table 8-18 shows that the mean values of the failure rate of gamma distributions, i.e., using models GEG, GEL, and GUU, are close to each other and smaller than the mean values calculated using other models. It is suspected that the results are adversely affected by the issue that the likelihood function obtained assuming a gamma distribution is unbounded [Hofer 1997]. In addition, experience in performing the HBM analysis indicates that uniformly distributed hyperpriors tend to produce consistent results. Therefore, it is decided that lognormally distributed failure rates and uniformly distributed hyperpriors be used in the HBM analysis based on the sensitivity analysis.

8.7.4 HBM Analysis of Other Digital Components

Other data of digital components which were also extracted from PRISM and the HBM are applied to them to estimate the failure rate of each type of component. Table 8-19 lists the components that are analyzed using HBM. In order to prevent inappropriate use of the data, the details of the results are not listed (only the calculated error factors are provided in the table). The failure rates were assumed to be lognormally distributed and the hyperpriors were assumed to be uniformly distributed, as suggested by the sensitivity analysis. The upper and lower bounds of the hyperpriors were selected such that the resulting posterior distributions of the hyperparameters are covered by the hyperpriors. Sensitivity studies show that uniform priors selected this way always produce reasonable results with 10,000 samples. Many failure records do not have any failures, and those failure records that do have failures have widely scattered point estimates of failure rates. Therefore, the results of the analysis depend on the choice of hyperprior distributions. The point estimates of individual failure records were used to estimate the approximate ranges of the population variability distributions, which provide information on how to select the parameters of the hyperprior distributions. The selected values of the parameters for the hyperpriors were verified to be wide enough to cover the confidence intervals of the posterior distributions of the hyperparameters.

The hyperpriors and their parameters were carefully selected, often by performing sensitivity calculations. However, the population variability curves obtained using the MCMC method are still very wide with large error factors, as indicated in Tables 8-18 and 8-19. This is due to the large variability in the different sources of data. In general, application specific data should be collected and used in a Bayesian analysis to further update the population variability curves to reduce uncertainty. In later tasks of this project, the population variability curves are only used to demonstrate the reliability methods and exercise the reliability models. Due to lack of information about the sources of the raw data and the method used in collecting them, the correctness of the raw data collection was not validated. It is questionable whether the data is applicable to the components of the DFWCS. The applicability of the data is an important issue that directly contributed to the large uncertainties shown in the table. Because of the lack of validation and the large uncertainties, the data as obtained in this study using HBM are not appropriate for quantifying models that are to be used in support of decision-making.

Table 8-19 Error factors based on a Hierarchical Bayes Analysis.

Component	Error factor
Buffer	88
Control	142
Counter/Divider	147
Decoder	16
Encoder	170
EPROM	23
Error Detection/Correction	173
Gate	21
Latch	4.7
Line bus driver	55
Line bus receiver	10
Linear amplifier	4.8
Linear comparator	26.8
Linear converter	15
Linear multiplexer	12.3
Linear operational amplifier	43.5
Linear timer	9.1
Linear voltage regulator	8.8
Micro controller	50
Microprocessor	16
Multiplexer	25
Optoisolator	8.7
Processing unit	339
PROM	5.3
RAM	76
Receiver/Transmitter	21
Register	22
ROM	14
Transceiver	11
UVEPROM	16

9. MODELING TO ADDRESS DESIRABLE CHARACERISTICS

Chapters 6 and 7 describe the approaches for using Markov and event tree/fault tree (ET/FT) methods to model the digital feedwater control system (DFWCS), respectively. In order to demonstrate how the reliability models developed using these two methods would address the desirable characteristics identified in Chapter 2, this chapter discusses what will be done in the next task of this project for each category of the characteristics. It also serves as a technical description of the scope of the benchmark studies.

Level of Detail of Modeling

The level of detail of the modeling of the DFWCS is demonstrated by the model of the Main central processing unit (CPU) module described in Section 6.1.1. The failure modes and failure rates of the major components of the Main CPU module, e.g., buffers, multiplexers, analog/digital converters, and digital/analog converters, can be explicitly considered in developing the reliability models of the module. Modeling at this level allows (1) the failure rates of the CPU module to be estimated using the available generic failure data and failure mode distributions of digital components; and (2) the control logic implemented in the software of the CPU to be considered in developing a reliability model, e.g., if a sensor fails to generate the input signal, and the CPU is operating normally, then the failure will be detected by the CPU and the input from the other sensor will be used instead.

In the next task, all modules/components of the DFWCS will be modeled in the same way.

Identification of Failure Modes of the Components of Digital Systems

The failure modes and effects analysis (FMEA) of the DFWCS performed so far needs to be further expanded to include the FMEA of the internal components of all modules of the DFWCS. A simulation tool will be developed that will systematically determine system's response to postulated failures. It is expected that this tool will help to identify system failures due to incorrect design requirements.

Regarding unique digital features of the system, the FMEA performed so far found that the communication and clock synchronization among the controllers do not affect the control function of the DFWCS. Neither voting nor synchronization takes place in the system. The fail-over operation that takes advantage of the CPU redundancy will be explicitly modeled.

A review of the operating experience of the system will be performed by reviewing the licensee event reports associated with the system. Relevant failure modes and effects identified in this way will be considered when developing the model.

Modeling of Software

Both normal and failure behaviors of software will be considered. Modeling of the normal behavior of the application software is included in construction of the Markov and FT models. The software failures that will be explicitly modeled include those of (1) the application software, and (2) the support software which includes vendor-developed platform software, and operating systems of the CPUs and controllers. The software failures will not be quantified; instead, place holders will be identified in the models. For example, the Markov model of the Main CPU module

described in Section 6.1.1 includes a contribution of software failures to the failure mode of undetectable failures, but the relevant parameters of these failures will not be quantified. Instead, a range of values will be used to demonstrate the sensitivity of the system reliability to software failures.

In general, it is desirable to include the software failure modes and failure rates in the calculations of the reliability models. However, this is beyond the scope of the benchmark study.

Modeling of Dependencies

The modeling of different types of dependencies of the DFWCS is described below.

- Modeling Dependencies due to Communication

The connections/interfaces between the modules/components of the DFWCS were described in Chapter 4. Two types of failures associated with these connections/interfaces were considered in the FMEA documented in Chapter 5 and Appendix B: (1) propagation of failures through them, and (2) loss of connections/communications. The propagation of failures was considered when failure modes were postulated in the FMEA and the failure effects were determined, and losses of the connections/communications were postulated as failure modes.

The information obtained in the FMEA will be used in the next task. Both types of failures will be considered. Propagation of failures will be considered when modeling the effects of any failure modes, and loss of connections/communication will be failure modes in the reliability model. The propagation of failure represents successful operation of the connection/communication, and will only be implicitly modeled. The following are two examples of this latter type of failure:

1. Between the two DFWCSs, the main feedwater valve (MFV) demand signals are interchanged and used in calculating feedwater pump demand. Loss of the signal or the connection can be modeled as a failure mode for the DFWCS being modeled.

2. The Main and Backup CPUs exchange different analog and digital signals. The failure modes involving failures of these signals were analyzed in Appendix B, and will be considered in developing the probabilistic models of the CPU modules.

- Modeling Support Systems

Internal power supplies will be modeled as a part of the component. For example, in the Markov model, 120V AC buses can be modeled using a simple model with two states. Dependencies on heating, ventilation and air conditioning (HVAC) probably can be neglected by using the initiating event frequency for loss of control room cooling and the probability of not recovering HVAC from a probabilistic risk assessment (PRA) of the plant.

- Sharing of Hardware

No sharing of hardware of the DFWCS with other systems was identified, except for sensors. Sensors are probably shared with other systems, e.g., reactor protection system (RPS) and engineered safety features actuation system (ESFAS). Such sharing represents a dependency between an initiating event and the systems that are needed to mitigate the initiating event. For

the FT method, standard fault tree linking can account for the sharing. In general, to properly account for the sharing using a Markov model, a joint Markov model of the systems would be needed, adding complexity to what may be already too complicated a model. Alternative approaches will have to be developed, e.g., transforming the results of the Markov models of the two systems that shared hardware into a cutsets format and combining the cutsets of the systems. The modeling of the dependency requires that a model of the RPS and ESFAS be available.

The DFWCS only has limited redundancy, e.g., having two CPUs. Controllers' selection of redundant CPU signals can be considered performing a voting function on the output of redundant channels. The controllers themselves do not have redundancy and are single failures of the system. The dependency on the controllers will be explicitly modeled.

- Modeling of Fault-Tolerant Features

The following fault-tolerant features will be modeled explicitly: (1) watchdog timers associated with the CPUs, (2) CPU capability to detect sensor failures and deviations as implemented in the application software (in an approximate way), and (3) controller capability to determine the status of the CPUs. The ability to detect failures was considered in the FMEA, i.e., for each postulated failure mode, whether or not the failure can be detected was determined based on engineering knowledge. The results of the FMEA can be used in grouping failure modes and building FT and Markov models, as demonstrated in the example of the Main CPU module.

The fault-tolerant features of the DFWCS design are specific to the system, not standard fault-tolerant features built in the digital components. Therefore, they are not included in the raw data of component failures that were used to estimate component failure parameters. That is, the fault tolerant features will not be double-credited.

- Modeling Type I and II Interactions

As defined for this project, traditional methods only model interactions with the physical processes (Type I) in an approximate way; that is, if the DFWCS fails, the steam generator level is assumed to be either too high or too low. Interactions between the components/modules of the DFWCS (Type II) will be explicitly modeled.

- Common Cause Failures (CCFs)

Hardware CCFs between the Main and Backup CPUs, and between the MFV and pressure differential indication (PDI) controllers will be modeled. Due to lack of redundancy in the rest of the DFWCS, no other CCF will be modeled.

Software CCF between the Main and Backup CPUs will be modeled by including a transition and a basic event in the Markov and FT models, respectively. Contributions from application and support software will be considered separately. Due to scope limitation, the software failure rates will be assigned a range of values, without developing a software reliability model.

Probabilistic Data

- Hardware Failure Data

Chapter 8 reviewed different sources of data and documented the Bayesian analysis performed as part of this study. The Bayesian analysis provides generic hardware failure data with large uncertainties at a level of detail which is consistent with the expected level of detail of the reliability models in the next task. Because of the lack of validation of the raw data and the large uncertainties in the results, the failure rates estimated using Bayesian analysis are used in this project only to demonstrate the usefulness of the model for the DFWCS. For those components that are not included in the Bayesian analysis, other sources of data, such as PRISM, will be used. For failure mode distributions, Meeldijk [1996] and RAC [1997b] will be used.

The review documented in Chapter 8 did not find many CCF parameters of digital components at the level of detail of the expected reliability models of the next task. Since development of a database for CCF parameters of digital components is beyond the scope of this project, the Advanced Light Water Reactor (ALWR) Utility Requirement Document (URD) [EPRI 1993] will be used as the source of CCF parameters. The ALWR URD does not specifically address digital components and uses only very generic CCF parameters for components whose specific parameters are not set out in the document. These data are used in this study due to lack of applicable data.

- Software Failure Data

As stated earlier, no software reliability quantification method will be used because developing this method is beyond the scope of the current project. Place holders will be created in the reliability model with failure rates and probabilities assigned with a range of values.

Treatment of Uncertainty

- Parameter Uncertainty

The hierarchical Bayesian analysis performed in Chapter 8 on failure rates of digital components captures the large variability of data from different sources. The results of this analysis will be used in the quantification of the Markov and ET/FT models. For those components whose failure rates were not quantified in the hierarchical Bayesian analysis, failure data from other sources, e.g., PRISM, will be used, and large uncertainty parameters will be assumed. CCF parameters will be assumed to have large uncertainties also. Large uncertainties for these parameters are suggested to reflect the lack of information about the variability of the parameters. If time permits, uncertainties associated with failure mode distributions will be addressed by using alternative distributions to determine the sensitivity of the results to the distributions.

- Uncertainty Propagation

Monte Carlo simulations will be used to propagate the parameter uncertainties discussed above in the fault tree model to determine the distribution of the initiating event (IE) frequency. Approaches for carrying out this propagation in the Markov model also will be explored.

- Modeling Uncertainty

Key assumptions will be identified and described. Sensitivity calculations will be performed on key assumptions to determine their potential impacts on the results. If time permits, the assumptions will be ranked according to their effects on the estimated IE frequency. The ranking serves as a tool for determining where additional research would be needed to reduce the uncertainty.

- Completeness Uncertainty

Aspects related to the completeness of the Markov and ET/FT models will be briefly discussed.

Integration of the Digital System Model with a PRA

The next task completes the models to estimate the frequency that a loss of feedwater control takes place. Typically, such an initiating event is modeled in a PRA simply in terms of its annual occurrence frequency and no integration of the model of the initiating event with the PRA model is needed. However, it is desirable to carry out an integration to correctly account for this sharing of sensors and support systems, such as 120V AC buses. Modeling shared sensors was discussed earlier. Due to lack of a model for the RPS and ESFAS and, in general, lack of instrumentation and control (I&C) modeling in a PRA, demonstrating the integration of the DFWCS models with a PRA to account for the sharing of sensors and 120V AC buses may not be easy. Surrogate models of other I&C systems may be necessary.

Modeling of Human Errors

The detailed human reliability analysis (HRA) that would be necessary to evaluate the impact of human errors due to upgrading of hardware and software, and the impact of the design of the human-system interface, are beyond the scope of this project. The likelihood of successful manual control when automatic control is failed will be considered in this study using a simplified human reliability method, since a detailed HRA is beyond the scope of the project. It is desirable to consider these aspects of HRA when developing and quantifying a digital system reliability model.

Documentation and Results

Key assumptions will be identified and discussed. The effects of a few selected alternative assumptions will be discussed. Dominant contributors to the system failure will be identified and documented.

10. SUMMARY AND CONCLUSIONS

The U.S. Nuclear Regulatory Commission (NRC) is conducting research to identify and develop methods, analytical tools, and regulatory guidance to support (1) using information on the risks of digital systems in nuclear power plant (NPP) regulatory decisions, and (2) including models of these systems into NPP probabilistic risk assessments (PRAs). In support of this research, traditional methods will be used to develop and quantitatively assess reliability models of digital systems. As part of this work, two selected traditional methods will be applied to two benchmark systems, a digital feedwater control system (DFWCS) and a reactor protection system (RPS).

This report addresses the following principal work performed so far in preparation for the benchmark studies:

- Selection of two traditional reliability methods that have been used in modeling digital systems for further exploration of their capabilities and limitations,
- Development of desirable characteristics for reliability models of digital systems,
- A failure modes and effects analyses (FMEA) of the first benchmark system (i.e., a DFWCS),
- Approaches for developing the models of the DFWCS using the two selected methods, and
- Review of available sources of failure data, and a Bayesian analysis that provides some failure rate estimates of digital modules/components.

The insights and lessons learned from the above work are discussed in the following sections.

The methods and approaches in this report are applied to attempt to develop as complete a probabilistic model of a digital system as possible, given the current limitations of the state of the art. This maximizes the insights that may be gained about aspects of digital system models, even if some of these aspects are ultimately determined to not be significant or necessary.

10.1 Selection of Traditional Methods

This project includes the application of traditional reliability modeling methods to example digital systems to support the development of tools and methods for including probabilistic models of these systems into PRAs. In determining which traditional methods to select for trial application, two factors were considered. First, because the ultimate goal of this project is to support the NRC in developing regulatory guidance for using risk information related to digital systems in the licensing actions of current or future NPPs, heavy emphasis was placed on those methods likely to be used by the nuclear industry. Secondly, many dynamic methods (i.e., methods that explicitly attempt to model the interactions between a plant system and the plant's physical processes, and the timing of these interactions) were not considered because they are the subject of a parallel NRC research project.

Considering the above factors, the two traditional reliability modeling methods selected for trial application as part of this project are the traditional Event Tree/Fault Tree (ET/FT) method and the Markov method. The traditional ET/FT method has been commonly used by the U.S. nuclear power industry and in other countries and industries. The Markov method can be a powerful tool for analyzing digital systems because it can explicitly model system configurations arising from the ability of some digital systems to detect failures and change their configuration during operation. The Markov method can also explicitly treat failure and repair times. Further, the Markov method has been used previously to model NPP systems and digital systems. A number of other methods that may be useful for developing and quantifying reliability models of digital systems are discussed in an appendix to this report. While it is not practical to further explore all of these methods as part of the current project, some of them may warrant further attention if other studies demonstrate their capability and practicality.

10.2 Development of Desirable Characteristics for Reliability Models of Digital Systems

Desirable characteristics were developed that address those aspects of a model that capture the design features of a digital system that could affect system reliability and plant risk. An external expert panel meeting was held to review the draft characteristics before they were finalized. The characteristics could provide input to the technical basis for risk evaluations. A total of 52 characteristics were classified into the following nine broad categories:

1. Level of Detail of the Probabilistic Model
2. Identification of Failure Modes of the Components of a Digital System
3. Modeling of Software Failures
4. Modeling of Dependencies
5. Probabilistic Data
6. Treatment of Uncertainty
7. Integration of the Digital System Model with a PRA Model
8. Modeling of Human Errors
9. Documentation and Results

The focus of the characteristics is on the modeling of the design features of digital systems. In addition, the PRA model is expected to meet the general PRA guidelines provided in documents, such as the PRA Procedures Guide [Hickman 1983] and the American Society of Mechanical Engineers (ASME) standard for PRA for NPP applications [ASME 2005].

The desirable characteristics are potentially relevant to any kind of probabilistic model of a digital system. It may be possible to address them by different methods.

There are some characteristics for which methods and/or data may not be currently available. Furthermore, it is debatable whether some of the characteristics are relevant. The intent was to include all characteristics addressing modeling of the design features that are potential contributors to system unreliability and plant risk. Some characteristics may be modified later using the findings from the benchmark studies.

10.3 Performance of an FMEA of the DFWCS

FMEA was used as a tool for the study team to become familiar with the system design detail, and to identify the failure modes and effects of the components of this system. Specifically, a detailed FMEA of the Main central processing unit (CPU) module was used to demonstrate how the FMEA could be used in developing probabilistic models of the DFWCS.

In the FMEA, the Main CPU module was broken down into its individual digital components. The failure modes of each component then were analyzed to determine the capability of the system to detect them and the effects of each failure mode on the Main CPU module. The FMEA at the component level was also used to develop the failure modes of the module that in turn can be used for modeling the system.

The insights learned from the process of performing the FMEA are summarized below.

- FMEA is a well-known method used to identify failure modes of a system and their effects or consequences on the system. A few guidance documents for performing an FMEA are available, i.e., Institute of Electrical and Electronics Engineers (IEEE) Standard 352 [IEEE 1987], Military Standard 1629A [DOD 1984], Military Handbook 338b [DOD 1998], and the British Standard Institute 5760-5 [BSI 1991]. However, the current documents provide general information, and do not give specific guidance on performing an FMEA for a digital system.

- Specific guidance about how to perform FMEA of digital systems appears to be lacking (at least in the public domain). IEEE 352 and other publications provide generic guidance on FMEA, but no specific guidance on FMEA of digital systems. An even bigger issue is that there is no generic or standard list of failure modes of digital systems/components. Therefore, if an FMEA of a system is performed by different analysts, they may arrive at different sets of failure modes.

- Due to the great complexity of the DFWCS, it is very difficult to reliably predict the response and effects of a single failure. If several failures are analyzed concurrently, the analysis becomes even more difficult. FMEA is an excellent tool for learning about and understanding the design, the operation, and some possible safety weaknesses of the system. However, FMEA by itself is not a sufficient tool to determine how specific component-level failure modes affect systems as complex as a digital system. Hence, it is advisable that other more sophisticated tools, such as simulation tools, be used to analyze the interactions between the components of a digital system and the effects of one or more failures. For the DFWCS, an integrated simulation tool would model all the components in the system in detail and allow determination of the system response to postulated failures identified in the FMEA. Development of the simulation tool would require detailed knowledge of all of the software used by the system. Ideally, the FMEA and these tools would be used in combination to identify the vulnerabilities of the system in a more comprehensive way than using FMEA alone.

10.4 Modeling Approach

The DFWCS system controls a steam generator level during full power, low power, and shutdown conditions. The main modules of the system include two redundant microprocessors, i.e., the Main and Backup CPUs, and controllers for the main feedwater regulating valves, bypass feedwater regulating valves, and main feedwater pump. In the first benchmark study, two models for estimating the frequency of loss of the DFWCS during power operation will be developed using the two selected traditional methods. They essentially estimate the frequency of the initiating event of loss of control of feedwater to the steam generator.

10.4.1 Development of Markov Model

A Markov model of the DFWCS can be expressed in terms of a set of linear differential equations modeling the transitions among system states. It is assumed that the system is initially in an operable state with all modules in a good condition. Every time a module of the DFWCS fails, the DFWCS transits to another state. In general, the system experiences several "jumps" of states until it reaches the failed state, i.e., the state that causes an initiating event.

The development of a Markov model of the DFWCS builds on the information obtained and insights gained in the FMEA. Markov models of the modules of the system, i.e., controllers, watchdog timers, sensors, and main feedwater regulatory valve positioners, will be developed in the same way the Markov model of the Main CPU module was developed (in Chapter 6) using a detailed FMEA of the components of the modules.

System-level Markov states can be defined in terms of the module-level Markov states. In general, a very large number of system states can be defined, and possible transitions among them have to be determined. The total number of system-level states could be as large as the product of the numbers of possible module states, which would make solving the model impractical. However, since the modules are interconnected, and affect each other's operation, not every combination of module states is possible, nor is every transition between the system states possible. In addition, a few system-specific considerations potentially can significantly reduce the number of system-level states and the size of the associated transition matrix. For example, the DFWCS system does not have multiple redundancies and, therefore, many of the failure modes of the controllers are single failures of the system. Once a system-level failure state is reached, it is modeled as an absorbing state with no transition out of it. That is, once the system is failed, no repair of the system is modeled.

Construction of the transition matrix is expected to be an important and difficult part of the model development. The main reason is that the modules of the system are connected, exchange information, and affect each other's operation. Consideration of these dependencies requires detailed knowledge of the software involved.

The process of developing the transition matrix is an iterative process. Each successive step involves postulating one additional failure mode/transition for each possible system-level state, and determining whether or not system failure occurs. For each postulated failure mode of a module, the interactions with other modules must be accounted for to determine how the failure mode of the module affects the system. The transition rates of the possible transitions are then estimated. The process continues until all transition paths to the failure state are identified. The

formulation of the transition matrix essentially is a manual process requiring an extensive knowledge of how the system works, especially the software.

The system failure state referred to above only involves loss of automatic control by the DFWCS. Typically, for those failures that cause loss of automatic control, manual control of the system may still be possible. The degree of difficulty in assuming manual control depends on the specific failures, e.g., whether or not the failure(s) is (are) annunciated in the control room or only indicated at the plant computer, and whether or not the hardware needed for manual control is adversely affected by the failure(s). The detailed human reliability analysis (HRA) that would be necessary to accurately evaluate the likelihood of successful manual control is beyond the scope of this proof-of-concept study, and therefore only a very simplified treatment of HRA will be undertaken. For some regulatory applications, a more detailed HRA may be necessary.

10.4.2 Development of Fault Tree Model

Chapter 7 delineates the application of the traditional ET/FT method to construct and solve a probabilistic model of the DFWCS. The deductive approach used to build a fault tree is used to illustrate the construction of a tree modeling the failure of a DFWCS. The DFWCS is a control system that is normally running during power operation. Accordingly, failure of the DFWCS is defined as loss of automatic and manual control of the loop associated with a DFWCS. As discussed in Chapter 3, failure of the DFWCS will be modeled to determine its frequency of occurrence as an initiating event. Hence, the top event is defined as "Loss of control of the loop associated with a DFWCS," and the fault tree is modeling the probability of this loss within an one-year period. As described in Chapter 2, it is important to include in the fault tree all relevant failures of the modules of a system contributing to system failure, such as the independent failure modes of these modules and dependent failures including common cause failures (CCFs). The fault tree to be developed in the next task is expected to include the hardware and software failure modes of the modules of a DFWCS.

The information from the FMEA is used to build the fault tree by determining the immediate cause of each event in the tree. This process of refinement is continued until reaching the level of detail considered appropriate for capturing the relevant contributors to the failure of the system, that is, the level of the failure modes of the digital modules determined in Chapter 5. In general, these failure modes become basic events in the fault tree. Thus, the modeling in the fault tree of the failure modes of some relevant digital modules of the DFWCS also is illustrated. Since the DFWCS is running during power operation, the components of this system also are running during this time. Hence, a basic event of the fault tree usually represents the failure of a module to run over one year. The exponential distribution can be employed to calculate the probability of this failure.

10.5 Development of Failure Parameter Database

In an attempt to develop a failure parameter database, (1) currently available failure data analysis methods and failure rate databases of digital components were reviewed, (2) studies that estimated failure rates of selected digital components using raw data from different industries were reviewed, and (3) an approach for estimating failure parameters using the data extracted from the PRISM [RAC PRISM] was applied.

To identify the available data and analysis methods on digital systems and components, a systematic search was conducted. The different data sources are organized into the following three groups:

- Commercially available databases, such as Reliability Prediction Methods (RPMs), to obtain reliability parameters of digital systems and components (e.g., Military Handbook 217F and PRISM).

- Analyses used in different industries to obtain reliability parameters of specific digital components for specific applications (e.g., NRC RPS unavailability studies and new reactor vendor PRAs).

- Databases containing digital system and components failures in NPP operation (e.g., licensing event reports and equipment performance and information exchange).

The review of the available databases and studies on failure experience found that the only generic databases for digital components are those based on RPMs that lack accuracy and treatment of uncertainty, and the studies of selected digital components only provide failure estimates for some specific components. Rapid obsolescence of digital equipment is one reason data collection is difficult. In general, component-specific data, i.e., failure data collected from the same component, is needed for reliability modeling. However, raw data of digital components for nuclear applications are lacking, at least, in the public domain, due to lack of operating experience and effort to collect them. Failure parameter data is an area of weakness for modeling digital systems.

In order to develop a generic database of digital components, the raw data of the PRISM database was extracted in the form of the number of failures in a number of operating hours, and a Hierarchical Bayesian analysis was applied to the data in order to obtain the generic failure rates of digital components. The PRISM data are the only publicly available data that are at the desired/lowest level of detail and covers a wide collection of components. Due to lack of information about the sources of the raw data and the method used in collecting them, the correctness of the raw data collection was not validated. It is questionable whether the data are applicable to the components of the DFWCS. The applicability of the data is an important issue that directly contributed to the large uncertainties obtained in the failure rate estimates. Because of the lack of validation and the large uncertainties, the failure rate estimates are not appropriate for quantifying models that are to be used in support of decision-making. In specific applications, an attempt to collect component-specific data should be made; these data then can be used to update the failure estimates from a generic database using Bayesian analysis.

In the area of CCF failure parameters, there is no established database for digital components. The need for failure mode specific CCF parameters makes it more difficult to model digital systems. A few studies that modeled CCFs of digital components did not document how the parameters were estimated. Additional research in this area is needed.

10.6 Next Steps

This report describes the approaches for developing Markov and ET/FT models of the digital feedwater control system. It also includes the progress made so far in the model development. In the next task of this project, the detailed approaches will be developed and implemented, and

the models and results will be documented in a NUREG/CR report. The lessons learned in developing the approaches and the models for the first benchmark study will benefit the second benchmark study (of a reactor protection system). The two benchmark studies will serve as a demonstration of the use of the Markov and ET/FT methods in modeling two different types of digital systems, i.e., a control system and a protection system. They will also help identify potential areas of weakness in the state-of-the-art using traditional PRA methods and where additional research and development may be needed.

10.7 Recommendations for Research

The activities on delineating the probabilistic models of the DFWCS indicated the following preliminary list of areas where additional research could enhance the state-of-the-art:

- Methods for defining and identifying failure modes and effects of digital systems. The methods would determine how failure modes propagate from their sources to the rest of the system and other systems of the plant, e.g., by using simulation models, taking into consideration the communication networks, voting, and synchronization.

- Methods and parameter data for modeling self-diagnostics, reconfiguration, and surveillance, including using other components to detect failures, e.g., watchdog timers and microprocessors. The data would include the fraction of failures that can be detected, e.g., coverage, and break down the failure rates by failure mode.

- Better data for hardware failures of digital components, avoiding the potential issue of double crediting fault-tolerant features, such as self-diagnostics.[4]

- Better data for CCFs of digital components.

- Methods for estimating the risk from software faults in both application and support software.

- Methods for modeling software CCF across system boundaries (e.g., due to common support software).

- Methods for considering modeling uncertainties in modeling of digital systems.

- Methods for human reliability analysis associated with digital systems.

[4] Double-crediting of fault-tolerant features can be an issue for software failures also.

11. REFERENCES

Aldemir, T., Miller, D.W., Stovsky, M.P., et al., "Current State of Reliability Modeling Methodologies for Digital Systems and Their Acceptance Criteria for Nuclear Power Plant Assessments," NUREG/CR-6901, February 2006.

Aldemir, T., Stovsky, M.P., Kirschenbaum, J., et al., "Dynamic Reliability Modeling of Digital Instrumentation and Control Systems for Nuclear Reactor Probabilistic Risk Assessments," NUREG/CR-6942, August, 2007.

American Society of Mechanical Engineers, ASME RA-Sb-2005 Addendum to ASME RA-S-2002, Standard for Probabilistic Risk Assessment for Nuclear Power Plant Applications. Date of Issuance: December 30, 2005.

Apostolakis, G.E., and Chu, T.L., "Unavailability of Systems Under Periodic Test and Maintenance," *Nuclear Technology*, v 50, n 1, Mid-Aug, 1980, p 5-15.

Atwood, C., "Handbook of Parameter Estimation for Probabilistic Risk Assessment," NUREG/CR-6823, September 2003.

B&B Electronics, "RS-422 and RS-485 Application Notes," Oct. 1997.

Barlow, R.E. and Proschan, F., *Statistical Theory of Reliability and Life Testing,* Holt, Rinehart and Winston, Inc., 1975.

Bickel, J., "Risk Implications of Digital Reactor Protection System Operating Experience," *Reliability Engineering & System Safety*, 2006.

Blanton, C.H., and Eide, S.A., "Savannah River Site Generic Data Base Development (U)," WSRC-TR-93-262, Westinghouse Savannah River Company, June, 1993.

British Standard Institute, "Reliability of Systems, Equipment and Components - Part 5: Guide to Failure modes, Effects and Criticality Analysis (FMEA and FMECA)," BS 5760-5:1991.

Brown, L.M., "Comparing Reliability Predictions to Field Data for Plastic Parts in a Military, Airborne Environment, Northrop Grumman Electronic Systems," Proceedings of Annual Reliability and Maintainability Symposium, 2003.

Bunea, C. et al., "Two-stage Bayesian Models - Application to ZEDB Project," *Reliability Engineering & System Safety*, Vol. 90, pp. 123 - 130, 2005.

Chu, T.L., Martinez-Guridi, G., Lehner, J., and Overland, D., "Issues Associated with Probabilistic Failure Modeling of Digital Systems," NPIC&HMIT2004, Columbus Ohio, September 2004.

Chu, T. L., Martinez-Guridi, G., Yue, M., and Lehner, J., "A Review of Software-Induced Failure Experience," NPIC&HMIT2006, Albuquerque, NM, November 12-16, 2006.

Chu, T.L., Martinez-Guridi, G., Yue, G., and Lehner, J., "Basis for Using Software Failure Rates and Probabilities in Probabilistic Failure Modeling of Digital Systems of a Nuclear Power Plant," IAEA Technical Meeting on Common Cause Failures in Digital Instrumentation and Control Systems of Nuclear Power Plants, June 19-21, 2007, Bethesda, Maryland, USA.

Code of Federal Regulation, Licensee Event Report System, 10 CFR 50.73, 2007.

Department of Defense, "Procedures for Performing a Failure Mode, Effects, and Criticality Analysis, Military," Standard 1629A, Notice 2, November 1984.

Department of Defense, "Reliability Prediction of Electronic Equipment," Notice 2, MIL-HDBK-217F, February 18, 1995.

Department of Defense, "Electronic Reliability Design Handbook," Military Handbook 338B, October 1998.

Dylis, D.D., and Priore, M.G., "A Comprehensive Reliability Assessment Tool for Electronic Systems," Proceedings of the Annual Reliability and Maintainability Symposium, 2001.

Eide, S.A., et al., "Reliability Study: Westinghouse Reactor Protection System, 1984 - 1995," Idaho National Engineering and Environmental Laboratory and Lockheed Martin Idaho Technologies Company, NUREG/CR-5500, Vol. 2, INEEL/EXT-97-00740, April 1999.

EPRI, Advanced Light Water Reactor (ALWR) Utility Requirements Document, Vol II, ALWR Evolutionary Plant, Ch. 1, App. A, "PRA Key Assumptions and Ground Rules," Rev. 6, December 1993.

EPRI, "Generic Requirements Specification for Qualifying a Commercially Available PLC for Safety-Related Applications in Nuclear Power Plants," EPRI TR-107330, December 1996.

Fisher Controls International, Inc., FIELDVUE DVC5000 Series Digital Valve Controller, Bulletin 62.1: DVC5000, January, 2001.

Fisher Controls International, Inc., Instruction Manual: 3570 Series Pneumatic Valve Positioners, July, 2006.

Generic Electric, "ESBWR Probabilistic Risk Assessment," NEDO-33201, Revision 1, 2006.

Hauge,S., Langseth, H. and Onshus, T., "Reliability Data for Safety Instrumented Systems," PDS Data Handbook, 2006 Edition, SINTEF, April, 2006a.

Hauge, S., Hokstad, P., Langseth, H., and Oien, K., "Reliability Prediction Methods for Safety Instrumented Systems: PDS Method Handbook," SINTEF, April 2006b.

Hickman, J.W., et al., "PRA Procedures Guide, A Guide to the Performance of Probabilistic Risk Assessments for Nuclear Power Plants," NUREG/CR-2300, January 1983.

Hofer, E., et al., "On the Solution Approach for Bayesian Modeling of Initiating Event Frequencies and Failure Rates," Risk Analysis, Volume 17, No. 2, 1997.

Humphreys, M. and Daniels, B.K., "How do Electronic System Failure Rate Predictions Compare with Field Experience?," SRS/GR/58, Systems Reliability Service, United Kingdom Atomic Energy Authority, Culcheth, Warrington, WQA3 4NE, UK, October, 1982.

IEEE Guide for General Principles of Reliability Analysis of Nuclear Power Generating Station Safety Systems, IEEE Std. 352, Institute of Electrical and Electronics Engineers, Inc., 1987.

IEEE Standard Reliability Data for Pumps and Drivers, Valve Actuators, and Valves, IEEE Std. 500-1984, Institute of Electrical and Electronics Engineers, Inc., September 16, 1986.

International Electrotechnical Commission, "Functional Safety of Electrical/Electronic/ Programmable Electronic Safety-Related Systems," IEC 61508, Parts 1 to 7, various dates.

Kaplan, S., "On a 'Two-stage' Bayesian procedure for determining failure rates," IEEE *Transactions on Reliability*, R-33, 227-232, 1984.

Marshall, F.M. et al., "Common-cause Failure Database and Analysis System: Event Definition and Classification," NUREG/CR-6268, Vol. 2, June 1998.

Masoneilan North American Operations, Masoneilan Electropneumatic Positioner Instructions, September, 1997.

Meeldijk, V., *Electronic Components Selection and Application Guidelines*, John Wiley & Sons, 1996.

MicroMod Automation, Instruction Manual: Multi-loop Process Controller 53MC5000, Rev. 1, 2004.

Mitchell, C.M. and Williams, K., "Failure Experience of Programmable Logic Controllers Used In Emergency Shutdown Systems, *Reliability Engineering and System Safety*," 39:329-331, 1993.

Niedzballa, G., "Teleperm XS Implementation & Upgrading of Safety I&C," Framatome ANP, International Nuclear Forum, Varna, June 2004.

Nuclear Energy Agency, "Operation and Maintenance Experience with Computer-Based Systems in Nuclear Power Plants," Committee on the Safety of Nuclear Installations, A Report by the PWR-1 Task Group on Computer-based Systems Important to Safety, NEA/CSNI/R(97)23, September 10, 1998.

Nuclear Energy Institute, "Industry Guideline for Monitoring the Effectiveness of Maintenance at Nuclear Power Plants," NUMARC 93-01, Revision 2, February 22, 2002.

Nuclear Regulatory Commission, "Use of Probabilistic Risk Assessment Methods in Nuclear Regulatory Activities," Final Policy Statement, August 16, 1995.

Nuclear Regulatory Commission, "Assessing and Managing Risk Before Maintenance Activities at Nuclear Power Plants," Regulatory Guide 1.192, May 2000a.

Nuclear Regulatory Commission, "Event Reporting Guidelines 10 CFR 50.72 and 50.73," NUREG-1022, Rev. 2, October 2000b.

Nuclear Regulatory Commission, "NRC Digital System Research Plan, FY 2005 – FY 2009," Revision 06/2, April 2006.

Nuclear Regulatory Commission, "Effects of Ethernet-Based, Non-Safety Related Controls on the Safe and Continued Operation of Nuclear Power Stations," NRC Information Notice 2007-15, April 17, 2007a.

Nuclear Regulatory Commission, "An Approach for Determining the Technical Adequacy of Probabilistic Risk Assessment Results for Risk-Informed Activities," Regulatory Guide 1.200, Rev. 1, January 2007b.

Oconee Nuclear Station, "Defense-in Depth and Diversity Assessment of the RPS/ESFAS Digital Upgrade," March 20, 2003.

"Offshore Reliability Data Handbook," OREDA Participants, PO Box 370, N-1322 HOVIK, Norway, 1984.

Paula, H.M, Roberts, M.W. and Battle, R.E., "Operational Failure Experience of Fault-Tolerant Digital Control Systems," *Reliability Engineering and System Safety*, 39:273-289, 1993a.

Paula, H.M, "Failure Rates for Programmable Logic Controllers," *Reliability Engineering and System Safety*, 39:325-328, 1993b.

Pinho, L.M., Vasques, F., and Tovar, E., "Integrating inaccessibility in response time analysis of CAN networks," Proceedings of 2000 IEEE International Workshop on Factory Communication.

Poloski, J.P., et al., "Rates of Initiating Events at U.S. Nuclear Power Plants: 1987-1995," NUREG/CR-5750, February, 1999.

Porn, K., "Two-Stage Bayesian Method Used for the T-book Application," *Reliability Engineering and System Safety* 51 (1996) 169-179.

Priore, M.G., Goel, P.S., and Itabashi-Campbell, R., "TRW Automotive Assesses PRISM Methodology for Internal Use," The Journal of the Reliability Analysis Center, First Quarter, 2002.

Radio Technical Commission for Aeronautics (RTCA) Special Committee 167, Software Considerations in Airborne Systems and Equipment Certification, RTCA/DO-178B, December 1992.

Reliability Analysis Center (RAC), "PRISM User's Manual, Version 1.4," Prepared by Reliability Analysis Center Under Contract to Defense Supply Center Columbus.

RAC, "Nonelectronic Parts Reliability Data," EPRD-95, 1995.

RAC, "Electronic Parts Reliability Data," EPRD-97, 1997a.

