ORNL/SR-2016/148

A Qualitative Assessment of Current CCF Guidance Based on a Review of Safety System Digital Implementation Changes with Evolving Technology

Kofi Korsah
Michael Muhlheim
Richard Wood

Approved for public release.
Distribution is unlimited.

April 2016

DOCUMENT AVAILABILITY

Reports produced after January 1, 1996, are generally available free via US Department of Energy (DOE) SciTech Connect.

Website http://www.osti.gov/scitech/

Reports produced before January 1, 1996, may be purchased by members of the public from the following source:

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
Telephone 703-605-6000 (1-800-553-6847)
TDD 703-487-4639
Fax 703-605-6900
E-mail info@ntis.gov
Website http://www.ntis.gov/help/ordermethods.aspx

Reports are available to DOE employees, DOE contractors, Energy Technology Data Exchange representatives, and International Nuclear Information System representatives from the following source:

Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831
Telephone 865-576-8401
Fax 865-576-5728
E-mail reports@osti.gov
Website http://www.osti.gov/contact.html

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/SR-2016/148

Electrical and Electronics Systems Research Division
Reactor and Nuclear Systems Division

A QUALITATIVE ASSESSMENT OF CURRENT CCF GUIDANCE BASED ON A REVIEW OF SAFETY SYSTEM DIGITAL IMPLEMENTATION CHANGES WITH EVOLVING TECHNOLOGY

Kofi Korsah
Michael Muhlheim
Richard Wood

Date Published: April 2016

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831-6283
Managed by
UT-BATTELLE, LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

LIST OF FIGURES

ACRONYMS

ADAMS	Agency-wide Documents Access and Management System
ALS	advanced logic system
ALWR	advanced light water reactor
BTP	branch technical position
CCF	common-cause failure
CFR	code of federal regulations
CLB	core logic board
COM	communication board
CRC	cyclic redundancy check
DFP	digital filter processor
DI&C	digital instrumentation and control
EEPROM	electrically erasable programmable read only memory
EPRI	Electric Power Research Institute
FB	function block
FD	function diagram
FDG	function diagram group
FEPROM	flash erasable programmable read only memory
FPGA	field programmable gate array
FSM	finite state machine
HDL	hardware description language
IAEA	International Atomic Energy Agency
I&C	instrumentation and controls
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
I/O	input/output
IPB	input board
ISG	interim staff guidance
LCP	loop calculation processor
MMI	man-machine-interface
MSI	monitoring and service interface
NAND	negative AND
NPP	nuclear power plant
NRC	US Nuclear Regulatory Commission
NUREG	NRC regulatory guide
OPB	output board
ORNL	Oak Ridge National Laboratory
PROM	programmable read only memory
RAB	reliable ALS bus
RAM	random access memory
RTD	resistance temperature detector
RTE	run time environment
SAR	safety analysis report
SDD	system description document
SDRD	system design requirements document
SECY	Secretary of the Commission, Office of the NRC
SEU	single event upset
SPACE	specification and coding environment
SQA	software quality assurance

TAB	test ALS bus
TSP	test sequence processor
TXS	TELEPERM XS
UTK	University of Tennessee, Knoxville
V&V	verification and validation
VHDL	VHSIC Hardware Description Language)
VHSIC	very high speed integrated circuit

EXECUTIVE SUMMARY

The US Nuclear Regulatory Commission (NRC) is initiating a new rulemaking project to develop a digital system common-cause failure (CCF) rule. This rulemaking will review and modify or affirm the NRC's current digital system CCF policy as discussed in the Staff Requirements Memorandum to the Secretary of the Commission, Office of the NRC (SECY) 93-087, *Policy, Technical, and Licensing Issues Pertaining to Evolutionary and Advanced Light Water Reactor (ALWR) Designs,* and Branch Technical Position (BTP) 7-19, *Guidance on Evaluation of Defense-in-Depth and Diversity in Digital Computer-Based Instrumentation and Control Systems,* as well as Chapter 7, "Instrumentation and Controls," in NRC Regulatory Guide (NUREG)-0800, *Standard Review Plan for Review of Safety Analysis Reports for Nuclear Power Plants* (ML033580677).

The Oak Ridge National Laboratory (ORNL) is providing technical support to the NRC staff on the CCF rulemaking, and this report is one of several providing the technical basis to inform NRC staff members. For the task described in this report, ORNL examined instrumentation and controls (I&C) technology implementations in nuclear power plants in the light of current CCF guidance. The intent was to assess whether the current position on CCF is adequate given the evolutions in digital safety system implementations and, if gaps in the guidance were found, to provide recommendations as to how these gaps could be closed.

The methodology adopted was to review the vendors' technology and software implementation processes for digital safety systems as technology evolved. The following three representative safety systems were selected to provide illustrative examples:

- Eagle 21 was selected to represent vintage microprocessor-based technology used from the 1980s to mid-1990s,
- TELEPERM XS (TXS) was selected to represent second generation microprocessor-based technology used from the mid-1990s to the 2000s, and
- The advanced logic system (ALS) was selected to represent the latest trend of using technology based on field programmable gate arrays (FPGAs).

Technology implementations were reviewed in light of the basic premise of the NRC in BTP 7-19 (Revision 6), that software-based or software-logic-based digital system development errors are a credible source of CCF and therefore are susceptible to CCF because identical copies of the software-based logic and architecture are present in redundant divisions of safety-related systems. BTP 7-19 categorizes firmware and logic developed from software-based development systems all under software.

The study first examined the unique capabilities and characteristics of digital technology that distinguish it from traditional analog technology, and therefore make the above assertion likely. The most relevant characteristics were found to be the following:

- ***Digital systems typically have multiple functionality***—A digital system may be designed to perform multiple functions (e.g., acquire input data, process the data, perform onboard diagnostics, monitor alarmed conditions). With today's sub-micron integrated circuit feature sizes, this usually means that all the functions could reside in a very small space. Thus, failure of one integrated circuit could result in failure of multiple functions. In addition, important functionality is often integrated into servers and processors. The implication of this is that some performance parameters such as transmission speed and response times may deteriorate with a growing size of the I&C system due to higher processing loads. This characteristic has the

potential to negatively affect important plant or I&C functions such as the quality of closed loop control and reaction times of the human-system interactions.

- ***Information processing is fundamentally sequential in nature***—Analog and digital circuitry at NPPs are a means of acquiring signals from sensors and communicating the measured values to guide safety or control actions. Analog circuitry is traditionally hard-wired and dedicated to specific tasks. In contrast, digital systems signals are sampled and digitized, and the resulting information is transmitted and processed sequentially. This means that existing functional specifications such as response time and dead time must be reconsidered in detail before they are applied to the new DI&C design.

- ***The complexity of digital systems makes licensing more challenging***—When licensing DI&C, it is difficult to assure sufficient testing of the software. Even a small software module can exhibit enough complexity to make a full verification of its correctness within reasonable cost and schedule impractical. The assumption is that there is some probability that a latent error not discovered during the verification and validation (V&V) process may disrupt its function in a crucial situation. In this scenario, building (software) redundancy into the system cannot remedy the situation because the software is deterministic in its operation, and each redundant channel will have the same embedded error. Even the use of software diversity cannot guarantee adequate protection against such potential for CCF because the requirement specification may be the ultimate cause of a software error. In essence, a (complex) digital system is fundamentally non-linear, so it is difficult to model and/or predict its behavior.

  On the other hand, additional complexity enables additional functionality that can provide substantially greater confidence in correct operation in analyzed circumstances. This is accomplished through self-checking and health monitoring. Moreover, additional capabilities of digital systems can reduce the conceptual workload of the human operators (and thereby increase the probability of taking correct actions) by providing interpreted data in more easily understood formats. The goal of the nuclear power regulatory process is to provide reasonable assurance of adequate safety. The greater confidence in correct operation offered by self-diagnostics must be balanced against the potential for digital instrumentation to contain unrevealed errors that are technically difficult to correct.

The findings from the review of the three representative systems are summarized as follows:

1. Early microprocessor-based safety system implementations such as the Eagle-21 process protection system was designed as a modular *functional replacement for existing analog equipment*. Starting from the premise that analog systems were mature technologies and their review processes were stable, a strict adherence to digital functional replacement for existing analog equipment was seen as limiting the potential for digital CCF. This appears to be the baseline upon which subsequent guidelines such as BTP 7-19 and the DI&C ISG were developed. This is a reasonable baseline, and although it is not quantitative, the authors believe that the state of the art does not currently warrant using any quantitative approach.
2. Although early digital implementations were typically one-for-one replacements of the proven analog designs as exemplified by the Eagle 21, some advantages of digital technology (e.g., onboard diagnostics) were nevertheless also implemented. For example, the Eagle 21 implemented automatic surveillance testing (to reduce the time required to perform surveillance tests), self-calibration (to eliminate rack drifts and time consuming calibrations), and self-diagnostics (to reduce the time required for troubleshooting). The drawback of implementing these software enhancements was the need to assure deterministic software behavior in spite of

the additional software overhead. In the Eagle 21, deterministic performance was implemented as follows;

a. Use a modular approach in the software design, with all executable code contained in modules or subroutines.
b. No interrupts are allowed.
c. No re-entrance is allowed.
d. Code format conforms to standards for both high-level and assembly language routines
e. GO TO statements are not allowed.
f. All modules are single task (no operating system or multi-tasking system).
g. All modules are single entry, single task.
h. Modules exit to points of call
i. Each module has a design performance specification and a verification test specification. However, these implementations alone do not necessarily guarantee sufficient determinism. For example, with the added overhead of onboard diagnostics and surveillance software, each module, as well as each complete cycle, should also be guaranteed to complete in a pre-determined time.[a]

3. The software V&V and digital communication standards and guidelines available in this period (i.e., in the era of the Eagle 21) were generally adhered to in the system development. Since then, there have been considerable improvements in these standards and guidelines (e.g., DI&C-ISG-02) which now address issues such as interdivisional communication. However, because the early digital safety system implementations tended to be one-for-one replacements of analog systems with no inter-divisional communication, etc., the early standards and guidelines were adequate for the period.
4. Evolution of safety system implementations simply made use of more sophisticated microprocessors and increased online self-testing and surveillance, as exemplified by the TXS. However, these systems (TXS) also made use of the improving guidance for digital safety system implementations (e.g., updated V&V standards) and improved on implementation of deterministic performance. For example, the digital system architecture of the TXS included procedures that improved determinism such as (a) monitoring of cycle time by means of software and a hardware watchdog, (b) automatic testing of the watchdog, (c) bus systems with constant load, and (d) no processing of absolute time or date. Improvements in safety system software also included self-testing of the inputs from the input modules and automatic readback of the outputs from the output modules.
5. Because digital safety system implementations were also accompanied with improvements in regulatory guidance, the issue of CCF was also a greater focus in safety systems implemented beginning in the mid-1990s to the 2000s. For example, the preferred measure against CCF, especially in connection with design errors, was functional diversity. This involves ensuring that the safety I&C subsystems, while equipped with the same hardware and system software, execute different I&C functions for handling one and the same event. For example, a reactor trip resulting from a steam generator tube rupture event may be monitored by two I&C subsystems: one monitoring main steam activity, and one monitoring steam generator level and pressurizer level. The assumption here is that the same hidden fault will not take effect simultaneously in two

[a] It is possible that this was also implemented in the early digital software safety systems such as the Eagle 21. However, the authors were unable to ascertain this from the available documentation.

different functions at the same time, causing both of them to fail simultaneously. In the absence of a quantitative measure, BTP 7-19 and DI&C-ISG-04 provide good additional guidance for addressing digital CCF.

