Rolls-Royce

Rolls-Royce
5959 Shallowford Road, Suite 511
Chattanooga, Tennessee 37421
www.rolls-royce.com

April 4, 2013

U.S. Nuclear Regulatory Commission
Document Control Desk
11555 Rockville Pike
Rockville, MD 20852

ATTENTION: To whom it may concern

SUBJECT: Rolls-Royce Response to Request for Additional Information (Set 3) Re: Rolls-Royce Civil Nuclear "***SPINLINE 3*** Digital Safety Instrumentation and Control Platform" Topical Report (TAC NO. ME3600)

REFERENCES: (1) Project Number 0773: ***SPINLINE 3*** Digital Safety Instrumentation and Control Platform (TAC No. ME3600)
(2) Letter, Joseph J. Holonich (NRC) to Mark Burzynski (Rolls-Royce), "Request for Additional Information Re: Rolls-Royce Civil Nuclear "***SPINLINE 3*** Digital Safety Instrumentation and Control Platform" Topical Report (TAC NO. ME3600)", March 28, 2013

NRC provided a request for additional information regarding the review of the Rolls-Royce ***SPINLINE 3*** Digital Safety Instrumentation and Control Platform Topical Report. The Rolls-Royce response to this request for additional information is provided by an enclosure to this letter.

Rolls-Royce hereby submits the following documents in connection with the referenced NRC project:

Document Title	Rolls-Royce Document Number	Versions: Proprietary (P), Non-proprietary (NP)	Notes
Response To Request For Additional Information (Set 3) - ***SPINLINE 3*** Digital Safety Instrumentation and Control Platform - Project No. 773	N/A	P	New document
Response To Request For Additional Information (Set 3) - ***SPINLINE 3*** Digital Safety Instrumentation and Control Platform - Project No. 773	N/A	NP	New document

Rolls-Royce considers some of the material contained in the response to be proprietary and requests that the proprietary documents be withheld from public disclosure. In accordance with 10 CFR 2.390, "Public inspections, exemptions, requests for

T007
NRR

withholding", an affidavit is enclosed identifying the specific portions of the above documents that are proprietary and the basis for making that determination. Proprietary and non-proprietary versions of the response to the request for additional information are provided.

All documents are submitted electronically and in hard copy.

If you have any questions related to this submittal, please contact me at 423-756-9730 extension 12 or by e-mail at mark.j.burzynski@ds-s.com.

Sincerely,

Mark J. Burzynski
US I&C Licensing Manager
Rolls-Royce

Rolls-Royce

Rolls-Royce Civil Nuclear US
5959 Shallowford Road, Suite 511
Chattanooga, Tennessee 37421
Tel: (423) 756-9730
www.rolls-royce.com

Affidavit

STATE OF TENNESSEE)

)

COUNTY OF HAMILTON)

1. In accordance with 10 CFR 2.390, "Public inspections, exemptions, requests for withholding", Rolls-Royce requests withholding from public disclosure of the documents listed in Table 1, which is attached to this affidavit.
2. I am familiar with the criteria applied by Rolls-Royce to determine whether certain Rolls-Royce information is proprietary. I am familiar with the policies established by Rolls-Royce to ensure the proper application of these criteria.
3. As required by 10 CFR 2.390, Rolls-Royce has included in Table 1 the following information:
   - Identity of the document or part sought to be withheld;
   - Declaration of the basis for proposing the information be withheld, encompassing considerations set forth in § 2.390(a);
   - Specific statement of the harm that would result if the information sought to be withheld is disclosed to the public; and
   - Locations in the documents of all information sought to be withheld.
4. As required in § 2.390(b)(4); Rolls-Royce wishes to note that the request for withholding from public disclosure applies to pages that contain commercially sensitive information that Rolls-Royce normally discloses only under a Non-Disclosure Agreement (NDA). This commercially sensitive information is not available in public sources and is the type of information customarily held in confidence by Rolls-Royce and our competitors.

5. Rolls-Royce is transmitting this information to NRC in confidence.

6. As noted in Table 1, release of this information in a public forum could cause harm to Rolls-Royce by revealing trade secrets and/or commercially sensitive design and operational details and technical processes related to designing, building, and/or operating a ***SPINLINE 3*** digital safety instrumentation and control system.

7. As Rolls-Royce US I&C Licensing Manager, I have been specifically delegated responsibility for reviewing the information sought to be withheld, and I am authorized to apply for its withholding on behalf of Rolls-Royce.

8. The foregoing statements are true and correct to the best of my knowledge, information, and belief.

Mark J. Burzynski

Mark J. Burzynski
US I&C Licensing Manager
Rolls-Royce

Sworn to and subscribed before me
this 4th day of April, 2013

Whitney D. Hickman

Notary Public

My commission expires: July 21st, 2015

Table 1 - Documents requested for withholding from public disclosure

Document Title	Document Number	Part of document sought to be withheld from public disclosure	Basis for proposing the information be withheld, encompassing considerations set forth in § 2.390(a)	Specific statement of the harm that would result if the information sought to be withheld is disclosed to the public	Location(s) in the document of all information sought to be withheld (Notes 1 and 2)
Response To Request For Additional Information (Set 3) - ***SPINLINE 3*** Digital Safety Instrumentation and Control Platform - Project No. 773	N/A	Portions of response, as marked by brackets **[[ ]]**.	Trade secrets and / or commercial information as per § 2.390(a)(4)	Rolls-Royce would be harmed by disclosure of aspects of the identified commercially sensitive information, which is of value to a competitor because it would enable them to make direct comparisons between the platform design and communication features for their ***SPINLINE 3*** safety I&C platform.	Some or all of the responses to questions 70, 71, 72, 73, 74, 75, 76, 77, 78, and 79 as marked by brackets **[[ ]]**.