RAC, "Failure Mode/Mechanism Distributions," A DOD Information Analysis Center, FMD-97, December 1997b.

RAC, and Performance Technology, "New System Reliability Assessment Method," Prepared for Rome Laboratory, IITRI Project No. A06830, June 1, 1998.

Rouvroye, J.L., and Brombacher, A.C., "New Quantitative Safety Standards: Different Techniques, Different Results," *Reliability Engineering and System Safety*, 66 (1999) 121-125.

Schneidewind, N.F., and Keller, T.W., IBM Federal Services Company, "Applying Reliability Models to the Space Shuttle," *Software, IEEE*, Volume 9, Issue 4, Page(s):28-33, July 1992.

Shooman, M.L., "*Probabilistic Reliability: An Engineering Approach*," McGraw-Hill Book Company, 1968.

Siewiorek, D.P., and Swarz, R. S., "*Reliable Computer Systems, Design and Evaluation*," Second Edition, Digital Press. 1992.

Siu, N., and Kelly, D. L., "Bayesian Parameter Estimation in Probabilistic Risk Assessment," *Reliability Engineering and System Safety* 62, pp. 89–116, 1998.

Smith, C.L., and Womack, J.B., "Raytheon Assessment of PRISM as a Field Failure Prediction Tool," Proceedings of the 2004 Annual Reliability and Maintainability Symposium.

Spiegelhalter, D., et al., "WinBUGS User Manual," Version 1.4, January 2003.

Sudarno, W., et al., "Reliability Study: Digital Engineered Safety Feature Actuation System of Korean Standard Nuclear Power Plant," KAERI/TR-2467, 2003.

Telcordia, "Reliability Prediction Procedure for Electronic Equipment," SR-332 Issue 1, May 2001.

Triconex Corporation, "TRICON Reliability Analysis," Irvine, California.

Varde, P.V., et al., "Reliability Analysis of Protection System of Advanced Pressurized Water Reactor-APR 1400," Korea Atomic Energy Research Institute, KAERI/TR-2468/2003

Vesely, W.E., Goldberg, F.F., Roberts, N.H., et al., "Fault Tree Handbook," NUREG-0492, January 1981.

Westinghouse Electric Company, "AP600 Probabilistic Risk Assessment," Westinghouse Electric Corporation, ENEL, Revision 7, June 28, 1996.

Westinghouse Electric Company, "AP1000 Probabilistic Risk Assessment," AP1000 Document No. APP-GW-GL-700, Revision 8, 2004.

Wierman, T.E., et al., "Reliability Study: Combustion Engineering Reactor Protection System, 1984 - 1998," Idaho National Engineering and Environmental Laboratory, NUREG/CR-5500, Vol. 10, INEL/EXT-97-00740, July 2002.

Yue, M., and Chu, T.L., "Estimation of Failure Rates of Digital Components Using a Hierarchical Bayesian Method," PSAM8, New Orleans, Louisiana, May 14-19, 2006.

APPENDIX A

SUMMARY REPORT OF THE EXTERNAL REVIEW PANEL MEETING ON RELIABILITY MODELING OF DIGITAL SYSTEMS

(MAY 23-24, 2007)

TABLE OF CONTENTS

		Page
A.1.	INTRODUCTION	A-1
A.1.1	Background	A-1
A.1.2	External Review Panel Meeting Process	A-2
A.2.	PRESENTATION OF PRELIMINARY COMMENTS BY EACH PANEL MEMBER	A-3
A.3	TRADITIONAL METHODS AND THEIR APPLICATIONS	A-12
A.3.1	Defining and Identifying "Traditional" Methods	A-13
A.3.2	Conclusions Stated versus Applications of the Criteria Presented in the Report	A-14
A.3.3	Criteria Applied to Methods or to Models/Applications	A-15
A.4.	COMMENTS ON REVIEW CRITERIA	A-16
A.4.1	General Comments on Review Criteria	A-16
A.4.2	Category 1: Level of Detail of the Model	A-16
A.4.3	Category 2: Identification of Failure Modes of the Components of Digital Systems	A-17
A.4.4	Category 3: Software Failures	A-18
A.4.5	Category 4: Modeling of Dependencies	A-19
A.4.5.1	Subcategory 4.1: Communication Networks/Buses	A-20
A.4.5.2	Subcategory 4.3: Support Systems	A-21
A.4.5.3	Subcategory 4.4: Sharing Hardware	A-21
A.4.5.4	Subcategory 4.5: Interactions of Digital Systems with Other Systems	A-22
A.4.5.5	Subcategory 4.6: Modeling of Fault Tolerance Features	A-23
A.4.5.6	Subcategory 4.2: Common Cause Failures	A-23
A.4.6	Category 7: Probabilistic Data	A-24
A.4.7	Criteria 8.4 and 8.5: Uncertainty	A-25
A.4.8	Category 6: Ease of Integration with a PRA Model	A-26
A.4.9	Category 5: Human Errors	A-26
A.4.10	Criteria 8.1 - 8.3: Documentation and Results	A-27
A.5.	CONCLUDING REMARKS	A-27
Attachment A	Agenda of External Review Panel Meeting on Selection of Traditional Methods for Reliability Modeling of Digital Systems	A-29
Attachment B	List of Documents Sent to Panel Members	A-31
Attachment C	Expert Panel Meeting Attendees	A-32
Attachment D	Biographies of Panel Members	A-33
Attachment E	Written Comments Provided by Reviewer A	A-35
Attachment F	Written Comments Provided by Reviewer C	A-44
Attachment G	Written Comments Provided by Reviewer D	A-46
Attachment H	Order for Addressing Review Criteria Categories/Subcategories	A-50

A.1. INTRODUCTION

This summary report documents the discussions that took place at an external review panel meeting on traditional methods for modeling digital systems as part of a probabilistic risk assessment (PRA). The meeting was held at Brookhaven National Laboratory (BNL) on May 23 and 24, 2007.

At the time of the meeting, the scope and content of the project differed somewhat from those presented in the main body of this NUREG/CR. A notable difference is that the desirable characteristics presented in Chapter 2 were considered to be "review criteria." Accordingly, the text in this appendix refers to these criteria, but the discussions are applicable to the characteristics.

A.1.1 Background

The U.S. Nuclear Regulatory Commission (NRC) is currently performing research on digital system risk assessment. Their objective is to identify and develop methods, analytical tools, and regulatory guidance to support (1) using risk information on digital systems in nuclear power plant (NPP) licensing decisions, and (2) including models of digital systems into NPPs' PRAs.

The NRC is exploring, in parallel, both the dynamic and traditional methods of modeling digital-system reliability. For this research, the latter can be thought of as the more well-established and commonly used methods of NPP system reliability modeling (e.g., fault tree modeling). Dynamic methods can be thought of as methods of NPP system-reliability modeling that attempt to explicitly model the coupling between a digital system and the plant's physical processes. (Note: The distinction between traditional and dynamic methods was further discussed during the meeting, as documented in Section 3.1 of this report.)

Under a contract with the NRC, BNL conducted the following work as part of the research on traditional methods of modeling digital system reliability:

1. Identified for further exploration two traditional methods that represent a spectrum of capabilities for modeling and quantitatively assessing the reliability of digital systems.

2. Developed criteria for evaluating reliability models of digital systems, which could eventually provide input to the technical basis for risk-informed decision-making.

3. Reviewed reliability models developed using traditional methods, such as fault tree and Markov methods, against the criteria to assist in determining the capabilities and limitations of the state-of-the-art of traditional reliability models.

The findings of this work are documented in a draft letter report (T. L. Chu, G. Martinez-Guridi, M. Yue, and J. Lehner, "Probabilistic Modeling of Digital Systems at Nuclear Power Plants: Traditional Methods Selection," Brookhaven National Laboratory, Draft Letter Report, April 2007).

A.1.2 External Review Panel Meeting Process

To more fully involve the technical community in identifying the most promising traditional methods for reliability modeling of digital systems, and in developing criteria for evaluating such reliability models, an external review panel was established to review the findings from the BNL activities. The panel was comprised of six members, all of whom have expertise in modeling and quantifying digital-system reliability, as well as in PRA. The objectives of the external review panel were to assess the following:

1. The identification of traditional methods and their application.
2. The draft criteria used to review reliability models of digital systems.
3. The limitations of the state-of-the-art in modeling digital systems.

The responsibilities of the panel members were to (1) study BNL's draft letter report before the meeting of the expert panel, and (2) attend the two-day meeting to satisfy the objectives listed above. Attachment A of this report contains the agenda for the meeting.

Panel members received a set of background information before the meeting. In addition setting out the objectives of the meeting and describing the process to be followed during the meeting, the background information contained the BNL draft letter report, and other related papers and reports. The materials sent to the panel members are listed in Attachment B. Furthermore, each member was asked to judge beforehand whether the draft criteria for evaluating reliability models of digital systems were appropriate and to identify any additions, deletions, or modifications to them.

The meeting took place at BNL on May 23 and 24, 2007. It was conducted with the help of a facilitator who was responsible for aiding the discussions and keeping them focused. In addition to the panel members and the facilitator, the meeting was attended by the authors of the BNL report and the NRC Project Manager. Their role was to provide answers and clarifications in response to the panel members' questions. Attachment C of this summary report lists the attendees' names; brief biographies of the experts are given in Attachment D.

The remaining sections of this report summarize the discussions of the meeting roughly in chronological order. At the beginning of the meeting, each panel member gave a short presentation of his preliminary thoughts on the BNL report (documented in Section 2 of this report). The next major topic for discussion was the identification of traditional methods and their relevant applications, as summarized in Section 3. Section 4 of this report summarizes the discussions on the proposed criteria, in the same order that their categories were reviewed. The concluding remarks are summarized in Section 5.

It should be noted that it was not a goal of the meeting to obtain a consensus among the panel on any particular aspect of the work under review. Rather, the goal was to receive feedback from a broad spectrum of individuals who have significant experience in the subject. Accordingly, while the following sections report any points of general agreement among the panel members, most of the information is in the form of comments from individual panel members.

A.2. PRESENTATION OF PRELIMINARY COMMENTS BY EACH PANEL MEMBER

During the morning of May 23, 2007, each member of the external review panel gave a short presentation of his preliminary thoughts on using traditional methods for reliability modeling of digital systems, such as suggestions for alternative methods/applications, and on the draft criteria used for reviewing the digital system reliability models.

Points of general agreement among the panel members include the following:

- The BNL report contains valuable information about traditional methods.
- A substantial amount of probabilistic data of digital components/systems has been generated, but most of it is not publicly available. An important and difficult issue is how to obtain this data.

- The term "criterion" should be used instead of the term "requirement" in the report due to the regulatory implications of the word "requirement."

The comments by individual members of the panel are presented next, in the order in which they gave their presentations.

Reviewer (A)[1]

General comments relative to the approach to be used in the evaluation of modeling methods and their application to digital systems:

- The term "digital systems" is very broad and covers a whole spectrum of systems with very different characteristics, therefore:
  - Different modeling methods may be needed for different areas of application within this broad spectrum.
  - It may not be possible to apply the same general criteria for evaluation of methods that are intended to address a specific area of application rather than another. For example, by and large, one can see a digital system as comprising three basic layers of components: hardware, operating system, and application software, plus the "external balance-of-system" with which the digital system interface. Each layer has different a different type of functionality and is subject to different types of faults and failures. Thus different types of modeling and model evaluation criteria may apply to the different layers.

- There is a big difference between evaluating the suitability of a modeling framework or method to cover a range of possible applications, and evaluating one specific application of the method, in light of a set of criteria that particular application of the method was not set out to satisfy in the first place. By adopting the latter mode of evaluation it is very difficult to understand and assess the true strength of a framework or method.

[1] In addition to his comments during the panel meeting, reviewer A provided some comments after the meeting that clarify and expand his points of view. Accordingly, this Appendix includes all his comments.

Observations on operating experience of mission-critical digital systems and NPP digital systems modeling needs:

- National Aeronautics and Space Administration (NASA) operational experience indicates that a majority of the mission failures that have occurred and in which digital control systems and associated software were involved occurred due to system and software design errors, which became failures in mission execution because the digital system and software had to face unanticipated system conditions. Failures due to other causes (e.g., software coding errors) have not occurred in mission-critical systems, probably because they can be identified and eliminated with traditional validation and verification techniques before mission execution.

- In addressing NPP digital system modeling needs, it is important to have the ability to adapt the modeling approach to the particular type of system and interactions that need to be modeled. It is not prudent to suggest, as often is suggested by the industry, that for risk assessment and safety purposes it is sufficient to address only those systems categorized as "safety related" (i.e., reactor protection system [RPS] and engineered safety features actuation system [ESFAS]), whereas "non-safety related" systems are of secondary importance. There are a number of good reasons why a flexible portfolio of modeling tools that covers both "safety related" and "non-safety related" systems.

  - In a NPP, serious challenges to operational safety may come from systems that are nominally categorized as "non-safety related." For example, the Three-Mile Island accident occurred not because of safety related systems failures, but because of triggering events in the non-safety related feedwater system, and interactions between the plant systems and their human operators.

  - An RPS is based on open loop logic and in that respect can be probably tested satisfactorily using traditional validation and verification techniques. The same is not true of a digital feedwater and level control system, which has logic and timing-dependent control loops and is also potentially subject to the effect of human errors introduced by its interface with human operators.

  - In general, interconnected control systems are considerably more complex than safety systems (as defined in the NPP context), because of their combination of logic, algorithms, and human interfaces.

- In light of all the above, one can conclude that "traditional methods" may be perhaps adequate to deal with safety systems that are relatively simple (in their logic and degree of permitted interaction with other systems). However, more advanced dynamic modeling methods appear to be definitely needed to address the potential of system failures initiated by control systems and unnecessary challenges to safety systems that may progress to unexpected and undesirable consequences.

Comments on the nature and suitability of some of the methods evaluated by BNL:

- Military Handbook 217 (MIL-HDBK-217) is hardly a method for modeling digital systems, even though it contains information of how to assess the reliability of certain hardware components of digital system. Since 217 has not been supported by the Government since the early 1990's, its information is also based on outdated data. The more recent evolution of 217, 217 Plus, is based on unverified data and is still confined to estimating the reliability of electronic hardware components.

- "Traditional Markov" models may be a good way of modeling fault interactions within a digital system, but it doesn't necessarily address in a satisfactory way the interactions between the functions of a digital system and the operational behavior of the "balance-of-system" and controlled equipment.

Comments on the objective of modeling and the establishment of criteria to judge the quality of modeling approaches and modeling results:

- Basic objectives of digital system risk modeling that appear to be realistically pursuable are:

  - Identification of significant digital system failure modes, with particular emphasis on those that are related to interactions between digital system software and controlled system functions, since these are often the most difficult to understand and uncover.

  - Identification of the type and degree of testing, explicitly including systematic software testing, that is needed to "bound" the level of risk contribution that can be expected from a particular digital system. This may not be as difficult to achieve as commonly believed to be, because software failures are usually triggered by the occurrence of specific system conditions, which in turn may occur with a certain frequency. The condition frequencies are equivalent to unconditional "hazard rate," which can be assessed independently from the software test process, whereas the actual software failure probabilities are conditional probabilities that can be determined by testing the software within the input space defined by each system condition of interest, i.e., if an input condition is expected to occur with a frequency of 10^{-3} per year, then it may be sufficient to "explore" the software input space defined by such a condition by means of random, but systematic, testing repeated a minimum of 1000 times without encountering a failure, in order to "bound" the risk level associated with the occurrence of that condition at an order of magnitude of 10^{-6} per year.

- With respect to the development of good criteria for evaluation of approaches to digital systems failure and risk modeling, the following consideration should apply:

  - A fundamental distinction needs to be made between establishing criteria to judge the quality and effectiveness of a modeling method, and criteria to judge whether one specific application of a method meets certain specific objectives. For example: fault tree analysis is sometimes used - without cut set quantification - to aid a system failure investigation process. It would erroneous to pick up one such fault tree analysis application and conclude that fault tree

analysis, as a method in general, fails to meet a criterion requiring risk quantification.

- Consistent with the above it would be appropriate and advisable for the BNL study to shift emphasis from evaluating specific past methodology applications to evaluating the suitability of specific aspects of a methodology to being applied effectively for the purpose of digital systems failure and risk modeling, according to foreseeable NRC regulatory evaluation needs.

- Reference in the above to evaluation of potentially useful aspects of a methodology, rather than a methodology as a whole, is not accidental. In fact mixing and matching particular features of different approaches to the needs of a particular type of application may be the best approach with the use of "traditional methods," which were not per se created to address the issue of digital system modeling and, therefore, cannot individually be expected to cover in an acceptable way all the many facets of the issue. For example, a traditional fault tree analysis may be adequate for the modeling of a relatively simple digital RPS logic, but a Markov model approach may be needed to address the fault handling features of a digital system software and redundant central processing unit (CPU) architecture.

– No matter what criteria are used, they should include the evaluation of methods in terms of whether they are effective at identifying and uncovering types of critical failure modes that have actually been observed in the operational experience of safety-critical digital systems.

  - In this respect, a categorization of types of systems in use and failures that have occurred should be adopted and/or developed and the suitability of methods to address the various categories should then be assessed.

Reviewer A also provided written comments after the meeting (see Attachment E).

Reviewer B

- The problem statement significantly lacks clarity (i.e., stable regulatory environment vs. trying to become risk-informed vs. ....). For example, are the criteria for the regulatory review of modeling? Hence, the conclusions of the report are not necessarily tied to the report's text or the regulatory premise (i.e., they are disjointed ideas).

  – Doesn't necessarily disagree with conclusions, but the text does not support them. In particular, conclusions about the methods are not supported.

  – The report starts by discussing regulatory items, and then moves into technical discussions.

- Each application reviewed using the report's criteria had different objectives. Hence, conclusions cannot be reached by comparing the extent to which the applications met the criteria.

- There is confusion between what is meant by traditional and dynamic methods that needs to be clarified.

- Other methods for reliability modeling of digital systems may exist in other industries, and modified versions of them might be used for systems in the nuclear industry.

- May want/need to pursue a "blended" approach with the best features of traditional and dynamic methods (or more "advanced" traditional methods).

- Sophisticated methods may not be necessary because there is no good data, anyway.

Reviewer C

- A link or relationship should be established between the criteria and other standards or procedures, such as the American Society of Mechanical Engineers (ASME) Level 1 PRA standard, and various guidelines on common cause failure (CCF).

- It is easy to implement modifications to digital systems, which can complicate modeling and quantifying data.

- Advanced reactors have many digital systems that perform control functions, as opposed to the RPS and ESFAS that are actuation systems. The former group needs detailed modeling, including considering many additional failure modes.

- Some modeling approaches reviewed may meet additional criteria, but it just wasn't documented.

- Some methods may only meet some of the criteria, but could play a role as part of the solution.

- The evaluation against criteria is limited by available information.

- A (quantitative?) method should be used for assessing software reliability. However, the resulting model does not have to be integrated with the overall PRA.

Reviewer C also sent written comments before the meeting (see Attachment F).

Reviewer D

- Hardware (HW) and software (SW) reliability cannot be evaluated separately, otherwise HW/SW interactions cannot be captured (philosophical issue).

- Traditional methods and the proposed draft criteria don't necessarily capture all of the Type I and Type II interactions.

- Type I - dependencies due to communication through the controlled/monitored process
- Type II - dependencies due to direct communication (e.g., networking, multiplexing, or hardware linkages)

- Fault tree/event tree (FT/ET) and Markov as "traditional" methods probably can't address all the issues of modeling digital systems
  - There probably is a need to add a "twist."
- Detailed models are needed. There is probably a need for different modeling methods for different applications. A graded approach should be used depending on system function (i.e., safety or control).
- Statistical dependencies between failure events may require Markov treatment.[2] FT cannot account for these dependencies. J. Dugan (University of Virginia) proposed a method that
  - Uses timed "AND" gates to model conditional occurrence of events given certain events have occurred
  - Doesn"t cover process interactions
- Dynamic methods may be needed to model communication, as well as dynamic methods for certain portions of the PRA, and then map them back into the PRA.
  - However, dynamic methods may not be necessary to model systems such as the RPS and ESFAS.
- [Reviewer A]: One can have a general framework using traditional criteria, and identify areas that require other approaches. Considering the issues related to Type I and Type II interactions is another way of looking at the different types and levels of digital system implementations that can be found in real life applications.
- [Reviewer B]: The complexity/accuracy of modeling may need to be driven by where data is available.
- The scarcity of probabilistic data is a big concern.
  - [Reviewer A]: Developing methods/models will point us to what data is needed,
  - [Reviewer A]: ...and also how to test the system, since testing is the only good source of data (as opposed to generic databases).

---

[2]After the meeting, Reviewer D sent the following statement: "...an explanation within the context of the relevant statement would be when the sequencing of events lead to different consequences. Then the consequences would be statistically dependent on the precursor events. For example, if an event B in the precursor sequence cannot occur before the previous event A occurs, then $P(B)=P(B|A)P(A)$. The standard ET/FT approach will not account for the conditional in $P(B|A)$ in the quantification process. However, Markov approach is not the only way such a dependency can be accounted for..."

- Uncertainties should be propagated through the model.
- Some statements and terms need to be more specific, e.g., what is meant by Markov modeling.

Reviewer D also provided written comments before the meeting (see Attachment G).

Reviewer E

- Cybersecurity also might be an issue that should be addressed in the criteria.
- Timing can be included in ETs/FTs through a phased-mission analysis.
- Dynamic FT gates can be used to capture dependency and timing.
  - Reviewer E showed a book that used Dugan's method. [M. Sonza Recorda, Z. Peng, and M. Violante, Editors, "System-Level Test and Validation of Hardware/Software Systems, Springer Series in Advanced Miicroelectroncs, 2005]
- MIL-HDBK-217 and PRISM have been superseded by 217 Plus.
- Even though an approach for software reliability analysis is included in NASA 's PRA procedures guide, NASA has not agreed yet on an approach for analyzing software reliability.
- NASA's approach (dynamic flowgraph methodology [DFM]) may not qualify as a traditional method.
  - Reviewer A disagrees for several reasons: a) the NASA approach is not based on DFM, but on traditional ET/FT modeling where possible, combined with traditional SW reliability estimation methods; b) the application example in the NASA PRA Procedures Guide shows a traditional ET/FT analysis of a Defense Meteorological Satellite Program (DMSP) satellite attitude control system in combination with the Schneidewind SW reliability estimation method; DFM is used as an example of what can be done when more detailed dynamic modeling is necessary; c) DFM itself is documented in at least four NUREGs and two NASA reports, dating back to the mid-nineties.
- Level of detail:
  - Needs to be tied to purpose of analysis.
    - This is discussed in the report but not listed in the final criteria.
    - The level of detail need not capture design features that could affect unreliability if sufficient data are available to bound unreliability and that is the output needed for decision-making.

- If system unreliability is dominated by components, such as circuit breakers and valves, there is no need to go to microprocessor level to ascertain it. .
    - System may also include operators who can over-ride failed digital controllers.
        - Controller failure ANDed with operator failure to over-ride.
    - This can limit level of detail needed, for example, need for controller FMEA.

- Software failures:
    - Cannot understand Criterion 3.2 for software (the reference [Chu, 2006b] is not available for review).
    - May not need logic models for software; depends on purpose of analysis.
        - It may be sufficient to bound software contribution.

- Modeling of dependencies:
    - Failure of communication network is important consideration.
        - Recent "data storm" at Browns Ferry is an example.

- Human errors:
    - Human reliability analysis also must consider operator recovery from failure of hardware/software.

- Probabilistic data:
    - Statement that digital hardware data are "scarce or non-existent" is probably too strong because some data are available.
    - Recent discussion with Honeywell (Netherlands) suggests there is a large amount of (non-nuclear) data on programmable logic controllers (PLCs).
        - HW and SW data, but the latter may not be applicable.
        - May need to analyze data for a particular application.
        - It is not clear whether these data are publically available.
    - Hardware data requirements should specifically address the Bayesian approach.
        - Use of "generic" data or allied-industry data as prior distribution.
        - Adjustment of data from other applications or environments.
        - Dealing with uncertain data (e.g., uncertainty in failure count).

- For software failure, testing data could be appropriately used in one of the software reliability growth models discussed in ANSI/AIAA Std. R-013-1992 and implemented in Computer Aided Software Reliability Estimation (CASRE) software.

Reviewer F

- Challenges associated with modeling digital systems:
  - software reliability
  - common cause failures (including software)
  - hardware/software interactions
  - failure data
  - interfacing digital system models into a PRA
  - time dependencies
  - diagnostics/fault tolerance/coverage
  - failure modes (including unknown or unforeseen failure modes).

- Challenges associated with developing a review process consistent with current regulations/guidance:
  - level of modeling detail
  - acceptance guidelines
  - PRA quality -attributes for digital system modeling
  - open issues - use of PRA with a deterministic defense-in-depth philosophy/methodology.

- The objective of the criteria is to support regulatory decision-making, e.g., a decision on giving credit to certain fault-tolerant features.

- It is helpful to link the criteria to standards, such as the ASME PRA standard.

- In establishing evaluation criteria, it is not as easy as saying one needs to be able to adequately model the unique aspects of digital systems:
  - Evaluation criteria must reflect both the characteristics of digital systems and how they are used in nuclear plants.
  - A method must be developed for categorizing digital systems.
  - The community needs to continually be looking at operational experience, for example:
    - Emergency Diesel Generator (EDG) load sequencer failure at Turkey Point, and
    - Data storm at Browns Ferry.

- Operational experience will affect how the evaluation criteria should be written. For example, the Category 1 criteria on level of detail currently include the phrases "design features that affect reliability" and "at the microprocessor level." System state and cross

system inter-connectivity as failure modes have been observed, and whether they should be explicitly part of the criteria should be considered.

- A method of categorizing digital systems may need to be developed to help determine the level of modeling detail needed.
  - Consider the need for additional Category 2 criteria for determining the level of detail to be used in identifying failure modes.
- In modeling software reliability, the concepts are often stated very differently than in traditional PRA terminology.
  - We should use terms that software people understand, or at least note the analogous terms (e.g., "operational profile" instead of "context" and "software-centric").
- The criteria need to be more consistent and/or may need to be applied in a particular order, for example failure modes (Category 2), before CCF modeling (Category 4), before level of detail (Category 1).
- In the evaluation criteria for human errors (5.1 and 5.2)
  - Criterion 5.1 needs to be reworked to include the way humans introduce faults into the software.
  - Man-machine interface (MMI), or more appropriately human-system interface (HSI), is outside the scope of the system model.
- Hybrid analysis methods should be considered for developing applications. For example, NASA's study of the International Space Station (ISS) used traditional FT/ET modeling for the most part, but Markov modeling for many digital systems.
- A possible definition of "traditional" method is one that is commonly used, well established, including large-scale applications.

A.3. TRADITIONAL METHODS AND THEIR APPLICATIONS

The discussions focused on three main issues:

- What is the definition of "traditional" methods, and therefore, what methods should be included in identifying and selecting "traditional" methods?
- Are the conclusions stated in the report clearly supported by applying the criteria presented in the report?
- Should the proposed criteria be applied to the methods, or to the models/applications?

The panel members' discussions on these three issues, as well as about alternative methods and applications, are summarized in the sections below.

A.3.1 Defining and Identifying "Traditional" Methods

Points of general agreement among the panel members: "Traditional" methods are difficult to define precisely. Separating methods based on "dynamic" versus "traditional" does not really help. Some methods can be considered traditional if they are used for a part of a model, but non-traditional if they are used for the whole model. Ultimately, binning traditional methods versus non-traditional methods includes subjective elements. Traditional methods will involve methods commonly used by the nuclear industry, since the NRC is interested in evaluating licensees' submittals. Therefore, traditional methods can be defined as:

- Methods that can be used in near-term (or somewhat near-term) to address at least some aspects of digital system reliability modeling and quantification.
- Methods that have had real-world application in the nuclear industry.
- Methods applicable to the kinds of decisions that the NRC will face.

Individual panel members had the following observations:

Reviewer F

The interactions and dependencies of digital systems may require methods that can overcome some of the limitations of traditional FT methods. Binary Decision Diagrams (BDDs) can help with coherence problems. Already, some applications of BDDs are used by the telecommunications and aerospace industries. In a Norwegian study, Dahll applied the method. Bayesian Belief Networks (BBNs) also may be helpful. Ali Mosleh developed a technique combining the BBN approach with BDDs. However, so far there are no large-scale applications of these methods to digital systems.

Reviewer E

Simulation methods, such as discrete event simulation (DES), also could be considered as traditional methods. In a 2003 Finnish paper, BBN was used to analyze the software of a relay. One could also question whether DFM qualifies as traditional. While the NRC has traditionally relied on FT/ET models for reactor safety analysis, a contractor for the Office of Nuclear Material Safety and Safeguards (NMSS) used discrete event simulation in a medical application.

Reviewer D

Traditional Markov techniques are not really that useful for evaluating digital systems. Depending on the definition of traditional methods, DES and BBNs could be considered as such. It is not clear that DES is practical. For simulation methods, the problems are that the sequence of failure modes cannot be captured, and integrating the results with a PRA is problematic. Dugan tries to capture data dependencies, employing a method she calls "Dynamic Fault Tree."

Reviewer C

The difference between traditional and dynamic methods is not clear, and it is hard to differentiate between them. For example, some methods would qualify as "non-traditional" if used to model an entire digital system, but would as "traditional" if used for specific aspects of

the modeling, such as BBN or testing or simulation with fault injection. The only method everyone agrees is clearly traditional is the FT/ET approach.

Reviewer B

EPRI supplies the risk and reliabilty (R&R) workstation that could be considered the most widely used software for reliability analysis. Methods included in the R&R workstation are currently traditional. However, new capabilities for the R&R workstation, such as BDDs and Declarative Modeling, will be released by the end of the year, and may include the ability to use phased-mission times. However, EPRI may not release this capability due to the concern that it may not be used properly, thereby distorting some plant's risk profiles.

Reviewer A

It is difficult to distinguish between traditional and dynamic methods. For example, BBN and Petri net methods can be considered traditional. Another example is using multi-value logic methods (predecessors of DFM), which have been employed in the chemical industry, but not for digital systems. It should be noted that reliability prediction methods are not really methods but a source of data. There is a NASA report[3] on an application of the software reliability quantification method described in the NASA PRA procedures guide. He elaborated on this subject in his written comments (Attachment E).

A.3.2 Conclusions Stated versus Applications of the Criteria Presented in the Report

Points of general agreement among the panel members: The report draws, or implies, conclusions about the models reviewed without knowing what original objectives the models were intended to satisfy. The report needs to further clarify that the models may not meet many of the criteria because they were not developed with the intent of doing so, but rather with objectives that may be quite different. Some models could have met more criteria if they had different objectives. The conclusions in the report should be more focused on the capabilities of the methods, not the capability of the models with respect to the criteria.

Individual panel members had the following observations:

Reviewer B

Some of the so-called "conclusions" in the report should be moved to the front to make the flow of the report more logical. The report reflects the order in which the work was done, but rearrangement could help with clarity, and produce a better report.

Reviewer A

The report should be portrayed as a demonstration of the criteria, not as judging specific methods or models. He elaborated on this subject in his written comments (Attachment E).

[3]This report is: "Risk-informed Safety Assurance and Probabilistic Risk Assessment of Mission-critical Software-intensive Systems," AR 07-01, ASCA, June 2007.

A.3.3 Criteria Applied to Methods or to Models/Applications

Points of general agreement among the panel members: It would be desirable to have criteria to evaluate methods. However, since a digital-system model could involve combining methods to address different aspects of the model, one ultimately needs to apply criteria to the modeling. The report's conclusions should be more focused on the methods' capabilities, not that of the models. Criteria could be applied against applications and this information used to evaluate methods. In any case, to proceed with discussing individual criteria, it was generally agreed that criteria should be viewed as being model/application-oriented.

Individual panel members had the following observations:

Reviewer B

The project is proposing which methods to pursue. It will be seen how well they meet the criteria, based on the evidence from the two test cases to which they are applied. The next step is extrapolation, i.e., to extend the conclusions from the test cases to reach general ones that can be considered generally applicable to analyzing digital systems. It will be necessary to support this extrapolation unless it is a straightforward inference, i.e., unless it is obvious that a particular method could meet a particular criterion if applied for that purpose - this requires a judgment about whether substantial additional work would be necessary for the method to meet the criterion.

Reviewer C

Some of the criteria presented are overly specific, such as requiring modeling the loss of HVAC ; they should be more general, not design-specific.

Reviewer E

Ultimately, one needs to apply criteria to "modeling," because the digital system model could involve a combination of methods to address its different aspects.

Reviewer D

If the capabilities/limitations of the methods are extrapolated based on the models, the work may be criticized as being speculative.

Reviewers A, B, and F

The criteria can be used to evaluate the methods, and the models then used as evidence of the methods' capabilities. Reviewer A elaborated on this subject in his written comments (Attachment E).

A.4. COMMENTS ON REVIEW CRITERIA

A.4.1 General Comments on Review Criteria

The following general comments were made about the review criteria:

- In general, the panel members felt that the criteria did a good job in covering the desired characteristics of digital-system reliability modeling.

- The state-of-the-art of reliability modeling cannot support all of the identified criteria, and additional research is needed, e.g., in reliability data, CCFs, and software reliability.

- It was recognized that the criteria have different levels of detail, degrees of specificity, and importance, and some criteria include not only review criteria but also background information. It is recommended that the criteria are made more succinct, and that supporting rationale, examples, and guidance on how to satisfy them is moved to the background discussion.

- Some criteria are similar to those in PRA standards, e.g., the ASME Level 1 PRA standard. Any relationship to the ASME standard should be stated.

- The criteria should be general without specifying the methods that should be used. For example, failure modes and effects analysis (FMEA) is only one of the methods used for identifying failure modes (others include hazard analysis, and hazard and operability study [HAZOPS]).

- A consistent terminology should be developed for all of NRC's projects dealing with digital systems.

The rest of this chapter summarizes the discussions on the categories of review criteria, in the same order they occurred during the panel meeting (Attachment H gives the order in which the review criteria categories/subcategories were discussed). For each criteria category, there is a brief description of the category, followed by any general comments agreed-upon by the members of the expert panel, and the comments from individual experts.

A.4.2 Category 1: Level of Detail of the Model

While the criteria in the other categories represent a collective set of criteria for evaluating digital-system reliability models, the three criteria in this category are mutually exclusive ones at different levels of detail (i.e., a model would only be expected to meet one of the three criteria). Criterion 1.1 represents the ideal level of detail of a model, Criterion 1.2 represents the level of detail that the authors believe is reasonably achievable, and Criterion 1.3 represents the level of detail of the digital-system models included in the design certification PRAs for new reactors (i.e., AP1000 and the Economic Simplified Boiling Water Reactor).

The panel thinks that the level of detail of modeling should depend on the study's objective, and recommended that the existing proposed criteria in this category be replaced by the following alternative criterion: "Modeling should reflect all significant failure modes (functional and

physical), be developed to the level of detail of supporting information, and provide output needed for risk-informed decision-making."

The comments of individual panel members are provided below:

- [Reviewer A]: Criterion 1.2 makes an assumption and is problematic. It should instead indicate that the model should address both the physical and functional characteristics of a digital system, e.g., the timing of a CPU failure can affect the type of impact the failure may have at the plant level.

- [Reviewer B]: Criterion 1.1 is the only criterion; Criterion 1.3 is an exception, and Criterion 1.2 is a very specific criterion that can conflict with Criterion 1.1. In Criterion 1.3, "...can adequately support the objective of the modeling" should be replaced with "...capture all dependencies including software."

- [Reviewer C]: The final criterion should address "operational and functional characteristics". Since a circuit board may perform several functions, physical and functional features cannot be separated.

- [Unknown]: It is necessary to consider the difference between "functional" and "physical" failures.

- [Reviewer E]: An alternative criterion should be used, based on the panel discussion: "Modeling should reflect all significant failure modes (functional and physical), be developed to the level of detail of supporting information, and provide output needed for risk-informed decision-making."

A.4.3 Category 2: Identification of Failure Modes of the Components of Digital Systems

BNL indicated that from their experience it is very difficult to undertake an FMEA of digital systems and little guidance is available. For example, what is the level of detail at which an FMEA should/can be performed (subject to limitations on design detail and knowledge)? Are the failure modes realistic and complete? For example, can an output bit being stuck high be an isolated failure mode, knowing that the bit is physically connected to other parts of the system?

The panel thinks that a criterion should not advocate a particular method, i.e., FMEA, and that other methods also can be used, e.g., HAZOPS. The title of the category should be changed to include "components of" before "digital system." An alternative to Criterion 2.1 was proposed: "A technique for identifying failure modes of the basic components of a digital system, and their impact on the system, should be applied." The discussion about Criterion 2.3 generated a recommended new criterion that the "...failure modes that have occurred in operating experience should be examined." For example, important software failures have occurred as a result of problems with requirement specifications. The panel also recommended rewording Criterion 2.3 and including it as a sub-bullet to the new criterion.

The comments of individual panel members follow:

- [Reviewer D]: By design, FMEA is intended to identify immediate impacts, not to model fault propagation. In the nuclear field, FMEA is a precursor to fault trees, i.e., it considers the immediate impact of component failure, not the systemic impact. Defining failure mode according to functions might be wrong. Criterion 2.1 may be neither feasible nor necessary. Arbitrary output as a failure mode should be considered.

- [Reviewer A]: An FMEA (more so when applied in FMECA - Failure Modes, Effects and Criticality Analysis) normally considers the effects and consequences of a postulated failure. Military Standard 1629 and Handbook 338b provide guidance on FMEA.

- [Reviewer C]: FMEA should be related to deterministic criteria; other methods can be used to identify failure modes, e.g., hazard analysis and HAZOPS.

- [Reviewer F]: Standard Review Plan (SRP) Chapter 7 addresses Criterion 2.3. Failure modes should not be screened at this stage because their effects in combination with other failures have not been identified. The model should allow the possibility of design errors.

- [Reviewer E]: FMEA usually is done by "designers," and is a good starting point for system analysis. Fault trees can be used to model multiple failures.

- [Reviewer B]: The industry performs FMEA routinely, and General Public Utilities has guidance on FMEA that is not specific for digital systems. Criterion 2.3 should not be here.

- [Reviewer F]: Operational experience suggests that digital systems are vulnerable to faults associated with diagnostic features.

A.4.4 Category 3: Software Failures

Criterion 3.1 suggests that the contribution of software failures can be considered using either a "software-centric" or "system-centric" approach, while Criteria 3.2 to 3.4 apply only to the "software-centric" approach. Criterion 3.2 associates the occurrence of software failures with that of triggering events, and requires a model of software failures that is consistent with this concept. Criterion 3.3 suggests separately considering the application software and support software, including the operating system and platform software. Criterion 3.4 emphasizes the importance of exploring the context wherein a piece of software is challenged, and states that a quantitative software reliability model should be able to account for different contexts.

The panel discussed Criterion 3.2 extensively, and agreed that the term "triggering event" must be explained in the background discussion. The panel also agreed that separating application and support software, as indicated in Criterion 3.3, is important because of their differing amounts of operating experience. In addition, the panel recommended listing Criteria 3.2 and 3.4 next to each other.