6. Software V&V procedures, reviews, and audits are important parts of the effort to reduce the potential for CCF and to comply with NRC requirements. The review of software V&V procedures for safety system implementations showed that there were general improvements in software V&V as the technology implementations also evolved. However, these improvements resulted from updates and improvements in regulatory guidance rather than from technology evolutions. The revisions to regulatory guidance resulted from updates and improvements in the standards endorsed by the regulatory guides. For example, the 2013 version of RG 1.168, "Verification, validation, reviews, and audits for digital computer software used in safety systems of NPPs," has undergone a significant update as a result of revisions of the endorsed standards in the 1997 version. (The latter version was used to guide V&V for the TXS reviewed for this report). Examples include the addition of a security analysis and the recommended use of the software integrity system, as the previous version did not require the selection of an integrity level.
7. With regard to the migration to FPGA technology, the reviews did not show that common cause failures are any less plausible for FPGA-based safety systems than for microprocessor-based safety systems. For both FPGA-based systems and microprocessor-based systems, it is difficult to prove adequate test coverage, and the method of ensuring adequate quality of the product continues to be extensive documentation of the development process, qualification, testing, guidelines on how to address computer communication issues (DI&C-ISG-04), guidelines on how to address diversity and defense-in-depth issues (BTP 7-19 Rev 6), etc. In the absence of quantitative methodologies (which the present state-of-the-art do not support), the current standards and guidelines provide very good guidance to assure quality and reduce the potential for CCF in DI&C for NPPs and should continue to be applied.
8. As a result of the above reviews, it is the authors' conclusion that current guidance aimed at reducing the potential for CCF as found in BTP 7-19 (Rev. 6) and DI&C-ISG-04 should continue to be relied upon. Operational experience could also be investigated in a future study to *support* current guidance. Operational experience alone cannot be used as proof of adequate design against CCF: the (safety) system may have been operating well for years during which the plant may even have undergone abnormal conditions showing that it performed its safety function under those abnormal conditions. However, that does not necessarily demonstrate adequate functionality under all scenarios that may not have occurred during the plant's operation.

1. INTRODUCTION

1.1 BACKGROUND AND PURPOSE OF REPORT

The US Nuclear Regulatory Commission (NRC) regulations require licensees to incorporate adequate protection against software common-cause failure (CCF) into a nuclear power plant (NPP), as well as an overall safety strategy to ensure that NPP abnormal operating occurrences and design basis events do not adversely impact public health and safety. Those protective measures can be provided through diverse functions and systems.

The NRC is initiating a new rulemaking project to develop a digital systems CCF rule. This rulemaking will review and modify or affirm the NRC's current digital system CCF policy as discussed in the Staff Requirements Memorandum to the Secretary of the Commission, Office of the NRC (SECY) 93-087, , *Policy, Technical, and Licensing Issues Pertaining to Evolutionary and Advanced Light Water Reactor (ALWR) Designs*, and Branch Technical Position (BTP) 7-19, *Guidance on Evaluation of Defense-in-Depth and Diversity in Digital Computer-Based Instrumentation and Control Systems*, as well as Chapter 7, "Instrumentation and Controls," in NRC Regulatory Guide (NUREG)-0800, *Standard Review Plan for Review of Safety Analysis Reports for Nuclear Power Plants* (ML033580677).The Oak Ridge National Laboratory (ORNL) is providing technical support to the NRC staff on the CCF rulemaking, and this report is one of several providing the technical basis to inform NRC staff members.

1.2 SCOPE OF REPORT

To support the CCF rulemaking study, this report examines the evolutions in technology and software implementation strategies from the 1990s to the present with regard to safety system development and in light of current regulatory guidance on CCF. While the high-level study of evolutionary changes does not provide detailed descriptions of specific technologies, some detail is provided as appropriate to highlight the changes discussed. Key issues examined during the study include the following questions regarding evolutions in technology and software implementations:

a. What is NRC's current position on CCF?
b. Is the current position adequate given the evolutions in digital safety system implementations from the 1980s to date? What are the gaps in the current NRC position where the move from the old single board computer technology to FPGA technology is not being addressed?)
c. If the current guidance is not adequate, what should be done to make it acceptable?

The methodology adopted was to review the vendors' technology and software implementation processes for digital safety systems as technology evolved. The following three representative safety systems were selected to provide illustrative examples:

- Eagle 21 was selected to represent vintage microprocessor-based technology used from the 1980s to mid-1990s,
- TELEPERM XS (TXS) was selected to represent second generation microprocessor-based technology used from the mid-1990s to the 2000s, and
- The advanced logic system (ALS) was selected to represent the latest trend of using technology based on field programmable gate arrays (FPGAs).

2. SUMMARY OF REGULATORY POSITION ON CCF

Current regulatory guidance on digital CCF is discussed in SECY 93-087 [1] and in BTP 7-19 [2]. On the basis of experience in digital instrumentation and control (DI&C) reviews, NRC staff members also established further guidance with the development of DI&C-ISG-02 [3]. However, Revision 6 of BTP 7-19 (issued July 2012) incorporates the content of DI&C-ISG-02 and is therefore the most relevant.

The basic premise of the NRC in BTP 7-19 (Revision 6) is that "software-based or software-logic-based digital system development errors are a credible source of CCF. . . . generally, digital systems cannot be proven to be error free and, therefore, are considered susceptible to CCF because identical copies of the software-based logic and architecture are present in redundant divisions of safety-related systems." BTP 7-19 categorizes firmware and logic developed from software-based development systems all under software.

The NRC's guidance on defense against CCF in BTP 7-19 is provided in a four-point position, which may be summarized as follows:

- Evidence shall be provided that the DI&C system has been adequately analyzed to identify and address any vulnerabilities to CCF.
- An analysis shall be made of each postulated common-mode failure for each event evaluated in the safety analysis report (SAR), and it shall be demonstrated that adequate diversity has been provided in the design for each of these events.
- A diverse means of performing a safety function shall be provided if the safety system providing that safety function is identified as being subject to a common-mode failure.
- An independent and diverse set of displays and controls for manual, system-level actuation of critical safety functions, and the monitoring of parameters that support the safety functions, shall be provided.

Additional guidance on digital CCF is also provided in the Interim Staff Guidance DI&C-ISG-04, "Highly-Integrated Control Rooms−Communications Issues." In particular, Item 2, "Command Prioritization," of DI&C-ISG-04, provides guidelines on priority modules used to combine diverse actuation signals with the actuation signals generated by the digital system to which they are diverse [4]. The guidance in the document may be summarized from the first two paragraphs of the staff position in DI&C-ISG-04:

> *. . . the priority modules that combine the diverse actuation signals with the actuation signals generated by the digital system should not be executed in digital system software that may be subject to common-cause failures (CCF). . . .*
>
> *An applicant should demonstrate that adequate configuration control measures are in place to ensure that software-based priority modules that might be subject to CCF will not be used later for credited diversity. . . .*

The reviews and discussions in this report are made in light of these basic guidelines.

3. KEY CHARACTERISTICS AND CAPABILITIES OF DIGITAL SYSTEMS

3.1 WHAT IS A DIGITAL SYSTEM?

Since the evolutions in safety system platforms essentially involve digital technology, this analysis sets the stage for the potential impact of digital technology on CCF. This section defines digital systems from an NPP regulatory perspective. The term *DI&C system* may vary in meaning, depending on the audience. Defining the term and scope as it is typically used in the NPP environment establishes the basis for addressing digital system issues of concern in NPP safety system applications.

Instrumentation and control (I&C) systems can be directly described in terms of their physical elements as implemented. Block diagrams and schematics are a convenient depiction of the detailed physical or functional layout of a system, module, or circuit. Figure 1 shows a functional representation of a typical NPP instrument channel from sensor to actuator. This functional representation represents analog I&C systems in which discrete elements may be represented by each block, and each block typically provides dedicated, hardwired functionality. The same figure with the same number of blocks could be used to represent a "digital instrument channel." However for digital I&C systems (unlike their analog counterparts), multiple functions can be realized in a single module via software implementation, and the complexity of the computational element makes it difficult to isolate specific functions or capabilities, therefore making it difficult to analyze the effect of failures of single functions.

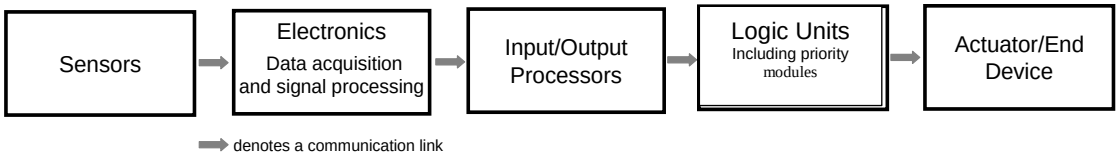

Figure 1. Block diagram of main elements of an instrument channel.[b]

A DI&C system contains a software- or firmware-based computational element [5]. The Institute of Electrical and Electronics Engineers (IEEE) Std. 7-4.3.2, "IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations" [6], describes the term *computer* as "a system that includes computer hardware, software, firmware, and interfaces." Thus, a DI&C system can be further described as a computational element that involves some kind of software- or firmware-based platform, or any type of reconfigurable and/or reprogrammable digital platform. FPGAs are included in this category.

This characterization of a DI&C system eliminates the inclusion of such digital-based hardware as digital panel meters. Such digital-based hardware typically performs a limited set of sequential tasks without allowing for reconfiguration or reprogramming. However, if such a device does contain an embedded microprocessor with the ability for a user to reconfigure it, then it may be classified as a digital system that falls within the scope of this report.

From a regulatory perspective, the term *digital equipment* is not only used for digital technology of new designs, but it is also used in the context of a digital upgrade to existing analog equipment in an NPP. These upgrades are often made to plant I&C systems, but they may also involve the replacement of mechanical or

[b]Note that IEEE 603's definition of *safety system* (or *safety channel* in the narrower context in this block diagram) does not include the sensor and actuator, only the interfaces to these devices. However, they are included here for completeness.

electrical equipment when the (new) equipment contains a digital device, such as an embedded microprocessor that performs control or monitoring functions [7].

Another important functional element of a DI&C system in the context of this definition is two-way communication. A digital two-way communication capability usually indicates the ability to remotely influence at least one of the parties involved in the communication (e.g., remote calibration, configuration, or diagnostics of a smart transmitter). While two-way communication within a safety system or between safety subsystems may be desirable, two-way communication between a safety system and a non-safety system is definitely not desirable due to issues such as the possibility of erroneous communication from a failure in the non-safety system that prevents the safety system from performing its safety function.

3.2 DRIVERS FOR THE USE OF DIGITAL TECHNOLOGY IN THE NUCLEAR POWER INDUSTRY

Some major drivers toward the use of digital technology in NPPs include the following:

- ***Technology obsolescence:***
  Although there have been digital upgrades in NPPs since the 1980s, and more recently there have been certified fully digital designs (with some operating worldwide), the majority of existing safety I&C systems in NPPs in the United States are still based on discrete component analog electronics and relay technologies. Developed in the 1960s and1970s, these systems have become difficult and costly to maintain, as most of the original equipment manufacturers are no longer in business and/or they are no longer providing technical support because they have not maintained their credentials as nuclear-qualified suppliers.

- ***Widespread use of digital technology in the non-nuclear industry:***
  Digital technology has been widely used in the non-nuclear environment for several decades, so data are available for assessing their reliability and safety. Many non-nuclear applications require high reliability (e.g., medical, aviation) similar to the nuclear industry.

- ***Lessons learned from accidents:***
  Accidents such as those which occurred at Three Mile Island, Chernobyl, and Fukushima tend to force I&C system designers to reevaluate operating principles, the ability to maintain functionality in spite of errors (i.e., system robustness), safety margins, etc. Three Mile Island and Fukushima both highlighted the essential role I&C plays in enabling operators to understand the nature of the accident they are facing. In particular, the Three Mile Island accident helped stimulate new research and development into signal validation, ultimately spawning the discipline of online monitoring [8].

3.3 STRENGTHS AND WEAKNESSES OF DIGITAL SYSTEMS RELATIVE TO NPP SAFETY SYSTEM CONSIDERATIONS

This section summarizes the unique capabilities and characteristics of digital technology that distinguish it from traditional analog technology:

- ***Digital systems have potential for high reliability***—Reliability is a measure of component ability (in this case a computer-related component) to consistently perform according to its specifications. In this context, reliability implies the probability that a system will be able to identify and/or remove a fault before it prevents a system from performing its function. Testing is the most common fault-identification and removal technique. Software introduces a powerful means of providing online embedded diagnostics and self-checking capabilities.

