UNIVERSIDAD DE EXTREMADURA

TESIS DOCTORAL

APPLICABILITY OF FIELD PROGRAMMABLE GATE ARRAYS IN INSTRUMENTATION AND CONTROL SYSTEMS IN NUCLEAR POWER PLANTS

APLICABILIDAD DE LAS MATRICES DE PUERTAS PROGRAMABLES EN CAMPO EN LOS SISTEMAS DE INSTRUMENTACIÓN Y CONTROL EN CENTRALES NUCLEARES

VÍCTOR JOSÉ ROJAS MORENO

Departamento de Expresión Gráfica

Conformidad de los Directores:

Fdo.: Mª Teresa Miranda García-Cuevas **Fdo.: Irene Montero Puertas**

2014

A mi familia, en especial a mis dos princesas, Mencía y Coral.

INDEX

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK.

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK.

SUMMARY

Early nuclear power plant instrumentation and control systems employed conventional hardware technologies such as relays, analog electronic circuit boards, and discrete digital electronics. These had a long service life, but they have now become obsolete. Also, they were limited in their functional capabilities, and the safety justification for these systems was based largely on review of the hardware design and functional testing.

When faced with obsolescence of the conventional hardware, many operating plants and new plant designers moved to microprocessor-based systems. These are much more capable than conventional hardware and, most important, were widely used and available from multiple vendors.

But they have proved to have a short lifetime, becoming obsolete much more quickly than the conventional hardware did. Also, the safety justification proved to be much more difficult, primarily due to the added complexity associated with software. Additionally, microprocessor-based systems execute tasks in a sequential way, being slower than their predecessors. This is a big handicap when talking about high speed (low latency) requirements in certain control systems and in protection systems. And moreover, this kind of technology is very prone to cyberattacks.

The objective of this thesis is to analyze the suitability and applicability of field programmable gate array technologies (FPGAs) for instrumentation and control systems in nuclear power plants, mainly for safety and protection systems.

The study has been structured in the following way. First, a state of the art study has been carried out, identifying advantages and limitations, applications in other high tech industries, as well as the use in nuclear industry up to the present time. Afterwards, guidelines have been developed to have into account during planning and design stages of projects using this type of technology. Later, a complete set of design guidelines has been issued for the selected option. These results have been used in the preparation of a specification for a diesel load sequencer based on the use of FPGAs. Finally, the thesis is completed with main conclusions and future research proposals.

Main conclusion that can be extracted from this thesis is that FPGA is the technology of the future for instrumentation and control systems in nuclear power plants, both for upgrade projects as well as new systems for future plants. FPGAs have a perfect balance between applicable requirements and with the exact amount of complexity, allowing a relative simple development and licensing process, and guarantying at the same time resilience against obsolescence and cyber threats.

Key words:

FPGA, field programmable gate array, instrumentation and control, nuclear power plant.

THIS PAGE INTENTIONALLY LEFT BLANK.

RESUMEN

Los primeros sistemas de instrumentación y control de las centrales nucleares empleaban tecnologías convencionales como relés, tarjetas analógicas y electrónica digital discreta. Estas tecnologías tenían una larga vida de servicio, pero han empezado a convertirse en obsoletas. Además, estaban limitadas en sus capacidades funcionales, y la justificación de seguridad para estos sistemas se basaba fundamentalmente en la revisión del diseño y las pruebas funcionales.

Cuando se enfrentaron al hecho de la obsolescencia de los equipos convencionales, muchas plantas en operación y los diseñadores de nuevas plantas optaron por los sistemas basados en microprocesador. Estos sistemas tienen más capacidades que los convencionales y, mucho más importante, eran ampliamente utilizados y estaban disponibles por parte de numerosos proveedores.

Por otro lado, han demostrado tener un ciclo de vida corto, convirtiéndose en obsoletos más rápido que los equipos convencionales. Además, la justificación de seguridad demostró ser más difícil, principalmente debido a la complejidad añadida asociada al software. Adicionalmente, los sistemas basados en microprocesadores ejecutan las tareas de forma secuencial, lo que los hace más lentos que sus predecesores. Esto supone un obstáculo cuando se trata de sistemas de control con requisitos de alta velocidad (baja latencia), así como en sistemas de protección. Y más aún, este tipo de tecnología es muy propensa a ciberataques.

La presente tesis tiene por objetivo analizar la idoneidad y aplicabilidad de las tecnologías basadas en matrices de puertas lógicas programables en campo (FPGAs, por su acrónimo en inglés) en los sistemas de instrumentación y control de las centrales nucleares de generación eléctrica, principalmente en aquellos de seguridad y protección de las mismas.

El trabajo se ha estructurado de la siguiente manera. En primer lugar, se realiza un estudio del estado del arte asociado a estas tecnologías, identificando sus ventajas e inconvenientes, aplicaciones en otro tipo de industrias tecnológicamente avanzadas, así como su empleo hasta el momento en la industria nuclear. Posteriormente se desarrollan las directrices que deben tenerse presentes en el proceso de diseño y planificación de proyectos que empleen este tipo de tecnologías. A continuación se desarrollan unas completas guías de diseño para la solución elegida. Los resultados obtenidos han permitido desarrollar, como caso práctico, la especificación para el suministro de un secuenciador de cargas de salvaguardias basado en el uso de FPGAs. La tesis se completa con las principales conclusiones que han podido extraerse de este trabajo así como con una propuesta de líneas de investigación futuras.

La conclusión principal a la que se llega es que las FPGAs son la tecnología del futuro para los sistemas de instrumentación y control de centrales nucleares, tanto en la modernización de plantas actuales como para los sistemas que se desarrollen para plantas futuras. Las FPGAs armonizan de manera especialmente adecuada los requisitos aplicables a estos sistemas con

el grado justo de complejidad, lo que permite un desarrollo y licenciamiento relativamente sencillo, al tiempo que garantizan una adecuada fiabilidad y resistencia frente a la obsolescencia y las ciberamenazas.

Palabras clave:

FPGA, matriz de puertas programable en campo, instrumentación y control, central nuclear.

BACKGROUND

Field-programmable gate arrays (FPGAs) are gaining increased attention worldwide for application in nuclear power plant instrumentation and control (I&C) systems, particularly for safety applications. Utilities see potential advantages of this technology as compared to the more common microprocessor-based digital I&C system implementations, because FPGA-based systems can be made simpler, more testable, less reliant on complex software (e.g., real time operating system or RTOS), and easier to qualify for safety-related applications. In addition, the use of FPGAs may provide a significant advantage in terms of hardware independence of the design and capability for cost-effective, long-term support over extended plant lifetimes. Finally, FPGAs have also significant advantages over cybersecurity issues. FPGA-based systems and equipment are beginning to appear in new plant I&C designs, as well as in retrofits for operating plants.

Conventional relay and analog electronics technologies used in the original I&C systems of most operating plants are increasingly becoming obsolete, apart from aging related issues. Utilities have been replacing these older systems with more modern equipment. Most replacements have used microprocessor-based equipment such as microcontrollers, programmable logic controllers (PLCs) or distributed control systems (DCSs). Several suppliers have received generic approval of their digital systems and components from the United States Nuclear Regulatory Commission (US NRC) and many microprocessor-based systems are in operation in plants worldwide. New plant designs have been based largely on microprocessor-based I&C systems.

However, experience has shown that for safety applications, gaining regulatory approval for microprocessor-based systems can be difficult and expensive. The systems are complex, involving large amounts of software in the form of operating systems and other platform software along with the custom software that performs the application. Also, microprocessors and the associated software tools tend to become obsolete much more quickly than the analog equipment they have replaced. Additionally, cybersecurity has become another major issue to be considered, not only in the design and implementation phase, but along the whole life cycle of the system. Therefore, plants are faced with the need for repeat replacement projects, which again can be costly and time-consuming. As a result, there has been increasing interest in exploring other electronic technologies for use in nuclear I&C systems to alleviate some of these difficulties.

Application-specific integrated circuits (ASICs) are one alternative technology that has been considered for nuclear plant I&C applications. A joint project involving the Westinghouse Owners Group, Electric Power Research Institute (EPRI), and Westinghouse Electric Company developed ASIC-based replacements for circuit boards in Westinghouse protection systems [1]. The equipment was reviewed by NRC and received a favourable safety evaluation in 2001 [2]. It has been installed in South Texas Project (largest installed population), Beaver Valley Unit 2,

Braidwood, Wolf Creek, VC Summer and Vogtle nuclear power plants (NPP) in the US. Nevertheless, nowadays the last three plants have no ASIC-based printed circuit board (PCB) installed, and Beaver Valley and Braidwood have only one application.

Programmable logic devices (PLDs[1]) are another alternative for use in nuclear applications, having been widely used in many industrial, military and aerospace applications for years. Figure 1 illustrates where PLDs fit into the overall landscape of electronic hardware technologies, including conventional technologies, ASICs and microprocessors.

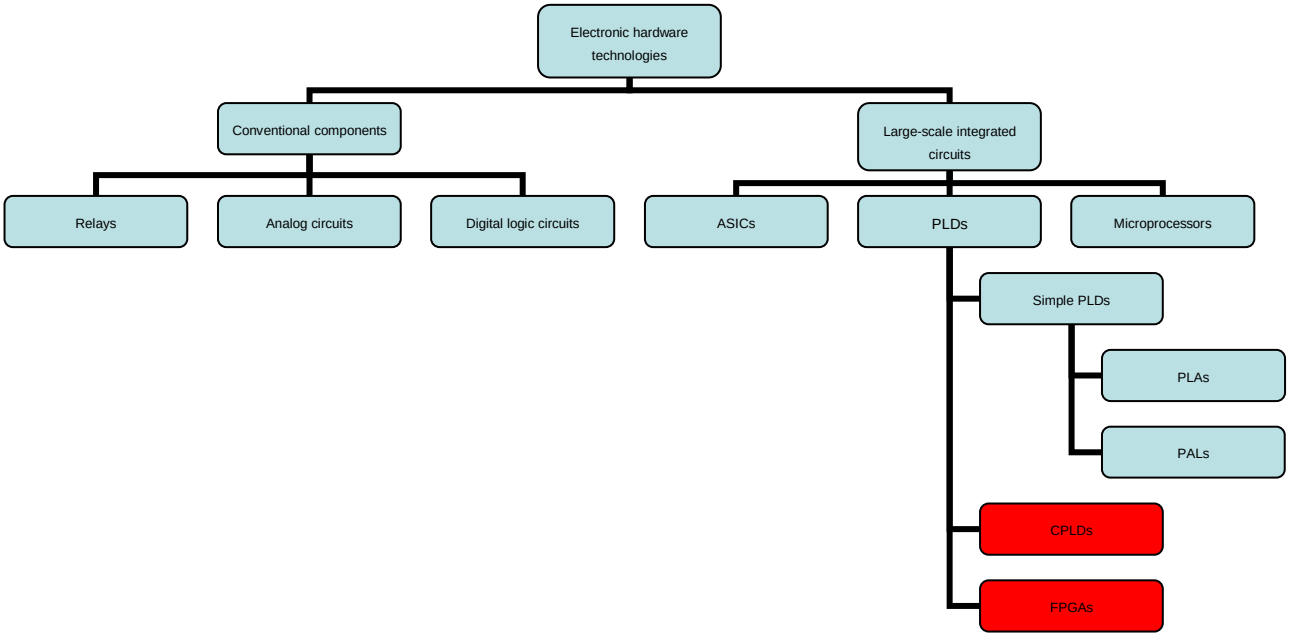

Figure 1 – Electronic hardware technologies

ASICs are custom-designed and fabricated at the integrated circuit foundry for a specific application. This can be very costly, and as a result it has generally been found that ASICs are cost-effective only when a large number of copies of the circuit is going to be deployed, but this is not the case for nuclear industry.

On the other hand, PLDs contain arrays of logic elements that can be interconnected by the user to perform the functions required for a particular application. The first to become available were simple PLDs, which include programmable logic arrays (PLAs) and programmable array logics (PALs). PLAs have been used in nuclear power plant I&C systems in the past. One US plant has used PLAs extensively for component control logic in both safety and non-safety systems for over 20 years, and they are still operating.

Complex PLDs (CPLDs) evolved from PALs (they essentially combine multiple PALs on a single chip, with the ability to interconnect them to perform more complex functions).

[1] The term “programmable logic device” or PLD is sometimes used to refer only to simple PLDs. Nevertheless, it will be used in this document for any programmable hardware logic device, both simple and complex PLDs and FPGAs.

FPGAs also can be considered PLDs, but they have a different internal architecture from the other types of PLDs shown in Figure 1. FPGAs use “gate arrays” as opposed to “logic arrays” among other differences. FPGAs available today contain millions of gates that can be interconnected to perform functions with a wide range of complexity. FPGAs are often used to develop ASICs, providing a convenient way to develop and refine an application using a reprogrammable device, leading to a prototype that can be fully tested prior to committing to fabrication of the ASIC at the foundry. In fact, the ASIC solution discussed above that was approved by NRC was prototyped using an FPGA [3].

FPGAs are used today in consumer electronics and in a large number of industrial control applications. Having started as relatively simple and inexpensive logic devices, FPGAs have quickly evolved into inexpensive substitutes for ASICs. Because of their reconfigurable capability, FPGAs are widely used in mass-market applications. However, FPGAs are also used to perform functions that are safety critical and require high reliability, like in surgery, automobiles, aircraft control and assistance and mission-critical applications in the aerospace industry.

FPGAs have been applied in nuclear plant I&C systems as well, including safety and non-safety systems. Safety applications include Reactor Trip Systems and Engineered Safety Features Actuation Systems in Ukraine and Bulgaria, and neutron monitoring systems in Japan. In the US, the first safety-related application of FPGAs occurred in 2009 in the Wolf Creek Main Steam and Feedwater Isolation System. FPGAs are also being applied in new plant designs, but mainly as basis for the Diverse Actuation System required by regulation in the case the main safety system is microprocessor-based.

Previous experience with CPLD-based equipment can also be cited. Westinghouse Electric Company developed, in the frame of a PWROG project, new design replacement PCBs for its Solid State Protection System (SSPS). This new design boards make use of CPLD technology from Xilinx (flash technology) and are aimed to addressed aging and Motorola high threshold logic (MHTL) technology obsolescence, which was the core technology of old boards. Main requirements for new design were elimination of single point vulnerabilities (SPVs), by introducing the necessary redundancy in the design, and incorporation of complete coverage self-testing (for which a second test CPLD is used). Nowadays, numerous plants worldwide have installed these new-design boards, including Almaraz and Krsko in Europe and Comanche Peak, Braidwood, Byron, Salem, Shearon Harris and Vogtle in the US.

Another example from Westinghouse, but in this case for a non-safety system, is the new Digital Rod Position Indication (DRPI) Advanced Display System (DADS). This new development is aimed to replace older Analog Rod Position Indication System (ARPI) or DRPI displays, and makes use of a COTS product from National Instrument (CompactRIO^{TM}).

Table 1 – Brief comparison between instrumentation and control technologies

Technology	***Product Life Cycle***	***Performance (Drift)***	***Testability (Verification)***
Analog-based	Parts obsolescence	Problematic	Good
CPU-based	Short life cycle	Good	Complete verification is difficult.
FPGA-based	Good	Good	Good

Nowadays, main FPGA manufacturers include Xilinx, Altera, Actel and Lattice. Figure 2 shows a timetable of main FPGAs manufactures.

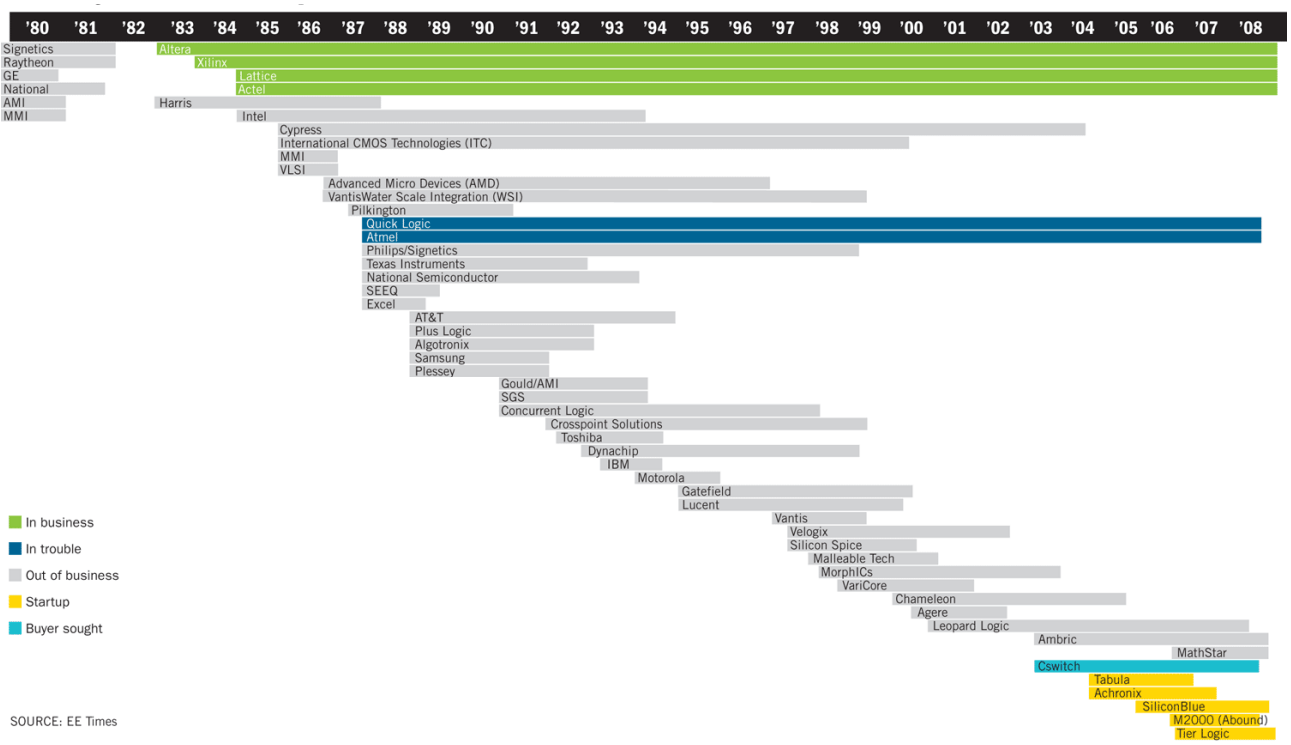

Figure 2 – History of programmable logic devices

Source: EETimes Europe, August 2009, pages 6-7

OBJECTIVES

Main objective of this thesis includes the following:

- Study of the state of the art on programmable logic hardware technologies, mainly focuses on field programmable gate arrays, looking at existing alternatives, their characteristics and their advantages and disadvantages.
- Analysis of advantages and limitations on the use of field programmable gate arrays in instrumentation and control systems in nuclear power plants.
- Study of the experience on the use of programmable logic hardware in high-tech industrial sectors.
- Study of the experience on the use of programmable logic hardware in the nuclear industry worldwide.
- Analysis of the current regulatory framework on the use of field programmable gate arrays in safety systems in the nuclear generation industry.
- Development of a full set of recommendations and good practices for planning and designing modifications using field programmable gate arrays in nuclear power plants.
- Development of design guidelines for field programmable gate array based solutions for nuclear power plants.
- Practical application of the guidelines in a diesels load sequencer upgrading specification for a generic nuclear power plant.

THIS PAGE INTENTIONALLY LEFT BLANK.

1 PRIMER ON FIELD PROGRAMMABLE LOGIC DEVICES

1.1 INTRODUTION TO FIELD PROGRAMMABLE LOGIC

1.1.1 FIELD PROGRAMMABLE GATE ARRAY

1.1.1.1 FIELD PROGRAMABLE GATE ARRAY ARCHITECTURE

Although FPGAs from different vendors and product lines differ in their detailed designs, they generally share a common basic architecture illustrated in Figure 3. This includes:

- A set of configurable logic blocks (CLBs)

  A CLB can be configured to implement any logic functions (AND, OR, XOR, NOT, etc.). To this end, each CLB features N Boolean inputs and M Boolean outputs. Each CLB can be configured to implement an N-to-M Boolean function using simple logic gates, or may be configured to use a look-up table (LUT) to implement the logic function. Multiple CLBs can also be interconnected to generate more complex functions. The output of each CLB includes a flip-flop for synchronizing the data flow within the FPGA.

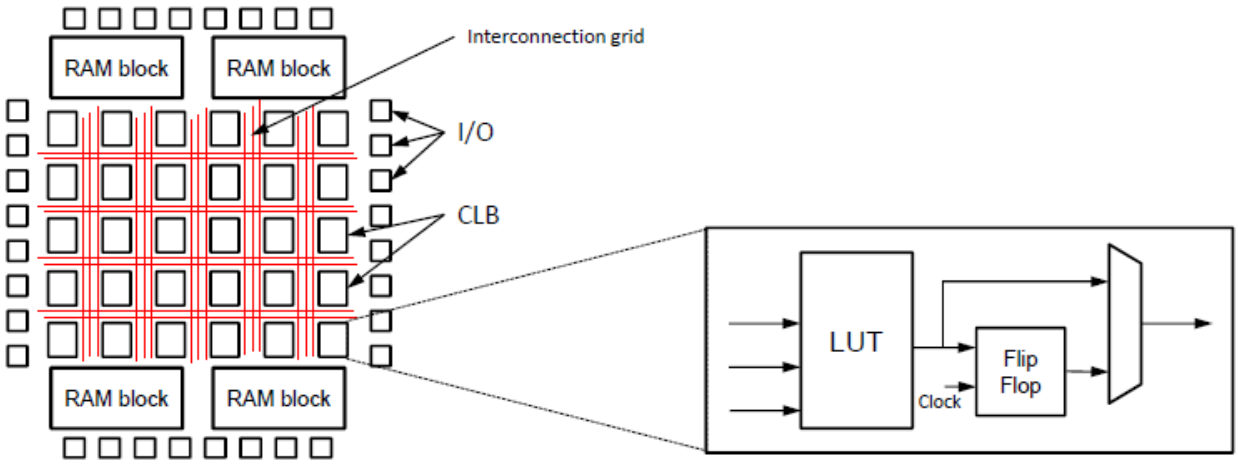

Figure 3 - Typical field programmable gate array architecture

- A set of programmable Input/Output (I/O) blocks

  These are the electrical interfaces between the low voltage, low current signals within the FPGA, and the higher voltages and currents required by the external electronic components connected to the FPGA. Each I/O block can be configured as an input or an output and is connected to one or more CLBs. Some I/O blocks can perform analog-to-digital conversion.

- An internal interconnection grid

  This is a set of unconnected horizontal and vertical wires. It is possible to create a contact at each intersection. The contacts constitute a connection pattern that links a CLB output to one or more CLB inputs. It also links the FPGA I/O blocks to specific CLBs inputs and output.

- Application data memory

  Almost all FPGAs contain additional memory dedicated to application data, compensating for the limited memory capacity in the CLBs. Memory blocks in most FPGAs are based on static random access memory (SRAM). However, non-volatile flash memory blocks are used in FPGAs requiring instant-on and greater resistance to single-event upsets (SEUs).

Some FPGAs can include other elements, like hard-wired microprocessors, which are linked to the CLBs through the interconnection grid.

1.1.1.2 FIELD PROGRAMMABLE GATE ARRAY TECHNOLOGIES

FPGA technologies distinction can be made in the way they store or memorize the configuration of the interconnection grid and the configurations of the CLBs and I/O blocks. Thus, we can differentiate between:

- SRAM – static random access memory

  SRAM is re-writable, which means that the implemented functionality can be modified without physically replacing the FPGA component. Because it is volatile memory, with SRAM the programming is not retained by the circuit on a loss of power. Also, a power glitch may alter the FPGA programming (interconnection grid, CLBs and/or I/O blocks). As a result, measures may need to be taken to protect against power glitches and other SEUs when SRAM is used.

- Flash and EPROM – erasable programmable read-only memory

  EPROM and flash technologies are re-writable and non-volatile. In both cases the FPGA programming is unaltered by power glitches. Flash is a modern derivative of the older, slower EPROM technology.

- Antifuse

  This technology is non-rewritable and non-volatile. A contact between two wires of the interconnection grid is created by sending a high current through the wires. Rather than breaking a connection or fuse to form the current flow, the connection is created between two logic blocks by means of heated nickel-alloy links, thus the name “anti-fuse.” The same process is applied to configure the CLBs and the I/O blocks. If the programming of the FPGA needs to be modified, it will be necessary to physically replace the component.

Table 2 – Comparison between field programmable gate array technologies

DESCRIPTION	***SRAM***	***FLASH***	***ANTIFUSE***
Typical use-case	Commercial, high volume	Military&Space	Military&Space
Technology	Standard CMOS (typical deep-submicron)	CMOS with flash technology	Special antifuse technology many additional process steps
Prone to SEU	Very sensitive to SEU. SRAM devices prone to neutron induced configuration errors	Insensitive to SEU. Immune to neutrons	Immune to SEU. Immune to neutrons
Prone to SEL	Very susceptible	Less susceptible	Less susceptible
Configuration integrity (bit flipping)	Susceptible	Leakage issue	Fuse defects, electron migration, weak oxide
Configuration retention time	Until loss of power (20-50 years)	20-50 years (temperature <70℃)	Indefinite
Device configuration lock (intruders capability to change content)	Transferred at startup (possible to modify setup)	FlashLock (impossible)	FuseLock (impossible)
Security (read-back of content)	Transferred at startup (none)	FlashLock (impossible)	FuseLock (impossible)
Development friendliness	Fast and easy	Optimal	Slow and difficult

Independent and comprehensive reports from industry neutron-effects experts iRoC Technologies, determines that SRAM-based devices are susceptible to functional failure when exposed to neutron radiation. Failure rates are significant even when exposed to the naturally-occurring background neutron radiation present at ground level. On the contrary, the fuse and flash based FPGAs are immune to the effects of neutron radiation.

Radiation-hard FPGA options are available but would not be typically required for target applications in the case of nuclear industry, as systems are usually installed in mild environments. Nevertheless, for local instrumentation and controls (e.g., field transmitters), it

could be beneficial to used radiation-hard or radiation-tolerant FPGAs already available in the market, or design appropriate radiation shielding at the component level.

1.1.2 COMPLEX PROGRAMMABLE LOGIC DEVICE

1.1.2.1 COMPLEX PROGRAMMABLE LOGIC DEVICE ARCHITECTURE

CPLDs from different vendors and product lines vary in their detailed designs. Nevertheless, they all share a common basic architecture illustrated in Figure 4:

- An “AND-plane” and an “OR-plane”

  A “plane” is composed of crossing vertical and horizontal wires. It is configured by creating a contact at appropriate intersections.

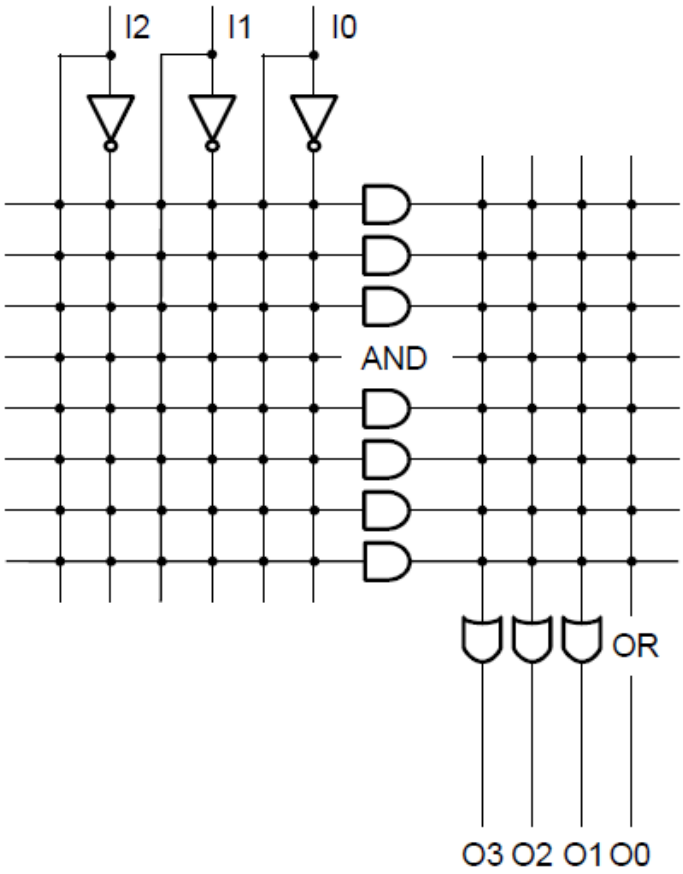

Figure 4 – Typical Complex Programmable Logic Device configurable logic block

- A set of programmable I/O blocks

  CPLD I/O blocks serve as electrical interfaces between the CPLD CLBs and the outside. However, one significant difference is that the synchronization flip-flops are

integrated to the I/O blocks and not to the CLBs. This forces the outputs of many CLBs to be connected to an I/O block and its corresponding pins. This may limit the possible number of sequential steps in the logic processing.

- A global routing plane

  This is a set of unconnected horizontal and vertical wires. It is possible to create a contact at each intersection. The contacts constitute a connection pattern that links together all the CLBs − see Figure 5.

Such architecture can very efficiently implement Boolean logic equations.

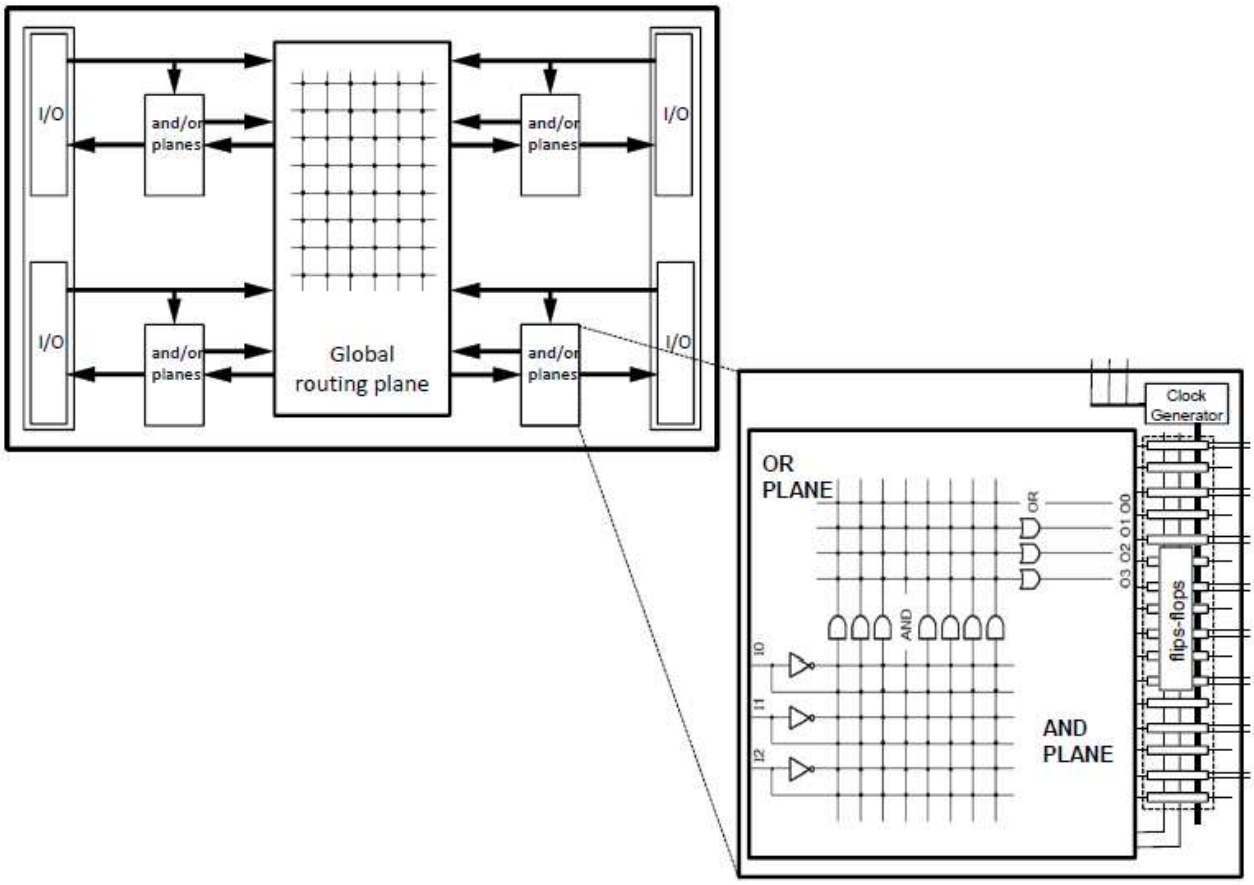

Figure 5 – Typical Complex Programmable Logic Device architecture

1.1.2.2 COMPLEX PROGRAMMABLE LOGIC DEVICE TECHNOLOGIES

For CPLDs, the memory elements needed to hold the contact configuration of the global routing plane, the configuration of the CLBs and the I/O blocks, employ similar technologies as for FPGAs:

- Antifuse

  A contact between two wires of the interconnection grid is created by sending a high current through the wires. The same process is applied to configure the CLBs and the I/O blocks. This is a similar method to that used in FPGAs but applied to a different architecture and device. As in the one-time-programmable FPGA, if the configuration of the CPLD needs to be modified, it will be necessary to physically replace the component. Traditionally a large majority of CPLDs have been antifuse-based.

- Reconfigurable

  Some modern CPLDs can be reconfigured (they are sometimes called Electrically Programmable Logic Devices or EPLDs). In these circuits, fuse arrays are replaced by transistors arrays, each transistor being driven by internal, non-volatile EPROM elements.

1.1.3 COMPARISON BETWEEN COMPLEX PROGRAMMABLE LOGIC DEVICES AND FIELD PROGRAMMABLE GATE ARRAYS

FPGAs have large numbers of CLBs that can be interconnected, but each CLB can implement only simple functions. There are fewer CLBs in a CPLD, but each can implement more complex combinatorial functions.

In a CPLD it is often possible to implement inter-dependent functions in a single CLB. This makes maximum clock frequency more easily predictable. For FPGAs, the determination of the maximum clock speed often depends on the application and design details, and in that sense is less predictable.

Because of their higher logic-to-interconnect ratio, CPLDs generally can yield a faster solution for simple applications. However, FPGAs demonstrate much greater flexibility and larger design capacity. Also, as noted earlier, FPGAs often have embedded hard-wired cores to perform complex functions, including microprocessors, whereas CPLDs typically do not.

Table 3 – Comparison of key attributes of Complex Programmable Logic Devices and Field Programmable Gate Arrays

Attribute	***CPLD***	***FPGA***
Configurable logic block	Logic array – like a PAL	Gate array
Density	<500K gates	>500K gates
Speed	Fast, predictable	Application and design dependent
Interconnect	Crossbar	Routing
Power consumption	High	Medium to low

Table 4 – Strengths and weaknesses of Complex Programmable Logic Devices and Field Programmable Gate Arrays

	CPLD	***FPGA***
Strengths	Maximum frequency can be determined precisely at beginning of design. Better suited for complex logic decoding due to more capable CLBs. Well adapted to address decoding and state machine designs. Simpler programming. Lower price.	Large choice of interconnect technologies, capacities and architectures. Technology widely used, not a technological dead end. Verification tools are continually being improved (but the number of available independent tools may be limited). Some vendors are targeting critical industrial, military and/or aerospace applications (components are designed for harsh environment conditions and have long commercial lifetimes).
Weaknesses	Design is much more constrained, less flexible – too limiting for some applications. High power consumption. Trend seems to be more toward FPGAs, so some risk of earlier obsolescence.	Programming and verification are more complex. Higher price. Rapid evolution of the technology and associated tools.

1.1.4 OTHER CONSIDERATIONS ON FIELD PROGRAMMABLE GATE ARRAYS

1.1.4.1 CONFIGURATION LOGIC BLOCK TYPES

CLBs can have different structures, but there are mainly two different types:

- Look-Up Tables

  A CLB can be based on LUTs. A LUT is a logic element that can implement any simple logic function. It is constructed out of a small RAM and a multiplexer.

  For example, to produce the logic function *y=a&b|c* ((aANDb)ORc), where *a*, *b* and *c* are the three inputs and *y* is the single output, the truth table of *a&b|c* is implemented in an 8 bit RAM. The output bit is selected by an 8-to-1 multiplexer. This is illustrated in Figure 6.

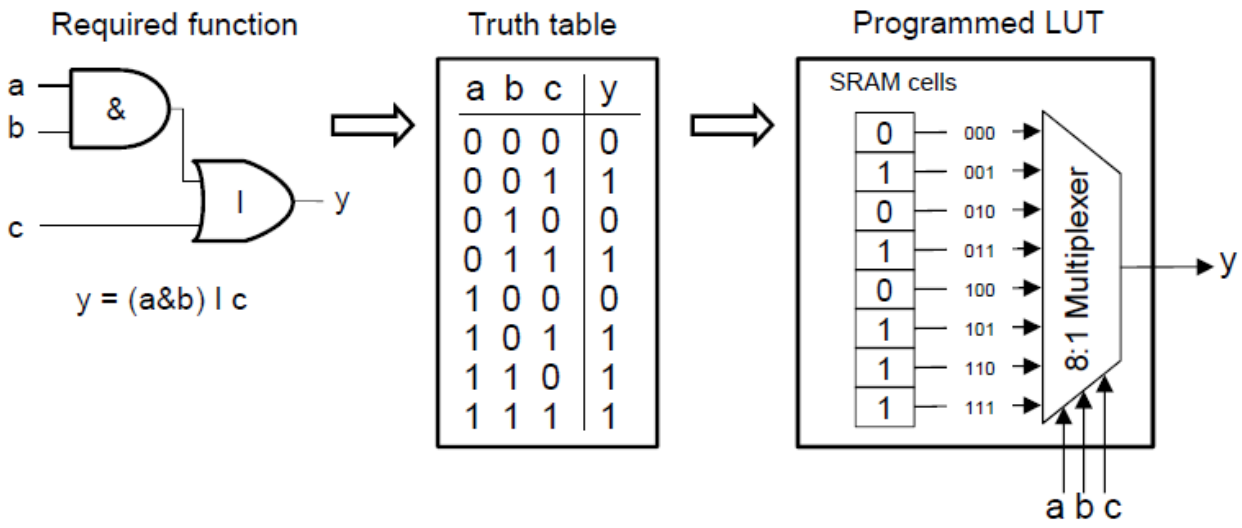

Figure 6 – Look-up table based configurable logic block

- Multiplexers

  The second type of CLB is purely multiplexer-based. It is similar to a tree of basic (2:1) multiplexers. Each multiplexer performs an operand of the logic equation implemented into the CLB.

  Figure 7 shows how the logic function *y=a&b|c* can be implemented using 4 multiplexers. The FPGA design tool calculates the most efficient configuration of the multiplexers to perform the required function.

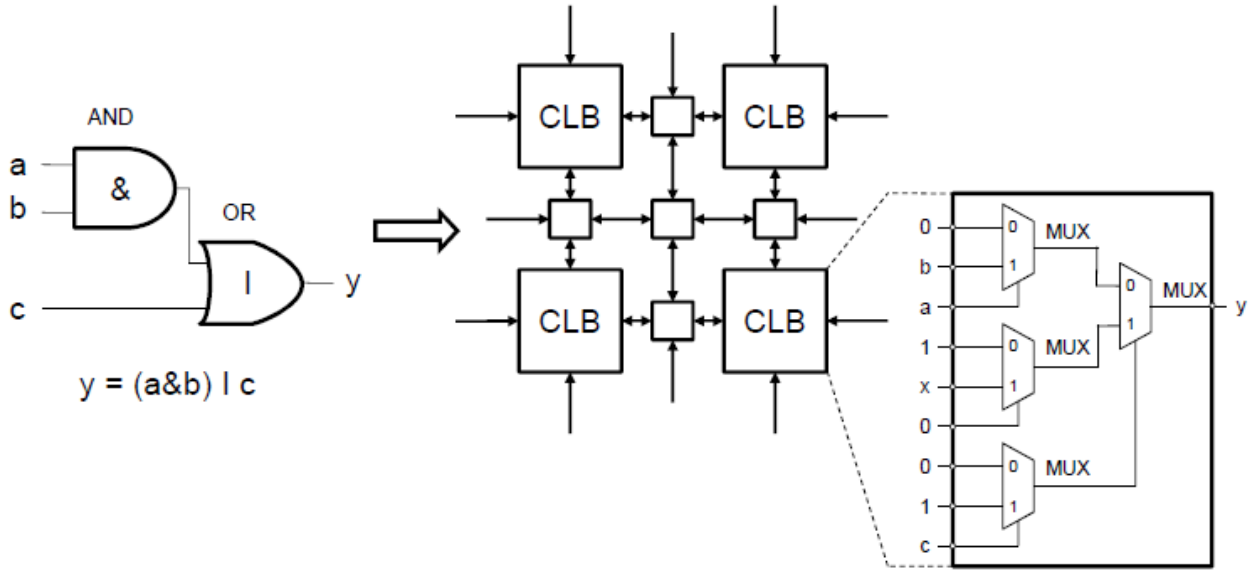

Figure 7 – Multiplexer-based configurable logic block

These two CLB types are classical. However, FPGA architectures are evolving rapidly, and the logic capability and complexity of the CLBs is increasing as this evolution continues.

1.1.4.2 CONFIGURATION LOGIC BLOCK ARRAYS

Each FPGA product line has its own architecture, and offers a choice of sizes and capacities. Typically, the architecture includes a number of sub-modules called “CLB arrays”. A CLB array is composed of multiple CLBs. CLB arrays can be used to implement large combinational logic functions.

1.1.4.3 INTERNAL INTERCONNECTIONS ARCHITECTURE

An interconnection grid can include different routing resources to connect the CLBs and I/O blocks in the device (see Figure 8). There are 4 types of routing resources:

- General Purpose Interconnect (GPI)

  These can be found in all types of FPGAs and are the main resource for routing signals between CLB arrays and between CLB arrays and I/O blocks. A GPI is a grid of horizontal and vertical wires. A switching matrix is located at each intersection.

- Direct interconnects

  These can also be found in all types of FPGAs and have two main purposes:

  - To connect a CLB array to the nearest GPI wires.
  - To directly transmit high-speed signals between adjacent CLB arrays or adjacent CLB arrays and I/O blocks.

- Longlines

  These are present in a large majority of available FPGAs. Longlines are used when bidirectional data busses are required. They are also useful to connect critical CLBs that are physically far from each other and to limit transmission delays. Longlines include low-impedance clock lines for fast signal propagation.

- Internal routing

  A CLB array may include an internal routing grid to link the CLBs within the array.

These routing resources are managed automatically by the vendor mapping, placement, and routing tools.

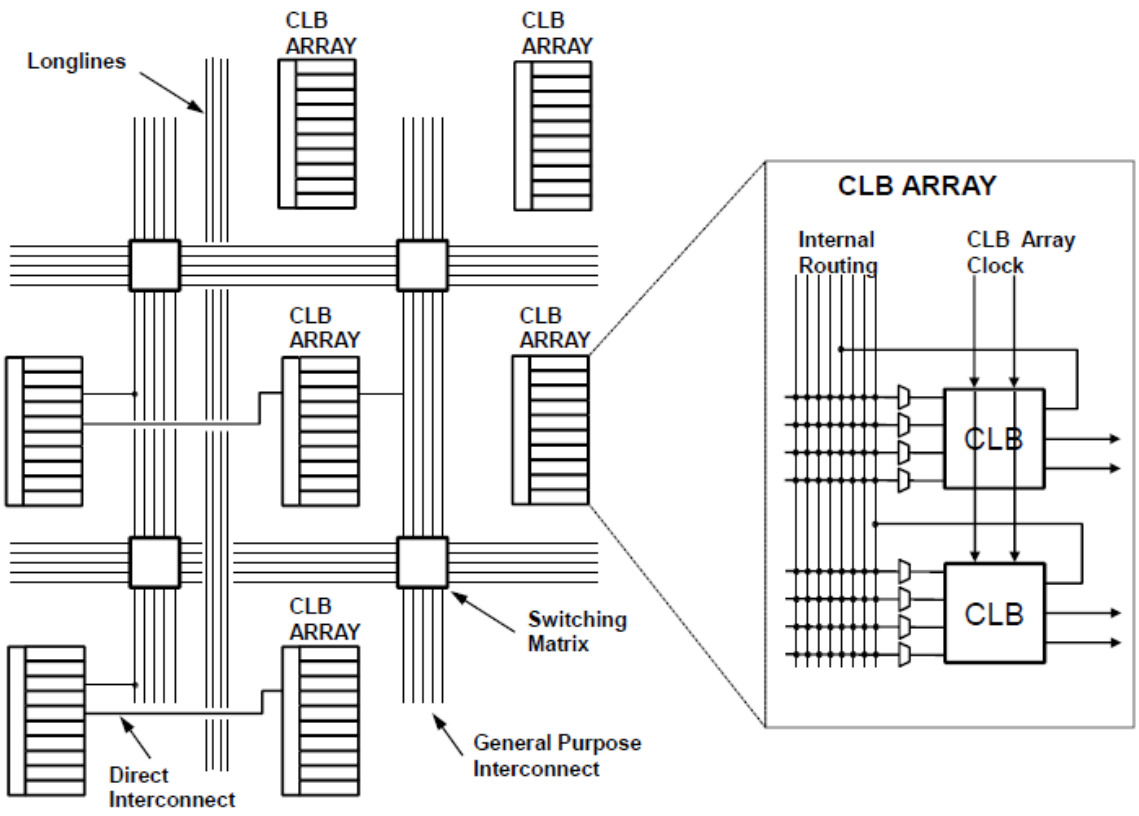

Figure 8 – Field Programmable Gate Array interconnection grid architecture

For example, to transmit a signal from one CLB to a CLB in another CLB array, all the resources of the interconnection grid are used (except longlines). The signal goes first through the CLB array internal routing, then uses a direct interconnect wire to reach the GPI. After going through some switching matrices, the signal switches to another direct interconnect wire and joins the destination CLB through the CLB array internal routing.

1.1.4.4 SWITCHES PROGRAMMING

There are four kinds of programmable switches: Static Radom Access Memory, flash, Electrically Erasable Programmable Read-Only Memory and antifuse.

- Static Radom Access Memory

  The majority of FPGA families are SRAM-based, with static RAM elements driving transistor-based switches: a zero in a RAM element turns the corresponding switch off, while a one turns the switch on. Figure 9 shows how routing wires between two logic blocks can be SRAM-programmed in a switch matrix. On the left side is an 8x8 switching matrix with 64 possible interconnections. The right side shows details of one interconnection using six SRAM elements (“SR” in the figure) driving six transistors.

  The weakness of this option is that they are volatile. Thus, at power-up they must be reloaded from an external configuration system. Also, most (but not all) SRAMs are

susceptible to random, radiation-induced hardware alterations, or SEUs. When this is the case, no part of the FPGA can be fully trusted, not even the FPGA internal self-monitoring functions, because the programming may have been affected by the SEU. This susceptibility should be addressed as required in the design, based on the expected environment and the risks associated with the impact of SEUs.

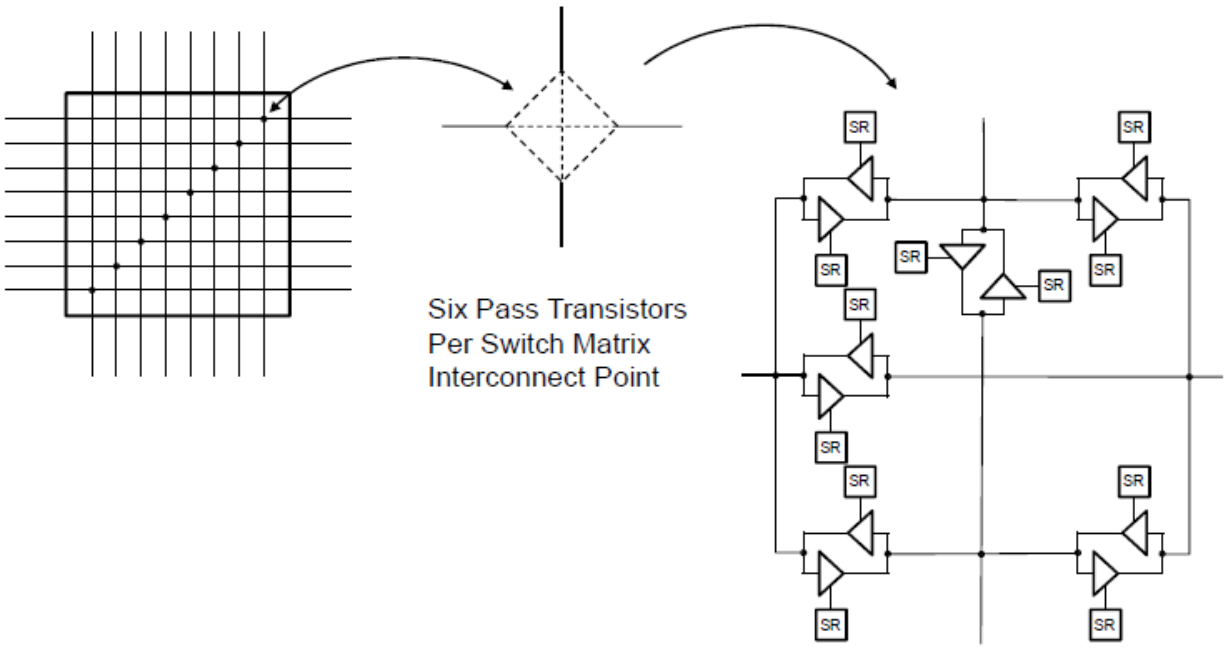

Figure 9 – Static Random Access Memory based interconnection using six elements

- Flash and Electrically Erasable Programmable Read-Only Memory

  A number of flash-based and EEPROM-based FPGA families are available. These technologies use re-writable and non-volatile switches. Flash memory is a modern evolution of EEPROM. The main difference is that in EEPROM memory, bits are written one after another, whereas for flash memory, large domains can be written simultaneously. Both technologies are in general slower and less dense than SRAM.