The comments of individual panel members are given below:

- [Reviewer B]: Many criteria are overly wordy, i.e., they should simply state the criterion, and the justification or "how to" should be moved to the background discussion.

- [Reviewer F]: The terminology of "system-centric" and "software-centric" can lead to misunderstanding.

- [Unknown]: Criterion 3.2 is too specific on "how to." There are other approaches for quantifying software reliability, such as parametric and non-parametric methods. The way Criterion 3.2 is stated suggests there are hidden assumptions pointing at a certain direction, and should be removed. Instead, it should state that the model needs to articulate how it arrives at its failure rates or probabilities.

- [Reviewer A]: Regarding Criterion 3.2, the software of a control system may have a conditional failure probability that if combined with a rate of occurrence of some condition becomes a failure rate.

- [Reviewers D and F]: Criterion 3.2 could be worded "...don't consider software failures in the abstract, consider them in the context of the system."

- [Reviewer F]: Regarding Criterion 3.2, there are other ways to get software failure besides just "triggering events" (e.g., bit drops, specification errors, hardware/software interface errors).

- [Reviewer F]: Criterion 3.3 should include the interactions between applied software and support software, and between software and hardware. Due to the potential for CCF, there may be a need to consider some software development tools that generate application software. Commercial off-the-shelf (COTS) and communication software should be considered.

- [Reviewer C]: Backbone (support) software and software development tools should be considered.

- [Reviewers A and F]: Criterion 3.4 is just an extension of Criterion 3.2.

- [Reviewer B]: Criterion 3.2 is on triggering events, and Criterion 3.4 is on functions; both are needed.

A.4.5 Category 4: Modeling of Dependencies

The criteria in this category consider different types of dependencies that should be accounted for in developing a reliability model of digital systems.

The panel agreed that Criterion 4.2 (CCFs) is a "catch-all" bin for dependencies that are not explicitly modeled; therefore, it should be moved to the end of this category and will be discussed last.

The comments of individual panel members are shown below:

- [Reviewer B]: The criteria essentially are a list of dependencies. One doesn't need to specify separate criteria for each dependency (that wasn't done elsewhere, e.g., specifying devices). Reviewer B recommends including the dependencies as a bulleted list instead of as separate criteria.

- [Reviewer F]: The subcategories are "uneven."

- [Reviewer D]: Type I dependencies (interactions with physical processes) are not clearly captured. They should be added, maybe under Subcategory 4.5. The time constant of the system changes, and the response times differ. The coupling (with physical processes) may become much tighter. Uncertainty in discrete sampling times may lead to system failure, even if the Nyquist criteria are satisfied; this is unique to digital systems.

- [Reviewer C]: Self-testing differs from other fault-tolerant features, and should be modeled separately (though not necessarily in this category).

- [Reviewers B and E]: The dependency breakdown may be too fine, and may result in the summation of a large set of overly conservative (uncertain) values. It may be preferable to lump more of the individual dependencies into the "catch-all" CCF bin, based on current state-of-knowledge.

- [Reviewers A, B, and D]: It is only possible to model to the level of the state-of-knowledge. However, this is an evolving boundary because there even is uncertainty about what is the current state-of-knowledge.

- [Reviewer F]: It is important to explicitly model aspects of the system that risk-informed applications are trying to address; therefore, many of these dependencies may need to be explicitly modeled.

A.4.5.1 Subcategory 4.1: Communication Networks/Buses

Components of digital systems are interconnected through buses, hardwired connections, and communication networks. Through the connections, information is exchanged and used in calculations and decision-making. The criteria in this sub-category address modeling of failures associated with the connections of components of digital systems at different levels (e.g., intra-channel communications and inter-channel communications).

The panel agreed that the Browns Ferry data storm incident is a good example indicating the importance of modeling communication-related failures.

The comments of individual panel members are provided below:

- [Unknown]: The lengthy explanation should be eliminated from Criterion 4.1.1 to make it more succinct.

- [Reviewer B]: The criteria should be kept broad so people do not leave things out.

- [Unknown]: A distinction should be made between the four criteria in this sub-category.

- [Reviewer E]: The meaning of communication network is unclear.

- [Reviewer B]: Any available operating experience should be included in the discussion, but it must be emphasized that examples and operating experience are not the "end all and be all."

- [Reviewer E]: It is not clear what is really meant by "failure of the communication network."

A.4.5.2 Subcategory 4.3: Support Systems

The criteria in this subcategory are applicable to support systems that are shared by the components of the digital system and the rest of the plant, as modeled in a PRA. They ensure that the dependencies are properly accounted for when the model is integrated with the PRA.

There are no general comments on the overall category. The comments of individual panel members appear below:

- [Reviewer F]: Power quality and power availability issues also should be considered in the criteria of this subcategory. Other issues in 50.49 or RG 1.97 and the issue of radio-frequency interference should also be included. A distinction should be made between the support systems that should be modeled and the data that capture the impacts of support systems.

- [Reviewer B]: These criteria are an example of being too specific. They imply that the only support- system dependencies that need to be addressed are the two specific dependencies identified in them. It is better to have general criteria and provide examples in the discussion, making it clear that these are only representative examples.

- [Reviewer D]: The HVAC in Criterion 4.3.2 should be generalized as "operating environment."

- [Reviewer A]: Developing overly specific criteria makes it very unlikely that any application can address them all.

A.4.5.3 Subcategory 4.4: Sharing Hardware

This criterion is intended to emphasize that some digital systems may be implemented by running different software on the same digital hardware, and this dependency has to be modeled.

There are no general comments on the overall category. The comments of individual panel members follow:

- [Reviewer F]: This subcategory should focus on shared digital components, i.e., chips. There should be more discussion in the rationale on what is meant by shared hardware (e.g., sensors, multiplexers, and voters) that should indicate what types of dependencies

are addressed, and which ones must be modeled explicitly. It is also important to consider specific possible regulatory decisions about putting RPS and ESFAS on the same hardware platform, or assuming what systems fail in the diversity and defense-in-depth analysis.

- [Reviewers A, D, and F]: The statement on RPS/ESFAS modeling in Criterion 4.4 should be removed. The phrase, "e.g., by linking fault trees" is too specific, and should also be removed from this Criterion.

- [Reviewer D]: Another type of sharing hardware could be the data sharing by processes of software. Two processes might try to access data in a storage device simultaneously, causing data race and system lock-up.

A.4.5.4 Subcategory 4.5: Interactions of Digital Systems with Other Systems

The intent of the criteria is to ensure that the interfaces between the digital system and the rest of the plant are properly accounted for.

There are no general comments on the overall category. The comments of individual panel members are provided below:

- [Reviewer D]: Type I interactions, i.e., devices communicating through the controlled/monitored process, are not captured by the criteria in this subcategory. An additional criterion that reflects this interaction might be needed.

- [Reviewer F]: Loosely coupled dependencies are also not captured by this subcategory criteria; an example is a low reactor pressure signal for reactor trip. Definitions of loosely coupled dependencies and tightly coupled dependencies are given in C. Perrow's book [Normal Accidents, Living with High-Risk Technologies, Princeton University Press, Princeton, New Jersey (1999)] or Steven Arndt's paper ["Development of Regulatory Guidance for Risk-Informing Digital System Reviews," NPIC&HMIT 2006].

- [Reviewer B]: The meaning of "incorrect sensor input" in Criterion 4.5.1 is not clear. Generally, the sensor inputs go to digital systems only, not to other components or systems.

- [Unknown]: The meanings of systems and channels are not clear. It may be better to use systems and sub-systems.

The following comments were made on Criterion 4.5.2 that addresses the modeling of voters and other logic devices:

- [Reviewer F]: It is too specific and should be either dropped or modified.

- [Reviewer D]: It should be modified to be made more general by removing the reference to specific devices.

- [Reviewer C]: It has already been covered and can be dropped.

- [Reviewer B]: It should be dropped.

- [Reviewer A]: It should be modified.

- [Reviewer E]: It should either be dropped completely or included in the supporting rationale.

A.4.5.5 Subcategory 4.6: Modeling of Fault Tolerance Features

The criteria are based on known weaknesses of models of digital systems, and intended to ensure that they are avoided. Criterion 4.6.1 considers identifying the failure modes that can be detected. Criterion 4.6.2 concerns the potential of doubly crediting fault coverages. Criterion 4.6.3 considers modeling the dependency on fault tolerant features. Criterion 4.6.4 relates to modeling failures to properly cope with detected failures.

There are no general comments on the overall category. Individual panel member's' comments are given below:

- [Reviewer F]: Fault tolerance includes design and tests for fault tolerance. A high-level criterion should state that fault tolerance design features should be modeled explicitly, including all ways in which they function, and both their potential positive and negative impacts. Some specific information can be included in sub-bullets and sub-criteria.

- [Reviewers C and F]: Fault identification features and continuous self-test features should be distinguished.

A.4.5.6 Subcategory 4.2: Common Cause Failures

The criteria specify that hardware and software CCF should be modeled within a channel if redundancy exists, between redundant channels, and between digital systems. It also was pointed out to the panel that some people from industry are claiming that "...digital hardware failures often can automatically be detected and possibly alarmed; therefore, it is very unlikely that two hardware failures take place at exactly the same time."

The panel recommended that "identical" in Criterion 4.2.2 should be changed to "common," or "similar," or "very similar." Also, there should be more discussion in this subcategory on its "residual" nature. An example statement is "The previous criteria in this category addressed dependencies that should be explicitly included in this model. This criterion is intended to address other potential dependencies that are not explicitly modeled."

The comments of individual panel members are provided below:

- [Reviewer A]: "Intra-system" and "inter-system" should be combined into the same criterion.

- [Reviewer E]: Consideration of inter-system CCF may be pushing the envelop beyond the state-of-the-art, since these types of failures are not currently included in other system models in PRAs.

- [Reviewer B]: It may not be credible for two different digital systems to be subjected to the same trigger mechanism.

- [Reviewer E]: If someone tries to model inter-system CCF for digital systems, there will be no data. They will generate a conservative model that might distort the plant's actual risk profile.

- [Reviewer B]: The inclusion of the inter-system CCF criteria could entail spending much money for analysis and possibly redesign, all for nothing.

- [Reviewer F]: Questionable criteria should be included for now, and can always be eliminated later.

- [Reviewer A]: The timing aspects of potential common cause failures may preclude the concern over inter-system CCF.

- [Reviewers A and B]: It makes sense to include exceptions or "wriggle-room" in the criteria, e.g., including the phrase "...or else demonstrate that it does not have to be included."

- [Reviewer F]: We may want to say "should be considered" instead of "should be modeled."

A.4.6 Category 7: Probabilistic Data

Probabilistic data for both hardware and software failures are needed to quantify reliability models of digital systems. While component-specific data are preferred, generic data also can be used if there are no specific data, and the generic data are properly collected. The same consideration applies to CCF parameters and estimates of fault coverage.

There are no general comments on the overall category. Individual panel member's' comments follow:

- [Reviewer A]: Regarding Criterion 7.1, it is very unlikely that truly component-specific data will ever be available at the desired level of statistical confidence and fidelity, due to the short design cycles of digital hardware components (i.e., digital systems are upgraded very quickly).

- [Reviewer B]: Many of Criteria 7.1 - 7.10 (for hardware data) are good practices, but not necessarily qualified to be criteria. Thus, Criterion 7.8 about the CCF data is particularly problematic since generally plant-specific CCF data cannot be obtained.

- [Reviewer F]: Regarding the criteria in this category, referring to the good practices of the ASME PRA standard or the data handbook is suggested. However, unique practices specifically related to digital systems should be emphasized. An example is whether the data for the same card, or similar versions of cards, are used in the reliability model, and whether the difference in data sources matters.

- [Reviewer E]: Replacing "data" with "information" is suggested. It would be good to provide some guidance on how to use generic data (e.g., from Military Handbook 217), and on how to account for uncertainty.

- [Reviewer B]: EPRI can talk with new reactor vendors about starting an effort to build a digital instrumentation and control (I&C) database.

- [Reviewer D]: Criterion 7.11, "quantifying the software failure probabilities" suggests using a software-centric model, but this does not necessarily have to be the case.

- [Reviewer F]: Some of the intrinsic issues in Criterion 7.11 associated with software reliability data, should be highlighted, such as software revisions, software evolvement (e.g., conglomerating operational data, test data), and other issues well-known in the software- reliability community.

- [Reviewer D]: Criterion 7.11 can be reworded to "A method for quantifying the contribution of software to digital system reliability should be used and documented."

- [Reviewer F]: Since Criterion 7.11 is data-related, the word "contribution" might not be enough, and this criterion may need re-wording.

- [Reviewer E]: Having to come up with a value for failure probability is a concern. An alternative choice could be "bounding" the failure probability instead of "quantifying."

- [Reviewer A]: An encouragement to develop quantitative estimates of the failure probability of digital systems is appropriate, otherwise everyone will stick with the current status quo, by which arbitrary assumptions of perfect digital system reliability are often made without justification.

A.4.7 Criteria 8.4 and 8.5: Uncertainty

Uncertainty criteria include both model uncertainty and parameter uncertainty. With the state-of-the-art in modeling digital systems, it is important to consider uncertainty.

There are no general comments on the overall category. The comments of individual members are provided below:

- [Reviewer F]: The criteria could refer to some literature on uncertainty in software modeling, which can be provided after the meeting.

- [Reviewer B]: For Criterion 8.4, it is not clear how many and which alternative assumptions should be discussed and documented. In some EPRI guidance, only those alternative assumptions that impact the CDF by a factor of two or more are required to be documented. Another issue is how to conduct the uncertainty analysis.

- [Reviewer B]: For Criterion 8.5, the point estimate may be the only value used in many applications. Therefore, propagation of uncertainties is not necessarily possible in these instances.

- [Reviewer F]: There is an Office of New Reactors' (NRO's) project on using sensitivity studies for digital system screening criteria that may be available by the end of this summer.

A.4.8 Category 6: Ease of Integration with a PRA Model

Most PRA models are built using the fault tree/event tree method. Thus, it is desirable to build the reliability model of a digital system in such a way that it can be integrated into the existing PRA framework.

There are no general comments on the overall category. Below are the comments of individual panel members:

- [Reviewer D]: There is a procedure for converting Markov model results to cutsets that can be implemented in software.

- [Reviewer F]: Criterion 6.1 should be modified. An alternative criterion for it is, "For the digital system reliability model to be fully effective, it should be possible to integrate it into the plant PRA model. The process for integrating the model should be relatively straightforward so that it can be mechanized through software and can be easily verified."

- [Reviewer D]: The title of this category is questionable. The ease of accomplishing something is subjective and differs from person to person. "Ease" should not be considered a criterion.

- [Reviewers A and E]: Both suggest removing "ease" from the title and criterion.

- [Reviewer F]: The word "ease" can be removed from the title and criteria, but the concept of "ease" should be included in the discussion or rationale. The discussion should also address the issue of the process being "mechanized," as mentioned in his previous comment above.

- [Unknown]: In the discussion, refer back to all of the modeling features that were previously described.

- [Reviewer C]: The digital system reliability model should be compatible with the PRA model, i.e., should avoid other means of arriving at a likelihood of digital system failure that may not be compatible with the established PRA framework.

- [Reviewer C]: Another aspect of integration is that the failure of a digital system may generate an initiating event with possible additional failures of mitigation features. This should also be integrated with the PRA model.

A.4.9 Category 5: Human Errors

Generally, human errors related to digital systems can be treated in the same way as analog systems. They mainly are due to two factors: errors introduced during upgrading digital systems, and errors related to the MMI.

There are no general comments on the overall category. The comments of individual panel members are set out next:

- [Reviewer F]: Criterion 5.1 is fine. It is probably worthwhile to add some caveats or sub-bullets to illustrate it. In Criterion 5.2, MMI should be replaced with HSI (human-system interface). However, HSI issues are beyond the scope of modeling digital system reliability, except for the dependency of human reliability analysis (HRA) on the state of the digital systems.

- [Reviewer D]: You cannot model the human errors associated with a digital feedwater control system in a traditional PRA because there are so many interactions.

- [Reviewer C]: An HRA associated with digital I&C is not simple, especially recovery actions. Manual actions depend on the design. From an MMI perspective, consideration should be given to how the operator will recognize the situation with a particular set of process signals.

A.4.10 Criteria 8.1 - 8.3: Documentation and Results

These criteria consider documentation of key assumptions and results.

There are no general comments on the overall category. The comments of individual panel members are provided below:

- [Reviewer B]: In Criterion 8.1, a qualifier should be applied to the word "assumptions" as a condition for needing to be documented, e.g., "key" or "unique" assumptions related to digital systems. Authors may refer to the ASME PRA standard or RG 1.200 errata, which will be available soon, on uncertainties and assumptions.

- [Reviewer F]: In Criteria 8.1 and 8.2, the term "logic model" is too specific and should be replaced with "model."

A.5. CONCLUDING REMARKS

Two members said that they discussed everything they wanted to at the meeting, and had nothing more to add. The concluding remarks of other members are summarized below:

Reviewer B

The report is a good product and contains valuable information. Its purpose should be further clarified. It is necessary to make sure that the terminology and criteria is consistent through the whole report. Another issue for clarification is whether the criteria were developed to evaluate the modeling or the methods, as this will affect the conclusions and recommendations on method selection presented in the report, which are not supported by the text. The report should better capture the authors' philosophy and intent, since such knowledge might lead readers directly to the criteria. Sometimes, this is not always given in the report and the criteria are not easy to follow. It is also recommended that criteria be consistent with the Level 1 PRA standard. Generally, data issue and the treatment of software are big challenges, and are not

expected to be resolved in this document; however, they should be considered. A good starting point on the issue of data is to collect it from other industries or sources named by panel members. Currently, the failure modes of digital systems are based on FMEA or operational experience. It is desirable to have a means to systematically identify the failure modes of digital systems.

Reviewer C

Although the report mentioned that the criteria were developed for specific applications, Chapter 4 in this report is trying to evaluate methods. It is not clear how to fix this, but it definitely needs some clarification. Also, the report clearly suggests that some criteria are mutually exclusive. However, this is not discussed in rating the individual methods in Chapter 4. It is expected to have certain impacts on the rating. Rating scores should be downplayed because this information is misleading. Readers might think something is wrong with digital systems that already have been used in the nuclear industry since so many criteria cannot be met, as indicated in the report.

Reviewer F

Other potential modeling methods should be articulated somewhere in the report, though it is not necessary to carry them throughout the report. The inclusion is suggested of a discussion of why some methods, such as DES, BBN, DFM, and various hybrid methods were selected for assessment in this report. The report needs restructuring so that model evaluation can be considered as supporting evidence of method evaluation, as previously suggested. Terminology and assumptions need to be better standardized for the NRC programs. A convergence of the philosophy and terminology between the PRA-world and the software-world also is desired; this is a bigger issue than this project. Operational experience, knowledge of how systems work, and their characteristics should be highlighted as part of the strategy for developing criteria.

Reviewer A

The purpose and product of the report should be restated in a different light. After discussion, it is clear that the real objective is to arrive at criteria that could evolve into regulatory review criteria and to try to match these to what is judged feasible. This has not been articulated clearly in the report. Some better words in writing will be provided. There exists the appearance of potential conflicts between some of the current criteria. While they might not be actual conflicts, in certain cases some criteria should include a supporting rationale to explain how they complement other criteria, e.g., that the known CCF mechanisms complement the unknown CCF. It is not clear whether this applies elsewhere in the report, but a systematic review might identify these cases (e.g., using a Venn diagram).

Attachment A

Agenda of External Review Panel Meeting on Selection of Traditional Methods for Reliability Modeling of Digital Systems

Brookhaven National Laboratory
May 23-24, 2007

Day 1

Start Time	Duration (minutes)	Topic	Speaker
8:30	5	Welcome	Lehner, BNL
8:35	45	Background, overview, and objectives	Kuritzky/Siu, NRC
9:20	70	Presentation of preliminary comments by each member of the panel (10 minutes per member)	Members
10:30	15	Break	
10:45	75	Discussion on identifying "traditional" methods and their relevant applications	Panel
12:00	60	Lunch break	
1:00	120	Discussion on each review criterion: 1. Modification/deletion/addition 2. Limitations of the state-of-the-art and recommendations for additional research	Panel
3:00	15	Break	
3:15	105	Discussion on each review criterion: 1. Modification/deletion/addition 2. Limitations of the state-of-the-art and recommendations for additional research	Panel
5:00		Adjourn for the day	

Attachment A (Cont'd)

Agenda of External Review Panel Meeting on Selection of Traditional Methods for Reliability Modeling of Digital Systems

Brookhaven National Laboratory
May 23-24, 2007

Day 2

Start Time	Duration (minutes)	Topic	Speaker
8:30	90	Discussion on each review criterion: 1. Modification/deletion/addition 2. Limitations of the state-of-the-art and recommendations for additional research	Panel
10:00	15	Break	
10:15	105	Discussion on each review criterion: 1. Modification/deletion/addition 2. Limitations of the state-of-the-art and recommendations for additional research	Panel
12:00	60	Lunch break	
1:00	60	Discussion on each review criterion: 1. Modification/deletion/addition 2. Limitations of the state-of-the-art and recommendations for additional research	Panel
2:00	45	Concluding remarks	Members
2:45	15	Next steps and action items	NRC/BNL
3:00		Adjourn	

Attachment B

List of Documents Sent to Panel Members

1. Brief summary of the USNRC Office of Research's digital risk research program.

2. External peer review meeting description and agenda.

3. Executive summary and Chapter 6 of the National Research Council report (The full report is available at http://www.nap.edu/catalog.php?record_id=5432).

4. Paper by Arndt, Siu, and Thornsbury on "What PRA Needs from a Digital System Analysis" from PSAM6.

5. Draft BNL letter report: T. L. Chu, G. Martinez-Guridi, M. Yue, and J. Lehner, "Probabilistic Modeling of Digital Systems at Nuclear Power Plant: Traditional Methods Selection," Brookhaven National Laboratory, Draft Letter Report, April 2007.

Attachment C

Expert Panel Meeting Attendees

Expert Panel Members (in alphabetical order):

Tunc Aldemir (Ohio State University)
Steven Arndt (USNRC)
Ken Canavan (Electric Power Research Institute)
Sergio Guarro (ASCA)
Dana Kelly (Idaho National Laboratory)
Taeyong Sung (Canadian Nuclear Safety Commission)

Facilitator:

Nathan Siu (USNRC)

NRC Project Manager:

Alan Kuritzky

BNL Authors:

Tsong-Lun Chu
Gerardo Martinez-Guridi
Meng Yue
John Lehner

Attachment D

Biographies of Panel Members

Tunc Aldemir

Tunc Aldemir received his PhD in nuclear engineering from the University of Illinois and is a Professor of Nuclear and Mechanical Engineering at The Ohio State University. His broad area of specialization is nuclear reactor safety. His research in reliability and probabilistic risk assessment (PRA) focuses on systems that may be difficult to model using conventional techniques. He has published more than 80 refereed articles on dynamic methodology development for the reliability modeling of such systems.

Currently, Dr. Aldemir is involved in developing methodologies that will allow quantifying the risk impacts of upgrades of the digital I&C system in nuclear power plants and in developing computational tools to automate Level 2 PRAs and perform seamless Level 1-2-3 PRAs. He is a Fellow of the American Nuclear Society and on the editorial board of Reliability Engineering and System Safety.

Ken Canavan

The bulk of Mr. Canavan's 20 plus years of experience is in application of risk technology with the utility sector of the nuclear power industry. Mr. Canavan began his nuclear career at Toledo Edison's Davis-Besse nuclear power station, where he was involved in all aspects of the development of the plant specific Probabilistic Risk Assessment (PRA). At GPU Nuclear, Mr. Canavan was a lead risk analysis engineer working on the Three Mile Island and Oyster Creek nuclear generating station risk management programs.

Following consultant experience as Manager of Risk Analysis for Data, Systems and Solutions SAIC and a Supervisor at ERIN Engineering, Mr. Canavan joined the Electric Power Research Institute (EPRI). Mr. Canavan is currently the Program Manager of the Risk and Safety Management (RSM) and Nuclear Asset Management (NAM) programs.

Mr. Canavan's experience is primarily in the area of risk technology and safety analysis. Specialty areas include methodology and tools development and unique applications of risk technology. Over his 20+ year of service, Mr. Canavan has participated or led the peer reviews of approximately a dozen large scale applications of risk technology within the nuclear and aerospace industries.

Sergio B. Guarro

Dr. Sergio Guarro is a Distinguished Engineer in the Systems Engineering Division (SED) of The Aerospace Corporation in El Segundo, California, a non-profit corporation that operates a Federally Funded Research and Development Center (FFRDC) supporting the acquisition of all U.S. Government military and reconnaissance satellites. He is also the founder and Chief Scientist of ASCA Inc., a small company dedicated to risk and systems engineering research in

the nuclear and space systems application arenas. At The Aerospace Corporation Dr. Guarro has held several management positions, including those of Manager of the Reliability and Risk Assessment Section in the Electronic Systems Division and of Director of the Risk Planning and Assessment Office in SED.

Dr. Guarro has pioneered research in the specific area of software safety and software risk, with developments and publications dating back to the mid 80's and early 90's. He has also authored the chapter of the NASA Probabilistic Risk Assessment Procedures Guide dedicated to software and software intensive space systems. Dr. Guarro has developed over his career broad expertise and experience in the development of systems engineering, risk management and mission assurance disciplines and techniques, and their application to complex systems such as nuclear power plants and orbital or planetary spacecraft. He has developed risk and safety assessment methodologies for launch and space systems, such as the one adopted for the launch approval of the NASA Cassini mission, and has served on National Research Council committees as an expert panelist for space systems risk and safety assessment. He provides leadership in establishing and disseminating space systems risk management and mission assurance best practices in the National Security Space (NSS) and NASA communities.

Dr Guarro has authored and has been the co-editor of technical textbooks, and has published close to eighty papers in refereed journals and conference proceedings. His latest work in the more general area of mission assurance is documented in the Aerospace Corporation Mission Assurance Guide, which has recently been published and distributed in the space systems community.

Taeyong Sung

Taeyong Sung is a probabilistic and safety assessment (PSA) and reliability technical specialist in Canadian Nuclear Safety Commission (CNSC).

He began his PSA career from 1989 in Korea Atomic Energy Institute with B.S. and M.S. degrees in nuclear engineering from Kyung Hee Universities in Korea. Since then he performed various areas of Level 1 PSA including digital I&C system reliability analysis.
He performed digital I&C system analysis for CANDU reactors as well as PWRs and leaded research projects to develop a PSA methodology for digital I&C system in NPPs for years in Korea.

In 2002, he joined Atomic Energy Canada Limited in Canada and he has worked in CNSC since 2003. He is reviewing various risk and reliability analyses and developing regulatory documents and involving a research project for digital I&C system quantitative analysis.

Biographies of the other panel members are not available.

Attachment E

Written Comments Provided by Reviewer A

Review Comments on Brookhaven National Laboratory Draft Letter Report "Probabilistic Modeling of Digital Systems at Nuclear Power Plants: Traditional Methods Selection"

1. Introduction

This document provides comments that address key issues concerning the subjects covered by the BNL (Brookhaven National Laboratory) cited in the title, as well as related issues that emerged at the expert panel meeting held at BNL on May 23 and 24, 2007.

The intent of the comments provided is to assist the authors of the BNL report, who are in the process of completing and improving its content before publishing it in its final version. The subjects discussed hereinafter are a selected subset of topics that have been addressed verbally at the above-mentioned expert panel. However, this subset is covered again here in a more in-depth and organized fashion, primarily because it includes topics that the reviewer believes to have special relevance and/or have not been fully addressed at the meeting. Specific recommendations have been formulated and are offered in this review, as a possible solution for the most pressing and critical issues associated with the reviewed subjects.

The following primary areas of the BNL draft letter report are addressed in the following:

- Report objectives and their reflection in the report contents
- Report review of NASA reliability models
- Report conclusions and recommendations

Comments on the criteria for evaluation of digital systems models are not included in this review, because they were the more specific target of the expert panel discussions held at BNL on May 23-24, 2007. As such, they are extensively and more than sufficiently addressed by all the comments provided by the panel experts, and these comments are well documented on the BNL written compilation of the comments.

The principal findings of this review and the key recommendations that these findings suggest are summarized upfront, along with pointers to the sections of the comment text that provide the supporting rationale. Each section also lists at the end the specific recommendations that pertain specifically to the subjects discussed in that section.

2. Summary of Findings and Recommendations

The main findings of this review and related recommendations are summarized in the following. The section number identified after each finding or recommendation refers to later sections of this review where the reader can find a more detailed explanation of the finding/recommendation itself.

Findings:

1. A key objective of the BNL study is the identification of criteria for evaluation of future digital systems risk models and assessments in support of regulatory decision processes (Section 3).

2. The distinction between "traditional" and "advanced / dynamic" methods is not always clear (Section 3).

3. The Report objectives that refer to the evaluation of "models" and "methods (2 and 3 in BNL report Section 1.1) lead to practical contradictions both within the report evaluations and conclusions and with respect to what emerged in regard during the expert panel discussions on May 23 and 24, 2007 (Section 3).

4. The BNL reviewers have not unable, for lack of information or other factors, to develop a good understanding of the NASA "conditional risk method," both as a framework for digital systems and software integration into PRA, and in terms of the application examples contained in the NASA PRA Procedures Guide (Section 4).

5. Some of the conclusions of the BNL report concerning "methods" do not seem to be supported by the evidence gathered in the course of the evaluations conducted on the "models." Other methods appear to have been excluded from consideration primarily because examples of application (i.e., "models") were not easily accessible by BNL (Section 5).

6. The BNL authors appear to be more oriented towards the development of hard to produce and update generic databases of digital system failure data, than on the development of component-specific test-oriented assessment methodologies. (This is an indirect deduction by the reviewer, based in equal measure on what is stated and what is not stated in the report and its conclusions and recommendations) (Section 5).

Recommendations

1. A better distinction and definition of "traditional" and "advanced / dynamic" methods should be provided upfront in the report. One such definition is offered in Section 3.

2. The Report objectives that refer to the evaluation of "models" and "methods (2 and 3 in BNL report Section 1.1) should be reconsidered. More specifically, it is recommended that:

    a. Objective 2 be reformulated in terms of pursuing the trial application of evaluation criteria to available models (Section 3).

b. Objective 3 be reformulated in terms of seeking the identification of key useful features of existing methods that may be assembled within an overall "hybrid" and flexible framework (Section 3).

3. The BNL assessment of the NASA PRA Procedures Guide approach and framework should be revisited in light of the more recent and detailed information that is now available (Section 4). Particular attention should be given to this approach as an example of "hosting framework" for a hybrid combination of methods and models (see also discussion in Section 3).

4. Some of the BNL report conclusions should be adjusted to better reflect: a) the actual evidence gathered in the report, and b) any modification of aim and emphasis that may be put in effect as result of the current review process and incorporation of reviewers' recommendations (Section 5).

3. Report Objectives

The objectives of the BNL report are stated in Section 1.1, as follows:

1. Develop criteria for evaluating reliability models of digital systems. These draft criteria could eventually provide input to the technical basis for risk-informed decision-making.

2. Review reliability models developed using traditional methods, such as fault tree and Markov methods, against the criteria to determine the capabilities and limitations of the state-of-the-art of digital system reliability models using traditional methods.

3. Identify traditional methods to further explore that represent a spectrum of capabilities for modeling and quantitatively assessing the reliability of digital systems."

Much discussion took place at the May 23-24, 2007 expert panel meeting concerning the correct interpretation of these objectives and whether the work documented in the body of the letter report consistently reflected and fulfilled them. With the benefit of that discussion one can add the following explanatory observations:

- Objective 1 is oriented towards developing criteria that may provide a basis for regulatory evaluation of analytical models of NPP digital systems that may be developed to produce risk scenarios and associated risk estimates.

- Objective 2 is oriented towards a trial application of the developed criteria to a set of pre-existing "models" that were developed by various sources, using a variety of "traditional methods."

- Objective 3 is oriented toward down-selecting, from the initial set of traditional methods used in the various "models" evaluated and/or initially reviewed, a limited subset to further explore and evaluate for applications in the regulatory review arena.

Objective 1 is an easy-to-understand objective, which is also fully consistent with the general context and scope of the work carried out by the BNL team. No further comments are needed, except that the regulatory perspective of the model evaluation criteria is a key element of the objective that needs to be made clear to the reader for his/her correct interpretation of the objective.

Objective 2, taken at face value, also appears to be easy to understand and justify. However, when this objective is considered in combination with its companion Objective 3, several issues come forward, some of which were discussed at some length at the May 23-24 expert panel meeting:

- One issue concerns the selection of models to evaluate based on the distinction between "traditional" and "advanced" (and/or "dynamic") modeling methods, i.e.: what is the definition of "traditional method" as opposed to "advanced dynamic method"?

  This issue cannot be truly resolved in a clear-cut fashion, but one can recognize that it may be of limited importance in the context of the BNL activity and of the overall NRC research on digital systems risk and reliability. This is for two reasons:

  1. The main drivers in the selection of models to evaluate with the criteria developed under Objective 1 appear in practice to have been:

     a. the availability of a documented "real life" application of a traditional method (which the report refers to as a "model") to a relatively large scale system, and:

     b. the perceived relevance of the existing application to the subject of NPP digital systems risk.

  2. Advanced methods are being evaluated in a separate, but coordinated NRC research project.

- A second, and more serious issue, concerns what appears to be a logical disconnect between Objective 2 and 3, which has not yet been resolved, even after discussion at the review panel meeting. At the panel review, the report authors and sponsors clarified that the intent of the evaluation was to evaluate digital system "models" that were available and accessible, against the set of initially developed criteria, and not to apply the criteria to evaluate "methods."

  The logical disconnect occurs because the evaluation conducted under Objective 2 is de-facto translated, under Objective 3, into a down-selection of methods to further explore. Thus, although the declared intent of the report is to evaluate models, a de-facto judgment of goodness and suitability is transferred to the underlying methods, each taken as if it were a monolithic block, rather than a combination of many features matching or not the spectrum of criteria developed under Objective 1. The undesirable outcome that ensues is that, by following the above reasoning and course of actions, methods with potential good features and suitability for application in the regulatory context are excluded from further consideration, ostensibly because a good example of application fitting the report evaluation criteria was not readily available to the report authors. A related outcome, perhaps even more undesirable, is that given the way the

two objectives are stated and applied, outside readers will almost without doubt interpret the report evaluations to be general judgments passed on the suitability of the methods used in the various applications examined.

3.1 Recommendations on Statement and Application of the Report Objectives

The good news concerning the above is that the identified issues appear to be addressable in a reasonable fashion that would not be disruptive with regard to the work already carried out by BNL:

A. Distinction between "traditional" and "advanced" methods

As mentioned above, there is some good reason not to consider this a pressing issue. One of the experts in the review panel suggested a possible definition of "traditional" method as "one that is commonly used, well established, including large-scale applications." This reviewer's recommendation is that a definition along those lines be used, perhaps refined to read as follows: "A traditional method is one that has been fully demonstrated used, and established, including production-scale applications." This definition tempers the requirements following from the terms "commonly used" and "large-scale," mostly in recognition of the need to keep the horizon of methods open across industries and beyond the NPP world. What is "large scale" in one industry may be not so large when viewed from another industry's perspective. Moreover, most digital system analysis applications, even in production environments, have been at least in part exploratory and limited for one reason or another to a specific subsystem or set of subsystems.

B. Evaluation of "models" vs. "methods"

The recommendations to address this issue - in this reviewer's opinion a serious one that should be corrected with high priority - are two-fold:

1. Objective 2 should be restated to say that the primary purpose of the evaluation of models against criteria is test the use of the criteria against existing applications, with an accompanying objective of also better understanding the features of the methods used in these applications.

2. Objective 3 should also be restated to say that the results of the evaluations conducted per Objective 2 are used to identify features of existing methods that can be further explored and applied within an overall PRA type of framework to model and quantitatively assess the reliability of digital systems.

Applying Recommendation a) has limited impact on the existing contents of the report, whereas Recommendation b) would require a shift in the way Objective 3 is intended to lead to follow-on activities. That is, instead of a selection of a "method" as a whole for follow-on use and evaluation, the selection would have to identify specific aspects and portions of a method that can assembled into a PRA implementation. The selection of what one may call a "hybrid method" (or methods) was suggested by more than one reviewer at the expert panel sessions, and in fact there is nothing novel or unusual about

this approach since a typical PRA framework is indeed a hybrid model that cobbles together a number of different modeling and assessment techniques, i.e.,:

- event trees (and in some cases, especially in the aerospace industry, event sequence diagrams)
- fault trees (and in some cases reliability block diagrams)
- a whole assembly of failure rate and failure-on-demand probability quantification techniques and formulations.

In summary, there is really little reason to select a method on the basis on one "model" that has been evaluated, assuming all along that such application is a good representation and illustration of all the features of the method. The two recommendations presented here substantially reduce the potential "political liabilities" that one may incur in the technical community by making such a, real or perceived, leap of judgment. This is important, given the environment and type of audience for the BNL study. Aside from political considerations, an identification of specific method features that well match criteria, leading to the trial application of a hybrid method (or methods) using such features, appears to be the most technically sound, useful and insightful path towards practical and effective applications of digital system reliability and risk modeling methodology.

4. Report Review of NASA Reliability Models

The BNL draft letter report contains, in Appendix D, a review and evaluation of the "NASA Reliability Methods" (concerning digital systems and software). Because of his professional background and involvement, the reviewer is quite familiar with NASA PRA applications in general and with digital systems applications in particular. This specifically includes the "conditional risk model" presented and documented in the NASA PRA Procedures Guide.

This position of familiarity has made it possible for the reviewer to identify some misinterpretations and inaccuracies in the BNL review of the NASA methodologies and applications, which it is appropriate to address here, also in relation to the recommendations made earlier in Section 3.

The main misunderstanding is relative to the concept of "conditional risk model" set forward in the NASA PRA Procedures Guide, and associated confusion with regard to the use of some specific analytical technique or other within the conditional risk model framework.

The BNL report states:

"In the conditional risk model, hardware failure conditions are used to define the boundary conditions for modeling software failures. For each boundary condition, a software failure probability, independent of how long the software is running, is estimated using a reliability growth model. In two examples, a spacecraft attitude control system and a fluid tank control system, the method was applied. As discussed in [Chu 2006b], for control systems, software failure rate is a more appropriate parameter because the longer the control system is operating, the more likely that a triggering event would take place. Therefore, the conditional risk model of

the NASA PRA procedures guide is not consistent with the framework for probabilistic modeling of software failures. It should be revised to include consideration of the duration of operation in estimating the software failure probabilities."

The above interpretation of the NASA model is erroneous, as essentially the framework expressed the probability of a digital system software failure as:

- an unconditional rate of occurrence per unit time or per mission (i.e., rate per unit time multiplied by mission time duration) of the system condition/triggering event,

- multiplied by the conditional probability of digital system/software failure, given the occurrence of the triggering event.

Modeling digital system software failures in the fashion set forward by the NASA PRA procedures guide is based on the actual NASA and general space system (i.e., including DOD) experience with software related failures that have led to loss of missions. In addition, modeling a risk scenario via the frequency of an "initiating event" multiplied by the conditional probability of the event or chain of events that may follow is standard PRA modeling and quantification practice.