- ***Digital systems have high flexibility***—Flexibility is a key attribute of digital systems. A digital system may be designed to be configurable and portable. The capability for remote reconfiguration (such as smart transmitters) typically implies capability for two-way communication between the digital system and human interface. The capability for remote configuration may be seen as both a strength and vulnerability of the digital system. In particular, the ability to remotely configure a system also increases the risk of deliberate unauthorized intrusion to cause harmful changes. Closely related to the ability to remotely configure a system is the ability for remote calibration. The strengths and vulnerabilities are also similar.

- ***Digital systems typically have multiple functionality***—A digital system may be designed to perform multiple functions (e.g., acquire input data, process the data, perform onboard diagnostics, monitor alarmed conditions). With today's sub-micron integrated circuit feature sizes, this usually means that all the functions could reside in a very small space. Thus, failure of one integrated circuit could result in failure of multiple functions. In addition, important functionality is often integrated into servers and processors. The implication of this is that some performance parameters such as transmission speed and response times may deteriorate with a growing size of the I&C system due to higher processing loads. This characteristic has the potential to negatively affect important plant or I&C functions such as the quality of closed loop control and reaction times of the human-system interactions [9].

- ***Digital systems generally are configured to employ common networking hardware***—Depending on the location of the multiplexing, networked communication has the advantage of reducing the number of containment penetrations. However, sharing common bandwidth can result in information bottlenecks during high load conditions. Moreover, digital systems can also suffer dead time due to shared components employing sequential access to the communication network (see below).

- ***Information processing is fundamentally sequential in nature***—Analog and digital circuitry at NPPs are a means of acquiring signals from sensors and communicating the measured values to guide safety or control actions. Analog circuitry is traditionally hard-wired and dedicated to specific tasks. In contrast, digital systems signals are sampled and digitized, and the resulting information is transmitted and processed sequentially. This means that existing functional specifications such as response time and dead time must be reconsidered in detail before they are applied to the new DI&C design [9,10].

- ***CCF in Analog systems is fundamentally different from that in digital systems***—The error modes of analog components and circuitry (hysteresis, drift, signal failure, etc.) are well understood, and CCFs in analog-based systems have been attributed to slow processes [11]. However, CCF in digital systems may be triggered by a latent fault in the software, and the resulting response in all redundant systems is typically too fast to be corrected by operator intervention. In addition, software does not age in the conventional sense, so any embedded error cannot be identified by periodic maintenance as in analog systems.

- ***The complexity of digital systems makes licensing more challenging***—When licensing DI&C, it is difficult to assure sufficient testing of the software. Even a small software module can exhibit enough complexity to make a full verification of its correctness within reasonable cost and schedule impractical. As described above, the assumption is that there is some probability that a latent error not discovered during the verification and validation (V&V) process may disrupt its function in a crucial situation. In this scenario, building (software) redundancy into the system cannot remedy the situation because the software is deterministic in its operation, and each redundant channel will have

the same embedded error. Even the use of software diversity cannot guarantee adequate protection against such potential for CCF because the requirement specification may be the ultimate cause of a software error [9]. In essence, a (complex) digital system is fundamentally non-linear, so it is difficult to model and/or predict its behavior.

On the other hand, additional complexity enables additional functionality that can provide substantially greater confidence in correct operation in analyzed circumstances. This is accomplished through self-checking and health monitoring. Moreover, additional capabilities of digital systems can reduce the conceptual workload of the human operators (and thereby increase the probability of taking correct actions) by providing interpreted data in more easily understood formats. The goal of the nuclear power regulatory process is to provide reasonable assurance of adequate safety. The greater confidence in correct operation offered by self-diagnostics must be balanced against the potential for digital instrumentation to contain unrevealed errors that are technically difficult to correct.

4. A REVIEW OF TECHNOLOGY IMPLEMENTATION CHANGES IN DIGITAL SAFETY SYSTEMS FROM THE 1980s TO PRESENT

4.1 INTRODUCTION

This chapter examines technology changes in digital safety system implementations from the 1980s to present. The objective is to qualitatively assess the adequacy of current guidance on digital CCF by identifying any gaps in the current regulatory positions that need to be addressed. The following three representative safety systems were selected to provide illustrative examples:

- Eagle 21 was selected to represent vintage microprocessor-based technology from the 1980s to mid-1990s,
- TXS was selected to represent second generation microprocessor-based technology (mid-1990s to 2000s), and
- The ALS was selected to represent the latest trend of using FPGA-based technology.

The information described in this section is presented to highlight key characteristics, including hardware and software architecture, memory, communication, redundancy, and the software V&V process.

4.2 THE EAGLE 21: REPRESENTATIVE MICROPROCESSOR-BASED SAFETY SYSTEM FROM THE 1980s TO MID-1990s

4.2.1 Architecture of the Eagle-21

In the 1990s, digital safety systems based on the Westinghouse Eagle-21 platform and the Foxboro Spec 200 Micro platform were licensed and implemented in a few US NPPs. The Eagle-21 platform is discussed in this section as a digital safety system representative of that period.

Hardware Architecture

The Eagle-21 Process Protection System was designed as a modular microprocessor-based functional replacement for existing analog equipment (Figure 2). It was designed to provide three or four instrumentation channels and outputs to two trip logic trains for each protective function. The input, processor, communication, and output modules all used separate single board computers or controllers. The processor used was the Intel iSBC 286/12 single board computer. This was a 16-bit single board computer designed as a board level solution for real-time, multi-tasking, and multiprocessor system applications. Intel iSBC 286/12 boards served as the loop calculation processor (LCP), test sequence processor (TSP), and man-machine-interface (MMI) processor [12].

Another single-board computer, the Intel iSBC 88/40A, was used as the measurement and control computer. The iSBC 88/40A provided 16 differential input channels and was used as both an A/D converter and a digital filter processor (DFP). This subsystem (iSBC 88/40A [DFP]) reads analog inputs, performs analog-to-digital conversion, makes input calibration readings and adjustment, performs onboard diagnostics, and performs digital filtering. After the iSBC 88/40A reads the input data, they are placed into memory for access by the LCP [12].

The Intel iSBC 88/45 was used as the communication controller for the loop processor subsystem multibus, the tester subsystem multibus, and the MMI test cart multibus. The loop processor subsystem uses an iSBC 88/45 to transmit data to the tester subsystem. The tester subsystem uses its iSBC 88/45 to receive data from the loop processor subsystem and to transmit and receive data to and from the MMI test cart. Likewise, the MMI test cart uses its iSBC 88/45 to transmit and receive data to and from the tester subsystem.

The Intel iSBC 519 was a programmable input/output (I/O) expansion board used in the Eagle-21 design to process digital I/O signals for both the loop processor and tester subsystems. The loop calculation and test sequence processors interfaced with their associated iSBC 519 and either read a signal which represents a digital input or wrote a value that the 1SBC 519 converts to a digital output.

The main features of the Eagle-21 system included the following [12]:

- automatic surveillance testing to significantly reduce the time required to perform surveillance tests,
- self-calibration to eliminate rack drift and time consuming calibration procedures,
- self-diagnostics to reduce the time required for troubleshooting,
- significant expansion capability to allow for rack consolidation and easily accommodate functional upgrades and plant improvements, and
- modular design to allow for a phased installation into existing process racks and use of existing field terminations.

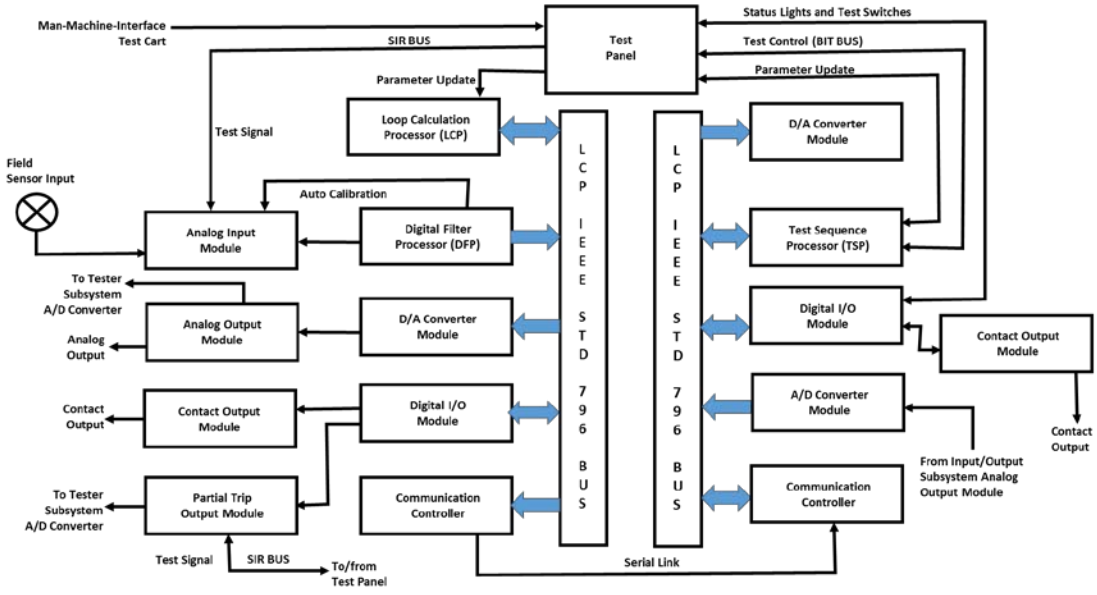

Figure 2. Eagle-21 architecture [12].

Software Architecture

The Eagle-21 process protection system software uses a modular approach in its design, with all executable code contained in modules or subroutines. This modular approach is depicted in Figure 3 [12]. The main program contains a restart section (modules A–D in Figure 3) and a looping section (modules 1 through N). The restart section contains initialization routines and is executed only once on restart, while the looping section, which contains process function routines, continually executes. The overall software development follows a general format of four layers, with the first and bottom layers containing the main program and support functions.

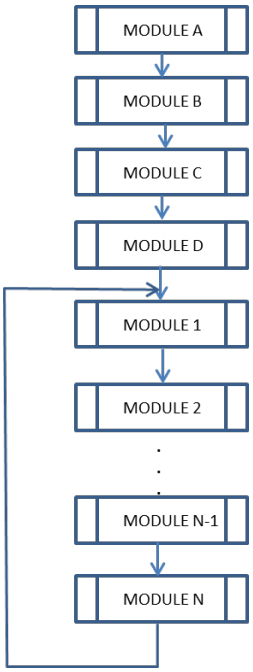

Figure 3. Modular approach used in the software development of the Eagle-21 [12].

The second layer is a library of general purpose modules. The third layer is a library of standard protection functions built primarily from general purpose modules. The fourth and top layer is the configuration layer. This layer contains plant-specific information, which tailors the generic functions to project-specific applications. The configuration layer typically represents approximately 0.5 percent of all code. Reference [12] indicates that this provides a high degree of confidence in the overall software code because the bottom three layers are standardized and do not change from project to project. Only the configuration layer requires programming for specific applications.

Reference [12] provides representative samples of each of the four layers of software, as follows:

Layer A: *Main Program and Support Routines*

- online diagnostics
- engineering unit conversion
- self-calibration
- limit checking
- program sequencing

Layer B: *Library of General Purpose Modules*

- summation
- square root
- multiplication
- division
- lead/lag
- high select
- low select
- high trip comparator

- low trip comparator

Layer C: *Library of Standard Protection Functions*
- average temperature and delta temperature
- pressurizer pressure
- pressurizer level
- containment pressure
- steam generator water level

Layer D: *Configuration Layer*
- plant-specific tag numbers
- analog input assignments (channel trip assignments, setpoints and tuning constants)

All of the executing software is supplied in programmable read only memory (PROM), so the information is permanent and cannot be erased or deleted. Configuration parameters are stored in electronically erasable PROM, which allow updates without having to remove the chips. Reference [12] states that all software follows the standards established for software design, which include the following:

a) High-level, easily maintained language is used in system development except where necessary for reasons such as timing.
b) No interrupts are allowed.
c) No re-entrance is allowed.
d) Code format conforms to standards for high-level and assembly language routines.
e) GO TO statements are not allowed.
f) All modules are single task (no operating system or multi-tasking system).
g) All modules are single entry, single task.
h) Modules exit to points of call.
i) Each module has a design performance specification and a verification test specification.