  Tests and experience in other industries have shown that these technologies can have a low sensitivity to SEUs, and that programming can be retained for more than 20 years. This is a great advantage for this technology.

- Antifuse

  A few specialized FPGA families use antifuse technology. A high current creates a contact between two wires of the interconnection grid. This makes the devices non-volatile and immune to SEUs. These devices are programmable only once. If the configuration of the FPGA needs to be modified, it will be necessary to physically replace the component. This technology is the best option for harsh environment, like in the aerospace industry, and it is also good option for nuclear industry in the case of high

radiation environment (e.g. field instrumentation or local controls). It is also the best option in terms of speed.

1.1.4.5 RANDOM ACCESS MEMORY BLOCKS

Most applications require data memory capabilities apart of that required to store the application program.

One option is to use the RAM elements of the CLBs (for example, the RAM that implements look-up tables). However, this is not recommended except for very basic memory needs.

For applications with more extensive needs, two options are possible: use of external memory circuits, and use of FPGA circuits with embedded memory blocks. External memory may create bottlenecks, depending on how intensive in memory the application could be and the communication bus architecture. Embedded memory blocks are important parts of many FPGA architectures. They compensate for the shortage of memory capacity in CLBs, and they come in various sizes and architectures.

1.1.4.6 INPUT/OUTPUT BLOCKS

I/O blocks are the electrical interfaces between the low-voltage, low-current signals within the FPGA, and the higher voltages and currents required by the external electronic components connected to the FPGA.

Programmable I/O blocks can perform many useful functions. For example, an I/O block can be used to implement:

- A carrier data recovery circuit that extracts the clock from the incoming data.
- An embedded monitoring algorithm that compensates for data skew and process delays.
- A FIFO (First In, First Out) buffer to synchronize the data to a clock signal.
- Analog-to-digital (A/D) and digital-to-analog (D/A) conversion.
- A programmable decoder/encoder that can read/write data in different formats to handle different communication protocols.

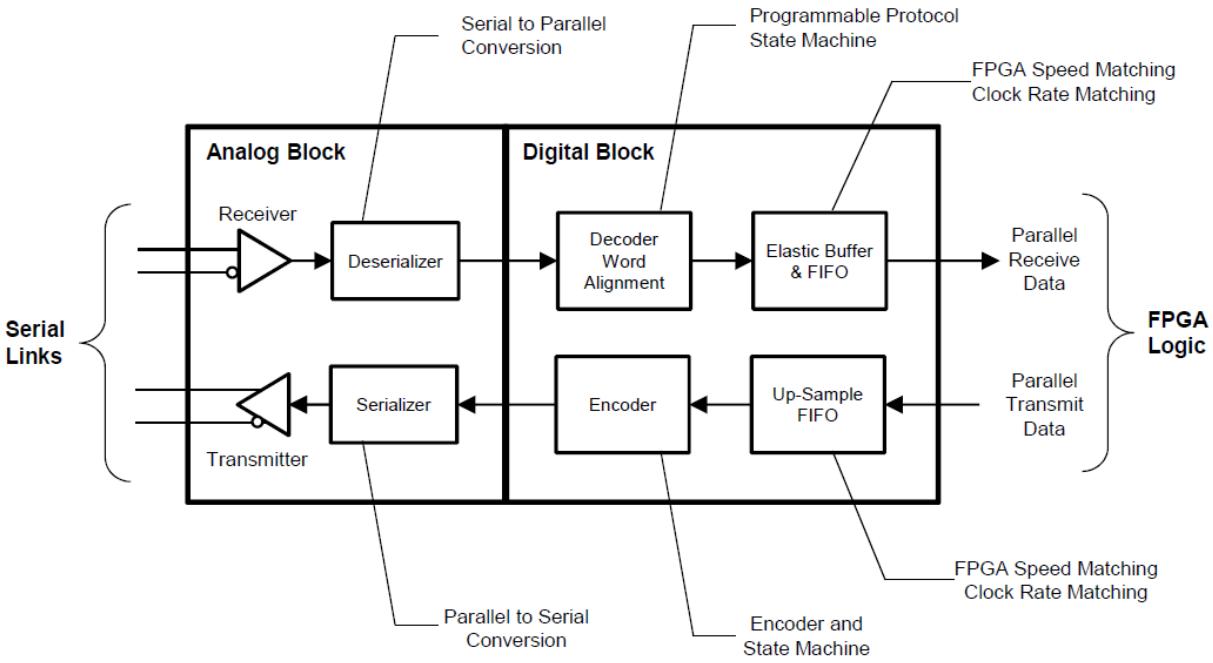

Figure 10 –Example I/O Block

Figure 10 shows an example of a configurable I/O block. The high-speed analog portion performs clock extraction and data serialization/de-serialization. The other block performs protocol-specific data extraction and formatting, word boundary identification, and clock rate matching with buffering.

1.1.5 FIELD PROGRAMMABLE GATE ARRAY PROGRAMMING

The FPGA programming process is usually composed of four main phases, along with the associated verification and validation activities, as illustrated in Figure 11:

- Component requirements specification

  The objective of this phase is to systematically and precisely state all the requirements that apply to the final FPGA circuit. These requirements usually result from the I&C system architectural design that decomposes the system into components and allocates system functional and safety/dependability requirements to each component.

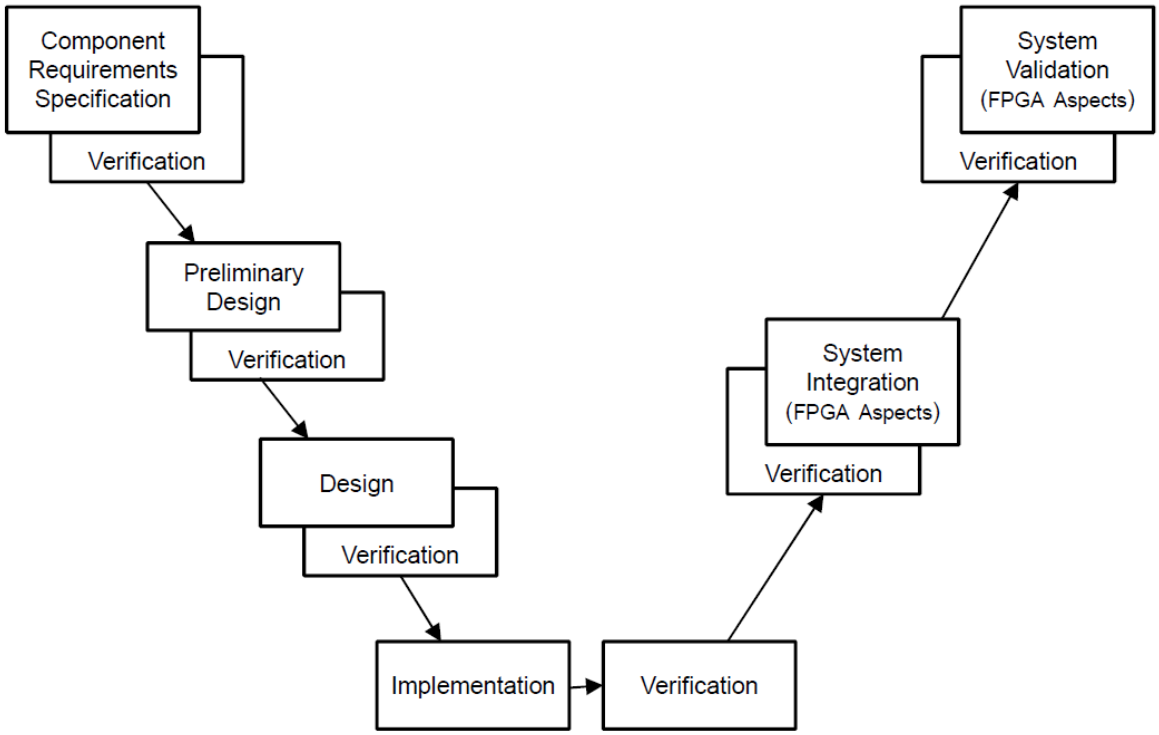

Figure 11 – V-shaped Field Programmable Gate Array programming lifecycle

- Preliminary design

  The objective of preliminary design is to decide on the major design choices such as the balance between combinatorial logic and sequential design (see Figure 12), decomposition into modules (application-specific or pre-developed), the implementation of defensive design practices, the identification of the circuit or circuit family to be used, and the needed library functions and IP cores.

- Design

  The objective of the design phase is to develop a detailed description of the logic processing to be performed by the FPGA component (like software source code for microprocessor based applications). This step is often circuit independent (i.e., portable to different FPGA architectures), unless specific features of a specific circuit or circuit family are relied upon.

  The design is usually expressed using a Hardware Description Language (HDL). A number of languages are available such as Verilog and Very-High-Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL).

  A design can be described at different levels of detail. The most common level used today is called Register Transfer Level (RTL). This level describes the functions of the FPGA in terms of a flow of signals (or data transfers) between registers (flip-flops or other memory elements), and logical operations performed on those signals. Changes in registers are simultaneous and follow the ticking of a clock signal (see Figure 12).

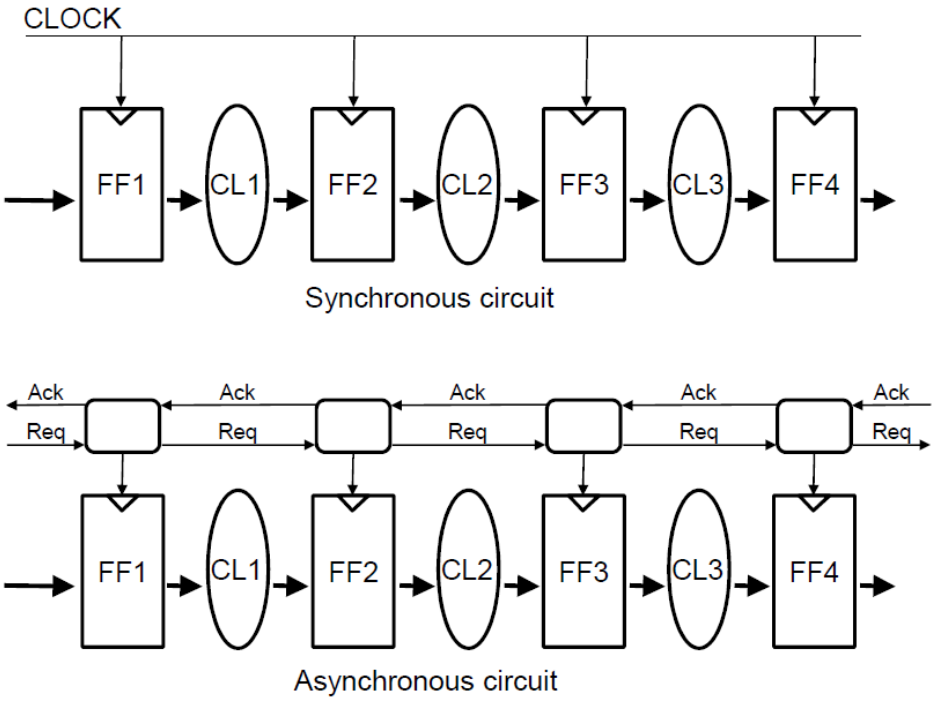

Figure 12 – Synchronous vs asynchronous design

Testing of the design (which is not integrated yet into a physical FPGA) is usually based on simulation. Timing errors usually need more detailed information on the FPGA circuit, and are looked after in the implementation phase.

Formal verification techniques and tools are also available. Such techniques usually start with a formal specification of the required design properties (using for example languages like Property Specification Language, or PSL). Formal verification tools can then systematically check that these properties are indeed satisfied by the RTL. If not, most tools can exhibit a counter-example that will help the system designer locate the error.

- Implementation

  Once the design is verified to satisfaction, the project goes on with the implementation phase. Implementation is usually divided into two main steps: synthesis and place&route. Both steps are in large part supported by appropriate tools provided by the circuit vendor (vendor specific).

  - Synthesis

    The objective of synthesis is to translate the circuit-independent RTL into an equivalent description that is expressed in terms of the resources provided by the selected FPGA circuit. This circuit-dependent description is called a netlist.

    Synthesis is itself divided into two main steps: elaboration and logic synthesis.

The elaboration step is independent of the hardware targeted for the final implementation. It may include an optimization phase, where the designer must be careful to avoid any unwanted modifications that might be performed as part of optimization. For example, redundant blocks could be removed by the optimization tool. To prevent this, it may be necessary to check that elaboration outputs are consistent with the HDL code. This task is greatly facilitated when the synthesis tool provides results also in diagrammatic forms.

The logic synthesis step is circuit dependent. For example, from a function described by the FPGA designer in HDL, the synthesis tool may proceed to a functional decomposition to determine specific sub-functions to be mapped into the LUTs provided by the selected FPGA circuit. The degree of functional decomposition is dependent on the LUT size in the FPGA.

When a design needs to be ported to a different type of FPGA, the implementation process must be repeated from synthesis onward.

Like for the design stage, testing of the synthesis is mostly done by simulation. In addition to the netlist, logic synthesis tools can produce a timing file. This file is used by simulation tools to simulate timings more accurately than was possible with the RTL. Formal verification techniques are also available to demonstrate that the netlist is equivalent to the RTL.

- Place & Route

  The netlist is expressed in terms of resources provided by the FPGA circuit. However, they are not yet identified by their physical position in the circuit. The place&route tool calculates the best physical positions and mapping for the FPGA resources that are used. In particular, it aims to reduce inter-CLB bottlenecks by distributing data transfers uniformly over the interconnection grid.

  Many factors can affect the place&route process, including items such as clock frequency and maximal signal setup times.

  The place&route tool also generates a timing file that is more accurate than the one produced by synthesis, since it also includes timing associated with routing.

  Ultimately a bitstream is generated. The bitstream is the digital data that is sent to the FPGA circuit to program it.

To design an FPGA application, several steps must be taken, and each step needs specific tools. Some of these tools can be obtained independently from FPGA vendors, while others are provided by the FPGA vendors themselves. Experience has shown that for a given FPGA family, using the vendor's tools typically gives better results because they have been specifically designed for the particular circuit and represent the accumulated experience of working with it, whereas an independently developed tool may be more generic. However, the tool selection

process should include a careful evaluation of the available tools and determination of which ones will be most suitable for the given project.

Vendors usually provide all their tools in an integrated package. This facilitates interactions between the tools. Having a toolset (and also sets of IP cores, discussed further below) provides a significant commercial advantage for the vendor. Therefore, there is an incentive for the vendors to ensure that the toolsets they provide are of high quality.

The programming process for a CPLD is much the same as that used for an FPGA, and some of the same tools or types of tools can be used.

1.1.6 PREDEVELOPED HARDWARE DESIGN

1.1.6.1 INTELECTUAL PROPERTIES

Pre-developed designs for logic functions or general-purposes interfaces, usually called Intellectual Properties (IPs), can be obtained independently from the FPGA circuit, or from the circuit vendor, and then integrated into the complete design as required. There are two main types of IPs:

Soft core IPs are provided to system designers in a format independent of any FPGA circuit (in RTL for instance).

Hard core IPs are provided in a format that allows their use only with a given circuit or circuit family. Hard core IPs are often offered by the circuit vendor, or by IP vendors who want to protect their intellectual property.

1.1.6.2 ANALOG INPUT/OUTPUT BLOCKS

Reducing the number of physical components on critical I&C electronic boards can improve reliability. When inputs originally come from analog signals, hybrid or mixed-signal FPGA solutions that have embedded analog I/O blocks are worth considering.

These blocks can be connected directly to analog input signals and perform analog-to-digital conversion (ADC), eliminating the need for external resistor divider networks, reducing component count, and increasing accuracy. The ADCs can support different conversion modes (8 bits, 12 bits, etc.). They can also drive analog outputs, eliminating the need for external level shifters and drivers.

Another type of field-programmable devices, but made of analog components instead of logic ones (gates), are Field-Programmable Analog Arrays, or FPAAs. The concept is very similar, and there are several manufacturers with commercial products available in the market, like Anadigm. Nevertheless, it seems that FPAAs has not reach the same level of acceptance as FPGAs, neither for analog design.

1.1.6.3 DIGITAL SIGNAL PROCESSORS

When complex and intense mathematic computations are required, FPGAs with embedded Digital Signal Processing (DSP) blocks can be considered.

A conventional (not FPGA-based) DSP is a specialized microprocessor well-suited to extremely complex math-intensive tasks. Its performance is determined by the clock rate, and the number of useful operations it can do per clock cycle. This is fixed by the DSP architecture.

In contrast, an FPGA-based DSP is a configurable logic block dedicated to perform complex mathematic operations. DSP blocks typically include a multiplier block, an adder/subtractor/accumulator block, and a summation block, and can be configured to maximize the number of mathematical operations that can be performed in parallel during each clock cycle.

1.1.6.4 MICROPROCESSORS

A microprocessor IP can be useful to integrate in the FPGA circuit for secondary functions (for example, human-system interfaces) that are more easily implemented using conventional software. This allows taking advantage of the benefits of the FPGA technology for the primary functions, while facilitating the implementation of complex secondary functions in the same device.

Many FPGA vendors offer RTL-level soft core microprocessors. They even offer a few customization options (e.g., instruction set and data and address size). Hard core microprocessors are not customizable but are usually designed to achieve higher performance than what can be achieved with soft core microprocessors.

Finally, an FPGA can be used to emulate an existing microprocessor. This allows existing legacy software to be maintained while replacing an obsolete microprocessor chip with a new FPGA-based emulation.

Probably, the most limiting aspect of FPGAs for used in I&C systems in NPPs is the poor performance for Human-Machine Interface (HMI). Nevertheless, it should be taken into account that, for safety systems, HMI requirements are generally not necessary or are minimal compare to control systems. Additionally there are some options on the market for simple HMIs that can be used for this kind of applications, even with qualification requirements, although for HMIs qualification is not usually required (except for post-accident monitoring purposes, for instance, according to Nuclear Regulatory Commission Regulatory Guide 1.75).

1.2 ADVANTAGES AND LIMITATIONS OF FIELD PROGRAMMABLE LOGIC TECHNOLOGIES

Following there is a discussion on advantages and limitations of FPGA-based solutions for nuclear plant I&C systems. Because of the similarities between CPLDs and FPGAs, much of this information applies to CPLDs as well.

1.2.1 ADVANTAGES

1.2.1.1 ADEQUATE CAPABILITIES FOR A WIDE RANGE OF APPLICATIONS

Most critical I&C functions involve relatively simple processing of a limited number of input signals, and require response times that are not very demanding – in the order of tens of milliseconds or longer. Traditionally, such functions have been implemented using relay-based components, analog components, or low-integration electronic components. Thus, the functional and performance capabilities offered by current FPGA circuits, which feature from tens of thousands to millions of gates and operate in the hundreds of MHz range, are more than adequate to support such functions, even when taking consideration into more ambitious I&C functional requirements, for example to support increased self-monitoring and fault-tolerance capabilities.

Because FPGAs can process independent functions in a parallel hardware implementation and with high clock speeds, they may be well suited to applications that require very short response times for certain functions. The cycle times of microprocessor-based systems may be too long to meet very short response time requirements. This is one of the major concern for suppliers and regulators related to the used of microprocessor-based systems, along with testing coverage capabilities and, as result, confidence in non-existence of software common cause failure (SCCF).

Although FPGA applications are particularly well suited to logical function processing (e.g., safety system actuation logic and component control logic) they are not limited to this. FPGA applications can be built to provide closed-loop control functions such as a proportional-integral-derivative (PID) control as well as more complex control functions and those that include mathematical calculations. Some FPGAs contain embedded microprocessors that can be used for even greater functionality, such as a simple human-system interface (HSI). However, it should be noted that as greater complexity is introduced and certainly when a microprocessor is used, requiring run-time software, some of the advantages of using FPGAs can be lost or significantly reduced.

The complexity referred to here comes from the presence of hardware and software components that are not directly involved in performing the primary I&C functions but are not independent, and thus could interfere with the I&C functions. Examples are operating systems, peripheral hardware and software and associated drivers, and ancillary functions that may be desired (e.g., self-testing and diagnostics) but are implemented in such a way that they could impact the primary I&C functions. These additional, potentially interdependent hardware and software components complicate the design, assessment of reliability, and safety justification. This is discussed further later.

The important point here is that the greatest number of advantages from use of FPGAs is obtained when the application is designed and implemented in “flat hardware logic” without use of embedded processors or other IP cores that bring added complexity. Limiting the application to flat hardware logic is especially helpful for safety applications as it greatly reduces effort required in design verification and safety justification.

1.2.1.2 SIMPLER, MORE EFFECTIVE SAFETY AND RELIABILITY JUSTIFICATION

Compared to possible FPGA-based solutions, the software-based distributed control systems (DCSs), programmable logic controllers (PLCs) and programmable automation controllers (PACs) that are currently proposed and used for I&C applications are significantly more complex, even when specifically designed to support safety applications and to comply with nuclear industry standards. In addition to application software that implements the required I&C and auxiliary functions, such systems necessarily include other components such as operating systems and microprocessors, which introduce a high complexity overhead. Even the simplest operating systems, specifically designed to the safety requirements and constraints of the nuclear industry, usually feature tens of thousands of lines of code. And the microprocessors used in PLC/PAC/DCS are typically not developed specifically for the nuclear industry. They are designed for larger markets where intense competition and drive for performance mean that individual models have brief commercial lifetimes and increasing levels of complexity. In fact, there is some concern within the nuclear industry as to whether it will even be practical to provide appropriate safety justification at a reasonable cost for the microprocessors that will be available for I&C and process control applications just a few years from now.

On the other hand, FPGA-based solutions can be designed using “flat hardware logic”, such that their complexity is in more direct relationship with the complexity of the I&C and auxiliary functions that are required, more like relay-based or analog-based solutions. Figure 13 shows a qualitative comparison between different technology options for a certain system.

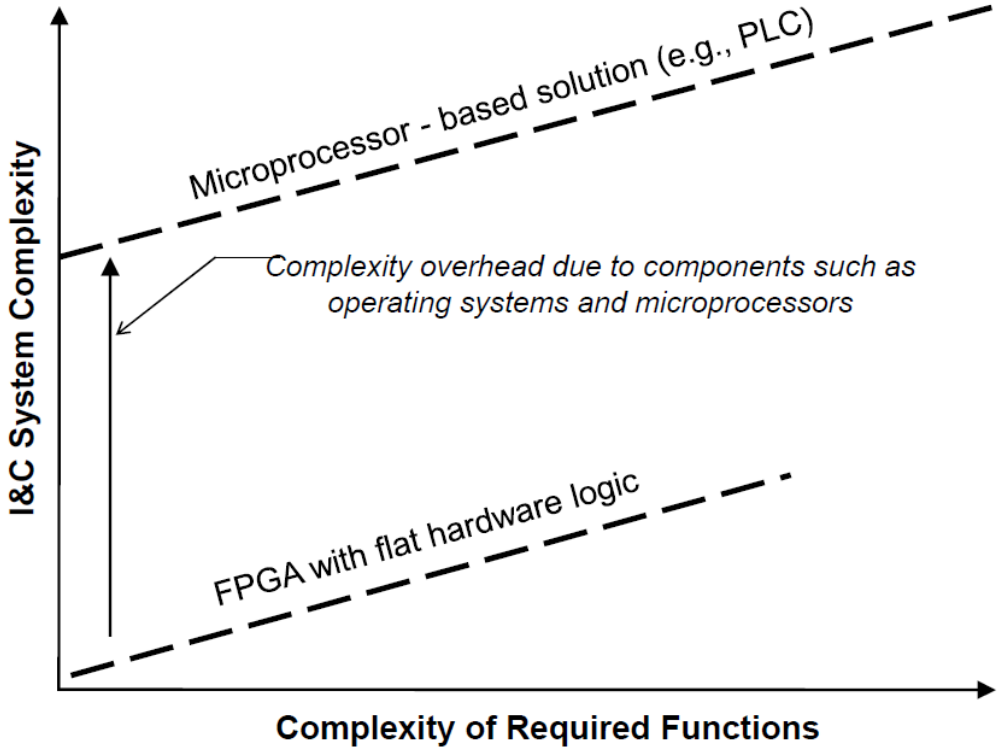

Figure 13 – Complexity of instrumentation and control solutions

The term *complexity* as used here should not be confused with *capacity* – a modern FPGA can have millions of gates, capable of a great deal of functionality, just as a microprocessor can have significant computing capacity and memory for program storage, for example. But capacity (gate counts in an FPGA or in memory available to a microprocessor) is not the type of complexity referred to here. Complexity in this context refers to hardware and software components that are present in the design and could impact the proper functioning of the I&C application. A primary difference between an FPGA flat hardware solution and a microprocessor-based solution, as noted above, is the presence of software components including an operating system plus other peripheral software and hardware components and drivers. In a typical software-based system, these cannot easily be shown to be independent of the application software and thus must be scrutinized in determining potential failure modes and demonstrating reliability and safety of the solution.

Also, a microprocessor-based system often has functions built-in that are not needed for a given I&C application but cannot easily be removed from the system, especially when using equipment that is designed for use in many different types of applications. It may be difficult to show that the unused software components are independent of the I&C application if, as is often the case, they could potentially impact the processing of the primary functions due to a software error, memory conflict, or other type of error. Also, as noted above, ancillary functions such as self-testing and diagnostics that are desired as part of the solution, but are not necessary for performing the primary functions, can be implemented in an FPGA such that they are

independent of the primary functions. This may not be the case with a software-based solution, again due to potential interdependencies through the sequential processing of these functions by the microprocessor.

These additional hardware and software components with their potential interdependencies are what contribute to the complexity overhead. In contrast, a flat hardware solution in an FPGA is purely hardware, with parallel signal paths from inputs through functional logic to outputs, more like the conventional hardwired electronics that was originally used in plants. I&C functions that are logically independent can be implemented such that they are independent and separated in the circuit, and ancillary functions, if needed or wanted, can be similarly segregated and kept independent. Thus the complexity of the end product depends primarily on the complexity of the I&C application itself, without a lot of additional complexity overhead. This can enhance reliability and testability, and ease the burden of demonstrating adequate reliability and safety for an FPGA-based flat hardware logic solution.

It is said by some suppliers that FPGA-based solution are completely or 100% testable. Nevertheless, this is not usually the position of the regulator, as can be seen in recommendations form NUREG CR-7006 [4] or Safety Evaluation Report for Wolf Creek Licensee Amendment Request (LAR) for implementation of Main Steam and Feed Water isolation based on Advanced Logic System (ALS^{TM}) [5].

It is important to note that this discussion refers to complexity of the end product installed in the plant. Although a flat hardware logic solution in an FPGA is relatively simple, the design process used to create it is not. Complex software tools are used to design and verify the application before it is implemented in the actual circuit. However, this is true of microprocessor-based solutions as well – relatively complex software tools (e.g., graphical application-builders, compilers, etc.) are used in developing the run-time software that is installed in the plant system. Also, software tools were used in creating the design of the microprocessor itself and the related circuitry of the system in which the microprocessor is embedded. In either case, whether FPGAs are used or microprocessors are applied, part of the safety justification is demonstrating that the tools are of high quality and that any errors introduced by the tools would be detected through appropriate verification and validation (V&V) activities.

As discussed earlier, some FPGAs incorporate embedded functionality beyond the basic logic elements used to create a flat hardware logic solution. Functionality can be embedded directly in the hardware of the chip or provided in the form of IP cores. This includes the capability to embed a microprocessor in an FPGA. While such functionality may be appropriate for non-safety applications or HMI functionality, it is important to remember that the advantages of lower complexity and independence of functions discussed above can be lost when more processors and run-time software are introduced into the product. Demonstration of adequate reliability and safety is more straightforward when the safety function is implemented in flat hardware logic.

That is why it is highly recommended to be implemented in a completely independent way and that independence be completely verified.

Another beneficial feature of FPGAs is their ability to process separate functions independently and in parallel on the same integrated circuit (provided of course that these functions are logically independent). Thus, ancillary functions (such as self-monitoring or configuration functions) may be separated from the main safety I&C functions, so that a postulated failure of an ancillary function will not prevent the correct execution of the safety I&C functions. In the same way, logically independent safety I&C functions may be prevented from interfering with one another. This is in contrast to microprocessor-based solutions, where achieving independence between functions running on the same microprocessor is much more difficult due to the shared operating system and other software services that all functions rely upon to operate correctly. This is also true even with modern virtualization technologies (hypervisors) or the use of multithread and multicore microprocessors.

In addition to limiting the scope of a failure within a circuit, maintaining separation and independence between functions is also useful in lowering complexity levels and facilitating verification, analysis, testing, and ultimately, safety justification.

1.2.1.3 CYBER-SECURITY

Cyber security is a concern with FPGA-based systems as it is with computer-based systems. However, FPGA-based solutions have characteristics that tend to increase the level of difficulty that would be faced by a would-be attacker as compared to conventional microprocessor-based systems:

- FPGA-based systems that directly implement the required I&C functions do not contain high-level, general-purpose components that could be easily diverted or hijacked for malicious purposes; malicious functions must be introduced as complete designs, using technology-specific engineering tools – this raises the level of difficulty a would-be attacker would face in attempting to make malicious modifications.
- Some of the FPGA technologies currently used for safety or critical applications (antifuse technology, for example) can be used in such ways as to require physical access to, and disabling of, the I&C equipment in order to alter the current programming. In these cases, more than a cyberattack, it would be a sabotage. With antifuse technology, there is not even a non-volatile memory to store FPGA program that could be change and take effect after the subsequent power-up or schedule re-setup of the FPGA.

Any safety I&C system is required, by Plant Technical Specifications, to pass its corresponding surveillance tests before being declared operational, ensuring the system performs as required and expected. This is another layer of protection against cyberattacks.

The worst case scenario would be a logical bomb programmed on the system. In this case, the system would perform as appropriate, passing all kind of testing, including Plant Technical Specifications surveillance testing, until the bomb is activated (by means of a signal status or a combination of several ones, or by time). In this case, physical access control to programming ports, even in storage (warehouse), and disabling those ports once the device is programmed, appears to be the most effective measures to be implemented. Tight configuration control and use of hashing for electronic files (RTL, netlist and bitstreams) are highly recommended during the design and implementation phases.

Additionally, safety systems usually require less communication capabilities and integration with other systems or operator panels than control systems do. In case communication protocols are needed, these can be developed as serial one-way communication datalinks to prevent any attempt to perform malicious activities over the system.

1.2.1.4 APPLICATION AS A DIVERSE ACTUATION SYSTEM

FPGA-based equipment provides an attractive design option when Diverse Actuation Systems (DAS) are required according to current regulations in the case of safety functions implemented using microprocessor-based systems. NUREG 800 Standard Review Plan and its associated Branch Technical Positions, NUREG CR-6303, NUREG CR-7006 and Interim Staff Guidance DI&C-ISG-02 are good references for diversity requirements and recommendations.

FPGA-based equipment can provide a more practical and cost-effective solution compared to providing a diverse microprocessor-based system, or when diversity requirements demand a non-microprocessor-based solution. It should be noted, however, that this may only provide equipment diversity; other forms of diversity such as functional diversity may need to be considered depending on the application and the types of failures the diversity is intended to address.

FPGA-based solutions for DAS are being employed both in modernization projects, like in the Diablo Canyon NPP replacement of Eagle21™ Reactor Protection System with Tricon ™ (a triple modular redundant, or TMR, micro-processor based pre-qualified platform), or in new builds, like in the case of Westinghouse's AP1000 ™ reactor, where the safety systems are based on Common-Q ™ (Asea Brown Bovery AC160 ™) platform.

Other possibilities also exist without employing FPGAs, like in the case of Oconnee NPP, where Nuclear Regulatory Commission approved the replacement of its Reactor Protection System with a new one based on the Teleperm ™ XS platform from AREVA/Siemens, making use of classical relay-based technology for the implementation of the required DAS.

1.2.1.5 UPGRADES TARGETING SPECIFIC COMPONENTS

An I&C upgrade using a PLC-based solution typically must be performed on a system basis: it addresses the complete I&C functionality (processing of inputs, control algorithms, and outputs) for an entire system, or even multiple plant systems. This implies significant costs, including recapturing of requirements, system development and validation, installation on site (including removal of the existing system, and rewiring as necessary), commissioning, and training of operations and maintenance personnel. Installation can be particularly costly if it requires extending the normal outage time, resulting in lost revenue.

FPGA-based solutions can be implemented at the system level as well. However, an I&C upgrade using an FPGA-based solution can alternatively be targeted just to the specific components of the existing I&C system that need to be replaced. Components may be individual circuits on a portion of a circuit board, or complete boards or sets of boards. In some cases it has been estimated that an upgrade that is focused on replacing solely the problematic boards, does not require extensive rewiring, and can be accomplished within a normal outage period, reducing costs by an order of magnitude.

Examples of this are the new-design CPLD-based Solid State Protection System boards from Westinghouse or the Motorola 68000 processor emulation daughter board for EDF reactors.

1.2.1.6 REDUCED NUMBER OF COMPONENTS AND LOWER POWER CONSUMPTION

FPGA-based solutions can be designed to integrate in one circuit (one IC or integrated circuit) what originally required multiple boards in the current digital (discrete digital) system architecture. The reduced number of hardware components and associated interconnections can have a favourable impact on cost and also reliability.

In addition, FPGA-based solutions typically consume less power than conventional electronics due to the reduced number of parts. They also tend to consume less power than microprocessor-based systems. Further, the heat load for FPGAs is lower, potentially reducing cooling system requirements and cost, and reducing or eliminating the need for cabinet fans.

Reducing moving parts and components contribute to increase the reliability of the system, as moving parts are more prone to failures due to wear. Even if this is accomplished in auxiliary equipment, like ventilation or cooling, this can have a significant beneficial impact in the system, for instance, temperature impact in electronics as electronics tend to drift with temperature and age faster. According to Arrhenius Lay, an increase in 10 ºC of temperature reduces electronic components life by a factor of two.

Although in nuclear sector it is not important at all, in consumer electronics, where FPGAs are massively used, reduce footprint and power consumption is critical (for example, in cell phones, in order to reduce the size and increase battery life). This has led to, for example, “3D” FPGA projects, where the term 3D can be understand in two different ways. On one hand, there are

project for development of FPGA integrated circuits constructed in a 3-spatial dimensional way. One example is Monolithic3D (former NuPGA) products. On the other hand, other research projects aim to reconfigurable FPGAs, that is, in order to implement more logic in the same circuit, the FPGA is reprogrammed more than once in each application execution cycle, being each programming different from one another. Due to the fast setup times and execution, this appears transparent to the user. It could be said it is a "multiplexed programming" implementation. One example of this is the ABAX Tabula™ family, with a reconfiguration time of only 8 ps. Signals can even being interchanged between different time-folds.

Lower feature sizes development also helps in these purposes, as they allow lower footprints and require less power consumption. Todays' technologies reach the 20 and 18 nm. Tendency follows Moore's Law, but there are nowadays concerns about if the limit of Moore's Law is near to the end. This, as will be discussed later, has negative effects on the integrity of the FPGA, as it becomes more prone to SEUs.

For all these reasons, 2D, non-reprogrammable and high feature size FPGAs are best suited for safety application in the nuclear industry.

1.2.1.7 PORTABILITY OF INSTRUMENTATION AND CONTROL APPLICATIONS

Besides the difficulties of obtaining appropriate safety justification and reliability evidence, a significant concern with microprocessor-based technology is its rapid obsolescence and short lifetime cycles. It is not unusual for relay-based and analog components to be maintained in full operation for thirty years. It is unlikely this will be possible with microprocessor-based equipment (hardware and software). Thus, considering the average age of nuclear power units at the time of their first major I&C upgrades (around thirty years), and with the life extensions to sixty years that have been granted to many units (more than 30 in the US), utilities need to consider that the digital I&C systems that are installed in the first round of I&C upgrades will also need to be replaced in their turn. In the case of new reactors, where the initial life is 60 years, the situation in this sense is even worse.

Some microprocessor-based safety systems suppliers have developed strategic plans to assure very-long term support for these digital platforms, but as far as many of them make use of information technology commercial of the shelf (IT COTS) dedicated equipment, this is not true in all cases or for all the components within the system. Even though new qualified hardware could be supplied, in many times it is not compatible with older software versions, so the need to move to newer ones appears, with the associated costs and licensee uncertainties.

Cybersecurity for these systems is also a concern, and malware prevention products are usually third party products and are supported for very short times, apart from the compatibility and configuration requirements in order to not affect system's functionality, which is not usually easy to define at all, neither for the system supplier.

Increase level of spares for long-term needs could be another response to these concerns, but it implies higher immobilized costs. Virtualization can help also in alleviating some of these concerns. This, along with used of cluster technology, can improve system dependability in terms of availability and maintainability. Nevertheless, once again, in case of safety system, all hardware and software (including OS and hypervisors, and virtualized machines) shall be qualified and licensed, and the need for periodic changes will not disappear.

As noted previously, PLC/PAC/DCS platforms are complex, integrated systems that use components that are driven by mass markets and become obsolete very rapidly. Thus, these platforms used in the first round of I&C upgrades are unlikely to be still available at the time of the second round of upgrades, and few if any PLC vendors can guarantee full upward replacement compatibility over two decades or more. It is therefore possible that the second round of I&C upgrades will be as difficult, risky, costly and traumatic as the first one if microprocessor technology continues to be applied. The same reasoning also applies for new plants; their I&C systems may have to be upgraded several times over the operating lifetime of the plants.

By contrast, FPGA-based solutions can be designed to facilitate future replacements of aging and/or obsolete FPGA circuits. In the FPGA design process, only the final steps (synthesis plus place&route) are dependent on the particular FPGA circuit chosen. Thus, provided that the initial I&C system design incorporates appropriate provisions for this, when the circuit becomes obsolete it can be replaced by another one using the currently-available technology and the same RTL level representation of the design (see Figure 14).

In the event of a circuit replacement, the three leftmost activities may not have to be repeated assuming that appropriate circuit design practices and coding guidelines were followed (e.g., avoiding circuit-specific constructs) in the initial design. Naturally, system-level activities such as validation and hardware qualification still need to be performed. Also, if the new FPGA has a different footprint or pin-out, the circuit board may require some redesign.

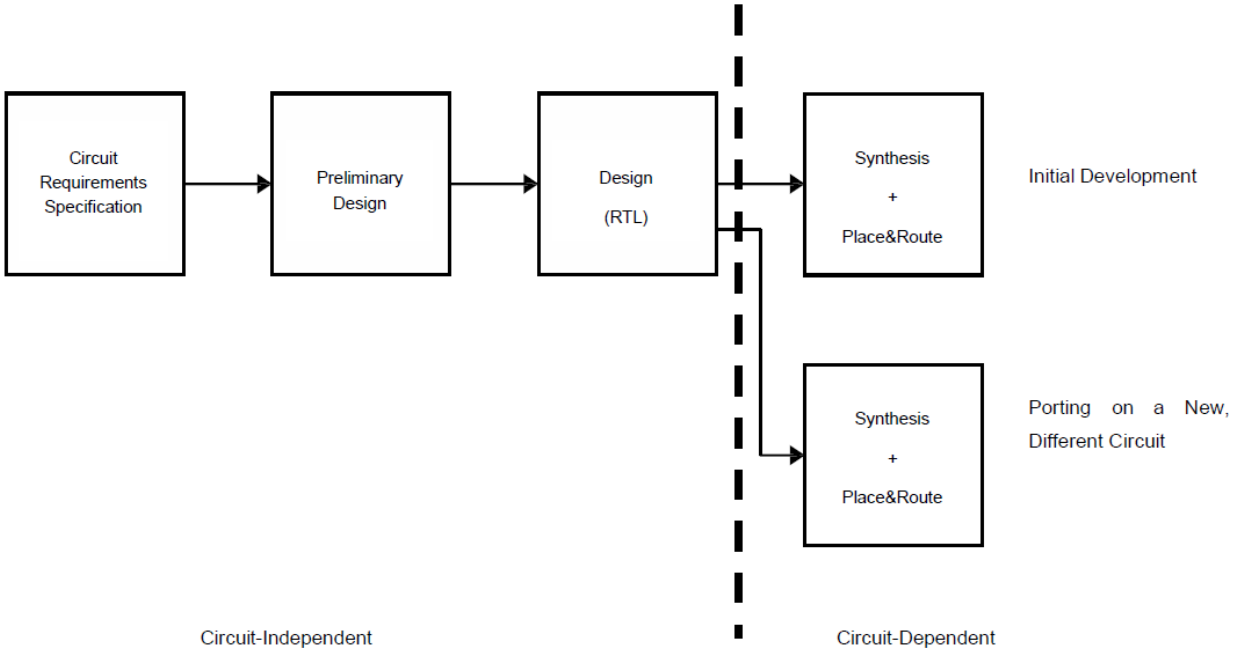

Figure 14 – Portability of Field Programmable Gate Array design

1.2.1.8 COST-EFFECTIVENESS

Many of the advantages cited so far contribute to cost-effectiveness in the case an FPGA-based solution be adopted.

Simpler and more effective safety justification and/or dependability assessment can be a very significant contributor to cost reduction. Targeting specific components (like PCBs) in a vintage system, and not the whole system itself, can reduce dramatically the cost of an upgrade. Finally, portability in RTL format is a great advantage to deal with obsolescence and manage the lifecycle of a system.

Additionally, operating costs derived from software, mainly due to patches, actualizations, etc., and mainly for antimalware appliances or applications, would be significant, if not completely, removed.

1.2.2 LIMITATIONS

1.2.2.1 INEXPERIENCE OF THE NUCLEAR INDUSTRY

Though FPGAs have been widely used in various industries and in consumer products for decades, they are still very new in the nuclear industry. One can find many FPGAs and CPLDs in digital I&C systems already in operation in NPPs, but they are typically buried in the design of electronic circuit boards (I/O boards, network interface boards, even CPU boards) and do not

play as prominent a role as do the microprocessor and its software in a PLC/PAC/DCS-based system.

Thus, until recently, little direct attention has been paid to these ancillary circuits on the part of both regulators and licensees. It is only recently that FPGAs have started to play a central role in the implementation of nuclear plant I&C functions.

Reflecting this relative inexperience is the fact that there is currently only one published international standard that provides guidance and requirements for FPGA-based solutions for the nuclear industry (IEC 62566). Additionally, there is no national specific regulation on this subject. In the case of USA, there are some references, like NUREG CR-7006 [4], the Safety Evaluation Report for the Licensee Amendment Request of Wolf Creek NPP for the replacement of the Main Steam and Feed Water Isolation System [5] and, more recently, the Safety Evaluation Report for the Control System Innovations, Inc. (now a Westinghouse subsidiary) topical report on the Advanced Logic System safety platform [6].

The International Atomic Energy Agency is also developing a publication on the use of FPGAs in the nuclear industry and is planned to be issue by the middle of this year.

1.2.2.2 LIMITED AVAILABILITY OF PRODUCTS

Also reflective of this relative inexperience in the nuclear industry is the fact that at present there are only a limited number of FPGA-based I&C platforms and products that are available and ready to be used in nuclear plant applications. Some examples of these available products are described in the appendixes.

1.2.2.3 HARDER TO ACCESS SIGNALS FOR TESTING AND TROUBLESHOOTING

As compared to conventional electronics (relays, analog, discrete component digital electronics), going to an FPGA-based solution can result in less observability and access to signals within the functional logic. Note that this issue applies to microprocessor-based solutions as well. Special effort may be required during design to provide access to important signals for monitoring, testing and troubleshooting. In many cases this is accomplished through use of the boundary scan technologies, like Joint Test Action Group (JTAG) test access port on the FPGA chip. The specific signals that will need to be accessed and when (on-line or during off-line testing or troubleshooting) must be determined up-front so that the appropriate measures can be taken in design to make them accessible.

1.2.2.4 SUITABILITY FOR COMPLEX HUMAN-SYSTEM INTERFACE FUNCTIONS

Although FPGAs generally offer more than sufficient functional and performance capabilities to implement most I&C control and protection functions, they are generally not well suited to implement complex human-system interfaces such as those used by the main control room

operators; software-based solutions (e.g., industrial workstations) are still the preferred approach.

FPGAs are very well suited to data processing functions including mathematical calculations, image processing, etc., that may be needed to prepare data for display on a control room HSI. However, they may not be the best choice for the user interface aspects – the functionality provided through menu systems and windows-based interfaces allowing selection of different means of information display and providing soft controls, procedure management, and so forth.

The main reason for this is that there are already extensive sets of pre-developed software components supporting the user interface capabilities required for systems such as control room HSIs. This is not the case for FPGAs, and indeed, it would be quite an undertaking to develop full-featured alarm functions, procedure management, and information display systems almost from scratch using FPGAs. And they would not have the benefit of the extensive experience and maturity of commercially-available HSI systems currently used.

Another reason is that functions that are more complex (both in terms of functionality and design) and functions that are user-oriented tend to require more frequent modifications than do the control algorithms, either because of evolving needs, evolving support platforms, or the need for error correction. For such complex functions, the needed modifications are usually easier to implement in software.

Nevertheless, the higher the HSI integration, the higher the risk in case of common cause failures or cyberattacks. The use, for instance, of individual panel-mounted auto-manual stations for a control system that interact directly with system I/O modules, with or without software operator workstations in parallel connected to the system network, greatly help in the cybersecurity assessment of the system, and this scheme with auto-manual stations can be perfectly implemented within an FPGA control system solution.

1.2.2.5 NEED FOR SPECIALIZED EXPERTISE ON DESIGN TEAM

Modern FPGA toolsets provide the capability to develop an I&C application at a workstation using various tools to connect functional blocks, simulate their behaviour, verify and validate the design, and implement it in the FPGA hardware. The design can also be changed and re-verified relatively easily using the provided toolset. Aiding this process is the nearly self-documenting nature of some of these tools. As the tools have become easier to use, the ability to create an application has extended beyond the IC hardware design community from which FPGA technology was born, allowing personnel with a minimal amount of training in the toolset to design and implement an I&C application. The reprogrammability of FPGAs also makes modifications to the application relatively easy to accomplish, often without having to modify the printed circuit board.

An example of this is the commercially available CompactRIO™ (RIO stands for Reconfigurable Input-Output) solution from National Instrument. As commented before, this platform has been used by Westinghouse in the development of the new DADS (DRPI Advanced Display System).

FPGA designs today are represented at the RTL level as “code” written in a hardware description language, and the design implementation is accomplished through successive application of software tools to synthesize the design and place and route connections within the FPGA. The HDL description is similar to software code written in other languages such as Ada, and the design process is quite similar to software design processes, including associated V&V activities performed at successive stages of design development. Therefore, the design team needs to have HDL coding expertise and an understanding of software-like development and V&V processes to ensure that the design meets the application requirements.

At the same time, it is important to remember that in designing the application, one is essentially designing the internal configuration of an integrated circuit – interconnecting fundamental logic elements to each other within the IC and to input and output pins on the chip. The HDL description is not software code that will be translated into computer instructions to be run on a machine, but rather is a description of the behaviour of a circuit, which will go through synthesis and place&route steps to create logic circuits that implement the described behaviour in a physical hardware design. There are hardware design issues that must be addressed properly in order to ensure a reliable and safe initial design, and that reliability and safety are maintained when any changes are made to the design.

Examples of hardware design issues are things such as ground bounce occurring as multiple outputs change state simultaneously, timing glitches that can occur if propagation delays internal to the circuit are not handled correctly, problems with multiple clock domains causing timing issues, etc. The vendor toolsets provide some protection against problems like these, but they are not fool-proof. Personnel with electronic hardware design expertise and a thorough understanding of the particular integrated circuit and its peculiarities need to be involved in the design and in design reviews to ensure that the design will behave correctly and meet reliability and safety goals, particularly for safety-critical applications.

1.3 APPLICATIONS OF FIELD PROGRAMMABLE GATE ARRAYS IN OTHER INDUSTRIES

Today FPGAs are distributed throughout consumer electronics and in numerous industrial applications. Having started as relatively simple and inexpensive “glue logic” devices to achieve compatible interfaces between two or more integrated circuits, FPGAs have quickly evolved into inexpensive substitutes for ASICs. For instance, digital signal processing is effectively monopolized by FPGAs because they provide the capability for hardware-based parallel processing.

1.3.1 MEDICINE

FPGAs offer several advantages in medical applications such as image processing or controlling the dose rates medicines in medical pumps. Computed tomography, magnetic resonance imaging or positron emission tomography (PET) makes intensive use of FPGAs. Also cardiac rhythm management applications like pacing systems or automatic external defibrillators employ FPGAs. Ventilation and life support systems are other medical examples.