Part of the confusion in the assessment of the NASA method may be ensuing from the NRC and BNL concern with "digital systems" failures in general, as opposed to the specific emphasis on "software related failures" in Section 11 of the NASA PRA Procedures Guide. The primary intent of Section 11 is indeed to address software-related risk modeling, which is seen by NASA as the part of digital systems modeling with which PRA practitioners are mostly unfamiliar. However, the section provides a real-life example of modeling and quantification of digital space system risk, in which some failure scenarios are driven only by the hardware portion of the digital system (i.e., sensor interface, CPU, etc.), some only by the software portion, and some by a combination of the two.

Another important point apparently missed in the BNL review is that the NASA PRA Procedures Guide digital system modeling approach is intended to provide not a recipe for one specific combination of techniques (i.e., DFM or the Schneidewind software reliability growth model, which are used as individual elements of the application examples provided) but a flexible framework, tailorable in different types of detailed implementation, but maintaining in general the following characteristics:

- "upward" (i.e., scenario-level compatibility with the "standard" PRA event-tree/fault-tree modeling paradigm;

- ability to be "appended" and completed "downward" (i.e., in the direction of more detailed model development) with either traditional PRA modeling and quantification methods (e.g., ET/FT, Bayesian failure rate estimation, etc.) or more "advanced" or specifically software-oriented methods (DFM, Dynamic FT/Markov, SW reliability growth methods of various nature etc.).

A recently published NASA report ("Risk-Informed Safety Assurance and Probabilistic Risk Assessment of Mission-Critical Software-Intensive Systems," AR 07-01, ASCA, June 2007) more fully illustrates and documents the NASA PRA Procedures Guide Section 11 framework.

4.1 Recommendations Concerning Report Review of NASA Reliability Models

Given the obvious sensitivity carried by evaluations and assessments of methodologies developed by other parties, and especially in this case by another U.S. Government agency, it seems appropriate to recommend a more careful review and re-examination of the substance and contents of the NASA methods and models being assessed, also in light of the more in depth and easily accessible information provided by the NASA report identified above.

The above may also apply to some of the other models and methods reviewed by BNL, but the reviewer cannot extend any recommendations in such direction since he is not as specifically familiar with these models and methods as he is with those developed by NASA.

5. Report Conclusions and Recommendations

In its concluding Section 6.2, the report indirectly indicates a favorable view of "FT/ET" and Markov methods, by stating:

" ... The identified weaknesses of the FT/ET and Markov methods are not believed to be inherent weaknesses of these methods themselves, but rather weaknesses in the application of these methods in the studies reviewed. The FT/ET and Markov methods are very general and flexible, and it may be possible to use them to develop reasonable digital system reliability models if the identified weaknesses in the studies are addressed."

The above statement is probably correct with respect to the use of ET/FT modeling as a "classical" PRA top level framework, but it is less true for Markov modeling, which is not quite as flexible and presents a whole array of problems in terms of the "quantification burden" that it carries alongside. Markov models have in fact been used in PRA occasionally, and primarily as a complement to FT techniques, but not as the "hosting" framework. If accepted for Markov modeling, the statement would also arguably be true, or truer, for other methods that were either not considered (e.g., Bayesian Belief Networks, Petri Nets) or given only a summary examination leading to not quite accurate conclusions (see Section 4 above).

The indirect suggestions concerning FT/ET (perhaps to be better referred to as ET/FT, since that is the typical logic order of PRA model development) and Markov, or any other conclusions that may be drawn concerning the selection of methods for further examination (e.g., using the "models" with the higher scores in Section 6.1.3 of the BNL report as "templates" in future research) lead back to the issues and contradictions intrinsic to the evaluation of "models" versus "methods." This has already been addressed and discussed at some length above in Section 3.

With respect to the recommendations for areas of needed improvement in the state-of-the-art which are listed in Section 6.2 of the BNL report one can generally agree, with the following caveats:

- It is unclear if the call for the "development of methods for defining and identifying failure modes and effects of digital systems" is meant to be general, or limited to the improvement of "traditional methods"; if the latter is the case, the obvious objection is that a large portion of the research community in this area does not believe this to be

possible without introducing what the BNL reports considers "advanced dynamic methods."

- Many in the community, including this reviewer, are skeptical about the feasibility of developing generic databases for digital systems failure modes and failure rates. This is because digital system "generational" design and usage-span cycles have become shorter and shorter, so that any such database becomes obsolete in validity and applicability for the newest generation of digital hardware and software components, by the time enough data has been collected from the preceding generation.

- The above comment leads to one conclusion that this reviewer has been able to draw from his own experience with modeling and assessing digital system risk, that is, that perhaps the best hope for realistic quantification of such a risk has to rely on methods that can utilize direct test results or realistic simulation results for the systems of interest. Ref. 1 discusses how this can be done, at least for certain scenarios of especially critical concern.

5.1 Recommendations

It would be inappropriate, besides being also highly logically suspect, to provide here "recommendations on recommendations."

The only general and obvious recommendation concerning this portion of the report is that, if some of the other recommendations previously presented in this review were to be incorporated in the final version, the general aim of the concluding sections would have to be adjusted accordingly.

For example, adjustments to the report conclusions would have to be made if the evaluation of "models" were instead presented more as a means to try out the evaluation criteria contained in Section 3 of the report, while at the same time developing experience with methods and their application models.

Similarly, adjustments would be in order if the project moved more towards the idea of trying to identify the desirable features of a framework to host hybrid combinations of models and method applications, optimized to address specific types of decisions and assessment, instead of focusing on monolithic method applications.

Attachment F

Written Comments Provided by Reviewer C

1. Continuous Control Function

It is mainly depend upon specific plant design nevertheless the report should take into account possibility.

Digital I&C can be used for safety related continuous control function, which may have different attributes to be modeled in a quantitative model. For instance, KSNPP's auxiliary feed water actuation signal controls injection flow according to a SG level measurement. I believe new reactor designs use digital technology to accomplish the kind of continuous control functions. Different criteria may be applicable to the function.

2. Test

Test is discussed in a criterion, modeling of fault tolerance features, which is a subpart of criterion of 3.4, modeling of dependencies. Even though it is generally consider that test is a method to accomplish fault tolerance, modeling the test features in digital I&C should be handled separately with consideration of the widened test capability in digital technology. For instance, short test frequency reduce using dedicated test computer reduce failure probability of tested components, but the model should take into account the test coverage.

3. Capability Evaluation

This study evaluate whether six reviewed approaches satisfy the criteria that capture the design features of a digital system and can affect the system reliability. Besides the evaluation, it should be evaluated if the approaches are able to meet the criteria.

4. Mutually Exclusive Requirement

Report (page 31) indicates that some criteria are mutually exclusive or presents alternatives for a desirable characteristic. The study should describe the mutual exclusion and alternatives in Chapter 3 and take into account them to evaluate the six approaches in the comparison.

5. Definition of Criterion

Some criteria define specific requirements to model a certain characteristic, but the necessity of the model is depended upon a detailed design feature. Chapter 3 should define the requirement to take into account the certain design characteristic. For example, when an approach explicitly provides a justification without modeling, the approach is considered to satisfy the criterion. For instance, requirement 4.3.2 requires to model loss of HVAC and requirement. If a specific cabinet design has a temperature switch that initiate an automatic cabinet trip and/or operator actions, the necessity of HVAC modeling is negligible.

6. KSNPP Information

General information for KSNPP is provided in a few places in report, but the information is inconsistent. I will provide more information regarding KSNPP later.

Attachment G

Written Comments Provided by Reviewer D

Reviewer D provided written comments prior to the external review panel meeting. These comments were provided as annotations to the text of a pdf version of the BNL report. The context and essence of these comments are provided below.

1. Section 2.1, "Discussion of Methods," of BNL's report states that the FT/ET method can quantitatively evaluate the detailed failure modes of the plant. He pointed out that an exception is when the failures are statistically dependent.

2. Section 2.1, "Discussion of Methods," of BNL's report states that the FT/ET method does not explicitly treat the timing of events in accident sequences, but only accounts for them in an implicit way (i.e., through the specific events included in the ETs and their order of occurrence). He pointed out that this implicit way may not be able to account for the competition between top events.

3. Section 2.1, "Discussion of Methods," of BNL's report states that the FT/ET method considers interactions with plant processes only implicitly in an approximate way (primarily through the system success criteria). He pointed out that in addition, non-coherence due to diagnostic and recuperative capabilities of digital I&C systems may be a limitation.

4. Criterion 1.2 states "A probabilistic model of a digital system should be modeled at least at a level of detail for which the microprocessors are separately modeled." He pointed out that this criterion is unclear.

5. Section 3.2, "Identification of Failure Modes of the Digital System," of BNL's report states that ideally, the FMEA and these tools would be used in combination to identify more vulnerabilities of the system in a more reliable way than using FMEA alone. He pointed out that by design, FMEA is intended to identify immediate impacts, not to model fault propagation. So while the statement is correct, it should emphasize this intent.

6. Section 3.2, "Identification of Failure Modes of the Digital System," of BNL's report states that failure modes, failure causes, or failure effects are frequently mixed up, defined ambiguously, and sometimes they overlap or are even contradictory. He pointed out that these statements should be supported by citations from the literature.

7. Section 3.2, "Identification of Failure Modes of the Digital System," of BNL's report states that in an attempt to address the aforementioned problems with the current software failure categorization methods, a software failure categorization framework that involves definition of generic failure modes and failure causes was developed. He asked in which report this framework is presented.

8. Section 3.2, "Identification of Failure Modes of the Digital System," of BNL's report states that an obvious way of defining failure modes is in terms of the functions of the system or components, e.g., an analog input module of a system may fail to convert the input signal to the correct digital signal for the system to process, and a CPU may fail to

generate the correct output signals. He pointed out that this is not necessarily the case, and mentioned the example that a controller may generate arbitrary outputs or may undergo Byzantine failures.

9. Criterion 2.1 states "A technique such as FMEA should be applied at least to a level of detail corresponding to the basic components of the system, such as microprocessors." He pointed out that this level of detail may be neither feasible nor necessary, depending on device functionality and data availability.

10. Criterion 2.2 states "Supporting analysis should be carried out to determine how specific features of a design such as communication, voting, and synchronization could affect the operation of the system. It should determine if the specific design feature could introduce dependent failures that should be modeled." He has the same comment as the one for Criterion 2.1.

11. Criterion 2.3 states "The information associated with the probabilistic model of a digital system should provide justification that the design requirements of the digital system are unambiguous, complete and consistent, and that these requirements have been implemented in the system." He pointed out that this criterion is unclear.

12. Section 3.3, "Modeling of Software Failures," of BNL's report states that hardware fails due to factors such as wear and tear, while a software failure happens due to the presence of a fault in the software and the occurrence of a specific set of input data. He asked the question "Due to specification error?"

13. Section 3.3, "Modeling of Software Failures," of BNL's report states that the occurrence of the input is random and can be modeled in terms of failure rates and failure probabilities. He pointed out that this statement is debatable.

14. Section 3.3, "Modeling of Software Failures," of BNL's report states that for a protection system, a failure rate can be used to model errors introduced during software updates. He pointed out that this statement needs substantiation by references.

15. Section 3.3, "Modeling of Software Failures," of BNL's report states that one way in which software failures may be explicitly included in the logic model is by somehow developing a model of behavior of the software and including it in the logic model of the rest of the NPP, and that this approach has been named "system-centric." He does not believe a software model is necessary for the system-centric approach.

16. In the bottom paragraph of page 12 of BNL's report, "Thereport" is a typo.

17. He suggests to rephrase Criteria 4.1.3 and 4.1.4 in a more non-device-specific manner.

18. Criterion 4.4 states that the digital systems of a plant should be examined to determine if there are dependencies due to sharing digital hardware. Such a dependency should be modeled, e.g., by linking fault trees. He pointed out that "e.g., by linking fault trees" is unclear.

19. Criterion 4.4 also states that if RPS and ESFAS are implemented using the same digital hardware, a conservative approach for accounting for this dependency is assuming that RPS is failed in those sequences with ESFAS failed due to digital failures, and vice versa. He pointed out that this statement is unclear.

20. In the discussion on "Modeling of Fault Tolerance Features," BNL's report states that the objective of a fault-tolerant feature is to have a positive impact on the risk metrics of a system, such as the system's reliability. On the other hand, a fault-tolerant feature may fail to detect and/or fix a failure mode that it was designed to catch. He pointed out that if fault-tolerance relies on self-testing, the system may be vulnerable during the testing process.

21. Section 3.5, "Human Errors," of BNL's report states that once a digital system has been installed and is operational in a nuclear power plant (NPP), an upgrade may introduce new errors into the system. This type of failure also may happen when upgrading an analog system. However, it appears that it has a higher probability of occurring when upgrading a digital system due to the greater complexity of these systems. He pointed out that the last statement needs to be justified by a reference.

22. Under "Requirements" of the Section 3.5 "Human Errors," the BNL's report states that two types of human errors that are related to a digital system are the introduction of faults when upgrading its hardware or software, and poorly designed or implemented man-machine interfaces (MMI). The human reliability analysis of the probabilistic model should take into account these types of failures. Regarding the last sentence, he would simply say "The probabilistic model should take into account these types of failures." He pointed out that there may be data available.

23. Criterion 5.2 requires modeling of human errors due to poor design of MMI. He asked if the term MMI has been defined earlier.

24. Section 3.6, "Ease of Integration with a PRA Model," of BNL's report states that one way to integrate a model of a digital system with an existing PRA model is by directly integrating the system model with the PRA model. Since the current PRAs use the FT/ET method, this approach can only be achieved by using a fault tree model of the digital system. He recommended to replace the words "fault tree" by "ET/FT" in the last sentence.

25. Section 3.6, "Ease of Integration with a PRA Model," of BNL's report states that one way to integrate a model of a digital system with an existing PRA model is by using the results from the model of a digital system in a PRA in a consistent way. This approach basically consists of developing a model of a digital system using a technique such as the Markov method. He recommended to replace the words "such as the Markov method" with "that can account for the all the relevant features of the digital system."[4]

[4]Reviewer D pointed out the following publication for information on these features: J. Kirschenbaum, M. Stovsky, P. Bucci, T. Aldemir, S.A. Arndt, "Benchmark Development for Comparing Digital Instrumentation and Control System Reliability Modeling Approaches", PSA'05, on CD-ROM, American Nuclear Society, LaGrange Park, IL (September 2005).

26. Criterion 6.1 states that a fault tree model of a digital system should be easy to integrate with a PRA model. All other methods require additional efforts in integration. He recommended to replace the words "fault tree" by "ET/FT."

27. A paragraph in the middle of page 23 of BNL's report starts with the sentence "Thereview of the availability of hardware data for digital components reveals that ..." He pointed out that "Thereview" is a typo.

28. Point 6 in pages 32 and 76 of BNL's report states that based on the review of the 6 studies, and previous research on PRAs of digital systems, it is still believed that the two main types of "traditional" methods, i.e., fault tree/event tree and Markov, are capable of assessing these systems. He pointed out that this statement needs substantiation in view of the criteria formulated, even if the "and" in the part "fault/tree and Markov" is meant as "in conjunction with". Neither of them will pick up all the Type I or Type II interactions. It is true that Markov/CCMT is capable of meeting all the requirements, but then Markov/CCMT is not a traditional method.

29. In Section 5.2, "Application-Specific Observations," under the heading "Modeling of the AP 1000 using the FT/ET method," the BNL's report states that a strength of this modeling is that software failures were explicitly included in the logic model. He asked where did the data come from?

30. In the first paragraph of Section 6.2 "Conclusions and Recommendations," the BNL's report states that strengths and weaknesses have been identified for the studies reviewed. The weaknesses represent limitations of the current state of the art in modeling digital systems. The identified weaknesses of the FT/ET and Markov methods are not believed to be inherent weaknesses of these methods themselves, but rather weaknesses in the application of these methods in the studies reviewed. The FT/ET and Markov methods are very general and flexible, and it may be possible to use them to develop reasonable digital system reliability models if the identified weaknesses in the studies are addressed. He pointed out that the last sentence is debatable.

31. Section 6.2, "Conclusions and Recommendations," of the BNL's report identified several areas of research that would enhance the state of the art. In addition to these areas, he suggested dependencies arising from Type I and Type II interactions.

Attachment H

Order for Addressing Review Criteria Categories/Subcategories

1. Category 1. Level of Detail of the Model

2. Category 2. Identification of Failure Modes of the Components of Digital Systems

3. Category 3. Software Failures

4. Category 4. Modeling of Dependencies

   Subcategory 4.1 Communication networks/buses
   Subcategory 4.3 Support systems
   Subcategory 4.4 Sharing hardware
   Subcategory 4.5 Interactions of digital systems with other systems
   Subcategory 4.6 Modeling of fault tolerance features
   Subcategory 4.2 Common cause failures

5. Category 7. Probabilistic Data

   Subcategories 7.1-7.10 Hardware failure data
   Subcategories 7.11-7.12 Software failure data

6. Criteria 8.4-8.5. Uncertainty

7. Category 6. Ease of Integration with a PRA Model

8. Category 5. Human Errors

9. Criteria 8.1-8.3. Documentation and Results

APPENDIX B

DETAILED FAILURE MODES AND EFFECTS ANALYSES (FMEA) OF THE DIGITAL FEEDWATER CONTROL SYSTEM (DFWCS) AT DIFFERENT LEVELS[1]

[1]The FMEAs presented here make extensive use of the hazard analyses performed by the plant.

TABLE OF CONTENTS

Section		Page
B.1	Top-Level FMEA of DFWCS	B-1
B.2	FMEA at Level of DFWCS Modules	B-2
B.3	FMEA at Level of Major-Component-of-Module of DFWCS	B-74
B.4	References	B-106

LIST OF TABLES

Table		Page
B.1-1	Top-Level FMEA of DFWCS	B-1
B.2-1	FMEA of Analog Backplane A (FY1111B/C1)	B-3
B.2-2	FMEA of Analog Backplane B	B-12
B.2-3	FMEA of Digital Backplane (I/O) of Main CPU	B-17
B.2-4	FMEA of MFV Controller (FIC-1111/1121)	B-35
B.2-5	FMEA of BFV Controller (FIC-1105/1106)	B-44
B.2-6	FMEA of FWP Controller (FIC-4516/4517)	B-55
B.2-7	FMEA of Pressure Differential Indicating (PDI) Controller PDI-4516/4517)	B-62
B.2-8	FMEA of Optical Isolator (PB4R)	B-72
B.3-1	FMEA at Level of Components of DFWCS Modules: Main CPU	B-75
B.3-2	FMEA of F&P 53MC5000 Controller	B-101

APPENDIX B.1: TOP-LEVEL FMEA OF DFWCS

Table B.1-1 Top-level FMEA of DFWCS.

Mode of operation of the plant: Power operation **Mode of operation of the MFW: High power**		
Failure Mode	**Detection of Failure Mode**	**Failure Effect on Main Feedwater System**
No or "low" signal from DFWCS to controlled components	Indications in control room of low feedwater flow and low level in steam generator(s) (SGs)	Low level in SGs can cause reactor trip
		Reduction of level in SG(s) can possible contribute to steam generator tube rupture (SGTR)
"High" signal from DFWCS to controlled components	Indications in control room of high feedwater flow and high level in SGs	Excessive feedwater to steam generator(s) can cause reactor trip
Abnormal fluctuations of signal from DFWCS to controlled components	Depending on frequency and severity of fluctuations, operators in control room may be able to detect changes in feedwater flow and in level in SGs	Effects are expected to be similar to those resulting from the previous two failure modes
Failure to transfer to low-power mode when reactor power decreases below 15% and remains above about 2%	Indications in control room of high level in SGs	A mismatch between the power produced by the reactor and the cooling of the SGs by the DFWCS. The mismatch may result in excessive feedwater to SGs causing a reactor trip.

APPENDIX B.2: FMEA AT LEVEL OF DFWCS MODULES

The next level of the FMEA includes the modules of the DFWCS. The major modules of the DFWCS include the Main CPU, Backup CPU, MFV Controller, BFV Controller, FWP Controller, PDI Controller, and the optical isolator that is related to the WDT signal. The FMEA is performed based on failures of input/output signals that reflect the failure modes of these modules. Thus, the FMEAs for the Main and Backup CPUs are actually the same as those for their respective analog and digital backplanes, because the backplanes contain all of the input/output signals of the CPUs. The FMEAs for the analog and digital backplanes associated with the Main CPU are provided below. The failure modes for the backplanes associated with the Backup CPU are expected to be similar to those for the Main CPU, but the failure effects are expected to be different.

Table B.2-1 FMEA of analog backplane A (FY1111B/C1).

Failure mode	Detection of failure mode	Failure effects	Comments
Analog Backplane - Loss of communications	There is no direct indication of the loss of communication. Failure of the CPU would send an alarm to the Plant Computer.	The application software checks that there are no errors (UMAC_NO_ERROR). Loss of communication would result in an error due to no response from the analog board causing the analog inputs to be invalid which would result in a failure of the CPU. A failover to the B/U CPU will take place.	It is assumed that the loss of communication to the CPU has no effect on other functions of the ISA bus.
Analog Backplane - Loss of power (5V source)	There is no direct indication of the loss of 5V source. Failure of the CPU would send an alarm to the Plant Computer.	The application software checks that there are no errors (UMAC_NO_ERROR). Loss of 5V source would result in an error due to no response from the analog board causing the analog inputs to be invalid which would result in a failure of the CPU. A failover to the B/U CPU will take place.	
Channel 1 - Feedpump A Demand (Output) Failed Hi OOR	There is no direct indication of the failure. The Main CPU deviation logic will be alarmed at the Plant Computer.	The Main CPU deviation logic will detect a large deviation between the CPU demand and the main FWP track signal and the B/U CPU will initially track the FWP signal to the failed value. The Main CPU will then fail, transferring control to the B/U CPU. When the B/U CPU assumes control it will retrieve the pre-failure demand signal to use for its initial output. Control will be maintained by the B/U CPU and the Operator will be alerted by the DFW system trouble alarm on the Plant Computer. The Lovejoy Control system will detect a short duration increase in pump demand and will interpret this as an HIC failure. Lovejoy will then transfer control to Diagnostic Manual and maintain the FWP speed at the pre-failure speed demand.	

Table B.2-1 FMEA of analog backplane A (FY1111B/C1) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 1 - Feedpump A Demand (Output) Failed Low OOR	There is no direct indication of the failure. The Main CPU deviation logic will be alarmed at the Plant Computer.	The Main CPU deviation logic will detect a large deviation between the CPU demand and the FWP tracking signal and the B/U CPU will initially track the FWP signal to the failed value. The Main CPU will then fail, transferring control to the B/U CPU. When the B/U CPU assumes control, it will retrieve the pre-failure demand signal to use for its initial output. Control will be maintained by the B/U CPU and the Operator will be alerted by the DFW system trouble alarm on the Plant Computer. The Lovejoy Control system will detect a short duration increase in pump demand and will interpret this as an HIC failure. Lovejoy will then transfer control to Diagnostic Manual and maintain the SGFP speed at the pre-failure speed demand.	Need to confirm how the Lovejoy controller detects the failure.
Channel 1 - Feedpump A Demand (Output)- Excess Drift, or Step Change (in range)	There is no direct indication of this failure.	Because rate failures are not detected by the DFW system, a step change with a magnitude less than the deviation setpoint may result in a system flow transient. Because of this, the deviation setpoint should be set at a value at which the DFW control system can compensate for without resulting in a plant trip.	
Channel 2 - Bypass Valve Demand (Output) Failed Hi OOR	There is no direct indication of the failure.	The BFV demand signal is commanded to zero during high power mode. Should the BFV demand increase, the MFV demand will decrease as during a valve transfer, limiting the induced transient. The CPU deviation logic for the BFV demand signal is inhibited during High Power Mode Operations.	

Table B.2-1 FMEA of analog backplane A (FY1111B/C1) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 2 - Bypass Valve Demand (Output) Failed Low OOR	This is no direct indication of the failure.	The BFV demand signal is normally at zero during high power mode. If the BFV demand signal remains at zero, nothing will happen. This fault will remain undetected until a valve transfer occurs. At this time, the Main CPU deviation logic becomes active and will detect a large deviation between the CPU demand and the BFV signal. The B/U CPU will track the BFV signal to the failed value. The Main CPU will then fail, transferring control to the B/U CPU. When the B/U CPU assumes control, it will use the tracked demand signal for its initial output. Control will be assumed by the B/U CPU and the Operator will be alerted by the DFW system trouble alarm on the Plant Computer.	
Channel 2 - Bypass Valve Demand (Output) Excess Drift or Step Change (in range)	There is no direct indication of the failure.	Because rate failures are not detected by the DFW system, a step change with a magnitude less than the deviation setpoint will result in a system flow transient. Because of this, the deviation setpoint should be set to a value at which the DFW control system can compensate for without resulting in a plant trip.	

Table B.2-1 FMEA of analog backplane A (FY1111B/C1) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 3 - Main Valve Demand (Output) Failed Hi OOR	There is no direct indication of the failure. The Main CPU deviation logic will be alarmed at the Plant Computer.	The failed signal will be sent to the MFV controller causing the MFRV to open wider. The Main CPU deviation logic will detect a large deviation between the CPU demand and the MFV track signal. The B/U CPU will initially track the MFV signal to the failed value. The *Main CPU will then fail, transferring control to* the B/U CPU. When the B/U CPU assumes control, it will retrieve the pre-failure demand signal to use for its initial output. Control will be maintained by the B/U CPU and the operator will be alerted by the DFW trouble alarm on the Plant Computer. The FWP speed will be momentarily affected because the high auctioneered MFV signal is used to control the FWP speed. This SGFP demand transient could cause the Lovejoy Control system to interpret an HIC failure which would transfer SGFP control to Diagnostic Manual and maintain SGPR speed at the pre-failure value.	
Channel 3 - Main Valve Demand (Output) Failed Low OOR	The PDI controller will display a "MFV Fail" message. A deviation message is activated by the Main CPU, after a settable, predetermined time delay. (This message may not be generated, because the PDI controller is expected to take over.)	The MFV controller will initially forward the failed demand signal to the MFRV positioner, PDI controller, and the CPUs of the other S/G. The PDI controller will then detect the signal failure and automatically become the manual controller for the MFRV using the old value in the circular buffer. The MFRV must be manually controlled from the PDI controller. The failed signal will be sent to the CPUs of other SG, and probably will not affect the FWP speed calculation.	The response specified in plant analysis probably will not take place, because the PDI controller has a scan time of not exceeding 100 milliseconds, while the CPU failover logic has a 1 second delay. The MFV demand signal is also sent to the CPUs of the other S/G and used in the FWP speed calculation.

Table B.2-1 FMEA of analog backplane A (FY1111B/C1) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 3 - Main Valve Demand (Output) Excess Drift, or Step Change (in range)	There is no direct indication of the failure.	Because rate failures of the BFV demand signal are not detected by the DFW system, a step change with a magnitude less than the deviation setpoint will result in a system flow transient. Because of this, the deviation setpoint should be set at a value at which the DFW control system can compensate for without resulting in a plant trip.	
Channel 4 - S/G 12 FW Temp (Input) OOR	A deviation alarm will be sent from the Main CPU to the Plant Computer.	The OOR condition and deviation will be detected by the Main CPU. The signal becomes invalid and the other signal is used. There is no effect on control. The signal is only used during low power operation.	
Channel 4 - S/G 12 FW Temp (Input) Excess Drift or Step Change (in range)	If the deviation is large enough, an alarm will be sent from the Main CPU to the Plant Computer.	The temperature signals are averaged and a deviation will not significantly affect low power control.	
Channel 5 - S/G 11 FW Temp (Input) OOR	A deviation alarm will be sent from the Main CPU to the Plant Computer.	The OOR condition and deviation will be detected by the Main CPU. The signal becomes invalid and the other signal is used. There is no effect on control. The signal is only used during low power operation.	
Channel 5 - S/G 11 FW Temp (Input) Excess Drift or Step Change (in range)	When the deviation is large enough, an alarm will be sent from the Main CPU to the Plant Computer.	The temperature signals are averaged and a deviation will not significantly affect low power control.	

Table B.2-1 FMEA of analog backplane A (FY1111B/C1) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 6 - S/G 11 Feedpump A Bias (Input) Fails High or Low OOR	A deviation alarm will be sent to the Plant Computer from the Main CPU. The FWP controller will activate a local alarm when the Main CPU demand signal differs from the B/U CPU demand signal by an amount exceeding a setpoint. The FWP controller will activate a local deviation alarm when the Main CPU demand signal differs from the B/U CPU demand signal by an amount exceeding a setpoint.	The OOR condition will be detected by the Main CPU, and a deviation alarm will be sent to the Plant Computer. The Main CPU will send the pump demand calculated with the failed bias signal to the FWP controller.	The effects depend on whether or not the failed signal at the Main CPU would also be received by the BFV controller. It is assumed the Main CPU will send the pump demand calculated with the failed bias signal to the FWP controller.
Channel 7 - S/G 12 Main Valve Tracking (Input) Failed Hi OOR	There in no direct indication of the failure. The increase in pump speed could be alarmed using the future D/P signals.	This is one of two signals (S/G 12 and S/G 11) used for high select to determine the FWP speed. A failed high signal will increase the pump speed which would cause a controllable disturbance at high power but could result in loss of control at low power. The Lovejoy Control System may detect this large change and transfer to Diagnostic Manual mode.	
Channel 7 – S/G 12 Main Valve Tracking (Input) Failed Low OOR	There is no direct indication of the failure.	This is one of two signals used for high select to determine the FWP speed. A failed low signal will have minimal effect on system operation.	
Channel 7 - S/G 12 Main Valve Tracking (Input) Excess Drift, or Step Change (in range)	There in no direct indication of the failure.	A decrease in value would have no effect; an increase in value would have little effect and be compensated for by an adjustment in the controlling valve.	

Table B.2-1 FMEA of analog backplane A (FY1111B/C1) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 8 - S/G 11 FWP A Tracking (Input) Fail High OOR, Fail Low OOR, and Excess Drift, or Step Change (in range)	There in no direct indication of the failure. The Main CPU failure will result in an alarm being sent to the Plant Computer.	The deviation between the CPU and controller will cause a failover to the B/U CPU as long as the deviation setpoint is exceeded. If the deviation setpoint is not exceeded, control will continue as the controlling demand signal is valid.	
Channels 9-12 are spares			
Channel 13 - MFRV LVDT #2 Fail High or Fail Low	A MFV Large Deviation alarm will be activated on the plasma display unit and the associated CPU deviation annunciator will activate, if the deviation between the two LVDT inputs exceeds the MFV_Deviation setpoint. If the Diagnostic Transfer mode is enabled, then it will transfer to Lockout.	If the deviation between the two LVDT inputs exceeds the MFV_DEVIATION setpoint, the Diagnostic Transfer mode will transfer to Lockout. If the MFV_DEVIATION setpoint is not exceeded and the MFV_DEADBAND setpoint is exceeded by the Demand-LVDT deviation, where the LVDT is the average of the two LVDT signals, the DMD-LVDT deviation will be accumulated over the subsequent cycles. If the accumulation exceeded the MFV_ACCUMULATION setpoint, and the Diagnostic Transfer control mode is ENABLED, the opposite positioner will be put into service and the control mode will be shifted to LOCKOUT.	It is not known what the CPU deviation annunciator is. It probably is a local annunciator on the PDU. The CPUs do not have direct connection to the control room annunciators.
Channel 14 - MFRV LVDT #1 Fail High or Fail Low	Same as above.	Same as above.	Same as above.

Table B.2-1 FMEA of analog backplane A (FY1111B/C1) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 15 or 16 - MFRV Differential Pressure #2 or #1 Fail High	The incorrect gooseneck flow and accumulated volume will be displayed on the Analog Inputs display page of the PDU.	The failed high signal will falsely indicate a larger than normal differential pressure, which would result in an incorrectly high accumulated gooseneck volume and prevent needed gooseneck purge. It is not clear what adverse effects would result when the needed purge is not performed.	The gooseneck is an upward bend and loop installed down stream of the feedwater nozzle of replacement steam generators to prevent flow of steam generator fluid upstream. When a gooseneck purge is needed, as determined by the accumulated gooseneck volume being less than the minimum volume setpoint, the BFRV alarm status becomes GSNECK PURGE and the associated CPU deviation annunciator will activate. The operator has to manually purge the Gooseneck. It is not known what the CPU deviation annunciator is. It probably is a local annunciator on the PDU. The CPUs do not have direct connection to the control room annunciators.

Table B.2-1 FMEA of analog backplane A (FY1111B/C1) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 15 or 16 - FRV Differential Pressure #2 or #1 Fail Low	The incorrect gooseneck flow and accumulated volume will be displayed on the Analog Inputs display page of the PDU.	The failed low signal will falsely indicate a smaller than normal differential pressure, which would result in an incorrectly low accumulated gooseneck volume and premature gooseneck purge. It is not clear what adverse effect would result when the premature purge is performed.	The gooseneck is an upward bend and loop installed down stream of the feedwater nozzle of replacement steam generators to prevent flow of steam generator fluid upstream. When a goose-neck purge is needed, as determined by the accumulated gooseneck volume being less than the minimum volume setpoint, the BFRV alarm status becomes GSNECK PURGE and the associated CPU deviation annunciator will activate. It is not known what the CPU deviation annunciator is. It probably is a local annunciator on the PDU. The CPUs do not have direct connection to the control room annunciators.

Table B.2-2 FMEA of analog backplane B[2].

Failure mode	Detection of failure mode	Failure effects	Comments
Analog Backplane - Loss of Communications	Same as Analog Backplane A.	Same as Analog Backplane A.	Same as Analog Backplane A.
Analog Backplane - Loss of Power	Same as Analog Backplane A.	Same as Analog Backplane A.	Same as Analog Backplane A.
Channel 6 - S/G 11 Level #1: Failed Hi OOR, Failed Low OOR, and Rate	Main CPU (the controlling CPU) Fail alarm status will be displayed on PDU if the failover occurs.	The other level input #2 is used and control continues. After a delay, if the B/U CPU is healthy, a failover to the B/U CPU will occur.	Channel 6 is an input signal. Failure effects are the same for both high and low power control modes.
Channel 6 - S/G 11 Level #1: Excess Drift or Step Change (the change is within the range)	A deviation alarm status will be actuated in the plant computer. Main CPU Fail alarm status will also be displayed if failover occurs.	A deviation between S/G 11 Level #1 signal and S/G 11 Level #2 signal will occur. A small deviation will result in a deviation alarm to the plant computer. If the deviation continues, a large deviation will result and after some delay, a failover to the B/U CPU will occur if it is healthy. Otherwise, the control will continue with the average of the two level inputs.	Channel 6 is an input signal. Failure effects are the same for both high and low power control modes.

[2]According to plant information, Channels 1 and 2 are reserved for test point output. It is assumed that these are for off-line test of the digital control system. Channels 3-5 are spares. The failures of these analog signals are not considered here.

Table B.2-2 FMEA of analog backplane B (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 7 - S/G 11 Level #2: Failed Hi OOR, Failed Low OOR, and Rate	Main CPU (the controlling CPU) Fail alarm status will be displayed on PDU if the failover occurs	The other level input #1 is used and control continues. After a delay, if the B/U CPU is healthy, a failover to the B/U CPU will occur.	Channel 7 is an input signal. Failure effects are the same for both high and low power control modes.
Channel 7 - S/G 11 Level #2: Excess Drift, or Step Change (the change is within the range)	A deviation alarm status will be actuated in the plant computer. Main CPU Fail alarm status will also be displayed if failover occurs.	A deviation between S/G 11 Level #1 signal and S/G 11 Level #2 signal will occur. A small deviation will result in a deviation alarm to the plant computer. If the deviation continues, a large deviation will result and after some delay, a failover to the B/U CPU will occur if it is healthy. Otherwise, the control will continue with the average of the two level inputs.	Channel 7 is an input signal. Failure effects are the same for both high and low power control modes.
Channel 8 - S/G 11 FW Flow #1: Failed Hi OOR, Failed Low OOR, and Rate	Main CPU (the controlling CPU) Fail alarm status will be displayed on PDU if the failover occurs.	The other FW flow input #2 is used and control continues. After a delay, if the B/U CPU is healthy, a failover to the B/U CPU will occur.	Channel 8 is an input signal. Failure effects are the same for both high and low power control modes.
Channel 8 - S/G 11 FW Flow #1: Excess Drift, or Step Change (the change is within the range)	A deviation alarm will be actuated in the plant computer. Main CPU Fail alarm will also be displayed if the failover occurs.	A deviation between S/G 11 FW Flow #1 signal and S/G 11 FW Flow #2 signal will occur. A small deviation will result in a deviation alarm to the plant computer. If the deviation continues, a large deviation will result. A large deviation will result in single element mode control.	Channel 8 is an input signal. At low power control mode, a large deviation will result in inhibiting a low to high power transfer and an inhibiting transfer alarm will be displayed in the plant computer.
Channel 9 - S/G 11 FW Flow #2: Failed Hi OOR, Failed Low OOR, and Rate	Main CPU (the controlling CPU) Fail alarm status will be displayed on PDU if the failover occurs.	The other FW flow input #1 is used and control continues. After a delay, if the B/U CPU is healthy, a failover to the B/U CPU will occur.	Channel 9 is an input signal. Failure effects are the same for both high and low power control modes.

Table B.2-2 FMEA of analog backplane B (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 9 - S/G 11 FW Flow #2: Excess Drift, or Step Change (the change is within the range)	A deviation alarm status will be displayed in the plant computer. Main CPU Fail alarm status will also be displayed if failover occurs.	A deviation between S/G 11 FW Flow #1 signal and S/G 11 FW Flow #2 signal will occur. A small deviation will result in a deviation alarm to the plant computer. If the deviation continues, a large deviation will result. A large deviation will result in single element mode control.	Channel 9 is an input signal. At low power control mode, a large deviation will result in inhibiting a low to high power transfer and an inhibiting transfer alarm will be displayed in the plant computer.
Channel 10 - S/G 11 Main Steam Flow: Failed Hi OOR, Failed Low OOR, and Rate	Main CPU (the controlling CPU) Fail alarm status will be displayed on PDU if the failover occurs.	The other steam flow input is used and control continues. After a delay, if the B/U CPU is healthy, a failover to the B/U CPU will occur.	Channel 10 is an input signal. Failure effects are the same for both high and low power control modes.
Channel 10 - S/G 11 Main Steam Flow: Excess Drift, or Step Change (the change is within the range)	A deviation alarm status will be displayed in the plant computer. Main CPU Fail alarm status will also be displayed if failover occurs.	A deviation between S/G 11 Main Steam Flow and S/G 12 Main Steam Flow signals will occur. A small deviation will result in a deviation alarm to the plant computer. If the deviation continues, a large deviation will result. A large deviation will result in single element model control.	Channel 10 is an input signal. At low power control mode, a large deviation will result in inhibiting a low to high power transfer and an inhibiting transfer alarm will be displayed in the plant computer. Note the steam flow small deviation alarms and messages are disabled when reactor power is below the low to high power mode transfer setpoint.
Channel 11 – S/G 12 Main Steam Flow: Failed Hi OOR, Failed Low OOR, and Rate	Main CPU (the controlling CPU) Fail alarm status will be displayed on PDU if the failover occurs.	The other steam flow input is used and control continues. After a delay, if the B/U CPU is healthy, a failover to the B/U CPU will occur.	Channel 11 is an input signal. Failure effects are the same for both high and low power control modes.