Notes:

(1) As required in NRC Information Notice (IN) 2009-07, documents containing proprietary information are marked with the word "Proprietary" at the top of the first page of the document and at the top of each page containing such information. In proprietary documents, brackets ("**[[ ]]**") denote proprietary information. In the proprietary document, the two brackets denoting the end of a proprietary segment of a report may appear one or more pages following the bracket indicating the start of the proprietary segment. In a nonproprietary edition of a proprietary document, the material within the brackets is removed.

(2) As noted in IN 2009-07, in instances in which a nonproprietary version would be of no value to the public because of the extent of the proprietary information, the agency does not expect a nonproprietary version to be submitted.

RESPONSE TO REQUEST FOR ADDITIONAL INFORMATION (SET 3)
ROLLS-ROYCE ***SPINLINE 3*** DIGITAL SAFETY
INSTRUMENTATION AND CONTROL PLATFORM
PROJECT NO. 773

To date, the U.S. Nuclear Regulatory Commission (NRC) Staff has made reasonable progress on the safety evaluation for the licensing topical report (LTR) on the Rolls-Royce Civil Nuclear (RRCN) ***SPINLINE 3*** Digital Safety Instrumentation and Control Platform. However, there are a handful of areas where the NRC staff needs to ensure that it clearly understands the terminology and some very specific ***SPINLINE 3*** Operating System Software (OSS) operations.

In addition, the NRC staff has a reasonable understanding of the NERVIA network communications from discussions during the regulatory audit performed in Grenoble, France. For the NRC staff to rely on some of the NERVIA features in its assessment of how ***SPINLINE 3*** meets Interim Staff Guidance (ISG)-04 (ADAMS No. ML083310185), the NRC staff will need to have some of the LTR-provided NERVIA communications material augmented.

Please answer the following requests for additional information on the ***SPINLINE 3*** Platform.

I. Operating System Software

RAI-68 Operating System Software

The LTR, Section 4.4.2 and 6.1, briefly describe that the Core System Software (CSS) and the Basic Functions (BFs) are part of the OSS. However, it is not clear how the CSS and BF work. In particular the LTR does not clearly describe how the CSS manages the environment in which the OSS elements operate. Also, the LTR does not describe how the BF transfers or exchanges data between the OSS and the CSS, and the input/output (I/O) modules, and then how this data is transferred and used by the application software.

RRCN submitted Document 1 207 108, "Operational System Software – Software Requirement Specification". This document seems to make a clear distinction between the functions performed by the OSS, BF, and CSS that is not explained in the LTR. Further, it is not clear why functions or attributes explained in the LTR are not consistent with the description provided in this document. For example, Section 3 of RRCN Document 1 207 108 explains error management for the ***SPINLINE3*** system, which states that errors are generated and reported by the CSS. However, the LTR, Section 4.4.3.6, states that the errors are generated and reported by the OSS, without referencing the CSS.

Please clarify the relationship between the OSS, BF, and the CSS. Also please explain the use of configuration tables by the OSS. Note if drawings are provided; please include a description of the drawing and how the information depicted relates to operation of the SPINLINE 3 platform.

Rolls-Royce Answer – The response below is organized to address software organization, self-test and error management, and OSS configuration tables.

Software Organization

The Operating System Software (OSS) is the assembled software including the CSS (Core System Software), the collection of BFs (Basic Functions), and the LDU_CPU (Local Display Unit Driver) module. In Rolls-Royce document 1 207 108 (OSS – Software Requirement Specification), Figure 4 shows a simple layout of the OSS. This same layout is shown in ***SPINLINE 3*** Licensing Topical Report (LTR) Figure 4.4-1.

The CSS should be considered as the "main" program which ensures the cyclic, sequenced, and deterministic behavior of the configured Unit. The CSS runs the sequence and calls the modules as "sub-programs" that, in turn, call their own sub-modules. All the calls are performed in a cyclic and sequenced manner. The BFs are the drivers that are used to interface between the CSS and the Input/Output (I/O) boards. The LDU_CPU is the interface module between the CSS and the Local Display Unit (LDU).

In Rolls-Royce document 1 207 228 (Safety of Processing Unit Software) Section 2.3.1 provides an overview of the OSS architecture and its operating sequence. The OSS is organized into high-level modules (called Treatment Modules). The Treatment Modules consist of the following activities: initialization, self-testing, acquisition, application processing, transmission, etc. A Treatment Module calls lower level Basic Modules. For example, the Transmission (sometimes referred to as Emission) Treatment Module calls the Basic Modules for the various types of Output Modules (e.g., analog, digital, relay, etc.). The Basic Module will make a call to the specific Basic Function module associated with the particular I/O board.

This software architecture is characteristic of software developed in accordance with IEC 60880. The goal is to minimize the changes to the OSS to accommodate the potential changes in I/O hardware initiated by possible technological evolution, operating experience or obsolescence management. Ideally, the hardware adaptations can be accommodated with changes to just the Basic Function module for the I/O board.