Nevertheless, some manufacturers still prefer ASICs for certain applications because their small footprint. This is also possible as this is a mass market where the number of units manufactured is enough high to compensate for the higher costs of development.

Robotic assisted surgery (like DaVinci^{TM}) is another example where FPGAs has an important role, as they can precisely sense and control position, provide force or torque sensing and control these functions through motor speed control.

1.3.2 AUTOMOBILE

Automobile sector has been employing ASICs and FPGAs for many years. Factors that have driven their use include the need for higher levels of safety, pollution control or increasing efficiency of motors as the cost of fuels gradually increased over the years.

Automobile manufactures started using ASICs because the design of application-specific devices was the only way to achieve the low power, high reliability and endurance needed in these applications. Microcontroller units (MCU) provide flexibility but they have high power consumption and cannot meet the response time requirements needed for some applications, like airbags or injection control in real time for each cycle of operation of the motor. CPLDs are used in some of the older automotive process technologies, but they do not provide as much logic density or I/O capacity as FPGAs. FPGAs offer the flexibility to accommodate late-stage design changes, which is very important in nowadays automobile development cycles (between one to two years). So finally, the best choice for this industry is FPGA-based systems.

Automotive Electronics Council developed AEC-Q100 standard for test qualification for integrated circuits used in critical applications, like safety systems (airbags, ABS, ESP, etc.). This includes examinations as a wire bond shear test, electrostatic discharge (ESD) test, latch-up test, non-volatile memory (NVM) data retention and endurance, operational life test, electro-thermally induced parasitic gate leakage, fault simulation, etc. For other non-critical applications, like environment control (air conditioning and so on) or infotainment, this level of quality and reliability is not requested.

Another area of deployment in the automobile sector is the so call “driver assistance”, which includes things like parking assistance, systems to warn from dangerous situations, like front collision, lane crossing and the like.

1.3.3 CIVIL AERONAUTICS

Civil aviation has been employing FPGAs for many years in aircraft and supporting systems. Applications include signal acquisition, signal tracking, and data demodulation functions for GPS positioning, surface surveillance, automatic transponder or aircraft identification systems, as well as safety and conflict management functions like detect-and-avoid systems.

US Federal Aviation Authority (FAA) recognized through AC 20-152 in 2005 that DO-254 Design Assurance Guidance for Airborne Electronic Hardware is a good way for development of complex electronic hardware for airborne systems. EUROCAE ED-80 is the equivalent in the European Union. DO-254 provides an official set of requirements for suppliers of civil aviation and avionics systems. Rather than specify how to implement requirements or what tests must be completed, it specifies a process for achieving design assurance and certification. The certification is for the hardware system, not for individual components. IEC 62566 is widely based in these standards.

The equivalent standard for software-based systems is DO-178B Software Considerations in Airborne Systems and Equipment Certification, being EUROCAE ED-12B the equivalent standard in the European Union.

1.3.4 MILITARY AND AEROSPACE

Military and aerospace manufacturers have been employing ASICs and FPGAs for many years, in areas like secure communications in real-time field information systems (where capability for signal processing and parallel computation is crucial) or unmanned drones, like unmanned aerial vehicles (UAVs), where light weight, low power, signal processing and encryption in real time are basic requirements. Another particular advantage that FPGAs offer is their ability to be reprogrammed on-the-fly.

High-altitude avionics and extraterrestrial applications employ radiation-hardened FPGAs to address single-event upsets, for example, antifuse technology. Nevertheless, a related issue arises when hardened anti-fuse FPGAs are used and the designers do not fully understand what the hardening provides and what it does not. For example, with an anti-fuse device typically the device configuration and the flip-flops are hardened, but the combinatorial logic is not. Data paths and clock inputs into each flip-flop are shared by all flip-flops, so they are all vulnerable and triplicate voting approaches cannot vote these glitches out. So SEUs can affect the logic and this must be dealt with in the overall design.

For harsh environments, testing of flash devices has identified some issues and research into their efficacy under extreme conditions is continuing. NASA Goddard Space Flight Center has used anti-fuse devices in the past for space applications that are mission critical. But they have also used SRAM devices with appropriate hardening for space missions and more so for terrestrial applications that are not mission critical. Jet Propulsion Laboratory has helped bring space-qualified SRAM-based technology to commercial availability. There are numerous SRAM-based FPGAs being used in harsh space environments. For example, the Mars Rovers relied on SRAM devices to control its descent to the Mars surface, including control of rockets and parachutes.

Regarding standards, the European Space Agency (ESA) supports the European Cooperation for Space Standardization (ECSS). ECSS publishes standards and other documents containing requirements, recommendations and information. In 2008 they updated ECSS-Q60-02 ASIC and FPGA Development, which defines requirements for development of digital, analog and mixed analog-digital custom designed integrated circuits, including ASICs and FPGAs.

Space industry offers a very good example of the need for rigorous design review for mission-critical applications [7]. In March 1999 the NASA Wide Field Infrared Explorer (WIRE) experiment was launched. The launch was a success. However, on the second orbital pass ground control discovered that the telescope aperture cover opened prematurely due to what was believed to be a system anomaly. This event caused the Goddard Space Flight Center mission to fail because the telescope could not perform its observations. A principal element in the circuitry controlling the assemblies was an FPGA. Design and testing of this circuitry was conducted by Utah State University Space Dynamics Laboratory (SDL). SDL and many other government contractors had successfully employed this particular FPGA in different roles for numerous mission-critical applications. However, after this incident occurred it was discovered that the FPGA was not well-suited to this application. Post-accident analyses uncovered unnecessary complexity built into the design, faulty architecture that allowed the FPGA to bypass two out of three required inhibits, and a power-up behaviour of the particular FPGA that places the device and its outputs in an indeterminate state for a few milliseconds on power-up. The indeterminate state condition only occurs after the device has been idle for at least one and a half to two hours. During testing, the device was never powered down for this length of time and only on actual launch did the condition occur. Additionally, personnel involved within the

project were unaware of the transitory power-up behaviour of the FPGA, although the manufacturer did document the power-up phenomenon in technical specifications on their web site, but it was overlooked by the design engineers and review team personnel. The SDL team reporting lessons learned stresses the importance of ensuring the necessary design expertise and the widest possible expert review for mission-critical FPGA designs.

1.4 EXPERIENCE WITH FIELD PROGRAMMABLE GATE ARRAY TECHNOLOGIES IN NUCLEAR POWER PLANTS

FPGAs have already been used to implement both safety and non-safety applications in NPPs in a number of plants worldwide. All different FPGA technologies have been used (SRAM, flash and antifuse).

FPGAs have been used for some time to perform ancillary functions within control and protection system equipment (e.g., implementing a data communication interface as part of a microprocessor-based system, where the microprocessor performs the primary control or protection function). When talking about FPGA-based systems we refer to the use of FPGAs in a primary role, performing the primary functions of the system.

Detailed examples of existing FPGA applications in NPPs are presented in the appendices of this study.

1.4.1 EXPERIENCES IN OPERATING PLANTS

1.4.1.1 UNITED STATES OF AMERICA

Hope Creek NPP has used field programmable logic arrays (FPLAs) since the early 1980's, and the plant continues to operate with them today. The FPLA-based system was part of the original design of the plant, a boiling water reactor (BWR). The FPLAs are used to perform component control logic (e.g., pump motor control, valve control, etc.), both safety and non- safety.

FPLAs are a predecessor technology to FPGAs. They are simpler devices, but like FPGAs they contain arrays of logic gates that can be interconnected through field programming. The FPLA-based I&C system is no longer supported by the original vendor, and the FPLA chip itself is now obsolete. However, the plant has been able to continue using the same chips through use of spares purchased originally and obtaining additional chips from various sources.

Nuclear Regulatory Commission approved the first safety application of an FPGA-based system in 2009 for the Main Steam and Feed Water Isolation System (MSFIS) at Wolf Creek NPP. The system is based on the ALS^{TM} (Advanced Logic System) platform from CSI (Control System Innovations). CSI was bought by Westinghouse Electric Company in 2009 and now is a Westinghouse subsidiary company.

As there are no specific regulatory documents nowadays in the US for FPGA-based applications, the Safety Evaluation Report (SER) for Wolf Creek application is a very good reference both for suppliers and utilities [5].

There is growing interest among other US utilities in the use of FPGAs for upgrading obsolete or aged safety I&C systems. Wolf Creek NPP is planning to replace the core thermocouple and core cooling monitoring system (TC/CCM) using ALS^{TM} platform and continue to use this platform for other systems, like for Diesel Load Sequencer (DLS) or Solid State Protection System (SSPS).

Previous Westinghouse experiences can also be cited, like the new-design CPLD-based boards for its Solid State Protection System (the voting logic system that is part of the Reactor Protection System), or the DADS (DRPI Advanced Display System) in the case of non-safety systems. Both of them have been briefly commented previously.

Diablo Canyon DAS Diverse Actuation System as part of its Eagle21 replacement project is another undergoing project in the USA involving FPGAs, as discussed in 1.2.1.4.

Comanche Peak NPP is also developing, along with Westinghouse as the supplier, a new Diesel Load Sequencer based on CSI ALS^{TM} platform. Nevertheless, the project has been hold at this moment. It is noteworthy that originally the selected platform was Common-Q system (a microprocessor-based system pre-licensed in the US by Combustion Engineering), and it was the supplier who recommended Comanche Peak to move to the new ALS^{TM} platform, after CSI got the approval for the Wolf Creek NPP MSFIS replacement project from the Nuclear Regulatory Commission.

At the beginning of this year, CSI and Westinghouse have received approval for their topical report for the generic licensing of the ALS^{TM} platform [6]. This is the first kind of approval by the NRC for an FPGA-based safety platform.

1.4.1.2 CANADA

FPGAs have been used in several Canadian Deuterium Uranium (CANDU) reactors in Canada since late 1990s. One early application was and FPGA-based emulator for the PDP-11 computer that were used in non-safety systems.

An updated version of the emulator has been in Digital Control Computers (DCCs) for reactor control functions.

Safety applications of FPGAs also are being considered, now that experience has been gained with the technology in applications that are not safety-critical. For example, application of FPGAs in the safety shutdown system (SDS) for operating CANDU reactors is currently being studied [8] [9].

1.4.1.3 FRANCE

Electricité de France (EDF) began replacing the rod control system (RCS) and reactor in-core measurement system (RIC), which are non-safety systems, in their 900 MW series of plants (34

units) in 2009. Project plan is for 10 years. New system makes use of FPGAs for performing control and interface functions.

For RCS, the use of FPGAs provides millisecond time resolution needed for this application. This would be impossible with microprocessor based systems. 850,000 gates are used to control up to 39 functions [10].

Both RCS and RIC systems have been successfully installed at Tricastin Unit 1 NPP. Second and third installations are in progress at Fessenheim Unit 1 and Bugey Unit 2.

EDF is also planning to use FPGAs in safety-critical applications. One project is developing an FPGA-based emulator for the Motorola 6800 microprocessor, currently used in safety systems in French plants and now obsolete. This way, qualified software that implements the reactor protection functions can be retained [11] [12].

1.4.1.4 SWEDEN

FPGAs have been used in the Ringhals modernization project developed by Westinghouse (Ringhals Two Instrumentation and Control Exchange, or TWICE project). They are used in the Component Interface Module (CIM), which acts as the interface between primary safety actuation system (based on Common-Q^{TM} microprocessor based platform) and the actuated equipment, as well as between Diverse Actuation System and operator commands. CIM incorporates priority logic that ensures appropriate signal is passed to each component, and that component is placed in safe state in case of conflict.

1.4.1.5 CZECH REPUBLIC

From the middle of 1990s to 2000 FPGA-based systems have been used in the Czech Republic in Temelin NPPs. They are mainly used in Non-Programmable Logic (NPL), which has several functions. One of them is as safety load interface. Temelin has a Primary Reactor Protection System (PRPS) and a Diverse Protection System (DPS), both of them are micro-processor based. NPL is in charge of priority logic arbitration between the protection systems and ensures that loads go to a safe state in case of discrepancy. Another one is the Diesel Load Sequencer.

1.4.1.6 EASTERN EUROPE

FPGA-based systems have been installed in safety applications in Ukraine and Bulgaria, using the RPC platform from Raidy Company. First installation took place in 2003 in Ukraine, and a total of 17 plants in Ukraine and Bulgaria have installed Radiy FPGA-based systems for safety applications, including Reactor Trip Systems (RTS) and Engineered Actuation Systems (ESFAS).

For providing D&DiD, two diverse sets of equipment, both FPGA-based but using different ICs from different vendors, have been used for RTS systems.

1.4.1.7 JAPAN

Toshiba began using FPGA-based equipment for non-safety radiation monitoring systems in Japanese BWRs in 2003. The first safety-related radiation monitoring system employing FPGAs was in 2004. Toshiba installed an FPGA-based power range neutron monitoring systems (PRNMS), which is a safety system, in an Advanced Boiling Water Reactor (ABWR) in 2007. As of late 2008, Tokyo Electric Power Company (TEPCO) reported that over 200 FPGA-based radiation monitoring and neutron monitoring systems were in use in TEPCO's plants [13].

Toshiba has also developed a Reactor Trip and Isolation System (RTIS) and Startup Range Neutron Monitoring System (SRNMS) based on the same technology.

1.4.1.8 SOUTH KOREA

CPLDs are used in digital safety systems of operating Korean nuclear plants for performing ancillary functions such as system initialization, bus interface, I/O card control, memory control and peripheral channel control. FPGAs are also used to perform component interface functions for the engineered safety features (ESF) in new APR-1400 plants under construction in Korea [14] [15].

1.4.2 EXPERIENCES IN NEW BUILDS

As new builds are already under construction in countries like US, China, Finland or France, and many others are in the planning or design phase, the following information could suffer significant changes. Nevertheless, current future applications of FPGAs in new builds include:

- Diverse systems

  The most significant example is the Diverse Actuation System (DAS) for the Westinghouse's AP1000™ design. AP1000™ safety platform is based on Common-Q (Common Qualified), which is a safety platform already approved by NRC for safety applications. Selected control platform is OVATION™ distributed control system from Emerson Water and Power Solutions (Westinghouse awarded a strategic agreement for supplying OVATION™ systems to the nuclear industry). Because of current US regulation, a DAS is required in case safety platform is based on microprocessors and runtime software (operating system, application software and so on). Selected DAS

platform is ALS™ from Control System Innovations, Inc. (now, a Westinghouse company).[1]

- Priority logic

  Priority logic modules ensure that safety signals have priority over non-safety signals in actuating or controlling safety equipment. The AP1000™ and the Mitsubishi US-APWRTM (Mitsubishi Electric Total Advanced Control or MELTAC™ platform) both use FPGAs to implement priority logic. Triconex (part of Invensys Process Systems) is also employing FPGAs for the priority logic modules in new plants being built in China using Tricon™ platform.

- Primary safety systems

  Some new plants are going to use FPGA-based platform to implement primary safety functions, like the ABWRs planned for the South Texas Project, which will use Toshiba's FPGA-based platform to implement the safety-related neutron monitoring systems (NMS) and reactor trip and isolation system (RTIS). Atomic Energy of Canada Limited (AECL) is also evaluating the potential application of FPGAs in safety systems for the Advanced CANDU Reactor, ACR-1000™.

Other types of reactors are going to use PLDs, like the Enhanced Simplified Boiling Water Reactor (ESBWR™), of GE-Hitachi, in a proposed independent control platform for implementing Anticipated Transient Without Scram (ATWS) and several isolation functions. The European Pressurized Reactor (EPR™), of Areva, is also making use of programmable logic technology in the Priority Actuation and Control System (PACS).

[1] Information from the ICWG (Instrumentation and Control Working Group) of the PWROG (Pressurized Water Reactor Owners Group) held during July 2011 and July 2012 suggests that Westinghouse Electric Company (WEC) is considering to use already developed product for operating plants in support of life cycle management, like 7300 control and protection system PCBs and CPLD-based SSPS (Solid State Protection System) PCBs, in new builds.

2 PLANNIFICATION AND DESIGN OF MODIFICATIONS INVOLVING FIELD PROGRAMMABLE GATE ARRAYS

2.1 TYPES OF MODIFICATIONS INVOLVING FIELD PROGRAMMABLE GATE ARRAYS

2.1.1 REPLACEMENT OF RELATIVELY SIMPLE LOGIC CIRCUITS OR COMPONENTS

FPGAs can be used to replace obsolete discrete electronic components on circuit boards, allowing a change to be made that targets the specific components that need to be replaced, or entire circuit boards with one or more new boards using FPGAs to implement the same functionality. Obsolete analog circuits cards can be replaced in this way. Banks of relays also can be replaced by a new circuit card with one or more FPGAs. These types of changes often reduce the number of components and associated interconnections, potentially improving reliability, eliminating single point vulnerabilities (SPVs) an at a reduced cost and time for installation.

Some examples of replacements have been previously commented and/or are described in the appendixes.

2.1.2 REPLACEMENT OF COMPLEX DIGITAL CIRCUITS INCLUDING MICROPROCESSORS

FPGAs also can be used to replace more complex digital circuits, including microprocessors, microcontrollers, and other complex digital devices. Microprocessor-based equipment that was installed in I&C systems years ago is becoming obsolete, particularly the microprocessor itself. An FPGA can be used to emulate the obsolete microprocessor, replacing the hardware circuit but allowing the legacy software to be retained. For safety applications and other applications important to plant operation, significant resources were invested in developing and qualifying the original software. Moving to a new microprocessor would require an additional significant investment to modify and re-qualify the software for the new processor.

Using an FPGA-based hardware emulator also has advantages over use of a software-based emulator – the latter would require a new hardware platform, operating system and microprocessor emulation software, as well as the application software. Use of an FPGA emulator eliminates one layer of complexity. Also, if the emulator is designed for portability, this

provides longer-term protection against future obsolescence of the integrated circuit (improved lifecycle management).

At the same time, it should be noted that a microprocessor emulator is a complex FPGA application to develop, verify, and justify to a regulator. This needs to be considered along with the factors discussed above when deciding what type of solution to employ to address obsolete complex digital circuits.

2.1.3 SYSTEM-LEVEL REPLACEMENTS

FPGA-based solutions also can be used as full system replacements, with the FPGAs performing the control and protection functions directly. Compared to microprocessor-based solutions, FPGAs can provide a solution with significantly less complexity. This is particularly true if the primary system functions are implemented in flat hardware logic, that is, by interconnecting fundamental logic blocks to perform the functions in parallel using the FPGA's native logic directly, without use of IP cores or embedded processors.

2.2 CHARACTERISTICS OF APPLICATIONS THAT ARE SUITABLE FOR FIELD PROGRAMMABLE GATE ARRAY BASED SOLUTIONS

FPGAs are particularly well suited to performing combinatorial and sequential digital logic functions. FPGAs are also widely used to handle digital data communication interfaces, even in cases where they do not perform the primary control functions.

They also can be used for continuous control functions. In the past, FPGAs were limited to processing digital inputs and producing digital outputs – separate circuits were required to handle analog signals. However, mixed-signal FPGAs are now available that can handle both analog and digital inputs and outputs, with the analog-to-digital converters built in.

FPGAs can provide relatively simple human-system interfaces, such as panel interfaces for maintenance or troubleshooting. However, more complex and full-featured HSIs such as those used by control room operators to monitor and control plant systems may not be suitable for FPGA-based solutions, or alternative solutions should be provided independently to address this feature. The PC-node box from Westinghouse within the ALS™ platform is a good example of this.

Each potential application should be evaluated considering the advantages that can be offered by an FPGA-based solution, as well as the limitations or disadvantages. Criticality of the application also should be considered. For example, for safety-critical functions an FPGA might be used to implement the primary functions in flat hardware logic to keep the implementation

simple and more easily verified, and to simplify the safety justification. But other less-critical functions such as diagnostic functions, simple HSIs and other ancillary functions that do not affect the safety functions might be implemented in a more complex FPGA solution, potentially including an embedded microprocessor if required. However, it must be remembered that some of the advantages of FPGAs can be lost as more complexity is introduced and the application moves away from a flat hardware implementation.

2.3 PLANNING AND CONCEPTUAL DESIGN OF A MODIFICATION INVOLVING FIELD PROGRAMMABLE GATE ARRAYS

Following guidance is provided related to planning and developing preliminary or conceptual designs of modifications involving FPGAs, focussing on FPGA-specific aspects.

2.3.1 MODIFICATION DESIGN

KISS philosophy (Keep It Simple Stupid) is a base rule for any good design, and simplicity is especially important for safety-critical applications. The simpler the design, the more predictable, the more reliable, the more testable, the cheaper, the easier to licensed, etc., the system would be.

In order to achieve simple design, the following must be considered:

- If a function does not have to be safety-related, keep it out of the safety-related portions of the design if possible – the application will be simpler and the safety justification easier.
- Consider segmenting the design to keep individual functions separate – this simplifies the design and can increase reliability, verifiability and testability. It also can help in obtaining regulatory approval.
- Keep ancillary functions (e.g., self-testing, diagnostics, and data communications to other devices) separate and independent from the processing of the primary functions.
- When dealing with obsolete microprocessor-based equipment, consider the options available and their relative complexities. Replacing the microprocessor with an FPGA-based emulator allows re-use of the existing software, preserving that investment, but it maintains complexity in the solution. If the I&C functions performed by the equipment could just as easily be performed directly in a flat hardware logic solution in an FPGA, this should be given serious consideration. It could be a much simpler solution to implement, qualify if required, and maintain. Additionally, it should be taken into account that regulators also take advantage of utilities upgrades to update the licensing basis of

a plant, and things that were approved in the past may not comply with newer and applicable regulation.

Other design considerations to be taken into account could be:

- Define testing requirements. This requirement could come from regulation (like periodic surveillance tests required by Technical Specifications) or be something promoted from regulator or operation or maintenance staff. Additionally, one of the goals could be to reduce the burden of periodic testing, making used of built-in self-testing (BIST) technologies or any other external testing capabilities and automated test equipment. Self-test features add complexity to a system, but they also can improve reliability and reduce periodic testing burdens. Keeping the self-test functions separate and independent from the safety function processing is a good design practice that must be followed. Additionally, licensing work not only for the system itself but for possible changes in licensed documents like Technical Specification should be foreseen.
- Look at the impact of implementing an FPGA-based solution on power supply loadings and heat load. Moving to FPGAs should significantly reduce power requirements and heating, ventilation and air conditioning (HVAC) loads in previous installations, as FPGA-based equipment tends to use less power and generate less heat than both microprocessor-based systems and conventional analog or discrete digital logic systems.

2.3.2 ACCEPTANCE AND QUALIFICATION TESTS AND ANALYSES

Planning for the implementation of FPGA-based equipment should include plans for acceptance testing, including factory acceptance tests (FAT) and site acceptance tests (SAT).

Plans also should be made for qualification testing for equipment to be used in safety applications. Qualification tests include response time testing if applicable and environmental qualification testing. Environmental qualification includes[1]:

- Temperature.
- Humidity.
- Vibration.
- Radiation (in case of harsh environment).
- Electromagnetic compatibility (EMC), which includes emissions, susceptibility, surge withstand and electrostatic discharge (ESD) withstand.

[1] In the case of USA (NRC) requirements, refer to RG 1.89 (IEEE 323) for environmental qualification, RG 1.100 (IEEE 344) for seismic qualification, RG 1.180 (EPRI 102323 and others) for EMC, RG 1.75 (IEEE 384) for isolation between safety and non-safety equipment, etc.

- Seismic.

Isolation between safety and non-safety equipment must also comply with regulatory requirement.

This study did not identify any significant differences in how any of the acceptance or qualification tests are performed when using FPGA-based equipment versus other types of I&C equipment. Current standards and guidelines can be applied in planning and running these tests. The same can be said for analyses typically performed for I&C systems that are safety-related or critical to plant operation, including failure modes and effects analyses (FMEA) and reliability analyses. The details of the analyses and the results (e.g., specific failure modes, reliability values, etc.) that are obtained may be different, but the basic approach and guidelines to follow are the same as for other I&C equipment.

Quality assurance requirement according to those required by current regulation should also be taken into account (for instance, 10CFR50 Appendix B or ASME NQA-1a, in the case of US).

2.3.3 LICENSING PLAN

For safety applications, the plan for gaining regulatory approval of the modification should be considered early in the planning and conceptual design process. Determine whether there are FPGA-based systems or equipment that could meet the requirements of the modification and have already been reviewed and accepted by the regulatory authority. Evaluate the differences between the planned application and the bounds of the previous regulatory approval, and how those will be addressed.

Determine what strategy will be used for addressing potential common cause failures and the need for any diversity within the system or in the form of a separate diverse backup. In the US, this should be considered as part of the overall diversity and defense-in-depth (D&DiD) evaluation for the safety I&C systems. Such an evaluation is typically performed in accordance with NUREG-0800 Branch Technical Position 7-19 [16] and the NRC Interim Staff Guidance document DI&C-ISG-02 [17]. Additional important reference for D&DiD are NUREG CR-6303 Method for Performing Diversity and Defense-in-Depth Analyses of Reactor Protection Systems [18], EPRI TR 1002835 Guidelines for Performing Defense-in-Depth and Diversity Assessments for Digital Upgrades: Applying Risk-Informed and Deterministic Methods, and NUREG CR-7007 Diversity Strategies for Nuclear Power Plant Instrumentation and Control Systems [19], this last one with very detailed examples of implementation of D&DiD strategies in Darlington, Lungmen, Kasiwazaki-Kariwa, Sizewell B, Temelin or Olkiluoto-3 nuclear power plants.

In the US, EPRI Technical Report 102348 (NEI 01-01) Guidelines on Licensing Digital Upgrades has been used for long time as the main reference for the licensing process. EPRI Technical Report 104600 Proceedings: Licensing Digital Upgrades in Nuclear Power Plants, can also greatly help in driving the process.

In Spain, there are few regulatory requirements or guidance, and digital projects follow the regulation from the country of original reactor supplier. Nevertheless, efforts have been made inside Spanish nuclear sector, and UNESA CEN-6 (Guía para la Implantación de Sistemas Digitales en Centrales Nucleares) was issued and received approval from Consejo de Seguridad Nuclear.

2.3.4 LIFECYCLE SUPPORT

Plans for supporting the system over its expected lifetime should be considered, including plans regarding how maintenance and repairs will be handled, and the plant's strategy regarding the ability to make changes to the application design should they be required in the future. Protection against future obsolescence is a very important related issue to be considered that impacts from the very beginning of the modification process.

The following should be considered regarding maintenance and repairs over the lifetime of the equipment:

- Determine what the expected service lifetime of the equipment will be. With FPGAs, longer lifetimes can be achieved as compared to microprocessor-based systems, with 20-25 years being a typical goal for some previous installations, more like the original analog systems first used in nuclear plant I&C systems.
- Consider whether board-level repairs will be attempted by site personnel, or if the strategy will be board replacement and return to the supplier. Modern circuit board component mounting technologies can make board repairs difficult, as they are usually multilayer and surface-mount (SMT). For example, ball grid array (BGA) mounting requires special instrumentation to support IC mounting and inspection of solder joints.

  International recognized standards, such as IPC[1] 7711/7721, or JEDEC[2] standards, should be identified and put in place if repairs are going to be done at plant.
- Maintenance strategy and protection against obsolescence will drive the strategy regarding stocking of spares. This is also an issue that should be taken into account from the very beginning of the modification process, and must determine selection and negotiation with the supplier.

Longer lifetimes may be achievable with FPGA-based systems than with microprocessor-based solutions. Therefore, the potential need for modifications to the system or components should be considered. For safety systems, modifications are typically very infrequent and require

[1] IPC stands for Association Connecting Electronics Industries, former Institute of Printed Circuits.

[2] JEDEC stands for Joint Electron Device Engineering Council.

support from the supplier to make the changes and re-qualify the system/equipment as required. However, the utility may want to have some protection against potential loss of support from the supplier in the future, so that the system can continue to be maintained and modified if necessary.

For less critical applications, changes may be expected more frequently or the plant may simply want to have the ability to make changes when required.

Also, it is a good idea to plan ahead for future obsolescence – how the design could be ported to a newer, available FPGA circuit in case the original one is no longer supported.

Considerations during planning regarding these issues should include:

- In-house capability to modify the FPGA application if needed or desired or rely on supplier for maintaining the necessary tools and be provided to support doing this later if required. An approach taken by some utilities is to ask that all design documentation including HDL, tools, development environment, etc. be kept by supplier and in case it goes out of business or stops supporting the equipment, all tools and documentation be provided to the customer.
- It should be recognized that if the utility takes on the responsibility to maintain the design and make any required modifications, this requires taking on the management of the toolset and test benches (software and hardware), configuration control, decisions on whether and when to update tools as the circuit vendor issues new versions or, even worst, patches for fixing gaps, verification and validation of changes, and maintaining the safety justification or licensing approval. Also, considerations should be taken on what expertise would be required to do this successfully. Achieving a reliable and safe design requires more than just HDL programming, but electronic hardware design expertise.

RTL portability is a great advantage on the use of FPGA. This portability should be assured during planning and conceptual design and be based on trusted and fully standard-compliant tools.

2.3.5 IMPACT ON OPERATION AND TRAINING

When making changes that involve a new I&C technology, like in the case of FPGA, consideration on potential impact on plant engineering, operations and maintenance staff should be taken into account:

- Determination of what the relative roles of the utility and supplier will be in the design development, verification and validation, installation and testing:
    - If utility act as an independent reviewer during design development and testing, or just provide oversight.

- If utility has or not to develop appropriate test cases for verification and validation.
- If outside contractors, like independent third parties, may be required to support in different activities.

- Consideration on the potential impact on engineering and maintenance staff, the skills they will require and training that may be needed to be able to support the new system or equipment. Moving from older analog technology to FPGA-based equipment can reduce maintenance burdens and lessen the need for the utility to maintain skills related to very old technologies. However, some level of knowledge and skill related to FPGAs may be needed in order to monitor, troubleshoot and maintain the new FPGA-based equipment. Maintenance procedures should be revised in advance to the implementation of the new system (this should include procedures for changing configuration parameters, calibration, periodic testing, etc.).
- The modification could potentially have operational impact as well – for example, different failure modes, different behavior on power loss and restoration, and/or differences in error-reporting, alarms and alarm response. Planning of the modification should consider whether the plant has specific goals or requirements regarding operational impact of the change that need to be reflected in the modification requirements and conceptual design. Operation procedures should be revised in advance to the implementation of the new system.
- Consideration on the impact on the plant training simulator and the method of implementing the new FPGA-based system in the simulator. Different simulation techniques could be used, like stimulated simulation. Use of simulation for staff training, mainly operation staff, is something recommended that should be evaluated.
- Consider use of the simulator models to support engineering design and evaluation, particularly for control systems where controls tuning and performance need to be evaluated with dynamic plant models, and for HSIs where operator input and human factors engineering (HFE) evaluations can greatly benefit from early simulation [20].

2.4 SPECIFICATION AND EVALUATION OF FIELD PROGRAMMABLE GATE ARRAY BASED SYSTEMS

Following guidance is intended to cover FPGA-specific aspects. The guidance can be used as appropriate by a utility to specify information that a prospective vendor should provide as part of a proposal, and to identify items that should be considered when evaluating a proposed system.

2.4.1 SELECTION OF THE FIELD PROGRAMMABLE GATE ARRAY CIRCUIT

The following are items to consider related to the specific choice of FPGA circuit:

- Evaluate the choice of FPGA memory technology (SRAM, flash or antifuse). All three types of FPGAs have already been applied in the nuclear industry. Each has its own set of issues and tradeoffs, but for safety-critical application, where changes are less frequent, antifuse technology should be considered. Check what measures have been taken to deal with issues specific to the FPGA technology that has been deployed for this system.
    - For SRAM: measures taken to protect against or tolerate SEUs; measures to ensure security of the programming; measures to protect against or deal with momentary power losses or glitches, etc.
    - For flash: measures taken to protect against or tolerate SEUs (flash is considered less vulnerable than SRAM, but more so than antifuse); assurances of long-term reliability of flash memory (there is less information available on flash than there is for SRAM), etc.
    - For antifuse: measures taken to ensure longevity of the connections, and data to support this, etc.

  Quantifiable data to address the issues with whatever circuit technology has been chosen will provide the greatest assurance and will make justification for regulator easier.

- Consider the experience that has been gained with the particular integrated circuit, including the amount of operating experience that has been gained and how relevant it is to the planned nuclear plant application, considering the environment in which it was used, the type of application and its criticality. Also consider experience with the related toolsets.

  Experience from other similar nuclear application, if it exists, would be a good starting point, mainly in the case of safety systems (and so qualified). It is supposed that this work was properly done, properly documented, etc. Nevertheless, if it was long time ago, caution should be taken; confirmation of availability should be the first step, followed by assurance of same regulatory framework for qualification, etc.

- Determine what commitment has been made by the IC vendor to support the chip, like how long will support be provided, where else is the circuit used or existence of any commitments been made for support to the military, aerospace, or other industries such as aviation, medical or automotive. Those industries have been making use of FPGAs for years in safety-critical or mission-critical applications, and some circuit vendors have

in the past provided assurances of continued long-term support for specific models of FPGAs.

- Examine the quality control and security measures employed by the circuit vendor to ensure high quality and integrity of the circuit, including how the vendor ensures security of the design through the entire chain of custody including the foundry (often separate from the circuit supplier). Security of hardware devices, i.e., protection against unwanted or malicious design elements being introduced into the hardware at the factory, is an issue that needs to be addressed for safety-critical applications.

  In the case of safety-critical applications of FPGAs, compliance to quality assurance regulation, cybersecurity regulation and so on is a must.

- Determine how the utility would be notified of any errors or critical issues that may be found by the circuit vendor or other users of the circuit or toolsets after the modification has been installed in the plant. This is something clear for qualified equipment suppliers, subject to, for instance in the case of USA, Title 10 of Code of Federal Regulations, part 21 notifications, but not so clear for non-qualified equipment suppliers for FPGAs to be used in control or information systems.

2.4.2 DESIGN

The following are items to consider related to system design:

- Simplicity of the design, particularly for the primary I&C functions:
  - Ancillary, more complex functions are kept independent from the primary I&C function processing and critical signal paths.
  - Different functions are segmented to keep different parts of the design relatively simple.
  - No IP cores are used that may have hidden complexity.
  - Building block approach is used, with the application made out of simple logic blocks.
- Determinism of the design:
  - Use of finite state machine approach with all states well-defined.
  - Use of synchronous design principles.
  - Etc.
- Hardware configuration management:
  - Aspects of the hardware configuration that are automatically checked and verified/alarmed (power supply in case of redundancy, anchored of PCBs, etc.).

- Restrictions affecting maintenance or replacement of boards (e.g., hot swappable).
- Etc.

- Provisions for troubleshooting and maintenance:
  - BIST features.
  - External testing of circuit boards (accessibility of test points, JTAG ports, etc.).
  - Need for special test equipment, and coverage of built-in self-testing and diagnostics.
- Cybersecurity provisions that protect against unauthorized or inadvertent alterations in the application or data, particularly if SRAM technology is used.
- Evaluation on how potential common-cause failures (CCFs) are addressed:
  - The system includes any internal diversity, like redundant FPGA with comparison and voting algorithms, or redundant logic in the same FPGA, etc.
  - External solutions available for implementing diversity, like DAS.
  - Types and levels of diversity.
  - Used of diverse development tools.
  - Used of diverse test tools.
  - Different design teams.
  - Test tools independent of and diverse from design tools.
  - Etc.

2.4.3 DEVELOPMENT PROCESS

The following are items to consider related to the application development and V&V processes:

- Evaluate the design and V&V process:
  - Standards used to guide the process.
  - In safety application, compliance with standards and guidelines used by the regulator in reviewing and approving FPGA-based systems.
- Formal methods or techniques employed as part of the process. The use of formal methods can reduce dependence on V&V activities, build confidence in the system's integrity, and assist in gaining regulatory approval.
- Independence of V&V for safety-critical or high-reliability applications. Utility or a recognize third party can act as an additional independent reviewer.

- Evaluation of all software tools used in development and testing of the circuit:
  - Well-established and controlled tools.
  - Maintenance and configuration control of the tools by vendor (e.g. assurance that only the required programming is placed onto the programmable device, and that the programmable device is clean before any application programming is placed upon it).
  - Control upgrades and patches for the tools by vendor.
- Review of the method used to translate the application functional requirements into the design. In case a high-level software tool (application-oriented language) is used, HDL could be auto-generated by the tool.
  - Review against coding guidelines.
  - Modifications made to the code (this can be a plus if it brings the code more in compliance with coding rules, but a minus in the sense that it can introduce human error).
  - Avoidance of constructs that could cause later portability problems.
  - Verification that the code matches the functional requirements in case HDL is written directly.
  - Obtaining functional diagrams derived from the HDL that can be compared to the original requirements.
- In case IP cores are used:
  - Method for verification.
  - If it is a soft IP core, HDL independence verification is recommended.
- Assessment of level of expertise on the design team and V&V teams:
  - Existence of personnel with electronic design expertise and deep knowledge of the particular FPGA circuit used.
  - Performance of critical design reviews addressing electronic design issues (e.g., potential for timing glitches, impact on power and grounds of multiple signals changing state simultaneously, clock skew, etc.).

2.4.4 SUPPORT

The following are items to consider related to the long-term support of the new equipment:

- Evaluate future sufficient commercial support for the system and the development environment and tools over the expected lifetime of the system. Evaluate the circuit

vendor's practice regarding modification of the implementation tools and maintaining compatibility with applications developing using previous versions. In particular, look at any new optimizations provided and whether those can be turned off if necessary.

- Determination whether the supplier will provide or at least escrow the entire toolset used for application development, V&V, simulation, files of constraints or directives used in the implementation, etc., including all software and any specialized hardware needed for application development, modification and testing.

Assurance that the HDL representation of the design is either provided to the utility or placed in escrow to support future modification or porting to another type of FPGA if necessary. This should also include documentation of the constraints imposed on HDL programming by the peculiarities of the particular FPGA circuit. IP cores or libraries should also be adequately captured for future modification or porting, in case they are going to be used.

THIS PAGE INTENTIONALLY LEFT BLANK.

3 DESIGN GUIDELINES

3.1 SELECTION OF FIELD PROGRAMMALBE GATE ARRAY CIRCUIT

When selecting circuits suitable for NPP I&C applications, and mainly in case of safety or critical applications, the following factors should be considered.

3.1.1 MEMORY TECHNOLOGY USED

FPGA circuits have two main types of memory: memory that retains the electronic design of the FPGA application, and memory for application data. Application design memory is in principle not modified during operation, whereas data memory may be.

The technology used for application design memory can have a significant impact on the circuit's susceptibility to random alterations caused by single-event upsets. Unwarranted alterations of a single bit in the design may have unpredictable effects on the behavior of the circuit, including on its self-monitoring and fault-tolerance capabilities. Therefore, it is recommended that immunity to or protection against such changes be given serious consideration when selecting the circuit and related design features. In general, non-volatile technologies tend to be more immune to random alterations, but it is recommended that circuit vendors be required to provide quantified susceptibility levels, preferably based on test data and/or operating experience.

In case of safety applications with no foreseen changes, antifuse technology should be the choice.

Also, with current technologies, data frequently modified by the application needs to be stored in volatile memory that can be of different technology than the design memory. What needs to be considered here is the amount of necessary application memory, and where this memory can be located: in CLBs (very small amounts), in memory blocks internal to the circuit, or in external circuits (with appropriate consideration of possible bottleneck issues in the communication between the FPGA circuit and the memory circuits).

Additional issue to be considered is the memory retention period of the technology. It depends on the technology and the manufacturer, but for flash-based technology it usually varies between 10 and 20 years. Nevertheless, this information comes normally from generic FPGA manufacturer information, and is based on a certain reprogrammable cycles that, in the case of nuclear sector, and mainly for safety applications, is very difficult to reach that number of cycles, so probably a more realistic memory retention time should be higher.

It should be noted that, even for antifuse technology, manufacturers define a memory retention period, as they cannot guarantee that fused connections will last forever with the appropriate characteristics.

3.1.2 FEATURE SIZE

A general trend in electronic circuit design is the seemingly never-ending decrease in the size of elementary design features or components (transistors and interconnects). Circuit designers often use the expression deep sub-micron technologies to refer to sizes that are currently used (typically 32 or 45 nm, with even smaller sizes coming in the near future; nowadays technology reach the 20 to 18 nm). Such miniaturization has many benefits including lower power consumption per gate and faster switching times.

However, smaller feature sizes also present weaknesses that need to be taken into consideration. In particular, deep sub-micron technologies are not necessarily optimal for NPP I&C and critical industrial applications. For example, they are increasingly subject to electro-migration, a phenomenon where high electric current densities in ultra-thin connecting wires displace metal atoms and ultimately rupture the wires. While this may be acceptable for mass-market products with short life expectancy (typically a few years for a commercial device), this is not acceptable for applications demanding high reliability or decades-long life-cycles such as NPP I&C systems.

Also, the small electronic features and ultra-low operating voltages and currents of deep sub-micron technologies tend to be affected more frequently and more severely by heavy ion radiation. Whereas ions may not affect older, coarser technologies, or affect only one transistor at a time, they can more easily affect ultra-thin features and ultra-low voltages, and can affect multiple transistors simultaneously, jeopardizing error detection and correction mechanisms.

Fortunately, a number of FPGA circuit vendors offer products explicitly aimed at industrial and safety-critical markets, featuring coarser and more robust technologies (typically in the 90 to150 nm range) that can operate in harsher environmental conditions and with longer lifetimes.

Further, it is best to employ one or more small FPGAs with low capacity rather than a single, large-capacity device. As a rule of thumb, if the FPGA occupancy is below 30%, the FPGA is too big for the application, and if the occupancy is over 80%, the FPGA is too small for the application.

Considerations over the use of CPLDs instead of FPGAs, and their benefits, should be evaluated, as for safety applications CPLDs could be more suitable (more suitable for combinatorial logic, faster response times, etc.).

3.1.3 CIRCUIT ARCHITECTURE AND EMBEDDED FUNCTIONALITY

Although most FPGA circuits share a number of common features, there are many possible differences in the architectures such as memory (types, quantity, size and location), routing channels and the number of inputs to LUTs. In addition to these differences, an evolving differentiation may be occurring with some FPGAs offering specific built-in capabilities, such as configurable microprocessor cores, adjustable cache sizes, multipliers, dividers, floating point operations, accelerators, or analog-to-digital conversion. These functional capabilities may be provided in the form of IP cores and are often targeted at specific markets.

The selection of the appropriate circuit architecture (including the choice between FPGA and CPLD, and between no IP and hard or soft IP cores) should be based on the overall system design and on the requirements that this design imposes on the component being considered.

For applications important to safety, one also needs to take into consideration the safety assessment of such built-in capabilities. In particular, one should make sure that sufficient information on the circuit is available regarding design, verification, manufacturing and operating experience to perform the assessment.

3.1.4 CIRCUIT PERFORMANCE AND CAPABILITIES

Considering the functional and response time requirements of most I&C functions, which are often expressed in terms of milliseconds or tens of milliseconds, circuit time performance and capabilities (e.g., in terms of numbers of basic elements) usually rank lower in the priority list, as most vendors have products that easily match such requirements, at least for the primary control and protection functions. Some ancillary functions, for example self-testing and diagnostics or provision of a simple human-system interface for maintenance purposes, may demand more in terms of logic capacity and capabilities from the circuit. This should be evaluated when selecting circuits for particular functions.

3.1.5 DESIGN FOR TESTABILITY

Design for testability (DFT) is a name for design techniques that add certain testability features to a design, to make it easier to develop and apply off-line tests during manufacturing or application development. The purpose is to ensure that the manufactured and configured circuit complies with the verified and validated design, and in particular that all specified logic gates are present, connected, and operating correctly.

DFT often is associated with design features that provide improved access to internal circuit elements such that the local internal state can be controlled (controllability) and/or observed (observability) more easily.

Boundary scan technologies, such as JTAG (Joint Test Action Group, though an association, is the common name used for the IEEE 1149.1 standard entitled *Standard Test Access Port and Boundary-Scan Architecture*), although it was originally designed for testing printed circuit board assemblies, is also used for accessing internal elements of integrated circuits (including FPGAs and CPLDs) and to provide DFT capability.

3.1.6 LONG-TERM SUPPORT

Life expectancy of NPP I&C equipment is measured in decades. It is thus necessary to have appropriate guarantees from the circuit vendor that maintenance will be possible over such long time periods. Utilities can of course acquire enough spares to cover the expected lifetime, incurring in high financial cost due to the increase in capital assets. So it is often preferable to rely on vendors that already have long term commitments (possibly with other industries such as aerospace or the military) regarding the products that are selected.

It is also important that the circuit vendor provides appropriate guarantees that the plant operator will be informed in a timely manner of any changes in the design or manufacturing of the circuit (including discontinuation of circuit manufacturing), as well as keep informed on defect or problems in the products on other industries or similar applications. Nuclear qualified suppliers in the US, for instance, are subject to Title 10 of Code of Federal Regulations, part 21, on the reporting of failures and defects.

3.1.7 SOFTWARE TOOLS

The use of appropriate software tools can increase the integrity of the FPGA application design, and hence the reliability of the final circuit and of the system, by reducing the risk of introducing faults in the design that remain undetected. The use of tools can also have economic benefits as they can reduce the time and the effort required. For example, tools can be used to automatically check for adherence to design rules and standards, to generate proper records and consistent documentation in standard formats, and to support change control. Tools can reduce the effort required for testing and to maintain automated logs. Thus, although software tools may not be the primary selection criterion for FPGA circuits, they are very important and are considered by circuit vendors as a significant element in their product strategy.

The toolsets are continually improving, and this has advantages as well as disadvantages. Improved tools can assist with a new design in all the ways discussed above. However, once a design is completed there is the question of long-term availability and stability of the toolset to support future design modifications, or at a minimum, the ability to work on the source with compatible equivalent tools. Adherence to widely used and recognized standards is very important to avoid future problems.

In the case of safety-critical applications, some of the tools, in particular those that participate in the generation of the final design of the circuit (e.g., synthesis and place&route tools), and those that are relied upon in the integrity assessment of the final circuit (e.g., analysis and verification tools) are subject to specific safety requirements and are the object of regulatory scrutiny. The requirements for these tools are generally the same as those that apply to tools used to develop software applications important to safety. As an example, Safety Evaluation Report from the Nuclear Regulatory Commission on the Wolf Creek NPP Licensee Amendment Request for the MSFIS replacement project has established an evaluation approach completely similar to that used for digital micro-processor based systems. Tools must be subject to version control and configuration management.

Typically, application designers use the toolset provided by the circuit vendor for design implementation, verification and testing. In addition, for safety-critical and high-reliability applications, tools from independent sources typically are needed to support independent testing and verification of the design. This can be challenging due to the tendency for tool vendors to merge, giving fewer independent choices. For critical applications, availability of independent tools should be evaluated.

3.1.8 USER DOCUMENTATION

Circuit vendors should provide system designers with adequate user documentation for their products (circuits, IP cores and tools). This documentation should provide all information necessary for using the products, such as functions, interfaces, input/output protocols, description of control registers, data-sheets, failure rates and possible failure modes, operating conditions, etc. It should also describe all operating modes (including failure modes, power-on and reset) and transitions. Vendor recommendations for using their products in critical applications, for achieving high reliability, and for avoiding unacceptable failure modes also should be considered as part of circuit selection.

3.2 DESIGN

3.2.1 ELECTRONIC SYSTEM LEVEL AND CIRCUIT REQUIREMENTS SPECIFICATION

An electronic system may be described at a high level (e.g., ESL or Electronic System Level) as a set of interacting components, each component being formally represented by a dynamic process. The process representing a component specifies and emulates the behavior, response times and interfaces of the component. Typical components could be microprocessors with their software, memory circuits, specialized computing units or communication channels.

ESL descriptions are particularly useful for system designs based on multiple highly functional components including those that can be created with FPGAs. They allow system designers to assess system behavior and performance under different assumptions: system architecture, system interactions with the environment, individual components' functional and performance specifications, functions and performance requirements allocation to components, and component failures in various modes. They also allow system designers to identify, analyse and specify the constraints that a part of the design induces on other parts.

An important result of ESL descriptions is that they encourage precise, unambiguous and complete requirements specification of individual components, and provide future modification projects with very useful information.

ESL descriptions are typically performed with languages such as C++, SystemC (IEEE 1666), SystemVerilog (IEEE 1800), or Matlab®.

3.2.2 SELF-MONITORING

In addition to the application functions required of the system and the ancillary functions that allow or facilitate the testing, commissioning, operation and maintenance of the system, the FPGA circuit may include self-monitoring functions. The main purpose of such functions is to increase the likelihood that circuit defects that could appear during operation (mainly due to random hardware mechanisms) and circuit malfunctions are detected in a timely manner and reported to operation and maintenance staff.

Different monitoring techniques could be put in place:

- Internal redundancy and cross-checking may be used to detect incorrect processing due to random errors in a single channel.
- Runtime checking of design assertions and/or assumptions may be used against both random and systematic errors. For example, a logic block may check the validity of its inputs (as specified by pre-conditions) and of its outputs (as specified by post-conditions). One of the benefits of the FPGA technology is that such checks may be designed so that they do not interfere directly with the function being checked.
- Parity checks or cyclic redundancy checks (CRCs) may be used to detect random data alterations.
- Range and plausibility checks (of the values of key signals and registers, of internal states, etc.) may be used to detect obviously incorrect data.