Table B.2-2 FMEA of analog backplane B (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 11 - S/G 12 Main Steam Flow: Excess Drift, or Step Change (the change is within the range)	A deviation alarm status will be displayed in the plant computer. Main CPU Fail alarm status will also be displayed if failover occurs.	A deviation between S/G 11 Main Steam Flow and S/G 12 Main Steam Flow signals will occur. A small deviation will result in a deviation alarm to the plant computer. If the deviation continues, a large deviation will result. A large deviation will result in single element model control.	Channel 11 is an input signal. At low power control mode, a large deviation will result in inhibiting a low to high power transfer and an inhibiting transfer alarm will be displayed in the plant computer. Note the steam flow small deviation alarms and messages are disabled when reactor power is below the low to high power mode transfer setpoint.
Channel 12 - Neutron Flux #1: Failed Hi OOR, Failed Low OOR	A deviation alarm status may be displayed in the plant computer.	The other flux input (#2) is used and control continues.	Channel 12 is an input signal. No deviation logic for CPU failover for neutron flux inputs.
Channel 12 - Neutron Flux #1: Excess Drift, or Step Change (the change is within the range)	A deviation alarm and an inhibit transfer alarm will be displayed in the plant computer.	Valve transfers are inhibited and control continues as long as the other flux input (#2) is valid.	Channel 12 is an input signal. At low power control mode, the last valid flux signal is frozen to minimize any disturbance.
Channel 13 - Neutron Flux #2: Failed Hi OOR, Failed Low OOR	A deviation alarm status may be displayed in the plant computer.	The other flux input (#1) is used and control continues.	Channel 13 is an input signal. No deviation logic for CPU failover for neutron flux inputs.
Channel 13 - Neutron Flux #2: Excess Drift, or Step Change (the change is within the range)	A deviation alarm and an inhibit transfer alarm will be displayed in the plant computer.	Valve transfers are inhibited and control continues as long as the other flux input (#1) is valid.	Channel 13 is an input signal. At low power control mode, the last valid flux signal is frozen to minimize any disturbance.

Table B.2-2 FMEA of analog backplane B (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 14 - S/G 11 Level Setpoint: Failed Hi OOR, Failed Low OOR, and Excess Drift, or Step Change (the change is within the range)	A deviation alarm will be displayed in the plant computer.	A deviation between this input and the program value of the setpoint will occur. If the deviation is less than a fixed value (LEV_SPT), the control continues. Otherwise, an internal level setpoint will be used as the substitute.	Channel 14 is an input signal.
Channel 15 - S/G 11 BFRV Tracking: Failed Hi OOR, Failed Low OOR, and Excess Drift, or Step Change (the change is within the range)	No alarms are generated.	Control will continue and the CPU or BFV still sends demand output that will close the BFRV.	Channel 15 is an input signal. At low power control mode, the deviation between the CPU and controller will cause a failover to the B/U CPU some time (deviation time delay) after the deviation setpoint is exceeded.
Channel 16 - S/G 11 MFRV Tracking: Failed Hi OOR, Failed Low OOR, and Excess Drift, or Step Change (the change is within the range)	A deviation alarm will be displayed on PDU. Main CPU (the controlling CPU) Fail alarm will also be displayed on PDU if the failover occurs.	The deviation between the CPU output and controller output will cause a failover to the B/U CPU as long as the deviation setpoint is exceeded after some time delay. If the deviation setpoint is not exceeded, control will continue as the controlling demand signal is valid. For the Fail Hi OOR signal, the FWP speed will momentarily increase due to high MFV actioneering, which is used to compute the FWP pump demand.	Channel 16 is an input signal.

Table B.2-3 FMEA of digital backplane (I/O) of main CPU.

Failure mode	Detection of failure mode	Failure effects	Comments
Loss of Communications			
Loss of communications	The three controllers (Main, Bypass and FW Pump) alarm lights energize. The BFV controller transmits to the plant computer that one CPU has failed. The PDU shows that the Main microprocessor is failed.	Loss of communications would result in the digital signals maintaining their existing state which would cause a watchdog failure which would result in a failover to the B/U CPU.	
Loss of Power			
Loss of Power	The three controllers (Main, Bypass and FW Pump) alarm lights energize. The BFV controller transmits to the plant computer that one CPU has failed. The PDU shows that the Main microprocessor is failed.	Loss of power would result in the digital signals changing to their unpowered state which would cause a watchdog failure which would result in a failover to the B/U CPU.	
Digital Outputs			
Channel 0 - Watchdog Timer (Output) fails as is	There is no direct indication of this failure. Indirect indications are annunciations of failure of the Main CPU in this CPU's PDU, and in the plant computer (from the BFV).	A failure of this output to change state would result in a Main CPU failover to the B/U CPU.	Output state: toggling (not failed). Watchdog Timer failing as is indicates that the timer identified a failure of the Main CPU.

Table B.2-3 FMEA of digital backplane (I/O) of main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 1 - Unusable	Not applicable (NA)	NA	Need to confirm why this channel is unusable and whether it could have some failure mode.
Channel 2 - Power Fail (Output) fails as is	There is no indication of this failure.	A failure of this output would have no effect on control until some other condition caused the Main CPU to fail, at which time automatic control would be lost. The severity of the loss of control would depend on the CPU fault.	This channel indicates power failure or microprocessor not controlling. Output state: not energized (OK). Power Fail failing as is indicates that the Main CPU is OK.
Channel 2 - Power Fail (Output) fails to opposite state	There is no direct indication of this failure. Indirect indications are annunciations of failure of the Main CPU in the PDU of the DFWCS and in the plant computer (from the BFV).	A failure of this output would result in a CPU failover.	This channel indicates power failure or microprocessor not controlling. Output state: not energized (OK). Power Fail failing to opposite state indicates that the Main CPU is failed.
Channel 3 - Unusable	NA	NA	An analysis of this channel was not found in plant information. Need to confirm why this channel is unusable and whether it could have some failure mode.
Channel 4 - High Power Indication (Output) fails closed	There is no indication of this failure.	There is indication that the DFWCS is in high-power mode. Operation of DFWCS is unaffected, but it may be puzzling to the operators that the DFWCS remains in high-power mode even if the plant is operating in conditions corresponding to low-power mode.	Output state: energized (closed = high power). High Power Indication failing closed indicates that the DFWCS is in high-power mode.
Channel 4 - High Power Indication (Output) fails open	There is no indication of this failure.	Operation of DFWCS is unaffected, but it may be puzzling to the operators that there is no indication that the DFWCS is in high-power mode when the plant is operating in conditions corresponding to this mode.	Output state: energized (closed = high power). High Power Indication failing open does not give indication that the DFWCS is in high-power mode.

Table B.2-3 FMEA of digital backplane (I/O) of main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 5 - Transfer Indication (Output) fails closed	There is no indication of this failure.	There is indication that the DFWCS is transferring between power modes. Operation of DFWCS is unaffected, but it may be puzzling to the operators that the DFWCS remains in a transferring state.	Output state: energized (closed = transferring). Transfer Indication failing closed indicates that the DFWCS is transferring between power modes.
Channel 5 - Transfer Indication (Output) fails open	There is no indication of this failure.	Operation of DFWCS is unaffected, but it may be puzzling to the operators that there is no indication that the DFWCS is transferring when a transfer is taking place.	Output state: energized (closed = transferring). Transfer Indication failing open does not give indication that the DFWCS is transferring between power modes.
Channel 6 - Low Power Indication (Output) fails closed	There is no indication of this failure.	There is indication that the DFWCS is in low-power mode. Operation of DFWCS is unaffected, but it may be puzzling to the operators that the DFWCS remains in low-power mode even if the plant is operating in conditions corresponding to high-power mode.	Output state: energized (closed = low power). Low Power Indication failing closed indicates that the DFWCS is in low-power mode.
Channel 6 - Low Power Indication (Output) fails open	There is no indication of this failure.	Operation of DFWCS is unaffected, but it may be puzzling to the operators that there is no indication that the DFWCS is in low-power mode when the plant is operating in conditions corresponding to this mode.	Output state: energized (closed = low power). Low Power Indication failing open does not give indication that the DFWCS is in low-power mode.
Channel 7 - Bypass Override Indication (Output) fails closed	There is no indication of this failure.	There is indication that the DFWCS is in Bypass Override (BPO) mode. Operation of DFWCS is unaffected, but it may be puzzling to the operators that the DFWCS is in BPO mode when they have not set the DFWCS to operate in this mode.	Output state: energized (closed = BPO mode). Bypass Override Indication failing closed indicates that the DFWCS is in BPO mode.
Channel 7 - Bypass Override Indication (Output) fails open	There is no indication of this failure.	Operation of DFWCS is unaffected, but it may be puzzling to the operators that there is no indication that the DFWCS is in BPO mode when the DFWCS has been set to operate in this mode.	Output state: energized (closed = BPO mode). Bypass Override Indication failing open does not give indication that the DFWCS is in BPO mode.

Table B.2-3 FMEA of digital backplane (I/O) of main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 8 - Deviation Alarm (Output) fails closed	There is no indication of this failure. However, there is indication in the plant computer that the Main CPU detected a deviation.	Operation of DFWCS is unaffected.	Output state: energized (closed = deviation). Deviation Alarm failing closed indicates that the Main CPU detected a deviation.
Channel 8 - Deviation Alarm (Output) fails open	There is no indication of this failure.	If the Main CPU detected a deviation, there is no indication of the detection, but there is a failover to the B/U CPU. If the Main CPU did not detect a deviation, the Main CPU remains in control. In either case, operation of DFWCS is unaffected.	Output state: energized (closed = deviation). Deviation Alarm failing open does not give indication that the Main CPU detected a deviation.
Channel 9 - Transfer Inhibit (Output) fails closed	There is no indication of this failure. However, there is indication in the plant computer that the transfer of power modes is inhibited.	Operation of DFWCS is unaffected.	Output state: energized (closed = transfer inhibited). Transfer Inhibit failing closed indicates that the transfer of power modes is inhibited.
Channel 9 - Transfer Inhibit (Output) fails open	There is no indication of this failure.	If the transfer of power modes is not inhibited, there is no need for indication that the transfer is inhibited. If the transfer of power modes is inhibited, there is no indication in the plant computer that the transfer is inhibited. However, there would be indication that the transfer is inhibited in the PDU. In either case, operation of DFWCS is unaffected.	Output state: energized (closed = transfer inhibited). Transfer Inhibit failing open does not give indication that the transfer of power modes is inhibited.
Channel 10 - Spare Output	Not known	Not known	An analysis of this channel was not found in plant information. Need to confirm whether this channel could have some failure mode.

Table B.2-3 FMEA of digital backplane (I/O) of main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 11 - Positioner Selected (Output) fails closed (fails as is)	There is no indication of this failure.	The signal from the Main CPU will be that the active positioner is A. If the accumulated deviation between the demand from the Main CPU and the position of the MFRV exceeds a setpoint value, the Main CPU will try to put into service the opposite positioner (B). However, the signal from the Main CPU will remain that the active positioner is A. If the accumulated deviation exceeded a setpoint, the positioner A may not be working properly; in this case the Main CPU will not be able to control the MFRV correctly. The impact of this loss of control of the MFRV can vary from a slight deviation of the position of the valve (with respect to the demand from the Main CPU) to the valve fully closing, leading to a reactor trip.	Output state: not energized (we assumed open = B positioner selected). Positioner Selected failing closed indicates that the A positioner is selected as the active positioner. An analysis of this channel was not found in plant information.
Channel 11 - Positioner Selected (Output) fails open (fails to opposite state)	There is no direct indication of this failure. An indirect indication is that the PDU will show that the active positioner changed from A to B.	The signal from the Main CPU will be that the active positioner is B. If the accumulated deviation between the demand from the Main CPU and the position of the MFRV exceeds a setpoint value, the Main CPU will try to put into service the opposite positioner (A). However, the signal from the Main CPU will remain that the active positioner is B. If the accumulated deviation exceeded a setpoint, the positioner B may not be working properly; in this case the Main CPU will not be able to control the MFRV correctly. The impact of this loss of control of the MFRV can vary from a slight deviation of the position of the valve (with respect to the demand from the Main CPU) to the valve fully closing, leading to a reactor trip.	Output state: not energized (we assumed open = B positioner selected). Positioner Selected failing open indicates that the B positioner is selected as the active positioner. An analysis of this channel was not found in plant information.

Table B.2-3 FMEA of digital backplane (I/O) of main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 12 - No Failures in Microprocessor (Output) fails closed	There is no indication of this failure. However, status of the Main CPU will change to failed in PDU.	The Main CPU remains in control. On the other hand, the B/U CPU will receive signal from the Main CPU that the Main CPU failed, but it is expected that the B/U CPU will receive from the MFV the correct status (OK) of the Main CPU. We do not know how the B/U CPU handles this contradictory information.	Output state: not energized (we assumed closed = failed). No Failures in Microprocessor failing closed indicates that the Main CPU failed. This channel is named "Deviation Alarm Status of Other CPU" in plant information.
Channel 12 - No Failures in Microprocessor (Output) fails open	There is no indication of this failure. If the Main CPU failed, it would be annunciated by the PDU and the plant computer.	If the Main CPU is OK, operation of DFWCS is unaffected. If the Main CPU fails, the B/U CPU will receive signal from the Main CPU that the Main CPU is OK, but it is expected that the B/U CPU will receive from the MFV the correct status (failed) of the Main CPU. We do not know at this time how the B/U CPU handles this contradictory information.	Output state: not energized (*we assumed closed = failed*). No Failures in Microprocessor failing open indicates that the Main CPU is OK. This channel is named "Deviation Alarm Status of Other CPU" in plant information.
Channel 13 - No Deviations (from Main CPU to B/U CPU) (Output) fails closed	There is no indication of this failure. However, status of the Main CPU will change to failed in PDU.	The Main CPU remains in control. The incorrect signal may negatively influence the deviation decisions carried out by the B/U CPU. On the other hand, the B/U CPU will receive signal from Main CPU that the Main CPU failed, but it is expected that the B/U CPU will receive from the MFV the correct status (OK) of the Main CPU. We do not know at this time how the B/U CPU handles this contradictory information.	Output state: not energized (we assumed closed = failed). No Deviations from Main CPU to B/U CPU failing closed indicates that the Main CPU failed. This channel is named "CPU Failure Status to Other CPU" and stated unused in plant document.

Table B.2-3 FMEA of digital backplane (I/O) of main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 13 - No Deviations (from Main CPU to B/U CPU) (Output) fails open	There is no indication of this failure. If the Main CPU had deviations, it would be annunciated by the PDU and the plant computer.	If the Main CPU is OK, operation of DFWCS is unaffected. If the Main CPU had deviations, the B/U CPU will receive signal from the Main CPU that the Main CPU is OK, but it is expected that the B/U CPU will receive from the MFV the correct status (failed) of the Main CPU. We do not know at this time how the B/U CPU handles this contradictory information.	Output state: not energized (we assumed closed = failed). No Deviations from Main CPU to B/U CPU failing open indicates that the Main CPU is OK. This channel is named "CPU Failure Status to Other CPU" and stated unused in plant document.
Channel 14 - CPU Level Status to Other CPU (Output) fails closed	There is no indication of this failure.	The Main CPU sends a signal to the B/U CPU indicating that both S/G level signals are invalid. If the Main CPU is OK, operation of DFWCS is unaffected.	Output state: not energized (we assumed closed = invalid). CPU Level Status to Other CPU failing closed indicates that both S/G level signals from the Main CPU are invalid. This channel is stated unused in plant document.
Channel 14 - CPU Level Status to Other CPU (Output) fails open	There is no indication of this failure.	The Main CPU sends a signal to the B/U CPU indicating that both S/G level signals are valid. If the Main CPU is OK, operation of DFWCS is unaffected.	Output state: not energized (we assumed closed = invalid). CPU Level Status to Other CPU failing open indicates that both S/G level signals from the Main CPU are valid. This channel is stated unused in plant document.
Channel 15 - CPU Feedflow/Steamflow Status to Other CPU (Output)	NA	NA	It appears that this channel is connected to the other CPU, though it is not known how the information transmitted through this channel is used by the other CPU. This channel is stated unused in plant document.

Table B.2-3 FMEA of digital backplane (I/O) of main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
		Digital Inputs	
Channel 16 - A/M Status BFV (Input) fails closed (fails as is)	There is no indication of this failure.	Operation of DFWCS is unaffected.	Input state: open circuit (closed = auto). A/M Status BFV failing closed indicates that the BFV is in auto.
Channel 16 - A/M Status BFV (Input) fails open (fails to opposite state)	There is no indication of this failure. However, there would be a discrepancy between the A/M status shown in the PDU and the one shown by the BFV.	The Main CPU would think the A/M status was manual and track instead of control. The valve demand could slowly increase and drift open. Level would be maintained as the main valve compensates for level errors.	Input state: open circuit (closed = auto). A/M Status BFV failing open indicates that the BFV is in manual.
Channel 17 - A/M Status MFV (Input) fails closed (fails as is)	There is no indication of this failure.	Operation of DFWCS is unaffected. However, if the MFV is in manual mode, the Main CPU would "think" that it is controlling the MFRV, but actually would not be controlling it. If the deviation is large enough between the Main CPU's demand and the MFV's demand, the Main CPU would fail, and the B/U CPU would recognize that the MFV is in manual.	Input state: open circuit (closed = auto). A/M Status MFV failing closed indicates that the MFV is in auto.
Channel 17 - A/M Status MFV (Input) fails open (fails to opposite state)	There is no indication of this failure. However, there would be a discrepancy between the A/M status shown in the PDU and the one shown by the MFV.	The Main CPU would think the A/M status was manual and track instead of control. The MFRV would not be controlled, so the S/G level would drift from setpoint. Operators can take manual control based on indications of incorrect S/G level and the discrepancy between the A/M status shown in the PDU and the one shown by the MFV.	Input state: open circuit (closed = auto). A/M Status MFV failing open indicates that the MFV is in manual.

Table B.2-3 FMEA of digital backplane (I/O) of main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 18 - A/M Status FWP (Input) fails closed (fails as is)	There is no indication of this failure.	Operation of DFWCS is unaffected. However, if the FWP is in manual mode, the Main CPU would "think" that it is controlling the FWP, but actually would not be controlling it. If the deviation is large enough between the Main CPU's demand and the FWP controller's demand, the Main CPU would fail, and the B/U CPU would recognize that the FWP is in manual.	Input state: open circuit (closed = auto). A/M Status FWP failing closed indicates that the FWP is in auto.
Channel 18 - A/M Status FWP (Input) fails open (fails to opposite state)	There is no indication of this failure. However, there would be a discrepancy between the A/M status shown in the PDU and the one shown by the FWP.	The CPU would think the A/M status was manual and track instead of control. Pump demand would increase. The main valve would compensate for the pump speed change, giving the operators time to take control as S/G level changed when the main valve could no longer compensate. Operators can take manual control based on indications of incorrect S/G level and the discrepancy between the A/M status shown in the PDU and the one shown by the FWP.	Input state: open circuit (closed = auto). A/M Status FWP failing open indicates that the FWP is in manual.

Table B.2-3 FMEA of digital backplane (I/O) of main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 19 - Reactor Trip (Input) fails closed (fails as is)	There is no indication of this failure.	No effect on DFWCS control. The DFWCS would not be able to detect a reactor trip. If a reactor trip were to subsequently occur, then the DFWCS would not ramp the MFRV shut or run back the FWP demand to minimum speed. The BFRV would open to its post trip position as determined by the feedwater bypass trip set control (1-FC-1211, 1221). The MFRV would shut after the time delay positioning relay times out. When this occurs, the Main CPU will fail due to MFRV deviation. The B/U CPU will take over the automatic control of the DFWCS in low-power mode. The associated FWP demand signal will run back to minimum speed.	Input state: open circuit (closed = not tripped). Reactor Trip failing closed indicates that there is no reactor trip.
Channel 19 - Reactor Trip (Input) fails open (fails to opposite state)	A reactor trip will occur.	During a programmable validation period, no alarm messages will be actuated if a non-validated trip signal is present. After this period, a Reactor Power Large Deviation alarm will be activated on the Vuepoint alarm display alarm and event log entry will result in activation of all trip functions. If there is a concurrent invalid Reactor Power Input, control would be lost. The CPU would erroneously ramp the main valve shut. This failure would result in a reactor trip.	Input state: open circuit (closed = not tripped). Reactor Trip failing open indicates that there is a reactor trip.

Table B.2-3 FMEA of digital backplane (I/O) of main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 20 - Main / B/U CPU Identification (Input) fails closed (fails as is)	There is no indication of this failure.	Plant analysis states that "The Main CPU has no external field connections to fail." It appears that plant analysis concludes that this failure mode cannot occur for the Main CPU. We do not have enough information to assess whether this conclusion is correct. The B/U CPU digital input is grounded. If the external connection were to fail, the B/U CPU would think it was the Main CPU and start to control versus track. As the Main CPU is selected first by the DFWCS controllers, the DFWCS would continue to operate normally. However, the B/U CPU would increase its outputs causing the B/U CPU to fail due to a deviation between the demand and controller output.	Input state: open circuit (closed = main). Main / B/U CPU Identification failing closed indicates that the CPU is the Main CPU.
Channel 20 - Main / B/U CPU Identification (Input) fails open (fails to opposite state)	There is no indication of this failure.	Plant analysis states that "The Main CPU has no external field connections to fail." It appears that plant analysis concludes that this failure mode cannot occur for the Main CPU. We do not have enough information to assess whether this conclusion is correct. The B/U CPU is unaffected.	Input state: open circuit (closed = main). Main / B/U CPU Identification failing open indicates that the CPU is the B/U CPU.
Channel 21 - Turbine Trip (Input) fails closed (fails as is)	There is no indication of this failure.	No effect on DFWCS control. The DFWCS would not be able to detect a turbine trip. If a turbine trip were to subsequently occur, a reactor trip would follow, and the DFWCS would remain in automatic control.	Input state: open circuit (closed = not tripped). Turbine Trip failing closed indicates that there is no turbine trip. Plant document states that this channel is not used for FW control.

Table B.2-3 FMEA of digital backplane (I/O) of main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 21 - Turbine Trip (Input) fails open (fails to opposite state)	There is no indication of this failure. However, a reactor trip will occur. In addition, the turbine and reactor trips are annunciated in the PDU.	Plant information states "When the turbine trip signal is active, a digital signal is sent to the digital feedwater microprocessors which process the signal and causes the feedwater regulating valve to ramp shut. At the same time, control of the bypass feedwater regulating valve is changed from the BFV controller (1-FIC-1105, -1106) to the feedwater bypass trip set control (1-FC-1211, -1221). The trip set control provides a constant output signal to the electro-pneumatic converter (1-I/P-1105, -1106) which will position the bypass valve to provide 5 percent of full load feedwater flow." Accordingly, a reactor trip is expected since the MFRV will ramp shut. The Main CPU will automatically control the DFWCS after the trip.	Input state: open circuit (closed = not tripped). Turbine Trip failing open indicates that there is a turbine trip. Plant information states that this channel is not used for FW control.
Channel 22 - Main CPU Failed (Input) fails closed (fails as is)	A Main CPU failure will be annunciated in the PDU and in the plant computer.	A failover from the Main CPU to the B/U CPU will take place, and the B/U CPU will be in automatic control.	Input state: open circuit (closed = failed). Main CPU Failed (from the MFV) failing closed indicates that the Main CPU failed.
Channel 22 - Main CPU Failed (Input) fails open (fails to opposite state)	There is no indication of this failure.	The Main CPU would "think" that it is OK, regardless of its status. If the Main CPU is OK, operation of DFWCS is unaffected. If the Main CPU is failed, the MFV controller will detect this failure, and the B/U CPU will take control of the DFWCS.	Input state: open circuit (closed = failed). Main CPU Failed (from the MFV) failing open indicates that the Main CPU is OK.

Table B.2-3 FMEA of digital backplane (I/O) of main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 23 - B/U CPU Failed (Input) fails closed (fails as is)	A B/U CPU failure will be annunciated in the PDU and in the plant computer.	The Main CPU believes that the B/U CPU failed. Operation of DFWCS is unaffected. If the Main CPU fails, the MFV controller will detect this failure, but this controller knows that the B/U CPU is OK; hence, the B/U CPU will take control of the DFWCS.	Input state: open circuit (closed = failed). B/U CPU Failed (from the MFV) failing closed indicates that the B/U CPU failed.
Channel 23 - B/U CPU Failed (Input) fails open (fails to opposite state)	There is no indication of this failure.	The Main CPU would "think" that the B/U CPU is OK, regardless of the B/U CPU's status. If the Main CPU is OK, operation of DFWCS is unaffected. If the Main CPU failed, a failover to the B/U CPU would occur. If the B/U CPU is OK, this CPU would take automatic control, and operation of DFWCS is unaffected. If the B/U CPU fails, the controllers would go to manual.	Input state: open circuit (closed = failed). B/U CPU Failed (from the MFV) failing open indicates that the B/U CPU is OK.
Channel 24 - Time Sync (Input)	If the time is reset, it may be shown in the PDU.	No effect.	An external clock synchronization signal causes the time to reset to a pre-arranged value defined in the setpoints. Our understanding is that the input "Time Sync" is associated with this signal. It appears that this input is not used in the control of the DFWCS.
Channel 25 - Neutron Flux # 1 Bypass (Input) fails closed (fails as is)	There is no indication of this failure.	The Main CPU believes the neutron flux # 1 is not bypassed, regardless of the position of the external keyswitch. If the position of the keyswitch is "normal," i.e., not bypassed, operation of DFWCS is not affected. If the position of the keyswitch is "bypass," the Main CPU still will use the neutron flux # 1, possibly resulting in incorrect control of the DFWCS.	Input state: open circuit (we assumed closed = not bypassed). Neutron Flux # 1 Bypass failing closed indicates that this flux is not bypassed. An external keyswitch is used to bypass the neutron flux signal.

Table B.2-3 FMEA of digital backplane (I/O) of main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 25 - Neutron Flux # 1 Bypass (Input) fails open (fails to opposite state)	It appears that the status of the neutron flux signal # 1, i.e., normal or bypass, is not displayed in the PDU. If this assumption is correct, there is no indication of this failure.	The Main CPU believes the neutron flux # 1 is bypassed, regardless of the position of the external keyswitch. The neutron flux signal # 1 will be taken out of service but the other neutron flux signal will be used. If the position of the keyswitch is "bypass," the Main CPU's action is appropriate, so operation of the DFWCS is not affected. If the position of the keyswitch is "normal," i.e., not bypassed, the Main CPU won't use the neutron flux # 1, resulting in a degradation of the input data used by the DFWCS.	Input state: open circuit (we assumed closed = not bypassed). Neutron Flux # 1 Bypass failing open indicates that this flux is bypassed. An external keyswitch is used to bypass the neutron flux signal.
Channel 26 - Neutron Flux # 2 Bypass (Input) fails closed (fails as is)	There is no indication of this failure.	The Main CPU believes the neutron flux # 2 is not bypassed, regardless of the position of the external keyswitch. If the position of the keyswitch is "normal," i.e., not bypassed, operation of DFWCS is not affected. If the position of the keyswitch is "bypass," the Main CPU still will use the neutron flux # 2, possibly resulting in incorrect control of the DFWCS.	Input state: open circuit (we assumed closed = not bypassed). Neutron Flux # 2 Bypass failing closed indicates that this flux is not bypassed. An external keyswitch is used to bypass the neutron flux signal.

Table B.2-3 FMEA of digital backplane (I/O) of main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 26 - Neutron Flux # 2 Bypass (Input) fails open (fails to opposite state)	It appears that the status of the neutron flux signal # 2, i.e., normal or bypass, is not displayed in the PDU. If this assumption is correct, there is no indication of this failure.	The Main CPU believes the neutron flux # 2 is bypassed, regardless of the position of the external keyswitch. The neutron flux signal # 2 will be taken out of service but the other neutron flux signal will be used. If the position of the keyswitch is "bypass," the Main CPU's action is appropriate, so operation of the DFWCS is not affected. If the position of the keyswitch is "normal," i.e., not bypassed, the Main CPU won't use the neutron flux # 2, resulting in a degradation of the input data used by the DFWCS.	Input state: open circuit (we assumed closed = not bypassed). Neutron Flux # 2 Bypass failing open indicates that this flux is bypassed. An external keyswitch is used to bypass the neutron flux signal.
Channel 27 - Positioner Selected (Input) fails closed (fails as is)	There is no indication of this failure.	The Main CPU will keep the A positioner as the active positioner. If the A positioner is OK, operation of the DFWCS is unaffected. If the accumulated deviation between the MFV demand from the Main CPU and the position of the MFRV exceeds a setpoint value, the opposite positioner (B) will be put into service and the Diagnostic Transfer mode will be shifted to lockout. Operation of the DFWCS is unaffected.	Input state: open circuit (we assumed open = B positioner selected). Positioner Selected failing closed indicates that the A positioner is selected as the active positioner. An analysis of this channel was not found in plant information.

Table B.2-3 FMEA of digital backplane (I/O) of main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 27 - Positioner Selected (Input) fails open (fails to opposite state)	There is no direct indication of this failure. An indirect indication is that the PDU will show that the active positioner changed from A to B.	The Main CPU will select the B positioner as the active positioner. If the B positioner is OK, operation of the DFWCS is unaffected. If the accumulated deviation between the MFV demand from the Main CPU and the position of the MFRV exceeds a setpoint value, the opposite positioner (A) will be put into service and the Diagnostic Transfer mode will be shifted to lockout. Operation of the DFWCS is unaffected.	Input state: open circuit (we assumed open = B positioner selected). Positioner Selected failing open indicates that the B positioner is selected as the active positioner. An analysis of this channel was not found in plant information.
Channel 28 - No Failures in Other Microprocessor (Input) fails closed (fails as is)	There is no indication of this failure.	The Main CPU believes that the B/U CPU is OK. Operation of the DFWCS is unaffected. If the Main CPU fails, two cases are possible: 1) If the B/U CPU is OK, a failover to the B/U CPU occurs, so operation of DFWCS is unaffected. 2) If the B/U CPU also is failed, the controllers go to manual, so the operators will have to take manual control.	Input state: open circuit (we assumed closed= no failures). No Failures in Other Microprocessor failing closed indicates that there are no failures in the B/U microprocessor. Plant document names this channel "Deviation Alarm Status from Other CPU."
Channel 28 - No Failures in Other Microprocessor (Input) fails open (fails to opposite state)	There is no indication of this failure.	The Main CPU believes that the B/U CPU is failed. Operation of the DFWCS is unaffected. However, if the Main CPU fails, a failover to the B/U CPU will not occur. The operators will have to take manual control.	Input state: open circuit (we assumed closed= no failures). No Failures in Other Microprocessor failing open indicates that the B/U microprocessor is failed. Plant document names this channel"Deviation Alarm Status from Other CPU."

Table B.2-3 FMEA of digital backplane (I/O) of main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 29 - No Deviations in Other Microprocessor (Input) fails closed (fails as is)	There is no indication of this failure.	The Main CPU believes that there are no deviations in the B/U CPU. Operation of the DFWCS is unaffected. If the Main CPU fails, two cases are possible: 1) If there are no deviations in the B/U CPU, a failover to the B/U CPU occurs, so operation of DFWCS is unaffected. 2) If there are deviations in the B/U CPU, the controllers go to manual, so the operators will have to take manual control.	Input state: open circuit (we assumed closed= no failures). No Deviations in Other Microprocessor failing closed indicates that this status is OK, i.e., there are no deviations in the other microprocessor. Plant document names this channel "CPU Level Status from Other CPU" and states that it is not used.
Channel 29 - No Deviations in Other Microprocessor (Input) fails open (fails to opposite state)	There is no indication of this failure.	The Main CPU believes that there are deviations in the B/U CPU. Operation of the DFWCS is unaffected. However, if the Main CPU fails, a failover to the B/U CPU will not occur. The operators will have to take manual control.	Input state: open circuit (we assumed closed= no failures). No Deviations in Other Microprocessor failing open indicates that this status is failed, i.e., there are deviations in the other microprocessor. Plant document names this channel "CPU Level Status from Other CPU" and states that it is not used.
Channel 30 - Both Level Signals Valid in Other Microprocessor (Input) fails closed (fails as is)	There is no indication of this failure.	The Main CPU believes that both S/G level signals are invalid in the B/U CPU. The status of the level signals in the B/U is used by the Main CPU in its S/G level deviation logic. It appears that the Main CPU would fail itself when it only has one valid level signal and both S/G level signals are invalid in the B/U CPU. If the Main CPU fails due to this reason, there are two cases: 1) if both S/G level signals are valid in the B/U CPU, this CPU takes control, and 2) if both S/G level signals are invalid in the B/U CPU, it appears that the B/U CPU would fail itself, and the operators would have to take manual control.	Input state: open circuit (we assumed closed= invalid). Both Level Signals Valid in Other Microprocessor failing closed indicates that both S/G level signals are invalid in the B/U microprocessor. Plant document names this channel "CPU Steam Flow Status from Other CPU" and states that it is not used.

Table B.2-3 FMEA of digital backplane (I/O) of main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Channel 30 - Both Level Signals Valid in Other Microprocessor (Input) fails open (fails to opposite state)	There is no indication of this failure.	The Main CPU believes that both S/G level signals are valid in the B/U CPU. The status of the level signals in the B/U is used by the Main CPU in its S/G level deviation logic. If the Main CPU and its own S/G level signals are OK, operation of the DFWCS is unaffected.	Input state: open circuit (we assumed closed= invalid). Both Level Signals Valid in Other Microprocessor failing open indicates that both S/G level signals are valid in the B/U microprocessor. Plant document names this channel "CPU Steam Flow Status from Other CPU" and states that it is not used.
Channel 31 - Both Steam Flow and Both FW Flow Signals Valid in Other Microprocessor (Input)	NA	NA	Plant document names this channel"CPU Feedflow Status from Other CPU," and states that this channel is not used. This channel is connected to the other CPU, however, it is unknown how the information transmitted through this channel is used by the other CPU.

Table B.2-4 FMEA of MFV controller (FIC-1111/1121).

Failure mode	Detection of failure mode	Failure effects	Comments
		Loss of analog input (Fail to 0.0 VDC)	
ANI0 (S/G level) Fail to 0.0	No alarm or message. The display will be -116.5".	The display at the MFV controller will be low. The failure can affect the operator's ability to manually control the MFRV.	The signal is for display only.
ANI1 (Valve demand from the main CPU) Fail to 0.0	A deviation alarm will be activated by the MFV controller when the Main CPU demand signal differs from the B/U CPU demand signal by greater than a settable, predetermined setpoint after a settable predetermined time delay. The deviation status will be sent to the BFV controller via Microlink. The BFV controller will activate an alarm to the Plant Computer. The PDI controller will display a "MFV Fail" message.	The controller will initially forward the failed demand signal to the MFRV positioner, PDI controller, and the CPUs of the other S/G. The PDI controller will then detect the signal failure and automatically become the manual controller for the MFV using the old value in its circular buffer. The MFRV must be manually controlled from the PDI controller. The failed signal will be sent to the CPUs of the other S/G, and probably will not affect the FWP speed calculation, because the speed calculation selects the higher of the two flow demand signals, the flow demand signal calculated by the CPUs and the flow demand signal back calculated from the MFV signal received from the other S/G.	The response specified in plant document probably will not take place, because the PDI controller has a scan time of not exceeding 100 milliseconds, while the CPU failover logic has a 1 second delay. The MFV demand signal is also sent to the CPUs of the other S/G and used in the FWP speed calculation of the other S/G.
ANI2 (Valve demand from the B/U CPU) Fail to 0.0	A deviation message is activated, after a settable, predetermined time delay. The deviation message will be sent to the BFV controller through Microlink, and the BFV controller will activate a System Trouble alarm at the Plant Computer.	The MFV controller will continue to forward the signal from the main CPU to its output. No effect on system operation is expected.	

Table B.2-4 FMEA of MFV controller (FIC-1111/1121) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Loss of analog output (Fail to 0.0 VDC)			
ANO0 (Output to the MFRV positioner, PDI controller, and other S/G) Fail to 0.0	The PDI controller will display a "MFV Fail" message.	The demand signal to the MFRV positioner will fail to 0, and the valve will begin to shut. The PDI controller will detect the failure and automatically transfer to the MFV Fail mode. The PDI controller output will then rise to the pre-failure value of the MFV controller output and the MFRV will return to that position. The MFRV must be manually controlled from the PDI controller. The failed signal will initially be sent to the CPUs, of the other S/G, and probably will not affect the FWP speed calculation.	It is not expected that CPU failover would take place, because the PDI controller would take over.
ANO2 (S/G level setpoint output) Fail to 0.0	A system deviation alarm at the Plant Computer will be activated, if a setpoint deviation is detected. The setpoint display at the BFV controller will be low.	The CPUs may detect a setpoint deviation if the deviation setpoint limit is exceeded, and revert to a built-in setpoint.	The operator may use the MFV controller to manually adjust the SG level setpoint.

Table B.2-4 FMEA of MFV controller (FIC-1111/1121) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Digital Inputs Fail Open			
CCI0 (B/U CPU Power Fail or in Test) Fails Open	The controller will indicate that the B/U CPU is failed, and the B/U CPU status will be sent through Microlink to the BFV controller which will activate an annunciator in the control room.	The controller will block the B/U CPU demand signal from its output. System operation will not be affected. The B/U CPU status is sent to the CPUs and could affect the deviation logic of the CPUs.	The signal is normally closed indicating the B/U CPU is OK. It is not clear what the B/U CPU would do when it receives the failure status of its own from the MFV controller. How does the B/U CPU determine its status to send to the Main CPU?
CCI1 (B/U CPU Fail) Fails Open	None.	The controller will not be able to determine the correct status of the B/U CPU. The operation is not affected unless other failures occur.	The signal is normally open indicating the B/U CPU is OK.
CCI2 (Main CPU Power Fail or in Test) Fails Open	The BFV controller will actuate an alarm to the Plant Computer.	Failover from the main CPU to the B/U CPU will take place. The controller will send a Main CPU Fail signal to the CPUs and to the BFV controller through Microlink. The Main CPU Fail signal affects deviation logic of the B/U CPU.	The signal is normally closed indicating the Main CPU is OK. It is not clear what the Main CPU will do when it receives the Main CPU Fail signal from the MFV controller. How does the Main CPU determine its own status to send to the B/U CPU?
CCI3 (Main CPU Fail) Fails Open	None.	The controller will not be able to determine the status of the Main CPU. The operation is not affected unless other failures occur.	The signal is normally open indicating the main CPU is OK.
Digital Input Fail Closed			
CCI0 (B/U CPU Power Fail or in Test) Fails Closed	None.	The controller will not be able to determine the correct status of the B/U CPU. The operation is not affected unless other failures occur.	The signal is normally closed indicating the B/U CPU is OK.