4.2.2 Memory Organization of the Eagle 21

The Eagle 21 memory is organized in relation to the software category. Level 1 software functions reside in Level 1 memory, and Level 2 functions reside in Level 2 memory. Level 1 software is "associated with actuation and/or implementation of reactor trip, engineered safety features, and information displays for manually controlled actions as defined by IEEE Std. 279-1971 and IEEE Std. 603-1980"[12]. Reference [12] describes several criteria used to differentiate between Level 1 and 2 software with regard to verification testing. These criteria are organized under various categories. One category is *memory organization*. In particular, the following criteria are applied to all software units with regard to memory organization. If all the conditions are met, the software is Level 2. Otherwise, the software is Level 1:

a) The software design does not permit writing to areas of random access memory (RAM) used by Level 1 software functions.
b) The software design does not permit inhibiting access to memory locations used by Level 1 software functions.
c) Software is not part of, nor can it alter, the execution path for Level 1 software functions.

4.2.3 Communication in the Eagle 21

There is no interdivisional digital communication in the Eagle 21. The partial trip output module shown in Figure 2 provides the interface between the Eagle-21 hardware and the trip logic system. The partial trip

output module converts a signal from the loop calculation processor into an ON/OFF voltage used to drive relays in the trip logic system.

Communication between modules in one division to the outside world is by serial communication. For example, the tester subsystem communication controller provides a serial link to the test panel, which allows for information display and printing when connected to the MMI test cart.[c]

4.2.4 Redundancy in the Eagle 21

The Eagle 21 protection system receives inputs from sensors, performs calculations on the values, and compares the results to predetermined setpoints. If the limits are exceeded, a partial reactor trip is generated. The partial trip signals from four redundant protection divisions form the inputs to a voting logic system that then generates a reactor trip. The generation of engineered safeguard system actuations (intended to mitigate the effects of undesired events) follow a similar path. The process protection system also provides isolated signals for use by non-safety systems such as the control system, the plant computer, and portions of the control board.

4.2.5 V&V Approaches Used in the System Development of the Eagle 21

The Eagle 21 development involved a system definition stage, a system design stage, and a system implementation and testing stage. Each stage is briefly discussed below [12] (see also Figure 4):

The Definition Stage:
During the definition stage, project objectives and an initial project plan were defined, along with high-level system design and functional requirements. These elements were all documented in a system design requirements document. An independent V&V team reviewed the functional requirements.

Design Stage:
During the design stage, the system design requirements were decomposed into system design specifications, as well as hardware and software design specifications, in sufficient detail to enable system implementation.

Implementation and Test Stage:
Hardware construction, along with software coding and testing, were performed during the implementation and test stage. Each software entity that was completed was turned over to the code verifiers for independent review and testing, beginning at the unit level. After it was verified that all software modules necessary to accomplish a software subprogram met the applicable software design specifications, the subprogram itself was verified as meeting applicable software design specifications. It was also verified that the appropriate software modules were used to generate the subprogram entity.

As part of the testing, various hardware and software components were assembled in a stepwise manner, and additional testing was performed at each step to ensure that each component performed its required function when integrated with its associated components. After integration of the various software and hardware subsystems, factory acceptance testing (i.e., testing of the entire system) was performed. This included development of az system test plan based on functional requirements and design specifications. Tests were performed to confirm that the system met requirements.

[c]The tester subsystem provides the interface for human interaction with the Eagle 21 protection system. Together with the MMI test cart, it provides the interface which allows test personnel to adjust setpoints and tuning constants and perform surveillance tests on the protection system.

Formal design reviews were held during implementation stage to ensure that the system design specifications met the functional requirements. The design review team consisted of knowledgeable, multidisciplinary engineers who ensured that all aspects of the design were reviewed.

4.2.6 Software Testing Methods Used for the Eagle 21

The Eagle-21 applied both white box (structural) and black box (functional) testing during the system development. White box testing attempts to "look under the hood" to see what is happening inside the application. It comprehensively exercises the software and requires that the verifier inspect the code and understand how it works before selecting the test inputs. White box or structural testing is typically performed at the software unit level, and the test inputs are chosen to exercise all the possible control paths within the software components. A software unit may be viewed as the smallest testable part of an application. In procedural language programming such as that used in the development of the Eagle 21, a unit could be an entire module, although it more commonly refers to one function or procedure. The context in which the term *unit* is used is not clear based on the review of Eagle 21 documentation.

In black box functional testing, which was also used to test the software of the Eagle 21, the internal structure of the program is ignored during the test data selection. Tests are developed based on module or system design specifications. Random testing is a form of black box testing. In this test, an input sequence of tests is selected at random. The method [12] is used to :

- simulate real time events that are truly random,
- increase the confidence level in the correctness of a complex program,
- test a system or a subsystem where it is not necessary to test all the possible paths,
- get a quantitative measure on the accuracy of a numeric calculation, and
- get a measure of the average time required by some calculation.

In general, testing cannot guarantee that every error is caught. This is because testing cannot evaluate every possible execution path except for in the simplest of programs. In addition, even with white box testing, which is typically used at the software unit level, only unit-level errors can be caught. White box testing will not detect integration or system-level errors.

4.2.7 Observations and Insights from the Eagle 21 Technology Implementation Review

Following are the observations and insights resulting from the Eagle 21 technology implementation review:

a) Early microprocessor-based safety system implementations such as the Eagle-21 process protection system was designed as a modular *functional replacement for existing analog equipment.* Starting from the premise that analog systems are mature technologies and that their review processes are stable, strict adherence to digital functional replacement for existing analog equipment was seen as limiting the potential for digital CCF.

b) Although early digital implementations were typically one-for-one replacements of the proven analog designs as exemplified by the Eagle 21, some advantages of digital technology (e.g., onboard diagnostics) were also implemented. For example, the Eagle 21 implemented automatic surveillance testing (to reduce the time required to perform surveillance tests), self-calibration (to eliminate rack drifts and time consuming calibrations), and self-diagnostics (to reduce the time required for troubleshooting). The drawback of implementing these software enhancements was the need to assure deterministic software behavior in spite of the additional software overhead. In the Eagle 21, deterministic performance was implemented according to the following specifications:

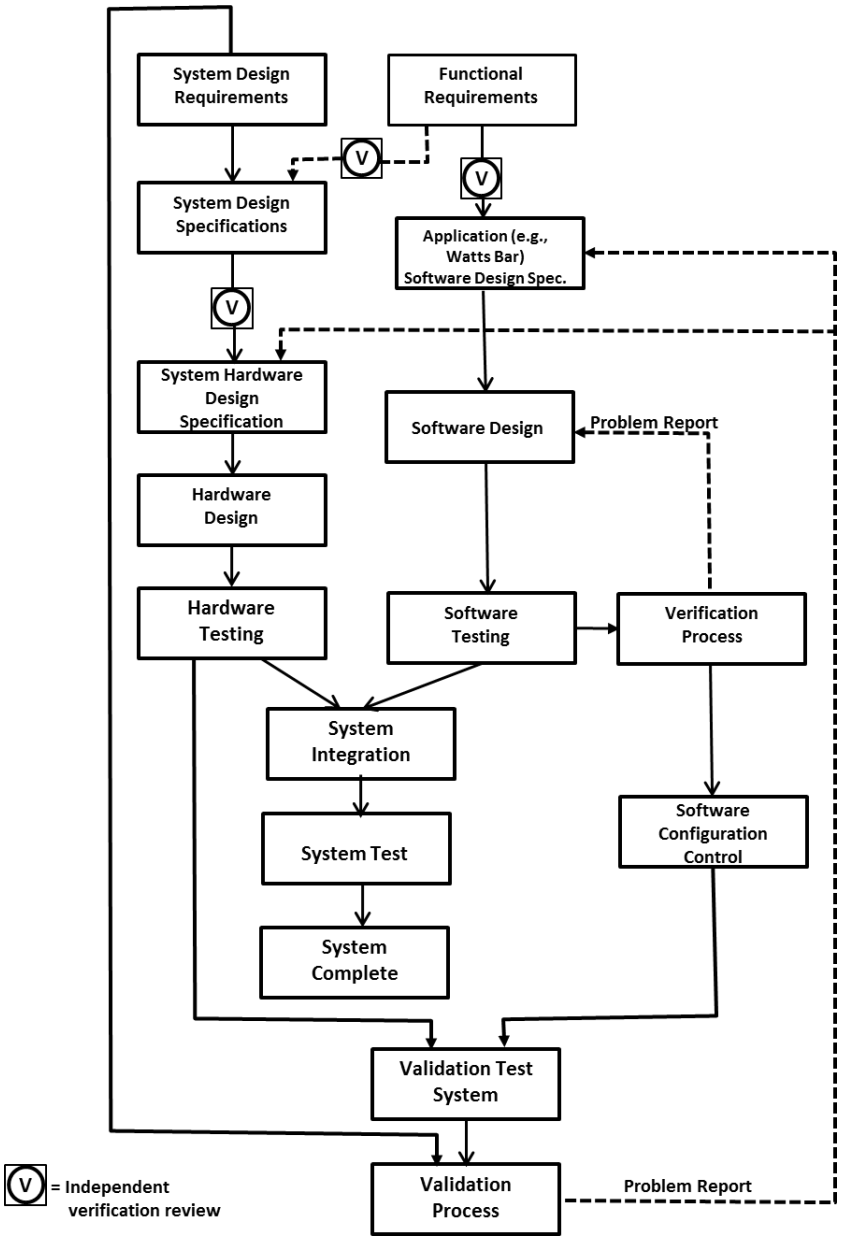

Figure 4. Design and V&V processes for Eagle-21 system development [12].

1. A modular approach is used in the software design, with all executable code contained in modules or subroutines
2. No interrupts are allowed.
3. No re-entrance is allowed
4. Code format conforms to standards for both high-level and assembly language routines.
5. GO TO statements are not allowed.
6. All modules are single task (no operating system or multi-tasking system).
7. All modules are single entry, single task.
8. Modules exit to points of call.
9. Each module has a design performance specification and verification test specification.

However, these specifications alone do not necessarily guarantee sufficient determinism. With the added overhead of onboard diagnostics and surveillance software, each module and each complete cycle should also be guaranteed to complete in a predetermined time.[d]

c) The software V&V and digital communication standards and guidelines available in this period were generally adhered to in the system development. Since then, there have been considerable improvements in these standards and guidelines (e.g., DI&C-ISG-02) which now address issues such as interdivisional communication. However, because the early digital safety system implementations tended to be one-for-one replacements of analog systems with no interdivisional communication, etc., the early standards and guidelines were adequate for the period.

4.3 TELEPERM® XS: REPRESENTATIVE MICROPROCESSOR-BASED SAFETY SYSTEM FROM THE MID-1990s TO 2000s

4.3.1 Architecture of the TXS

The TXS is an example of the evolution of DI&C safety systems from one-to-one replacements to fully digital computer-based systems suitable for new plants as well as upgrades. The TXS consists of all the necessary hardware and software components (including software tools) required for engineering, testing and commissioning, operation, and troubleshooting for an NPP I&C system.

Hardware Architecture
TXS is a distributed, redundant computer system. It consists of three or four divisions, each consisting of data acquisition, signal conditioning, data-processing and actuation signal voting running asynchronous with respect to each other. The communication between redundant channels uses end-to-end fiber optic cable connections. In conjunction with the SPPA-T2000 (which was initially named the TELEPERM XP for operational I&C), TELEPERM XS supports the configuration of an integrated overall plant architecture as shown in Figure 5. The architecture consists of the following building blocks [13,14]:

- The system hardware platform consists of distributed computers (processor modules, communication modules, and suitable I/O modules) which handle tasks including acquisition of process signals, signal conditioning, and filtering, actuation of final control elements, and annunciation of process conditions and faults. The processor module has a 32-bit processor and onboard RAM for executing programs, flash erasable programmable read only memory for storing program code, and electrical erasable programmable read only memory for storing application program data. The automation program is loaded from a FLASH memory and is executed cyclically. This cyclical execution

[d] It is possible that this was also implemented in the early digital software safety systems such as the Eagle 21. However, the authors were unable to ascertain this from the available documentation.

involves control of I/O modules, processing of the automation program and self-test routines, and data exchange via communication modules and bus connections.