Mainly in safety-critical applications, the monitoring functions should be specified and designed so that they will not directly interfere with the required application functions. This prevents the monitoring functions from causing unsafe failures of the application functions and also facilitates the verification (by test and analysis) and validation of the primary and safety functions, making

licensing process simpler, faster and cheaper. In these cases, external circuit monitoring could be a better choice.

3.2.3 EXTERNAL CIRCUIT MONITORING

When an FPGA is used for safety-critical functions, considerations for external monitoring and forcing of acceptable failure modes should be evaluated. For instance, use of external watchdogs could detect the absence of timely outputs. One way communication to another device (e.g. an external computer) sending data on inputs, outputs, internal states, etc., could be another possibility. As this external device does not perform any safety function, and could comply with isolation requirements from the safety portion of the system (both electrical and digital isolation), there would not be any special requirement for it.

3.2.4 SYSTEM ON CHIP

With advanced I/O blocks embedded on chip, system designers can contemplate the integration of all or nearly all electronic circuitry necessary for an I&C function into a single FPGA circuit: inputs acquisition, parameters acquisition, data communication, logic processing, and outputs transmission. This may bring a number of benefits, such as reduced number of hardware components (with possible enhanced reliability) and simpler overall design.

At the same time, maintaining separation or segregation among I&C functions within the circuitry can be beneficial, as it enhances simplicity, verifiability and testability of individual functions. Lower-cost solutions for logic processing using FPGAs may allow system designers to consider reversing the current trend where increasing numbers of functions are concentrated into the same processors in software-based solutions. This also relates to functional independence.

3.2.5 FUNCTIONAL INDEPENDENCE

In electronic design it is much easier than in software design to have a strong guarantee that one function will not interfere with another function implemented in the same circuit. Thus, when multiple functions are performed by the same circuit, it is important that system designers specify (in the Circuit Requirements Specification) the independence requirements between functions.

For example, when two applications' functions are performed by the same circuit but are otherwise functionally independent, it is preferable that they remain so in the hardware design and in the implementation. The same can be said for self-monitoring functions and most ancillary functions: they should normally not interfere directly with the application functions.

That independence is not necessarily symmetric: function A may not be influenced by function B, but B might be influenced by A.

3.2.6 COMPETENCIES

Though FPGA application development and software programming do have some similarities, they are fundamentally different skill sets, and software designers and designers of microprocessor-based systems do not necessarily have the competences and skills required for FPGA designers. The similarities lie mainly in the fact that the activities to develop the electronic design for an FPGA are typically organized into a series of steps summarized in the V-shaped lifecycle. Also, the activities lying in the upper part of the development cycle (requirements specification, architectural design, integration, validation) rely on the same general principles. But even then, there are differences in the details.

Big differences appear in the lower part of the development cycle: in the design and the implementation. Design of an integrated circuit is inherently parallel (electric signals propagate concurrently in multiple wires and through multiple blocks), whereas in single core microprocessors, software logic is inherently sequential: statements are executed one after the other, one statement at a time. Nowadays multi-core microprocessors are not used yet in I&C applications, or are used as single-core processors, so there is no parallel processing at all. Differences in implementation are even more evident: though both electronic implementation and software implementation may include an optimization step, there is no notion of synthesis and place&route in software implementation.

It is also important to recognize that the final product is at base, a piece of hardware and subject to electrical and physical phenomena against which software guidelines generally cannot protect. Therefore, it is equally important that experienced hardware engineers be heavily involved in design, implementation and validation. As in many other areas of design, a team-based approach is likely to work best for safety-related applications. While a software-like process is used to develop the application for an FPGA and software tools play a substantial role in V&V, the FPGA is not and should not be considered a software product. Nevertheless, current discussion with regulators on this issue is underway, as can be seen in the Safety Evaluation Report from the US Nuclear Regulatory Commission on the ALS^{TM} platform application for Wolf Creek NPP, or in NUREG CR-7006, where it is stated that “in most cases, FPGA designs are complex to the level that makes the 100% testability too costly and time consuming. The proposed design life-cycle including the V&V process should be used as a substitute for 100% testability”.

3.3 DEVELOPMENT

3.3.1 SAFETY STANDARDS

Since the use of FPGAs is relatively new in the nuclear industry, there is currently only one published nuclear-specific international standard providing requirements or recommendations

for this activity: International Electrotechnical Commission (IEC) standard 62566, issued at the end of January 2012, which is applicable to the development of Category A (according to IEC 61226) or Class 1 (according to IEC 61513) applications based on programmable complex electronic components (i.e., FPGAs and CPLDs), or as it has been finally stated, HDL Programmed Devices (HPD) [21]. Category A or Class 1 corresponds more or less to Class 1E applications in USA. Committee SC 45A on I&C for nuclear facilities has been in charge of its development.

It has to be noticed that this new standard has to be used in conjunction to IEC 60987 on hardware aspects of programmed digital computers important to safety in NPPs. Additionally, some recommendations from IEC 60880 Software for Computers in the Safety Systems (Category A) of NPPs, and IEC 62340 Requirements for Coping with Common-Cause Failure, has to be taken into account.

International Atomic Energy Agency (IAEA) is preparing a technical document (TECDOC series) on the use of FPGAs in nuclear power plants.

Other industries with safety concerns already have a long history of using FPGAs and have published their own guidelines and requirements, most notably the civil aviation industry, which uses RTCA DO 254 Design Assurance Guidance for Airborne Electronic Hardware [22] and DOT/FAA/AR-95/31 Design, Test, and Certification Issues for Complex Integrated Circuits [23]. Other sectors can be railways, military or aerospace. National Aeronautics and Space Administration (NASA) and European Space Agency (ESA), for instance, have published large amounts of bibliography related to electronic programmable devices for mission-critical applications.

The IEC standard draws from those civil aviation industry existing documents and from the lessons learned from projects that have already been implemented internationally.

3.3.2 DEVELOPMENT LIFECYCLE

3.3.2.1 V-SHAPED LIFECYCLE

Like with software development, the activities leading to the development of a final FPGA circuit actually performing the required functions may be presented in a V-shaped lifecycle. This lifecycle is consistent with the one suggested by IEC 62566 standard.

3.3.2.2 APPLICATION-ORIENTED DEVELOPMENT PROCESSES

When developing software for I&C applications, it is necessary to distinguish two main types of software components: system software and application software. The first one is part of the I&C platform used to develop the system, and comprises the operating system, function libraries and

so on. The second one refers to software components that are specifically developed for the target application and is based on system software.

Whereas system software is generally developed following a conventional software development process, application software is today generally based on diagrammatic application-oriented languages semantically close to the application domain and syntactically well understood by plant engineers. In these conditions, the development lifecycle for application software is different from the one followed for system software, with more emphasis on requirements specification and functional validation, and less on design, which is in large part supported by automatic code generators.

Similar concepts may be applied to electronic design. Ancillary functions like self-monitoring, and library functions may follow a conventional electronic design process, whereas application functions may follow a more application-oriented process. Indeed, some application-oriented languages used for application development can be translated into an RTL description ready for electronic implementation, along with appropriate verification and validation and technical review of the resulting electronic circuit design.

3.3.2.3 OVERALL PROJECT ORGANIZATION

Electronic design and software design have their similarities and their differences and it is important that design of integrated circuits be managed by a team including electronic engineering expertise with FPGAs and software design process expertise. One of the similarities is in the overall complexity level of FPGA application design and software programming. Both are susceptible to design errors that could lead to systematic failure each time the system is put in the same conditions (as opposed to random hardware failures that occur at unpredictable times and places).

Therefore, the overall project organization for an FPGA-related project is similar to a software-related project, with a strong focus on planning, appropriate types and levels of skills, quality assurance, configuration management, verification and reviews, independent verification and validation (IV&V) for the most critical projects and documentation of the development activities.

3.3.3 CIRCUIT REQUIREMENTS SPECIFICATION

As is the case for software development, it is essential that all the requirements that are applicable to the final circuit design are fully specified, and in an unambiguous manner. This specification is the starting point of, and the reference for, all the subsequent FPGA electronic design activities. It usually results from the overall system design which identifies the components that constitute the system, defines their interactions, identifies their mutual constraints, and allocates the system requirements to the components.

Items that should be addressed in the specification include:

- Operating modes of the circuit (e.g., initialization, reset, normal modes and failure modes) and their transitions.

  A state machine focus is recommended. Behaviour of a state machine should be defined not only for the used states but also for the unused states. Most synthesis tools will ignore unused states and synthesize a state machine that can become stuck in an undefined state after entering it unexpectedly. A similar situation occurs if a state in the state machine is not defined for all possible combinations of its inputs. Therefore, all states and state transitions in a state machine should be explicitly defined.

- Functions to be performed by the circuit in each mode and when transitions occur. This generally includes the application functions that are the primary purpose of the circuit, and ancillary functions. Ancillary functions may include functions that facilitate or are needed for operation (e.g., configuration settings, setpoints, etc.), self-monitoring functions, forcing of internal signals and registers, etc.

- Response times. In safety applications this is usually critical, as the system must comply with Plant Technical Specification time response requirements that have been considered in the safety analysis of the plant.

- Independence requirements (both between primary and ancillary and even between primary functions) to guarantee that one function will not directly interfere with another, even in case of failure.

- Related to failure modes, which ones can be considered acceptable, reliability requirements (failure rates calculations), etc., which may influence the selection of the hardware circuit and its technologies.

- Mechanisms to prevent spurious activation of particular functions (e.g., activation of configuration or calibration modes).

- Inputs, outputs and parameters (modifiable during operation or not) of each function, type, format, ranges, limits, resolution and accuracy, time stamp requirements, etc.

- Data communication interfaces and protocols, covering all applicable layers of communication.

- Circuit electrical interfaces, including power supplies and grounding.

- Environment requirements, like temperature, humidity, vibration, EMC, radiation or seismic.

- Tin whiskers protection.

  Use of lead-based soldering, soldering mask, etc., is recommended, especially in this type of electronic devices, using pin grid array (PGA), ball grid array (BGA) and quad flat pack (QFP) technologies, because of the high pin count and small pitch. If not possible, error detection and BIST could be alternatives to consider.

- Diversity and defense in depth requirements. NUREGs CR-6303 and CR-7007 are good references for D&DiD, and NUREG 800 SRP and BTP 7-14. EPRI TR 1002835 is also a very good industry reference.
- Cybersecurity requirements (access control, authentication, authorization, ciphering requirements, anti-malware countermeasures, like intrusion detection or intrusion protection devices, both host or network based, antivirus, patch managements, whiltelisting technologies and so on).

  Nevertheless, FPGA-based solutions themselves provide much more defensive concept due to the fact that, following design guidelines provided here, there are implemented in flat hardware logic completely independent from ancillary functions. Only in the case of IP-cores or other more complex solutions available for FPGAs, involving runtime software, and communications protocols with external systems or devices for HMI, both for operation or maintenance, a complete and deep cybersecurity analysis and protected measures must be applied. In this case, some regulatory and standards requirements should be followed, like Title 10 of Code of Federal Regulations, part 73.54, Regulatory Guide 5.71, ISA/IEC 62443 (formerly ISA S99 series standards), National Institute of Standards and Technology (NIST) SP800-53 and SP800-82, Nuclear Energy Institute guidelines, etc.
- Standards to be applied and followed. This is crucial in the case of safety-critical applications for the licensing process.

  Information about regulation and standards has been previously discussed in 2.3.3 and 3.3.1.

3.3.4 PRELIMINARY DESIGN

In preliminary design phase decisions are made on major design issues, like:

- Organization for initialization and the other required modes of operation.
- Organization into sequential, synchronous steps and combinatorial logic.
- Organization of defensive design, fault-tolerance, self-monitoring and safe failure modes, and determination of internal redundancies.
- Determination of internal independencies (in addition to those already required by the previous requirements specification).
- Organization for testing and access to internal signals.
- Selection of library functions and IP cores to be used, if needed.
- Decomposition into, and specification of, simpler modules (application-specific or pre-developed).

- Selection of the FPGA circuit.

3.3.4.1 DESIGN FOR RELIABILITY

Currents FPGAs are capable of processing all the logic and operations required by a typical NPP I&C system. This also would provide better performance in terms of time response.

Nevertheless, and derived from functional segmentation recommendation, and from diversity and defense in depth, it is considered more appropriate to split different set of independent functions into different circuits. Additionally, extra capabilities of the different circuits can be used for verification and enhance fault-tolerance.

This leads to the concept of fault containment regions (FCR). Proper FCRs should have:

- Independent clocking with bounded clock skew.
- Independent power and grounding.
- Dielectric isolation from other FCRs.

3.3.4.2 INITIALIZATION

Particular attention should be given to initialization, as the circuit's internal electrical and temporal characteristics could be different at power-up. Also, outputs, registers and internal states should be forced to known values during initialization.

3.3.4.3 TESTABILITY AND OBSERVABILITY

Preliminary design needs to address testability and observability issues. Most signals and registers will be not accessible, so it will be impossible to observe what actually happens in the real physical circuit. Provisions can be made during requirements specification and preliminary design to allow access to key signals and registers, both for observation or to force their values for test purposes.

3.3.5 DESIGN

In this phase a detailed design description of the logic processing to be performed by the circuit is generated, usually at the RTL level.

3.3.5.1 SYNCHRONOUS DESIGN

A synchronous component is a component in which the internal registers and outputs are modified simultaneously and at times defined by a clock signal. A synchronous circuit is a circuit

in which the components that communicate with one another are synchronized by the same clock signal. The set of components synchronized by the same clock signal constitutes a *clock domain*.

Because asynchronous designs are prone to glitches, bus skews, and other timing issues, it is highly recommended, particularly for safety applications and for applications critical to plant performance, that FPGA designs are made synchronous whenever possible. This means that different clock domains should not exchange data with one another. The main reason is that so-called *clock-domain crossings*, where data is transmitted from one clock domain to another, can be very problematic and require extreme care. Otherwise, they could lead to non-deterministic behaviours.

Synchronous design also helps provide modularity and clarity of the design. Furthermore, the FPGA design tools do not generally support asynchronous timing constraint and analysis.

Nevertheless, asynchronous designs are more suitable for 100% testability. This constitutes a great advantage. If asynchronous designs are used for 100% testability or any other reason, appropriate measures need to be taken to make sure that the output glitches and the bus skews are not affecting safe operation of the FPGA design. These measures may include use of registered I/Os or analog filtering of the FPGA outputs.

3.3.5.2 METASTABILITY

Metastability can occur when an asynchronous input gets clocked within the FPGA, and it is expressed as an undetermined state at the output of a flip-flop (Figure 15). The undetermined state resolves itself after the recovery time, which is on the order of several nanoseconds to several tens of nanoseconds for most of FPGAs. Metastability with a recovery time of 1 ns occurs on average every microsecond (mean time between metastability occurrences or MTBMO) while the metastability with a 3 ns recovery time is extremely rare (more than one million years of MTBMO).

As a consequence, it is important to consider the recovery time when determining the maximum clock speed of the design.

In order to mitigate the occurrence of metastability, it is recommended to use double registers for edge-sensitive transfers between different clock domains.

3.3.5.3 POWER SUPPLY

Most FPGAs use multiple power supplies, at least one for the FPGA core and one for the input/output (I/O) buffers. In addition, some FPGAs use separate reference voltage supplies for differential I/O buffers. Proper power sequencing is required in some FPGAs to avoid an unpredictable behaviour of the I/O buffers during power-up and power-down. Additionally, most

FPGAs have power ramp-up timing requirements which set the upper and the lower limit of the power ramp time.

An additional issue is the startup current which in some cases can exceed the operational current. Care should be taken to implement voltage regulators that can support excess currents during the power-up.

After power-up, reconfigurable FPGAs are configured from an internal or an external memory which can take up to several hundreds of microseconds depending on the FPGA size and the configuration clock. Care should be taken to prevent transient behaviour of the external circuitry driven by the FPGA during the configuration time. Appropriate measures include keeping the interfacing circuitry disabled during the configuration time or pulling up or down the critical inputs of the interfacing circuitry.

3.3.5.4 POWER PIN DECOUPLING

Due to a simultaneous switching of a large number of gates in an FPGA, the supply current has a shape of periodic spikes which can cause spike-like voltage bounces at the ground pins. To reduce the ground bounce and the digital noise, every power pin should be decoupled using decoupling capacitors. Some of the FPGAs have their decoupling capacitors built in.

3.3.5.5 UNUSED INPUT/OUTPUT PINS

To reduce the power dissipation, the unused I/O pins should be properly programmed or tied to the ground or the power, depending on the particular FPGA I/O specifications. All unused differential I/Os should be configured as single-ended to save the power used by the differential I/O bias circuitry.

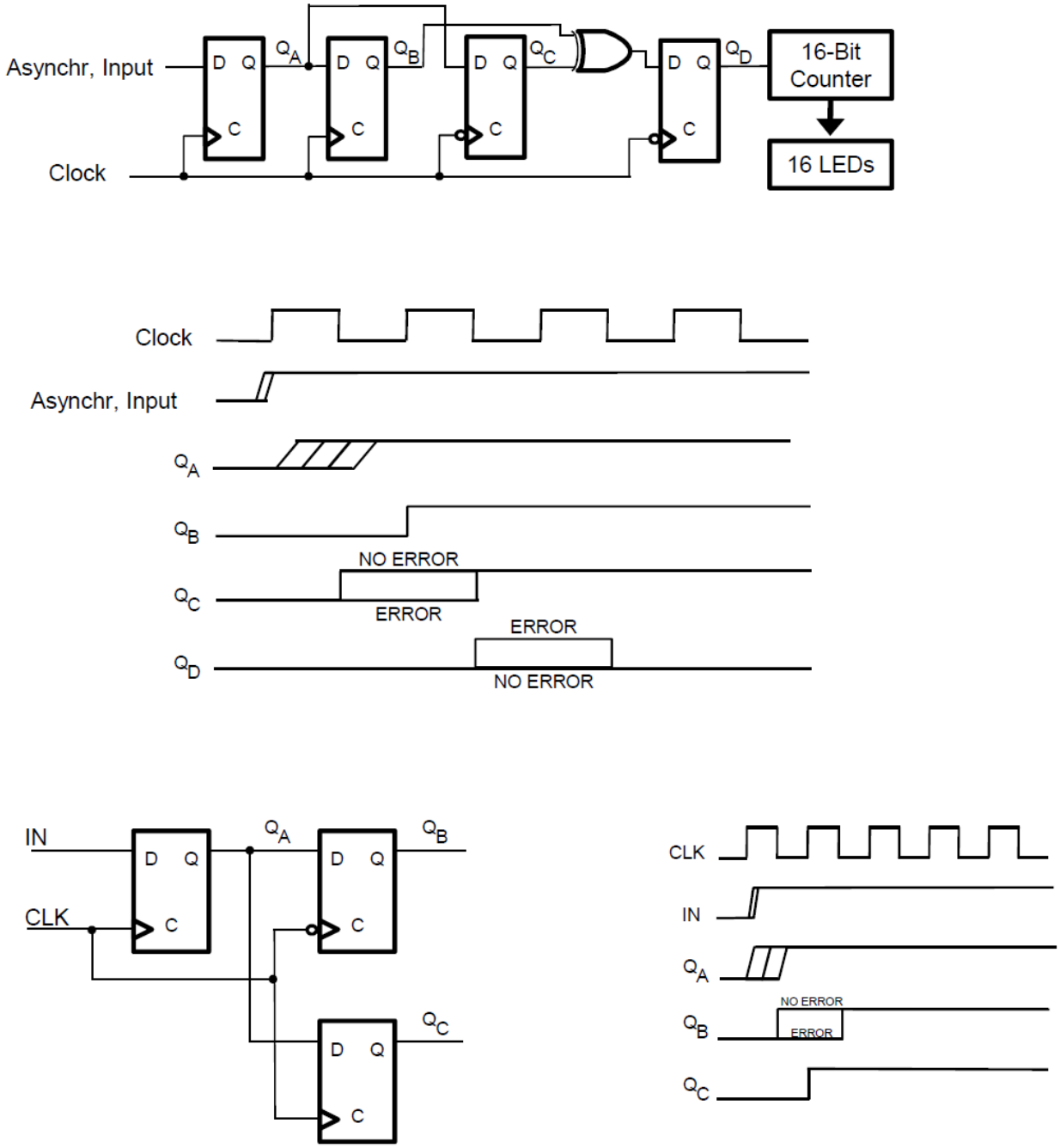

Figure 15 – Metastability

3.3.5.6 PROGRAMMING PINS

FPGA programming pins, such as JTAG port pins, may be subject to noise if not pulled up or down by resistors. This noise can either alter or erase the configuration memory. Also, FPGAs with external configuration memory require specific rules for the layout of the connections between the external configuration memory and the FPGA programming pins.

3.3.5.7 SECURITY

To prevent accidental or malicious reconfiguration of FPGAs or external configuration programmable read-only memories (PROMs), the configuration memory should be write-protected by using write security bits.

Malicious reconfigurations should be prevented by design and/or by administrative controls. In the first case, the FPGA design can include error detection circuitry or built-in self-test (BIST) to constantly monitor the health of the FPGA system. In the second case, making JTAG connector physically accessible only to the trusted personnel could be an example.

3.3.5.8 INPUT OVERFLOW

A robust HDL design should be able to properly handle a potential input overflow. In the first step, the overflow should be recognized either by sampling an external overflow flag or by checking whether the input is in an acceptable range. In the second step, proper actions are taken such as ignoring the current input sample, replacing it by a default value, accepting the overflow, or reporting the overflow.

3.3.5.9 INPUT ACTIVITY

To prevent the hang-ups in the design due to an inactive or disconnected input, the HDL code should implement a logic that will respond to an inactive input. The response may include a specific internal action, such as ignoring the input or setting an error flag.

3.3.5.10 SIMULTANEOUS SWITCHING OUTPUTS

FPGA outputs that transition at the same time should be distributed across the FPGA output pins so that the simultaneous switching output requirement satisfies the FPGA technical specifications.

3.3.5.11 OUTPUT SLEW RATE

To reduce the digital noise due to the output switching, the slew rate of the FPGA output pins should be set to the minimum value defined in the FPGA technical specifications. The use of high-slew-rate outputs should be limited to the case where the fast output switching is essential for proper FPGA interfacing.

3.3.5.12 OUTPUT CURRENT DRIVE

Most of the FPGAs have a selectable current drive at the output pins. To reduce the digital noise due to the output switching, the current drive of the FPGA output pins should be set to the minimum value unless a higher current drive is needed for proper FPGA interfacing.

3.3.5.13 CLOCK TRACES

Long signal lines, if not properly terminated, can cause reflections and crosstalk. This is especially important for the clock and other high-speed lines that should be terminated at the source and/or the end of the line using serial and/or parallel termination networks.

3.3.5.14 LATCHES

Even though latches use fewer gates than conventional flip-flops, special care should be taken when the use of latches is necessary as the noise occurring at the latch inputs may propagate to the latch outputs. Use of flip-flops eliminates this problem.

3.3.5.15 EXTERNAL RESET

If an external reset signal is provided to an FPGA, it should be bounce-free and have a sufficiently fast transition between the logic levels. Since different FPGAs may require different rise times at their inputs; the use of a dedicated reset integrated circuit is suggested.

Usually, all flip-flops in FPGAs are cleared after the power-up unless specified differently by the designer, in which case a dedicated synchronous/asynchronous reset network controlled by the reset logic inside the FPGA is used. Care should be taken to ensure synchronous reset of all flip-flops across the FPGA. This is especially critical when an asynchronous reset signal is used to clear counters or state machines that need to run synchronously.

The clock period must be greater than the resolution time for the flip-flops as well as skews on the reset network are less than half of the clock period. Both of these requirements are easily satisfied in modern FPGAs, since most FPGAs have dedicated low-skew lines for the reset signal and use low-resolution time flip-flops.

Another critical timing issue is the maximum delay and/or maximum skew on high-fan signals such as RESET or ENABLE. Designers should ensure that the maximum delay on these lines is less than the clock period to ensure the resetting and/or enabling of all flip-flops in the design occurs within the same clock period.

3.3.5.16 PRINTED CIRCUIT BOARD LAYER STACKING

The analog and the digital parts of the design residing on the same board should be physically separated to reduce injection of the digital noise into the analog signals. Also, separate power supplies and separate ground and power planes should be used. The analog and the digital ground planes should be connected in a single point to reduce the current loops in the planes.

To ensure a high integrity of the FPGA board design, the stacking of the board layers is critical. Preferred layout-stacking alternates the signal layers and the power/ground layers. Below are shown two examples of a preferred 8-layer PCB stacking.

Table 5 – Printed circuit board layer stacking options

LAYER	OPTION 1	OPTION 2
1	Signal	Signal/Power
2	Ground	Ground
3	Signal	Signal/Power
4	Power	Ground
5	Ground	Power
6	Signal	Signal/Power
7	Ground	Power
8	Signal	Signal/Power

3.3.5.17 LANGUAGES

It is highly preferable that the design is expressed in a high level language that can be understood by the designers (current and future, considering the long lifetimes of I&C applications) and the verifiers. Conventional electronic design languages are usually called Hardware Description Languages or HDLs. Application-oriented languages may also be used for application functions. They tend to be semantically closer to the functional needs and more easily understood by application specialists, thus reducing the potential for errors. In general, automatic translation is used to produce an equivalent HDL representation, so that later on in the development lifecycle, the code for application functions can follow the same path as the conventionally developed code.

Using different languages to code different blocks in the same design should be avoided due to potential language incompatibility and different synthesis rules. However, different languages

can and should be used to design redundant identical blocks, redundant FPGAs, or redundant boards to provide additional diversity.

3.3.5.18 CODING RULES

It is often preferable to define and adhere to coding rules (i.e., rules for using the selected design languages, conventional or application-oriented). These rules may serve several purposes:

- Avoid introduction of coding errors.
- Facilitate understanding by current and future developers.
- Facilitate verification, by reviews, simulation or formal methods.
- Remain within the guaranteed domain of the supporting tools.
- Limit the potential for differences between the simulated and the actual behaviours.
- Facilitate future modifications.
- Enhance portability to future hardware circuits.

Coding rule sets should address topics such as the following:

- Naming conventions.
- Formatting conventions.
- Language features and constructs to be avoided or kept under tight control.
- Standardized constructs, e.g., for redundancy or module independence.
- Organization of computations on critical paths.
- Use of the structuring features (such as functions blocks, macros, etc.).
- Initialization of variables.

There is lot of bibliography regarding best practices and coding rules, both generic and language specific, from high level to assembler languages. NUREG CR-6463 [24] on guidelines on software languages for use in nuclear power plant safety systems is a very good example because of its completeness and being a nuclear sector document coming from US regulator. This document analyses different programming attributes, like reliability, robustness, traceability and maintainability, of several languages that can be used in such safety applications, like Ada, C/C++, IEC 61131-3 ladder logic and sequential function charts, Pascal and PL/M. Many of these recommendations could be applicable to FPGA programming, while others not.

Other references could be NASA documents GB-8719.13 [25] on guide for safety-critical software, TM-2000-210616 [26] on software fault tolerance or GB-A302 [27] on software formal inspections.

Following there is a series of coding rules proposed as highly recommended in the case of FPGA programming, with examples in the two main HDLs (VHDL and Verilog).

A. *If* and *case* statements may be used to design purely combinatorial logic such as multiplexers, encoders, de-multiplexers, and decoders as well as sequential logic such as state machines. When using *if-elseif* or *case* statements, all the branches should be defined explicitly for all possible input combinations. Also, all the outputs should be defined for every branch. If priority is not needed, use *case* statements instead of *if* statements to minimize the logic created by synthesis tools. Failing to completely define *if* or *case* statements can cause some synthesis tools to insert latches that are difficult to simulate.

```
--VHDL 4-to-1 priority mux
process (RST,SEL,IN1,IN2,IN3,IN4)
begin
    if RST = '1' then O <= '0'; -- reset
    elsif SEL = "00" then O <= IN1;
    elsif SEL = "01" then O <= IN2;
    elsif SEL = "10" then O <= IN3;
    else O <= IN4;
    end if;
end process;

//Verilog 4-to-1 priority mux
always @ (RST or SEL or IN1 or IN2 or IN3 or IN4)
    begin
            if (RST == 1'b1) O = 1'b0; //reset
            else if (SEL == 2'b00) O = IN1;
            else if (SEL == 2'b01) O = IN2;
            else if (SEL == 2'b10) O = IN3;
            else O = IN4;
    end
```

```
--VHDL 4-to-1 mux using case statement
process (SEL,IN1,IN2,IN3,IN4)
begin
    case SEL is
            when "00" => O <= IN1;
            when "01" => O <= IN2;
            when "10" => O <= IN3;
            when others => O <= IN4;
    end case;
end process;

//Verilog 4-to-1 mux using case statement
always @ (SEL or IN1 or IN2 or IN3 or IN4)
    begin
            case (SEL)
                    2'b00 : O = IN1;
                    2'b01 : O = IN2;
                    2'b10 : O = IN3;
                    2'b11 : O = IN4;
            endcase
    end
```

```
--VHDL 4-to-2 priority encoder
process (RST,IN1)
begin
    if RST = '1' then O <= "00";
    elsif IN1 = "0000" then O <= "00";
    elsif IN1 = "0010" then O <= "01";
    elsif IN1 = "0100" then O <= "10";
    elsif IN1 = "1000" then O <= "11";
    else O <= "00";
    end if;

//Verilog 4-to-2 priority encoder
always @ (RST or IN1)
    begin
            if (RST == 1'b1) O = 2'b00;
            else if (IN1 == 4'b0001) O = 2'b00;
            else if (IN1 == 4'b0010) O = 2'b01;
            else if (IN1 == 4'b0100) O = 2'b10;
            else if (IN1 == 4'b1000) O = 2'b11;
            else O = 2'b00;
    end
```

```
--VHDL 4-to-2 encoder using case statement
process (IN1)
begin
    case IN1 is
            when "0001" => O <= "00";
            when "0010" => O <= "01";
            when "0100" => O <= "10";
            when "1000" => O <= "11";
            when others => O <= "00";
    end case;
end process;

//Verilog 4-to-2 encoder using case statement
always @ (IN1)
    begin
            case (IN1)
                    4'b0001 : O = 2'b00;
                    4'b0010 : O = 2'b01;
                    4'b0100 : O = 2'b10;
                    4'b1000 : O = 2'b11;
                    default : O = 2'b00;
            endcase
    end
```

```
--VHDL 1-to-4 priority demux
process (RST,SEL,IN1)
begin
    if RST = '1' then O <= "0000"; -- reset
    elsif SEL = "00" then O <= "000"&IN1;
    elsif SEL = "01" then O <= "00"&IN1&'0';
    elsif SEL = "10" then O <= '0'&IN1&"00";
    else O <= IN1&"000";
    end if;
end process;

//Verilog 1-to-4 priority demux
always @ (RST or SEL or IN1)
    begin
            if (RST == 1'b1) O = 4'b0000; -//reset
            else if (SEL == 2'b00) O = {3'b000,IN1};
            else if (SEL == 2'b01) O = {2'b00,IN1,1'b0};
            else if (SEL == 2'b10) O = {1'b0,IN1,2'b00};
            else O = {IN1,3'b000};
    end
```

```
--VHDL 1-to-4 demux using case statement
process (SEL,IN1)
begin
    case SEL is
            when "00" => O <= "000"&IN1;
            when "01" => O <= "00"&IN1&'0';
            when "10" => O <= '0'&IN1&"00";
            when others => O <= IN1&"000";
    end case;
end process;

//Verilog 1-to-4 demux using case statement
always @ (SEL or IN1)
    begin
            case (SEL)
                    2'b00 : O = {3'b000,IN1};
                    2'b01 : O = {2'b00,IN1,1'b0};
                    2'b10 : O = {1'b0,IN1,2'b00};
                    2'b11 : O = {IN1,3'b000};
            endcase
    end
```

```
--VHDL 2-to-4 priority decoder
process (RST,IN1)
begin
    if RST = '1' then O <= "0000"; -- reset
    elsif IN1 = "00" then O <= "0111";
    elsif IN1 = "01" then O <= "0110";
    elsif IN1 = "10" then O <= "1100";
    else O <= "1110";
    end if;
end process;

//Verilog 2-to-4 priority decoder
always @ (RST or IN1)
    begin
            if (RST == 1'b1) O = 4'b0000; // reset
            else if (IN1 == 2'b00) O = 4'b0111;
            else if (IN1 == 2'b01) O = 4'b0110;
            else if (IN1 == 2'b10) O = 4'b1100;
            else O = 4'b1110;
    end
```

```
--VHDL 2-to-4 decoder using case statement
process (IN1)
begin
    case IN1 is
            when "00" => O <= "0111";
            when "01" => O <= "0110";
            when "10" => O <= "1100";
            when others => O <= "1110";
    end case;
end process;

//Verilog 2-to-4 encoder using case statement
always @ (IN1)
    begin
            case (IN1)
                    2'b00 : O = 4'b0111;
                    2'b01 : O = 4'b0110;
                    2'b10 : O = 4'b1100;
                    2'b11 : O = 4'b1110;
            endcase
    end
```

B. Addition of multiple numbers should be implemented as a synchronous cascade of two-number adders to avoid different synthesis implementations and to improve timing performance. Also, care should be taken to properly size the data path to avoid overflows in the addition tree.

```
--VHDL pipelined adder tree with 5 inputs and 3-cycle latency
process (CLK)
begin
  if CLK'event and CLK = '1' then
        LEVEL_ONE_SUM1 <= IN1 + IN2;
        LEVEL_ONE_SUM2 <= IN3 + IN4;
        LEVEL_ONE_SUM3 <= IN5;
        LEVEL_TWO_SUM1 <= LEVEL_ONE_SUM1 + LEVEL_ONE_SUM2;
        LEVEL_TWO_SUM2 <= LEVEL_ONE_SUM3;
        FINAL_SUM <= LEVEL_TWO_SUM1 + LEVEL_TWO_SUM2;
  end if;
end process;

//Verilog pipelined adder tree with 5 inputs and 3-cycle latency
always @ (posedge CLK)
  begin
        FINAL_SUM = LEVEL_TWO_SUM1 + LEVEL_TWO_SUM2;
        LEVEL_TWO_SUM1 = LEVEL_ONE_SUM1 + LEVEL_ONE_SUM2;
        LEVEL_TWO_SUM2 = LEVEL_ONE_SUM3;
        LEVEL_ONE_SUM1 = IN1 + IN2;
        LEVEL_ONE_SUM3 = IN5;
        LEVEL_ONE_SUM2 = IN3 + IN4;
  end
```

C. Among all the choices for counters, the binary counters are sufficient to cover most of the design needs. Gray counters may be used in cases when the single-bit-per-count change is absolutely necessary. Ripple counters should be avoided in safety-critical designs due to difficult timing simulation and verification.

```
--VHDL 8-bit binary counter
process (CLK)
begin
    if CLK'event and CLK = '1' then
            if RST = '1' then
                    CNT <= X"00";
            else
                    CNT <= CNT + '1';
            end if;
    end if;
end process;

//Verilog 8-bit binary counter
always @ (posedge CLK)
begin
    if (RST)
            CNT = 8'h00;
    else
            CNT = CNT + 1;
end
```

D. Random Access Memory (RAM) can be in different forms, including single-port synchronous, dual- port synchronous and dual-port asynchronous. Single-port synchronous RAM can be used whenever simultaneous writes and reads from two different memory locations are not required. This is the simplest RAM implementation that can be easily tested and verified. If simultaneous reads and writes are necessary, dual-port synchronous RAM should be used. Care should be taken to explicitly define RAM behaviour during simultaneous reading and writing to the same memory cell. Depending on the application, there may be either write-first or read-first RAM implementations. Dual-port asynchronous RAM should be generally avoided because of their cross-clock boundaries.

```
--VHDL single-port synchronous RAM
entity sp_sync_RAM is
  port (CLK : in std_logic;
        WE : in std_logic;
        ADDR : in std_logic_vector(7 downto 0);
        DIN : in std_logic_vector(7 downto 0);
        DOUT : out std_logic_vector(7 downto 0));
end sp_sync_RAM;
architecture behavioral of sp_sync_RAM is
  type ram is array(255 downto 0) of std_logic_vector(7 downto 0);
  signal SP_RAM : ram;
begin
process (CLK)
begin
  if CLK'event and CLK = '1' then
        if WE = '1' then
              SP_RAM(conv_integer(ADDR)) <= DIN; -- RAM write
        end if;
        DOUT <= SP_RAM(conv_integer(ADDR)); -- RAM read
  end if;
end process;
end behavioral;
```

```
//Verilog single-port synchronous RAM
module sp_sync_RAM (CLK, WE, ADDR, DIN, DOUT);
  input CLK;
  input WE;
  input [7:0] ADDR;
  input [7:0] DIN;
  output [7:0] DOUT;
  reg [7:0] SP_RAM [15:0];
  reg [7:0] DOUT;
  always @ (posedge CLK)
        begin
              DOUT = SP_RAM[ADDR]; // RAM read
              if (WE) SP_RAM[ADDR] = DIN; // RAM write
        end
endmodule
```

```
--VHDL dual-port read-first synchronous RAM
entity dp_sync_RAM is
  port (CLK : in std_logic;
        WE : in std_logic;
        RE : in std_logic;
        WADDR : in std_logic_vector(7 downto 0);
        RADDR : in std_logic_vector(7 downto 0);
        DIN : in std_logic_vector(7 downto 0);
        DOUT : out std_logic_vector(7 downto 0));
end dp_sync_RAM;
architecture behavioral of dp_sync_RAM is
  type ram is array(255 downto 0) of std_logic_vector(7 downto 0);
  signal DP_RAM : ram;
begin
  process (CLK)
        begin
              if CLK'event and CLK = '1' then
                    if WE = '1' then
                          DP_RAM(conv_integer(WADDR))     <=
                          DIN; -- RAM write
                    end if;
                    if RE = '1' then
                          DOUT                            <=
                          DP_RAM(conv_integer(RADDR))   ;   --
                          RAM read
                    end if;
              end if;
  end process;
end behavioral;
```

```
//Verilog dual-port read-first synchronous RAM
module dp_sync_RAM (CLK, WE, RE, WADDR, RADDR, DIN, DOUT);
  input CLK;
  input WE;
  input RE;
  input [7:0] WADDR;
  input [7:0] RADDR;
  input [7:0] DIN;
  output [7:0] DOUT;
  reg [7:0] DP_RAM [255:0];
  reg [7:0] DOUT;
  always @ (posedge CLK)
        begin
              if (WE) DP_RAM[WADDR] = DIN; // RAM write
        end
  always @ (posedge CLK)
        begin
              if (RE) DOUT = DP_RAM[RADDR]; // RAM read
        end
endmodule
```

E. First-in-first-out (FIFO) structures are used for memory storage, data delay, data rate change, and data format change. They are usually designed as dual-port synchronous RAM with write and read counters controlling the write and the read address. The asynchronous FIFO should be generally avoided in safety-critical designs because their cross-clock boundaries are difficult to simulate and verify.

```
--VHDL synchronous FIFO
RAM_WRITE:process (CLK)
  begin
        if CLK'event and CLK = '1' then
              if WE = '1' and FF = '0' then – write to FIFO if not full
                    FIFO_RAM(conv_integer(WADDR)) <= DIN;
              end if;
        end if;
end process;
RAM_READ:process (CLK)
  begin
        if CLK'event and CLK = '1' then
              if RE = '1' and EF = '0' then – read from FIFO if not
              empty
                    DOUT <= FIFO_RAM(conv_integer(RADDR)) ;
              end if;
        end if;
end process;
WRITE_POINTER:process (CLK,CLR)
  begin
        if CLR = '1' then – clear write pointer
              WADDR <= X"00";
        elsif CLK'event and CLK = '1' then
              if WE = '1' and FF = '0' then – increment write pointer if
              not full
                    WADDR <= WADDR + '1';
```

```
                end if;
        end if;
end process;
READ_POINTER:process (CLK,CLR)
  begin
        if CLR = '1' then – clear read pointer
                RADDR <= X"00";
        elsif CLK'event and CLK = '1' then
                if RE = '1' and EF = '0' then – increment read pointer if
                not empty
                        RADDR <= RADDR + '1';
                end if;
        end if;
end process;
FFLAG:process (CLK,CLR) -- active high
  begin
        if CLR = '1' then -- clear full flag
                FF <= '0';
        elsif CLK'event and CLK = '1' then
                if RE = '1' then -- clear full flag when read
                        FF <= '0';
                elsif WE = '1' and WADDR = RADDR - '1' then
                        FF <= '1'; -- set full flag
                end if;
        end if;
end process;
EFLAG:process (CLK,CLR) -- active high
  begin
        if CLR = '1' then -- clear empty flag
                EF <= '1';
```

```
        elsif CLK'event and CLK = '1' then
            if WE = '1' then
                EF <= '0'; -- clear empty flag when write
            elsif RE = '1' and WADDR = RADDR + '1' then
                EF <= '1'; -- set empty flag
            end if;
        end if;
    end process;
```

```
//Verilog synchronous FIFO
always @ (posedge CLK) // write to FIFO if not full
  begin
        if (WE & !FF) FIFO_RAM[WADDR] = DIN;
  end
always @ (posedge CLK) // read from FIFO if not empty
  begin
        if (RE & !EF) DOUT = FIFO_RAM[RADDR];
  end
always @ (posedge CLK or posedge CLR) // write pointer
  begin
        if (CLR) WADDR = 8'h00; // clear write pointer
        else if (WE & !FF) WADDR = WADDR + 1; // increment write
        pointer if not full
  end
always @ (posedge CLK or posedge CLR) // read pointer
  begin
        if (CLR) RADDR = 8'h00; // clear read pointer
        else if (RE & !EF) RADDR = RADDR + 1; // increment read
        pointer if not empty
  end
always @ (posedge CLK or posedge CLR) // full flag active high
  begin
        if (CLR) FF = 1'b0; // clear full flag
        else if (RE) FF = 1'b0; // clear full flag when read
        else if ((WE) & (WADDR == RADDR - 1)) FF = 1'b1; // set full
        flag
  end
always @ (posedge CLK or posedge CLR) // empty flag active high
  begin
```

```
        if (CLR) EF = 1'b1; // clear empty flag
        else if (WE) EF = 1'b0; // clear empty flag when write
        else if ((RE) & (WADDR == RADDR + 1)) EF = 1'b1; // set empty flag
end
```

F. Safe state-machine design assumes explicit assignments for all states for all possible input combinations. Also, the state-machine outputs should be completely defined for every state. If a state machine contains states that transition to themselves for all input combinations (deadlock states), an external reset signal should be implemented to move the state-machine from the deadlock state. Furthermore, the reset signal is necessary to ensure that the state machine initially starts in a known state. If the reset signal is not synchronous, it should be resynchronized properly.