Figure 1 (from Rolls-Royce document 1 207 228) illustrates the chained calls from the Cycle Time Management module (Core) to the Treatment Modules, then to the Basic Modules, and down to the Basic Functions. The Core Module, Treatment Modules, and Basic Modules are part of the CSS.

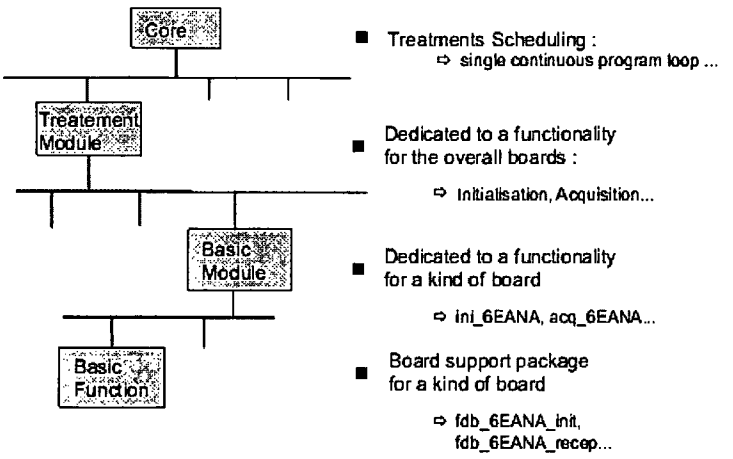

Figure 1 – OSS Architecture Illustration

Figure 2 (from Rolls-Royce document 1 207 228) illustrates the sequential operation of the OSS.

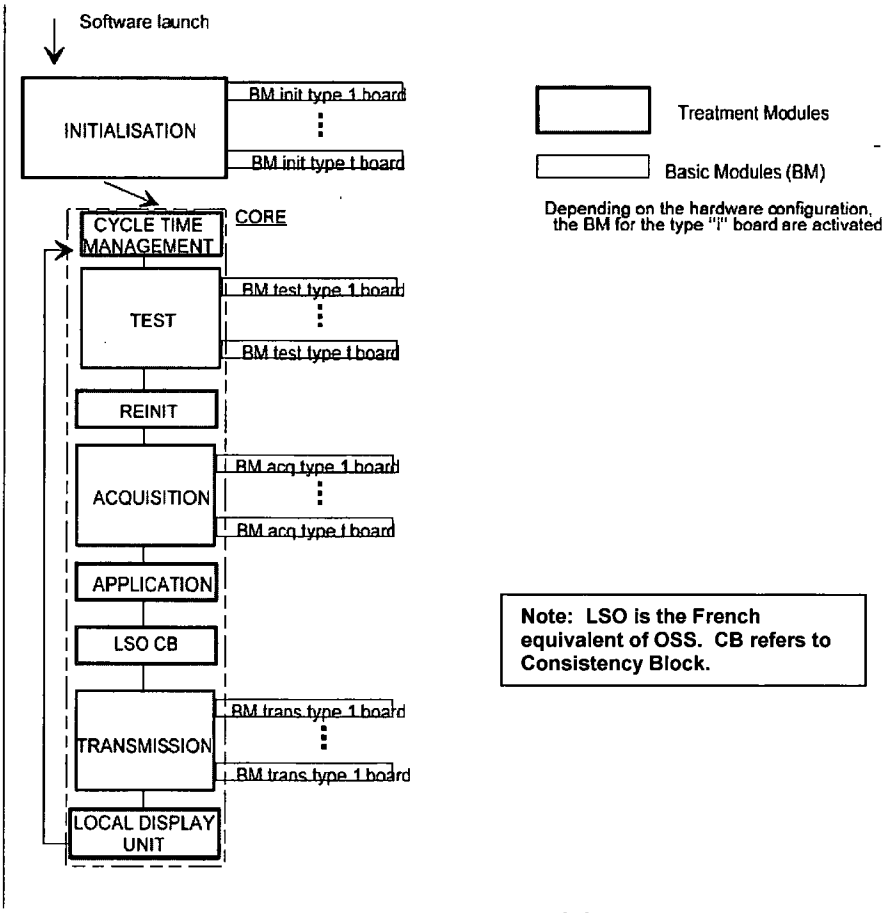

Figure 2 – OSS Sequence of Operation

Self-Test and Error Management

The I/O self-tests are ensured by calling the Basic Modules, which then call their own Basic Functions specifically for each type of board or peripheral.

The LTR simplified the description and aimed to explain that the hardware failures are covered by the OSS and not the Application Software, which only uses the test result information to manage the Unit outputs and signaling. Typically, the Unit sets the outputs in a safe or predefined state and sets the invalidity flags for data transmitted on the network.

OSS Configuration Tables

The OSS uses parameter tables produced by the CLARISSE System and Software Development Environment to communicate with the I/O modules, the Application Software, and the Local Display Unit. The OSS also uses this information to produce the System Consistency Block (CB). The use of the tables by the OSS is illustrated in Rolls-Royce document 1 207 108 Figure 6.

The Hardware Configuration Table defines the Unit's hardware configuration and this data is used by modules handling the different OSS functions. The Hardware Configuration Table is described in Rolls-Royce document 1 207 108 Section 2.5.1. All details on the Hardware Configuration Table specifications are found in Rolls-Royce document 1 207 110 (***SPINLINE 3*** Interface Specifications) Section 2.4.