- System software consists of a set of quality-controlled software components, the execution of which is based on operating system software developed by Siemens specifically for the TXS systems. The operating system communicates with the platform software and the application software. The platform software includes the runtime environment program that provides a unified environment for execution of the function diagram modules.

- Application software performs the plant-specific TXS safety-related functions using function block modules which are grouped into function diagram modules. The application software is generated by specification and coding environment (SPACE) tools which use the qualified software modules from the function block library to construct a specific application. Such programs are stored in flash erasable programmable read only memory (FEPROM).

- The SPACE tool is an engineering software system used to implement the requirements of plant-specific I&C features. In particular, the functions to be implemented in TXS I&C systems are specified by means of SPACE in graphical form. SPACE software tools are discussed further in the next (software) section.

A complete I&C system for an NPP of course includes more than just the safety I&C system. Figure 6 shows a fully automated I&C system for an NPP consisting of the TXS (for safety I&C) connected to a Siemens SPPA-T2000 (for non-safety balance of plant I&C) via one-way communication. The SPPA-T2000 is connected to the priority modules[e] of the TXS via the PROFIBUS DP. The PROFIBUS DP link also implements isolation between the two systems.

The TXS includes several standard self-monitoring mechanisms, some of which form part of the system platform, and some of which are configured by means of application-specific engineering. Self-monitoring mechanisms that are part of the system platform are the following [14]:

- cyclic testing of program memories,
- permanent communication monitoring,
- monitoring of cycle time by means of software and a hardware watchdog,
- automatic testing of the watchdog,
- self-testing of the inputs of input modules, and
- automatic read-back of the outputs of output modules.

Reference [14] discusses “system features that support high reliability,” including deterministic system behavior, the most important features of which include the following:

- strictly cyclic processing of application software,
- bus systems with a constant load,
- complete absence of process-driven interrupts,
- static memory allocation,
- no processing of absolute time or date (no real time clock), and
- no long-term data storage and no use of external data storage media.

[e]A priority module is needed to manage the priorities assigned to individual commands in situations where final control elements are used by both the operational I&C and the safety I&C.

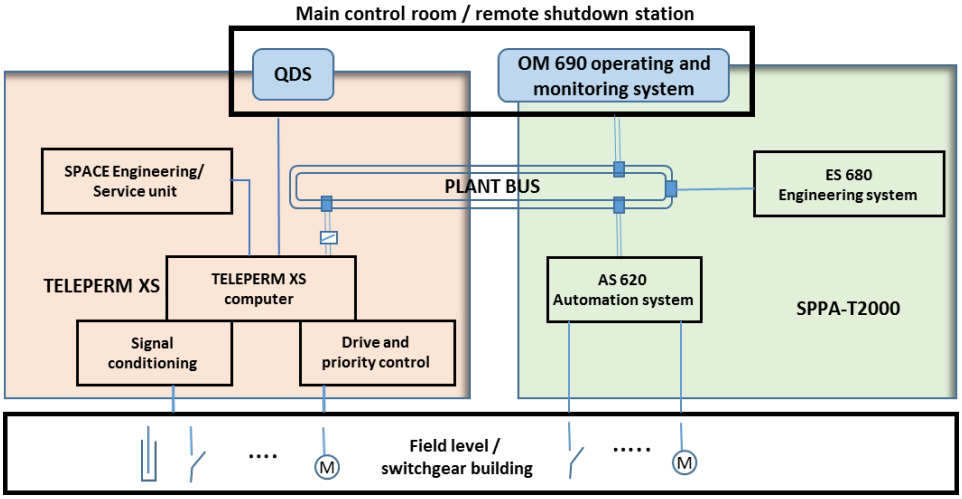

Figure 5. Complete plant I&C architecture based on TELEPERM XS safety system [14]. Note that the SPPA-T2000 was originally called "TELEPERM XP."

Software Architecture

The TXS consists of three layers of software: application software, platform software, and operating system software. These layers are depicted in Figure 6 and briefly described below:

Application Software:

The application software consists of the following modules [13]:

a) Function diagram group (FDG) modules include sets of function diagram (FD) modules. An FDG-module groups together all the FD-modules executed on the same processor in the same cyclic frequency.

b) The function block (FB) modules are the basic software function primitives of a library. This library consists of the implementations of about 120 common I&C function elements.

The FD and FDG modules provide the functionality of the I&C functions that control the technical process. FB modules are used to compose the FD and FDG modules. The FB modules are the elementary software components of the application software, providing the basic logic and arithmetic functions (e.g., AND, OR, 2 out of 3, limit monitor, etc.). Each block has a graphical representation in the function diagrams. The function blocks can be thought of as the vocabulary of a formal specification language.

As indicated in the previous section, the application software for the TXS is generated by SPACE tools which use the (qualified) software modules from the FB library to construct a specific application. Program and data that do not change over the fuel cycle of the plant are stored in FEPROM. This data cannot be changed without first erasing the data stored in the applicable FEPROM 64 Kb sector and then rewriting the entire FEPROM sector. To ensure that the application software and data stored in the flash EPROM stay unchanged, FEPROM data integrity is checked by a self-diagnostic routine that calculates the cyclic redundancy check (CRC) value of each 64Kb sector of the FEPROM and compares the results to the CRC values stored in each 64 Kb sector of the FEPROM with the application software and invariable data [15].

Data that are subject to change over the nuclear plant fuel cycle are stored in redundant electrically erasable programmable read only memory (EEPROM). Examples of data that may be stored in EEPROM include plant system parameters and setpoints that may require changing by the plant operator, as well as the loader for programming the flash EPROM. Unlike the permanent data and programs stored on flash EPROM, data stored in EEPROM may be changed without first erasing all the data stored in a block of memory in the EEPROM [15].

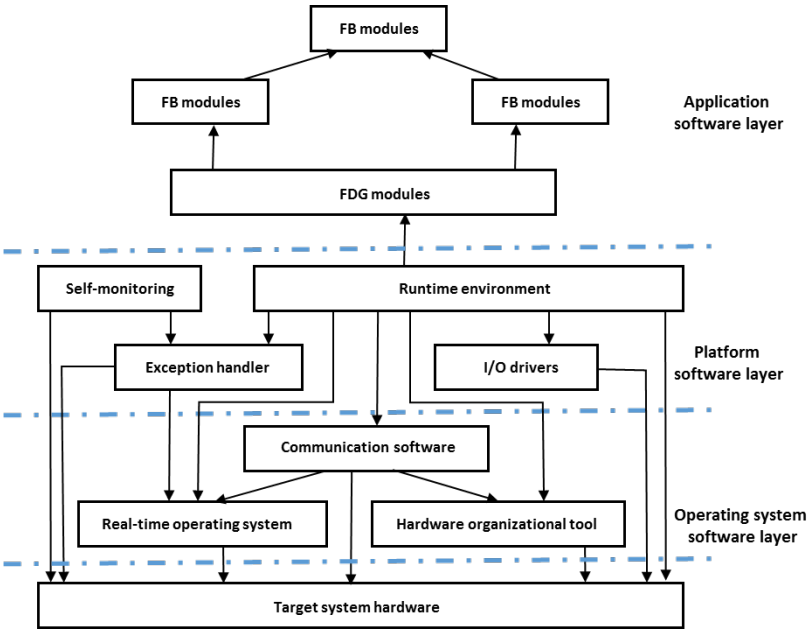

Figure 6. Software layers of the runtime system in one processing module [13].

The process described below is followed to create a specific application project [15]:

1) Define the hardware specification. The hardware specification contains the complete hardware structure of the target system with all of its components and is created using the SPACE editor. The SPACE editor is a graphical user interface tool to create I&C function diagrams and hardware diagrams. The information is stored in the specification database.
2) Use SPACE code generators to interpret the contents of the specification database and to automatically generate high-level language code (in C language) for each function diagram. Communication between function diagrams is accomplished using data messages. These are automatically generated by interpreting the hardware specification and the software-hardware assignment. Thus, the complete code for all function diagrams is automatically generated. Independent tools are developed to perform automatic code verification. The SPACE tools parse the generated code, transform it into an internal representation, and compare this representation to the information stored in the specification database.

Platform Software:
The platform software consists of the run time environment (RTE) and its modules, the I/O drivers for the I/O module interface, the exception handler, and the self-test software [17]. The RTE has two major interfaces: the interface to the FDG modules (in the application layer), and the interface to the operating system's software layer/target system hardware. The RTE provides a unified environment for executing the

FDG modules. It also controls the cyclic processing of the FDG modules and controls signal transfers via messages or directly by I/O modules. Finally, it provides the interface of the runtime system software to an external service unit through which it can be monitored and controlled. In effect, the RTE is essential to all aspects of TXS communication. The internal system clock triggers every millisecond and controls all actions during the processing cycle. The central control unit sequentially starts the main processing phases in each processing cycle (typical cycle time is 50 minutes). Below is the sequence of operations for a typical real-time processing cycle:

1) Read input data.
2) Perform checks on received messages.
3) Process application software.
4) Handle transmitted messages.
5) Transfer output messages.
6) Process diagnostic programs for the remaining processing cycle time.
7) Process self-monitoring programs.

4.3.2 Communication in the TXS

Communications within Safety Divisions and from Safety to Non-Safety Equipment
In the TXS design, there is communication between redundant Class-1 E channels and from Class-1 E channels to non-Class-1 E devices. The communication between Class-1 E channels uses end-to-end fiber optic cables. Communication from the safety I&C system to the non-safety plant information system is accomplished via the monitoring and service interface (MSI), which serves as a means of isolation within the TXS architecture. The MSI serves as a gateway to non-safety-related systems such as service units, process control computers, and monitoring computers. The safety protection system signal passes through the MSI to display information at the main control board. As specified in the Software Program Manual for TELEPERM XS™ Safety Systems [15], the non-safety-related service unit requests access through the MSI to perform the diagnostic function at the safety-related processor. The manual also indicates that "the TXS design requires that in case of a single failure of one of the independent processing channels or within one communication path in the same processing channel, the channels still available will continue to operate as designed on the basis of the remaining information to ensure the required safety functions do not fail."

4.3.3 V&V Approaches Used in the System Development of the TELEPERM® XS

Figure 7 summarizes the software quality assurance (SQA) processes for the TELEPERM® XS [15]. The SQA plan describes the necessary processes to ensure that the (safety system) software has been developed to a level of quality commensurate with the prevailing standards and guidelines. The manual states that the documentation required for each (software) project is produced and independently reviewed, taking into consideration the quality factors listed in BTP 7-14 Section B.3.3, "Acceptance Criteria for Design Outputs." Furthermore, software reviews are conducted in accordance with IEEE 730-2002 and 1028-1997 (endorsed by Regulatory Guide 1.168) [16, 17, 18]. Reviews were conducted throughout the software lifecycle and were meant to verify that the software products of each phase were correct with respect to the phase inputs and outputs. The manual indicates that several audits were also conducted throughout the software life cycle and that these audits "provided an independent evaluation of conformance of the software products and processes to applicable regulations, standards, and procedures." These audits were also reported to have been performed using an independent qualified lead auditor and included, wherever possible, technical resources such as V&V personnel, as necessary.

According to the manual, the application software V&V for the TXS was performed in accordance with IEEE Std. 1012-1998 [19], in which Section 1.6 allows for customization of the task lists[f] (e.g., combination or elimination). In this regard, the V&V for the TXS was modified from the stipulations in IEEE 1012-1998 as follows:

1. The test tasks were modified to address generic TXS platform testing.
2. A separate hazard analysis was not performed for TXS technology. Instead, the set of analyses performed as described in the software safety plan constituted the hazard analysis for TXS systems.
3. An additional implementation activity criticality analysis task was not performed. Criticality was determined during the requirements and design phases based on the TXS technology attributes and the project-specific network design.

Apart from the modifications described above, software V&V procedures were performed according to the standard for all of the phases, as follows:

- Concept phase V&V tasks were performed to address system architectural design and system requirements analysis. The objectives were to verify the allocation of system requirements, verify the selected solution, and ensure that no false assumptions had been incorporated in the solution.

- Requirements phase V&V tasks were performed to address software requirements analysis. The objectives were to ensure the correctness, completeness, accuracy, testability, and consistency of the requirements.