Consider partitioning of state-machine in smaller state machines in case the number of states is very high. In this case, each sub-machine needs an IDLE state during which the control is passed to another sub-machine.

```
--VHDL state machine with 4 states, two inputs and three outputs using if-elsif statements
process (CLK)
begin
  if CLK'event and CLK = '1' then
        if RST = '1' then -- synchronous reset
              O <= "000";
              STATE <= "00";
        elsif STATE = "00" then -- state zero
              if IN1 = "00" then
                    O <= "000";
                    STATE <= "00";
              elsif IN1 = "01" then
                    O <= "111";
                    STATE <= "01";
              elsif IN1 = "10" then
                    O <= "101";
                    STATE <= "00";
              else
                    O <= "001";
                    STATE <= "11";
              end if;
```

```
elsif STATE = "01" then -- state one
        if IN1 = "00" then
                O <= "000";
                STATE <= "00";
        elsif IN1 = "01" then
                O <= "110";
                STATE <= "01";
        elsif IN1 = "10" then
                O <= "001";
                STATE <= "10";
        else
                O <= "110";
                STATE <= "11";
        end if;
elsif STATE = "10" then -- state two
        if IN1 = "00" then
                O <= "000";
                STATE <= "10";
        elsif IN1 = "01" then
                O <= "101";
                STATE <= "11";
        elsif IN1 = "10" then
                O <= "010";
                STATE <= "00";
        else
                O <= "110";
                STATE <= "01";
        end if;
elsif STATE = "11" then -- state three
        if IN1 = "00" then
```

```
                    O <= "001";
                    STATE <= "11";
                elsif IN1 = "01" then
                    O <= "100";
                    STATE <= "11";
                elsif IN1 = "10" then
                    O <= "001";
                    STATE <= "00";
                else
                    O <= "101";
                    STATE <= "01";
                end if;
            end if;
        end if;
    end process;
```

--VHDL state machine with 4 states, two inputs and three outputs using case statements

```
process(CLK,RST)
begin
  if RST = '1' then -- reset
        O <= "000";
        STATE <= "00";
  elsif CLK'event and CLK = '1' then
        case STATE is
              when "00" => -- state zero
                    case IN1 is
                          when "00" =>
                                O <= "000";
                                STATE <= "00";
                          when "01" =>
                                O <= "111";
                                STATE <= "01";
                          when "10" =>
                                O <= "101";
                                STATE <= "00";
                          when others =>
                                O <= "001";
                                STATE <= "11";
                    end case;
              when "01" => -- state one
                    case IN1 is
                          when "00" =>
                                O <= "010";
                                STATE <= "01";
                          when "01" =>
```

```
                O <= "111";
                STATE <= "01";
            when "10" =>
                O <= "101";
                STATE <= "00";
            when others =>
                O <= "110";
                STATE <= "11";
        end case;
    when "10" => -- state two
        case IN1 is
            when "00" =>
                O <= "000";
                STATE <= "10";
            when "01" =>
                O <= "101";
                STATE <= "01";
            when "10" =>
                O <= "100";
                STATE <= "00";
            when others =>
                O <= "001";
                STATE <= "11";
        end case;
    when others => -- state three
        case IN1 is
            when "00" =>
                O <= "101";
                STATE <= "11";
            when "01" =>
```

```
                O <= "110";
                STATE <= "01";
            when "10" =>
                O <= "000";
                STATE <= "00";
            when others =>
                O <= "001";
                STATE <= "10";
        end case;
    end case;
  end if;
end process;
```

//Verilog state machine with 4 states, two inputs and three outputs using if-else if statements

```
always @ (posedge CLK)
  begin
    if (RST == 1'b1) begin // reset
            O = 3'b000;
            STATE = 2'b00;
            end
    else if (STATE == 2'b00) begin // state zero
            if (IN1 == 2'b00) begin
                    O = 3'b000;
                    STATE = 2'b00;
                    end
            else if (IN1 == 2'b01) begin
                    O = 3'b111;
                    STATE = 2'b01;
                    end
            else if (IN1 == 2'b10) begin
                    O = 3'b101;
                    STATE = 2'b00;
                    end
            else begin
                    O = 3'b001;
                    STATE = 2'b11;
                    end
            end
    else if (STATE == 2'b01) begin // state one
            if (IN1 == 2'b00) begin
                    O = 3'b000;
                    STATE = 2'b00;
```

```
            end
        else if (IN1 == 2'b01) begin
            O = 3'b100;
            STATE = 2'b01;
            end
        else if (IN1 == 2'b10) begin
            O = 3'b001;
            STATE = 2'b10;
            end
        else begin
            O = 3'b110;
            STATE = 2'b11;
            end
        end
    else if (STATE == 2'b10) begin // state two
        if (IN1 == 2'b00) begin
            O = 3'b000;
            STATE = 2'b10;
            end
        else if (IN1 == 2'b01) begin
            O = 3'b101;
            STATE = 2'b11;
            end
        else if (IN1 == 2'b10) begin
            O = 3'b010;
            STATE = 2'b00;
            end
        else begin
            O = 3'b110;
            STATE = 2'b01;
```

```
                end
            end
        else if (STATE == 2'b11) begin // state three
            if (IN1 == 2'b00) begin
                O = 3'b001;
                STATE = 2'b11;
                end
            else if (IN1 == 2'b01) begin
                O = 3'b100;
                STATE = 2'b11;
                end
            else if (IN1 == 2'b10) begin
                O = 3'b001;
                STATE = 2'b00;
                end
            else begin
                O = 3'b101;
                STATE = 2'b01;
                end
        end
    end
```

//Verilog state machine with 4 states, two inputs and three outputs using case statements

```
always @ (posedge CLK)
  begin
    if (RST == 1'b1) begin // reset
          O = 3'b000;
          STATE = 2'b00;
          end
    else if (STATE == 2'b00) begin // state zero
          if (IN1 == 2'b00) begin
                O = 3'b000;
                STATE = 2'b00;
                end
          else if (IN1 == 2'b01) begin
                O = 3'b111;
                STATE = 2'b01;
                end
          else if (IN1 ==2'b10) begin
                O = 3'b101;
                STATE = 2'b00;
                end
          else begin
                O = 3'b001;
                STATE = 2'b11;
                end
          end
    else if (STATE == 2'b01) begin // state one
          if (IN1 == 2'b00) begin
                O = 3'b000;
                STATE = 2'b00;
```

```
            end
        else if (IN1 == 2'b01) begin
            O = 3'b100;
            STATE = 2'b01;
            end
        else if (IN1 == 2'b10) begin
            O = 3'b001;
            STATE = 2'b10;
            end
        else begin
            O = 3'b110;
            STATE = 2'b11;
            end
        end
    else if (STATE == 2'b10) begin // state two
        if (IN1 == 2'b00) begin
            O = 3'b000;
            STATE = 2'b10;
            end
        else if (IN1 == 2'b01) begin
            O = 3'b101;
            STATE = 2'b11;
            end
        else if (IN1 == 2'b10) begin
            O = 3'b010;
            STATE = 2'b00;
            end
        else begin
            O = 3'b110;
            STATE = 2'b01;
```

```
                end
            end
        else if (STATE == 2'b11) begin // state three
            if (IN1 == 2'b00) begin
                O = 3'b001;
                STATE = 2'b11;
                end
            else if (IN1 == 2'b01) begin
                O = 3'b100;
                STATE = 2'b11;
                end
            else if (IN1 == 2'b10) begin
                O = 3'b001;
                STATE = 2'b00;
                end
            else begin
                O = 3'b101;
                STATE = 2'b01;
                end
            end
    end
```

G. Any potential changes of internal states of the design caused by an SEU should be properly handled by the HDL code. To detect an error, double redundancy logic can be implemented at the module level or at the FPGA level, and the comparison of the outputs can be done. Proper action may include a scheduled internal reset of the entire FPGA or reporting the error to an operator. Some parts of the design, such as setting points or state machines, may require error correction to be implemented. Possible methods include triple module redundancy and Hamming coding.

```
--VHDL double module redundancy with reset-on-error
COMPONENT1:process (CLK)
begin
    if CLK'event and CLK = '1' then
            if RST = '1' or ERROR = '1' then
                    CNT1 <= X"00"; -- reset when error
            else
                    CNT1 <= CNT1 + '1';
            end if;
    end if;
end process;
COMPONENT2:process (CLK)
begin
    if CLK'event and CLK = '1' then
            if RST = '1' or ERROR = '1' then
                    CNT2 <= X"00"; -- reset when error
            else
                    CNT2 <= CNT2 + '1';
            end if;
    end if;
end process;
COMPARE:process (CLK)
begin
    if CLK'event and CLK = '1' then
```

```
            if CNT1 = CNT2 then
                ERROR <= '0';
            else
                ERROR <= '1'; -- error set
            end if;
        end if;
    end process;
```

```
//Verilog double module redundancy with reset-on-error
always @ (posedge CLK)
begin
    if (RST | ERROR)
            CNT1 = 8'h00; // reset when error
    else
            CNT1 = CNT1 + 1;
end
always @ (posedge CLK)
begin
    if (RST | ERROR)
            CNT2 = 8'h00; // reset when error
    else
            CNT2 = CNT2 + 1;
end
always @ (posedge CLK)
begin
    if (CNT1 == CNT2)
            ERROR = 1'b0;
    else
            ERROR = 1'b1; // error set
end
```

```
--VHDL Self-correctable, triplicated flip-flop
process (CLK)
begin
    if CLK'event and CLK = '1' then
            if RST = '1' then
                    Q1 <= '0';
                    Q2 <= '0';
                    Q3 <= '0';
            else
                    Q1 <= D; -- flip-flop one
                    Q2 <= D; -- flip-flop two
                    Q3 <= D; -- flip-flop three
            end if;
    end if;
end process;
--voter
Q <= (Q1 and Q2) or (Q2 and Q3) or (Q1 and Q3);
```

```
//Verilog Self-correctable, triplicated flip-flop
always @ (posedge CLK)
begin
    if (RST) begin
            Q1 = 1'b0;
            Q2 = 1'b0;
            Q3 = 1'b0;
            end
    else begin
            Q1 = D; // flip-flop one
            Q2 = D; // flip-flop two
            Q3 = D; // flip-flop three
    end
end
//voter
assign Q = (Q1&Q2)|(Q2&Q3)|(Q1&Q3);
```

H. Others:

- To have better readability of the code and to reduce coding errors, use proper indentation and spacing.
- Code lines should be restricted to 80 character spaces, as a rule of thumb, to improve code readability. Longer lines can be broken with the continuation character and aligned with the first line.
- Add comments to the code to describe the purpose and functionality of each of the components in the design. Also, comment on critical lines and segments of the code to help verification, traceability, and maintainability of the code.
- The design entry file names should match the entity/module name of the VHDL/Verilog code contained in the file. This helps code readability and simulation. Names for signals, variables, wires, instances, etc., should be concise but meaningful using underscore where appropriate.
- To improve code readability, use named rather than positional association for the port mapping when instantiating a sub-module. Also, a single port mapping per line is preferred over single-line port mapping.

```
--VHDL named port mapping
INST0 : tbuf port map (
I => DATA_IN,
O => DATA_OUT,
T => DATA_ENB
);


--VHDL positional port mapping
INST0 : tbuf port map (DATA_IN, DATA_OUT, DATA_ENB);
```

- Use constants and parameters to substitute numbers to help readability and portability of the code.

```
--VHDL constants
constant INIT : std_logic_vector(1 downto 0) := "00";
constant SLOW : std_logic_vector(1 downto 0) := "01";
constant FAST : std_logic_vector(1 downto 0) := "10";
signal STATE : std_logic_vector(1 downto 0);
begin
    if STATE = INIT then
        O <= '0';
    elsif STATE = SLOW then
        O <= IN1;
    elsif STATE = FAST then
        O <= not IN1;
    else
        O <= '1';
    end if;
```

```
//Verilog parameters
parameter INIT = 2'b00;
parameter SLOW = 2'b01;
parameter FAST = 2'b10;
wire STATE;
always @ (posedge CLK)
begin
    if (STATE == INIT) begin
            O = 1'b0;
            end
    else if (STATE == SLOW) begin
            O = IN;
            end
    else if (STATE == FAST) begin
            O = !IN;
            end
    else begin
            O = 1'b1;
            end
end
```

- Avoid using flat-module designs where all the code resides in a single file. Using hierarchical design makes the code easier to read, trace, and verify. Also, it facilitates team work on large designs. In hierarchical code, use the top-level code for component/module declaration and instantiation. The behavioural code should generally be placed at the lowest hierarchical level.

```
--VHDL file top.vhd
entity top is
port (
    IN1 : in std_logic_vector(15 downto 0);
    IN2 : in std_logic_vector(15 downto 0);
    IN3 : in std_logic_vector(15 downto 0);
    O : out std_logic_vector(15 downto 0)
    );
end top;
architecture Behavioral of top is
component adder is
port (
    IN1 : in std_logic_vector(15 downto 0);
    IN2 : in std_logic_vector(15 downto 0);
    O : out std_logic_vector(15 downto 0)
    );
end component;
signal SUM1 : std_logic_vector(15 downto 0);
begin
INST0 : adder port map (
    IN1 => IN1,
    IN2 => IN2,
    O => SUM1
    );
INST1 : adder port map (
    IN1 => IN3,
```

```
    IN2 => SUM1,
    O => O
    );
end Behavioral;

--VHDL file adder.vhd
entity adder is
port (
    IN1 : in std_logic_vector(15 downto 0);
    IN2 : in std_logic_vector(15 downto 0);
    O : out std_logic_vector(15 downto 0)
    );
end adder;
architecture Behavioral of adder is
begin
O <= IN1 + IN2;
end Behavioral;
```

```
//Verilog file top.v
module (IN1, IN2, IN3, OUT);
input IN1 [15 : 0];
input IN2 [15 : 0];
input IN3 [15 : 0];
output OUT [15 : 0];
wire SUM1 [15 :0];
always @ (IN1, IN2, IN3)
    begin
    adder INST0 (
            .IN1 (IN1),
            .IN2 (IN2),
            .OUT (SUM1)
            );
    adder INST1 (
            .IN1 (IN1),
            .IN2 (SUM1),
            .OUT (OUT)
            );
    end
end module;
```

```
//Verilog file adder.v
module (IN1, IN2, OUT);
input IN1 [15 : 0];
input IN2 [15 : 0];
output OUT [15 : 0];
always @ (IN1, IN2)
    begin
        OUT = IN1 + IN2;
    end
end module
```

- To allow for possible regeneration of the FPGA configuration file using different design tools or different FPGAs, the HDL code should be as generic as possible without tool-specific or FPGA-specific directives, structures or hard macros. Same for synthesis attributes and constraints. Avoid place&route directives in the HDL code that are not critical for the design, such as particular placement of design components or use of specific logic and routing resources. Essential place and route constraints, such as pin placements, global clock lines, and registered I/Os, are generally transparent to most of FPGA place&route tools. Examples below show Xilinx place&route constraints specified in the user constraint files (UCFs).

  Essential place and route constraints

  NET “CLK” LOC = V10; place external signal CLK to pin V10

  TIMESPEC “TS_CLK” = PERIOD “CLK” 10 ns; specify the period for signal CLK

  Non-essential place and route constraints

  INST “FF1” LOC = CLB_R4C4; place flip-flop FF1in the CLB in row 4, column 4

  INST “BUF1” LOC =TL; place clock buffer BUF1 in the top left corner

3.3.5.19 PORTABILITY

One very useful feature of the RTL that is worthwhile pursuing during design is its portability to a different hardware circuit: as the circuit chosen at initial design ages and becomes obsolete, one possible option could be to port the application onto a new different circuit. The new hardware circuit must be assessed (and qualified in the case of safety applications), and the implementation would need to be performed again, with all the requirements, but the preceding steps of the lifecycle could be reused.

3.3.5.20 TOOLS

Design languages are usually associated with extensive toolsets that allow:

- Facilitate the creation of new designs and the assessment of existing designs (e.g., pre-developed IP cores).
- Integrate design components coming from different sources.
- Perform static verification (adherence to coding rules, consistency of interfaces, type checking, out of range checking, detection of dead states in finite state machines, detection of side effects in functions or macros, detection of shared objects, etc.).
- Simulate functional behaviours and timings.
- Formally verify the satisfaction of requirements.
- Produce project documentation automatically.

As tools may come from different sources (generic or from circuit vendors), it is important to make sure that they are compatible with one another (can exchange information seamlessly). Also, considering the very long operational lifetimes of typical NPP I&C systems, adherence to stable and widely used standardized formats is an important criterion.

The tools are relatively complex and come from commercial vendors. Thus “qualifying” the tool software, in the sense of demonstrating that it is equivalent in quality to safety-related software, typically is impractical. As a result, tools used for safety applications are typically dedicated, which means they are assessed for quality, experience and maturity, and V&V activities are performed to check the outputs of the tools such that any errors will be detected. EPRI has lot of technical references, some of them endorsed by the Nuclear Regulatory Commission, about dedication processes. The requirements for these tools are generally the same as those that apply to tools used in developing software for safety applications or important to safety, as required by regulators. This implies, among other things, that tools should be assessed for quality and should be subject to version control and configuration management.

It should be noticed that commercial tools traditionally target on speed and/or power consumption of the FPGA, while on the other hand, a safety-related FPGA should perform its intended functions in a predictable and consistent manner with the desired timing.

3.3.6 IMPLEMENTATION

3.3.6.1 OPTIMIZATION

When RTL optimization is performed using automated tools, it is important to make sure that this will not remove or nullify important design features such as redundancy or function independence.

The opposite problem may also exist (e.g., to meet speed constraints or to cope with the saturation of routing channels, tools may duplicate gates). These replications may introduce states which do not exist in the RTL design. Such states occur when the replicates have different outputs, due to events such as asynchrony, metastability, or random fault. Therefore, it is important to analyse the replications to make sure that they are acceptable.

3.3.6.2 SYNTHESIS AND PLACE&ROUTE PARAMETERS AND CONSTRAINTS

Files of parameters and constraints, or directives, are given to the tools to guide the synthesis and place&route operations. Such files specify constraints such as needed operating frequency, timing relations between signals, or fan-out. Following these constraints, the tools may modify the placement to favour a given propagation path at the expense of other ones, duplicate one gate to reduce the load on each copy and thus increase their speed, and so on. The design of these files is a major activity of the implementation process because only the constraints explicitly given to the tools are taken into account. Errors or omissions in these files may result in non-deterministic faults, often not detectable during simulation, and sensitive to normal variations of the microelectronics process (mismatches).

Files of parameters and constraints need to be designed with great care by skilled and experienced electronic design engineers. Their completeness and correctness needs to be documented and verified, and the files should be placed under version control and configuration management.

3.3.7 VERIFICATION

3.3.7.1 TESTING AND SIMULATION

Simulation is run at successive stages of the design (functional specification, RTL, netlist, placement&routing) and with increasing levels of detail. Vendor-supplied toolsets are used

primarily, and of course, for circuit-specific implementation phases. As more detailed simulation requires more resources and computing power, the number of simulation cases tends to decrease while simulation accuracy increases.

One of the benefits of simulation is that it allows unlimited observability of the internal functioning of the simulated circuit, which is extremely limited in the case of real circuit testing. Another benefit is that it allows the injection of various types of faults and the direct assessment of fault tolerance capabilities.

FPGA verification should be performed using hardware generated input stimuli that are identical to the stimuli used in the software-based simulation. By comparing the hardware simulation outputs with the software simulation outputs, one can verify whether the hardware implementation conforms to the initial design requirements both at the module level and at the top level. The input stimuli should include input combinations that are not expected during a normal board/module operation to confirm that there are no unexpected responses.

Functional hardware verification should also include design issues related to the circuitry residing on the same board with the FPGA device. These issues include verifying FPGA device interfaces with external circuitry including proper I/O logic levels, output driving currents, external resets, external clock sources, FPGA power regulators, etc.

3.3.7.2 FORMAL VERIFICATION

Techniques and tools based on rigorous reasoning may be used to verify that new or existing designs have desirable properties. Tool vendors have made very significant progress in recent years.

For microprocessor-based systems, formal verification is more complicated to apply than for FPGAs, as they introduce lower levels of complexity and mainly in the safety functions as they should be implemented in flat hardware logic as much as possible.

Formal verification can be defined as a systematic, rigorous approach to ascertain that a design has specific properties. The systematic approach means that these properties are ascertained based not on specific sets of inputs, but on entire range of inputs.

Formal verification represents a diverse form of assurance that can help in the licensing process about the concerns of the complexities derived from de use of software for the development. Additionally, formal verification is usually employed in IV&V, which introduces another diverse characteristic as the individuals to use this technique are independent from design and review teams.

Because they are very costly, it is recommended that formal verification be focus in safety applications.

Formal verification can be divided into four categories:

- Functional and timing properties: relate the outputs to its inputs.

  Such techniques usually start with a formal specification of the required design properties (using for example languages like property specification language or PSL, standardized in IEEE 1850, which is compatible with VHDL, Verilog or SystemC). Formal verification tools can then systematically check that these properties are indeed satisfied by the design. If not, most tools can help the system designer locate the error.

  Verification of timing properties relies on back-annotations from the synthesis and place&route tools, so functional and timing properties formal verification is not completely independent from the final circuit nor tools.

- Integrity properties: verify that the design does not incorporate any type of intrinsic fault, that is, faults that can be recognized without any knowledge of functions or timings requirements. Example could be non-initialized variables of registers, overflows, etc.

  This technique can help in finding faults that could be difficult to detect with simulation and testing (for instance, counter overflows, that could require very long operation times or very special combination of circumstances before they could result in detectable incorrect behavior).

- Structural properties: assertion that a particular part of the design cannot influence another part. This help in determining that this other part can be verified independently of the first one. Structural properties techniques can also determine that a particular part of the design can influence another part only in specific ways and under specific conditions. This also helps in regression analysis to be made because of later modifications in the design process.

  Structural properties formal verification should focus on most critical parts of the design (safety functions, for instances). Once it has been shown that a non-critical part of the design cannot influence a critical part (for instance, that self-testing capabilities cannot influence safety or primary functions), following verifications can focus only in the critical parts.

- Equivalence properties: ascertain that the results from the synthesis and place&route phases are equivalent to the HDL code that has already been formally verified for other types of properties (functional, timing, integrity and structural). These techniques need obviously the back-annotations from synthesis and place&route tools.

3.3.7.3 STATIC TIMING ANALYSIS

Static timing analysis (STA) may be performed to analyse behaviour in worst and best cases to calculate the margins. The detailed timing information for the circuit (internal propagation times, etc.) is usually provided by the technology libraries and the relevant design tools. Static timing analysis does not require simulation.

The main goal is to verify that all signals will arrive neither too early nor too late, and hence proper circuit operation can be assured. Bad results could be:

- A hold time violation, when an input signal changes too quickly, after the clock's active transition.
- A setup time violation, when a signal arrives too late, and misses the time when it should advance.

Since STA is capable of verifying every path, it can detect other problems like glitches, slow paths and clock skews.

Nevertheless, behaviour of an electronic circuit is often dependent on various factors in its environment like temperature or local voltage variations. In such a case either STA needs to be performed for more than one such set of conditions, or STA must be prepared to work with a range of possible delays for each component, as opposed to a single value. If the design works at each extreme condition, then under the assumption of monotonic behaviour, the design will also work for all intermediate points.

The timing behaviour of the FPGA is guaranteed by the design tools only for a given junction temperature range. The power dissipation of the FPGA should be estimated and later measured to see if additional cooling is needed to keep the junction temperature within the required range.

In static timing analysis, the word static alludes to the fact that this timing analysis is carried out in an input-independent manner, and purports to find the worst-case delay of the circuit over all possible input combinations.

Statistical static timing analysis (SSTA) is a new methodology that is gaining importance to handle the complexities of process and environmental variations in integrated circuits.

3.3.7.4 VERIFICATION OF SYNTHESIS AND PLACE&ROUTE

Errors in synthesis and place&route tools may have a major influence on robustness and even on logic correctness. Tool selection and if necessary, rules to restrict their use to only a sub-set of tool functionality of adequate quality may be used as a first means of defense.

Synthesis tools optimize digital logic described in HDL code and produce either textual outputs such as electronic design interface format (EDIF) files or graphical (schematic) outputs. Most synthesizers generate FPGA-independent schematic representation of the HDL code as well as the FPGA-specific schematic representation. The FPGA-independent schematic uses AND, OR, and NOT gates and flip-flops to represent the design, while the FPGA-specific schematic utilizes look-up tables to replace the combinatorial logic and different macros to represent higher-level functional blocks such as memories, multipliers, adders, etc. To be able to understand what possible optimizations and design changes are applied during the synthesis, designers and reviewers should carefully examine the synthesis report documents generated by the synthesis

tool. Usually, the warnings in the report may give important information about how the synthesis tool understands the HDL code and, thereby, whether the original designer's intent remained unchanged.

Verification of the tool results, preferably with the support of other independent tools, may also be performed to ensure that the results of synthesis, and then of place&route, are cycle-by-cycle equivalent to the RTL, including the initialization steps.

THIS PAGE INTENTIONALLY LEFT BLANK.

4 PRACTICAL APPLICATION OF STUDY RESULTS: SPECIFICATION FOR A DIESEL LOAD SEQUENCER BASED ON FIELD PROGRAMMABLE GATE ARRAYS

Following is a template for a specification to tender for a diesel load sequencer (DLS) upgrade project for an operating plant based on a new FPGA-based system.

This specification has been issued taking into account all recommendations and guidelines provided in the present study.

Text/number in brackets [ ]/( ) are examples that should be customized by each plant according to its specific circumstances and requirements.

4.1 REASON FOR REPLACEMENT

Main reasons for replacement are [aging of the current system and obsolescence issues].

4.2 SCOPE OF SUPPLY

The scope of supply for the replacement system includes [2 trains] of Diesel Load Sequencer System. A Maintenance and Training System (MTS) [one train] shall be incorporated as optional.

Additionally, reuse of existing cabinets or installation of new ones shall be evaluated. In either case, current field cables shall be reuse to minimize project impact, costs, schedule, installation activities and uncertainties. This shall be taken into account in the final layout proposal of components inside the cabinets.

Support for licensing and regulatory approval of the project.

Support for plant installation and post-installation testing activities.

4.3 PROJECT SCHEDULE

Project schedule shall be established so that installation is performed during a normal plant refuelling outage without impacting outage schedule.

Two options shall be evaluated. Installation of one train first and the second one in a later outage or installation of both trains in the same outage. Uncertainty and risk evaluation, as well as cost-benefit study shall be performed for both options.

4.4 CODES AND STANDARDS

Latest revisions of following codes and standards shall be used.

A. Code of Federal Regulations:

- 10CFR50.49 Environmental qualification of electric equipment important to safety for nuclear power plants.
- 10CFR50.55 Conditions of construction permits, early site permits, combined licenses, and manufacturing licenses.
- 10CFR50.59 Changes, tests and experiments.
- 10CFR50 Ap. A General Design Criteria for Nuclear Power Plants.
- 10CFR50 Ap. B Quality Assurance Criteria for Nuclear Power Plants and Fuel Reprocessing Plants.
- 10CFR73.54 Protection of digital computer and communication systems and networks.
- 10CFR21 Reporting of defects and noncompliance.

B. Nuclear Regulatory Commission Regulatory Guides:

- RG 1.152 Criteria for Use of Computers in Safety Systems of Nuclear Power Plants.
- RG 1.153 Criteria for Safety Systems.
- RG 1.168 Verification, Validation, Reviews, and Audits for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.
- RG 1.169 Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.
- RG 1.170 Software Test Documentation for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.
- RG 1.171 Software Unit Testing for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.
- RG 1.172 Software Requirements Specifications for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.
- RG 1.173 Developing Software Life Cycle Processes for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.
- RG 1.180 Guidelines for Evaluating Electromagnetic and Radio-Frequency Interference in Safety-Related Instrumentation and Control Systems.
- RG 1.187 Guidance for Implementation of 10 CFR 50.59, Changes, Tests, and Experiments.
- RG 1.22 Periodic Testing of Protection System Actuation Functions.

- RG 1.47 Bypassed and Inoperable Status Indication for Nuclear Power Plant Safety Systems.
- RG 1.53 Application of the Single-Failure Criterion to Nuclear Power Plant Protection Systems.
- RG 1.75 Physical Independence of Electric Systems.
- RG 1.89 Environmental Qualification of Certain Electric Equipment Important to Safety for Nuclear Power Plants.
- RG 1.100 Seismic Qualification of Electric and Mechanical Equipment for Nuclear Power Plants.
- RG 1.118 Periodic Testing of Electric Power and Protection Systems.
- RG 5.71 Cyber Security Programs for Nuclear Facilities.

C. Nuclear Regulatory Commission Interim Staff Guidance:

- DI&C-ISG-01 Cyber Security.
- DI&C-ISG-02 Diversity and Defense-in-Depth.
- DI&C-ISG-03 Risk-Informed Digital Instrumentation and Controls.
- DI&C-ISG-06 Licensing Process.

D. Other Nuclear Regulatory Commission documents:

- RIS 2002-22 Use of EPRI/NEI Joint Task Force Report, "Guideline on Licensing Digital Upgrades: EPRI TR-102348, Revision 1, NEI 01-01: a Revision of EPRI TR-102348 to Reflect Changes to the 10 CFR 50.59 Rule".
- Safety Evaluation by the Office of Nuclear Reactor Regulation Related to Amendment No. 181 to Renewed Facility Operating License No. NPF-42, Wolf Creek Nuclear Operating Corporation, Wolf Creek Generating Station, Docket No. 50-482, ML090610317. Nuclear Regulatory Commission, Washington, DC: 2009.
- NUREG 800 Standard Review Plan, Chapter 7 Instrumentation and Controls.
    - BTP 7-14 Guidance on Software Reviews for Digital Computer-Based Instrumentation and Control Systems.
    - BTP 7-17 Guidance on Self-Test and Surveillance Test Provisions.
    - BTP 7-18 Guidance on the Use of Programmable Logic Controllers in Digital Computer-Based Instrumentation and Control Systems.
    - BTP 7-19 Guidance for Evaluation of Diversity and Defense-in-Depth in Digital Computer-Based Instrumentation and Control Systems.
    - BTP 7-21 Guidance on Digital Computer Real-Time Performance.

- NUREG 800 Standard Review Plan, Chapter 13.6.6 Cyber Security Plan.
- NUREG CR-5930 High integrity Software Standards and Guidelines.
- NUREG CR-6294 Design Factors for Safety-Critical Software.
- NUREG CR-6303 Method for Performing D-in-D&D Analysis for Reactor Protection System.
- NUREG CR-6463 Review Guidelines for Software Languages for Use in Nuclear Power Plants Safety Systems.
- NUREG CR-7006 Review Guidelines for FPGAs in Nuclear Power Plant Safety Systems.
- NUREG CR-7007 Diversity Strategies for Nuclear Power Plants Instrumentation and Control Systems.
- NUREG 700 Human-System Interface Design Review Guidelines.

E. Institute of Electrical and Electronic Engineers:

- IEEE 338 Criteria for the Periodic Surveillance Testing of Nuclear Power Generating Station Safety.
- IEEE 379 Application of the Single-Failure Criterion to Nuclear Power Generating Station Safety Systems
- IEEE 384 Criteria for Independence of Class 1E Equipment and Circuits
- IEEE 603 Criteria for Safety Systems for Nuclear Power Generating Stations.
- IEEE 467 Quality Assurance Program Requirements for the Design and Manufacture of Class 1E Instrumentation and Electrical Equipment for Nuclear Power Plants.
- IEEE 1050 Guide for Instrumentation and Control Equipment Grounding in Generating Stations
- IEEE 7-4.3.2 Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Plants.
- IEEE 730 Software Quality Assurance Plans.
- IEEE 828 Standard for Software Configuration Management Plans.
- IEEE 829 Standard for Software Test Documentation.
- IEEE 830 Guide to Software Requirement Specification.
- IEEE 1008 Standard for Software Unit Testing.
- IEEE 1012 Standard for Software Verification and Validation Plans.
- IEEE 1016 Recommended Practices for Software Design Description.

- IEEE 1028 Standard for Software Review and Audits.
- IEEE 1042 Guide to Software Configuration Management.
- IEEE 1063 Standard for Software User Documentation.
- IEEE 1074 Standard for Developing Software Life Cycle Processes.
- IEEE 1219 Standard for Software Maintenance.
- IEEE 1228 Standard for Software Safety Plans.

F. International Electrotechnical Commission:

- IEC 61226 Nuclear Power Plants – Instrumentation and Control Important to Safety – Classification of Instrumentation and Control Functions.
- IEC 61508 Functional Safety of Electrical/Electronic/Programmable Electronic Safety Related Systems.
- IEC 61513 Nuclear Power Plants – Instrumentation and Control Important to Safety – General Requirements for Systems.
- IEC 62566 Nuclear Power Plants – Instrumentation and Control Important to Safety – Development of HDL-programmed Integrated Circuits for Systems Performing Category A Functions.
- IEC 60880 Software for Computers in the Safety Systems of Nuclear Power Plants (parts 1 and 2).
- IEC 60987 Programmed Digital Computers Important to Safety in Nuclear Power Plants.
- IEC 61225 Nuclear Power Plants – Instrumentation and Control Important to Safety – Requirements for Electrical Supplies.
- IEC-61000 series – Electromagnetic Compatibility.
- IEC 27000 series – Information Technology.

G. International Society of Automation:

- ISA/IEC 62443 (former S99 Series) Cibersecurity.

H. Nuclear Energy Institute:

- NEI 96-07 Guidelines for 10CFR50.59 implementation.
- NEI 08-09 Cyber Security Plan for Nuclear Power Reactors.
- NEI 10-04 Scope of Systems for the NRC Cyber Security 10 CFR 73.54 and FERC Order 706-B Compliance.
- NEI 10-09 Cyber Security Common Controls for Nuclear Power Reactors

I. National Institute of Standards and Technology:

- SP800-53 Recommended Security Controls for Federal Information Systems and Organizations.
- SP800-82 Guide to Industrial Control Systems (ICS) Security.

J. Electric Power Research Institute:

- EPRI TR 103291 Handbook for V&V of Digital Systems.
- EPRI TR 106439 Guideline on Evaluation and Acceptance of Commercial Grade Digital Equipment for Nuclear Safety Applications.
- EPRI TR 107339 Evaluating commercial digital equipment for high integrity applications: a supplement to EPRI TR 106439.
- EPRI TR 102348 Guidelines on licensing digital upgrades.
- EPRI TR 1011710 Handbook for evaluating critical digital equipment and systems.
- EPRI TR 1002835 Guidelines for performing defense-in-depth and diversity assessments for digital upgrades: applying risk-informed and deterministic methods.
- EPRI TR 104595 Abnormal conditions and events analysis for I&C systems.
- EPRI TR 108831 Requirements engineering for digital upgrades.
- EPRI TR 1019182 Protecting Against Digital Common-Cause Failure.
- EPRI TR 1021077 Estimating Failure Rates in Highly Reliable Digital Systems.
- EPRI TR 1001045 Guidelines on the use of pre-qualified digital platforms for safety and non-safety applications on NPP´s.
- EPRI 1019187 Technical Guideline for Cyber Security Requirements and Life-Cycle Implementation Guidelines for Nuclear Plant Digital Systems.
- EPRI TR 1023502 Cybersecurity Procurement Benchmark.

K. Military Standards and Guidelines:

- DOD-STD-1701 Hardware Diagnostic Test System Requirements.
- MIL-STD-461 Testing for Electromagnetic Compatibility.
- MIL-STD-1686 Electrostatic Discharge Control Program for Protection of Electrical and Electronic Parts, Assemblies and Equipment.
- MIL-HDBK-263 Electrostatic Discharge Control Handbook for Protection of Electrical and Electronic Parts, Assemblies and Equipment.
- MIL-HDBK-338 Electronic Reliability Design Handbook.
- MIL-HDBK-344 Environmental Stress Screening of Electronic Equipment.

- MIL-STD-756 Reliability Modeling and Prediction.
- MIL-STD-1629 Procedures for Performing a Failure Mode, Effects, and Criticality Analysis.
- MIL-STD-2164 Environment Stress Screening Process for Electronic Equipment.

4.5 SYSTEM DESCRIPTION AND REQUIREMENTS

The DLS consists of [two redundant Trains, designated A and B]. Each Train is physically, functionally, and electrically separated from each other and are electrically isolated from non-safety systems. Redundant Trains are provided to satisfy single failure criteria and to improve plant availability.

The DLS provides the equipment that is necessary to perform a sequential start of the required plant vital bus loads in response to a Safety Injection (SI) actuation and/or Loss of Offsite Power (LOOP), with subsequent restoration of power to the vital buses. Specifically, the DLS includes a Safety Injection Sequencer (SIS) and a BlackOut Sequencer (BOS), each of which consists of its associated field inputs, actuation logic, timed step outputs, Operator Lockout (OL) outputs, and Automatic Lockout (AL) outputs.

The DLS also includes maintenance, test, and diagnostic functions that are used to verify proper system operation, including Plant Technical Specifications surveillance procedures.

Figure 16 represents a proposal for basic system architecture for one Train of the upgraded system.

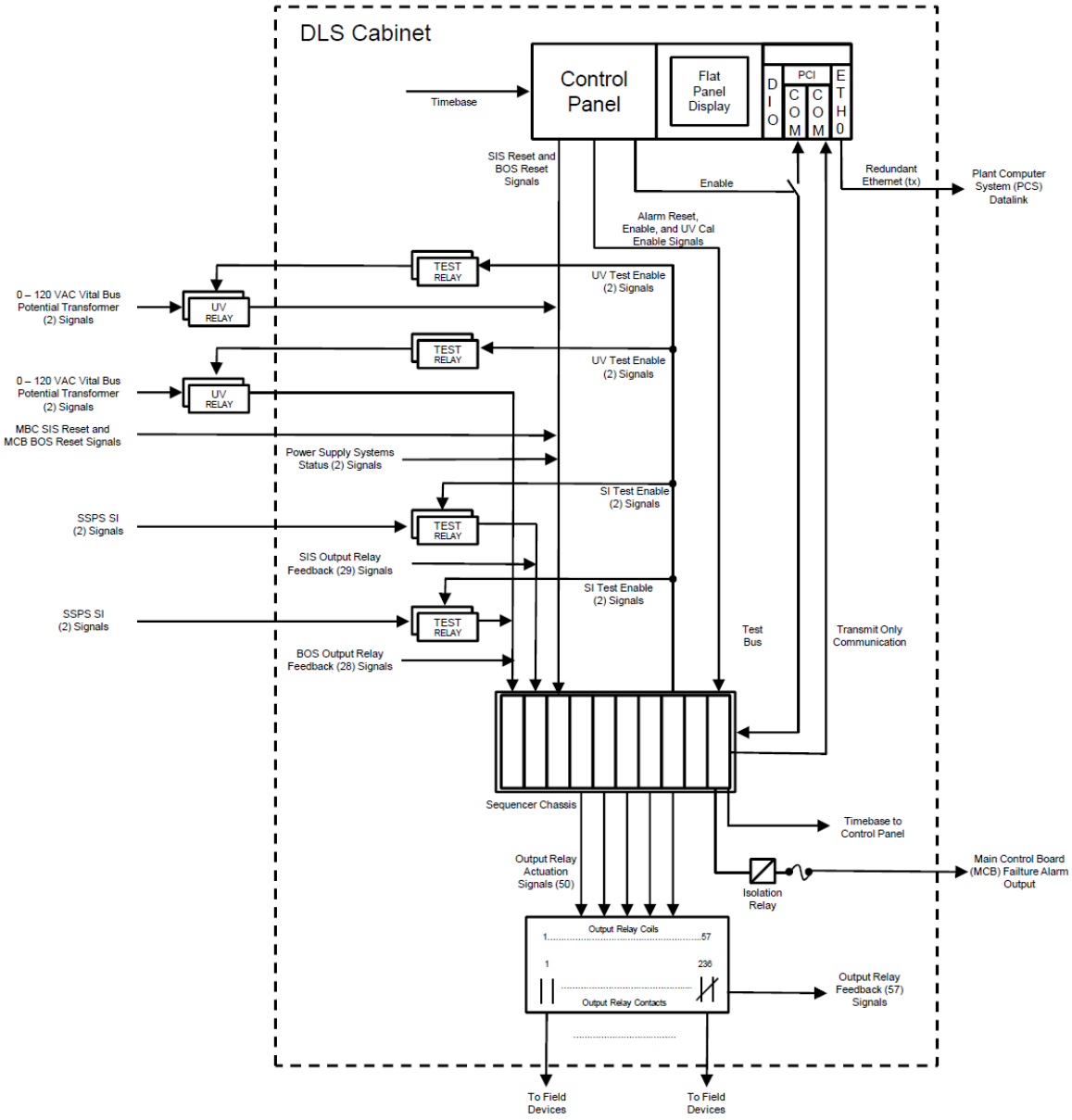

Figure 16 – Basic upgraded diesel load sequencer architecture (proposal)

Major components include:

- Sequencer Chassis

  The design utilizes a single rack that includes all Inputs/Outputs (I/O), logic, and communication hardware necessary to perform the Safety System Functions.

- Maintenance and Engineering Unit (M&EU)

  The Safety System Auxiliary Functions (monitoring and test) are implemented using an M&EU. The M&EU consists of a Process Computer (PC), Flat Panel Display (FPD), keyboard with integrated trackball, Plant Computer System (PCS) datalink (including Trouble Alarm), and Control Panel. It provides the Human-System Interface (HSI) for the system.

- System Communication

  The DLS includes the following communications interfaces:

  - One unidirectional (Chassis to M&EU) RS-422 (tx) communication bus, designated as Transmit Only Communication Bus 1 (TxB1). It is active at all times and continuously transmits system data from the Chassis to the M&EU for display.
  - One switched bidirectional (between Chassis and M&EU) RS-485 (tx/rx) communication bus, designated as the Test Bus (TB). It is active only during testing activities, which requires activation of the TB Enable Keyswitch in the Control Panel.

- System Inputs
  - Four (4) Undervoltage (UV) inputs (normally open mechanical dry contacts from the plant UV relays).
  - Four (4) SI inputs (normally open mechanical dry contacts from Protection System).
  - One (1) normally open mechanical dry contact from the Main Control Board (MCB) SIS Reset handswitch.
  - One (1) normally open mechanical dry contact from the MCB BOS Reset handswitch.

- System Outputs
  - (179) output relay mechanical dry contacts for actuation of plant equipment.
  - One normally open mechanical dry contact for actuation of the MCB Failure Alarm. This output is implemented in a fail-safe (de-energize to alarm) configuration.
  - Unidirectional Ethernet (tx) Plant Computer System (PCS) datalink. The PCS datalink transmits data to the PCS through redundant fiber-optic cables that provide the Class 1E electrical isolation barrier between the DLS and the non-safety PCS.

4.5.1 FUNCTIONAL LOGIC

Functional logic to be implemented is represented in functional diagrams (Figure 17 to Figure 25).

SYMBOL	FUNCTION	DESCRIPTION
	OR	A FUNCTION THAT PRODUCES A TRUE OUTPUT WHEN AT LEAST ONE INPUT IS TRUE.
	EXCLUSIVE OR	A FUNCTION THAT PRODUCES A TRUE OUTPUT WHEN ONLY ONE INPUT IS TRUE (USED FOR THE OUTPUT RELAY MISMATCH FUNCTION).
	NOT	A FUNCTION THAT PRODUCES A TRUE OUTPUT WHEN THE INPUT IS FALSE OR A FALSE OUTPUT WHEN THE INPUT IS TRUE.
	AND	A FUNCTION THAT PRODUCES A TRUE OUTPUT ONLY WHEN EVERY INPUT IS TRUE.
2/4	COINCIDENCE (2 OUT OF 4 SHOWN)	A FUNCTION THAT PRODUCES A TRUE OUTPUT WHEN THE PRESCRIBED NUMBER OF INPUTS (OR MORE) IS TRUE. FOR EXAMPLE SHOW, 2 OR 3 OR 4 INPUTS MUST BE TRUE FOR THE OUTPUT TO BE TRUE.
S R OR R S	SET / RESET MEMORY (RESET PRIORITY)	A FUNCTION THAT PRODUCES A TRUE OUTPUT WHEN THE SET INPUT IS TRUE WHILE AT THE SAME TIME THE RESET INPUT IS FALSE. WHEN THE RESET INPUT IS TRUE THE OUTPUT IT IS UNCONDITIONALLY FALSE.
S R OR R S	SET / RESET MEMORY (SET PRIORITY)	A FUNCTION THAT PRODUCES AN UNCONDITIONALLY TRUE OUTPUT WHEN THE SET INPUT IS TRUE. WHEN THE RESET INPUT IS TRUE WHILE AT THE SAME TIME THE SET INPUT IS FALSE THE OUTPUT IS FALSE.
TD ON TS	TIME DELAY ON	A FUNCTION THAT PRODUCES A TRUE OUTPUT FOLLOWING A DEFINITE INTENTIONAL TIME DELAY OF T SECONDS AFTER THE INPUT BECOMES TRUE. THE OUTPUT BECOMES FALSE INMEDIATELY IF THE INPUT BECOMES FALSE.
TD OFF TS	TIME DELAY OFF	A FUNCTION THAT PRODUCES A FALSE OUTPUT FOLLOWING A DEFINITE INTENTIONAL TIME DELAY OF T SECONDS AFTER THE INPUT BECOMES FALSE. THE OUTPUT BECOMES TRUE INMEDIATELY IF THE INPUT BECOMES TRUE.
RESET	SOFT CONTROL	A SOFT CONTROL THAT PRODUCES A MOMENTARY TRUE OUTPUT THAT REMAINS TRUE FOR A PERIOD OF TIME THAT IS SUFICIENT TO ALLOW PROCCESSING BY THE RECEIVING LOGIC BEFORE TRANSITIONING TO FALSE.
ACTUATE MOMENTARY	HARD CONTROL (MOMENTARY)	A HARDWARE SWITCH THAT PRODUCES A TRUE OUPUT AS LONG AS THE SWITCH IS MANUALLY HELD IN THE ACTUATE POSITION.
ACTUATE	HARD CONTROL (MAINTAINED)	A HARDWARE SWITCH THAT PRODUCES A TRUE OUPUT FOR AS LONG AS THE SWITCH REMAINS IN THE ACTUATE POSITION.
DI	DIGITAL INPUT	
DI	DIGITAL INPUT (INVERTED)	
DOL	DIGITAL OUTPUT LOGIC	
DO	DIGITAL OUTPUT	
K2x	OUTPUT RELAY	A HARDWARE DEVICE THAT IS USED TO ACTUATE PLANT EQUIPMENT.
A	TROUBLE EVENT INDICATOR	
A	FAILURE EVENT INDICATOR	
I	VARIABLE INDICATOR	
S	STATUS INDICATOR	
C	COMPUTER POINT	A SYSTEM LOGIC VALUE OR STATUS THAT IS ISOLATED AND MADE
TE VO	SI TEST RELAY	A HARDWARE DEVICE THAT PRODUCES A TRUE TEST OUPUT (TO) SIGNAL WHEN THE TEST ENABLE (TE) SIGNAL IS TRUE.
TS PT TE VO	UV TEST RELAY	A HARDWARE DEVICE THAT SELECTS THE VOLTAGE SIGNAL THAT IS INPUT TO THE SYSTEM, WHEN THE TEST ENABLE (TE) SIGNAL IS FALSE, THE POTENTIAL TRANSFORMER (PT) VOLTAGE INPUT IS PASSED TO THE VOLTAGE OUTPUT (VO). WHEN THE TEST ENABLE (TE) SIGNAL IS TRUE, THE TEST SIGNAL (TS) VOLTAGE INPUT IS PASSED TO THE VOLTAGE OUTPUT (VO).
	POTENTIAL TRANSFORMER	A HARDWARE DEVICE THAT PROVIDES A 0-120 VAC SIGNAL TO THE SSSS. THIS SIGNAL IS REPRESENTATIVE OF THE 0-7,200 VAC VITAL BUS VOLTAGE.
UV RELAY	UNDERVOLTAGE RELAY	A HARDWARE DEVICE THAT MONITORS AN INCOMING VOLTAGE SIGNAL (0-120 VAC) AND PROVIDES A TIME-DELAYED TRUE OUTPUT WHEN THE VOLTAGE SIGNAL IS REDUCED BELOW THE DROPOUT SETPOINT. WHEN THE VOLTAGE SIGNAL RISES ABOVE THE PICKUP SETPOINT, THE OUTPUT RESETS TO FALSE.