The System Status Table is used by the CSS and LDU to share information on rack hardware status and connected network status, LDU connection status data, and information on changes to individual parameters in the electrically erasable programmable read-only memory (EEPROM) made by the LDU. The System Status Table is described in Rolls-Royce document 1 207 108 Section 2.5.2. All details on the System Status Table specifications are found in Rolls-Royce document 1 207 110 Section 2.3.

The OSS/Application Interface Table is used for the data exchanges between the OSS and the Application Software. The OSS/Application Interface Table is described in Rolls-Royce document 1 207 108 Section 2.5.3. All details on the Hardware Configuration Table specifications are found in Rolls-Royce document 1 207 110 Section 2.2.

The Static Hardware Configuration Table is internal to the CSS and is shared with the CD_LDU program. It is used to provide information on the UC25 N+ board memory map. The Static Hardware Configuration Table is described in Rolls-Royce document 1 207 108 Section 2.5.5.

RAI-69 Platform Peripheral Boards

LTR, Section 4.4.3.5.1 describes the initialization functions that the OSS performs. This section identifies the following peripheral boards: microprocessor-based peripherals, non microprocessor-based peripherals, and configurable non-microprocessor-based peripherals. LTR Section 5.2 describes the following peripheral boards: intelligent peripherals, non-intelligent peripherals, configurable non-intelligent peripherals, and non-

configurable non-intelligent peripherals. Please clarify the different peripheral boards terminology used in the LTR and OSS software requirement specification (SRS). Also, please include a complete list of peripheral boards and their classification.

Rolls-Royce Answer – The following table provides a mapping between the LTR and SRS terminology:

LTR	SRS	Related Board
microprocessor-based peripherals	intelligent peripherals	NERVIA ICTO
non microprocessor-based peripherals	non-intelligent peripherals	6SANA 32ACT 32ETOR
configurable non-microprocessor-based peripherals	configurable non-intelligent peripherals	16E.ANA ISO

It should be noted that this LTR description is for software initialization. The SRS notes that only intelligent peripherals require software initialization.

The I/O boards listed in the above table are a complete list of peripheral boards associated with each classification.

These boards correspond to boards communicating through and addressed by the CPU through the Parallel Acquisition Bus and thus, managed by the OSS. Other boards (e.g., conditioning boards or power supplies) are not considered in this table.

RAI-70 Initialization Functions

LTR, Section 4.4.3.5.1 describes the initialization functions that the OSS performs. This section does not explicitly state that the initialization functions should be performed in a particular order. LTR Section 5.2 states that these functions should be performed in a particular order, which does not match the order followed in the LTR. Please clarify if these functions are performed in a specific and determined order, and, if so, which is the nominal order.

Rolls-Royce Answer – The LTR correctly describes the implementation of the sequencing. Indeed, the parameter copy is implemented as part of the initialization of the UC25 N+ board.

In Rolls-Royce document 1 207 108, the more detailed condition for activating and/or halting the functions are given in the Sections 5.2.1 to 5.2.7. The sequencing of the first four functions is indeed respected and Section 5.2.6 gives the condition for activating the copy of parameter "as soon as the UC25 N+ board's internal peripherals are initialized if there are EEPROM".

The introductory language describing this in the SRS is confusing. A better wording would be:

[[

]]

RAI-71 Use of Interrupts

LTR, Section 4.4.3.5.1, Sub-function 1, states that when the central processing unit (CPU) is initialized, it sets up interrupts. Other sections in the LTR (e.g., Section 6.2.3) state that interrupts are not used in the OSS for managing functions. Please explain what type of traps and interrupts are used in the ***SPINLINE 3*** platform, and how and when they are activated. Further, if these interrupts are generic to the platform review, then describe cases, conditions, and actions taken when these interrupts are activated.

Rolls-Royce Answer – The interrupts physically exist on the UC25 N+ CPU (pins) as well as the potential software exception. Both interrupts, even if unexpected, have to be managed and the CPU has to be configured to handle the occurrence of signals on the related pins (which may come from a hardware failure) or unexpected software exceptions.

The processor is, by design, always in one of three states: normal processing, exception processing, or halted. It is in the normal processing state when executing instructions, fetching instructions, and operands, and storing instruction occur. Exception processing is the transition from program processing to system, interrupt, and exception handling. Exception processing includes fetching the exception vector, stacking operations, and refilling the instruction sequencer after an exception.

The processor enters exception processing when an exceptional internal condition arises (e.g., tracing an instruction, an instruction results in a trap, or executing specific instructions). External conditions, such as interrupts and access errors, also cause

exceptions. Exception processing ends when the first instruction of the exception handler begins to execute.

The processor halts when it receives an access error or generates an address error while in the exception processing state. For example, if during exception processing of one access error another access error occurs, the MC68040 is unable to complete the transition to normal processing and cannot save the internal state of the machine. The processor assumes that the system is not operational and halts. Only an external reset can restart a halted processor. Note that when the processor executes a STOP instruction, it is in a special type of normal processing state, one without bus cycles. The processor stops, but it does not halt.

[[

]]

An interrupt consists of retrieving the CPU code execution associated with an error, displaying the error code on the front panel of the board, and pointing to the "stop CPU" Basic Function. Consequently, the Unit and its outputs fall in their safe (or predefined) positions.

SPINLINE 3 software units run by the UC25 N+ CPU do not make use of interrupt-driven tasks. In fact, no application software processing or OSS processing is done under hardware or software interrupts and no multitasking or multithreading is implemented. No interrupts are used for management of OSS functions.