- Design phase V&V tasks were performed to demonstrate that the software design correctly, accurately, and completely reflects the software requirements and that no unintended features are introduced.

- Implementation phase V&V tasks were performed to verify that the software design was correctly translated into code.

- The test phase V&V tasks were performed to (a) verify that the requirements in the system design requirements document (SDRD) were correctly implemented into the fully integrated system, (b) validate project-specific system performance, and (c) validate that the software requirements in the system design document (SDD) were correctly implemented into the application. Activities included verifying acceptance test documentation and validating the application software design with testing.

- **Independent Testing and Validation tasks:**
  The qualification testing process for the TXS was a two-part process: generic (application-independent) system qualification and specific (application-dependent) system qualification. The application-independent qualification of the TXS system included the type test of the hardware and software components and the plant-independent system test. The generic qualification work provided the foundation for the application-dependent system testing. That is, the application-dependent phase took credit for all application-independent qualification activities.

[f] V&V tasks based on the integrity level of the software.

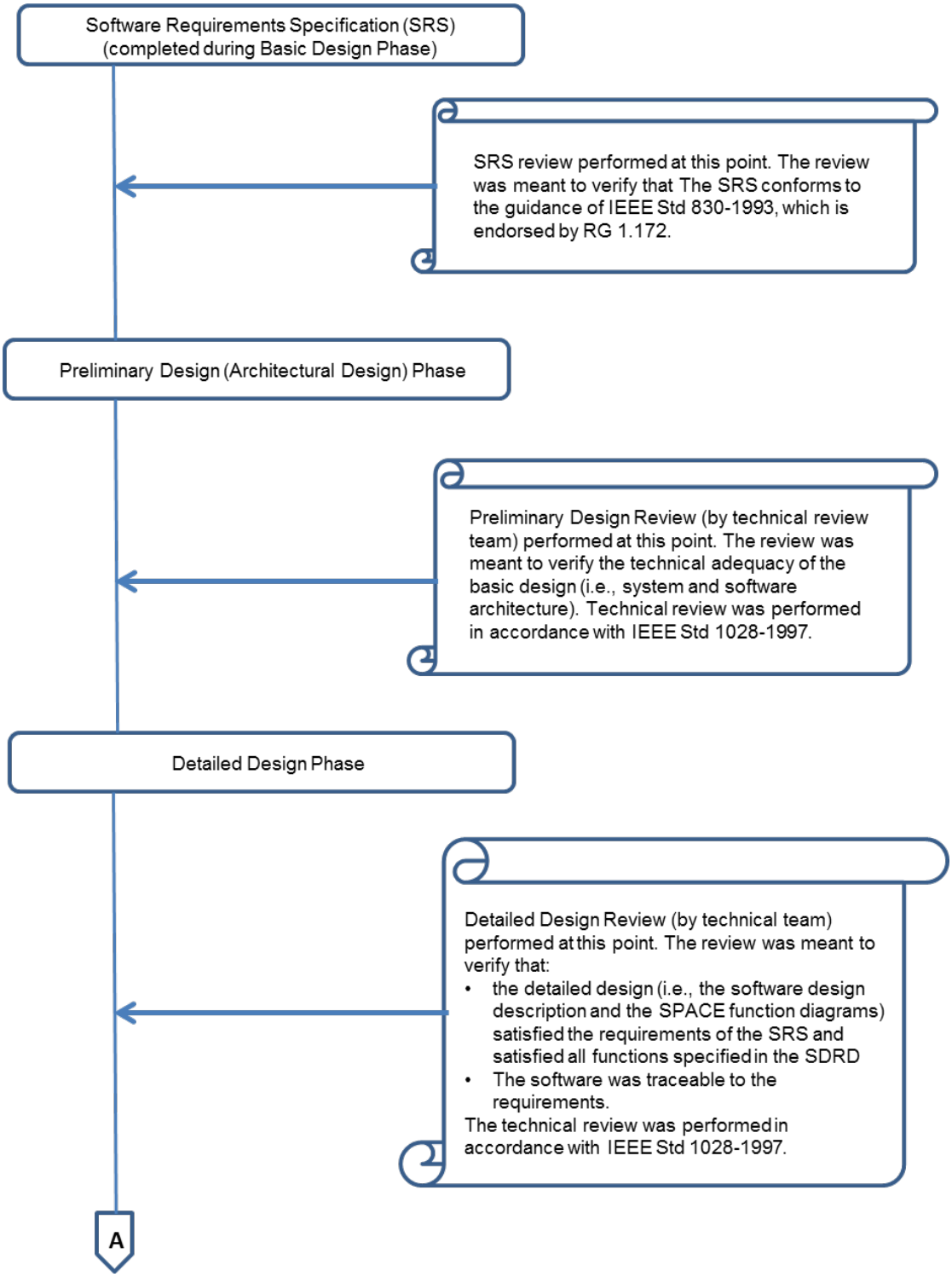

Figure 7. Software reviews conducted for quality assurance for the TELEPERM XS.

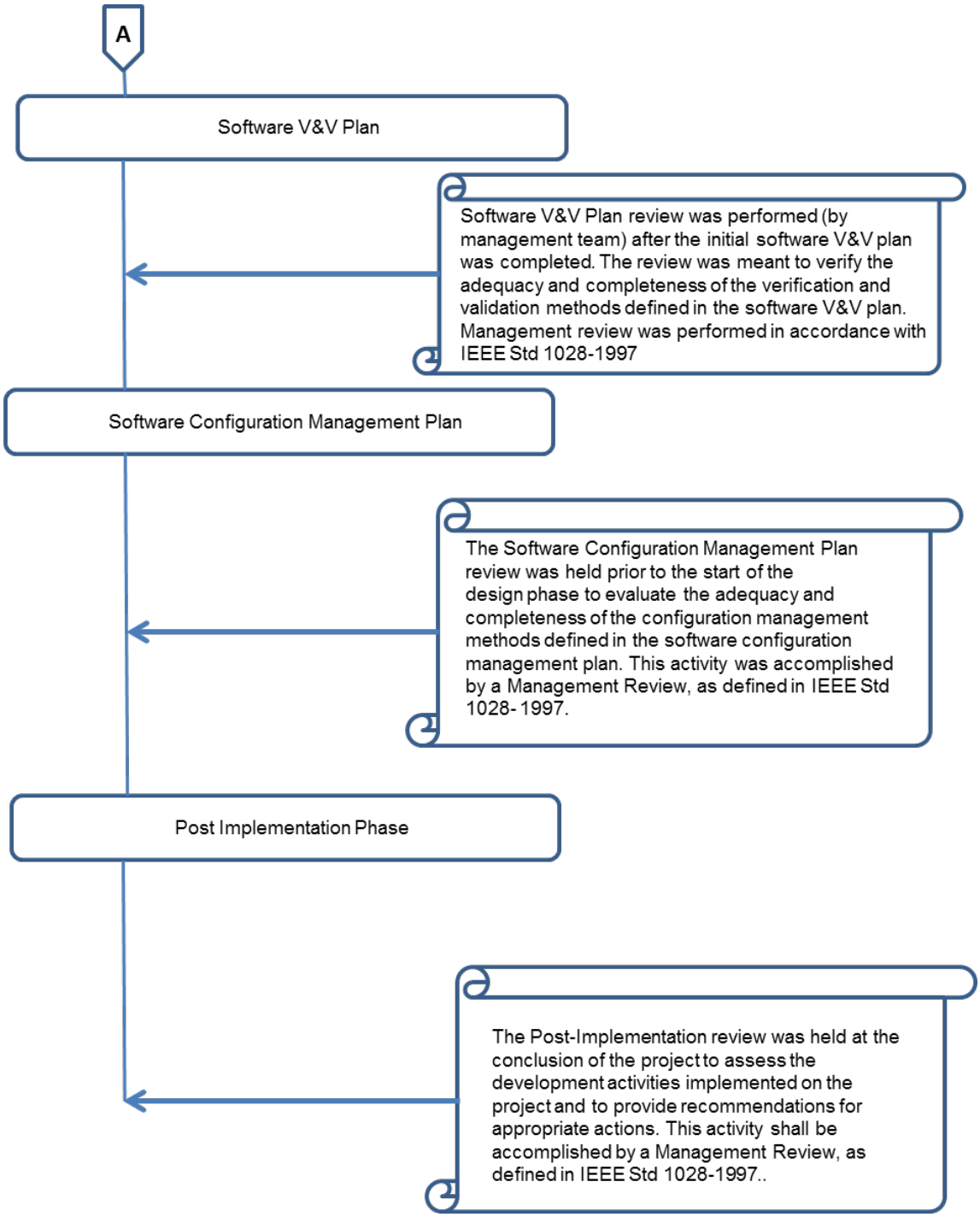

Figure 7. Software reviews conducted for quality assurance for the TELEPERM XS (continued).

Software Type Tests:
Standards that were applied in the development and evaluation of the software type-tests are IEEE 730, IEEE 828, IEEE 1012, and International Electrotechnical Commission (IEC) 880 [15]. The software development of each individual TXS component followed engineering procedures as described in Sect. 3.2.1.2 of the software manual. The same compiler and linker was used (with a restricted set of options) to develop each software component. According to the software manual, this was done "to qualify the compiler and the linker during the development by good service records, because each piece of software was extensively tested to meet the IEC 880 requirements for C0 and C1 test coverage". The development documentation, the source code, and the object code were then submitted for type-tests. The main objectives in the type-tests of TXS software were:

- To evaluate the software development process according to the software life-cycle, and
- To evaluate conformity to the coding recommendations of IEC 880.

Tests were performed according to the following procedure:

- the manufacturer provided a test specification of the software components,
- the independent body confirmed this test specification by checking it and asking for additional or different tests where appropriate,
- the manufacturer carried out the tests in line with the agreed-upon test specification and summarized the results in a test report, and
- on the basis of this test report, the independent body stated whether the software successfully passed the software type-tests.

The most significant evaluation criteria were the following:

- the completeness of the test
  - ✓ *Were all relevant features of the component covered by the test?*
- the suitability of the test
  - ✓ *Was the selected test method adequate to demonstrate the features?*
- the test coverage
  - ✓ *Has the test item been reached or "covered" by the test cases?*

According to Ref. [14], a subset of the following test methods was applied to verify the features in accordance with Appendix E of IEC 880, depending on the software to be tested:

- statistical tests,
- black-box test,
- white-box test,
- path testing,
- coverage testing,
- execution time testing, and
- boundary test.

4.3.4 Observations and Insights from the TXS Technology Implementation Review

The following are the observations and insights obtained from the technology implementation review of the TXS:

a) Evolution of safety system implementations simply made use of more sophisticated microprocessors and increased online self-testing and surveillance, as exemplified by the TXS. However, these systems (TXS) also made use of the improving guidance for digital safety system implementations (e.g., updated V&V standards) and improved on implementation of deterministic performance. For

example, the digital system architecture of the TXS included procedures that improved determinism such as (a) monitoring of cycle time by means of software and a hardware watchdog, (b) automatic testing of the watchdog, (c) bus systems with constant load, and (d) no processing of absolute time or date. Improvements in safety system software also included self-testing of the inputs of input modules and automatic read-back of the outputs from output modules.

b) Because digital safety system implementations were also accompanied with improvements in regulatory guidance, the issue of CCF was also a greater focus in safety systems implemented from the mid-1990s to the 2000s. For example, the preferred measure against CCF, especially in connection with design errors, was functional diversity, which involves ensuring that the safety I&C subsystems, while equipped with the same hardware and system software, execute different I&C functions for handling one and the same event. For example, a reactor trip due to a steam generator tube rupture event may be monitored by two I&C subsystems; one monitoring main steam activity and one monitoring steam generator and pressurizer levels. The assumption here is that the same hidden fault will not take effect simultaneously in two different functions at the same time, causing both of them to fail simultaneously. In the absence of a quantitative measure, BTP 7-19 and DI&C-ISG-04, both which are briefly discussed in Sect. 1, provide good additional guidance for addressing digital CCF.

c) Software V&V procedures, reviews, and audits are important parts of the effort to reduce the potential for CCF and to comply with NRC requirements. The review of software V&V procedures of the safety system implementations showed that there were general improvements in software V&V as the technology implementations also evolved. However, it is the authors' opinion that these improvements were more a result of updates and improvements in regulatory guidance (rather than a result of the technology evolutions), which was a result of updates and improvements in the standards endorsed by the regulatory guides. For example, the 2013 version of RG 1.168, "Verification, validation, reviews, and audits for digital computer software used in safety systems of NPPs," has undergone a significant update as a result of revisions of the endorsed standards in the 1997 version. (The latter version was used to guide V&V for the TXS that was reviewed for this report). Examples include the addition of security analysis and the recommended use of the software integrity system, as the previous version did not require the selection of an integrity level.