Table B.2-4 FMEA of MFV controller (FIC-1111/1121) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
CCI1 (B/U CPU Fail) Fails Closed	The controller will indicate that the B/U CPU is failed, and the B/U CPU status will be sent through Microlink to the BFV controller which will activate an annunciator in the control room.	The controller will block the B/U CPU demand signal from its output. System operation will not be affected. The B/U CPU status is sent to the CPUs and could affect the deviation logic of the CPUs.	It is not clear what the B/U CPU would do when it receives the failure status of its own from the MFV controller. How does the B/U CPU determine its status to send to the Main CPU? The signal is normally open indicating that the CPU is OK.
CCI2 (Main CPU Power Fail or in Test) Fails Closed	None.	The controller will not be able to determine the correct status of the Main CPU. The operation is not affected unless other failures occur.	The signal is normally closed.
CCI3 (Main CPU Fail) Fails Closed	The BFV controller will actuate an annunciator in the control room indicating the Main CPU Fail.	A failover from the Main CPU to the B/U CPU will take place. The controller will send a Main CPU Fail signal to the CPUs and to the BFV controller through Microlink. The Main CPU Fail signal affects deviation logic of the B/U CPU.	The signal is normally open indicating the main CPU is OK. It is not clear what the B/U CPU would do when it receives the failure status of its own from the MFV controller. How does the Main CPU determine its own status to send to the B/U CPU?

Table B.2-4 FMEA of MFV controller (FIC-1111/1121) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
		Digital Outputs Fail Open	
CCO0 (A/M Status to the Main CPU) Fails Open	The PDU of the Main CPU will display the Transfer Inhibit Alarm. The alarm will also be sent to the Plant Computer.	A manual signal will be sent to the Main CPU, and the Transfer Inhibit Alarm window will be activated. Assuming the Main CPU is in control, and the MFV controller is in auto, the Main CPU will track the MFV controller output. The MFV controller output will be sent from the Main CPU to the MFV controller. The automatic control is effectively lost. This failure may lead to a reactor trip. Normally, upon a reactor trip, the MFRV will be ramped closed and the post trip positioning relay circuit will ensure the MFV demand signal is reduced to zero. It is not obvious that the MFRV will be ramped closed, when the controller is in Manual. The post trip positioning relay circuit should ensure the MFRV be closed.	The signal is normally closed when in auto mode. The response to a reactor trip needs to be confirmed through review of the software.
CCO1 (A/M Status to the B/U CPU) Fails Open	The PDU of the B/U CPU will display the Transfer Inhibit Alarm. The alarm will also be sent to the Plant Computer.	Assuming the Main CPU is in control and the controller is in auto, the operation will not be affected.	The signal is normally closed when in auto mode.
CCO1 (A/M Status to the B/U CPU) Fails Open	The PDU of the B/U CPU will display the Transfer Inhibit Alarm. The alarm will also be sent to the Plant Computer.	Assuming the Main CPU is in control and the controller is in auto, the operation will not be affected.	The signal is normally closed when in auto mode.

Table B.2-4 FMEA of MFV controller (FIC-1111/1121) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
CCO2 (B/U CPU Failed Status to CPUs) Fails Open	There is no direct indication of the failure.	The failed signal will be sent to the Main and B/U CPUs. Assuming the Main CPU is in control, and the controller is in auto, the operation is not affected.	The signal is normally open indicating the B/U CPU is OK.
	If the MFV controller detects failure of the B/U CPU, it generates a local B/U CPU Fail message and sends the status through Microlink to the BFV controller which will actuate an annunciator in the control room.	Assuming the Main CPU is not available, and the B/U CPU is in control, when the failure occurs, the MFV controller should know the correct status of the B/U CPU, and use the MFV demand from the B/U CPU as the output. The system operation will not be affected. If, in addition, the B/U CPU fails, the MFV controller should be able to detect it and transfer to the manual mode.	We assumed that the failure mode is a local failure of the output circuitry, not the controller itself.
CCO3 (Main CPU Failed Status to CPUs) Fails Open	There is no direct indication of the failure.	The failed signal will be sent to the Main and B/U CPUs. Assuming the Main CPU is in control, the operation is not affected.	This signal is normally open indicating the Main CPU is OK.
	If the MFV controller detects failure of the Main CPU, it generates a local Main CPU Fail message and sends the status through Microlink to the BFV controller which will actuate an annunciator in the control room.	Assuming the Main CPU failed while in control, its failure should be detected by the MFV controller, and a failover to the B/U CPU will take place. The incorrect Main CPU status may affect the deviation logic of the B/U CPU.	We assumed that the failure mode is a local failure of the output circuitry, not the controller itself. It is not clear how the B/U CPU reconciles the conflicting information about status of the Main CPU; that is, the MFV controller indicates it is good, while the signal directly from the Main CPU probably indicates it has failed.

Table B.2-4 FMEA of MFV controller (FIC-1111/1121) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
		Digital Outputs Fail Closed	
CCO0 (A/M Status to the Main CPU) Fails Closed	No direct indication of the failure is available.	The failed signal will be sent to the Main CPU. If the Main CPU is in control, the system operation is not affected. If the operator switches the controller to manual, the Main CPU will not recognize it, and continues sending its output to the MFV. As a result, Transfer Inhibit will not be activated. As long as the operator properly takes control, the operation continues until the MFV output deviation from the Main CPU output exceeds the setpoint, in which case a failover from Main CPU to B/U CPU takes place. If the operator fails to manually control MFV, a loss of feedwater control may lead to a reactor trip. Is it possible that a Transfer is initiated with the failure? Upon a reactor trip, the MFRV will be ramped closed and the post trip positioning relay circuit will ensure the MFV demand signal is reduced to zero. The pre-existing failure of the CCO0 does not affect the response to a reactor trip.	The signal is normally closed when in auto mode.

Table B.2-4 FMEA of MFV controller (FIC-1111/1121) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
CCO1 (A/M Status to the B/U CPU) Fails Closed	No direct indication of the failure is available.	If the Main CPU is in control, and the controller is in auto, then the system operation is not affected.	The signal is normally closed when the controller is in auto.
	The deviation will actuate an alarm be sent to the Plant Computer.	If the B/U CPU is in control, and the operator changes the controller to manual, the B/U CPU will not be able to detect it, and the Transfer Inhibit will not be actuated. The B/U CPU continues sending its MFV demand to the controller until the deviation between the MFV demand calculated by the B/U CPU and the MFV controller output exceeds the setpoint, when the B/U CPU will fail and the MFV controller will transfer to manual.	
CCO2 (B/U CPU Failed Status to CPUs) Fails Closed	No direct indication of the failure will be available.	The failed signal will be sent to both CPUs. The MFV controller itself is aware of the correct status of the B/U CPU. If the Main CPU is in control and the controller is in Auto, system operation will not be affected. The failed signal may affect the deviation logic of the Main CPU.	The signal is normally open indicating the B/U CPU is OK. It is not clear how the Main CPU reconciles the conflicting information about status of the B/U CPU; that is, the MFV controller indicates it is failed, while the signal directly from the B/U CPU probably indicates it is good.
		If the B/U CPU is not failed, the Main CPU is failed, and the controller is in Auto, the failure will cause the controllers to switch to manual.	It is assumed that the B/U CPU will fail itself.

Table B.2-4 FMEA of MFV controller (FIC-1111/1121) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
CCO3 (Main CPU Failed Status to CPUs) Fails Closed	The Main CPU failure will be annunciated in the control room.	The failed signal will be sent to both CPUs. The MFV controller itself is aware of the correct status of the B/U CPU. If the Main CPU is in control, and the controller is in Auto, a failover to B/U CPU will take place.	The signal is normally open indicating the Main CPU is OK. It is assumed that the Main CPU will fail itself when it receives the failed signal.
		Loss of Power to Controller	
Loss of power	The MFV controller will be off. The PDI controller will display a "MFV Fail" message.	All analog outputs fail to 0. All digital outputs fail to Open status. The PDI controller will automatically switch to its MFV failure mode of operation and its output will raise to the pre-failure output level of the MFV controller. The MFRV has to be controlled manually using the PDI controller. The CPUs will use the built-in S/G level setpoint and track PDI controller output.	

Table B.2-5 FMEA of BFV Controller (FIC-1105/1106)[3].

Failure mode	Detection of failure mode	Failure effects	Comments
		Loss of Analog Input (Fail to 0.0 VDC)	
ANI0 (Steam generator (S/G) level) fails to 0.0 VDC	The controller will display a value and bargraph of S/G level equal to -116.5". No alarms will be activated.	The DFWCS will continue its operation in automatic mode.	The S/G level signal is used for display only.
ANI1 (Valve demand from the main CPU) fails to 0.0 VDC	During normal high-power mode (i.e., not high-power override mode) the failure is not detected.	The controller will forward the failed demand signal to the BFRV positioner per automatic mode of control. Since the signal corresponds to closure of the BFRV, and the BFRV is already closed, there is no negative effect on the operation of the DFWCS. The failure would manifest when the BFRV should open, but would receive a signal to remain closed. The BFRV is required to open when there is a transfer to 1) low-power mode, or 2) high-power override mode. A deviation would be detected by the Main CPU which will then fail. Subsequently, the backup (B/U) CPU also will fail due to the same reason. Hence, there would be a loss of automatic control of the DFWCS.	The BFRV is normally closed during high-power mode. Input signals to BFV controller are clamped within their range limits. This appears to mean that a failure to 0.0 VDC of ANI1 will be interpreted by the controller as a signal to close the BFRV. The PDI controller also receives the failed demand signal which is held in a circular buffer.

[3]In conducting the FMEA of the BFV Controller, the following assumptions were made:

1. The DFWCS is initially in automatic high-power mode with all system modules normally running, and the Main CPU is controlling the feedwater system. It appears that the plant hazards analysis of the BFV controller assumes that the DFWCS is operating in low-power mode.
2. The BFRV is normally closed during high-power mode. A signal of "0.0" to the BFRV positioner is interpreted in this analysis to result in the BFRV remaining closed.

Table B.2-5 FMEA of BFV Controller (FIC-1105/1106) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
ANI2 (S/G level setpoint) fails to 0.0 VDC	The controller will display a value and bargraph of S/G level equal to -116.5". No alarms will be activated.	The DFWCS will continue its operation in automatic mode.	The level setpoint signal is used for display only.
ANI3 (Valve demand from the B/U CPU) fails to 0.0 VDC	During high-power mode with no additional failures, the failure is not detected (see comments).	Since this signal is not used when the main CPU is controlling, there is no negative effect on the operation of the DFWCS. The failure would manifest when 1) the main CPU fails, and 2) the BFRV is required to open, but would receive a signal to remain closed. The BFRV is required to open when there is a transfer to 1) low-power mode, or 2) high-power override mode. A deviation would be detected by the Main CPU which will then fail. Subsequently, the backup CPU also will fail due to the same reason. Hence, there would be a loss of automatic control of the DFWCS.	Normally, the BFV controller sends the demand from the main CPU to the BFRV positioner. Plant information indicates that the BFV controller will detect the deviation between the main and backup CPU demand signals when they differ by greater than a settable, predetermined setpoint after a settable, predetermined time delay. However, since both signals demand the BFRV to be closed, it is not clear that the failure will be detected.

Table B.2-5 FMEA of BFV Controller (FIC-1105/1106) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
		Loss of Analog Output (Fail to 0.0 mADC)	
ANO0 (Output to the BFRV (positioner)) fails to 0.0 mADC	During normal high-power mode (i.e., not high-power override mode) the failure is not detected.	Since the signal corresponds to closure of the BFRV, and the BFRV is already closed, there is no negative effect on the operation of the DFWCS. The failure would manifest when the BFRV should open, but would receive a signal to remain closed. The BFRV is required to open when there is a transfer to 1) low-power mode, or 2) high-power override mode. A deviation would be detected by the Main CPU which will then fail. Subsequently, the backup CPU also will fail due to the same reason. Hence, there would be a loss of automatic control of the DFWCS.	The BFRV is normally closed during high-power mode. The PDI controller also receives the failed demand signal which is held in a circular buffer.
		Digital Input (Fail Open)	
CCI0 (B/U CPU Power Fail or in Test) fails open	The controller will display the message "B/U FAIL." The "Main or B/U CPU Fail" contact (CCO3) shall close, so a plant computer DFWCS Trouble Alarm will actuate.	The controller will block the B/U CPU BFRV demand signal from its output. As long as the Main CPU is available, system operation will be unaffected and the Main CPU BFRV demand signal will continue to be forwarded to the output. If the Main CPU is not available, the BFV controller will indicate that both CPUs are failed and will revert to Manual mode of operation. The operator will then be required to take action to control S/G level.	Contact CCI0 open means that the Power Fail / Test status of the B/U CPU failed.

Table B.2-5 FMEA of BFV Controller (FIC-1105/1106) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
CCI1 (B/U CPU Fail) fails open	This failure cannot be detected. If the B/U CPU actually fails its watchdog test, the failure will be detected by other controllers that, in turn, will send a "B/U CPU Fail" signal to the BFV controller. In this way, a plant computer DFWCS Trouble Alarm will actuate.	The controller will be unable to determine the watchdog status of the B/U CPU. The controller will assume that the watchdog status is normal. System operation is unaffected unless the B/U CPU actually fails its watchdog test (which will be detected) and the Main CPU becomes unavailable. When this happens, the BFRV demand signal from the failed B/U CPU will be sent to the BFRV positioner. The impact on the DFWCS will vary depending on the nature and severity of the B/U CPU fault.	Contact CCI1 open means that the watchdog status of the B/U CPU is OK.
CCI2 (Main CPU Power Fail or in Test) fails open	The controller will display the message "M FAIL." The "Main or B/U CPU Fail" contact (CCO3) shall close, so a plant computer DFWCS Trouble Alarm will actuate.	The controller will block the Main CPU BFRV demand signal and will forward the "tracking" B/U CPU BFRV demand signal to its output. The Main and B/U CPUs will remain in controlling and tracking modes, respectively. The controller's output will drift upward or downward. This may result in the BFRV opening to some extent.	Contact CCI2 open means that the Power Fail / Test status of the Main CPU failed. Plant information indicates that if the output drifts beyond the deviation limit of the Main CPU, it will fail, and the B/U CPU will assume automatic control of the BFRV. However, it appears that the Main CPU will not fail because the CPU deviation logic for the BFRV demand signal is inhibited during High Power Mode Operations.

Table B.2-5 FMEA of BFV Controller (FIC-1105/1106) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
CCI3 (Main CPU Fail) fails open	This failure cannot be detected. If the Main CPU actually fails its watchdog test, the failure will be detected by other controllers that, in turn, will send a "Main CPU Fail" signal to the BFV controller. In this way, a plant computer DFWCS Trouble Alarm will actuate.	The controller will be unable to determine the watchdog status of the Main CPU. The controller will assume that the watchdog status is normal. System operation is unaffected unless the Main CPU actually fails its watchdog test (which will be detected). When this failure occurs, the BFRV demand signal from the failed Main CPU will be sent to the BFRV positioner. The impact on the DFWCS will vary depending on the nature and severity of the Main CPU fault.	Contact CCI3 open means that the watchdog status of the Main CPU is OK.
Digital input (Fail Closed)			
CCI0 (B/U CPU Power Fail or in Test) fails closed	This failure cannot be detected. If the B/U CPU actually fails or is placed in test, the failure will be detected by other controllers that, in turn, will send a "B/U CPU Fail" signal to the BFV controller. In this way, a plant computer DFWCS Trouble Alarm will actuate.	The controller will be unable to determine the Power Fail / Test status of the B/U CPU. The controller will assume that this status is normal. System operation is unaffected unless 1) the B/U CPU actually fails or is placed in test (either of these events will be detected by the MFV controller which will send a failure signal to B/U CPU), and 2) the Main CPU becomes unavailable. When this happens, the BFRV demand signal from the failed B/U CPU will be sent to the BFRV positioner. The impact on the DFWCS will vary depending on the nature and severity of the B/U CPU fault.	Contact CCI0 closed means that the Power Fail / Test status of the B/U CPU is OK.

Table B.2-5 FMEA of BFV Controller (FIC-1105/1106) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
CCI1 (B/U CPU Fail) fails closed	The controller will display the message "B/U FAIL." The "Main or B/U CPU Fail" contact (CCO3) shall close, so a plant computer DFWCS Trouble Alarm will actuate.	The controller will block the B/U CPU BFRV demand signal from its output. As long as the Main CPU is available, system operation will be unaffected and the Main CPU BFRV demand signal will continue to be forwarded to the output. If the Main CPU is not available, the BFV controller will indicate that both CPUs are failed and will revert to Manual mode of operation. The operator will then be required to take action to control S/G level.	Contact CCI1 closed means that the watchdog status of the B/U CPU is failed.
CCI2 (Main CPU Power Fail or in Test) fails closed	This failure cannot be detected. If the Main CPU actually fails or is placed in test, the failure will be detected by other controllers that, in turn, will send a "Main CPU Fail" signal to the BFV controller. In this way, a plant computer DFWCS Trouble Alarm will actuate.	The controller will be unable to determine the Power Fail / Test status of the Main CPU. The controller will assume that this status is normal. System operation is unaffected unless the Main CPU actually fails or is placed in test. Either of these events will be detected by the MFV controller which will send a failure signal to the Main CPU. When either of these events occurs, the BFRV demand signal from the failed Main CPU will be sent to the BFRV positioner. The impact on the DFWCS will vary depending on the nature and severity of the Main CPU fault.	Contact CCI2 closed means that the Power Fail / Test status of the Main CPU is OK.

Table B.2-5 FMEA of BFV Controller (FIC-1105/1106) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
CCI3 (Main CPU Fail) fails closed	The controller will display the message "M FAIL." The "Main or B/U CPU Fail" contact (CCO3) shall close, so a plant computer DFWCS Trouble Alarm will actuate.	The controller will block the Main CPU BFRV demand signal and will forward the "tracking" B/U CPU BFRV demand signal to its output. The Main and B/U CPUs will remain in controlling and tracking modes, respectively. The controller's output will drift upward or downward. This may result in the BFRV opening to some extent.	Contact CCI3 closed means that the watchdog status of the Main CPU is failed. Plant information indicates that if the output drifts beyond the deviation limit of the Main CPU, it will fail, and the B/U CPU will assume automatic control of the BFRV. However, it appears that the Main CPU will not fail because the CPU deviation logic for the BFRV demand signal is inhibited during High Power Mode Operations.
Digital Output (Fail Open)			
CCO0 (Auto/Manual Status to the Main CPU) fails open	Plant analysis states that "...the Transfer Inhibit Alarm window will be activated." It appears that this "window" refers to an annunciator in the main control room. The CPUs include a digital output to provide indication for the plant computer whenever automatic valve transfer is inhibited.	A Manual status signal will be sent to the DFWCS Main CPU regardless of the actual status of the controller. Thus, the transfer of high power to low power mode is inhibited. If the controller is in Manual mode, or the B/U CPU is controlling S/G level, operation is unaffected. If the controller is in Auto mode, and the main CPU is controlling S/G level, this CPU will "think" that the controller is in Manual mode, so it appears that it (and the B/U CPU) will track the BFRV demand from the controller's output. The controller, in turn, will receive the tracked signal, and forward it to its output. The controller's output will drift upward or downward. This may result in the BFRV opening to some extent.	Contact CCO0 open means that the Auto/Manual Status to the Main CPU indicates Manual. Plant information states that "Main CPU Automatic control of S/G level is lost during this failure." However, the Main CPU will keep Automatic control of the rest of the modules of the DFWCS, so it appears that this CPU can remain in control of S/G level, unless there are additional failures.

Table B.2-5 FMEA of BFV Controller (FIC-1105/1106) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
CCO1 (Auto/Manual Status to the B/U CPU) fails open	Plant analysis states that "...the Transfer Inhibit Alarm window will be activated." It appears that this "window" refers to an annunciator in the main control room. The CPUs include a digital output to provide indication for the plant computer whenever automatic valve transfer is inhibited.	A Manual status signal will be sent to the DFWCS B/U CPU regardless of the actual status of the controller. Thus, the transfer of high power to low power mode is inhibited. If the controller is in Manual mode, or the Main CPU is controlling S/G level, operation is unaffected. If the controller is in Auto mode, and the B/U CPU is controlling S/G level, this CPU will "think" that the controller is in Manual mode, so it appears that it (and the Main CPU if available) will track the BFRV demand from the controller's output. The controller, in turn, will receive the tracked signal and forward it to its output. The controller's output will drift upward or downward. This may result in the BFRV opening to some extent. B/U CPU Automatic control of S/G level is lost during this failure if operating in low power mode.	Contact CCO1 open means that the Auto/Manual Status to the B/U CPU indicates Manual.
CCO2 (Main and B/U CPUs Failed Status) fails open	A status signal of "Both CPUs OK" will be sent to the Fail to Manual Alarm window (annunciator), regardless of the actual status of both CPUs. There is no detection of the contact CCO2 failing open.	If either the Main or the B/U CPU (or both) is OK, then the signal is correct. The operation of the DFWCS is unaffected. If both CPUs failed, the Fail to Manual Alarm annunciator is incorrect. This annunciation ("Both CPUs OK") would fail to alert the operators to take manual control of the DFWCS. The DFWCS would not be controlled neither automatically nor manually. It is not known at this time the consequences of this total loss of control.	Contact CCO2 open means that the Main and B/U CPUs Failed Status is OK, i.e., at least one CPU is not failed

Table B.2-5 FMEA of BFV Controller (FIC-1105/1106) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
CCO3 (Main or B/U CPU Failed Status) fails open	A status signal of "CPU OK" will be sent to the Plant Computer, regardless of the actual status of each CPU. There is no detection of the contact CCO3 failing open.	If both CPUs are OK, the signal is correct. The operation of the DFWCS is unaffected. If the main (B/U) CPU is failed, the signal is incorrect. However, the DFWCS is controlled by the B/U (main) CPU. If both CPUs failed, the Fail to Manual Alarm annunciator (fed from CCO2) would alert the operators to take manual control of the DFWCS.	Contact CCO3 open means that the Main or B/U Failed Status is OK, i.e., both CPUs are OK (not failed).
Digital Output (Fail Closed)			
CCO0 (Auto/Manual Status to the Main CPU) fails closed	This failure mode is not detected.	An Auto status signal will be sent to the DFWCS Main CPU regardless of the actual status of the controller. Operation is unaffected if the controller is in Auto mode, or the B/U CPU is in control. If the controller is in Manual mode, and the Main CPU "thinks" it is in control (due to the erroneous signal), this CPU will attempt to control the BFRV by keeping it closed, even though the BFV controller blocks this CPU's signal when it's in manual mode. Operation of DFWCS is unaffected.	Contact CCO0 closed means that the Auto/Manual Status to the Main CPU indicates Automatic. Plant information indicates that the Main CPU will fail when the Deviation Setpoint is reached. However, it appears that the Main CPU will not fail because the CPU deviation logic for the BFRV demand signal is inhibited during High Power Mode Operations.

Table B.2-5 FMEA of BFV Controller (FIC-1105/1106) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
CCO2 (Main and B/U CPUs Failed Status) fails closed	A status signal of "Both CPUs Failed" will be sent to the Fail to Manual Alarm window (annunciator), regardless of the actual status of both CPUs. This annunciator will actuate	If both the Main and B/U CPUs are OK, the DFWCS is controlled in the Automatic mode. The incorrect signal may be puzzling to the operators. However, the "Main or B/U CPU Failed Status" (from CCO3) indicates that no CPU failed; this indication, in turn, would give a clue to the operators that the incorrect signal is wrong. Nevertheless, the operators may decide to take manual control of the DFWCS. In this way, errors may be executed. If either the Main or the B/U CPU (but not both) failed, the DFWCS is controlled in the Automatic mode by the remaining CPU. Both the "Main and B/U CPUs Failed Status" and "Main or B/U CPU Failed Status" indicate failure. The operators are likely to take manual control of the DFWCS. In this way, errors may be executed. If both CPUs failed, the signal is correct. Operation of the DFWCS is unaffected, except for the failure of the CPUs.	Contact CCO2 closed means that the Main and B/U CPUs Failed Status is failed, i.e., the Main and B/U CPUs are failed.
CCO3 (Main or B/U CPU Failed Status) fails closed	A status signal of "CPU Failed" will be sent to the Plant Computer, regardless of the actual status of each CPU. The Plant Computer DFWCS Trouble Alarm will actuate	If both CPUs are OK, the signal is incorrect. However, the operation of the DFWCS is unaffected. The operators are expected to become aware of the Plant Computer DFWCS Trouble Alarm, and troubleshoot this erroneous signal. If the main or B/U CPU is failed, the signal is correct. The operation of the DFWCS is unaffected, except for the failure of one CPU.	Contact CCO3 closed means that the Main or B/U CPU Failed Status is failed, i.e., at least one CPU failed.

Table B.2-5 FMEA of BFV Controller (FIC-1105/1106) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
		Loss of Power to Controller	
Loss of power to controller	The display of the controller will be off.	ANO0 fail to 0.0 mADC: Since the signal corresponds to closure of the BFRV, and the BFRV is already closed, there is no negative effect on the operation of the DFWCS. The failure would cause a negative impact when the BFRV should open, but would receive a signal to remain closed. The BFRV is required to open when there is a transfer to other power modes, such as low-power mode. CCO0 (CCO1) open: A Manual status signal will be sent to the DFWCS Main (B/U) CPU regardless of the actual status of the controller. CCO2 open: A status signal of "Both CPUs OK" will be sent to the Fail to Manual Alarm annunciator, regardless of the actual status of both CPUs. CCO3 open: A status signal of "CPU OK" will be sent to the Plant Computer, regardless of the actual status of each CPU. Summary: please continue after * in the column "Comments".	The controller's analog output ANO0, will fail to 0.0 mADC, and the controller's digital outputs will fail to Open status. * Summary: Main and B/U CPUs will receive signals that controller is in manual. Thus, the automatic transfer of power modes is inhibited. The BFRV remains closed due to closure signal. The operators cannot take manual control of the BFRV using its controller. To control the BFRV using the PDI controller, the operators have to position the handswitch HS-4516(17)C in the "Bypass Fail" position. The DFWCS is unable to annunciate failures via BFV controller's contacts CCO2 and CCO3.

Table B.2-6 FMEA of FWP controller (FIC-4516/4517).

Failure mode	Detection of failure mode	Failure effects	Comments
		Loss of analog input (Fail to 0.0 VDC)	
ANI0 (Main CPU Speed Demand) Fails to 0.0	The display at the FWP controller will be low. A deviation alarm is activated at the controller when the Main CPU demand signal differs from the B/U CPU demand signal by greater than a set-point, after a time delay. The deviation alarm status will be sent to the BFV controller which will send the alarm to the Plant Computer (PC). The CPU failures and deviation will be annunciated in the control room and sent to the PC.	The failed signal will be sent to the Lovejoy FWP speed controller which will detect the failure and maintain the FWP speed at pre-failure value. The failed signal is sent to the CPUs for tracking, and after a delay will cause the CPUs to be failed due to deviation logic. As a result, the MFV, BFV and FWP controllers will transfer to manual control. It is not likely that the FWP controller can be used to manually control the FWP in this condition.	Need to confirm operation of the Lovejoy controller.

Table B.2-6 FMEA of FWP controller (FIC-4516/4517) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
ANI2 (Bias Signal from Potentialmeter, also sent to the CPUs) Fails to 0.0	The BFV controller will send an alarm to the Plant Computer, upon receipt of the Bias Potential Rate Alarm from the FWP controller.	The failed signal corresponds to a -100% bias. The rate of change of the bias is monitored by the FWP controller, and if a pre-set limit is exceeded, the FWP controller switches to manual mode with the pre-failure value, and a Bias Potential Rate Alarm signal is sent to the BFV controller via the Microlink connection. The BFV controller will then send the alarm to the Plant Computer.	The bias signal is also sent to the Main and B/U CPUs where it is added to the calculated pump speed. It is assumed that the failure is a local failure and a correct signal is sent to the CPUs.
ANI3 (B/U CPU Speed Demand) Fails to 0.0	A deviation alarm at the controller is activated when the main CPU demand signal differs from the B/U CPU demand signal by greater than a settable, predetermined setpoint after a time delay. The deviation alarm is also sent to the BFV controller via Microlink, and the BFV controller will send it to the Plant Computer.	The controller will continue sending the demand from the Main CPU to its output, and the system operation is not affected.	

Table B.2-6 FMEA of FWP controller (FIC-4516/4517) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
		Loss of analog output (Fail to 0.0 VDC)	
ANO0 (Output to the Lovejoy Control System) Fails to 0.0	The CPU failures and deviation will be detected by the BFV controller which will activate an annunciator in the control room and send the alarm to the Plant Computer.	The failed signal will be sent to the Lovejoy FWP speed controller which will detect the failure and maintain the FWP speed at pre-failure value. The failed signal is sent to the CPUs for tracking, and after a time delay will cause the CPUs to be failed due to deviation logic. As a result, the MFV, BFV and FWP controllers will transfer to manual control. A complete loss of automatic control will take place. It is not likely that the FWP controller can be used to manually control the FWP in this condition. The FWP has to be manually controlled using the Lovejoy controller.	Need to confirm operation of the Lovejoy controller.
ANO2 (Bias Potential Excitation) Fails to 0.0 (This failure mode is also applicable to failure to 0.0 of the potential meter.)	The BFV controller will send an alarm to the Plant Computer, upon receipt of the Bias Potential Rate Alarm from the FWP controller.	The failed signal corresponds to a -100% bias. The rate of change of the bias is monitored by the FWP controller, and if a pre-set limit is exceeded, the FWP controller switches to manual mode with the pre-failure value, and a Bias Potential Rate Alarm signal is sent to the BFV controller via the Microlink connection. The BFV controller will then send the alarm to the Plant Computer.	The failed bias signal is also sent to the Main and B/U CPUs where it is added to the calculated pump speed. At the CPU, a FWP bias deviation logic is used to detect out of range condition of the signal. It is probably not going to initiate an alarm, because the bias should be in the expected range. The output of the CPUs will not be used by the FWP controller which is in manual.

Table B.2-6 FMEA of FWP controller (FIC-4516/4517) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
		Digital Inputs Fail Open	
CCI0 (B/U CPU Power Fail or in Test) Fails Open	The controller will indicate that the B/U CPU is failed, and the B/U CPU status will be sent through Microlink to the BFV controller which will activate an alarm to the Plant Computer.	The controller will block the B/U CPU demand signal from its output. System operation will not be affected.	The signal is normally closed indicating the B/U CPU is OK. The B/U CPU status is not sent back to the CPUs. This is true for the BFV controller also.
CCI1 (B/U CPU Fail) Fails Open	None.	The operation is not affected unless other failures occur.	The signal is normally open indicating the B/U CPU is OK.
CCI2 (Main CPU Power Fail or in Test) Fails Open	The BFV controller will actuate an alarm to the Plant Computer.	Failover from the main CPU to the B/U CPU will take place. The controller will send a Main CPU Fail signal to the BFV controller through Microlink. The Main CPU status is not sent back to the CPUs and the CPUs do not know that the controller thinks the Main CPU has failed. The Main CPU continues thinking it is in control, and the B/U CPU continues tracking the output of the controller. Therefore, the FWP demand may remain unchanged, i.e., a loss of automatic control, until the Main CPU detects a deviation and fails itself, and the B/U CPU takes over. It is probably not likely that a reactor trip takes place due to loss of FWP control.	The signal is normally closed indicating the Main CPU is OK. It is assumed that the Main CPU status information to other controllers is correct.
CCI3 (Main CPU Fail) Fails Open	None.	The controller does not have the correct status of the Main CPU. The operation is not affected unless other failures occur.	The signal is normally open indicating the main CPU is OK.

Table B.2-6 FMEA of FWP controller (FIC-4516/4517) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
		Digital Input Fail Closed	
CCI0 (B/U CPU Power Fail or in Test) Fails Closed	None.	The controller does not have the correct status of the B/U CPU. The operation is not affected unless other failures occur.	The signal is normally closed indicating the B/U CPU is OK.
CCI1 (B/U CPU Fail) Fails Closed	The controller will indicate that the B/U CPU is failed, and the B/U CPU status will be sent through Microlink to the BFV controller which will activate an alarm to the Plant Computer.	The controller will block the B/U CPU demand signal from its output. System operation will not be affected unless other failures take place.	The signal is normally open indicating that the CPU is OK.
CCI2 (Main CPU Power Fail or in Test) Fails Closed	None.	The controller does not have the correct status of the Main CPU. The operation is not affected unless other failures occur.	The signal is normally closed.
CCI3 (Main CPU Fail) Fails Closed	The BFV controller will actuate an annunciator in the control room indicating the Main CPU Fail.	Failover from the main CPU to the B/U CPU will take place. The controller will send a Main CPU Fail signal to the BFV controller through Microlink. The Main CPU status is not sent back to the CPUs and the CPUs do not know that the controller thinks the Main CPU has failed. The Main CPU continues thinking it is in control, and the B/U CPU continues tracking the output of the controller. Therefore, the FWP demand may remain unchanged, i.e., a loss of automatic control, until the Main CPU detects a deviation and fails itself, and the B/U CPU takes over. It is probably not likely that a reactor trip takes place due to loss of FWP control.	The signal is normally open indicating the main CPU is OK. It is assumed that the Main CPU status information to other controllers is correct.

Table B.2-6 FMEA of FWP controller (FIC-4516/4517) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
Digital Outputs Fail Open			
CCO0 (A/M Status to the Main CPU) Fails Open	None.	A Manual status signal will be sent to the Main CPU. Assuming the Main CPU is in control, and the FWP controller is in auto, the Main CPU will switch to tracking mode and continue sending its output to the FWP controller, with the controller remaining in Auto. The B/U CPU will continue its tracking also. There will be no Transfer Inhibit Alarm. The automatic control is effectively lost. The output of the controller may drift with no direct indication.	The signal is normally closed when in auto mode. Need to confirm whether or not there will be a Transfer Inhibit Alarm.
CCO1 (A/M Status to the B/U CPU) Fails Open	None.	Assuming the Main CPU is in control and the controller is in auto, the operation will not be affected. There will be no Transfer Inhibit Alarm.	The signal is normally closed when in auto mode. Need to confirm whether or not there will be a Transfer Inhibit Alarm.
Digital Outputs Fail Closed			
CCO0 (A/M Status to the Main CPU) Fails Closed	None.	The system operation is not affected unless other failures occur.	The signal is normally closed when in auto mode.
CCO1 (A/M Status to the B/U CPU) Fails	None.	If the Main CPU is in control, and the controller is in auto, then the system operation is not affected.	The signal is normally closed when the controller is in auto.

Table B.2-6 FMEA of FWP controller (FIC-4516/4517) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
		If the B/U CPU is in control, and the operator changes the controller to manual, the B/U CPU will not be able to detect it. The B/U CPU continues sending its FWP demand to the controller, until the deviation between the FWP demand calculated by the B/U CPU and the FWP controller output exceeds the setpoint, when the B/U CPU will fail and the FWP controller will transfer to manual.	
Loss of Power to Controller			
Loss of power	The FWP controller will be off.	All analog outputs fail to 0. All digital outputs fail to Open status. The failed signal will be sent to the Lovejoy FWP speed controller which will detect the failure and maintain the FWP speed at pre-failure value. The failed signal is sent to the CPUs for tracking, and after a delay will cause the CPUs to be failed due to deviation logic. As a result, the MFV, BFV and FWP controllers will transfer to manual control.	Need to confirm operation of the Lovejoy controller.

Table B.2-7 FMEA of Pressure Differential Indicating (PDI) Controller (PDI-4516/4517)[4].

Failure mode	Detection of failure mode	Failure effects	Comments
		Loss of analog input (Fail to 0.0 VDC)	
ANI0 (Feed Regulating Valve Differential Pressure): Fail to 0.0 VDC	No alarms will be activated. The MFRV D/P bargraph will indicate D/P at 0.0 PSID.	MFRV differential pressure fail to 0.0 PSID. Operation of the DFWCS is not affected.	The signal is for display only.
ANI1 (Main FRV Tracking Signal): Fail to 0.0 VDC	High rate deviation flags will be raised on the PDI controller and reset after a preset time period.	A failed MFRV tracking signal is detected either because the current ANI1 signal is less than -20% or because its change rate is too high for the 0.0V DC input of the PDI ANI1. Upon this detection, the PDI controller thinks that the MFV fails although that ANI1 fails to zero does not mean the failure of MFV. The PDI controller will automatically take over by raising its output to the pre-failure value of the MFV output and enters the manual mode. If only the PDI ANI1 fails, the outputs of the normally running MFV and the PDI controllers will be summed together and should be twice as large as the output of the MFV controller. This will cause the MFRV to open more than designated by the CPU and the problem persists without operator's intervention. Plant analysis indicates that it will likely result in a failed open MFRV and transient. Without operator's action, the MFV demand deviation logic in the CPU software will fail the main CPU. After the B/U CPU takes over, the B/U CPU will fail for the same reason. Pump speed demand on another SG will be affected by the summed signal.	If the PDI ANI1 is the only failure, the operator may place the handswitch (HS) in the MFV Fail position. This will block the MFV output and only the PDI output will be sent to the MFRV. MFRV will be manually controlled by the operator via PDI controller. If, in addition to the PDI ANI1 failure, the MFV ANO0 demand output also fails to zero, the PDI controller will raise its output to the pre-failure MFV output. It is expected that the transfer from the MFV controller to the PDI controller is bumpless in this case. The MFV demand signal to the MFRV will be used by another S/G to calculate the pump speed demand.

[4]The PDI controller is assumed to be in normal mode initially.

Table B.2-7 FMEA of Pressure Differential Indicating (PDI) Controller (PDI-4516/4517) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
ANI2 (Bypass FRV Tracking Signal): Fail to 0.0 VDC	No alarms are generated.	In high power control mode, the BFV controller should output linear 0% demand to the BFRV such that the BFRV is closed. Thus, the ANI2 Fail to 0.0VDC cannot be detected by comparing the failed ANI2 to the previous value held in the circular buffer of the PDI controller. The operation of the DFWCS will not be affected.	In low power control mode, a failed ANI2 is assumed to be detected by comparing the current ANI2 signal to the previously sampled values held in a circular buffer. PDI controller will not take over the BFV controller unless the manual switch HS-4516C/4517C is placed in the position of BFV Fail. Thus, the BFV will be continuously running as normal (output a linear 0% demand to the BFRV) and the operation of the DFWCS will not be affected. However, if the operator mistakenly decides to switch to the BFV Fail position, the summed outputs of the normally running BFV controller and the PDI (pre-failure value of the BFV controller) will open the BFRV wider than designated. Without operator's further action, the BFV demand deviation logic in the CPU software might fail the main CPU depending on the deviation setpoint. After the B/U CPU takes over, the B/U CPU might fail for the same reason.

Table B.2-7 FMEA of Pressure Differential Indicating (PDI) Controller (PDI-4516/4517) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
		Loss of analog output (Fail to 0.0 VDC)	
ANO0 (Output to the MFRV or the BFRV): Fail to 0.0 VDC	During normal operation of the PDI, there is no alarm.	If neither the MFV controller nor the BFV controller fails, this has no effect on the system. If this failure occurs after the PDI controller takes over the MFV controller, the MFRV is expected to fail shut causing a loss of feedwater to the corresponding S/G. In high power mode, the BFRV is normally shut. Thus, if this failure occurs after the operator switches from the BFV controller to the PDI controller, no impacts are expected.	In low power mode, if this failure occurs after the PDI controller takes over the BFV controller, the BFRV is expected to fail shut causing a loss of feedwater to the corresponding S/G.