RAI-72 Self-Diagnostic Tests of the CPU Board

LTR, Section 4.4.3.5.2, describes the test and self-test diagnostics for the CPU and LTR Section 5.3.1 describes UC25 N+ board tests during operation. The OSS document states that these tests are performed over several cycles, and the operator defines how to spread the tests. The execution of these tests is not described in the LTR, although they are described in the OSS document. Please clarify how these self-diagnostic tests are performed, and what parameters are defined to spread out and perform these tests (e.g., all self-tests should be performed.

Rolls-Royce Answer – All of the self-tests of a Unit, related to board access and network communication, are performed within each software cycle. Self-test of memories based on checksum calculation need longer periods of time (i.e., cumulative time of a few seconds for all memories) and are consequently conducted over several cycles. As part of code generation, the designer can set:
[[

]]

The greater the number for each parameter is, the longer the time to execute the test is at each cycle. The value is selected to ensure cycle time requirements are satisfied. The smaller the number is, the larger the number of UC25 N+ cycles is, to complete the full scope of the memory test.

Characteristics of the Code RAM checksum:

- minimum element size to be summed = one 32 bit word (4 Bytes)
- checksum storage size = one 64 bit word (8 Bytes)

Characteristic of the EEPROM and Application RAM checksum (i.e., copy from the EEPROM to the RAM):

- minimum element size to be summed = one Byte (8 bits)
- Checksum storage size = one 32 bit word (4 Bytes)

The total numbers of Bytes to be summed are:

Element	Size (including checksum)	Area to be summed (excluding the checksum)
Code RAM	[[ ]]	[[ ]]
Application RAM	[[ ]]	[[ ]]
EEPROM	[[ ]]	[[ ]]
	Total Bytes to be summed	[[ ]]

Constraints: The overall time for the test of all memories shall not exceed 1 hour. Accordingly, the number of elements to be summed within each cycle is directly given by the number of cycles/hours.

For example, if a Unit cycle time is regulated at 30 millisecond, the number of cycles per hour equals 120,000 and the number of bytes to be summed per cycle shall be set at least at **[[ ]]** to ensure that the overall self-test is performed in less than 1 hour. This number of Bytes to be tested per cycle is set and defined through the CLARISSE tool during the design phase of the software as part of the Unit configuration.

RAI-73 Electrically Erasable Programmable Read-Only Memory (EPROM) Parameters

LTR Section 4.4.3.5.1, Sub-function 2, describes the initialization task for the EEPROM. LTR Section 5.2.6 also describes this initialization activity. However, the OSS document states that during initialization of the EEPROM this component is not accessible for a defined period. Please explain this initialization function and where the defined period is configured, and whether the EEPROM becomes available after the defined period has elapsed.

Rolls-Royce Answer – In document 1 207 108, Section 5.2.6 gives a more detailed reason to use the mechanism of copying the parameters from the EEPROM to the RAM. The EEPROM contains the initial application software parameters and it is also used for parameter changes made during the operational phase once the system is in service. The Application Software uses for its own processing only the parameters in the RAM.

During a parameter change (made through the LDU) the application remains functional; however, the EEPROM is not accessible and will not be accessible until the change is finished (.i.e., the hardware writing of the parameter change). The EEPROM becomes available after the write activity has finished.

[[

]]. At that time, parameters in the RAM and the EEPROM become identical.

II. Communication

RAI-74 Consistency Blocks and Dual-Port Memory

LTR, Section 4.5, describes the ***SPINLINE 3*** platform communications. In particular, Section 4.5.2.6 describes how data from the Station's MPC 860 microprocessor is transferred to the Unit's MC68040 microprocessor via writing to the Dual-Port Memory (DPM), so it can be used by the NERVIA network. This section explains that data is copied using consistency blocks. Section 4.5.4.3 goes into further detail by stating: "DPM memories are organized [[ ]]." However, this section does not explain how the CBs are [[ ]] in the DPMs, and how [[ ]] when data is transmitted in the NERVIA network. Please provide further explanation regarding how CBs are used in conjunction with the DPM – specifically, elaborate on the use of the [[ ]].

Rolls-Royce Answer – The response below discusses the sequencing of DPM use and the organization of the DPM.

Sequencing of DPM Use by the Unit and Related Station

As a reminder, the figure bellows shows the physical and functional overview of the data exchange using Figure 3 (from in LTR Section 4.5).
[[

]]

Figure 3 - Basic Processing and Communication Architecture for Data Exchange between a Station and a Unit

The following sequence of events describes the general ***SPINLINE 3*** network communication.

As a preliminary, both processors (i.e., MPC 860 managing the Station and 68040 managing the UC25 N+) have independent cyclical processing and operate independently of refreshment of data in the DPM. Exchanges between a Unit and an associated Station are not synchronized. Both processor access the DPM according to their own cyclic processing with both sides operating independently.

As an example, consider one Unit (indentified as UNIT1), connected to a given Network with one single related Station (identified as STATION U1_S1 among the 6 possible stations up to U1_S6) and that this Station communicates on the network with K other Units (UNITK) through their related Station (e.g., STATION UK_S1).