4.4 THE ADVANCED LOGIC SYSTEM (ALS) PLATFORM: REPRESENTATIVE SYSTEM BASED ON THE FPGA APPROACH

4.4.1 Architecture of the ALS

System Architecture

The ALS is a Westinghouse safety system platform known as "a logic-based platform that does not utilize a microprocessor or software for operation, but instead relies on simple hardware architecture. . . . It is a hardware-based architecture that uses a minimal set of hardware to implement a system with high reliability and integrity. The system incorporates self-test capability for detection and mitigation of the effects of failures within or external to the system"[20].

The ALS platform is based on the FPGA. The FPGA contains basic programmable logic components (e.g., negative AND [NAND] gate) and programmable interconnects. This allows the FPGA to be programmed and interconnected in various ways to perform various functions using the basic logic components. The logic components can also be combined into more complex structures to perform more complex functions, including math functions.

The ALS is designed around four primary board types, as follows [20]:

a) core logic board (CLB): the primary decision-making board containing the functional logic for the system; it provides data link interfaces to external systems.

b) input boards (IPBs): convert specific types of field signals into digital signals (e.g., thermocouples, resistance temperature detectors [RTDs], 0-5V, 4-20mA) and filters inputs.

c) output boards (OPBs) convert digital signals to specific types of field signals and provide interfaces to actuators, indicators, relays and other devices.

d) communication board (COM): provides standard, bidirectional datalink interfaces with other controllers.

An ALS system typically consists of a combination of ALS boards (b), (c) and (d), and application-specific ALS boards based on item (a). A generic ALS platform architecture is shown in Figure 8.

Since an FPGA is not software-based in the traditional sense, this report does not have a software architecture section as with the other previous platforms. Instead, the fundamental differences between a microprocessor-based architecture and an FPGA-based architecture are identified. The following comparisons have been excerpted from Ref. [21].

1. In an FPGA-based design such as the ALS platform, the programmed FPGA will be limited to combinatorial logic and finite state machine (FSM) designs. Each as-designed and as-tested FSM logic circuit should be deterministic. This means that each FSM in the FPGA should operate independently from the others using hardware digital logic resources dedicated to that FSM but not shared with any other FSM. Also, no FSM should support an undefined state, and for a given state, only one transition to a new state should occur per cycle. Finally, for each input event applicable to the current state, there should be only one associated transition to the next state. As an example of how a reactor trip circuit might work using FSM design, one FSM might periodically acquire the sensor input data. This sensor input data may be provided to a second FSM to perform the comparison of the sensor input against setpoint values. A third FSM may receive the trip/no-trip results of the comparisons and send the result to the final actuator. A fourth and independent FSM may monitor an equipment rack door latch and the bypass switch to determine an alarm status when the door is opened and the channel is not in bypass.

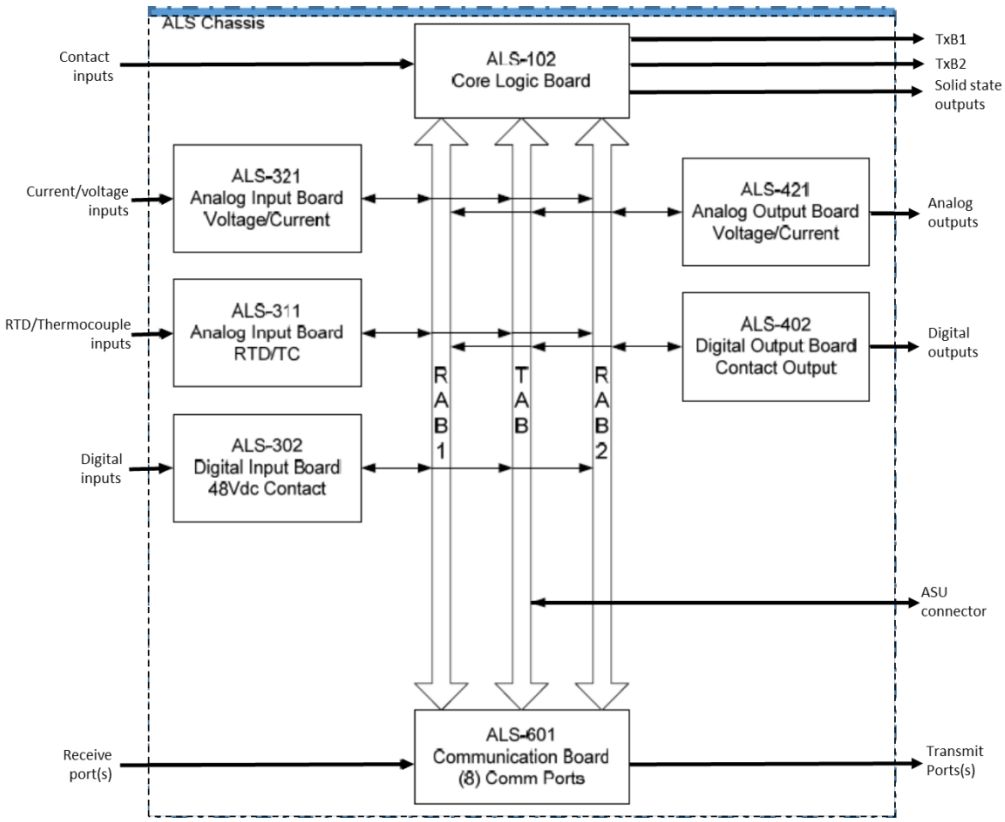

Figure 8. Advanced logic system platform architecture [21].

2. In contrast to the items above, operation of a software-based microprocessor system is fundamentally different. First, a microprocessor executes external instructions and should maintain an overall program flow control to perform the required functionality. A simple program flow control might cyclically repeat a single loop of instructions at a prescribed interval. More complicated program flow control might involve multiple tasks, task switching, and interrupts. In any case, program flow in all cases generally involves repetitive retrieval and storage from memory devices. Internal to the microprocessor, microcode uses dedicated registers to manipulate data and execute low-level operations. The memory and the circuitry within the microprocessor (registers, etc.) are shared resources for the overall software program. Resource sharing increases the potential for a latent error or another event to propagate, which can result in unpredictable behavior. In addition, an unplanned interruption of any portion of the software may cause the microprocessor or its program(s) to enter undefined states. By contrast, the FPGA circuit modules in the ALS platform always use dedicated resources and are designed and verified to preclude undefined states.

3. In microprocessor-based safety systems, online diagnostic programs are typically used to detect failures. The same is true for the ALS platform. The online diagnostics in this case are implemented by dedicated FSMs. In both microprocessor-based systems and FPGA-based systems, single event upsets (SEUs) and single event latch-ups (SELs) can corrupt a memory location and result in unpredictable behavior. To address this concern, microprocessor-based designs typically include a separate and distinct watchdog timer circuit. The watchdog timer is reset at a prescribed program control point before it times out to ensure that the system is functioning properly. This kind of watchdog timer is not applicable to FPGAs developed with constraints similar to the ALS platform

FPGAs. However, FPGA logic can include watchdog timer–like functionality to ensure other FPGA logic satisfies specified timing requirements. The ALS platform FPGAs include such logic.

4. For microprocessor-based systems, an additional operating system software layer and set of device drivers (typically commercial off-the-shelf or proprietary) may exist to support the application-specific software functions and processes. This additional software typically has very limited or no design disclosure documentation. Furthermore, the internal designs of commercial microprocessors are proprietary and lack design disclosure documentation. As reference [21] indicates, this lack of transparency into the design is undesirable from a safety evaluation standpoint because it restrains the ability to perform root-cause analysis and corrective actions as part of the equipment's life-cycle. In contrast to typical commercial software and microprocessor-based designs, an FPGA's internal design details are transparent and available for review and evaluation.

4.4.2 Communication in the ALS

The ALS platform limits the digital communications for safety signals to serial data transfers over the reliable ALS bus (RAB) within each safety division [21]. This is depicted in Figure 8. A separate bus, a test ALS bus (TAB), for each safety division is used for maintenance, diagnostics, and test data.

The ISG DI&C-ISG-04, Revision 1[22], establishes a means to ensure independence among redundant safety channels while permitting some degree of interconnection and shared resources among independent channels. The following brief descriptions of how the ALS platform implements communication are made with reference to Figure 8 and have been excerpted from Ref. 21.

Communication among Safety Divisions or with Safety Equipment
The ALS-601 communication board provides eight unidirectional interfaces that can be independently configured as transmit-only or receive-only. The FPGA device on the ALS-601 communication board will include separate logic resources to independently support the communication through each interface. The use of the ALS-601 communication board for interdivisional communications appears to be limited to a communication processor for sharing (i.e., transmitting and receiving) individual channel trip votes among redundant safety channels to support a coincidence voting application.

Communication with Non-Safety Equipment

- *Communication via TxB1 and TxB2 on the ALS-102 Core Logic Board*
  The ALS-102 core logic board provides two transmit-only interfaces: TxB1 and TxB2. The FPGA device on the ALS-102 core logic board will include the logic that performs safety functions and the logic that supports communication via TxB1 and TxB2. To use the ALS platform, application specifications are required for each system that enables either TxB1 or TxB2 because the programming of the ALS-102 core logic board's FPGA and the digital data communication content and format for TxB1 or TxB2 are application-specific.

- *Communication via Test ALS Bus (TAB) and Instrument Backplane with Each Circuit Board*
  Each ALS standardized circuit board shares a bidirectional interface over the TAB on the instrument backplane. The FPGA device on each ALS standardized circuit board will include the logic that performs safety functions and the logic that supports communication via the TAB.

4.4.3 Overview of ALS FPGA Development Process

The basic development life cycle of an FPGA-based system is similar to that of a software-based system. However, some of the specific details are different. As with software-based systems, the development process of the ALS platform basically consisted of five steps: requirements, design, implementation, integration, and validation.

- *Requirements Specifications*
  As with software, specifications must first establish the desired functions of the FPGA. As is typical with software-based systems, the ALS platform used natural language in its requirements specifications and then used a text-based high level language to specify the functions of the FPGA's circuitry. The high-level language used for the ALS development is hardware description language (HDL).

- *Design Phase*
  In this phase, a representation of each requirement was developed so that the implementation of the logic would meet the requirements. The typical procedure in this phase is to define candidate architectures and evaluate them in order to select the best possible design. However, this level of detail is assumed in this report rather than being specifically identified in the ALS platform documentation reviewed for this report. The design phase involved refining the architectural specifications into behavioral descriptions that describe the functionality of each module and their interactions. Behavioral simulation, also referred to as pre-synthesis simulation, uses standard simulation tools and is based solely on the graphical or textual description of the design requirements. The next step after the pre-synthesis is the synthesis, which is used to generate the gate-level representation of the register transfer level description contained in HDL code.

  As in software simulation, HDL simulation allows for modeling and testing for how a circuit would behave without making actual device interconnections. Thus, HDL simulation does not require an actual FPGA device. However, this also implies that the testing validates the designer's intent rather than an actual circuit [21]. As stated in Ref. 21, "HDL simulation and validation are independent of the underlying FPGA device technology. This independence is similar to portable standard language software that excludes all target and compiler specific directives, and similarly leads to greater portability of the HDL from one device to another. The ALS platform development includes use of HDL simulation and validation with formally established and configuration controlled test vectors to verify acceptable FPGA circuit design behavior. This HDL simulation and validation is integral to ALS platform FPGA program development plans."

- In the *implementation phase*, the ALS detailed development specifications were transformed into functioning logic. The implementation phase includes the generation, testing, and V&V of the ALS HDL code.