Table B.2-7 FMEA of Pressure Differential Indicating (PDI) Controller (PDI-4516/4517) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
		Digital Inputs Fail Open	
CCI0 (MFV Control Station Fail Flag - HS): Fail Open	No alarm is generated regarding this failure.	The PDI controller will not be able to know whether HS-4516C/4517C have been placed in the MFV Fail position. If only CCI0 fails and the HS is in the normal or BFV Fail position, then the DFWCS operation is not affected since both MFV and/or BFV controllers are running as usual. If, in addition to CCI0 Fail Open, the MFV also fails, the PDI controller can still detect the MFV failure by comparing ANI1 signal to its previous values held in the circular buffer and automatically takes over the MFV controller. If, in addition to the CCI0 Fail Open, the operator thinks that the MFV controller has a problem even though the MFV demand output does not fail to zero and the rate change of the MFV demand output is not high, and decides to manually switch to the PDI controller, the PDI controller is not able to take over the MFV controller and the MRV will fail shut. It is not certain about the response of the CPUs.	CCI0 Open=MFV OK and Closed=MFV Fail. The state of input CCI0 is decided by the position of HS-4516C/4517C. If the operator places the HS in the MFV Fail position, the output of MFV controller will be blocked and the output of the PDI controller will be sent to the MFRV.

Table B.2-7 FMEA of Pressure Differential Indicating (PDI) Controller (PDI-4516/4517) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
CCI0 (MFV Control Station Fail Flag - HS): Fail Closed	The PDI controller will display a message indicating that the MFV controller is failed although the MFV controller is not.	If only CCI0 fails, then the PDI controller will take over the MFV controller while the MFV is normally running. The output from PDI and the output from the MFV will be added together and sent to the MFRV, which will cause the MFRV to open more than designated by the CPU or the MFV controller. The operator must place HS-4516C/4517C to the Main Fail position in order to clear other contacts on the HS so that manual control of the MFRV using the PDI controller is obtained. Transients will be expected and instability may even be observed without this operator's action. Without operator's action, the MFV demand deviation logic in the CPU software will fail the main CPU depending on the deviation setpoint. After the B/U CPU takes over, the B/U CPU will fail for the same reason. The summed signal will be sent to another S/G to calculate the pump speed demand. The speed demand of the other SG will be affected. If both CCI0 and the MFV controller fail, the PDI controller will automatically take over the MFV controller bumplessly.	CCI0 Open=MFV OK and Closed=MFV Fail. S/G level can only be maintained by the operator's action in this situation.

Table B.2-7 FMEA of Pressure Differential Indicating (PDI) Controller (PDI-4516/4517) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
CCI1 (BFV Control Station Fail Flag - HS): Fail Open	No alarms will be generated.	The PDI controller will be unable to know whether HS-4516C/4517C have been placed in the BFV Fail position. If only CCI1 fails open and the HS is placed in the Normal or MFV Fail position, then the DFWCS operation is not affected since both MFV and/or BFV controllers are running as normal. If, in addition to the CCI1 Fail Open, the BFV also fails, and the operator placed the HS in the BFV Fail position, the output signal to the BFRV is the sum of the BFV output of linear 0% and the PDI output of linear -17%. BFRV might slight open. If, in addition to the CCI1 Fail Open, the BFV also fails, and the operator does not place the HS in the BFV Fail position, the operation of the system is not affected. If, in addition to the CCI1 Fail Open, the MFV controller also fails, the operation of the DFWS system is still not affected since the PDI controller will still take over the MFV controller automatically.	CCI1 Open=BFV OK and CCI1 Closed=BFV Failed The state of input CCI1 is determined by the position of HS-4516C/4517C.

Table B.2-7 FMEA of Pressure Differential Indicating (PDI) Controller (PDI-4516/4517) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
CCI1 (BFV Control Station Fail Flag - HS): Fail Closed	The PDI controller will display a message indicating that the BFV controller is failed although the BFV controller is not.	The CCI1 Fail Closed will make the PDI believe that BFV has failed and the PDI controller should raise the PDI's output to the pre-failure value of the BFV output. However, whether the output of the PDI controller should join the output of the MFV or the BFV is determined by the HS position, which is still at Normal position if the operator has not changed the position of the HS. Therefore, the output of the PDI controller will add to the output of the MFV. Because the BFRV is normally shut in high power control mode, the pre-failure value of the BFV controller held in the circular buffer of the PDI controller should be very small. The impacts of the summed signal on the MFRV may not be significant. Operators action that puts the HS at Bypass Fail position will regain the BFRV control via PDI. Without operator's action, whether the deviation logic will fail the controlling CPU depends on the deviation setpoint of the MFV demand although the deviation is small.	The failure effects in lower power control mode is discussed in plant analysis. This will cause the MFRV to move to the open position and feedwater flow to the affected S/G will increase rapidly. The operator must place HS-4516C/4517C to the Bypass Fail position in order to regain control of the MFRV. This failure mode creates an overfeed situation for the affected S/G. Operator action is required in order to prevent overcooling of the RCS.

Table B.2-7 FMEA of Pressure Differential Indicating (PDI) Controller (PDI-4516/4517) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
CCI2 (Time Sync Input): Fail Open	Inconsistent time may be noticed by comparing operator's clock (independent clock in the control room or even the wristwatch), which was used to define the time stamp table, to the controller clocks.	CCI2 will be sampled periodically. If the CCI2 is closed, the PDI clock will be updated using the pre-defined time stamp. In case of the CCI2 Fail Open, the real-time clock of the PDI controller will not be updated. The clock values of the PDI controller will be propagated to other device controllers for time synchronization via the Microlink every minute. As long as the Microlink is working correctly, a loss of synchronization between the device controllers will not happen. However, the times of device controllers are expected to be inconsistent with operator's clock. The synchronized times at individual controllers are not used in the control task but for the purpose of display only.	CCI2 Open=OK, do not update the PDI clock and CCI2 Closed=Sync, i.e., update the PDI clock. It is assumed that updating the real-time clock of the PDI controller is performed when the system starts running. However, the operator is able to update the PDI clock at any time. Updating the clock of the PDI controller can be either done manually or automatically. Automatic updating is not discussed in the available documentation.
CCI2 (Time Sync Input): Fail Closed	The time associated with the display does not change.	Real-time clock of the PDI controller will be updated using the same user-defined time-stamp table every cycle after sampling the CCI2. If the time-stamp is not changed (which is assumed to be the case here), the time on the PDI (and then the times on other controllers) will remain the same.	CCI2 Open=OK, do not update the PDI clock and CCI2 Closed=Sync, i.e., update the PDI clock.

Table B.2-7 FMEA of Pressure Differential Indicating (PDI) Controller (PDI-4516/4517) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
		Digital Outputs Fail Open	
CCO3 (Loss of Communication Alarm): Fail Open	No alarms are generated.	When CCO3 fails open, if Microlink is working properly, there will be no impact. If the Microlink fails, the loss of communication alarm will not be sent out and the plant computer will not be able to actuate the loss of communication alarm.	CCO3 Open=OK, i.e., the communication is normal. CCO3 Closed=A loss of communications alarm is actuated in the plant computer. Impacts on the operation of the DFWCS are not expected upon the failure of CCO3.
CCO3 (Loss of Communication Alarm): Fail Closed	A loss of communications signal will be sent to the DFWCS Trouble Alarm on the plant computer.	False alarm of a loss of communication will be sent to the plant computer if the Microlink is working properly. The alarm will persist until the failure is fixed.	

Table B.2-7 FMEA of Pressure Differential Indicating (PDI) Controller (PDI-4516/4517) (cont'd).

Failure mode	Detection of failure mode	Failure effects	Comments
		Loss of Power to Controller	
Loss of power that causes: 1. ANO0 fails to 0.0 mADC ; 2. CCO3 fails open; 3. Loss of the PDI controller	No alarms are generated for ANO0 Fail to 0.0 mADC.	If neither the MFV controller nor the BFV controller fails, this has no effect on the system. If this failure occurs after the PDI controller takes over the MFV controller, the MFRV is expected to fail closed causing a loss of feedwater to the corresponding SG. In high power mode, the BFRV is normally shut. Thus, if this failure occurs after the operator switches from the BFV controller to the PDI controller, no impacts are expected.	
	No alarms are generated for CCO3 Fail Open.	When CCO3 fails open, if the Microlink is working properly, there will be no impact. If the Microlink fails, the loss of communication alarm will not be sent out and the plant computer will not be able to actuate the alarm.	
	MFRV dP is no longer displayed on the PDI controller.	PDI fails its functions (display MFRV dP, detect failed MFV and BFV and change modes to take over manually or automatically). If other device controllers are working, this has no impact on the operation of DFWCS except for a loss of communication. Time synchronization will not be performed over the device controllers and FIX numbers from other three device controllers cannot be obtained.	

Table B.2-8 FMEA of Optical Isolator (PB4R)[5].

Failure Mode	Detection of Failure Mode	Failure Effects	Comments
Channel 1 (B/U CPU Watchdog Timer Signal): Fail Closed	No alarms are generated.	The operation of the systems is not affected. If, in addition to this failure, the B/U CPU truly fails in a way such that it can not send out this toggling watchdog timer signal, e.g., it gets hung, and this signal will remain low (contact fails closed), the watchdog timer will never timeout. If the B/U CPU is in control, a failover to the Main CPU will not occur and the system might lose automatic control.	The watchdog timer signal from the B/U CPU toggles every cycle. If the watchdog timer receives the low signal (contact becomes closed) within a preset time period, there will be no timeout, i.e., it is considered that the B/U CPU is working properly. Otherwise (contact becomes open), the watchdog timer will timeout and signal three device controllers.
Channel 1 (B/U CPU Watchdog Timer): Fail Open	Failover alarm will be displayed on the PDU and the B/U CPU failure will be alarmed via annunciator.	The operation of the system will not be affected since the Main CPU is in control. Watchdog timer of the B/U CPU will timeout and initiate a failure of the B/U CPU. The B/U CPU failure status will be sent to the controllers from the watchdog timer. If, in addition to this failure, the Main CPU fails, a failover to the B/U CPU will not occur and the system has to be controlled manually.	The watchdog timer signal from the B/U CPU toggles every cycle. If the watchdog timer receives the low signal (contact becomes closed) within a preset time period, there will be no timeout, i.e., it is considered that the B/U CPU is working properly. Otherwise (contact becomes open), the watchdog timer will timeout and signal three device controllers.
Channel 2 (Main CPU Watchdog Timer): Fail Closed	No alarms are generated.	The operation of the systems is not affected. If, in addition to this failure, the Main CPU truly fails in a way such that it cannot send out this toggling watchdog timer signal, e.g., it gets hung, this signal will remain low (contact fails closed) and the watchdog timer will never timeout. A failover to the B/U CPU will not occur and the system might lose automatic control.	The watchdog timer signal from the Main CPU toggles every cycle. If the watchdog timer receives the low signal (contact becomes closed) within a preset time period, there will be no timeout, i.e., it is considered that the Main CPU is working properly. Otherwise (contact becomes open), the watchdog timer will timeout and signal three device controllers.

[5] PB4R is an optical isolator which performs conversions between electrical signal and optical signal and isolates the electrical coupling between inputs and outputs. Inputs pass through this isolator device and become the outputs. Thus, FMEA of inputs and the corresponding outputs are the same. Failure analysis of PB4R signals is not considered in plant analysis.

Table B.2-8 FMEA of Optical Isolator (PB4R) (cont'd).

Failure Mode	Detection of Failure Mode	Failure Effects	Comments
Channel 2 (Main CPU Watchdog Timer): Fail Open	Failover alarm will be displayed on the PDU and the Main CPU failure will be alarmed via annunciator.	Watchdog timer of the Main CPU will timeout and initiate a failure of the Main CPU. The Main CPU failure status will be sent to the controllers from the watchdog timer. A failover to the B/U CPU will occur although the Main CPU is actually working properly.	The watchdog timer signal from the Main CPU toggles every cycle. If the watchdog timer receives the low signal (contact becomes closed) within a preset time period, there will be no timeout, i.e., it is considered that the Main CPU is working properly. Otherwise (contact becomes open), the watchdog timer will timeout and signal three device controllers.
Channel 3 (One Microprocessor Failed Signal): Fail Closed	See CCO3 Fail Closed in BFV FMEA	See CCO3 Fail Closed in BFV FMEA	Open=No microprocessor failed Closed=One microprocessor failed
Channel 3 (One Microprocessor Failed Signal): Fail Open	See CCO3 Fail Open in BFV FMEA	See CCO3 Fail Open in BFV FMEA	Open=No microprocessor failed Closed=One microprocessor failed
Channel 4 (Both Microprocessor Failed Signal): Fail Closed	See CCO2 Fail Closed in BFV FMEA	See CCO2 Fail Closed in BFV FMEA	Open=Not both microprocessors failed Closed=Both microprocessors failed
Channel 4 (Both Microprocessor Failed Signal): Fail Open	See CCO2 Fail Open in BFV FMEA	See CCO2 Fail Open in BFV FMEA	Open = Not both microprocessors failed Closed = Both microprocessors failed

APPENDIX B.3 FMEA AT LEVEL OF MAJOR-COMPONENT-OF-MODULE OF DFWCS

The detailed FMEA at the level of components of the Main CPU module is shown in Table B.3-1. In the FMEA of the Main CPU module, the Main CPU module was decomposed into individual digital components, which were identified in Chapter 5. FMEA of each component was then conducted to determine the failure impacts on the component, the detectability of the failure, and the associated effects on the Main CPU module, i.e., failure modes of the Main CPU module. It should be noted that the column of "Detection of Failure Mode" in Table B.3-1 indicates the detection by the watchdog timer and software only. Table B.3-2 is the FMEA of a controller at a similar level of detail.

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU

Failure mode	Detection of failure mode	Failure effects (In terms of states of Main CPU)	Comments
Application Software			
The application software on the main CPU seems to be normally running but sends erroneous output	No detection	Undetected Failure of Main CPU	1. Failure rate of the application software is the rate of occurrence of EFC. Further investigation is needed to determine it.
CPU is hung (CPU stops updating output)	Can be potentially detected by WDT if the WDT status is good.	Main CPU Fails to Send WDT the Toggling Signal	1. The WDT does not receive toggling signal and will trip the main CPU if the status of the WDT is normal.
Microprocessor of the Main CPU			
The CPU seems to be normally running but sends erroneous output	No detection	Undetected Failure of Main CPU	1. The failure modes used here are adapted from [RAC 1997]. There, the failure mode is "wrong data word" of a 16-bit CPU. However, the Intel 80586 of this study is a 32-bit processor. 2. Another source of failure modes is Meeldijk [1996]. The failure modes in Meeldijk [1996] include stuck high or low modes (this may correspond to the CPU stops updating outputs) and loss of logic (this may correspond to seemingly normal operation of the CPU).
CPU stops updating output	Can be potentially detected by WDT if the WDT status is good.	Main CPU Fails to Send WDT the Toggling Signal	1. The WDT does not receive a toggling signal and will trip the main CPU.

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects (In terms of states of Main CPU)	Comments
ISA Bus			
Loss of ISA bus	Can be potentially detected by both application software and WDT if the WDT status is good.	Main CPU Fails to Send WDT the Toggling Signal	1. Input and output of the CPU rely on the ISA bus and both the application software and the WDT can potentially detect this loss of the ISA bus. It is assumed the CPU is failed by the WDT if its status is normal. 2. The failure rate of the bus is the sum of failure rates of line/bus driver and receiver. They are considered the major components of the bus.
RAM			
Loss of RAM	Can be potentially detected by WDT if the WDT status is good.	Main CPU Fails to Send WDT the Toggling Signal	1. Application software has to be loaded into RAM in order to run it. Thus, the application software can not run upon a malfunction of RAM. It is assumed that WDT can detect it because the Main CPU does not send out a toggling signal any more.
ROM (BIOS)			
Loss of BIOS	Can be potentially detected by both application software and WDT if the WDT status is good.	Main CPU Fails to Send WDT the Toggling Signal	1. Input and output operation of CPU rely on BIOS routines. Both the software and the WDT can potentially detect this failure. It is likely that the CPU will be failed by the WDT.
Flash Disk			
Loss of Flash Disk	Can be detected by application software.	Main CPU Failed by Application Software (Needs further investigation)	

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects (In terms of states of Main CPU)	Comments
Serial Port			
Loss of Serial Port	No detection.	Main CPU Continues Normal Operation (Needs further investigation)	1. Serial port is used for communication between the Main CPU and PDU. Very likely the serial port is an RS-232 implementation.
Multiplexer			
Loss of all signals	Can be detected by application software.	Main CPU Failed by Application Software	1. Deviation logic will capture the loss of input signals. 2. Only a brief description of failure effects of individual input signals through the multiplexer is shown here. More details of the FMEA of these signals can be found in Appendix B.2.
Loss of one of the signals: S/G 12 Feedwater Temperature	Can be detected by application software.	Main CPU Continues Normal Operation	1. Channel 4 of Analog Backplane A (signal only used during low power operation). 2. Invalidity of the signal will be detected by the Main CPU but the other signal is used and it has no effect on operation. 3. A deviation alarm will be sent to plant computer from the Main CPU.
Loss of one of the signals: S/G 11 Feedwater Temperature	Can be detected by application software.	Main CPU Continues Normal Operation	1. Channel 5 of Analog Backplane A (signal only used during low power operation). 2. Invalidity of the signal will be detected by the Main CPU but the other signal is used and it has no effect on operation. 3. A deviation alarm will be sent to plant computer from the Main CPU.
Loss of one of the signals: S/G 11 FWP A Bias	Can be detected by application software.	Main CPU Continues Normal Operation	1. Channel 6 of Analog Backplane A. 2. It will be detected by the Main CPU. The pump demand will be sent to the FWP regardless. 3. A deviation alarm will be sent to the plant computer from the Main CPU.

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects (In terms of states of Main CPU)	Comments
Loss of one of the signals: S/G 12 MFV Tracking	Can be detected by application software.	Main CPU Continues Normal Operation	1. Channel 7 of Analog Backplane A. 2. Higher MFV tracking signals from both S/Gs will be used to calculate FWP demand. Therefore, this loss of the signal does not affect the FWP demand calculation. 3. There is no direct indication of the failure.
Loss of one of the signals: S/G 12 FWP A Tracking	Can be detected by application software.	Main CPU Failed by Application Software	1. Channel 8 of Analog Backplane A. 2. A deviation larger than the setpoint between the CPU and the controller will cause a failover. If the deviation is not large enough, there is no effect. Here, the deviation is assumed to be large. 3. There is no direct indication of failure. If the Main CPU is failed, there will be an alarm to the plant computer.
Loss of one of the signals: MFRV LVDT #2	Can be detected by application software.	Main CPU Continues Normal Operation	1. Channel 13 of Analog Backplane A. 2. If the accumulation exceeded the MFV-ACCUMULATION setpoint and the Diagnostic Transfer mode is enabled, the opposite positioner will be put in service and the control mode will be shifted to LOCKOUT. 3. PDU and the associated CPU deviation annunciator will be activated.
Loss of one of the signals: MFRV LVDT #1	Can be detected by application software.	Main CPU Continues Normal Operation	1. Channel 14 of Analog Backplane A. 2. If the accumulation exceeded the MFV-ACCUMULATION setpoint and the Diagnostic Transfer mode is enabled, the opposite positioner will be put in service and the control mode will be shifted to LOCKOUT. 3. PDU and the associated CPU deviation annunciator will be activated.
Loss of one of the signals: MFRV Differential Pressure #2	Can be detected by application software.	Main CPU Continues Normal Operation	1. Channel 16 of Analog Backplane A. It appears that a loss of this signal does not affect Main CPU operation. 2. Gooseneck purge related.
Loss of one of the signals: MFRV Differential Pressure #1	Can be detected by application software.	Main CPU Continues Normal Operation	1. Channel 16 of Analog Backplane A. It appears that a loss of this signal does not affect Main CPU operation. 2. Gooseneck purge related.

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects (In terms of states of Main CPU)	Comments
Loss of one of the signals: S/G 11 Level #1	Can be detected by application software.	Main CPU Failed by Application Software	1. Channel 6 of Analog Backplane B. 2. The other input is used for control. 3. Failover will be displayed on PDU. 4. If both S/G 11 Level signals are lost, there will be a loss of auto control. 5. A deviation alarm and failover (if any) will be displayed on PDU.
Loss of one of the signals: S/G 11 Level #2	Can be detected by application software.	Main CPU Failed by Application Software	1. Channel 7 of Analog Backplane B. 2. The other input is used for control. 3. Failover will be displayed on PDU. 4. If both S/G 11 Level signals are lost, there will be a loss of auto control. 5. A deviation alarm and failover (if any) will be displayed on PDU.
Loss of one of the signals: S/G 11 FW Flow #1	Can be detected by application software.	Main CPU Failed by Application Software	1. Channel 8 of Analog Backplane B. 2. The other input is used for control. 3. Failover will be displayed on PDU. 4. If both S/G 11 FW flow signals are lost, a single element control (high power mode) is adopted. Note that the Main CPU is conducting the single element control. If it is in low power mode, Low to High transfer is inhibited. 5. A deviation alarm and failover (if any) will be displayed on PDU.
Loss of one of the signals: S/G 11 FW Flow #2	Can be detected by application software.	Main CPU Failed by Application Software	1. Channel 9 of Analog Backplane B. 2. The other input is used for control. 3. Failover will be displayed on PDU. 4. If both S/G 11 FW flow signals are lost, a single element control (in high power mode) is adopted. Note that the Main CPU is conducting the single element control. If it is in low power mode, Low to High transfer is inhibited. 5. A deviation alarm and failover (if any) will be displayed on PDU.

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects (In terms of states of Main CPU)	Comments
Loss of one of the signals: S/G 11 Main Steam Flow	Can be detected by application software.	Main CPU Failed by Application Software	1. Channel 10 of Analog Backplane B. 2. The other input is used for control. 3. Failover will be displayed on PDU. 4. If both S/G 11 main steam flow signals are lost, a single element control (in high power mode) is adopted. Note that the Main CPU is conducting the single element control. If it is in low power mode, Low to High transfer is inhibited. 5. A deviation alarm and failover (if any) will be displayed on PDU.
Loss of one of the signals: S/G 12 Main Steam Flow	Can be detected by application software.	Main CPU Failed by Application Software	1. Channel 11 of Analog Backplane B. 2. The other input is used for control. 3. Failover will be displayed on PDU. 4. If both S/G 11 main steam flow signals are lost, a single element control (in high power mode) is adopted. Note that the Main CPU is conducting the single element control. If it is in low power mode, Low to High transfer is inhibited. 5. A deviation alarm and failover (if any) will be displayed on PDU.
Loss of one of the signals: Neutron Flux #1	Can be detected by application software.	Main CPU Continues Normal Operation	1. Channel 12 of Analog Backplane B. 2. The other input will be used and control continues. 3. If both inputs are lost, mode transfer is inhibited. 4. A deviation alarm will be sent to plant computer.
Loss of one of the signals: Neutron Flux #2	Can be detected by application software.	Main CPU Continues Normal Operation	1. Channel 13 of Analog Backplane B. 2. The other input will be used and control continues. 3. If both inputs are lost, mode transfer is inhibited. 4. A deviation alarm will be sent to plant computer.
Loss of one of the signals: S/G 11 Level Setpoint	Can be detected by application software.	Main CPU Continues Normal Operation	1. Channel 14 of Analog Backplane B. 2. A deviation between this signal and the setpoint inside the program will occur. If it is larger than a fixed value, the internal level setpoint will be used. Otherwise, there is no impact. 3. A deviation alarm will be sent to plant computer.

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects (In terms of states of Main CPU)	Comments
Loss of one of the signals: S/G 11 BFRV Tracking	Can be detected by application software.	Main CPU Continues Normal Operation	1. Channel 15 of Analog Backplane B. 2. Control continues and BFRV will be closed. There is no impact when it is in high power mode. If it is in low power mode, a failover will occur. 3. There is no alarm.
Loss of one of the signals: S/G 11 MFRV Tracking	Can be detected by application software.	Main CPU Failed by Application Software	1. Channel 16 of Analog Backplane B. 2. The deviation between the Main CPU output and controller feedback will cause a failover for a large deviation. If the deviation is small, the control continues. A large deviation is assumed to be the case here. 3. A deviation alarm and the failover (if any) will be displayed on PDU.
A/D Converter			
All 16 bits stuck at zeros or ones	Can be detected by application software.	Main CPU Failed by Application Software	1. Both A/D and D/A converters are linear ICs. 2. The failure modes are adapted from [Meeldijk 1996]. There, the failure modes of a linear IC are degraded/ improper output, no output, short circuit, open circuit, and drift. 3. Since the A/D converter is shared by all inputs, its loss results in a loss of all inputs.
Random bit failure	No detection.	Undetected Failure of Main CPU	1. Although some of random failures might be detected by the application software, the failures are conservatively assumed to be undetectable.

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects *(In terms of states of* Main CPU)	Comments
D/A Converter			
Output fails high	Can be detected by application software.	Main CPU Failed by Application Software	1. Main CPU will detect this failure via controller feedback if the status of the controller is normal. 2. Failure modes are from [Meeldijk 1996] (see comment 2 of A/D converter). 3. Since the D/A converter is shared by all inputs, its loss results in a loss of all inputs.
Output fails low	Can be detected by application software.	Main CPU Failed by Application Software	1. In addition to failure of the main CPU, the PDI controller will take over the MFV controller.
Drifted output	No detection.	Main CPU Continues Normal Operation	1. Drifted output within a certain range can be coped with.
Demultiplexer			
Loss of all output signals	Can be detected by application software.	Main CPU Failed by Application Software	1. The Main CPU has three analog outputs: one for each controller demand (MFV, FWP, BFV). 2. In addition to the failure of the Main CPU, the PDI controller will take over the MFV controller for this failure mode. 3. Only a brief description of failure effects of individual input signals through the demultiplexer is shown here. More details of the FMEA of these signals can be found in Appendix B.2.
Loss of one of the output signals: Feed Pump Demand	Can be detected by application software.	Main CPU Failed by Application Software	1. Channel 1 of Analog Backplane A. 2. There is no direct indication of this failure. Main CPU deviation (between its demand output and FWP tracking signal) will be sent to plant computer. 3. It seems Lovejoy controller will detect this failure and takes over but details are not available (Appendix B.2).

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects (In terms of states of Main CPU)	Comments
Loss of one of the output signals: Bypass Valve Demand	Can be detected by application software.	Main CPU Continues Normal Operation	1. Channel 2 of Analog Backplane A. 2. The BFV demand signal is normally zero in high power mode. Nothing will happen for loss of the signal. 3. There is no direct indication of this.
Loss of one of the output signals: Main Valve Demand	Can be detected by application software.	Main CPU Failed by Application Software	1. Channel 3 of Analog backplane A. 2. In addition to the failure of the Main CPU, the PDI controller will take over the MFV controller for this failure mode. 3. The PDI controller will display an "MFV fail" message. Main CPU will also activate a deviation message.
Current Loop			
Output current fails high: Feed Pump Demand	Can be detected by application software.	Main CPU Failed by Application Software	1. Channel 1 of Analog Backplane A. 2. A failover will occur due to the large deviation between the CPU demand and the FWP tracking signal. 3. There is no direct indication of this failure. Main CPU deviation (between its demand output and FWP tracking signal) will be sent to plant computer. 4. It seems Lovejoy controller will detect this failure and takes over but details are not available (Appendix B.2). 5. Current loop is a linear device. The failure modes are from [Meeldijk 1996]. 6. It is assumed there is a separate current loop for each output.
Output current fails low: Feed Pump Demand	Can be detected by application software.	Main CPU Failed by Application Software	1. Channel 1 of Analog Backplane A. 2. A failover will occur due to the large deviation between the CPU demand and the FWP tracking signal. 3. There is no direct indication of this failure. Main CPU deviation (between its demand output and FWP tracking signal) will be sent to plant computer. 4. It seems Lovejoy controller will detect this failure and takes over but details are not available (Appendix B.2).

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects (In terms of states of Main CPU)	Comments
Drifted output current: Feed Pump Demand	Can be detected by application software.	Main CPU Continues Normal Operation	1. Channel 1 of Analog Backplane A. 2. According to Appendix B.2, this failure can be compensated by the control algorithm. 3. There is no direct indication of this failure.
Output current fails high: Bypass Valve Demand	Can be detected by application software.	Main CPU Continues Normal Operation	1. Channel 2 of Analog Backplane A. 2. According to Appendix B.2, the CPU deviation logic for the BFV demand signal is inhibited in high power mode. However, if the BFV demand increases, the MFV demand will decrease to cope with this. Therefore, there is at most a transient. 3. There is no direct indication of this failure.
Output current fails low: Bypass Valve Demand	Can be detected by application software.	Main CPU Continues Normal Operation	1. Channel 2 of Analog Backplane A. 2. The BFV demand signal is normally zero in high power mode. Nothing will happen for loss of the signal. 3. There is no direct indication of this failure.
Drifted output current: Bypass Valve Demand	Can be detected by application software.	Main CPU Continues Normal Operation	1. Channel 2 of Analog Backplane A. 2. According to Appendix B.2, a proper setpoint can cope with this. 3. There is no direct indication of this failure.
Output current fails high: Main Valve Demand	Can be detected by application software.	Main CPU Failed by Application Software	1. Channel 3 of Analog Backplane A. 2. Main CPU will detect this failure via MFV controller feedback if the MFV controller status is normal. 3. There is no direct indication of this failure. A deviation alarm will be sent to the plant computer from the Main CPU.
Output current fails low: Main Valve Demand	Can be detected by application software.	Main CPU Failed by Application Software	1. PDI controller will take over the MFV controller. 2. According to Appendix B.2, the PDI controller will take over before the failure of the Main CPU. 3. The PDI controller will display an "MFV fail" message. The Main CPU will give a deviation message.

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects (In terms of states of Main CPU)	Comments
Drifted output current: Main Valve Demand	No detection.	Main CPU Continues Normal Operation	1. According to Appendix B.2, drifted output within a certain range can be compensated. 2. There is no direct indication of this failure.
V_{REF}			
Loss of V_{REF}	No detection	Main CPU Continues Normal Operation	1. According to plant information, V_{REF} is only used to correct for voltage offsets in the input signal path when the system is initialized.
Analog Address Logic			
Unintended address sent out and wrong component selected	Not likely to be detected by the application software and not detectable by the WDT.	Undetected Failure of Main CPU	1. Although some address logic failures might be detected by the application software, this failure mode is conservatively assumed to be undetectable. 2. Address logic is usually called decoder in current digital systems. 3. An analog address logic is a digital device and the failure modes are from [Meeldijk 1996]: stuck high, stuck low, and loss of logic.
Complete loss of analog address logic	Can be potentially detected by both application software and WDT if the WDT status is good.	Main CPU Fails to Send WDT the Toggling Signal	1. CPU should be able to detect the status of analog address logic but can not send out output properly.
Buffer			
Loss of buffer	Can be potentially detected by WDT if the WDT status is good.	Main CPU Fails to Send WDT the Toggling Signal	1. All digital input and output require the buffer.

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects (In terms of states of Main CPU)	Comments
Digital Address Logic			
Unintended address sent out and wrong component selected	Not likely to be detected by the application software and not detectable by the WDT.	Undetected Failure of Main CPU	1. Although some of these failures might be detected by the WDT, this failure mode is conservatively assumed to be undetectable.
Complete loss of digital address logic	Can be detected by WDT if the WDT status is good.	Main CPU Fails to Send WDT the Toggling Signal	1. CPU should be able to detect the status of digital address logic but can not send digital output properly.
Digital Output Module			
Failure to operate of the solid-state switch (Watchdog Timer fails as is)	Can be detected by WDT if the WDT status is good.	Main CPU Fails to Send WDT the Toggling Signal	1. Channel 0 of Digital Backplane: output to WDT. 2. PDU and the plant computer should indicate the failure of the Main CPU. 3. The main component of the digital output module is the solid-state switch. The failure modes, according to [RAC 1997], are Failure to Operate and False Operation.
Failure to operate of the solid-state switch (Power Fail fails as is)	No detection.	Main CPU Continues Normal Operation	1. Channel 2 of Digital Backplane: power failure or the CPU is not controlling. This failure indicates that the Main CPU is OK. Therefore, this failure does not affect the operation of the Main CPU or the system until there is a power failure of the Main CPU. In that case, there will be an undetected Main CPU failure and a loss of auto control. 2. There is no direct indication or detection of this failure.
False operation of the solid-state switch (Power Fail fails to opposite state)	No detection.	Main CPU Failed by Application Software	1. False operation of this switch will indicate a Main CPU power failure and a fail-over should occur. 2. There is no direct indication or detection of this failure. There should be indirect indication from the PDU and the plant computer.

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects (In terms of states of Main CPU)	Comments
Failure to operate of the solid-state switch (High Power Indication fails closed)	No detection.	Main CPU Continues Normal Operation	1. Channel 4 of Digital Backplane: high power indication. It is normally closed indicating the high power mode. 2. This failure does not affect the Main CPU or the system operation. It might, however, affect the operators since this failure indicates the high power mode but it is actually the low power mode. 3. There is no direct indication of this failure.
False operation of the solid-state switch (High Power Indication fails open)	No detection.	Main CPU Continues Normal Operation	1. There is no direct indication of this failure. 2. This failure does not affect the Main CPU or the system operation. It might, however, affect the operators since this failure indicates the low power mode but it is actually the high power mode.
False operation of the solid-state switch (Transfer Indication fails closed)	No detection.	Main CPU Continues Normal Operation	1. Channel 5 of Digital Backplane: transfer indication. It is normally open indicating there is no mode transfer. 2. There is no direct indication of this failure. This failure indicates that the system is transferring between power modes. 3. This failure does not affect the Main CPU or the system operation. It might affect the operators since this failure indicates an ongoing transfer but there is actually no transfer.
Failure to operate of the solid-state switch (Transfer Indication fails open)	No detection.	Main CPU Continues Normal Operation	1. There is no direct indication of this failure. This failure indicates there is no power mode transfer even if a transfer is ongoing. 2. This failure does not affect the Main CPU or the system operation. It might, however, affect the operators since this failure indicates no transfer but there is actually one ongoing.

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects (In terms of states of Main CPU)	Comments
False operation of the solid-state switch (Low Power Indication fails closed)	No detection.	Main CPU Continues Normal Operation	1. Channel 6 of Digital Backplane: low power indication. It is normally open (high power mode). This failure indicates that the system is operating in high power mode. 2. There is no direct indication of this failure. 3. This failure does not affect the Main CPU or the system operation. It might, however, affect the operators.
Failure to operate of the solid-state switch (Low Power Indication fails open)	No detection.	Main CPU Continues Normal Operation	1. There is no direct indication of this failure. 2. This failure does not affect the Main CPU or the system operation. It might, however, affect the operators.
False operation of the solid-state switch (Bypass Override Indication fails closed)	No detection.	Main CPU Continues Normal Operation	1. Channel 7 of Digital Backplane: bypass override (BPO) indication. It is normally open (not in BPO mode). This failure indicates that the system is in a BPO mode. 2. There is no direct indication of this failure. 3. This failure does not affect the Main CPU or the system operation. It might, however, affect the operators.
Failure to operate of the solid-state switch (Bypass Override fails open)	No detection.	Main CPU Continues Normal Operation	1. There is no direct indication of this failure. 2. This failure does not affect the Main CPU or the system operation. It might, however, affect the operators.
False operation of the solid-state switch (Deviation Alarm fails closed)	No detection.	Main CPU Continues Normal Operation	1. Channel 8 of Digital Backplane: deviation alarm and it is normally open, i.e., there is no deviation. This failure indicates that there is a deviation. If this output is closed then there is a deviation. 2. It seems that a fail-over will occur regardless of the state of this output. 3. There is no direct indication. However, the plant computer will indicate that the Main CPU detects a deviation.
Failure to operate of the solid-state switch (Deviation Alarm fails open)	No detection.	Main CPU Continues Normal Operation	1. This failure indicates there is no deviation even if there is. It does not affect operation of the CPUs or the system. 2. There is no direct indication.

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects (In terms of states of Main CPU)	Comments
Failure to operate of the solid-state switch (Transfer Inhibit fails open)	No detection.	Main CPU Continues Normal Operation	1. Channel 9 of Digital Backplane: transfer inhibit. It is normally open, i.e., transfer is not inhibited. 2. This failure indicates that the transfer is not inhibited. 3. Transfer is not considered in this study. 4. There is no direct indication.
False operation of the solid-state switch (Transfer Inhibit fails closed)	No detection.	Main CPU Continues Normal Operation	1. This failure indicates that control mode transfer is inhibited. 2. There is no direct indication. However, the plant computer will indicate that power mode transfer is inhibited.
Failure to operate of the solid-state switch (Positioner Selected fails closed)	No detection.	Main CPU Continues Normal Operation	1. Channel 11 of Digital Backplane: positioner selected, an output to positioner. 2. It is assumed here that positioner A is normally used, i.e., the output is closed. 3. This failure will not affect the operation of the Main CPU. However, if the accumulated deviation between the demand from the Main CPU and the position of the MFRV exceeds a setpoint value, e.g., the positioner A fails, the Main CPU can not switch to positioner B. It might lead to reactor trip.
False operation of the solid-state switch (Positioner Selected fails open)	No detection.	Main CPU Continues Normal Operation	1. This failure will not affect the operation of the Main CPU if the positioner B is in a good state. However, if the accumulated deviation between the demand from the Main CPU and the position of the MFRV exceeds a setpoint value, e.g., the positioner B fails, the Main CPU can not switch to positioner A. It might lead to reactor trip.

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects (In terms of states of Main CPU)	Comments
Failure to operate of the solid-state switch (No Failures in Microprocessor fails open)	No detection.	Main CPU Continues Normal Operation	1. Channel 13 of Digital Backplane: no failures in microprocessor. It is assumed to be normally open, i.e., the Main CPU does not fail. This output goes to the other microprocessor. 2. This failure indicates that the Main CPU is in a good state. This will not affect the operation of the Main CPU and the system. If the Main CPU truly fails, the Backup CPU will be able obtain the Main CPU's status directly from the Main CPU instead of from this output. Thus, it seems that a subsequent failure of the Main CPU will not directly cause a problem. 3. The PDU and plant computer will show the status of the Main CPU. 4. There is no direct indication of this failure.
False operation of the solid-state switch (No Failure in Microprocessor fails closed)	No detection.	Main CPU Failed by Application Software	1. This failure signals that the Main CPU fails and effectively, the MFV controller will block the demand from the Main CPU. The control demand will be from the Backup CPU. Thus, if the Backup CPU is in a good state, it causes a fail-over only and will not affect the operation of the system. 2. If, in addition to the Main CPU failure, the Backup CPU also fails, there will be a loss of auto control. 3. The PDU will show the status of the Main CPU. 4. There is no direct indication of this failure. Failure status of the Main CPU will be displayed by the PDU.