Considering UNIT1, independently of the activity of STATION U1_S1:
[[

]]

As such, UNIT1 can read and write to the DPM independently based on its own asynchronous cyclic process. The transmission and acquisition steps described above are the same for any UNITK

Considering STATION U1_S1, independently of the activity of UNIT1:
[[

]]

As such, STATION U1 can read and write to the DPM independently based on its own asynchronous cyclic process. It is not influenced by either UNIT1 or STATION UK operation. The transmission and acquisition steps described above are the same for any STATION UK.

Considering STATION UK_S1, independently of the activity of UNITK and STATION U1_S1:
[[

]]

As such, STATION UK_S1 can read and write to its own DPM independently based on its own asynchronous cyclic process. It is not influenced by either UNIT1 or STATION U1 operation.

More information on the temporal design of a NERVIA+ Network is found in the response to RAI 77.

Organization of the DPM

A Unit DPM is organized and contains a predefined allocation of addresses for all data. The organization contents are:
[[

]]

The algorithm describing the buffer selection by the producer and consumer is given in Figure 4.

[[

]]

Figure 4: Shared Memory Software Access Protocol

This protocol ensures that the UC25 N+ operates without interference from the NERVIA+ communication processor.

RAI-75 DPM data control

LTR Sections 4.5.2.6 and 4.5.4.3 state that data control flags would be used to access each side of the DPM, and [[

]]. Section 4.5.3 states that the use of this flag is explained in Section 4.5.4.3, but this section only states that access to the buffers is controlled using a pair of data control flags. Thus, it is not clear how the

data control flags work to indicate that a buffer is not available. Also, the LTR explains that in the case that one of the microprocessor is reading, the data will be written to the buffer not being accessed. It is not clear when the other buffer is updated, and how the data control flags will indicate this.

Then, Section 4.5.4.3 states that potential conflicts with both processors attempting to access the data control flags at the same time are **[[ ]]** by the Complex Programmable Logic Device (CPLD), and that this is explained in Section 4.5.3. However, Section 4.5.3 only repeats what was said in Section 4.5.4.3. Therefore, it is not clear how the data control flags and the CPLD interact and work to control access to the buffers.

Please explain how DPM and data control flags work and how they control access to the DPM's buffers.

Rolls-Royce Answer – In Figure 4, the UC25 N+ processor board is on the right side of the DPM and the NERVIA+ daughter board and I.NERVIA+ interface board are on the left side of the DPM.

At the start of the access to the DPM (i.e., read or write) cycle, each microprocessor working asynchronously will attempt to read and set the **[[**

]]

Figure 5: Control Flag Access Management Signal

RAI-76 Status of the DPM buffer

The LTR explains that when a microprocessor wants to read data from the DPM, it will read it from the buffer with the most recently updated data. It is not clear what flag or data is used to identify recently updated buffer. Please explain how the status of the buffer is kept.

Rolls-Royce Answer – The control flags **[[**

]]

This control ensures that a producer is not continuously writing in the same buffer and letting the other buffer containing older information.

The use of the two flags is described in the response to RAI 74. Access to the control flag itself is described in the response to RAI 75.

RAI-77 NERVIA communication

The LTR explains that the NERVIA network is a ***SPINLINE 3*** dedicated network, which was developed based on the Open System Interconnection model of the International Standard Organization. Further, the LTR provides a general description of the frame and protocol used to transmit messages in the network. However, this information is not sufficient to understand:

a) Additional information on the NERVIA logic code and how the protocol works is necessary to evaluate points 16 and 18 of ISG-04. Specifically, the staff needs information that describes the logic used to flag invalid data, network initialization and operation, network status, etc.

   For example, Section 4.5.5.2 of the LTR states that network and Station description tables (e.g., Table de Description de Station) are defined for the network communication. These tables would specify the [[

   ]]. This section states that if there is only one Station in the network, this Station will transmit data.

   However, if several Stations are in the network, the TDS will define the [[

   ]]. For example, if there are 4 Stations, the [[ ]] is defined for these Stations to be [[

   ]].

   What happens if Station [[ ]]?
   Can Station [[ ]] defined in the TDS? Will Station [[ ]] initialize?

   Please provide different scenarios of how events that could occur during initialization and how the Stations would behave.

Rolls-Royce Answer – The NERVIA+ Network is designed to allow each Station to operate independently; however, each Station in a NERVIA+ Network is configured to operate in a coordinated manner within the sequenced emission of all Stations in overall cycle. The deterministic behavior of the NERVIA+ Network is based on dedicated communication protocol that has a time-based element and a token bus element.

The time-based element defines for each Station when it can transmit and how long it is will take to transmit. This time scheme is calculated and configured during the development of the software for a project based on the specific network configuration. A sequence of cyclic and ordered transmitting Stations declared on the Network is established. The choice of the order is not important; instead, it is only important to establish a given order for the deterministic behavior of the sequence. This order of the

sequence is not linked to the order of powering up the Units and related Station because the NERVIA+ protocol does not rely on a "Master" Station. In other words, no Station has a specific role, either during network initialization or during operation. The amount of data transmitted by each Station, defined during the software development, is the input for the calculation of the transmission time windows allocated the respective Stations. The number of Stations and the time window for each Station, based on the amount of data transmitted by each Station, determine the maximum Network cycle time.