Figure 17 – Diesel load sequencer functional logic diagram symbols (Sheet 1)

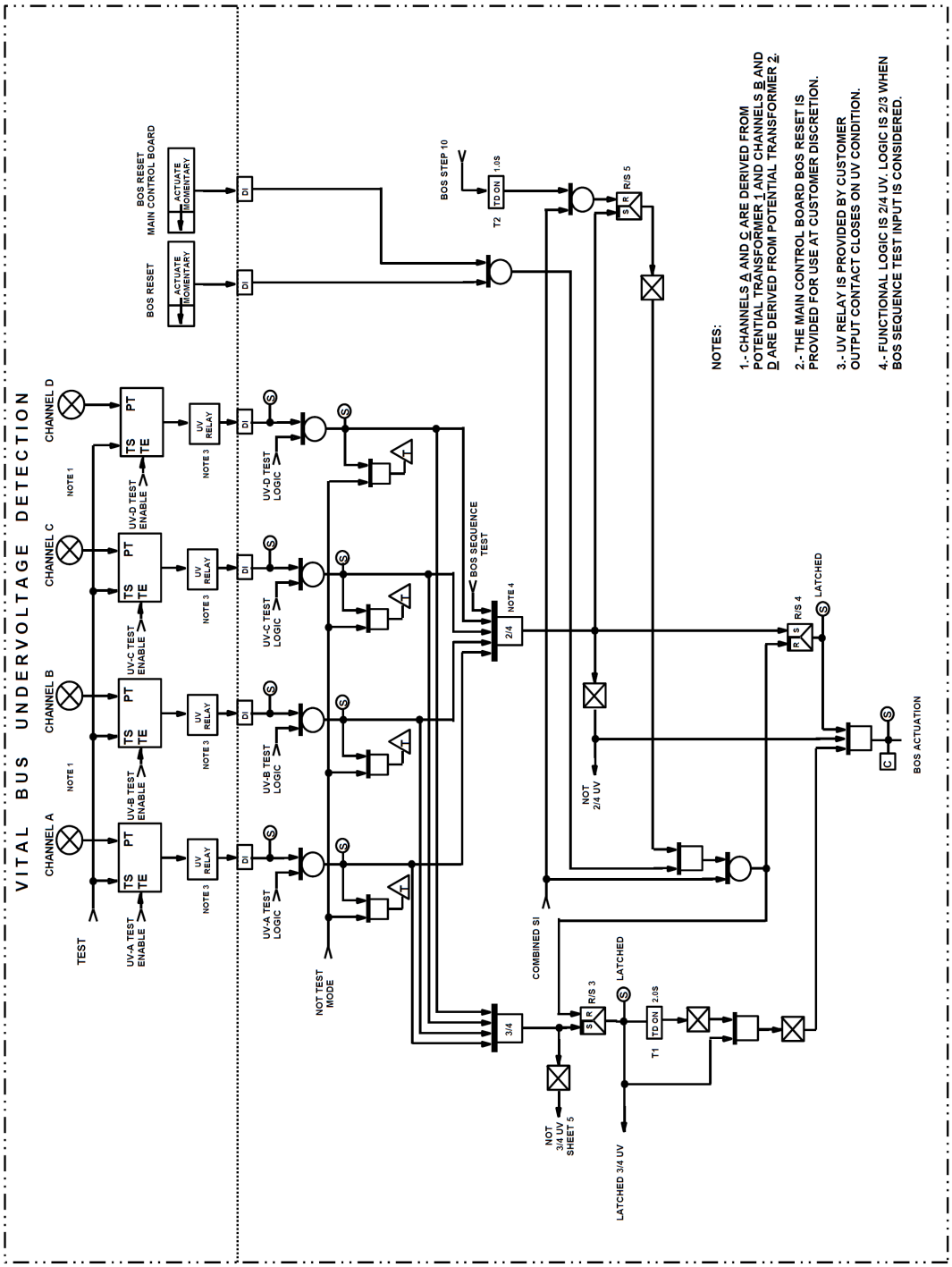

Figure 18 – Diesel load sequencer functional logic diagram for BOS actuation logic (Sheet 2)

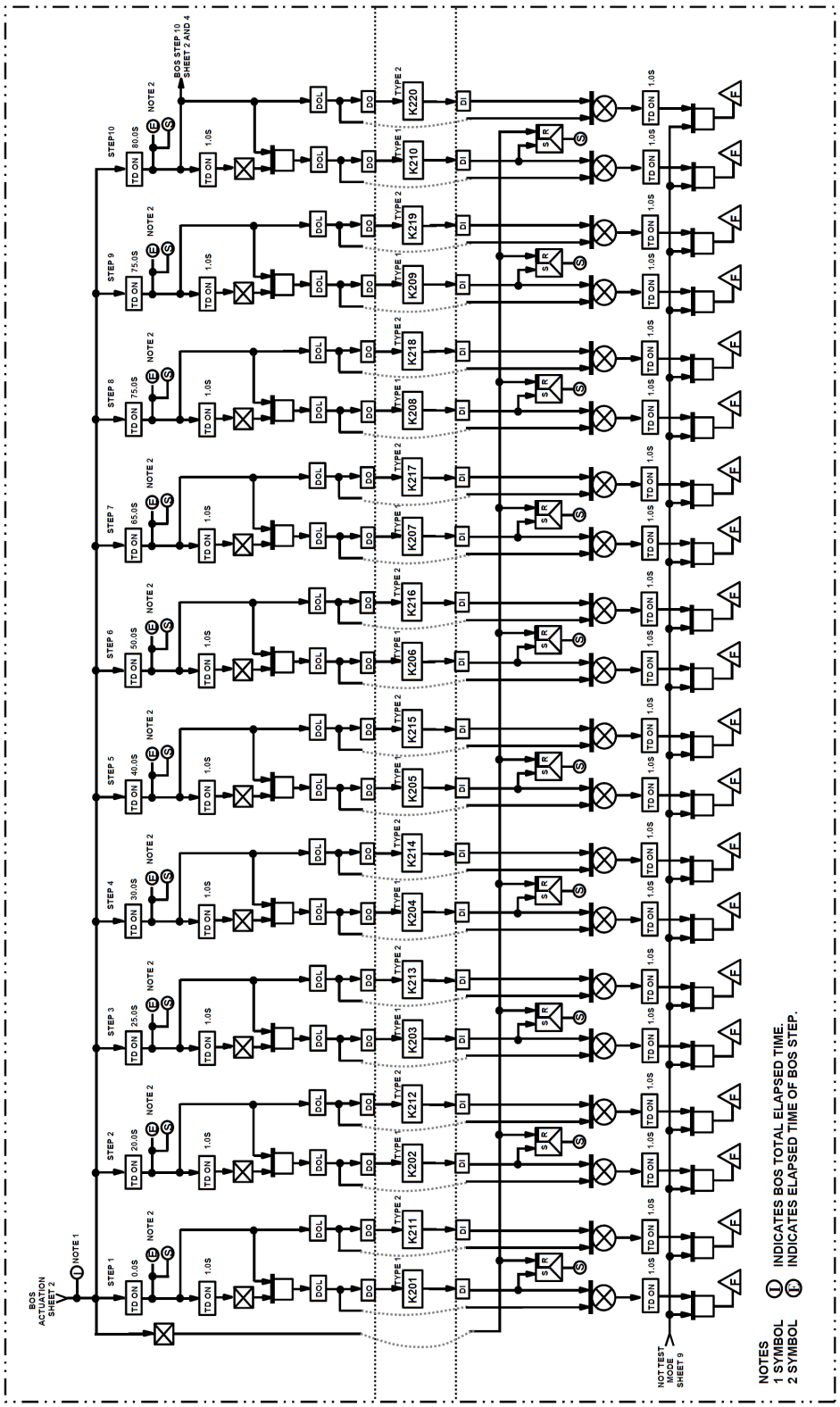

Figure 19 – Diesel load sequencer functional logic diagram for BOS step logic (Sheet 3)

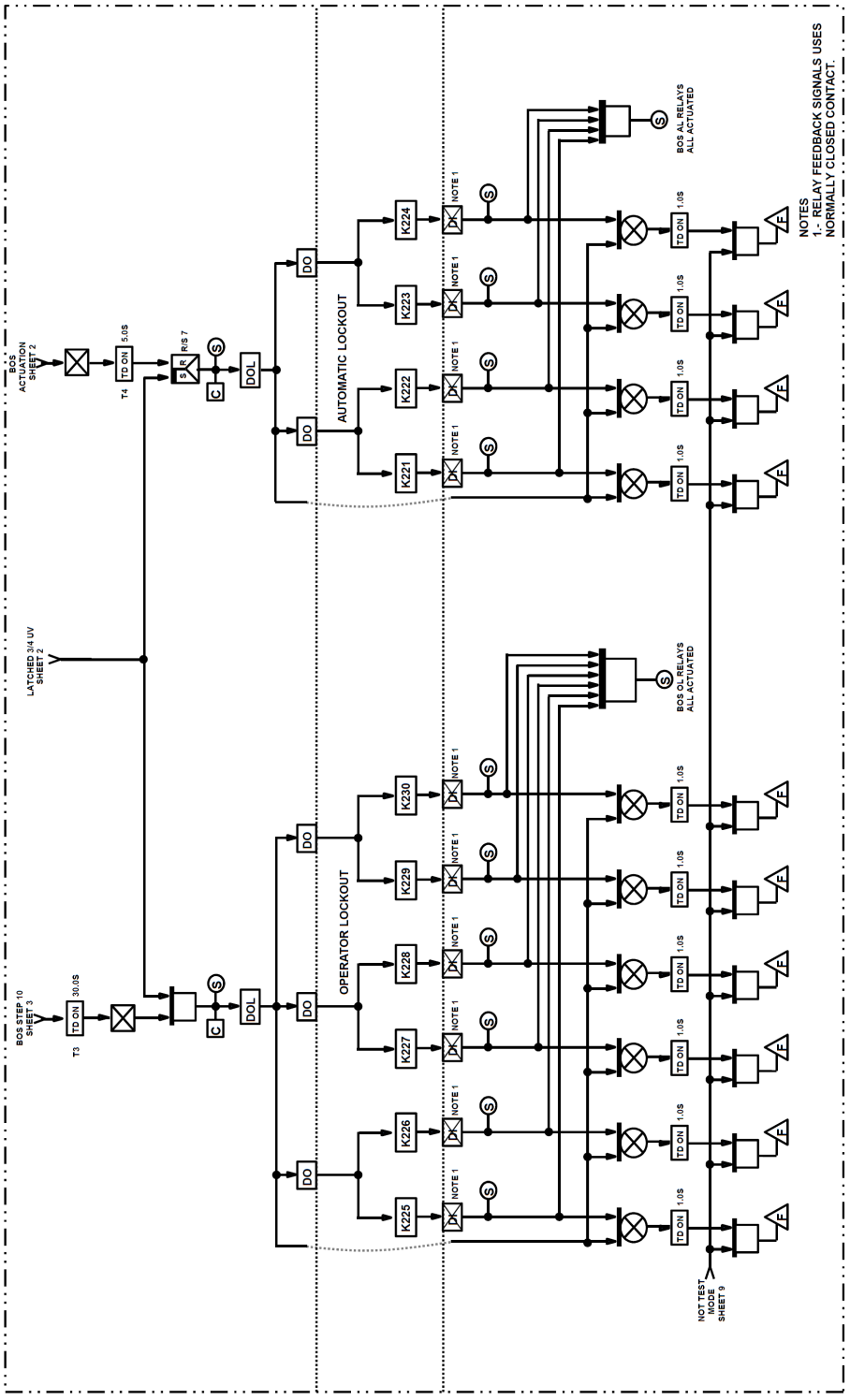

Figure 20 – Diesel load sequencer functional logic diagram BOS lockout logic (Sheet 4)

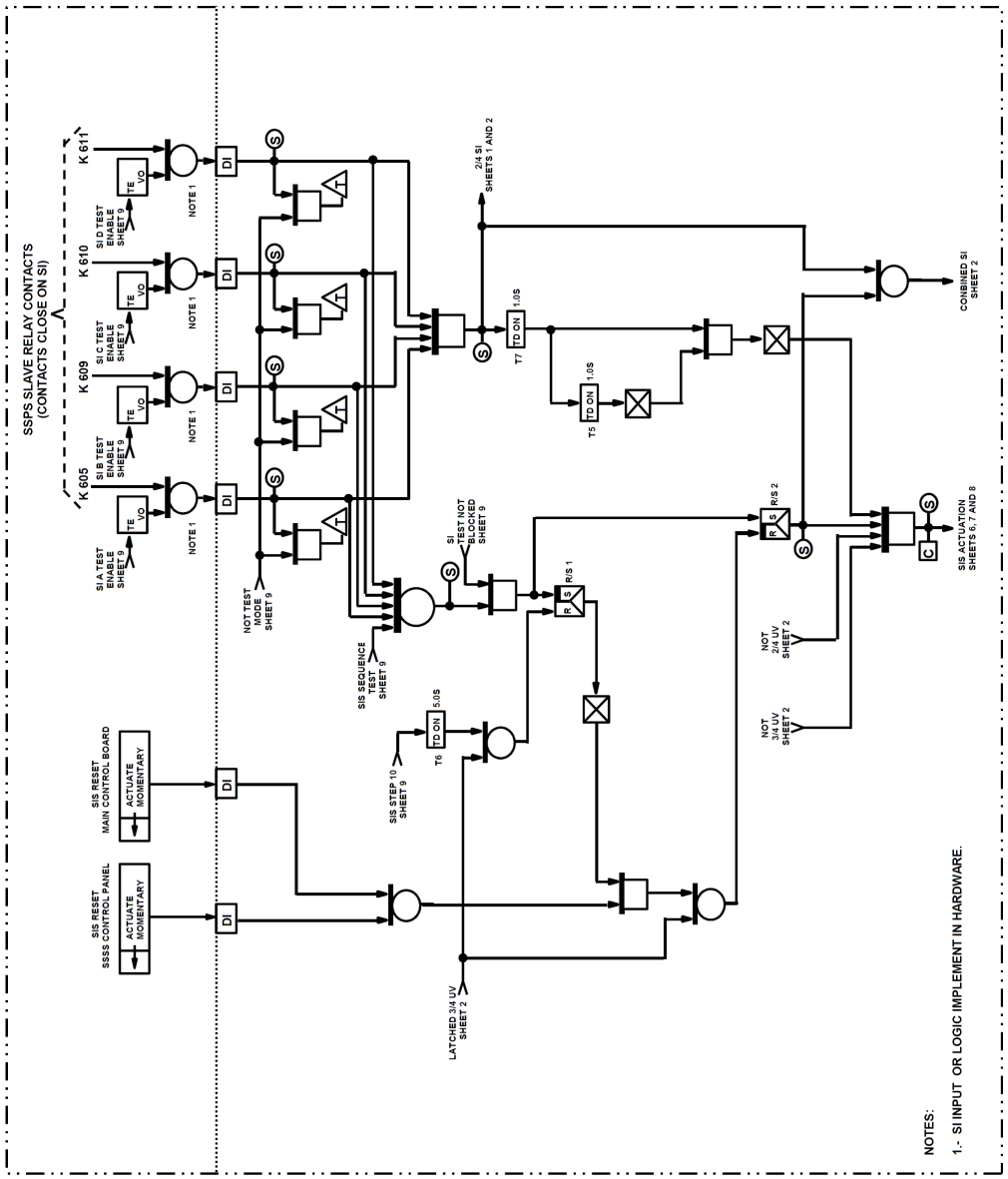

Figure 21 – Diesel load sequencer functional logic diagram SIS actuation logic (Sheet 5)

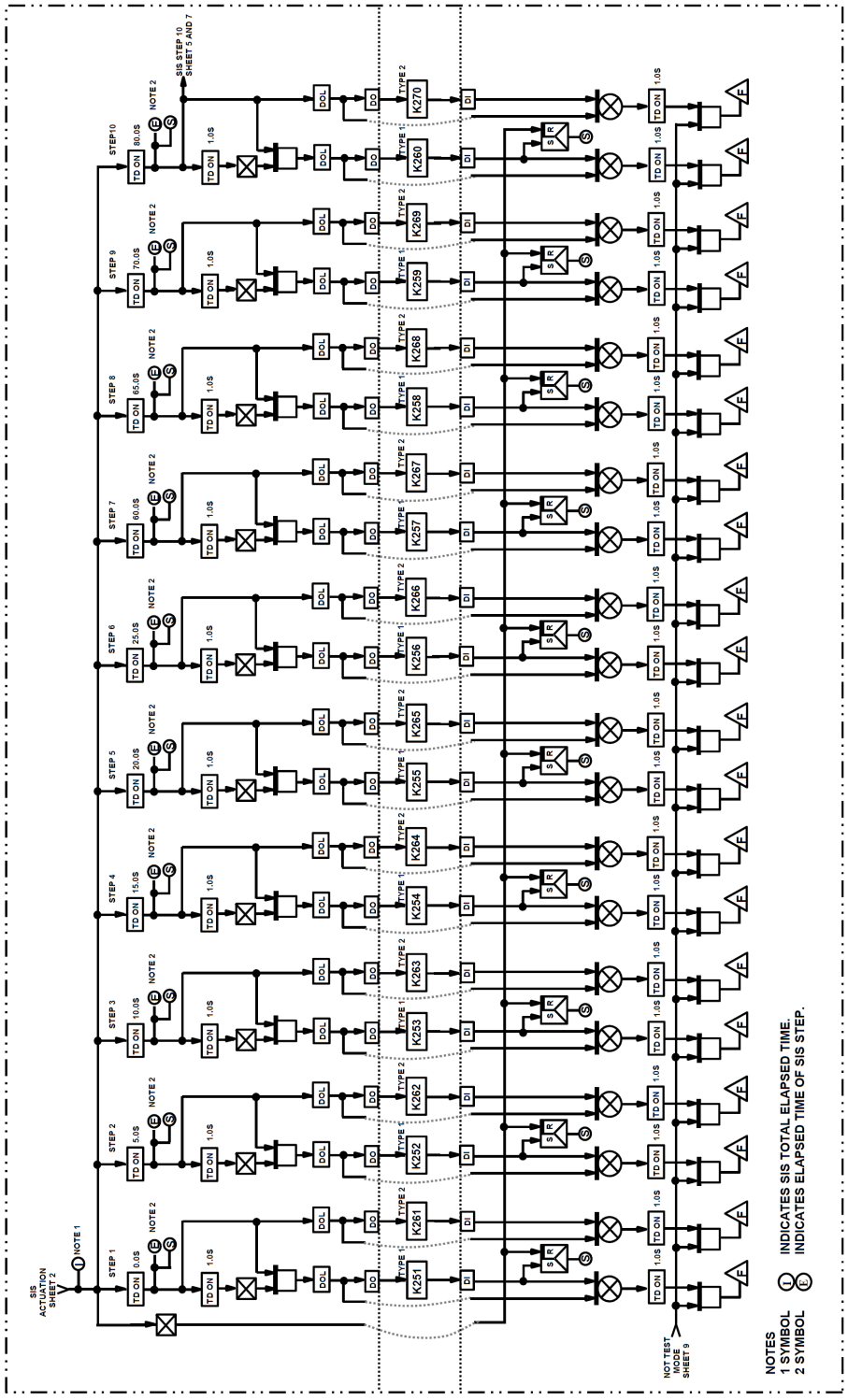

Figure 22 – Diesel load sequencer functional logic diagram SIS step logic (Sheet 6)

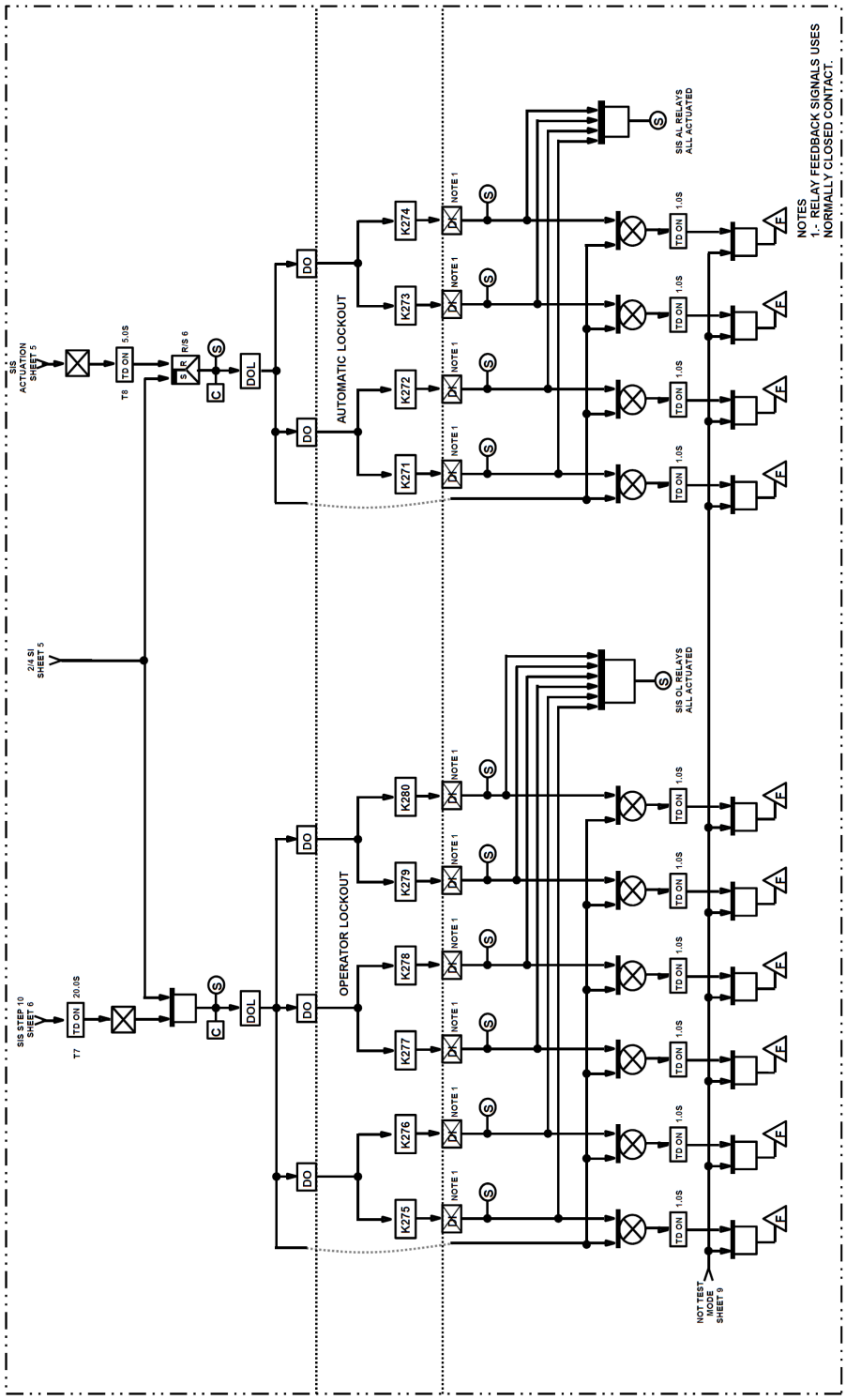

Figure 23 – Diesel load sequencer functional logic diagram SIS lockout logic (Sheet 7)

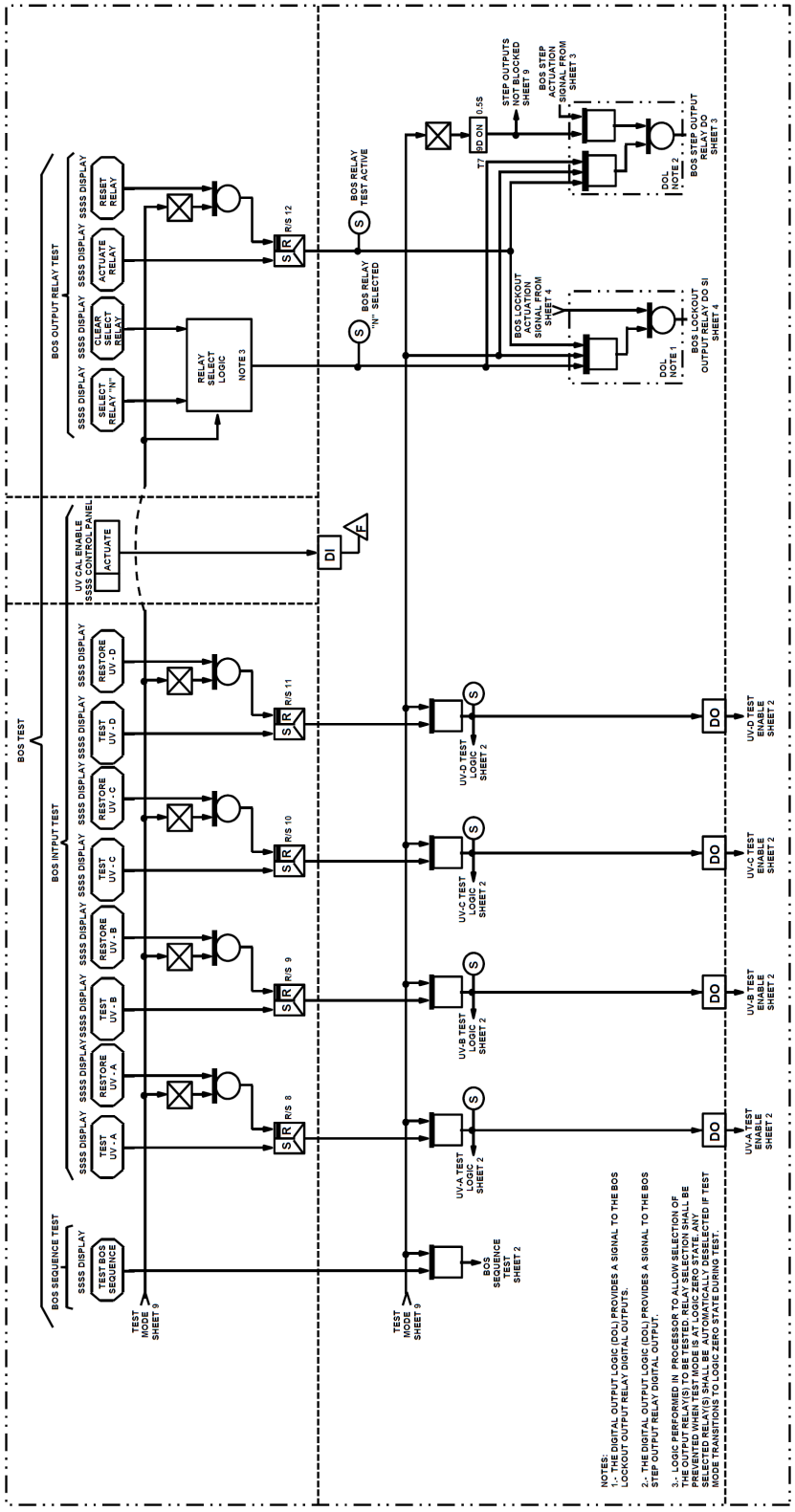

Figure 24 – Diesel load sequencer functional logic diagram BOS test logic (Sheet 8)

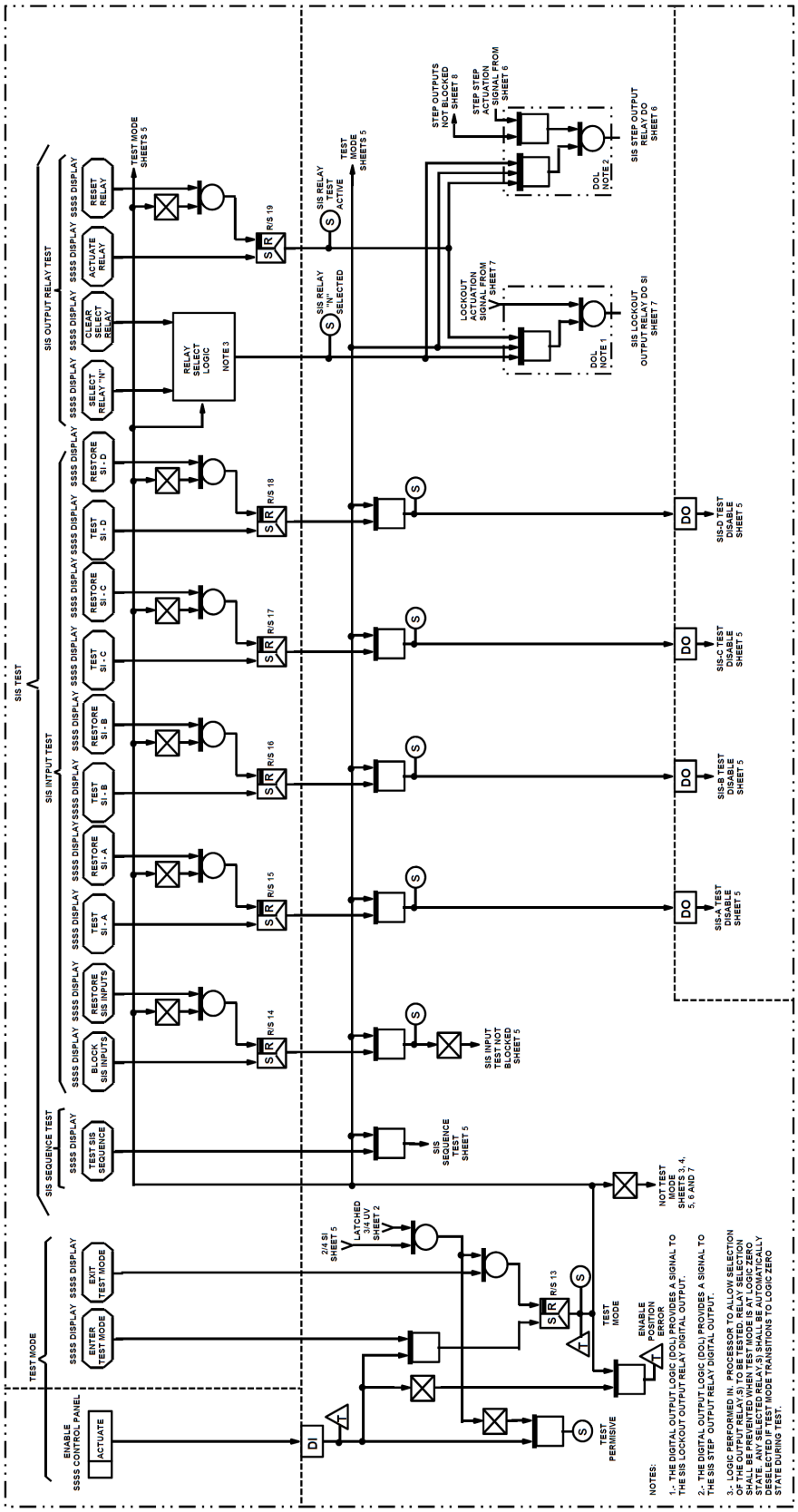

Figure 25 – Diesel load sequencer functional logic diagram test SIS test logic (Sheet 9)

Displays for M&EU shall include the following:

- Operational status monitoring.
- Surveillance testing.
- System status monitoring.
- Corrective maintenance.

Applicable Human Factor Engineering shall be applied in the specification and design of the displays (NUREG 700 and 711).

FPD could be a touch screen type. Nevertheless, keyboard and mouse/trackball is requested.

Navigation from any display to another shall require no more than three (3) operations.

4.5.2 DIAGNOSTICS

System-level and module-level self-diagnostics are required. Trouble and/or Failure Alarms shall be initiated, as appropriate, when faults are detected. Differentiation between Trouble and Failure shall be as follows:

- Failure Alarms indicate conditions that render the system Inoperable. These alarms shall be annunciated on the MCB as they require immediate operator attention. Example of this type could be a failure in a non-redundant logic module.
- Trouble Alarms indicate that the system is in a degraded, but Operable, condition. These alarms could be annunciated on the PCS as they do not require an urgent operator response. Periodic maintenance activities or surveillance testing of the system is considered enough to detect and correct these conditions. Example of this type could be the loss of a redundant power supply.

Both types of alarms shall be annunciated locally in the M&EU.

A detailed description of the self-diagnostic features and classification of alarms shall be provided. In any case, system diagnostic shall include, as a minimum:

- System-level diagnostics:
    - Power supply and Direct Current (DC) distribution breaker status monitoring – Trouble or Failure Alarm depending on partial or total loss of power supply.
    - Output relay mismatch monitoring, comparing each output relay demand actuation signal to the associated output relay feedback signal(s) – Failure Alarm.
- Logic modules self-diagnostics, perform on a timely basis (5 minutes or less) to verify the integrity of the digital input module circuits, control logic module circuits and relay

output driver module circuits (including the integrity of the associated test and output relay coils).

- Additionally, M&EU shall perform continuous self-diagnostics, including:
    - Verification of TxB1 integrity.
    - Verification of TB integrity (when the bus is active).
    - Verification that all required processes are running.
    - Checksum verification.
    - Processor diagnostics.
    - Temperature monitoring.

4.5.3 TESTING

Testing capabilities shall be provided for the following:

- SIS Input Test.
- BOS Input Test.
- SIS Sequence Test.
- BOS Sequence Test.
- SIS Output Relay Test.
- BOS Output Relay Test.
- Integrated Test.

The DLS shall be placed in Test Mode prior to performing any of the activities specified above. Placing the DLS in Test Mode shall require a two (2) step process that involves both hard and soft controls. The TB Enable Keyswitch shall be placed in the Enable position to activate the bidirectional TB and enable the M&EU soft controls. The DLS shall be then placed in Test Mode via the M&EU soft controls and logic.

Placing the DLS in Test Mode shall automatically block actuation of the SIS and BOS step output relays to prevent them from actuating during testing.

If a real SI or LOOP event is detected while in Test Mode, the DLS shall automatically terminates any test in progress and exits the Test Mode, removing the output relay block, and performing its Safety System Function in response to the event.

The SIS Input Test shall provide the capability to individually test each SI input signal path through the digital input module to the control logic module. The M&EU soft controls shall prevent the testing of more than one SI input at a time. In any case, SIS actuation logic shall be blocked during this test.

The BOS Input Test shall provide the capability to individually test each UV input signal path through the UV relay and input module to the control logic module. The M&EU soft controls shall prevent the testing of more than one UV input at a time. The BOS Input Test interrupts the voltage input from the plant vital bus transducers, causing the UV relay to trip due to loss of input voltage.

The SIS and BOS Sequence Tests shall provide the capability to manually test the SIS and BOS actuation sequences.

The SIS and BOS Output Relay Tests shall provide the capability to individually test each SIS and BOS Step, OL and AL output signal path from the control logic module through the relay output driver modules and output relay(s) to its associated end device(s).

The DLS shall provide the capability to test the system as a whole under conditions as close to normal as practical – sequence and integrated safeguard surveillance tests.

4.5.4 HARDWARE

4.5.4.1 CABINETS

In the case new cabinets are supply, they shall have key locks.

Adequate ventilation and thermal study shall be performed.

EMI gaskets in doors and any other removable cabinet part shall be installed.

Metal vent screens shall be installed in any fan, if applicable.

Care shall be taken with painting to not interrupt conductive paths.

Provisions shall be made for field cables entering the cabinet from the bottom.

Cabinet shall be anchored to the ground by means of bolts.

In the case old cabinets are reuse, ventilation and thermal study and EMI/RFI shielding shall be conveniently study.

4.5.4.2 CABLES AND GROUNDING

Internal cables shall comply with IEEE-383, specifically being halogen-free and flame retardant. Adequate color coding standards shall be used. Wire shall be adequately terminated and identified in each end.

Adequate system grounding (through ground lugs, for instance) shall be provided for connection to the plant ground network.

4.5.4.3 TERMINAL BLOCKS

Terminal blocks shall be DIN rail-mounted. Cables wire sizes shall vary between 24 and 12 AWG for signals. Terminal blocks for power shall vary between 12 and 6 AWG.

4.5.4.4 FIELD PROGRAMMABLE GATE ARRAY SUB-SYSTEM AND MAINTENANCE AND ENGINEERING UNIT

Enough modules shall be provided for the following:

- Digital Input Modules for at least (96) signals, normally close or normally open configurable, 48 VDC wetting with optical isolation, signal conditioning, surge protection, and input filtering.
- Relay Output Driver Modules for at least (60) signals, providing 48 VDC to its associated relay coil(s) via an isolated relay (preferable solid state relay) within the module.
- Control Logic Module.

Modules shall be hot-swappable.

M&EU could be based on a microprocessor-based system or soft or hard IP-cores and provide adequate interfaces.

4.5.4.5 CONTROL PANEL

The Control Panel shall provide the hard-wired controls and test points that are necessary to support system operation and testing. Figure 26 provides a proposal of Control Panel, including as a minimum:

- SIS Reset Pushbutton, that provides a normally open mechanical dry contact to manually reset the SIS following sequence completion.
- BOS Reset Pushbutton, that provides a normally open mechanical dry contact to manually reset the BOS following sequence completion.
- Alarm Reset Pushbutton, that provides a normally open mechanical dry contact to reset latched alarm indications after the associated triggering event(s) have cleared.
- Test Bus Enable Keyswitch, that provides two (2) normally open mechanical dry contacts that activate the bidirectional Test Bus. A third normally open mechanical dry contact provides an input to the Control Logic Module.
- UV Calibration Enable Keyswitch, that provides a normally open mechanical dry contact to allow calibration of BOS inputs (voltage from vital bus bars).

- Timebase Monitor Point, that provides the capability to verify the accuracy of the logic timebase (clock) using an external measurement and test equipment (MTE).
- UV Test Source, that provides a test jack injection point for MTE to calibrate BOS inputs (voltage from vital bus bars).

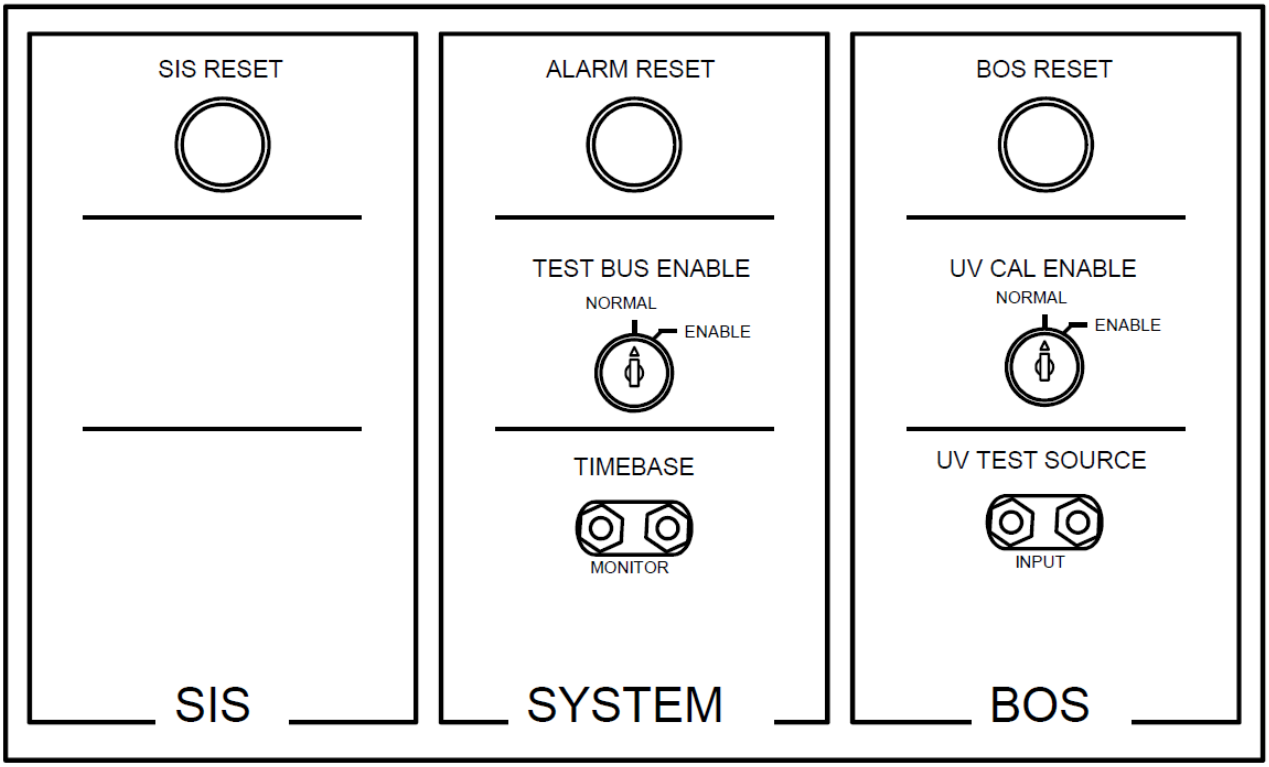

Figure 26 – Control panel layout (proposal)

4.5.4.6 TEST RELAYS

Four (4) test relays (K295 through K298) shall be provided to support testing of the SI input circuits.

Four (4) test relays (K291 through K294) shall be provided to support testing of the UV input circuits.

4.5.4.7 OUTPUT RELAYS

Output relays shall be electromechanical type industrial relays to provide normally open and normally closed output contacts for actuation of plant equipment. They shall be DIN rail-mounted and actuated by 48 VDC provided by the relay output driver modules. Contact ratings should be adequate for the loads commanded.

One (1) normally open contact on each step output relay and one (1) normally closed contact on each lockout output relay shall be used to provide feedback status signal to the system to support an output relay mismatch function.

4.5.4.8 ISOLATION RELAYS

An isolation relay (K290) shall provide the Class 1E electrical isolation barrier between the DLS and the

non-1E Plant Alarm System. The relay output circuit shall be fused to prevent fault propagation into the DLS.

The relay shall be normally energized by 48 VDC and shall use a normally open contact.

The relay shall be DIN-rail mounted and socket type.

4.5.4.9 SYSTEM POWER

Two (2) sets of 100% capacity of power supply systems in a redundant configuration (auctioneered) shall be provided inside each cabinet (per Train). Input shall be 118 VAC 50 Hz (±5%) and output shall be 48 VDC. Linear or switched type shall be evaluated, depending on the most convenient for the application.

Adequate protection devices, like OVPs, shall be provided.

Additionally, line filters and other EMC devices, as convenient, shall be provided.

Power supplies shall provide an isolated normally open mechanical dry contact that opens upon power supply failure to provide monitoring of power supply status.

DIN rail-mounted DC distribution breakers and DIN rail-mounted multi-point termination blocks shall be provided for power distribution inside de cabinet to the different system components.

4.5.5 FPGA MODULES DEVELOPMENT REQUIREMENTS

4.5.5.1 LIFECYCLE DEVELOPMENT

Development of the FPGA application shall follow a well-define and documented lifecycle, including as a minimum the following phases: requirements specification, preliminary design, design and implementation.

Required quality assurance activities for each phase, including design reviews and V&V activities, shall also be defined.

Input and outputs for each phase shall be identified and documented. A phase shall not be considered completed until all activities of that phase and the previous one have been completed and verified.

Project plan shall identified how later changes in the process will be handled and which activities should be repeated (regression tests criteria).

Project plan shall ensure that all necessary competencies and experience is represented in the design and verification teams, including competencies in electronic circuit design, the specific circuits and toolsets used and software-like development, V&V activities and configuration control and management.

Project plan shall include configuration management for, at least, design inputs, products and tools (including tools for simulation and testing, or other automated V&V tools). This shall include version control, change management for software and software-like items (developed HDL code, libraries used, pre-developed soft or hard cores, configuration setup of tools, constrains and directives files used in the synthesis and place&route, test vectors, etc.). IEEE 828 and IEC 60880 standards, although developed for software development, shall be adapted and followed.

Project management plan shall include provisions in all development phases to ensure adequate long-term support and protection against obsolescence. Aspects like FPGA and toolset suppliers long-term support policies, used of IP-cores or other pre-developed items and their portability, circuit independence of HDL code and design rules and its portability, access and completeness of the design and test documentation, etc., shall be considered. Maintenance and repair policy shall also be established, as well as later modifications policy. Minimum service life for the new system shall also be established, and spares policy and provisions for that service life.

4.5.5.2 REQUIREMENTS SPECIFICATION

Requirements shall be understandable, clear and unambiguous. For this purpose, terms definitions shall be included in the requirements specification documents. Information about the overall goals, basis for each requirement and assumptions shall be included.

Requirements shall be established in a way that is independent from a certain design approach.

Each requirement shall be achievable and verifiable.

All plant and I&C system operating conditions, both in normal or abnormal conditions, shall be considered when defining system requirements (environmental conditions, power dissipation, cooling, timing requirements, input and output loading capabilities, etc.).

Requirements for protection against CCF (like diversity) and cybersecurity shall be established.

The requirements shall specify the functions to be performed by the circuit and associated performance requirements (like time response), and shall indicate behaviours or output states to be avoided.

Primary and ancillary functions shall be perfectly identified separately, and independency requirements between all of them shall be clearly specified.

Operating modes and transitions between them shall be clearly specified, including outputs conditions and behaviour.

Interface requirements shall be included, like inputs and outputs requirements, parameters (configuration parameters, calibrations parameters, etc.), communications protocols, etc.

Circuit failure modes shall be identified and classified in acceptable or inacceptable. Target reliability values for acceptable failures shall be defined.

Requirements for failures malfunctions detection shall be specified, both by internal means (like BIST) or supporting external ones (like external watchdog). Measures taken at the system or sub-system levels shall be taken into account.

Behaviour in response to detected faults or malfunctions shall be specified, including state of outputs and times for transitioning to that state, alarming, etc. Fail-safe state definition for each error or malfunction shall be completely defined.

Interfaces for programming, testing or maintenance shall be specified (like JTAG ports). Provisions for protection against errors or problems introduced through these interfaces shall be established.

Requirements regarding longevity of support both for the circuits and the toolsets, protection against obsolescence or ease of later modifications shall be specified. This includes selection of the specific circuit to be used, definition of coding rules, selection of toolsets, use of pre-developed items, etc.

4.5.5.3 PRELIMINARY DESIGN

The architecture shall be defined to the greatest degree of simplicity achievable consistent with the required functionality.

Implement flat hardware logic whenever possible.

Synchronous design shall be used whenever possible.

Modular design shall be used.

Ancillary functions shall be kept separated and independent from primary ones.

Segregate primary functions if they do not need to interact between them.

Fault tolerance and other defensive design features shall be incorporated as a mean to meet the application requirements, but minimizing as much as possible the complexity of the system. Overall reliability requirements, anticipated failure rates (based on the electronic components data and the environmental conditions) and maintenance predicted practices, like surveillance testing, shall be taken into account.

An assessment of potential CCFs shall be performed.

Diversity provision shall be incorporated to address those possible CCFs of greater concerns. These provisions can include:

- Functional diversity – different functions to achieve the same goal.
- Signal diversity – use of different types or sets of signals or parameters.
- Electronic component diversity – use of circuits from different manufacturers, different technologies or different families.
- Specification diversity – use of different specification languages.
- Human diversity – different design teams, different and independent V&V teams.
- Logic design diversity – different algorithms to implement the same logic.
- Tool diversity – use of different tools and/or different configurations (for instance, constraints and directives files).

Adequate segregation and independence of diverse circuits or modules shall be established to avoid interdependencies.

Provisions, like reviews and analysis, shall be put in place to avoid creation of dependencies during the synthesis and place&route process, due to optimization features.

Circuit shall be design to provide appropriate observability of internal signals for V&V processes, as well as for troubleshooting and maintenance, commensurate with the application requirements.

These observability requirements can be fulfilled using light emitting diodes (LEDs), sending information to external HSIs or through programming ports, or a combination of them.

Circuit shall be design to meet long-term support and portability goals defined in the project plan.

For safety functions implementation, non-reprogrammable technology shall be used (antifuse technology). In case a different technology is chosen, the potential for faults, SEU events, etc., and their effects shall be precisely analysed and evaluated to assure circuit requirements will be met.

Adequacy of toolsets to be employed shall be assessed, including evaluation of quality and maturity, ease of use, level of expertise of team members with these tools, etc.

Failure data rates for the circuits chosen and evaluations against reliability requirements shall be accomplished.

Sufficient documentation on the selected circuits and toolsets shall be available, including operating modes and configurations, protocols, pins and registers, known programming issues, electrical o logical peculiarities, recommended programming rules, constraints and directives, as well as those to be avoided, optimizations that tools may perform, etc.

Long-term support for the specific circuits and toolsets selected shall be evaluated, including vendor's commitments regarding longevity of support, availability of replacement policy, expected shelf life of spares and support for porting to newer devices policy.

4.5.5.4 DESIGN

Language and tools used for design and simulation shall comply with a recognized standard. VHDL or Verilog shall be used.

Higher level languages, like ESL, can be used, but the RTL representation must comply with the above.

HDL coding shall make use of features that are synthesizable, avoiding circuit or tool dependent features.

Coding rules shall enforce good design practices, ensuring HDL code is well structured, uses appropriate coding constructs and promotes observability and testability of the design.

Coding rules shall minimize the potential for differences between simulated (based on HDL code) and synthetized behaviours.

Coding rules should facilitate detection of errors through static verification techniques (type checking, out-of- range conditions, completeness of instruction cases, dead states in finite state machines (FSMs), etc.).

Delays shall be avoided as much as possible. Static timing analysis shall be performed.

Design shall be synchronous.as much as possible. Signals at asynchronous interfaces shall be resynchronized. If some kind of asynchronous design is made, analysis of all paths shall be performed to demonstrate that the outputs comply with the requirements and no adverse glitches or metastability occurs.

Design shall be made so that its ability to perform its functions and meet the requirements will not depend on the internal propagation delays along wires or through gates.

Internal electrical and temporal characteristics of circuit's behavior during power-up, power-down or unintentional loss of power (partial or total) shall be documented and evaluated to meet application requirements.

Design shall include initialization input signals so that places all outputs, registers and FSMs in a predefined and documented state. These signals shall comply with specific circuit requirements (rise time, fall time, monotonicity, etc.).

Pins and registers that put the circuit in special configurations, like test, diagnostic or programming modes, shall be analyzed for potential impact during normal operation, and configured to avoid any adverse impact on application functionality.

Each of the primary functions shall be testable by internal or external means.

4.5.5.5 IMPLEMENTATION

Distribution of power to circuit elements shall make use of dedicated power paths provided by the circuit.

Files for constraints and directives used during synthesis and place&route phases shall be detailed documented and place under configuration control.

Equivalence of the post-route description and the RTL, including time information, shall be demonstrated for the fastest and slowest case, including initialization. Changes in environmental conditions, like temperature, shall be taken into account, according with operating conditions define in the requirements.

The coverage of self-testing define for the primary functions shall be assessed against the require coverage defined previously. This shall be done after place&route phase to take into account the final circuit topology implementation and the possible effects of optimization performed by tools.

It shall be verified that logic optimization performed by synthesis and place&route tools do not remove features that where introduce in the design intentionally as part of error detection and correction (EDAC) or diversity measures, such as redundancy or definition of cases that are rarely reachable but are addressed in the design.

Additionally, replication of gates performed by these tools to meet timing or other defined constraints shall be analyzed to verify that no additional states are introduced in the design, or if this happens, they are acceptable.

STA shall be performed for best and worst cases to determine timing margins. Environmental effects, like temperature, shall be taken into account.

4.5.5.6 VERIFICATION AND VALIDATION

The extent of V&V activities shall be commensurate with the level of complexity and risk significance.

Verification teams shall include individual with enough electronic design expertise.

Formal verification techniques shall be used by an independent V&V team, at least for the primary functions.

Tests shall be performed so that the circuit remains active in different conditions for very long time.

4.5.5.7 CYBERSECURITY

Industry regulations, guidelines and practices for dealing with cybersecurity shall be applied.

If reprogrammable technology is used, provisions for detecting any alteration of the circuit's programming shall be provided.

Protection shall be provided to preclude potential cybersecurity threats being introduce through maintenance or surveillance activities.

Spare parts inventory protection measures shall also be provided.

Hashing techniques for configuration files or other measures to assure file integrity for configuration items shall be established.

4.6 EQUIPMENT QUALIFICATION

The DLS system is to be located in a mild environment.

IEEE-323 and IEEE-344 shall be the basis for the replacement equipment environmental and seismic qualification, respectively, as they are endorsed by NRC Regulatory Guides 1.89 and 1.100.

EMC type test requirements shall be based on NRC Regulatory Guide 1.180, which includes type testing to various IEEE Standards, Military Standards, IEC Standards, and EPRI guidelines, as applicable.

Equipment qualification shall be accomplished using several different techniques, including type testing, analysis, or operating experience. These can be used individually or in any combination depending on the particular situation. With all qualification methods, the end result shall be traceable and auditable documentation with appropriate independent review demonstrating the adequacy of equipment to perform its safety system functions under design basis accident conditions.

Test sequence shall include, as a minimum:

- Initial inspection.
- Functional test.
- EMC tests.

- Functional Test.
- Abnormal environmental (temperature and humidity) test.
- Mechanical aging test.
- Resonant frequency search test.
- Functional Test.
- Seismic Tests.
- Post-test inspection.

Relays and switches used in Class 1E subsystems shall be mechanically aged prior to seismic testing. The number of aging cycles shall be established to simulate the expected equipment operating life.

An electromagnetic site survey where the system is going to be installed shall be performed.

4.7 CYBERSECURITY

Requirements of NRC Regulatory Guide 1.152 and NRC Regulatory Guide 5.71 shall be fulfilled.

The cybersecurity requirements could be applied to the DLS as a whole.

Assessments shall be performed during the conceptual, design, and implementation phases to ensure that the cybersecurity controls are adequate.

4.8 HUMAN FACTOR ENGINEERING

Human Factors Engineering (HFE) evaluation shall be performed according to NUREG 711 considerations.

The evaluation should focus as a minimum on the M&EU and associated displays as well as Control Panel.

4.9 DOCUMENTATION DELIVERABLES

The following documentation shall be provided along the project:

A. Functional Requirements Document (FRD).

   This document shall include, as a minimum:

   - Safety System functional requirements.

- Safety System Auxiliary functional requirements, including non-safety functional requirements.
- Performance requirements associated with the system functions (time response, accuracy, etc.).
- Setpoints and constants associated with the system functions.

B. Functional Logic Diagram (FLD).

This document shall depict the top level functional operation of the system, including all functional inputs and outputs, and shall provide the allocation of functional attributes to system components.

C. System Requirements Document (SRD).

This document shall specify the system requirements for the system, including as a minimum:

- System design requirements.
- System internal and external interface requirements.
- HSI requirements.
- Cybersecurity requirements.
- Equipment qualification requirements.

D. System Design Specification (SDS).

This document shall describe the design of the system, including as a minimum:

- System architecture diagram.
- Description of the functionality and architecture.
- Allocation of the functionality to the subsystem level.
- Description and definition of the system external interfaces, including I/O, plant computer system datalinks and main control board failure alarm.
- Description of the system internal communications, including the TxB1 and TB.
- Specification of the system hardware configuration.
- Description of system diagnostics.
- Description of system testing capabilities.
- Safety System response time analysis, according to system response times defined in the FRD.
- Safety System spare capacity analysis, which analyzes the spare capacity, including I/O.

- System power consumption and heat dissipation analysis.
- Description of cybersecurity attributes.

E. Software Requirements Specification (SRS).

This document shall specify the software requirements for the M&EU, including:

- Software design requirements (e.g., the required use of a particular programming tool or language, or the required use of particular platform software).
- Requirements associated with the functional processes.
- Timing requirements.
- Cybersecurity requirements.
- TxB1 and TAB requirements.
- Plant computer system datalink protocol requirements.

F. Software Design Description (SDD).

This document shall provide a description of the M&EU software, including:

- Description of the hardware and software environment, including the architecture of the subsystem and identification of the platform software that is to be used to support the application.
- Decomposition of the required functions into software modules.
- Screen prints of the as-built displays.
- Description of plant computer system datalink protocol.
- Description of cybersecurity attributes.

G. Software Licenses, Application Code and Databases, when applicable.

H. Requirements Traceability Matrix (RTM).

This document shall track all requirements that must be satisfied by the system. The Independent Verification and Validation (IV&V) team shall review the adequacy and accuracy of this document.

I. Verification and Validation (V&V) Plan.

This document shall define the V&V process to be applied during system development and implementation. The process shall define how, when and by whom specific V&V activities are to be performed, including options and alternatives, as required.

J. Verification and Validation (V&V) Report.

K. Failure Modes and Effects Analysis (FMEA).

This document shall provide a qualitative assessment of system reliability by examining the failure mode of each system element and determining the effect of those failures on the capability of the system to perform its Safety System Functions and Safety System Auxiliary Functions.