- In the *integration phase*, all the ALS lower level components were assembled into an overall structured component.

- The *validation phase* assesses all the ALS components to verify that the logic operates according to the ALS requirements.

4.4.4 Observations and Insights from the ALS Technology Implementation Review

Following are the observations and insights obtained from technology implementation review of the ALS platform:

a) The reviews did not show that CCFs are any less plausible for FPGA-based safety systems than for microprocessor-based safety systems. For both FPGA-based systems and microprocessor-based systems, it is difficult to prove adequate test coverage, and the method to ensure adequate quality of the product continues to be extensive documentation of the development life cycle based on current standards. In general, the perception that FPGAs are much less complex than microprocessors and are not software-based and therefore may have less potential for CCF is somewhat misleading. First, present-day FPGAs can be very complex. Second, the development steps are similar and also prone to similar errors. In effect, microprocessors are complex hardware devices *programmed* by software engineers using conventional software programming methods such as C, while FPGAs are complex hardware devices *programmed* by hardware engineers using hardware programming methods such as VHDL or Verilog. The significant difference is in the *method* of programming. Both are complex, programmed digital devices and are therefore likely to be prone to similar embedded, undetected errors leading to the potential for CCF.

b) Based on the reviews described above, current guidance aimed at reducing the potential for CCF as found in BTP 7-19 (Rev. 6) should continue to be relied upon. Operational experience could also be investigated in a future study to *support* current guidance. Operational experience alone cannot be used as proof of adequate design against CCF: the (safety) system may have been operating well for years, during which time the plant may even have undergone abnormal conditions while showing that it performed its safety function under those abnormal conditions. However, this does not necessarily demonstrate adequate functionality under all of the scenarios that may not have occurred during the plant's operation.

5. CONCLUSIONS

This report reviews digital safety system implementations with evolving technology and provides a qualitative assessment of the adequacy or otherwise of current regulatory guidance on CCF. The following issues are addressed:

- What is NRC's current position on CCF?
- Is the current position adequate given the evolutions in digital safety system implementations from the 1980s to date? (Where are the gaps in the current NRC position that are not addressing the move from the old single board computer technology to FPGA technology?)
- If the current guidance is not adequate, what should be done?

Based on these questions, the findings are summarized as follows:

1. Early microprocessor-based safety system implementations such as the Eagle-21 process protection system was designed as a modular *functional replacement for existing analog equipment*. Starting from the premise that analog systems were mature technologies and their review processes were stable, a strict adherence to digital functional replacement for existing analog equipment was seen as limiting the potential for digital CCF. This appears to be the baseline upon which subsequent guidelines such as BTP 7-19 and the DI&C ISG were developed. This is a reasonable baseline, and although it is not quantitative, the authors believe that the state of the art does not currently warrant using any quantitative approach.
2. Although early digital implementations were typically one-for-one replacements of the proven analog designs as exemplified by the Eagle 21, some advantages of digital technology (e.g., onboard diagnostics) were nevertheless also implemented. For example, the Eagle 21 implemented automatic surveillance testing (to reduce the time required to perform surveillance tests), self-calibration (to eliminate rack drifts and time consuming calibrations), and self-diagnostics (to reduce the time required for troubleshooting). The drawback of implementing these software enhancements was the need to assure deterministic software behavior in spite of the additional software overhead. In the Eagle 21, deterministic performance was implemented as follows;
    a. Use a modular approach in the software design, with all executable code contained in modules or subroutines.
    b. No interrupts are allowed.
    c. No re-entrance is allowed.
    d. Code format conforms to standards for both high-level and assembly language routines
    e. GO TO statements are not allowed.
    f. All modules are single task (no operating system or multi-tasking system).
    g. All modules are single entry, single task.
    h. Modules exit to points of call
    i. Each module has a design performance specification and a verification test specification. However, these implementations alone do not necessarily guarantee sufficient determinism. For example, with the added overhead of onboard diagnostics and surveillance software, each module, as well as each complete cycle, should also be guaranteed to complete in a pre-determined time.[g]

[g] It is possible that this was also implemented in the early digital software safety systems such as the Eagle 21. However, the authors were unable to ascertain this from the available documentation.

3. The software V&V and digital communication standards and guidelines available in this period were generally adhered to in the system development. Since then, there have been considerable improvements in these standards and guidelines (e.g., DI&C-ISG-02) which now address issues such as interdivisional communication. However, because the early digital safety system implementations tended to be one-for-one replacements of analog systems with no inter-divisional communication, etc., the early standards and guidelines were adequate for the period.

4. Evolution of safety system implementations simply made use of more sophisticated microprocessors and increased online self-testing and surveillance, as exemplified by the TXS. However, these systems (TXS) also made use of the improving guidance for digital safety system implementations (e.g., updated V&V standards) and improved on implementation of deterministic performance. For example, the digital system architecture of the TXS included procedures that improved determinism such as (a) monitoring of cycle time by means of software and a hardware watchdog, (b) automatic testing of the watchdog, , (c) bus systems with constant load, and (d) no processing of absolute time or date. Improvements in safety system software also included self-testing of the inputs from the input modules and automatic readback of the outputs from the output modules.

5. Because digital safety system implementations were also accompanied with improvements in regulatory guidance, the issue of CCF was also a greater focus in safety systems implemented beginning in the mid-1990s to the 2000s. For example, the preferred measure against CCF, especially in connection with design errors, was functional diversity. This involves ensuring that the safety I&C subsystems, while equipped with the same hardware and system software, execute different I&C functions for handling one and the same event. For example, a reactor trip resulting from a steam generator tube rupture event may be monitored by two I&C subsystems: one monitoring main steam activity, and one monitoring steam generator level and pressurizer level. The assumption here is that the same hidden fault will not take effect simultaneously in two different functions at the same time, causing both of them to fail simultaneously. In the absence of a quantitative measure, BTP 7-19 and DI&C-ISG-04 (both briefly discussed in Sect. 1) provide good additional guidance for addressing digital CCF.

6. Software V&V procedures, reviews, and audits are important parts of the effort to reduce the potential for CCF and to comply with NRC requirements. The review of software V&V procedures for safety system implementations showed that there were general improvements in software V&V as the technology implementations also evolved. However, these improvements resulted from updates and improvements in regulatory guidance rather than from technology evolutions. The revisions to regulatory guidance resulted from updates and improvements in the standards endorsed by the regulatory guides. For example, the 2013 version of RG 1.168, “Verification, validation, reviews, and audits for digital computer software used in safety systems of NPPs,” has undergone a significant update as a result of revisions of the endorsed standards in the 1997 version. (The latter version was used to guide V&V for the TXS reviewed for this report). Examples include the addition of a security analysis and the recommended use of the software integrity system, as the previous version did not require the selection of an integrity level.

7. With regard to the migration to FPGA technology, the reviews did not show that common cause failures are any less plausible for FPGA-based safety systems than for microprocessor-based safety systems. For both FPGA-based systems and microprocessor-based systems, it is difficult to prove adequate test coverage, and the method of ensuring adequate quality of the product continues to be extensive documentation of the development process, qualification, testing, guidelines on how to address computer communication issues (DI&C-ISG-04), guidelines on how to address diversity and defense-in-depth issues (BTP 7-19 Rev 6), etc. In the absence of quantitative methodologies (which the present state-of-the-art do not support), the current standards and guidelines provide very good guidance to assure quality and reduce the potential for CCF in DI&C for NPPs and should continue to be applied.

8. As a result of the above reviews, it is the authors' conclusion that current guidance aimed at reducing the potential for CCF as found in BTP 7-19 (Rev. 6) and DI&C-ISG-04 should continue to be relied upon. Operational experience could also be investigated in a future study to *support* current guidance. Operational experience alone cannot be used as proof of adequate design against CCF: the (safety) system may have been operating well for years during which the plant may even have undergone abnormal conditions showing that it performed its safety function under those abnormal conditions. However, that does not necessarily demonstrate adequate functionality under all scenarios that may not have occurred during the plant's operation.

6. REFERENCES

1. SECY-93-087, "Policy, Technical, and Licensing Issues Pertaining to Evolutionary and Advanced Light-Water Reactor (ALWR) Designs," July 21, 1993 (ADAMS Accession No. ML003708056).

2. BTP 7-19, Rev. 6, "Guidance for Evaluation of D3 in Digital Computer-Based Instrumentation and Control Systems," US NRC, July 2012. (ADAMS Accession No. ML110550791)

3. DI&C-ISG-02, "Task Working Group #2: Diversity and Defense-in-Depth Issues Interim Staff Guidance," Revision 2, June 2009.

4. DI&C-ISG-04, "Task Working Group #4: Highly-Integrated Control Rooms—Communications Issues," Interim Staff Guidance Revision 1, March 2009.

5. US Nuclear Regulatory Commission (NRC), "Digital Instrumentation & Control Training Module 2.0 - Architecture Overview," NRC Training Center, Rev. 20070905.

6. Institute of Electrical and Electronics Engineers (IEEE), "IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations," IEEE Std. 7-4.3.2-2010, Piscataway, New Jersey, 2010.

7. Electric Power Research Institute, *Guideline on Licensing Digital Upgrades* – TR-102348 Revision 1 - NEI 01-01: A Revision of Electric Power Research Institute (EPRI) TR-102348 to Reflect Changes to the 10 CFR 50.59 Rule, EPRI TR-102348, Rev. 1, March 2002.

8. H. M. Hashemian, in *Nuclear Power Plant Instrumentation and Control, Nuclear Power - Control, Reliability and Human Factors*, 2011, Dr. Pavel Tsvetkov (Ed.), ISBN: 978-953-307-599-0, InTech, <Available from: http://www.intechopen.com/books/nuclear-power-control-reliability-and-human-factors/nuclear-power-plantinstrumentation-and-control >.

9. Implementing Digital Instrumentation and Control Systems in the Modernization of Nuclear Power Plants, IAEA Nuclear Energy Series No. NP-T-1.4, 2009.

10. A. Avritzer and E. J. Weyuker, "Monitoring Smoothly Degrading Systems for Increased Dependability," Empirical Software Engineering, 2(1), March 1997.

11. "Protecting against Common Cause Failures in Digital I&C Systems of Nuclear Power Plants," IAEA Nuclear Energy Series No. NP-T-1.5, 2009.

12. L. E. Erin, "EAGLE-21 Microprocessor-Based Process Protection System," Westinghouse Topical Report, January 1987.

13. Topical Report EMF-2110(NP), Revision 1, "TELEPERM XS: a Digital Reactor Protection System" Project no. 702, Safety Evaluation by the Office of Nuclear Reactor Regulation Siemens Power Corporation.

14. TELEPERM XS System Overview, published 2012, AREVA NP GmbH.

15. "Software Program Manual for TELEPERM XS™ Safety Systems," Topical Report, ANP-10272, AREVA NP, Inc., 2006.

16. IEEE Std. 730-2002, “Standard for Software Quality Assurance Plans,” Institute of Electrical and Electronics Engineers, Inc., 3 Park Avenue, New York, NY 10016-5997, USA.

17. IEEE Std. 1028-1997, “Standard for Software Reviews,” Institute of Electrical and Electronics Engineers, Inc., 3 Park Avenue, New York, NY 10016-5997, USA.

18. Regulatory Guide 1.168, “Verification, Validation, Reviews, and Audits for Digital Computer Software Used in Safety Systems of Nuclear Power Plants,” U.S. Nuclear Regulatory Commission, September 1997.

19. IEEE Std. 1012-1998, “IEEE Standard for Software Verification and Validation,” Institute of Electrical and Electronics Engineers, Inc., 3 Park Avenue, New York, NY 10016-5997, USA.

20. “Advanced Logic System Platform,” http://www.westinghousenuclear.com, Westinghouse Electric Company LLC, May 2013.

21. Safety Evaluation for Topical Report 6002-00301, “Advanced Logic System Topical Report,” http://pbadupws.nrc.gov/docs/ML1321/ML13218A979.pdf, September 2013, U.S. Nuclear Regulatory Commission.

22. Interim Staff Guidance DI&C-ISG-04 (Revision 1), “Task Working Group #4: Highly-Integrated Control Rooms − Communications Issues,” March 2009.