The token bus element aims to define the order of the fixed cyclic sequence of transmissions. It consists of a simple number for each Station (called the token). This token is transmitted from one Station to the others through the "next station sequence number" field included in all of its network messages. The next Station to transmit knows its turn through the positive comparison of its own Station number and the "next station sequence number" set in the received CB. [[

]]

A NERVIA+ Network is initialized as soon as a first Station is initialized. It can be any Station, since there is no Master Station in the protocol. The first emitting Station to power up and initialized listens to the network traffic for a period of time greater than the defined Network cycle time. If no message is received during this period of time, this Station knows it is the first on the network and starts transmission. This first Station starts a timer used to identify its next transmission time window in the network cycle. The first Station will transmit again at the end of the timer, which corresponds to the network cycle time if no messages are received. This operation will proceed indefinitely if no other Station is inserted. In this manner, the first Station transmission operates independently of any other Stations. The first Station message serves as a synchronization signal for the next Station to join the Network. Station transmission and reception are mutually exclusive operations.

The second emitting Station to power up and initialize will begin listening. It will receive the message from the first Station. The received message contains the token (i.e., next Station sequence number). This second Station compares the sequence number contained in the transmitted message with its own sequence number.
[[

]]

All other new Stations which insert (when they are powered up and initialized) will follow the same procedure and steps as the second Station: listening, checking the token, setting its own timer, and transmitting in its pre-defined time window.

Special case

There is a particular case at start-up, where it may happen that multiple Stations try to start their insertion at the same time. [[

]]

As a summary, each message transmitted on the NERVIA+ Network media is received by all other Stations. The messages sent on the Network are used as the synchronization signal for all receiving Stations, as described above. This protocol ensures that the NERVIA+ Network meets its bounded response time requirement, even in case of failure of one or several Stations. Each Station operates independently and can transmit to the Network even if they do not receive the token from any Station. If a Station receives a message, it knows where it is located within the time scheme and can insert itself and start normal operation.

b) ISG-04, Staff Position 1, Point 12 contains 12 examples of credible communication faults to be considered as part of the evaluation. Table 3.7-1 of Appendix A in ISG-04 addresses each of the 12 examples. It is understood by the staff that the ***SPINLINE 3*** NERVIA network would not normally exhibit any of these errors. However, the staff is attempting to conclude that the system is robust to handle such errors or that the example error is not credible for the system.

   For examples 2 and 3, do the NERVIA messages contain any time-stamp or message sequence identifiers such that a repeated or out-of-sequence message would be identified by a recipient station, i.e., is the NERVIA network designed to handle and/or detect such errors?

   Alternately, is it RRCN's position that such errors are not credible for the NERVIA network? If so, please provide additional explanation as to why these errors are not credible.

Rolls-Royce Answer – NERVIA+ Network messages are created by each transmitting Station on the Network within each Network cycle. The deterministic nature of the network protocol ensures that each Network Station transmits its message at specified time in the Network cycle.

The NERVIA+ communication frames do not contain time-stamps. The synchronization necessary for keeping the sequenced emitting in the given time slots is ensured by the local time counter started when a message is received (refer to the response to RAI-77 Part a).

The ***SPINLINE 3*** technology is also designed to detect and safely handle communication errors. The self-diagnostic tests are performed at the Station level and at the Receiving Unit level (see LTR Section 4.5.6 on communication failures detection).

The self-test at the NERVIA+ board level consist checking correct functioning of the MPC 860, correct addressing of memories, integrity of the memories, clock drift, Ethernet frame CRC, watchdog status. The result of the local errors are set in the control zone of the DPM and accessed by the UC25N+.

At the Receiving Unit level, the tests consist of following checks and verification:
[[

]]

Any one of these errors is managed by the receiving Units to treat the information read in the CB as invalid.

A Station failure to transmit is detected by the refreshment indicator check by the receiving Unit. Stale data are flagged as invalid and handled in accordance with the engineered fault management features built into the Application Software of the receiving UC25 N+ processor.

A Station failure to receive or update the shared memory is detected by the refreshment indicator check by the receiving Station. Stale data are flagged as invalid and handled in accordance with the engineered fault management features built into the Application Software of the receiving UC25 N+ processor.

The refreshment check will detect messages that may be repeated at an incorrect point in time. The message identifier test will detect messages that may be sent in the incorrect sequence.

c) Additional information on the configuration of the network and definition of Stations categories is needed to evaluate ISG-04 and independence between 1E and non-1E communication.

   For example, how does the network know when to define a receive-only station? Where is this marked in the TDS?

Rolls-Royce Answer – The NERVIA+ Network configuration is definitively defined during the development process using the CLARISSE Tool. The characteristic of a Station as emitting/receiving or receiving only is defined in the TDS, which are loaded in the DPM at the start-up and initialization of the Unit. The table is used by the NERVIA+ software to definitively configure the Station. The transmitting Stations operate independently, as described in the response to RAI-77 Part a.

[[

]]. They have no messages defined for them to transmit, as part of the project-specific software development. As such, there are no data to transmit assigned to those Stations and no time window allocated for them in the Network Cycle Time. As such, they cannot influence the transmitting Stations with message data or the Network cycle time with message transmissions.

Non-1E Receive-only Stations would be isolated from the 1E potions of the system by qualified isolators (e.g., fiber optic cables, etc.).

d) Additional information is necessary to evaluate determinism and independence of the NERVIA and the CPU. Specifically, when both the NERVIA and CPU are writing data to the DPM, and then how data is updated in the buffer not being updated.