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects (In terms of states of Main CPU)	Comments
Failure to operate of the solid-state switch (No Deviation fails open)	No detection.	Main CPU Continues Normal Operation	1. Channel 14 of Digital Backplane: no deviations. It is normally open, i.e., there is no deviation. This output goes to the other CPU (the Backup CPU). 2. This failure indicates that there is no deviation. Thus, this failure does not affect the operation of the Main CPU or the system. However, if there is truly a deviation, the Backup CPU will not know due to this failure. 3. The Backup CPU can still receive the Main CPU failure (due to deviation) status from the MFV controller. Thus, it is still likely that the deviation will cause a fail-over. 4. There is no indication of this failure. If there is a deviation, the PDU and the plant computer will show the message.
False operation of the solid-state switch (No Deviation fails closed)	No detection.	Main CPU Continues Normal Operation	1. This failure indicates that the Main CPU has a deviation. However, this failure does not cause the Main CPU to fail and thus, the Main CPU remains in control. 2. There is no indication of this failure. The status of the Main CPU will be "Failure" in the PDU display.
Failure to operate of the solid-state switch (CPU Level Status to the Other CPU fails open)	No detection.	Main CPU Continues Normal Operation	1. Channel 15 of Digital Backplane: CPU level status. It is normally open indicating that both SG level signals are valid. This signal goes to the Backup CPU. This failure indicates the validity of signals and will not cause any problem with the operation of the Main CPU and the system. 2. If, in addition to this failure, both of the signal levels are invalid, the Main and the Backup CPUs will fail and there will be a loss of auto control. 3. If, in addition to this failure, the Main CPU fails, it will inform the Backup CPU of its status and the Backup CPU will take over. 4. There is no direct indication of this failure.

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects (In terms of states of Main CPU)	Comments
False operation of the solid-state switch (CPU Level Status to the Other CPU fails closed)	No detection.	Main CPU Continues Normal Operation	1. This failure indicates that both SG level signals are invalid. Since the Main CPU is in a good state and the Backup CPU can validate the signals, it should not cause any problem. 2. There is no direct indication of this failure.
Failure to operate of the solid-state switch	N/A		This is the status of feedflow/steamflow signals to the Backup CPU. This status signal is not used.
False operation of the solid-state switch	N/A		

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects (In terms of states of Main CPU)	Comments
Digital Input Module			
A/M Status BFV fails closed	No detection.	Main CPU Continues Normal Operation	1. Channel 16 of Digital Backplane: A/M Status BFV. It is normally closed, i.e., the BFV is in auto status. It is an input from the BFV controller. 2. It does not cause any problem with the Main CPU or the system. 3. If, in addition to this failure, the BFV controller is in manual mode (which is already defined as failure of the system due to loss of auto control), the Main CPU would still think it is controlling. When the deviation is large, there will be a failover. However, the Backup CPU knows the BFV is in manual mode. 4. The major component of digital input is again a solid-state switch [Eurotherm 2000]. 5. There is no direct indication of this failure.
A/M Status BFV fails open	No detection.	Main CPU Tracking	1. This failure indicates that the BFV is in manual status. The Main CPU would track instead of control and the BFRV may drift open. However, it will be compensated by the MFV. 2. There is no indication of this failure. BFV status displayed on the PDU and the BFV controller is different.
A/M Status MFV fails closed	No detection.	Main CPU Continues Normal Operation	1. Channel 17 of Digital Backplane: A/M Status MFV. It is normally closed, i.e., the MFV is in auto status. It is an input from the MFV controller. 2. It does not affect the operation of the Main CPU. 3. If, in addition to this failure, the MFV controller is in manual mode (which is already defined as failure of the system due to a loss of auto control), the Main CPU would still think it is controlling. When the deviation is large, there will be a failover. However, the Backup CPU knows the MFV is in manual mode. 4. There is no direct indication of this failure.

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects (In terms of states of Main CPU)	Comments
A/M Status MFV fails open	No detection.	Main CPU Tracking	1. This failure indicates that the MFV is in manual status and the Main CPU will track instead of control. The MFRV will drift from setpoint. Eventually, the system will fail without operator actions. 2. There is no indication of this failure. MFV status displayed on the PDU and the MFV controller is different.
A/M Status FWP fails closed	No detection.	Main CPU Continues Normal Operation	1. Channel 18 of Digital Backplane: A/M Status FWP. It is normally closed, i.e., the FWP is in auto status. It is an input from the FWP controller. 2. This failure indicates that the FWP is auto. Operation of the Main CPU or the system is not affected. 3. If, in addition to this failure, the FWP controller is actually in manual status (which is already defined as system failure due to loss of auto control), the Main CPU is still thinking it is controlling the FWP. When the deviation is large enough, there will be a failover. After the failover, the Backup CPU will know the correct status of the FWP controller. 4. There is no direct indication of this failure.
A/M Status FWP fails open	No detection.	Main CPU Tracking	1. This failure indicates that the FWP controller is in manual status. The Main CPU will track instead of control. The pump demand may increase but it is expected to be compensated by the MFV controller. 2. There is no direct indication of this failure. However, the FWP controller status displayed by the PDU and the FWP controller is different.

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects (In terms of states of Main CPU)	Comments
Reactor Trip fails closed	No detection.	Main CPU Continues Normal Operation	1. Channel 19 of Digital Backplane: Reactor Trip. It is normally closed, i.e., there is no reactor trip. It is an input from post reactor trip position relay. 2. This failure does not affect the system. However, the Main CPU can not detect whether there is a reactor trip. 3. If, in addition to this failure, there is a reactor trip, the Main CPU will fail and a failover will occur (Appendix B.2). 4. There is no direct indication of this failure.
Reactor Trip fails open	No detection.	Main CPU Continues Normal Operation	1. This failure indicates that there is a reactor trip. Trip functions will be activated after certain time period (Appendix B.2). 2. A reactor trip will occur.
Main/Backup CPU Identification fails closed	No detection.	Main CPU Continues Normal Operation	1. Channel 20 of Digital Backplane: Main/Backup CPU Identification. It is normally closed, i.e., the pre-selected CPU is the Main CPU. This failure mode can not occur to the Main CPU (Appendix B.2). It is a pre-selected input. 2. However, the failure will make the Backup CPU think that it is the Main CPU and start controlling. The Backup CPU will fail due to deviation. 3. There is no indication of this failure.
Main/Backup CPU Identification fails open	No detection.	Main CPU Continues Normal Operation	1. This failure can not occur to the Main CPU (Appendix B.2). 2. It does not affect the Backup CPU. 3. There is no indication of this failure.

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects (In terms of states of Main CPU)	Comments
Turbine Trip fails closed	No detection.	Main CPU Continues Normal Operation	1. Channel 21 of Digital Backplane: Turbine Trip. It is normally closed, i.e., there is no turbine trip. It is an input from the turbine relay. 2. This failure does not affect the operation of the system. The system can not detect the occurrence of turbine trip. 3. If, in addition to this failure, there is a turbine trip, a reactor trip will follow and the system remains in automatic control. 4. There is no indication of this failure.
Turbine Trip fails open	No detection.	Main CPU Continues Normal Operation	1. This indicates that there is a turbine trip. 2. The MFRV will be shut down but the Main CPU remains in automatic control. 3. There is no indication of this failure except a reactor trip. The PDU will display the trip events.
Main CPU Failed fails closed	No detection.	Main CPU Failed by Application Software	1. Channel 22 of Digital Backplane: Main CPU Failed. It is normally open, i.e., the Main CPU is not failed. It is an input from the MFV controller. 2. This failure indicates that the Main CPU is failed, a failover is expected. 3. Main CPU failure will be displayed by the PDU and the plant computer.
Main CPU Failed fails open	No detection.	Main CPU Continues Normal Operation	1. This failure indicates that the Main CPU is OK even if it is not. Thus, it does not affect the operation. 2. If, in addition to this failure, the Main CPU fails, its true status can be detected by the MFV controller and there will be a fail over to the Backup CPU. 3. There is no indication of this failure.

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects (In terms of states of Main CPU)	Comments
Backup CPU Failed fails closed	No detection.	Main CPU Continues Normal Operation	1. Channel 23 of Digital Backplane: Backup CPU Failed. It is normally open, i.e., the Backup CPU is OK. It is an input from the MFV controller. 2. This failure indicates that the Backup CPU failed. It does not affect the operation of the system. 3. If, in addition to this failure, the Backup CPU fails, it is still OK since the Main CPU is controlling. 4. If, in addition to this failure, the Main CPU failed, there will be a failover because the MFV knows the true status of the Backup CPU. 5. The PDU and the plant computer will show the failure status of the Backup CPU.
Backup CPU Failed fails open	No detection.	Main CPU Continues Normal Operation	1. This failure indicates that the Backup CPU is OK. It does not affect the operation of the system. 2. If, in addition to this failure, the Backup CPU failed, it is still OK because the Main CPU will be controlling. 3. If, in addition to this failure, the Main CPU fails, there will be a failover to the Backup CPU. 4. There is no indication of this failure.
Time Sync	N/A	N/A	Not used. It is an input from the external clock.
Neutron Flux #1 Bypass fails close	No detection.	Main CPU Continues Normal Operation	1. Channel 25 of Digital Backplane: Neutron Flux #1 Bypass. It is normally closed, i.e., the flux signal is not bypassed. It is an input from the keyswitch. 2. This failure indicates that the flux #1 is not bypassed. If the external keyswitch is "normal," it does not affect the operation of the system. 3. However, even if the external keyswitch is "bypass," it does not seem that the operation will be affected (Appendix B.2). 4. There is no indication of this failure.

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects (In terms of states of Main CPU)	Comments
Neutron Flux #1 Bypass fails open	No detection.	Main CPU Continues Normal Operation	1. This failure indicates that the flux #1 is bypassed even if the external keyswitch is "normal." It does not affect the operation of the system. 2. There is no indication of this failure.
Neutron Flux #2 Bypass fails closed	No detection.	Main CPU Continues Normal Operation	1. Channel 26 of Digital Backplane: Neutron Flux #2 Bypass. It is normally closed, i.e., the flux signal is not bypassed. It is an input from the keyswitch. 2. This failure indicates that the flux #2 is not bypassed. If the external keyswitch is "normal," it does not affect the operation of the system. 3. However, even if the external keyswitch is "bypass," it does not seem that the operation will be affected (Appendix B.2). 4. There is no indication of this failure.
Neutron Flux #2 Bypass fails open	No detection.	Main CPU Continues Normal Operation	1. This failure indicates that the flux #2 is bypassed even if the external keyswitch is "normal." It does not affect the operation of the system. 2. There is no indication of this failure.
Positioner Selected fails closed	No detection.	Main CPU Continues Normal Operation	1. Channel 27 of Digital Backplane: Positioner Selected. It is normally closed, i.e., positioner A is selected. It is an input from the positioner. 2. This failure indicates that the positioner A is selected as the active positioner. It does not affect the operation of the system. 3. There is no indication of this failure.
Positioner Selected fails open	No detection.	Main CPU Continues Normal Operation	1. This failure indicates that positioner B is the active positioner. It does not affect the operation of the system. 2. There is no direct indication of this failure. The PDU will show the active positioner.

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects (In terms of states of Main CPU)	Comments
No Failures in Other Microprocessor fails closed	No detection.	Main CPU Continues Normal Operation	1. Channel 28 of Digital Backplane: No Failures in Other Microprocessor. It is normally closed, i.e., the other microprocessor is not failed. It is an input from the other microprocessor. 2. This failure indicates that the other microprocessor is OK. It does not affect the operation of the Main CPU. 3. There is no indication of this failure.
No Failures in Other Microprocessor fails open	No detection.	Main CPU Continues Normal Operation	1. This failure indicates that the other microprocessor is failed. It does not affect the operation of the Main CPU. 2. If, in addition to this failure, the Main CPU failed, there will be no failover. A loss of automatic control occurs. 3. There is no indication of this failure.
No Deviation in Other Microprocessor fails closed	No detection.	Main CPU Continues Normal Operation	1. Channel 29 of Digital Backplane: No Deviations in Other Microprocessor. It is normally closed, i.e., there is no deviation in the other microprocessor. It is an input from the other CPU. 2. This failure indicates that the other CPU is OK. It does not affect the operation of the system. 3. There is no indication of this failure.
No Deviation in Other Microprocessor fails open	No detection.	Main CPU Continues Normal Operation	1. This failure indicates that the other CPU has a deviation. It does not affect the operation of the system. 2. If, in addition to this failure, the Main CPU failed, the failover will not occur and there will be a loss of automatic control. 3. There is no indication of this failure.
Both Level Signals Valid in Other Microprocessor fails closed	No detection.	Main CPU Continues Normal Operation	1. Channel 30 of Digital Backplane: Both level signals are valid in the other microprocessor. It is normally open, i.e., both signals are valid. 2. This failure indicates that level signals in the Backup CPU are invalid. It does not affect the operation of the Main CPU. 3. There is no indication of this failure.

Table B.3-1 FMEA at Level of Components of DFWCS Modules: Main CPU (cont'd).

Failure mode	Detection of failure mode	Failure effects (*In terms of states of* Main CPU)	Comments
Both Level Signals Valid in Other Microprocessor fails open	No detection.	Main CPU Continues Normal Operation	1. This failure indicates that both level signals in the Backup CPU are valid. It does not affect the operation of the system. 2. There is no indication of this failure.
Both Steam Flow and Both FW Flow Signals Valid in Other Microprocessor	N/A	N/A	Not used.

Table B.3-2 FMEA of F&P 53MC5000 Controller

Failure Mode	Detection of Failure Mode	Failure Effects	Comments
Loss of Power Supplies			
Loss of +15V Supply	Display goes blank	Loss of analog input and output channels due to loss of D/A and analog comparator powered by +15V. The RS-232 serial communication link is lost. Loss of power supply to a level shifter (from 5V to 12V digital signals) in the display circuit.	Loss of D/A has the same effects on the analog input and output channels.
Loss of -15V Supply	No indication of failure unless the configuration port is in use.	Loss of RS-232 serial communication link.	This failure has no effect on the controlled application (unless the port is used to receive control information which is usually not the case).
Loss of +26V	Not detectable unless the analog output is monitored using extra circuits.	Loss of analog outputs.	If analog outputs are monitored, then the failure of the monitoring circuits and/or the sampling and holding circuits has the same failure effects.
Loss of +5V	The display probably goes blank	Most of functions will be lost. The drive lines to the display will be lost. The display was designed to blank if the input data stream stops to protect the display. Analog outputs will drift.	Display interface design is not tested yet for this situation.
Loss of +80V	Display goes blank	Only display is affected.	
Loss of -110V	Display goes blank	Only display is affected.	

Table B.3-2 FMEA of F&P 53MC5000 Controller (cont'd).

Failure Mode	Detection of Failure Mode	Failure Effects	Comments
ROM Error	The processor performs a checksum test of ROM at startup or reset. If the ROM fails at this time, it will be detected. Otherwise, it is not detectable.	When the processor is running, the failure effects of a ROM error is not predictable.	ROM is usually used to store programs and constants used in the programs developed by vendors. There is no continuously executing memory error detection algorithm.
RAM Error	The processor performs a test of RAM at startup. If the RAM fails at this time, it will be detected. Otherwise, it is not detectable.	When the processor is running, the failure effects of a RAM error is not predictable.	For 53MC5000, there is no background running process that performs read/write test (however, 53MC2000 does) and there is no continuously executing memory error detection algorithm.
PAL (Programmable Array Logic) Error	Unknown	Some functions provided possibly by user-written F-TRAN software stored in RAM will not be available.	
Computational Errors	Unknown	Unknown except that CPU outputs will be incorrect.	Plant information indicates that the risk of computational problems caused by the math library is low.
Initialized Data Errors	Unknown	If the presence of wrong data is noticed before the operation, there will be no impacts. Otherwise, severity of impacts on the DFWCS depends on individual errors of data.	Re-initialization of the software should fix the non-default data errors since all non-default database values are hardcoded into the various software modules used in the controllers.
Loss of Lithium Battery	Error will be obvious only after the external power supply is also lost.	Loss of all functions of the processor due to a loss of program and database if the external power supply is unavailable. Otherwise, the processor will continue to run.	

Table B.3-2 FMEA of F&P 53MC5000 Controller (cont'd).

Failure Mode	Detection of Failure Mode	Failure Effects	Comments
Loss of RS-485 Serial Communications Interface	N/A	53MC5000 will not be able to receive data upon problem with the receive circuit. 53MC5000 will not be able to transmit data upon problem with the transmit circuit.	From available documentation, RS-485 serial communication is not used in the DFWCS (RS-232 is used for development only).
Loss of RS-485 Jabber	A DFWCS trouble alarm will be actuated.	53MC5000 does not use the communication network to transmit control related information. The failure effects could be losses of warning messages or time.	It is assumed that RS-485 Jabber indicates the Microlink communication link.
Loss of PWR_ON Signal	Flashing display.	Watchdog time out due to loss of reset signal from PWR_ON. The processor will halt. The control task stops updating outputs and the display task stops updating display memory. All the contact outputs will be at "Open" state. Analog outputs will go to zero mA.	
Run-time Error: FIX (Function Index) 0	The F & P logo appears on the display.	The control program stops and the inputs are still measured. The processor continues to run but the control outputs will not be updated. The display memory is no longer updated. The contact and the analog outputs stay the same.	The FIX number determines the functionality of the controller by selecting various control strategies and operations once the FIX number is entered into the database System Module Function Index data point B000.
Failure of Display or Display Circuitry	Blank or weird display.	The processor continues to run. Both control task and display task continues to operate. The contact outputs and the analog outputs are set by control.	

Table B.3-2 FMEA of F&P 53MC5000 Controller (cont'd).

Failure Mode	Detection of Failure Mode	Failure Effects	Comments
Failure of Digital Input/Output	Generally not detectable.	There could be a generic failure mode of IC, where an IC fails and causes a momentary short across the +5V supply. This generic failure mode will cause a reset of the processor. The total reset time of a 53MC5000 unit is around 1.2 to 1.5 seconds. During this time period, the analog and digital outputs of the controller will go to off state (typically ground). The failures effects on the actuators are difficult to estimate and depend on individual designs.	The failures effects on the actuators are difficult to estimate and depend on individual designs.
ASIC Failure	Failure of the DISP-controller or the DISP-memory is visible in the display.	Loss of display.	
	Failure of the core block, the processor, will cause flashing display.	The ASIC design is hierarchical and partitioned into different blocks. If the processor fails, the watchdog timer, the DISP-controller and the DISP-memory are still working properly, which will produce a flashing display.	
	Failure of the processor's display interface or the DISP-memory (1K dual-ported RAM) is detected if the heartbeat pixel does not flash.	Loss of display.	
Clock Reference	The display will freeze and the display is possibly destroyed.	All functions of the ASIC will stop. The core block (8051 processor) will fail to execute software. Both the watchdog timer and display will freeze. Analog outputs will drift because the watchdog timer has not expired.	

Table B.3-2 FMEA of F&P 53MC5000 Controller (cont'd).

Failure Mode	Detection of Failure Mode	Failure Effects	Comments
Analog Output Drift	Very difficult to detect.	Under certain conditions, e.g., a total ASIC failure, the output sample and hold circuits will no longer be refreshed but the watchdog timer will not act to pull the outputs to zero. Analog outputs will drift to unknown values in unknown directions.	In critical applications, analog outputs should be monitored using input channels and a bypass circuit should be used in case of the analog output failure.

B.4 REFERENCES

Eurotherm Ltd., Using 2604/2704 Fixed Digital I/O, Technical Information, No. TIN 137, pg. B-113, 2000.

Meeldijk, V., *Electronic Components Selection and Application Guidelines*, John Wiley & Sons, 1996.

Reliability Analysis Center (RAC), "Electronic Parts Reliability Data," EPRD-97, 1997.

APPENDIX C

OTHER METHODS FOR MODELING DIGITAL SYSTEMS

TABLE OF CONTENTS

		Page
C.1	Introduction	C-1
C.2	ET/FT Method	C-2
C.3	"Traditional" Discrete-State Continuous-Time Methods	C-3
C.4	Other Methods	C-5
C.5	References	C-6

C.1. INTRODUCTION

As discussed in the main report, the Event Tree/Fault Tree (ET/FT) and Markov methods were selected for further exploration of their capabilities and limitations. Several other methods that may be useful for developing and quantifying reliability models of digital systems are discussed in this appendix. While it is not practical to further explore all of these methods as part of the current project, some of them may warrant further attention if other studies demonstrate their capability and practicality.

Traditional methods are defined here as those that are well-established but that do not explicitly model the interactions between the plant system being modeled and the plant physical processes, nor the timing of these interactions. This definition affords a somewhat blurry, and hence, debatable boundary between traditional and dynamic methods mainly because some of the former have been extended to address interactions and timing. Accordingly, it is possible that the identified set of traditional methods is incomplete, or that some of these methods might be considered dynamic. Nevertheless, an attempt was made to include all methods that can be considered traditional.

To facilitate their analysis, these methods were grouped into three major categories:

1. ET/FT methods. This category (Subsection C.2) includes the traditional ET/FT method that has been commonly used in the U.S. nuclear power industry and in other countries and industries. Other methods that are variations or refinements of this basic method also are included: "Dynamic" FT [Dugan 1992], GO-FLOW [Gately 1978, Matsuoka 1988], and Binary Decision Diagrams [Rauzy 1993].

2. "Traditional" discrete-state continuous-time methods. This category (Subsection C.3) addresses the classical state-transition method, i.e., Markov modeling, and two of its variations: Petri Nets and the "SINTEF PDS" method [Hauge 2006b]. SINTEF is a Norwegian acronym for "Scientific and Industrial Research at the Norwegian Institute of Technology," and PDS is an acronym for "reliability of computer-based safety systems."

3. Other methods. Subsection C.4 presents other methods considered worthwhile for discussion. These methods include Bayesian Belief Networks (BBNs) [Eom 2004, Dahll 2002], a "conditional risk model" for including quantitative software failure probabilities in a probabilistic risk assessment (PRA) proposed by the National Aeronautics and Space Administration (NASA) [NASA 2002a], Reliability Prediction Methods (RPMs) [DOD 1995, RAC PRISM, Telcordia 2001], discrete event simulation (DES) [Siu 1994], and a simplified analytical method used for the (PRA) of a Japanese Advanced Boiling Water Reactor (ABWR) [Sugawara 2000].

C.2 ET/FT METHODS

Traditional ET/FT Method

In the United States, probabilistic models of nuclear power plant (NPP) systems are typically developed using FTs, and integrated into an overall plant response model using ETs. Thus, all the PRA models developed as part of the Individual Plant Examinations (IPEs) were constructed by this traditional ET/FT method, which is also currently used by the worldwide NPP PRA community. The ET/FT method already has been used in several applications for modeling digital systems.

The ET/FT method has proven its flexibility. Its building blocks can be used for constructing models of the relevant features of the many varied systems of a NPP. Besides the nuclear industry, it long has been used by the computer, aerospace and chemical industries in a wide variety of applications.

The ET/FT method is a powerful tool for reliability analysis of complex systems. It is well-suited to identify detailed plant failure modes, represented by combinations of failures of system components, by combining the system models into an overall model of the NPP. The method can quantitatively evaluate the detailed failure modes of the plant.

The traditional ET/FT method has several limitations. It does not explicitly treat the timing of events in accident sequences, but only accounts for them implicitly (i.e., through the specific events included in the ETs and their order of occurrence). Similarly, it only considers interactions with plant processes implicitly and approximately (primarily through the system success criteria). Also, while it may be possible to model most or all types of digital system fault tolerant features using the traditional ET/FT method, the process for doing so may not be straightforward.

"Dynamic" FT

The dynamic FT method of Dugan et al. [1992] is a straightforward extension of the traditional FT analysis that introduces special gates to handle the order in which events occur. For example, a functional-dependency gate models a network element as a trigger event whose failure isolates the connected components. A cold-spare gate allows the cold spare on standby to have zero failure rate. A priority AND gate generates an output only if the inputs occur in a particular sequence, and a sequence-enforcing gate forces events to occur in a particular order.

The dynamic FT method requires that the model be transformed into a Markov model in order to be quantified [NASA 2002b], due to its modeling of the order in which events take place. Therefore, it can be considered a tool for constructing a Markov model of digital systems, and is subject to the limitations of the Markov method. Its practicality in modeling of digital systems has not been demonstrated in full-scale applications.

GO-FLOW

The GO methodology originally was developed for modeling complex systems [Gately 1978]. Later, it was modified into the GO-FLOW methodology [Matsuoka 1988] and used in many applications in Japan [Matsuoka 1998]. GO-FLOW is a phased-mission methodology that models a system in terms of signals that are the input and output of different operators. The signals can represent time duration, flow in a pipe, an electrical current in a circuit, a demand for an operation, and events such as power being available. They take on a discrete number of states or values, with the

associated probabilities. The different operators perform various probabilistic operations on the input signals to generate the output signals and their probabilities.

GO-FLOW can be seen as a success-oriented complement of the fault tree method. However, Siu [1994] pointed out that this method has some drawbacks: "GO-FLOW is not designed to easily provide structural information regarding the system (i.e. the minimal cut sets), nor are importance measures computations provided. This information, routinely provided by fault tree and event tree analyses, is quite important when trying to decide how to reduce the system risk/unavailability. Further, GO-FLOW does not directly treat common cause failures, which frequently dominate the unavailability of redundant systems..."

Binary Decision Diagrams

An ET/FT logic model is qualitatively solved by obtaining the minimal cut sets (MCSs). Depending on the model's size and complexity, the computer resources and time required for its solution can be substantial. A method was established for generating minimal cut sets called the Binary Decision Diagram (BDD) (see for example [Rauzy 1993]) to reduce these requirements. The logic model is first transformed to a BDD from which the MCSs can be directly obtained. A BDD is a directed acyclic graph. All paths through the BDD end in one of two states, either system failure or success. Since the BDD method is another way of solving an ET/FT model, it does not offer improved techniques for modeling a digital system.

C.3 "TRADITIONAL" DISCRETE-STATE CONTINUOUS-TIME METHODS

The "Traditional" Markov Method

The Markov method has been used both for modeling NPP systems and digital systems. It is a flexible method because it can explicitly model the different states that a system can reach during its operation, regardless of the type of system.

The Markov method can be a powerful tool for analyzing digital systems because such systems may be able to detect failures and change their own configuration during operation. In addition, the system may be repaired and thus return to its original configuration. The Markov method allows explicit detailed modeling of these reconfigurations or states of the system. In addition, it treats failure and repair times explicitly.

The fault tree models of systems can be considered simplified representations of the associated Markov model. In the early development of PRA, the Markov method was used to model systems [Papazoglou 1978], but integrating it into traditional plant PRA models is not straightforward.

The limitations of the Markov method include the fact that the number of states can grow very rapidly, usually due to the complexity of the system, thereby making the analysis of the model very difficult. Furthermore, integrating it with an ET/FT model, i.e., the usual risk model for a NPP, is not straightforward because the Markov method models the time dependence of component failures and repairs, adopting a format not directly compatible to that of the ET/FT method. Similar to the ET/FT method, the traditional Markov method considers interactions with plant processes only implicitly in an approximate way, i.e., primarily in terms of which plant systems must operate for given plant conditions, and in terms of system success criteria.

Petri Nets

The Petri-net method has been used to model the behavior of software [Lawrence 1993]. Its advantage is its ease of modeling the behavior of a control system. It can be used in modeling changes in the system's state caused by triggering events. For modeling computer-based systems, the advantage of Petri nets is that, in general, they can model the state- and time-dependent behavior of these systems. In addition, they can encompass finite-state machines, concurrent (parallel) processes, software (data flow), communication protocols, synchronization control, and multiprocessor systems. Analyses of a Petri-net model will reveal the presence or absence of safety properties, such as hazardous conditions, system deadlock, or unreachable states. The method was employed by Goddard [1996] as a tool for a Failure Mode and Effects Analysis (FMEA) to identify failures and their effects. While its use has not been common in the nuclear industry, it recently was applied to the reliability modeling of computer-based systems [Malhotra 2002]. The Stochastic Petri net is a Petri net that includes probabilistic information. This model is similar to a Markov model and must be converted into the latter to be solved. Hence, the value of Petri nets for modeling digital systems may not be different than that of a Markov model. However, given the potential usefulness of Petri nets, it may be worthwhile to explore them further.

The SINTEF PDS Method

The PDS (Norwegian acronym for "reliability of computer-based safety systems") handbooks [Hauge 2006a and 2006b] published by the SINTEF have guidance on reliability analyses of computer- based safety systems. The SINTEF work involved international oil companies, digital-system vendors, and engineering companies. It represents an adaptation of the Reliability Block Diagram (RBD) method specified in Appendix B of Part 6 of IEC standard 61508 [IEC 61508] for the Norwegian oil industry. The data handbook [Hauge 2006a] is based on earlier work on Offshore Reliability Data Handbooks (OREDA), e.g., [OREDA 2002], that covered data collected at offshore platforms. International oil companies participated in the OREDA project.

IEC 61508 specifies the qualitative requirements for different safety integrity levels (SILs) of the software and hardware of safety-related systems, along with quantitative reliability goals in terms of failure rates and failure probabilities. IEC 61508 also describes different methods for quantifying the reliability of the systems. The "safety-related" systems are those that perform safety functions during certain abnormal operating conditions. There is a disconnect between the qualitative requirements and the quantitative goals. Appendix B of Part 6 of IEC 61508 has example analyses using Markov-type models with simplified analytical solutions. The intent is that the quantitative models potentially can be used to demonstrate that the hardware satisfying the qualitative requirements of a given SIL also meets the quantitative goals. However, little guidance is available on how to use the qualitative requirements to develop quantitative reliability models.

The PDS method [Hauge 2006b] essentially is the same as that of IEC 61508. Accordingly, it models a system in terms of a Markov model and solves the model by introducing simplifying assumptions, such that analytical expressions can be derived, rather than requiring the solution of differential equations. Hence, the PDS method is a simplified version of Markov modeling. One limitation of the PDS method is that it considers that common cause failure (CCF) dominates subsystem unavailability and, therefore, fails to account for independent random failures of components. As such, it ignores the combinations of failures of individual components from different subsystems that could trigger overall system failure. Also, the estimates of coverages and hardware failure fractions of the dangerous failure rates, and the beta factors, are based on expert judgment, which is not documented.

C.4 OTHER METHODS

The following methods also were considered worthwhile discussing:

1. BBNs have been used mainly for modeling some specific aspects of systems, such as software reliability.

2. An approach for including software failure probabilities in a logic model is described in the NASA PRA Procedures Guide [NASA 2002a]. According to page 5 of the NASA PRA procedures guide, this approach has not been fully endorsed by NASA for agency-wide application because the field of quantitative software reliability is considered to be too immature for a specific approach to be recommended. This approach is not considered a general method for modeling digital systems because, as mentioned in Section 11.3.3.1 of the NASA PRA Procedures Guide, it only refers to traditional and dynamic methods as approaches for identifying conditions that may trigger software failures. This approach does not specify a method for quantifying software reliability.

3. Reliability Prediction Methods (RPMs), such as PRISM [RAC Manual] and the Military Handbook 217 [DOD 1995], are methods for estimating the failure rate of a circuit board in terms of its components' failure rates. Their main use is as a source of probabilistic data, and they are described in Chapter 8.

4. Discrete event simulation DES is a Monte Carlo simulation method for solving stochastic models that involve continuous process states and discrete system states. Siu [1994] pointed out that "...discrete event simulation has the ability to treat all the issues of interest in dynamic accident scenario analysis: hardware state, process variables, operator state of mind, scenario history, and time..." Hence, this method satisfies the definition of a dynamic method (mentioned above). Since it is not "traditional," it is not considered to be within the scope of this project.

5. A simplified analytical method was employed for the reliability modeling of the Digital Protection Systems (DPSs) of a Japanese ABWR [Sugawara 2000]. Logic models of the DPSs, such as fault trees, were not developed. Instead, three formulas were used for assessing the probability of failure of software and the probability of the failure of a DPS due to the CCF of identical components in all divisions. The basic method for determining the unavailabilities of the DPSs consists of independently evaluating the hardware failures and software failures of a DPS. The following are the main considerations: The failure due to the CCF of the same component in all divisions causes the system to fail; a β factor is used to account for the CCF of hardware; and the β factor equals one for software, i.e., a complete dependency is modeled. Thus, the simplified method accounts for software failures in that they are explicitly included in the logic model. However, while a model for quantifying software reliability is employed, its technical basis is not provided. Due to the extent of the simplifications associated with the simplified analytical method as described here, this method is not considered further in this study.

C.5 REFERENCES

Dahll, G., Gran, B.A., and Liwang, B., "Decision Support for Approval of Safety Critical Programmable Systems," NEA/CSNI/R(2002)1/Vol 1.

Department of Defense, "Reliability Prediction of Electronic Equipment," MIL-HDBK-217F, Notice 2, February 18, 1995.

Dugan, J. B., et al., "Dynamic Fault-Tree Models for Fault-Tolerant Computer Systems," IEEE Transactions on Reliability, Vol. 41, N. 3, September, 1992.

Eom H., Son, H.S., Kang, H., and Ha, J., "A Study on the Quantitative Reliability Estimation of Safety-Critical Software for Probabilistic Safety Assessment, " Fourth American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Controls and Human-Machine Interface Technologies, Columbus, Ohio, September 2004.

Gately, W. Y., and Williams, R.L., "GO Methodology- System Reliability Assessment and Computer Code Manual", Electric Power Research Institute, NP-766, May 1978.

Goddard, P. L., "A Combined Analysis Approach to Assessing Requirements for Safety Critical Real-Time Control Systems," Hughes Aircraft Company, IEEE Proceedings Annual Reliability Maintainability Symposium, 1996.

Hauge, S., Langseth, H. and Onshus, T., "Reliability Data for Safety Instrumented Systems," PDS Data Handbook, 2006 Edition, SINTEF, April, 2006a.

Hauge, S., Hokstad, P., Langseth, H, and Oien, K., "Reliability Prediction Methods for Safety Instrumented Systems: PDS Method Handbook," SINTEF, April, 2006b.

International Electrotechnical Commission, "Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems," IEC 61508, Parts 1 to 7, various dates.

Lawrence, J.D., "A Software Reliability and Safety in Nuclear Protection Systems," Lawrence Livermore National Laboratory, NUREG/CR-6101, November 1993.

Malholtra, M., and Trivedi, K.S., "Dependability Modeling Using Petr-Net, " IEEE Transaction on Reliability, vol. 44, vol. 3, 1995, and Trivedi, Probability and Statistics with Reliability, Queing, and Computer Science Applications, 2002.

Matsuoka, T., and Kobayashi, M., "GO-FLOW: A New Reliability Analysis Methodology", Nuclear Science and Engineering: 98, 64-78, 1988.

Matsuoka, T., et al., "An Application of the GO-FLOW Methodology Evaluation of Component Cooling Water System for New Type Marine Reactor", Proceedings of PSAM4, New York City, USA, 1998.

NASA, "Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners," Version 1.1, NASA, August 2002a.

NASA, "Fault Tree Handbook with Aerospace Applications," Version 1.1, NASA, August 2002b.

OREDA Participants, "Offshore Reliability Data Handbook," 2002, 4th Edition.

Papazoglou, I.A., and Gyftopoulos, E. P., Markovian Reliability Analysis Under Uncertainty with an Application on the Shutdown System of the Clinch River Breeder Reactor," NUREG/CR-0405, September, 1978.

Rauzy, A., "New algorithms for fault tree analysis, Reliability Engineering & System Safety," Volume 40, pages 203-211, 1993.

Reliability Analysis Center (RAC), "PRISM User's Manual, Version 1.4, "Prepared by Reliability Analysis Center Under Contract to Defense Supply Center Columbus.

Siu, N., "Risk assessment for dynamic systems: An overview," Reliability Engineering & System Safety, Volume 43, Issue 1, pages 43-73, 1994.

Sugawara, M., "Reliability Analysis of Digital Safety Systems used in an Advanced Boiling Water Reactor (ABWR) Plant in Japan," slides of a presentation to the 5th COOPRA Steering Committee Meeting in Tokyo, Japan, December 4, 2000.

Telcordia, "Reliability Prediction Procedure for Electronic Equipment," SR-332 Issue 1, May 2001.

NRC FORM 335
(9-2004)
NRCMD 3.7

U.S. NUCLEAR REGULATORY COMMISSION

BIBLIOGRAPHIC DATA SHEET

(See instructions on the reverse)

1. REPORT NUMBER
(Assigned by NRC, Add Vol., Supp., Rev., and Addendum Numbers, if any.)

NUREG/CR-6962
BNL-NUREG-80141-2008

2. TITLE AND SUBTITLE

Traditional Probabilistic Risk Assessment Methods for Digital Systems

3. DATE REPORT PUBLISHED

MONTH	YEAR
October	2008

4. FIN OR GRANT NUMBER

N6413

5. AUTHOR(S)

T.L. Chu, G. Martinez-Guridi, M. Yue, J. Lehner, P. Samanta

6. TYPE OF REPORT

Technical

7. PERIOD COVERED *(Inclusive Dates)*

8. PERFORMING ORGANIZATION - NAME AND ADDRESS *(If NRC, provide Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address; if contractor, provide name and mailing address.)*

Brookhaven National Laboratory
P.O. Box 5000
Upton, NY 11973

9. SPONSORING ORGANIZATION - NAME AND ADDRESS *(If NRC, type "Same as above"; if contractor, provide NRC Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address.)*

Division of Risk Analysis
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, D.C. 20555

10. SUPPLEMENTARY NOTES

Alan Kuritzky, Project Manager

11. ABSTRACT *(200 words or less)*

The U.S. Nuclear Regulatory Commission is conducting research to determine the existing capabilities and limitations of using traditional reliability modeling methods to develop and quantify digital system reliability models, with the ultimate goals of (1) supporting the use of information on the risks of digital systems in nuclear power plant (NPP) regulatory decisions and (2) including models of digital systems into NPP probabilistic risk assessments (PRAs). This report documents the initial tasks of this research, which include (1) development of a set of desirable characteristics for reliability models of digital systems, (2) identification of two traditional reliability methods for trial application to two example digital systems, and (3) laying out the process by which the first reliability study of an example digital system will be performed. The results of this initial workk indicate that the traditional methods of Event Tree/Fault Tree and Markov modeling appear to be useful for the PRA of digital systems, but also reveals limitations in the state-of-the-art for modeling digital systems using traditional PRA methods and where additional research and development are needed. The report offers other insights and conclusions obtained during this work and proposes activities to be conducted when applying these methods to the first reliability study. Note, in keeping with the principal objective stated above, this project generally does not involve advancements in the state-of-the-art, such as the estimation of risk from software faults.

12. KEY WORDS/DESCRIPTORS *(List words or phrases that will assist researchers in locating the report.)*

Digital system, digital instrumentation and control, digital I&C, probabilistic risk assessment ,PRA, digital system PRA, reliability models, digital system risk, digital I&C risk, digital system modeling

13. AVAILABILITY STATEMENT

unlimited

14. SECURITY CLASSIFICATION

(This Page)

unclassified

(This Report)

unclassified

15. NUMBER OF PAGES

16. PRICE

PRINTED ON RECYCLED PAPER

Printed
on recycled
paper

Federal Recycling Program

UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, DC 20555-0001

OFFICIAL BUSINESS