L. Reliability Analysis Report.

This document shall provide a study of system reliability and availability.

M. Diversity and Defense-in-Depth Analysis.

This document shall specifically address NUREG CR-6303 criteria.

N. Cybersecurity Assessment Report.

This document shall specifically address assessment of system cybersecurity attributes to ensure compliance with Regulatory Guides 1.152 and 5.71.

O. Project-Specific Test Plan.

This document shall define the process and requirements for testing the system.

P. M&EU Test Procedure.

This procedure shall verify the functionality of the M&EU and demonstrates compliance with applicable requirements specified in the SRD and SRS.

Q. M&EU Test Report.

R. Cabinet Hardware Test Procedure.

This document shall verify satisfactory assembly of the system hardware.

S. Cabinet Hardware Test Report.

T. Qualification Report.

This document shall provide traceable evidence of qualification activities of system equipment and components. It shall also include all qualification procedures and results.

U. Factory Acceptance Test Procedure.

This procedure shall demonstrate compliance with the applicable requirements specified in the FRD, SRD, and SRS.

V. Factory Acceptance Test Report.

W. System Certificate of Conformance.

This document shall certify that the system meet its specific requirements.

X. Operation and Maintenance Manual.

This document shall provide the information necessary to support and maintain the system, including as a minimum:

- Hardware and software description.
- Principles of operation.
- System operation.
- Complete set of system drawings, including functional logic diagrams, cabinet layout, internal wiring, etc.
- Complete list of system components, including manufacturer, model and main characteristics.
- Corrective maintenance, including diagnostics, troubleshooting, and component replacement.
- Preventive maintenance, including testing and calibration.
- Receipt, storage, and handling.
- Technical information, including sub-vendor manuals.

4.10 RECOMMENDED SPARE PARTS

Startup spares for installation and commissioning shall be provided.

In addition, operational spares for at least 10 years shall also be provided.

A proposal based on reliability and aging analysis shall be provided, shall be according to Operation and Maintenance Manual instructions and recommendations, and shall be subject to customer approval.

4.11 TRAINING

A recommended training program and schedule, both for Operations and Maintenance, shall be prepared.

The program and contents will be subject to customer prior approval.

4.12 QUALITY ASSURANCE

A Quality Management System shall be put in place for the project, and shall be in accordance with 10CFR50, Appendix B. Additionally, accordance to other standards, like ISO 9000 or ASME NQA-1 is recommended.

A Specific Project Quality Plan shall be issued to document additional quality requirements that are applicable or unique to the project.

Applicable Inspection Point Plan shall be issued for the project.

Additionally, project specific audits by customer could be done.

5 CONCLUSIONS AND FUTURE RESEARCH ACTIVITIES

5.1 CONCLUSIONS

The conclusions from the present study are the following:

- FPGA technology is a modern but mature technology that is widely used in many industries, from consumer electronics through medicine, transportation, military and defense.
- FPGA technology is a new technology for the nuclear sector, although some deployments, both in safety and non-safety systems, have already taken place worldwide.
- There is currently a lack of regulation and guidance regarding the use of FPGA technology in the nuclear industry. First international standard has been issued recently from the International Electrotechnical Commission.
- FPGAs can be used in several different forms or architectures with differing complexities. The simpler the implementation, the easier the licensing process.
- For safety systems, flat hardware logic implementation with parallel processing is the most suitable and covers all possible needs for safety systems.
- Self-testing and diagnostic functions, communications interfaces, and more auxiliary functions can be implemented in an independent way, not affecting primary, critical or safety functions.
- Testability and ease of review and V&V is much easier than in microprocessor-based systems.
- However, because typical I&C applications are not trivial and have multiple inputs and outputs, verification solely through inspection and testing is not considered sufficient by regulators. Thus the application development process must be scrutinized as part of ensuring adequate reliability and safety of the design.
- Because this process typically involves development of HDL code and use of complex, commercial software tools, the process is viewed as very similar to a software development process, and thus regulatory approval once again relies upon documentation and demonstration of adequacy of this process.
- More complex functionalities, typically for auxiliary functions, or for non-safety systems, can also be implemented using FPGAs making use of IP cores. Use of pre-developed

IP cores is possible, reducing the need to build them from scratch, and providing capabilities that become more like conventional microprocessor-based systems.

- FPGA technology presents also very important advantages over microprocessor-based systems, regarding the system life cycle, thanks to its portability to another type of FPGA. Other advantages include faster processing speed and cybersecurity robustness.
- Use of FPGA-based solutions allows for more flexibility in the case of replacement or modernization projects, as they can be applied not only at the system level, but at the component and sub-component levels, with associated benefits, including costs.
- Guidance for implementation of FPGA-based systems in nuclear power plants has been provided along this study and for all phases of the project, from planning and conceptual design, through design and development, to implementation and verification.
- Finally, a practical application of the study results has allowed the development of a specification for a new FPGA-based Diesel Load Sequencer safety system.

5.2 FUTURE RESEARCH ACTIVITIES

Starting from the results of this thesis, the following research activities are proposed for future work:

- Development of the specification for a safety platform based on the use of field programmable gate arrays for implementing primary functions in nuclear power plants.
- Design, manufacture and testing of main components of the safety platform.
- Development of a specific nuclear safety system based on the safety platform.
- Study of alternatives for development a completely generic qualified platform with application-specific add-on logic modules.

  This way, only those application-specific add-on modules (FPGAs containing application specific control and protection strategies) would have to be requirement defined, designed, programmed, verified, validated, tested and licensed in a case by case basis, reducing efforts and costs for FPGA deployment along nuclear industry.
- Development of automatic testing tools for full coverage of the safety platform.

  This, along with the previous research line, would allow simplifying licensing and approval process, making it more like a traditional one for hardware-only systems, equipments and components, and like in the present time, which more similar to a software-based system.

THIS PAGE INTENTIONALLY LEFT BLANK.

ACRONYMS

ABWR	Advanced Boiling Water Reactor
AC	Alternative Current
ADC	Analog-to-Digital Converter
AECL	Atomic Energy of Canada Limited
AL	Automatic Lockout
ALS™	Advanced Logic System
APRM	Average Power Range Monitor
APWR	Advanced Pressurized Water Reactor
ASIC	Application-Specific Integrated Circuit
ASU	ALS™ Service Unit
ATE	Advanced Test Equipment
ATU	ALS™ Test Unit
ATWS	Anticipated Transient Without Scram
BGA	Ball Grid Array
BIST	Built-In Self-Test
BOP	Balance of Plant
BOS	Black-Out Sequencer
BRAM	Block Random Access Memory
BTP	Branch Technical Position
BWR	Boiling Water Reactor
CAD	Computer-Aided Design
CANDU	CANadian Deuterium Uranium
CCF	Common Cause Failure
CDR	Critical Design Review or Count Down Register
CEC	Complex Electronic Component
CFR	Code of Federal Regulations (US)
CJC	Cold Junction Compensation
CJT	Cold Junction Temperature
CLB	Configurable Logic Block / Core Logic Board

CMOS	Complementary Metal-Oxide Semiconductor
COC	Cross Output Cabinet
COTS	Commercial Off-the-Shelf
CPLD	Complex Programmable Logic Device
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
D&DiD	Diversity and Defense in Depth
DAS	Diverse Actuation System
DC	Direct Current
DCC	Digital Control Computer
DCM	Digital Clock Manager
DEC	Digital Equipment Corporation
DFT	Design for Testability
DLS	Diesel Load Sequencer
DoD	Department of Defense (US)
DoE	Department of Energy (US)
DPS	Diverse Protection System
DSP	Digital Signal Processing
ECSS	European Cooperation for Space Standardization
EDA	Electronic Design Automation
EDAC	Error Detection and Correction
EDE	Electronic Design Environment
EDF	Electricité de France
EDIF	Electronic Design Interchange Format
EEPROM	Electrically Erasable Programmable Read-Only Memory
EFT	Electrical Fast Transient
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
EPLD	Electrically Programmable Logic Device
EPR	European Pressurized Reactor

EPRI	Electric Power Research Institute
EPROM	Erasable Programmable Read-Only Memory
ESA	European Space Agency
ESBWR	Enhanced Simplified Boiling Water Reactor
ESD	Electrostatic Discharge
ESF	Engineering Safety Feature
ESFAS	Engineered Safety Features Actuation System
ESL	Electronic System Level
FAT	Factory Acceptance Test
FCO	Full Capacity Operation
FCR	Fault Containment Region
FE	Functional Element
FEC	Functional Equivalence Check / Formal Equivalent Checking
FET	Field Effect Transistor
FIFO	First In First Out
FLD	Functional Logic Diagram
FMEA	Failure Modes and Effects Analysis
FPAA	Field Programmable Analog Array
FPD	Flat Panel Display
FPGA	Field Programmable Gate Array
FPLA	Field Programmable Logic Array
FSM	Finite State Machine
FTA	Fault Tree Analysis
FRD	Functional Requirement Document
GPI	General Purpose Interconnect
HDL	Hardware Description Language
HFE	Human Factors Engineering
HMI	Human-Machine Interface
HPD	HDL Programmed Device
HSI	Human-System Interface

HVAC	Heating, Ventilation and Air Conditioning
I&C	Instrumentation and Control
I/O	Input/Output
IAEA	International Atomic Energy Agency
IC	Integrated Circuit
IDE	Integrated Design/Development Environment; see also EDE
IEC	International Electrotechnical Commission
IEEE	Institute of Electric and Electronic Engineers
INSA	Independent Nuclear Safety Assessor
IP	Intellectual Property
IPB	InPut Board
IPC	Association Connecting Electronics Industries (former Institute of Printed Circuits)
IT	Information Technology
IV&V	Independent Verification and Validation
JEDEC	Joint Electron Device Engineering Council
JTAG	Joint Test Action Group
LAR	License Amendment Request
LED	Light Emitting Diode
LOOP	Lost Of Outside Power
LPRM	Local Power Range Monitor
LSB	Least Significant Bit
LSELS	Load Shed and Emergency Load Sequencer
LUT	Look-Up Table
LVDS	Low-Voltage Differential Signaling
M&EU	Maintenance and Engineering Unit
MCR	Main Control Room
MCU	Micro-Controller Unit
MHD	Moving Head Disk
MHTL	Motorola High Threshold Logic
MOV	Metal Oxide Varistor

MSB	Most Significant Bit
MSFIS	Main Steam and Feedwater Isolation System
MTBF	Mean Time Between Failures
MTBMO	Mean Time Between Metastability Occurrences
MTE	Measuring and Test Equipment
MTS	Maintenance and Training System
NASA	National Aeronautics and Space Administration
NCC	Normalizing Converter Cabinet
NMS	Neutron Monitoring System
NPL	Non-Programmable Logic
NPP	Nuclear Power Plant
NRC	Nuclear Regulatory Commission
NRO	NRC Office of New Reactors
NRR	NRC Office of Nuclear Reactor Regulation
NVM	Non-Volatile Memory
OEM	Original Equipment Manufacturer
OL	Operator Lockout
OPB	OutPut Board
OPG	Ontario Power Generation
ORNL	Oak Ridge National Laboratory
OS	Operating System
PAC	Programmable Automation Controller
PACS	Priority Actuation and Control System
PC	Personal Computer
PCB	Printed Circuit Board
PCEC	Programmable Complex Electronic Component
PCS	Plant Computer System
PDED	Pre-Developed Electronic Design
PDR	Preliminary Design Review
PET	Positron Emission Tomography

PGA	Pin Grid Array
PID	Proportional Integral Derivative
PLA	Programmable Logic Array
PLC	Programmable Logic Controller
PLD	Programmable Logic Device
PPS	Process Protection System
PRNMS	Power Range Neutron Monitor System
PROM	Programmable Read-Only Memory
PRPS	Primary Reactor Protection System
PSL	Property Specification Language
PSU	Power Supply Unit
PWR	Pressurized Water Reactor
QA	Quality Assurance
QED	Quickware Engineering and Design
QFP	Quad Flat Pack
QMR	Quadruple Modular Redundancy
RAB	Reliable ALS^{TM} Bus
RAM	Random Access Memory
RBM	Rod Block Monitor
RCC	Remote Control Cabinet
RCO	Reduced Capacity Operation
RCS	Rod Control System
RCTM	Requirements Compliance and Traceability Matrix
RFI	Radio Frequency Interference
RIC	Reactor In-Core measurement system
ROM	Read-Only Memory
RPS	Reactor Protection System
RRCN	Rolls Royce Civil Nuclear
RTCA	Radio Technical Commission for Aeronautics
RTD	Resistance Temperature Detector

RTIS	Reactor Trip and Isolation System
RTL	Register Transfer Level
RTM	Requirements Traceability Matrix
RTS	Reactor Trip System
SAT	Site Acceptance Test
SC	Signalling Cabinet
SCCF	Software Common Cause Failure
SDD	Software Design Description
SDL	Space Dynamics Laboratory (Utah State University)
SDS	Shut-Down System / System Design Specification
SEE	Single-Event Effect
SER	Safety Evaluation Report
SEU	Single-Event Upset
SFC	Signal Forming Cabinet
SI	Safety Injection
SIS	Safety Injection Sequencer
SLU	Slave Logic Unit
SMT	Surface Mount Technology
SoC	System-on-Chip
SOE	Sequence Of Events
SPI	Serial Peripheral Interface
SPST	Single Pole Single Throw
SPV	Single Point Vulnerability
SRAM	Static Random Access Memory
SRD	System Requirement Document
SRNMS	Startup Range Neutron Monitoring System
SRS	Software Requirement Specification
SSPS	Solid State Protection System
SSR	Solid State Relay
SSTA	Statistical Static Timing Analysis

STA	Static Timing Analysis
STB	Service and Test Board
STP	South Texas Project
TAB	Test ALS™ Bus
TEPCO	Tokyo Electric Power Company
TMR	Triple Modular Redundancy
TVS	Transient Voltage Suppressor
UCF	User Constraint File
UK	United Kingdom
US(A)	United States (of America)
V&V	Verification and Validation
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
VVER	Vodo-Vodyanoi Energetichesky Reaktor

DEFINITIONS

Antifuse – a technology for storing the "programming" or configuration of the interconnects in a programmable logic device such as an FPGA or CPLD. This technology is non-rewritable and non-volatile. A contact between two wires of the interconnection grid is created by sending a high current through the wires. Rather than breaking a connection or fuse to form the current flow, the connection is created between two logic blocks by means of heated nickel-alloy links, thus the name "anti-fuse."

Application-specific integrated circuit (ASIC) – an integrated circuit customized for a specific use and configured by means of a "mask" at the factory.

Complex programmable logic device (CPLD) – a programmable logic device that contains a number of "macro cells" that are essentially the same as programmable array logic (PAL), and the means to interconnect them.

Dependability – a property of a system such that reliance can justifiably be placed on the service it delivers. Dependability incorporates concepts like availability, reliability, safety, integrity and maintainability.

Die – a small block of semiconducting material, on which a given functional circuit is fabricated. Typically, integrated circuits are produced in large batches on a single wafer of electronic-grade silicon through processes such as photolithography. The wafer is cut ("diced") into many pieces, each containing one copy of the circuit. Each of these pieces is called a die. "Shrinking the die" refers to making the die and the associated circuit features smaller, for example, going from 250 nanometers (nm) to 180nm, 130nm, 90nm, 65nm, 45nm, and so on.

Electronic design automation (EDA) – a category of integrated design and production tools used to create integrated circuits and printed circuit boards.

Erasable programmable read-only memory (EPROM) – a type of solid state storage containing non-volatile memory that can be erased and reprogrammed.

Fan-out – maximum number of digital inputs that can be fed by the output of a single logic gate.

Flash – a type of solid state storage containing non-volatile memory that can be erased and reprogrammed. Flash is similar to EPROM but its memory is erased in larger blocks, making it faster than EPROM.

Field-programmable gate array (FPGA) – an integrated circuit designed for configuration or "programming" in the field after manufacture; it includes configurable logic blocks, programmable interconnections among them and programmable blocks for inputs and/or outputs.

Finite state machine (FSM) – an abstract model of a machine that has a primitive internal memory and a behaviour composed of a finite number of states, transitions between those states, and actions (e.g., entering a state, exiting a state, input action). The behaviour of a finite

state machine can be represented in a state transition diagram. Within a given state, for each input there is only one possible transition from the present state to the new state. An application may contain multiple finite state machines interacting with each other through their inputs and outputs.

Flat logic or flat hardware logic – this refers to logic that is implemented directly in a circuit's electronic design using configurable logic blocks and interconnections, and not using any microprocessors or run-time software. This is contrasted with an implementation that uses a microprocessor core and associated software to implement the logic, in which case the logic functions are executed serially rather than in parallel as they would be in a "flat" logic implementation.

Flip-flop – a bi-stable state circuit providing a single bit of memory. A flip-flop is usually controlled by one or two control signals and/or a gate or clock signal. The output often includes the complement as well as the normal output.

Hard IP core – an intellectual property (IP) core that is provided in the form of physical circuit layout; with a hard IP core the end-designer does not need to perform the synthesis and place&route process as would be required for a soft core. These are necessarily circuit technology- specific.

IP core – a reusable unit of logic, cell design or chip layout design belonging to one party and licensed for use by another party. These are typically offered for ASIC and FPGA design components as *netlists*, but may be either *soft* or *hard* IP cores. Vendors offer libraries of IP cores to end users as a means for faster development.

JTAG port – Joint Test Action Group, the IEEE 1049.1 boundary scan test method, "Standard Test Access Port and Boundary-Scan Architecture" is widely used on FPGAs and CPLDs for programming the device, debugging the program, gaining visibility to internal signals/states, and testing for faults.

Netlist – a logical or physical description of an electronic design defining the connectivity. A netlist is typically circuit-dependent.

Programmable array logic (PAL) – a type of simple programmable logic device that consists of a programmable AND-plane followed by a fixed OR-plane.

Programmable logic array (PLA) – a type of simple programmable logic device that consists of two levels of logic, an AND-plane and an OR-plane, both of which are programmable.

Place&route – the step in integrated circuit or printed circuit board design that determines the physical locations of components, circuitry and logic elements, and the wiring paths required to connect the components.

Programmable logic device (PLD) – an electronic device that can be configured as an integrated circuit one or more times following production at a factory. This is a general-purpose

device as opposed to an application-specific integrated circuit (ASIC), which is manufactured to perform a specific application and cannot be changed after manufacture.

Register transfer level (RTL) – a description of signal flow between registers (flip-flops) and the combinatorial logic functions of the gates through which signals flow.

Single-event latch-up (SEL) – a condition that causes loss of device functionality due to a single-event-induced high current state (occasional short circuit between power and ground). A SEL may or may not cause permanent device damage, but requires power strobing of the device to resume normal device operations.

Single-event upset (SEU) – a change of state caused by ions or electromagnetic radiation striking a sensitive node in a microelectronic circuit, resulting in an error (e.g., a memory bit error or "bit-flip"). An SEU is usually a recoverable event since it is a "soft error" effecting a state change in the logic node or memory bit, but not permanently damaging the circuit.

Soft IP core – an intellectual property (IP) core that is in the form of a *netlist* or hardware description language (HDL). A soft IP core requires verification of function following implementation (synthesis and/or place&route), unlike a hard IP core.

Static random access memory (SRAM) – a form of storage that allows memory locations to be accessed for reading or writing data in any order (hence "random access"), and which does not require periodic refreshing of the memory; although it is static and does not have to be refreshed, it is volatile and thus the data are eventually lost when the memory loses power.

Synthesis – a process by which an abstract expression of a digital circuit's behaviour at the register transfer level (RTL) – for example, in a hardware description language or HDL – is translated into an equivalent description that is expressed in terms of the resources provided by the selected FPGA circuit. The circuit-dependent description is called a *netlist*.

Toolset – "packages" or "sets" of tools used in electronic design of a programmable logic device. Circuit vendors typically provide toolsets specific to their circuits, with multiple design and simulation tools that work together and are specific to the particular circuit technology. Other tools may be obtained from independent third parties as well to support design, verification and validation.

Verilog – a hardware description language (HDL) used to model electronic systems at the register transfer level (RTL). Verilog is described in IEEE Std. 1364.

VHDL – a very-high-speed integrated circuit (VHSIC) hardware description language used to model electronic systems at the register transfer level (RTL). VHDL is described in IEEE Std. 1076.

THIS PAGE INTENTIONALLY LEFT BLANK.

REFERENCES

[1] EPRI TR-110044, Design and Testing Description of an Application-Specific Integrated Circuit for Reactor Protection and Control, Electric Power Research Institute, Palo Alto, CA: 1999.

[2] Review of Westinghouse Topical Report WCAP-15413, "Westinghouse 7300a ASIC-Based Replacement Module Licensing Summary Report" (TAC No. M96513), Safety Evaluation Report, ML0103090526, U.S. Nuclear Regulatory Commission, Washington, DC: 2001.

[3] EPRI TR-109390, Design Description of a Prototype Implementation of Three Reactor Protection System Channels Using Field Programmable Gate Arrays, Electric Power Research Institute, Palo Alto, CA: 1997.

[4] NUREG/CR-7006, Review Guidelines for Field Programmable Gate Arrays in Nuclear Power Plant Safety Systems, Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, 2010.

[5] Safety Evaluation by the Office of Nuclear Reactor Regulation Related to Amendment No. 181 to Renewed Facility Operating License No. NPF-42, Wolf Creek Nuclear Operating Corporation, Wolf Creek Generating Station, Docket No. 50-482, ML090610317. Nuclear Regulatory Commission, Washington, DC: 2009.

[6] Safety Evaluation by the Office of Nuclear Reactor Regulation Related to Topical Report 6002-00301 "Advanced Logic System Topical Report", C.S. Innovations, Project No. 779, ML13291A328. Nuclear Regulatory Commission, Washington, DC: 2013.

[7] Gibbons, W, Ames, H. Use of FPGAs in Critical Space Flight Applications - A Hard Lesson, Space Dynamics Laboratory, Utah State University.

http://klabs.org/richcontent/MAPLDCon99/Papers/B1GibbonsAmesP.pdf

[8] de Grosbois, J., The CANDU® Reactor, I&C Architectures, and Applications of FPGA Technology, 2nd IAEA Workshop on Applications of FPGAs in Nuclear Power Plants, 29 September – 1 October 2009 in Kirovograd, Ukraine.

[9] She, J., Jiang, J., Application of FPGA to Shutdown System No. 1 in CANDU, 6th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies NPIC&HMIT, 5-9 April 2009, Knoxville, Tennessee.

[10] Bach, J., Tavolara, I., Use of FPGA Technology in Implementation of the Logic of the Modernized Rod Control System (RCS) of the 900 MW EDF Fleet, 7th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies NPIC&HMIT, 7-11 November 2010, Las Vegas, Nevada.

[11] Nguyen, T., EDF's Projects with FPGAs, 1st IAEA Workshop on Applications of FPGAs in Nuclear Power Plants, October 2008, Chatou, France.

[12] Zhao, J., Regulatory Perspectives on Applications of FPGA/PLD Technology to New Reactor Control Systems in US, 3rd IAEA Workshop on Applications of FPGAs in Nuclear Power Plants, September 2010, Hamilton, Ontario, Canada.

[13] Nakagawa, Y. Application of FPGAs in Japanese BWR, 1st IAEA Workshop on Applications of FPGAs in Nuclear Power Plants, October 2008, Chatou, France.

[14] Koh, J.S., Licensing Experience for FPGA/CPLD in Digital-Based Safety Systems in Korea, 1st IAEA Workshop on Applications of FPGAs in Nuclear Power Plants, October 2008, Chatou, France.

[15] Choi, J.G., Experiences of an FPGA Based Safety Critical System Development for an Application to Nuclear Power Plants in Korea, 1st IAEA Workshop on Applications of FPGAs in Nuclear Power Plants, October 2008, Chatou, France.

[16] BTP 7-19, Guidance for Evaluation of Diversity and Defense-in-Depth in Digital Computer-Based Instrumentation and Control Systems, NUREG-0800, Standard Review Plan Chapter 7, US Nuclear Regulatory Commission, Washington, DC: 2007.

[17] DI&C-ISG-02, Digital Instrumentation and Controls Task Working Group #2, Diversity and Defense-in-Depth Issues, Interim Staff Guidance, Revision 1, US Nuclear Regulatory Commission, Washington, DC: 2007.

[18] NUREG/CR-6303, Method for Performing Diversity and Defense-in-Depth Analyses of Reactor Protection Systems, U.S. Nuclear Regulatory Commission, 1994.

[19] NUREG/CR-7007, Diversity Strategies for Nuclear Power Plant Instrumentation and Control Systems, Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, 2010.

[20] EPRI 1015088, Guidance on Use of Simulation to Support Digital I&C and Control Room Modifications, Electric Power Research Institute, Palo Alto, CA: 2008.

[21] IEC 62566, Nuclear Power Plants – Instrumentation and Control Important to Safety – Development of HDL-programmed Integrated Circuits for Systems Performing Category A Functions, Ed. 1, International Electrotechnical Commission: 2012.

[22] DO-254, Design Assurance Guidance for Airborne Electronic Hardware, RTCA/DO-254 (EUROCAE ED-80), Radio Technical Commission for Aeronautics, Washington, DC: 2000.

[23] DOT/FAA/AR-95/31, Design, Test, and Certification Issues for Complex Integrated Circuits, Federal Aviation Administration, Office of Aviation Research, Washington, DC: 1996.

[24] NUREG/CR-6463, Review Guidelines on Software Languages for Use in Nuclear Power Plant Safety Systems, Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, 1996.

[25] NASA-GB-8719.13, Software Safety Guidebook, National Aeronautics and Space Administration, Washington, DC, 2004.

[26] NASA/TM-2000-210616, Software Fault Tolerance: A Tutorial, National Aeronautics and Space Administration, Langley Research Center Hampton, Virginia, 2000.

[27] NASA-GB-A320, Software Formal Inspection Guidebook, National Aeronautics and Space Administration, Washington, DC, 1993.

[28] Clarkson, G., FPGA Based Safety Related I&C Wolf Creek Generating Station, 1st IAEA Workshop on Applications of FPGAs in Nuclear Power Plants, October 2008, Chatou, France.

[29] Clarkson, G., Implementing the ALS at Wolf Creek, 6th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies NPIC&HMIT, 5-9 April 2009, Knoxville, Tennessee.

[30] 6002-00.301-NP, Advanced Logic System Topical Report, Revision 1, August 2010, Westinghouse CS Innovations, Scottsdale, AZ: 2010. ADAMS ascension number ML102570797.

[31] U.S. Nuclear Regulatory Commission Approval Letter for Westinghouse Topical Report 6002-00301, “Advanced Logic System Topical Report” (TAC NO. MF3103), ML13291A320, Nuclear Regulatory Commission, Washington, DC: 2014.

[32] Sorenson, S. ALS Platform Overview, Document 6002-00026, Revision 2, CS Innovations, Scottsdale, AZ: 2008.

[33] Review of Triconex Corporation Topical Reports 7286-545, "Qualification Summary Report" and 7286-546, "Amendment 1 to Qualification Summary Report," Revision 1 (TAC NO. MA8283), ML013470433, Nuclear Regulatory Commission, Washington, DC: 2001.

[34] Diablo Canyon Power Plant Topical Report, Process Protection System Replacement Diversity&Defense-in-Depth Assessment, Revision 0, March 2010, Pacific Gas & Electric Company, Avila Beach, CA: 2010. ADAMS ascension number ML101100647.

[35] Safety Evaluation by the Office of Nuclear Reactor Regulation Regarding Diablo Canyon Power Plant, Units 1 and 2 Topical Report ‘Process Protection System Replacement Diversity&Defense-in-Depth Assessment’ Docket Nos. 50-275 and 50-323, ML 110480845, US Nuclear Regulatory Commission, Washington, DC: 2011.

[36] Zupan, A., et al., Darlington Digital Control Computer System Replacement Approach and Experience, IAEA Technical Meeting on Impact of Digital I&C on the Operation and Licensing of Nuclear Power Plants, November 2008, Beijing, China.

[37] Hohendorf, R., OPG Current and Potential Use of FPGAs, AECL Workshop on Field Programmable Gate Arrays in Nuclear Safety Applications, May 2009, Mississauga, Canada.

[38] Miyazaki, T., et al., Qualification of FPGA Based Safety Related PRM System, 6th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies NPIC&HMIT, 5-9 April 2009, Knoxville, Tennessee.

[39] UTLR-0020NP, Licensing Topical Report for Toshiba NRW-FPGA-based Instrumentation and Control System for Safety-Related Application, Revision 0, October 2012, Toshiba Corporation, Nuclear Energy Systems & Services Division. ADAMS ascension number ML12292A320.

[40] TOS-CR-FPG-2013-0001, Licensing Topical Report for Toshiba NRW-FPGA-based Instrumentation and Control System for Safety-Related Application, Toshiba Corporation, ML13080A206, 2013.

[41] Bakhmach, E., et al., Safety Critical FPGA-based NPP I&C Systems: Assessment, Development and Implementation, 17th Pacific Basin Nuclear Conference, 24-30 October 2010, Cancun, Mexico.

[42] Yastrabenetsky, M., et al., Safety Assessment of FPGA Based ESFAS for Kozloduy NPP, 6th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies NPIC&HMIT, 5-9 April 2009, Knoxville, Tennessee.

BIBLIOGRAPHY

1 Sklyar, V., Lesson Learned from Using FPGA-based Platform for NPP I&C Refurbishment, 6th IAEA International Workshop on Applications of FPGAs in Nuclear Power Plants, 8-11 October 2013 in Kirovograd, Ukraine, Research and Production Corporation. 2013.

2 Kharchenko, V., Diversity Assesment of NPP I&C Systems Developed Using FPGA-based Platforms, 6th IAEA International Workshop on Applications of FPGAs in Nuclear Power Plants, 8-11 October 2013 in Kirovograd, Ukraine, RPC Radiy, Centre for Safety Infrastructure-Oriented Research and Analysis, National Aerospace University KhAI, Department of Computer Systems and Networks. 2013.

3 Chaoli, L., FPGA Testing for V&V under IEC62566 Guideline, 6th IAEA International Workshop on Applications of FPGAs in Nuclear Power Plants, 8-11 October 2013 in Kirovograd, Ukraine, China Nuclear Control System Engineering Co. Ltd. 2013.

4 Russomanno, S., FPGA Applications in I&C Modernization Projects, 6th IAEA International Workshop on Applications of FPGAs in Nuclear Power Plants, 8-11 October 2013 in Kirovograd, Ukraine, SunPort SA. 2013.

5 Thuy, N., Formal Verification of HDL Designs Using Free-Licensed Tools, 6th IAEA International Workshop on Applications of FPGAs in Nuclear Power Plants, 8-11 October 2013 in Kirovograd, Ukraine, Electricité de France. Research and Development. 2013.

6 Hai, Z., FPGA-based Safety Platform Architecture, 6th IAEA International Workshop on Applications of FPGAs in Nuclear Power Plants, 8-11 October 2013 in Kirovograd, Ukraine, State Nuclear Power Automation System Engineering Company. 2013.

7 Chunlei, Z., FPGA-based DAS System Used in Yangjiang Units 5/6, 6th IAEA International Workshop on Applications of FPGAs in Nuclear Power Plants, 8-11 October 2013 in Kirovograd, Ukraine, China Techenergy Co., Ltd. 2013.

8 Harber, J., Used of FPGA-based Components and Systems in CANDU Nuclear Power Plants, 6th IAEA International Workshop on Applications of FPGAs in Nuclear Power Plants, 8-11 October 2013 in Kirovograd, Ukraine, Candu Energy. 2013.

9 Illiashenko, O., et al., Cyber Security Assessment of FPGA-based NPP I&Cs, 6th IAEA International Workshop on Applications of FPGAs in Nuclear Power Plants, 8-11 October 2013 in Kirovograd, Ukraine, National Aerospace University KhAI, Center for Safety Infrastructure-Oriented Research and Analysis. 2013.

10 O. Illiashenko, V. Kharchenko, A. Kovalenko. Cyber Security Lifecycle and Assessment Technique for FPGA-based I&C Systems, Proceedings of IEEE East-West Design&Test Symposium (EWDTS 2012), pp. 432-436, 2012.

11 Bakhmach, I., et. al., Experience of I&C Systems Modernization Using FPGA Technology, 7th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies NPIC&HMIT, 7-11 November 2010, Las Vegas, Nevada.

12 Bakhmach, I., et al., Design and Qualification of I&C Systems on the Basis of FPGA Technologies, 7th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies NPIC&HMIT, 7-11 November 2010, Las Vegas, Nevada.

13 Dittman, B.F., Licensing Field-Programmable Gate Arrays in Safety Systems, 7th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies NPIC&HMIT, 7-11 November 2010, Las Vegas, Nevada.

14 Druilhe, A., Daumas, F., Nguyen, T., Formal Verification of an FPGA Emulation of the Motorola 6800 Microprocessor, 7th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies NPIC&HMIT, 7-11 November 2010, Las Vegas, Nevada.

15 Fink, R.T., et al., Guidelines and a Primer on Application of Field-Programmable Gate Arrays in Nuclear Plant I&C Systems. 7th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies NPIC&HMIT, 7-11 November 2010, Las Vegas, Nevada.

16 Kojima, A., et al., Qualification of Toshiba's FPGA Based Safety Related Systems, 7th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies NPIC&HMIT, 7-11 November 2010, Las Vegas, Nevada.

17 Lu, J.-J., Chou, H.-P., Wong, K.-W., Conceptual Design of FPGA Based RPS for the Lungmen Nuclear Power Plant, 7th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies NPIC&HMIT, 7-11 November 2010, Las Vegas, Nevada.

18 Wang, X., Holbert, K.E., Clark, L.T., Using TMR to Mitigate SEUs for Digital Instrumentation and Control in Nuclear Power Plants, 7th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies NPIC&HMIT, 7-11 November 2010, Las Vegas, Nevada.

19 Xing, A., et al., FPGA Based Controller in CANDU Nuclear Safety Reactor Applications, 7th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies NPIC&HMIT, 7-11 November 2010, Las Vegas, Nevada.

20 Bobrek, M., et al., FPGA Design Practices for I&C in Nuclear Power Plants, 6th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies NPIC&HMIT, 5-9 April 2009, Knoxville, Tennessee.

21 Alvarado, R., Herrell, D., Approach to Designing FPGA Based Digital I&C Systems for Nuclear Applications, 6th International Topical Meeting on Nuclear Plant Instrumentation,

Control, and Human-Machine Interface Technologies NPIC&HMIT, 5-9 April 2009, Knoxville, Tennessee.

22 Bakhmach, I., et al., Implementation Principles of FPGA Based ESFAS for Kozloduy NPP, 6th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies NPIC&HMIT, 5-9 April 2009, Knoxville, Tennessee.

23 Dittman, B.F., Regulatory Experience with a FPGA Based Digital I&C Review, 2nd IAEA Workshop on Applications of FPGAs in Nuclear Power Plants, 29 September - 1 October 2009, Kirovograd, Ukraine.

24 Zhao, J.Y., Application of FPGA/PLD Based Technology to New Reactor Systems in US, 2nd IAEA Workshop on Applications of FPGAs in Nuclear Power Plants, 29 September - 1 October 2009, Kirovograd, Ukraine.

25 Thuy, N., Formal Verification for FPGA Design, Regional Workshop on the Impact of Digital I&C Technologies on the Operation and Licensing of NPPs, 4-8 May 2009, Portoroz, Slovenia.

26 Thuy, N., Why Use FPGAs in NPP Instrumentation&Control, Regional Workshop on the Impact of Digital I&C Technologies on the Operation and Licensing of NPPs, 4-8 May 2009, Portoroz, Slovenia.

27 Gassino, J., Introduction of Programmable Electronic Devices in Nuclear Safety Systems: a New Challenge in Assessment, EUROSAFE 2009: Safety Implications of an Increased Demand for Nuclear Energy, 2-3 November 2009, Brussels.

28 Rozen, Y, et al., Operating Licensing Principles of FPGA-Based NPP I&C Systems, 17th International Conference on Nuclear Engineering (ICONE 17), 2009, Brussels, Belgium.

29 Sorenson, S., CS Innovations ALS Advanced Logic System, AECL Workshop on Field Programmable Gate Arrays in Nuclear Safety Applications, May 2009, Mississauga, Canada.

30 Berg, M. Effectiveness of Internal vs. External SEU Scrubbing Mitigation Strategies in a Xilinx FPGA: Design, Test, and Analysis, paper published in the 9th European Conference Radiation and Its Effects on Components and Systems (RADECS07), IEEE RADECS07 Proceedings: 2008.

31 Lee, J.K., Design and Verification Process for Developing the FPGA Based Firmware for NPPs, 1st IAEA Workshop on Applications of FPGAs in Nuclear Power Plants, October 2008, Chatou, France.

32 Hayashi, T., Oda, N., Qualification and Application of FPGA Based Safety Related I&C Systems, 1st IAEA Workshop on Applications of FPGAs in Nuclear Power Plants, October 2008, Chatou, France.

33 Pampagnin, P., DO254 ED 80 Design Assurance Guidance for Airborne Electronic Hardware, 1st IAEA Workshop on Applications of FPGAs in Nuclear Power Plants, October 2008, Chatou, France.

34 Drimer, S. Volatile FPGA design security – a survey. Computer Laboratory, University of Cambridge. 2008. http://www.cl.cam.ac.uk/~sd410/papers/fpga_security.pdf

35 Schwank, J.R., et al. Radiation Effects in MOS Oxides. IEEE Transactions on Nuclear Science, 2008, Vol. 55, No. 4, pp. 1833–1853.

36 Berg, M., LaBel, K. Determining the Best-Fit FPGA for a Space Mission: An Analysis of Cost, SEU Sensitivity, and Reliability, paper presented at the Microelectronics Reliability & Qualification Workshop, Los Angeles, CA: 2007.

37 Jung, I., Current Digital Instrumentation and Control Licensing Activities, 21st IAEA Meeting of Technical Working Group on Nuclear Power Plant Control and Instrumentation (TWG-NPPCI), May, 2007.

38 Hammarberg, J., Nadjm-Tehrani, S. Formal Verification of Fault Tolerance in Safety Critical Reconfigurable Modules. International Journal on Software Tools for Technology Transfer (STTT), Vol. 7, No. 3, pp. 268–279, 2005.

39 Mason, M., O'Neill, K. FPGA Reliability in Space-Flight and Automotive Applications. FPGA Journal. 2005.

40 iRoC Technologies. Radiation Results of the SER Test of Actel, Xilinx and Altera FPGA instances. 2004.

41 Berg, M., Methodologies for Reliable Design Implementation, 7th Military and Aerospace Programmable Logic Devices (MAPLD) International Conference, 2004, Washington, DC.

42 Roosta, R. A Comparison of Radiation-Hard and Radiation-Tolerant FPGAs for Space Applications, JPL D-31228, Jet Propulsion Laboratory, National Aeronautical and Space Administration, 2004.

43 Sexton, F.W. Destructive Single Event Effects in Semiconductor Devices and ICs. IEEE Transactions on Nuclear Science, 2003, Vol. 50, No. 3, pp. 603–621.

44 Erickson, K., Asynchronous FPGA Risks, 4th Military and Aerospace Programmable Logic Devices (MAPLD) International Conference, 2000, Laurel, MD, USA.

45 Fernández-León, A., et al., ESA FPGA Task Force: Lessons Learned, 5th Military and Aerospace Programmable Logic Devices (MAPLD) International Conference, 2002, Laurel, MD, USA.

46 EPRI TR-1022983, Recommended approaches and design criteria for application of Field Programmable Gate Arrays in Nuclear Power Plant Instrumentation and Control Systems Electric Power Research Institute, Palo Alto, CA: 2011.

47 EPRI TR-1019182, Protecting Against Digital Common-Cause Failure, Electric Power Research Institute, Palo Alto, CA: 2010.

48 EPRI TR-1021077, Estimating Failure Rates in Highly Reliable Digital Systems, Electric Power Research Institute, Palo Alto, CA: 2010.

49 EPRI TR-1019181, Guidelines on the Use of Field Programmable Gate Arrays in Nuclear Power Plant I&C Systems, Electric Power Research Institute, Palo Alto, CA: 2009.

50 EPRI TR-1010263, Control System Retrofit Guidelines Update, Electric Power Research Institute, Palo Alto, CA: 2005.

51 EPRI TR-106392, Generic Testability and Test Methods Guidelines for ASIC Devices, Electric Power Research Institute, Palo Alto, CA: 1995.

52 T. Huffmire, C. Irvine, T.D. Nguyen, T. Levin, R. Kastner, T. Sherwood. Handbook of FPGA Design Security. Springer Dordrecht Heidelberg London. New York, 2010. pp. 177.

53 Bernstein, J.B., et al. Electronic Circuit Reliability Modelling. Microelectronics and Reliability, Vol. 46, No. 12, 2006, pp. 1957-1979.

54 MathWorks. HDL Code Generation and Verification.

http://www.mathworks.se/hdl-code-generation-verification/index.html

55 Understanding Metastability in FPGAs. Altera 2009.

http://www.altera.com/literature/wp/wp-01082-quartus-ii-metastability.pdf

56 Carmichael, C. Triple module redundancy design techniques for Virtex FPGAs. 2006.

http://www.xilinx.com/support/documentation/application_notes/xapp197.pdf

57 European Space Agency. VHDL Modelling Guidelines. 1994.

http://www.eda.org/rassp/vhdl/guidelines/ModelGuide.pdf

58 10 CFR 50, Domestic Licensing of Production and Utilization Facilities, US Nuclear Regulatory Commission, Washington, DC.

59 IEEE Std. 603-1991, IEEE Standard Criteria for Safety Systems for Nuclear Power Generating Stations, Institute of Electrical and Electronics Engineers, Piscataway, NJ: 1991.

60 IEEE Std. 7-4.3.2-2003, IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations, Institute of Electrical and Electronics Engineers, Piscataway, NJ: 2003.

61 Standard Review Plan Chapter 7, Instrumentation and Controls, NUREG-0800 Chapter 7, US Nuclear Regulatory Commission, Washington, DC: 2007.

62 BTP 7-14, Guidance on Software Reviews for Digital Computer-Based Instrumentation and Control Systems, NUREG-0800, Standard Review Plan Chapter 7, US Nuclear Regulatory Commission, Washington, DC: 2007.

63 DI&C-ISG-01, Digital Instrumentation and Controls Task Working Group #1, Cyber Security, Interim Staff Guidance, Revision 0, US Nuclear Regulatory Commission, Washington, DC: 2007.

64 DI&C-ISG-04, Digital Instrumentation and Controls Task Working Group #2, Highly Integrated Control Rooms – Communications Issues, Interim Staff Guidance, Revision 1, US Nuclear Regulatory Commission, Washington, DC: 2009.

65 Regulatory Guide 1.168, Verification, Validation, Reviews, and Audits for Digital Computer Software Used In Safety Systems of Nuclear Power Plants, US Nuclear Regulatory Commission, Washington, DC: 2004.

66 IEEE Std. 1012-1998, IEEE Standard for Software Verification and Validation, Institute of Electrical and Electronics Engineers, Piscataway, NJ: 1998.

67 CSA Z299.2-85, Quality Assurance Program – Category 2, Canadian Standards Association, Montreal, Canada: 2000.

68 CSA Q396.1.1-89, Quality Assurance Program for the Development of Software Used in Critical Applications, Canadian Standards Association, Montreal, Canada: 2000.

69 AEC-Q100 Rev-G, Stress Qualification For Integrated Circuits, Automotive Electronics Council: 2009.

70 AEC-Q100-004 Rev-C, IC Latch-Up Test, Automotive Electronics Council: 1998.

71 AEC-Q100-005 Rev-C, Non-Volatile Memory Program/Erase Endurance, Data Retention And Operating Life Test, Automotive Electronics Council: 2004.

72 AEC-Q100-007 Rev-B, Fault Simulation and Fault Grading, Automotive Electronics Council: 2004.

73 AIAG, Production Part Approval Process, Automotive Industry Action Group: 2006.

74 DO-178B, Software Considerations in Airborne Systems and Equipment Certification, RTCA/DO-178B (EUROCAE ED-12B), Radio Technical Commission for Aeronautics, Washington, DC: 1992 (Errata issued: March 26, 1999).

75 ECSS-S-00A, ECSS System: Description and Implementation, European Cooperation for Space Standardization: 2005.

76 ECSS-Q-60-02C, Space Product Assurance: ASIC and FPGA Development, European Cooperation for Space Standardization: 2008.

APPENDICES

THIS PAGE INTENTIONALLY LEFT BLANK.

Appendix 1: **Detailed description of Wolf Creek field programmable gate array application for Main Steam and Feedwater Isolation System.**

Appendix 2: **Detailed description of ALS™ platform.**

Appendix 3: **Diablo Canyon Reactor Protection System upgrade project.**

Appendix 4: **Digital Control Computer upgrade in Darlington (Canada).**

Appendix 5: **Power Range Neutron Monitoring System in Advanced Boiling Water Reactor in Japan.**

Appendix 6: **Modernization of Engineered Safety Features Actuation System in Kozloduy Units 5&6 nuclear power plant in Bulgaria.**

Appendix 7: **Rolls Royce Civil Nuclear experience on the development of field programmable gate array solutions.**

Appendix 8: **Modernization of slave logic units of Rod Control System in Electricité de France 900 MW series reactors in France.**

THIS PAGE INTENTIONALLY LEFT BLANK.

Appendix 1

Detailed description of Wolf Creek field programmable gate array application for Main Steam and Feedwater Isolation System[1]

Wolf Creek is a US PWR from Westinghouse that has been granted with a license renewal and life extension to 60 years. The plant began operation in 1985.

Wolf Creek tried to replace obsolete safety systems using microprocessor based platform, but the project experienced difficulties in licensing during 2003-2004 that prevented regulatory approval, mainly due to non-acceptance by NRC of D&DiD analysis made by vendor.

This introduced uncertainty in both schedule and cost of the project. Additionally, Wolf Creek needed a short-term solution for Main Steam and Feedwater Isolation System (MSFIS). This led to the decision of developing a hardware-based solution making use of FPGAs. The objective was to license a hardware platform for use in all projected upgrades for the long-term plan.

Figure 27 shows Wolf Creek safety systems architecture, where MSFIS can be seen.

[1] ALS is a trademark of Control Systems Innovations (a Westinghouse Company).