Rolls-Royce Answer – The shared memory access protocol shown in Figure 4 controls the read and write access to the DPM. The NERVIA+ daughter board and UC25 N+

processors can act as data Producers (writers) and Consumers (readers) from the DPM. Access to the two buffers is controlled in the following way.

The Consumer is informed of which buffer to read by the Producer via the [[ ]]. The Consumer informs the Producer which buffer is being read via the [[ ]]. The Producer is informed that the Consumer is done reading the buffer via [[ ]].

The Producer is informed by the Consumer if a buffer is being read via the [[ ]]. The Producer writes to buffer B if buffer A is being read. The Producer informs the Consumer that it wrote to buffer B via the [[ ]]. Similarly, the Producer writes to buffer A if buffer B is being read. The Producer informs the Consumer that it wrote to buffer A via the [[ ]]. When the DPM is not being read by the Consumer, the Producer writes to the opposite buffer than it last wrote in. With [[ ]], the Producer writes to buffer B. With [[ ]], the Producer writes to buffer A.

e) Additional information is needed to demonstrate that network initialization is independent of Station initialization and Unit operation and to demonstrate that these functions are independent from the network operation. In other words, when the ***SPINLINE 3*** platform starts, the operating system performs a series of initialization functions, including the NERVIA daughter board. Then the system starts operating and performing its functions.

   It is not clear how the NERVIA network is initialized and configured. LTR Section 4.5.2.5 states that the network is initialized when the first Station is powered.

   Does the network have to wait for the OSS to finish configuration of this first station? How does this work when more than one station is connected to the network? How is the order established for them to start transmitting?

Rolls-Royce Answer – The Network is initialized when the first Station on the network begins transmitting. A Station cannot transmit to the Network until the OSS has finished configuring the Station.

The configuration of the Station consists in initialization of the MPC 860, self-test of local hardware (e.g., [[ ]]). Once the MPC 860 is initialized, the associated Stations are initialized and data used by the software are set in their initial state. This includes copying of the NERVIA+ software into RAM, loading the Network configuration tables in the DPM so that the Station and the related NERVIA+ software are set according to the Network specification (i.e., [[ ]]). As such, the Network is not initialized until the OSS finishes configuring at least one Station. The first Station (not necessary the Station with sequence number 1) and subsequent Stations insert into the network as described in the response to RAI-77 Part a.

The NERVIA+ network is designed to allow each Station to operate independently; the automatic insertion procedure of Station is explained in RAI-77 Part a. Once introduced in the in the NERVIA+ Network, each Station operates in a coordinated manner as defined by the cyclic and sequenced time windows.

RAI-78 Reinsertion of a Station in the NERVIA Network

a) LTR Section 4.5.5.2 states that when a Station wants to transmit, [[

]]. Please explain how this is indicated in the network message or frame. Is this simply reflected as "stale" data for the station that could not transmit, or is there a different method to acknowledge this occurrence?

Rolls-Royce Answer – The reinsertion is managed by the Station NERVIA+ software which starts by re-synchronizing with the Network: the Station sets to the reception mode, for a predefined time. Within this predefined time, if the Station cannot transmit because another Station is still transmitting, then an insertion default is detected by the Station and the Station will try again to reinsert.

In case of a repeated unsuccessful transmission attempt, the Station [[
]]. If the reinsertion does not succeed, the Station will [[
]], which is used by the UC25 N+ to adapt its own processing (according to its functional specification).

Since the Station is not able to transmit the information on the Network, the default is detected by all other Stations through the missing reception of message from that station and is also detected by the UC25 N+ of those Units through the "frozen" refresh indicator of the CBs. Stale data are flagged as invalid and handled in accordance with the engineered fault management features built into the Application Software of the receiving Unit.

Conditions of lock-out of a station are:

- During Unit initialization, as long as the Unit has not requested for the Station initialization, the Station is in a lock-out status (see LTR Section 4.5.5.2).
- During self-diagnostic tests: if the Station detects an unrecoverable error (see LTR Section 4.5.5.2), it informs its Unit, and then sets to a lock-out position.

An unrecoverable error for the Station is one of the following:
[[

]]

A Station comes out of its lock-out position by being reinitialized by its Unit.

Situations which lead to the need for Station reinsertion into the Network following its initialization or re-initialization:

- During Unit initialization, once the Station in initialized, its needs to be inserted in the Network
- Following detection of an unrecoverable error, the Unit requests for the Station to re-initialize. Re-initialization is performed only if the previous re-initialization (or

initialization) attempt has not already failed, in which case, re-initialization is not performed

b) Also, this section states that a [[ ]] waits to transmit. Please clarify if this [[ ]]. And if this is a NERVIA-specific [[ ]], please explain how this [[ ]] works.

Rolls-Royce Answer – In that situation described in the response to part a of RAI 78, the word "watchdog timer" is, indeed, inappropriate. In any case, the communication management is processed by the MPC 860 on the NERVIA+ daughter board and not by the MC68040 processor on the UC25 N+ side. The waiting delay (as intended to be explained in the sentence of the LTR) is the [[ ]] as determined in a normal cycle. The sentence should be understood as the station is not trying to retransmit until enough time is passed until its normal "time window".

RAI-79 DPM configuration

LTR Section 4.5.4.3 states that the [[ ]] are handled by the off-line configuration generation tool. Please clarify if this is intended to refer to the CLARISSE Workshop tool. If not, please provide a brief description of this tool.

Rolls-Royce Answer – The CLARISSE Tool is used to configure the application-specific software.