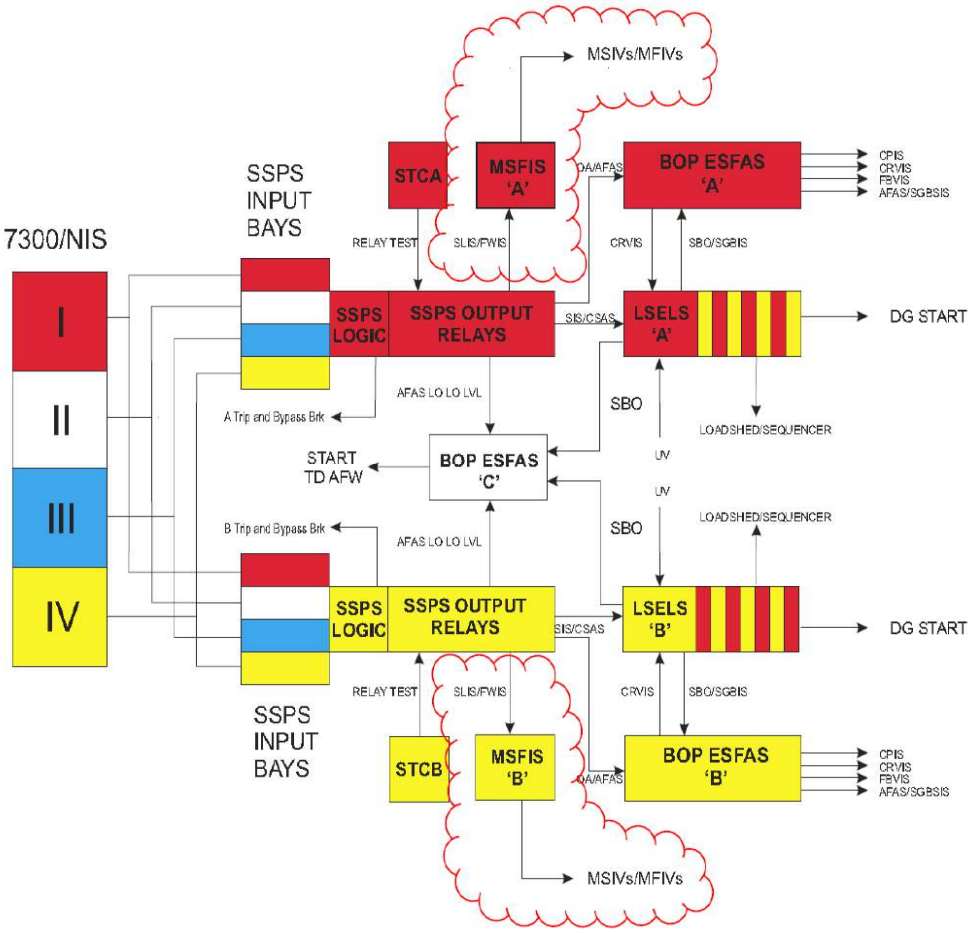

Figure 27 – Wolf Creek safety systems architecture

Source: Reference [28]

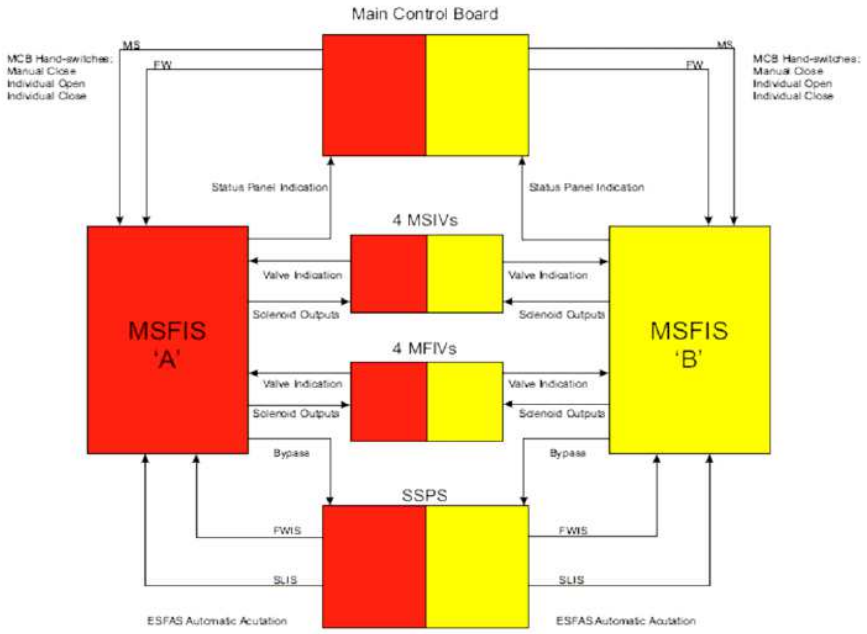

Figure 28 – Main Steam and Feedwater Isolation System detail

Source: Reference [28]

Figure 29 – ALS-based MSFIS at Wolf Creek (ALS rack)

Source: Reference [29]

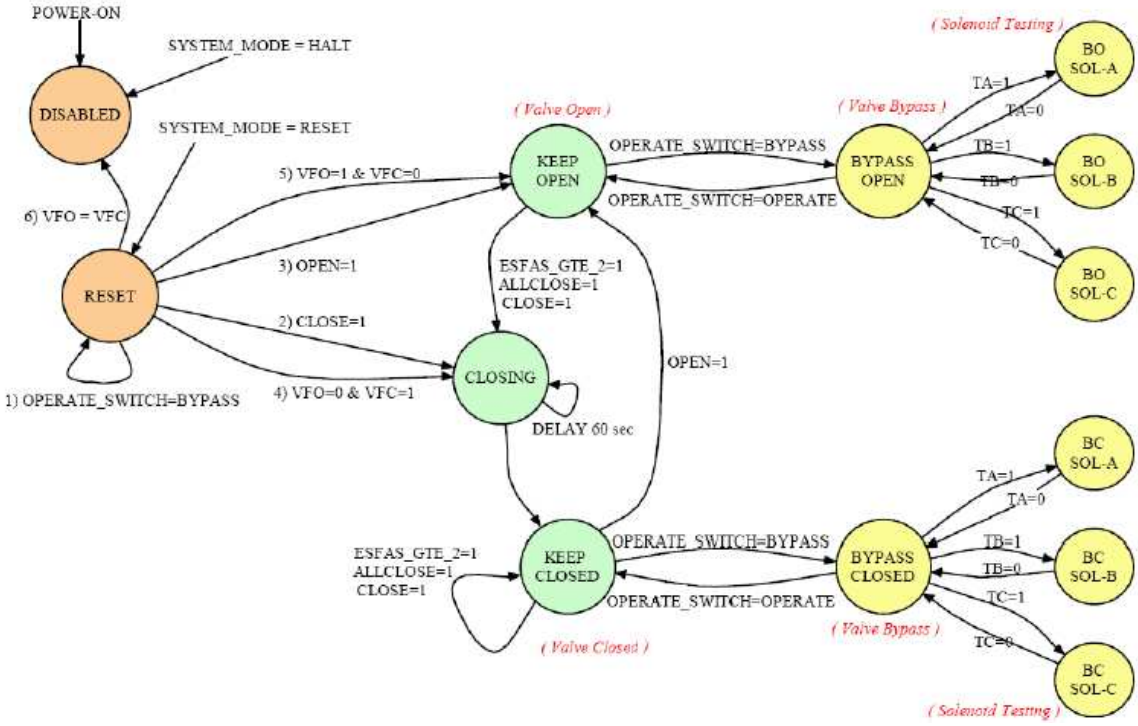

Figure 30 – Finite state machine model for MSFIS

Source: Reference [29]

The ALS platform was designed, developed and implemented under Control Systems Innovations (CSI) 10 CFR 50 Appendix B QA program. This program was audited by Wolf

Creek and Westinghouse in 2007, with acceptable results. Additionally, SER from NRC included evaluation of CSI QA program, also with acceptable results.

The platform, including the FPGA portion, was designed to meet the requirements of IEEE 603-1991. The FPGA programming was done using methods that meet requirements of IEEE 7-4.3.2-2003, adapted for an FPGA application. Additional guidance, like NRC I&C-ISG (Interim Staff Guidance), NUREG 800 (Standard Review Plan, Chapter 7 I&C and related Branch Technical Positions), cybersecurity guidance, etc. were followed.

FPGAs used are Actel ProASIC family (flash technology). Language used is VHDL.

No IP-cores and no chipsets are employed.

No formal methods were used in the V&V process, but requirements traceability matrix was put in place.

The tools used to develop and verify the FPGA implementation were not qualified as safety-related software. The supplier performed assessment and qualification of the tools to ensure that they were capable of performing the particular design or verification activity to an acceptable level of confidence.

A Licensed Amendment Request was submitted by Wolf Creek to NRC. Licensing process took 2 years since LAR submittal. NRC approval was obtained in March 2009, and the MSFIS replacement was installed in the fall of 2009.

For safety signal paths, two different logic cores are synthesized using identical HDL but different synthesizer directives, resulting in different (diverse) implementations of the same logic.

Development of test vectors and expected results was performed by personnel who were not otherwise associated with the design development.

An independent contractor was used for the independent V&V activities. Independent V&V activities used two test benches that are diverse and independent of the design engineer's test bench.

The board-level V&V was performed using ICT technologies (In-Circuit Testing). A suite of test fixture ALS boards developed for testing each board's compliance with its requirements and specifications was used.

NRC concluded that the review of the ALS platform and MSFIS application should be similar to the review of a traditionally programmed microprocessor-based application.

The staff used the same regulatory guidance and standards used for microprocessor-based systems as the basis for the review and approval of this application. These include regulatory guides that endorse IEEE standards such as IEEE 603, IEEE 7-4.3.2, and IEEE 1012. Chapter 7 of Standard Review Plan (NUREG-0800) and its various Branch Technical Positions were also used. In addition, recently issued Interim Staff Guidance documents were used to address digital system issues such as D&DiD, digital data communications, and cybersecurity. The

provisions in the various standards and regulatory guides had to be adapted in some cases in order to apply them to FPGA applications.

A goal in designing the ALS was to ensure sufficient quality, system integrity, and built-in protection against common cause failures such that they would be fully addressed without the need for any additional diversity. The intent was that no diverse backup to the ALS would be required in order to meet NRC expectations for D&DiD. This was evaluated by the NRC in their safety evaluation and for the MSFIS system the staff agreed that CCFs were adequately addressed and no additional Diverse (automatic) Actuation System (DAS) or manual operator action would be required for that system.

Safety function logic within the ALS includes a level of internal or built-in diversity. This is accomplished by incorporating two diverse implementations of the logic within the same FPGA. The two logic cores are made diverse by using different logic implementation strategies during the synthesis and place&route process. After the HDL is developed by expanding the specification into formal language, the synthesis of the HDL to produce netlists for the two cores was performed using two different hierarchical structures, finite state machine (FSM) encoding and state decoding for the two logic cores. The diversity of the two logic cores was verified by the IV&V team by comparing the netlist schematics of the two logic implementations to ensure they were diverse. Also, the two cores were compared in terms of the numbers and types of gates used for each implementation, providing further evidence of diversity.

The diverse core designs resulting from the different synthesis strategies were tested on two diverse test benches to verify that each core will adequately perform the required safety function. The two cores then underwent the place&route process and were tested again to verify proper operation of the safety application.

Equipment qualification was performed in the same manner as is done for other Class 1E equipment, applying the appropriate standards for each qualification activity (RG 1.89 and IEEE 323, RG 1.100 and IEEE 344, RG 1.180, etc.). Qualification testing included environmental qualification, electromagnetic compatibility testing, seismic testing, and response time testing.

The NRC approved use of the ALS for the MSFIS application at Wolf Creek. In addition, generic approval was given to a number of aspects of the system design, and the design and V&V processes used, which can be used as a basis for licensing other applications (follow-on applications can reference the SER for MSFIS for these generic items that were approved). However, there were aspects that gained approval only for the MSFIS application and could require further review for other applications. For example, MSFIS did not use a communications board, so that part of the ALS system was not approved. Also, simplicity of the MSFIS application was cited by the staff as part of the basis for approving certain aspects of the design, and the staff pointed out that more complex applications would need further review. The SER identifies which aspects of the ALS received generic approval, and which items would require additional review for other applications.

Westinghouse/CS Innovations submitted a supplemental topical report to the NRC in August 2010 to address the items that did not receive generic approval, so as to further reduce licensing effort for follow-on applications [30].

Enhancements were made to ALS platform, including incorporation of new hardware components (i.e., analog input board, analog output board and communication board), incorporation of new features (redundant Reliable ALS Bus, new ALS Service Unit interface, online setpoint adjustment and external power supply), enhancements to the design process in the area of independent verification and validation (IV&V) and the use of independent design teams for the development of diverse FPGA cores.

At the beginning of this year, CSI and Westinghouse have received approval for their topical report for the generic licensing of the ALS platform [6] [31].

Appendix 2

Detailed description of ALS™ platform[1]

An application implemented using the ALS platform typically consists of one or more ALS racks and an assembly panel which incorporates terminal blocks, fuse holders and other field interface hardware.

ALS rack is an industry standard 19" sub-rack. ALS boards are designed according to a proprietary standard for size and shape, assuring only ALS boards will fit into the ALS rack, and therefore ensuring the integrity of the safety system.

ALS racks can be chained together through an expansion bus if needed. The internal bus system architecture allows for up to 62 boards to be connected in up to six different racks. The racks must be located within 50 feet of one another.

Figure 31 shows an ALS rack configuration with the following parts:

- Sub-rack mountable in a 19" rack (1).
- Core Logic Board (2).
- Service and Test Board (3).
- A number of IO-boards (4-9) [up to 12].
- power supply boards (10).
- Customized back panel.
- Back plate w/ interconnects for cabling (12).

ALS rack may be powered directly from the Class 1E power source to a redundant pair of current-sharing internal power supply boards (DC-DC converters). Internal supply is 5V.

Rack cooling relies on natural convection in the cabinet and with no internal fans.

There is a set of ALS boards that can be retargeted either directly or after a simple configuration, and a set of dedicated ALS boards with FPGA logic that has been configured for a specific application. Reconfiguration requires the use of and off-line equipment (ALS Test Unit or ATU). Table 6 provides o list of some ALS types.

[1] Based on information from references [30] and [32].

ALS is a trademark of Control Systems Innovations (a Westinghouse Company).

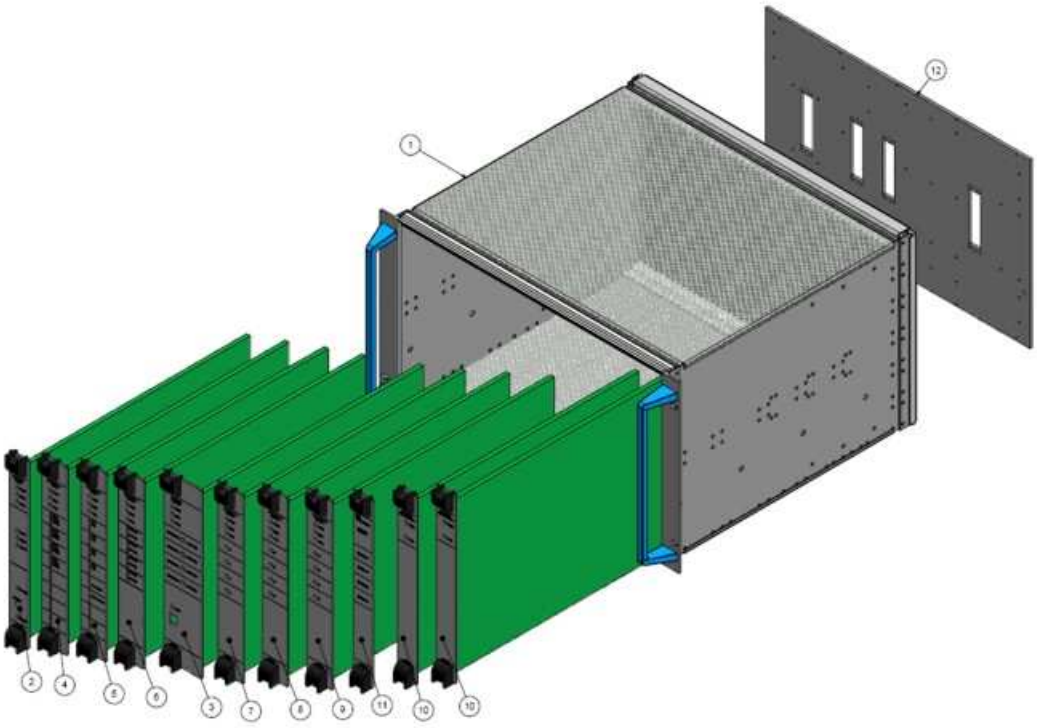

Figure 31 – Advanced Logic System rack

Source: Reference [32]

Table 6 – Advanced Logic System board types

Board Type	Configuration	Description
Core Logic Board (CLB)	Dedicated Board	Responsible for control related activities and primary communication in the system.
Input Board (IPB)	Generic Configurable	Responsible for conditioning, sensing and filtering of field input signals.
Output Board (OPB)	Generic Configurable	Responsible for controlling and conditioning of field output signals.
Service&Test Board (STB)	Generic Configurable	Provides diagnostics and monitoring capability to the ALS platform.
Communication Board (COM)	Generic Configurable	Provides secure communication links to external systems.
Power Supply Unit (PSU)	Generic	Generates supply voltages from external power source.

NOTES:

- Dedicated board – FPGA is configured with application specific logic.
- Generic configurable – Generic board, with configuration capability.
- Generic – Generic board without configuration capability.

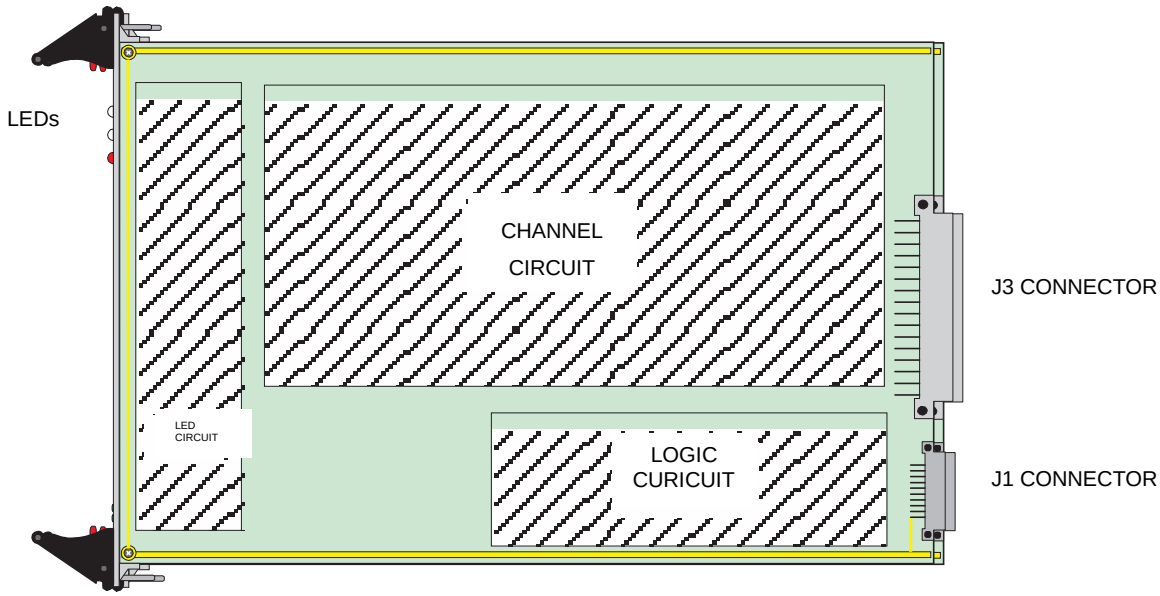

Figure 32 – Generic Advanced Logic System board

Source: Reference [32]

Channel Circuit area will vary with board type, due to different channel conditioning or output drivers. Logic Circuit area has the same components for all board types, and includes the voltage conditioning, FPGA and bus communication circuits.

FPGAs used in ALS are manufactured by Actel. Actel has focused on the military and aerospace product segment where high reliability, robust QA processes and procedures, and long product cycles are important.

Input cards make use of optoisolators devices. The input channels are protected against ESD and surge voltages using transient voltage suppressors (TVS). For output cards isolated solid-state devices are also used. No electromechanical relays are used due to reliability, long life and ability to work with inductive loads needed. Self-test capability which continuously verifies all components within the channel are operational and available both in input and output cards. Additionally, output cards can be provided with redundancy if needed. Protections against ESD and surge voltages are also present, in this case making use of metal oxide varistors (MOVs).

Board communication is supported using two different types of independent serial RS-485 point-to-point data buses: RAB (Reliable ALS Bus) and TAB (Test ALS Bus). RAB is used for all data transfers between ALS boards during normal system operation, while TAB is used for monitoring and diagnostics. CLB board is master on the RAB and STB board is master on the TAB. The buses are simple differential RS-485, point-to-point, master-slave communication protocol with proprietary communication protocol and standard cyclic redundancy checks protection to ensure the integrity of the communicated information between boards. Payloads for packages that do not meet CRC are discarded. All communication is acknowledged, except broadcasting. Communication is half-duplex. RAB bus is redundant.

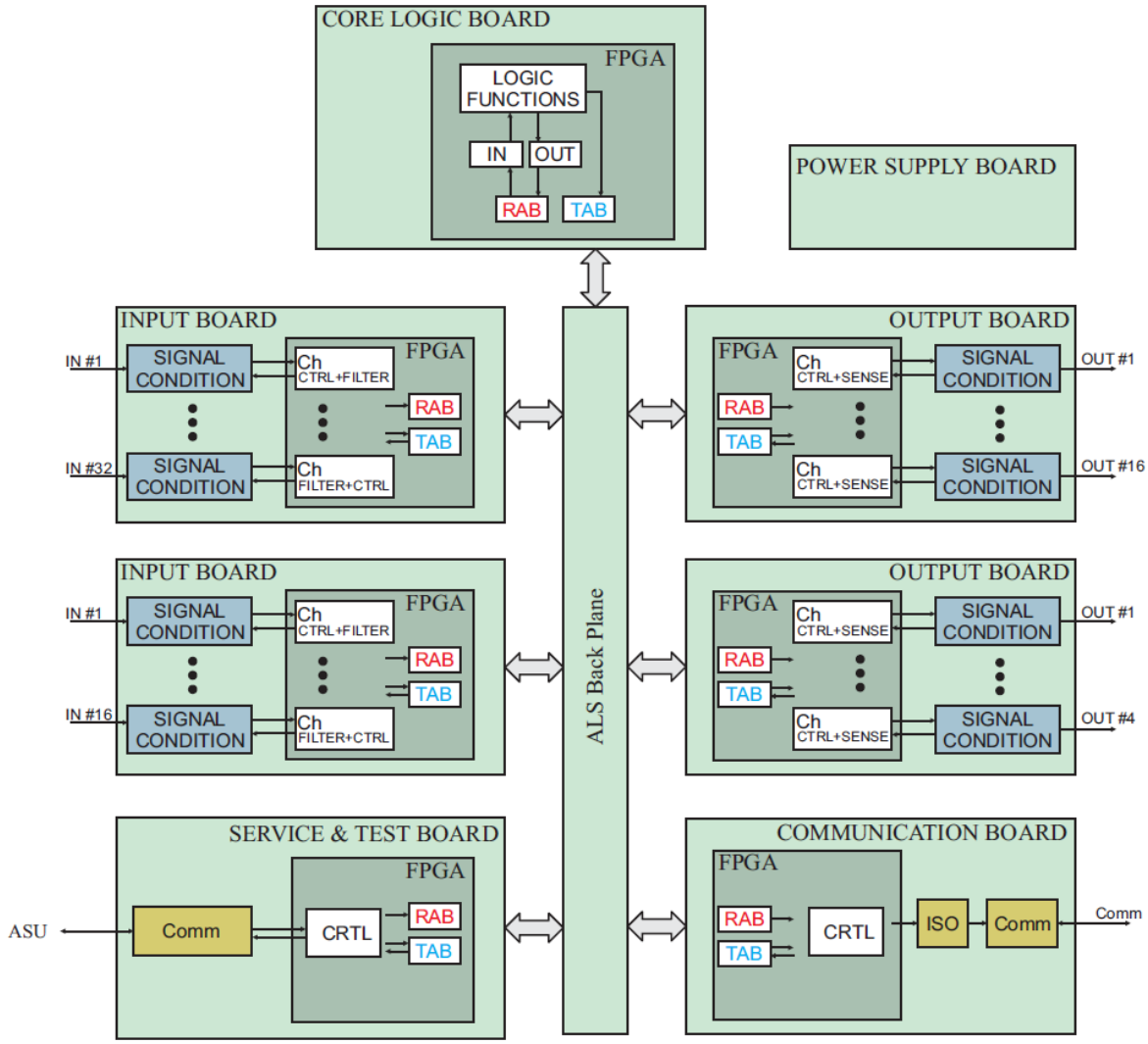

Figure 33 – Advanced Logic System architecture

Source: Reference [32]

ALS boards have dedicated flash devices to store application specific configuration and setpoints. These are stored in an external non-volatile memory and local copies are maintained/utilized within the FPGA. The configuration memory is particularly important for I/O boards, where the configuration memory allows for board reuse and common spare parts. Precautions have been made in the ALS platform to ensure that an incorrectly configured I/O board cannot cause unintended plant event if inserted into a rack. NVM contents are protected with a CRC checksum, which will detect and alert if any changes to the information have occurred.

ALS boards are designed to read out the NVM contents only at power-up. After power-up the ALS boards hold an internal copy of settings within the FPGA. While the ALS is running, the NVM contents will be tested at regular intervals to ensure there are no data integrity issues. If a data integrity issue is found, the ALS will actuate the plant alarm system to indicate an ALS

board requires maintenance. In this case, the ALS remains fully functional until the next power cycle.

BOARDS DESCRIPTION

GENERAL

ALS boards connect to the back plane using DIN41612 style connectors. Depending on the board type, there can be one or two connectors. All components, switches and LEDs are mounted directly on the printed circuit board.

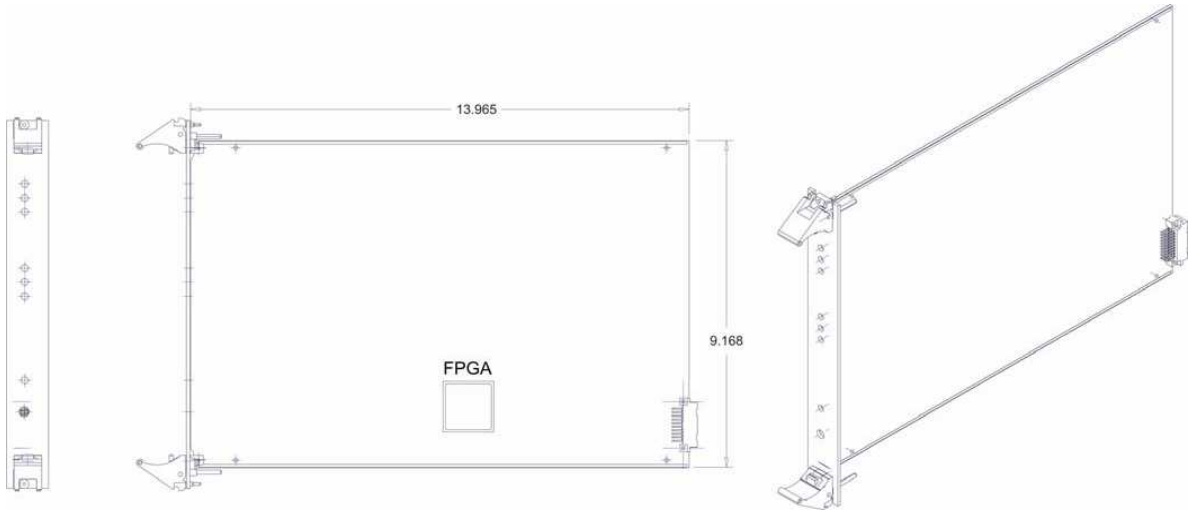

Figure 34 – Advanced Logic System board dimensions

Source: Reference [30]

ALS boards are configured with IEC 60297-3-102 style injector/ejector handles and IEC 60297-3-103 style card guides. This ensures ESD good practices when inserting or extracting the cards from the rack.

ALS slots are not designed to be interchangeable. The combination of backplane connectors, their locations and optional keying ensures that only the correct type of ALS board can be inserted into a given slot. In addition to this the ALS logic will also verify that the inserted board contains proper configuration before allowing it to become an active part of the system.

All ALS boards include 3 common system LEDs (see Table 7).

Boards are fastened and secured to the rack using board latches. Latches are IEC 60297-3-102 compliance and include a microswitch which allows the ALS board to enter a safe state before the board is ejected (see Table 8).

Table 7 – Advanced Logic System board LEDs

SYSTEM LED	INDICATION TYPE	COLOR	STYLE	DESCRIPTION
PWR	Board power indicator	Green	Solid	The board is powered – Both latches are locked
		Yellow	Solid	The board is powered – One latch is locked
		Off	Solid	None of the latches have been locked
RUN	Board running indicator	Green	Blink 1Hz	The board is running in FCO mode
		Yellow	Blink 1Hz	The board is running in RCO mode
		Off	Solid	The board has stopped execution and is in HALT mode
FAIL	Board fail indicator	Red	Solid	The board has experienced and detected a failure
		Off	Solid	The board has not detected any failures

Table 8 – Board states according to board latches states

LATCH STATE	BOARD STATE	PWR LED
Both OPEN (not inserted in rack)	Board without power	Off
Both OPEN (inserted in rack)	Board will be powered, but in HALT mode	Off
One OPEN, One CLOSED	Board will be operational	Yellow
Both CLOSED	Board will be operational	Green

CORE LOGIC BOARD

Core logic board (CLB) contains all the application specific core logic circuits, which define and control the operation of the system. The CLB controls all sequencing within the ALS system: it requests to input boards to provide field input information, makes decisions based on received inputs and commands the output boards to drive a specific output state. CLB is the primary decision making board in the ALS system and it has no direct field input or output capability except for the directly coupled system alarm output.

An ALS system typically contains one CLB, but may have more if redundancy is needed (dual redundancy, triple modular redundancy or quadruple modular redundancy).

Core logic is customizable based on the requirements of a given application and can contain any type of digital building blocks which can be generated from a NAND2 device, such as AND/OR/XOR-gates, Flip Flops (D, JK, SR), etc. These building blocks can then be combined to form more complex logic such as counters, timers, multiplexers, comparators or FSMs.

The core logic is implemented in the FPGA and the typical size of the core logic module is less than 5K gates (NAND2).

The CLB has a dedicated external flash device to store application specific setpoints and configuration. Local copies are maintained/utilized within the FPGA. Only parameters which are requested to be adjustable (like delays, trigger values, etc.) are included in the NVM configuration, otherwise they will be hardcoded.

The CLB is the RAB bus master. Communication is deterministic, with a system frame time of typically 10 ms. RAB can be provided with the desired redundancy (2 or 3 RAB busses typically).

Means for reliable communication are present, like CRC checksum. If a failure occurs, CLB will retry the transmission one time. An unsuccessful retry will result in the failing board being removed from the list of active boards and the system alarm being actuated. When a board has been removed from the list of active boards no further accesses will be attempted without acknowledgement. Data from the failing communication will be isolated and not used. Provisions are made in the logic to actuate accordingly.

The CLB is a slave on the TAB bus and will respond with the requested diagnostic and integrity information. The information is collected in a non-intrusive manner and does not affect the on-going operation of the system. Monitoring and diagnostic information (like inputs and outputs to the core logic module, states in a state machine, etc.) can be provided, but should be define early from the requirement specification.

The CLB maintains the System Mode which is used to control the run capability of the ALS system, and to provide easy to interpret diagnostics information.

CLB has a dedicated and independent alarm circuit to control and generate an isolated alarm output. Two types of failures can cause the alarm to be actuated:

- Application related failures: customizable alarms if inputs to the ALS are invalid (e.g., redundant inputs not being consistent).
- System related failures: caused by failures within the ALS circuitry such as failures on the RAB bus, supply voltages out of specified range, etc.

A redundant alarm circuit similar to the CLB alarm circuit is located on the STB.

The CLB also has the capability to accommodate 6 contact input channels, 4 solid-state output channels and 2 transmit-only EIA-422 communication channels. All input, output and communication channels are isolated from the ALS logic and can withstand 1,500 Vrms.

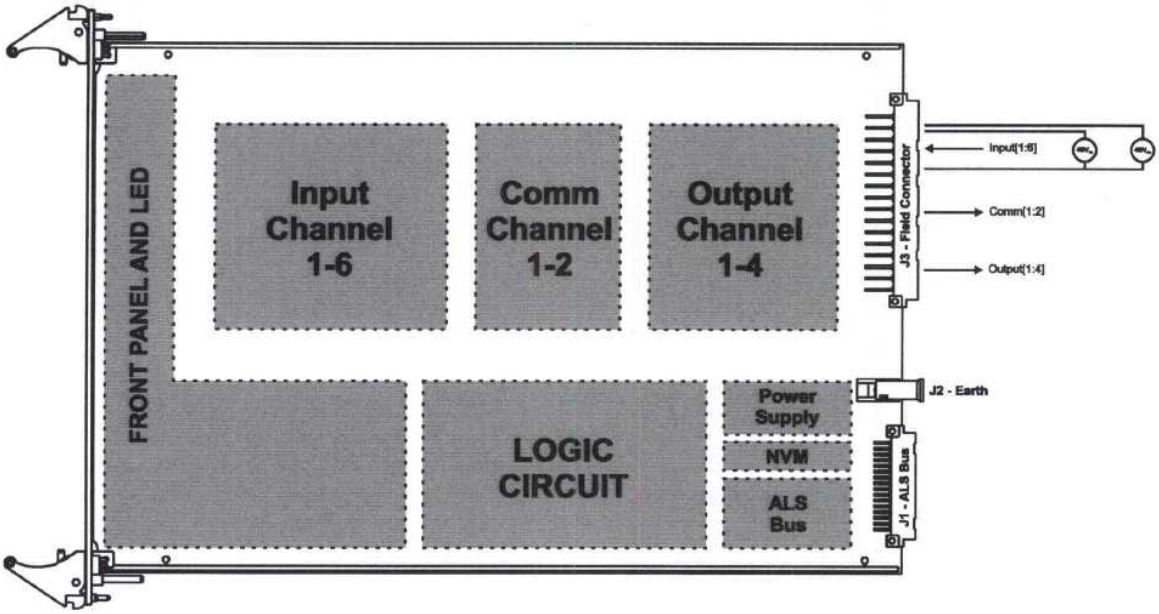

Figure 35 – Advanced Logic System Core Logic Board

Source: Reference [30]

The contact input channels are intended to be used for system related inputs. Examples of such inputs are:

1) Toggle switch for acknowledging and/or clearing alarms.

2) Detecting of state of maintenance keyswitch, i.e., COMM keyswitch.

3) Door Alarm.

4) Power supply diagnostic.

5) Function bypass toggle switches.

Communication channels can be used to send internal state and diagnostics data from the ALS chassis to local or remote safety equipment. For example, in a typical safety system application, the CLB will continuously transmit data to the ALS Service Unit (ASU) via one of the communication channels bus. The data stream includes information such as contact input/output states, analog input/output states, internal state (including counter values, analog computed values, etc.), board and system integrity, and application specific operational data. The second channel may, in the same application, continuously transmit data to a Qualified Display System (QDS) mounted on a main control board, or to a non-safety system(s) (such as the plant computer), as required by the specific application. These channels are independent, dedicated, serial, uni-directional (no handshake) EIA-422 communication channels with simple Universal Asynchronous Receiver Transmitter (UART) based (proprietary) protocol with standard baud rates.

SERVICE AND TEST BOARD

Service&Test Board (STB) provides on-line and off-line maintenance features such as monitoring and diagnostics though LiveView and BlackBox.

LiveView is a runtime diagnostics feature which provides a live-view of all important signals within the ALS. Signals include information transferred between input/output boards and the CLB as well as important internal CLB signals and health information for all the boards. LiveView is running both with the system on-line and off-line and provides real-time information with 100 μs resolution.

BlackBox is a runtime logging and post-event off-line diagnostics feature which provides the staff with sequence of events (SOE) information. The BlackBox circuit continuously monitors information transmitted on the RAB. When a change in the system occurs the information is stored into an NVM, which allows for high resolution analysis. Typical capacity is 18 months of operation.

Information provided by STB is retrieved using the ALS Service Unit (ASU) through a port in STB. ASU is not attached during normal operation. It is only attached and use for maintenance and troubleshooting activities. The alarm circuit is actuated immediately when an ASU is plugged in.

The diagnostics features are implemented in a passive and non-intrusive manner and do not affect ALS system performance. STB only uses TAB bus (in which it is the master) and does not write to or attempt to control other boards.

STB includes an alarm output, as explained before.

STB has a dedicated external flash device (NVM) to store application specific configuration. Local copies are maintained/utilized in the FPGA. Transfers take place during card energization.

ALS can work perfectly with STB removed. STB is not a necessary card in an ALS application.

INPUT BOARD

Input boards (IPBs) are responsible for conditioning, sensing, filtering, and sampling field inputs. IPBs are typically dedicated to a specific input type (digital 24 VDC or 48 VDC digital inputs, dry-contact inputs, 4-20 mA analog inputs, 0-10 VDC analog inputs, resistance temperature detector (RTD) or thermocouple inputs, etc.).

IPB provides LEDs at the front for channel status indication. IPB channels range from 4 to 32 channels, depending on the type.

Input channels consist of two circuits:

- An analog circuit for signal conditioning and ADC.
- A digital circuit located in the FPGA that performs all channel control, integrity checks, self-testing, digital filtering functions, bus communications, etc. All functions are implemented with redundant logic within the FPGA. The redundancy and self-test circuits ensures the detection of any single failure on the board.

Generally all input channels are galvanic isolated from the ALS logic and can withstand 1,500 Vrms. The input channels may also be individually isolated or isolated into groups with common reference.

As in previous cards, the IPB has a dedicated external flash device (NVM) to store specific configurations. Local copies are maintained/utilized in the FPGA. Transfers take place during card energization. Configuration can include filtering parameters, NO/NC configurations, etc.

IPBs are slave devices both on RAB and TAB busses and will respond as required by the masters (CLB or STB, respectively). Integrity information is provided with each input channel and allows the CLB to mitigate failures on channel basis.

IPBs expect to be accessed in every system frame. Thus, if no access takes place for a certain time period it will automatically time-out and enter HALT state. Integrity information about channels can make the card enter in RCO or HALT mode, depending on the severity.

Digital input board

Digital input board has 32 channels (Figure 36).

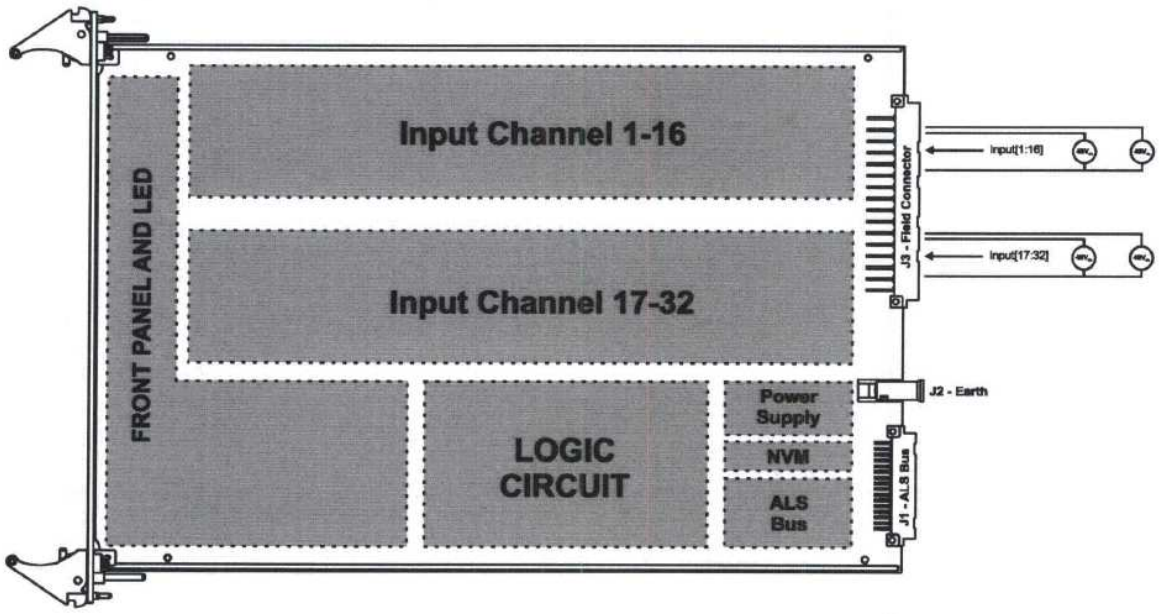

Figure 36 – Advanced Logic System digital input board

Source: Reference [30]

Analog input board (RTD/TC)

RTD/TC analog input board is a high-integrity, 8-channel temperature sensor board (Figure 37). Each temperature channel can be individually configured for 3-wire RTD, 4-wire RTD, or TC operation.

The board is presently compatible with the following temperature sensor types:

- RTDs: Pt100, Pt200, Pt300, Pt400 and Pt500.

- Thermocouples: J, K, N, E, T, R, S.

The board supports automatic cold junction compensation (CJC) of thermocouple sensors using a common cold junction temperature (CJT). The CJT is written to the board by the CLB and can originate from another RTD channel on the board itself, from a RTD on another input board, or from a combination of multiple temperature inputs. If needed by the application it is possible to implement complex CJT selection criteria in the CLB, such as voting and/or averaging between multiple RTD inputs.

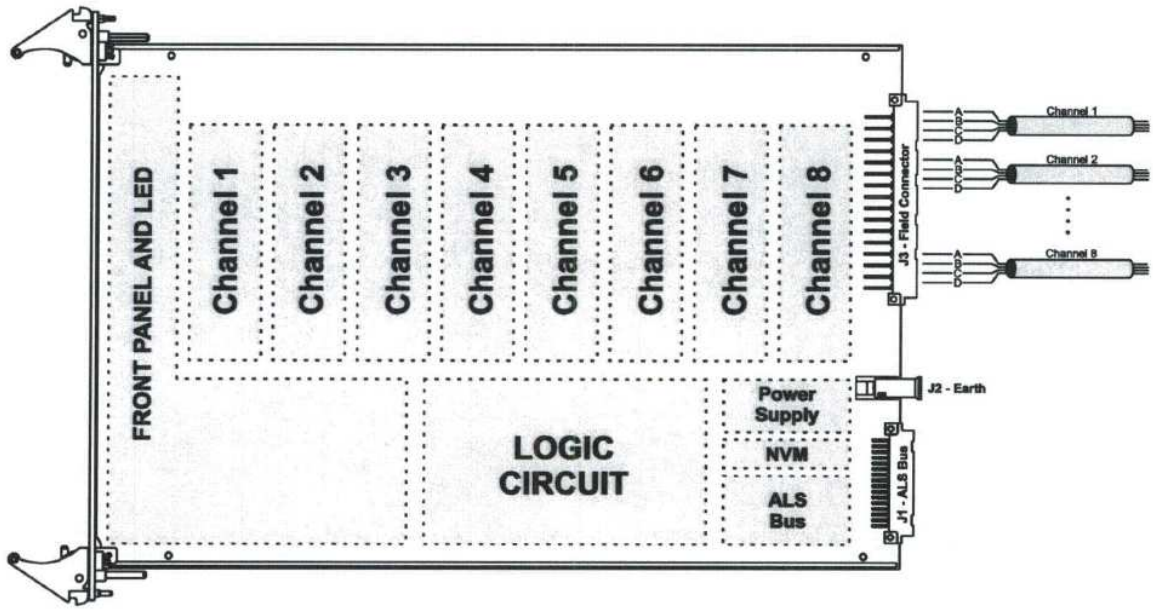

Figure 37 – Advanced Logic System RTD/TC input board

Source: Reference [30]

Analog input board (voltage/current)

Voltage or current analog input board is a high-integrity, 8-channel analog input board (Figure 38). Each input channel can be individually configured for voltage or current input operation.

The board channels can operate in the following standard modes:

- Current mode: [4 to 20 mA], [0 to 20 mA], [10 to 50 mA] or [0 to 50 mA].
- Voltage mode: [0 to 5 VDC], [-5 to +5 VDC], [0 to 10 VDC] or [-10 to +10 VDC].

ADC resolution is 20 bits.

Process instrument loop power is provided by external loop power supplies from a cabinet-mounted or remote power-supply.

Each channel has the following characteristics:

- A failure in one channel will not affect other channels.
- The calibration of one channel will not affect other channels.

- Each channel is independently (and individually) calibrated.
- Each channel supports out-of-range detection, as well as automatic recovery from an overload condition.
- Each channel performs a self-test to ensure integrity.

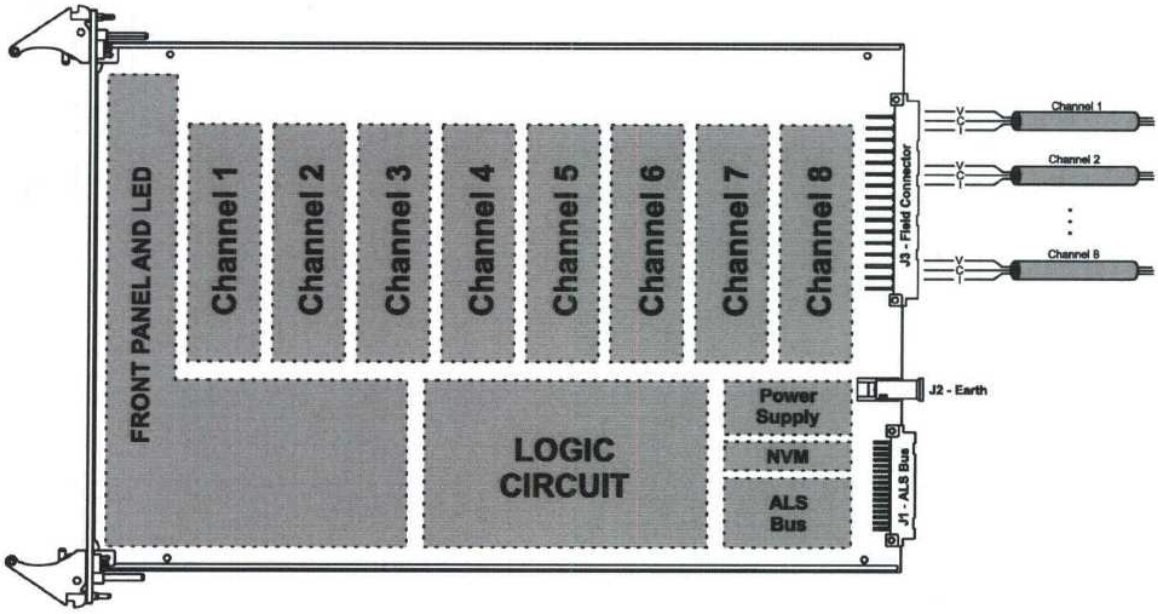

Figure 38 – Advanced Logic System voltage/current analog input board

Source: Reference [30]

OUTPUT BOARD

Output boards (OPBs) are responsible for driving outputs of the system. There are different types, such as 24 or 48 VDC digital outputs, relay outputs with 125 VAC or higher ratings, analog signals, both in voltage or current, etc.

LEDs are provided at the front of the card for channel monitoring. Number of channels range from 1 to 16, depending on the type. External circuit integrity monitoring can also be provided.

As for input cards, output channel has two main circuits:

- An analog circuit, responsible for signal conditioning from digital 3,3 VDC control voltage levels to the electrical output needed. The analog circuit is responsible for all integrity sensing and feedback loops, which provide information about the state of the output circuit.
- A digital circuit located in the FPGA that performs all channel control, integrity checks, self-testing, digital filtering and bus communications. All functions are implemented with redundant logic within the FPGA. The redundancy and self-test circuits ensures the detection of any single failure on the board.

ALS has the capability of driving field devices directly from the rack without the use of interposing relays. This is accomplished with the use of well protected FET transistor devices with proper isolation.

Output channels are divided up into groups (1 to 4 groups typically), with common ground and galvanic isolation (1,500 Vrms) from the other groups and from the digital portion.

OPBs are slave devices both on RAB and TAB busses and will respond as required by the masters (CLB or STB, respectively).

As in previous cards, the OPB has a dedicated external flash device (NVM) to store specific configurations. Local copies are maintained/utilized in the FPGA. Transfers take place during card energization.

OPBs incorporate a failsafe feature for autonomously assume a predefined state in case of system failure. The fail safe state for a particular OPB is defined during the system design cycle.

Output board (contact)

This board utilizes solid-state relays (SSR). Output channels are isolated from the ALS logic domain with optical-isolators capable of withstanding at least 1,500Vrms. Furthermore, all 16 output channels are individually isolated and are divided into 2 groups of 8 channels with isolation.

SSRs are single pole single throw (SPST) contact capable of switching up to 125 VDC or 120 VAC with a 1 A load current.

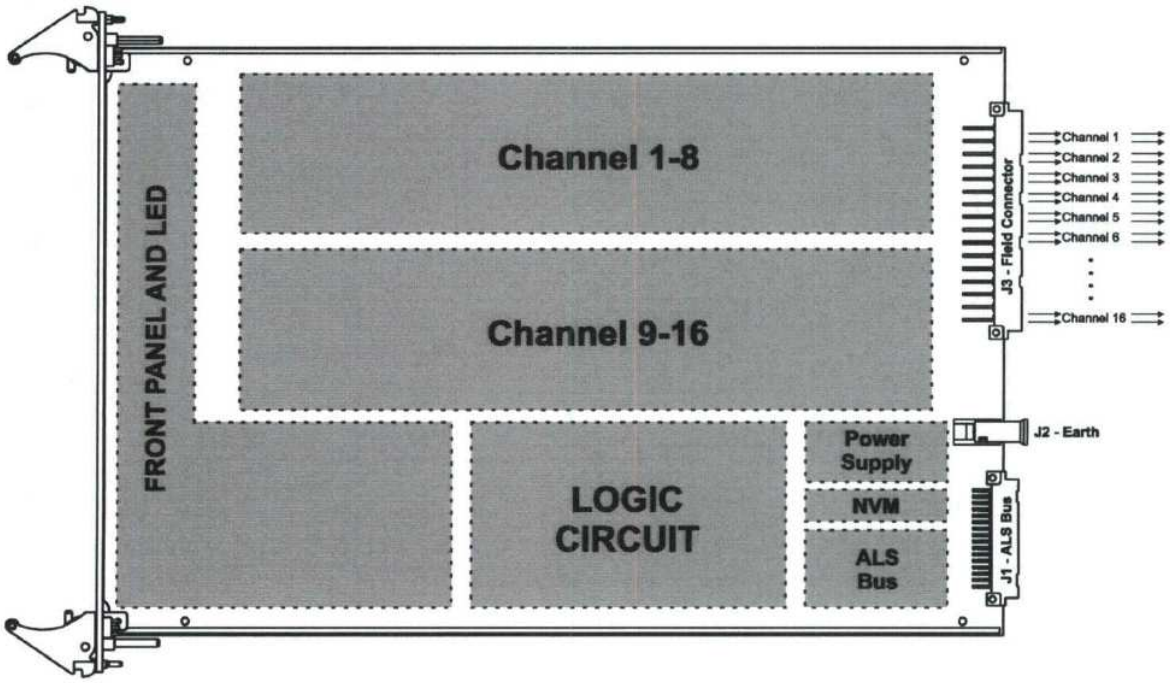

Figure 39 – Advanced Logic System contact output board

Source: Reference [30]

The board compares the demand from the CLB to the isolated feedback signal from the output circuit to check for possible mismatch. Board checks also include field continuity testing for each channel.

Digital output board (relay driver)

This board incorporates 12 field effect transistor (FET) driven output channels for energizing and de-energizing relay loads powered from a shared set of dual-redundant external power supplies.

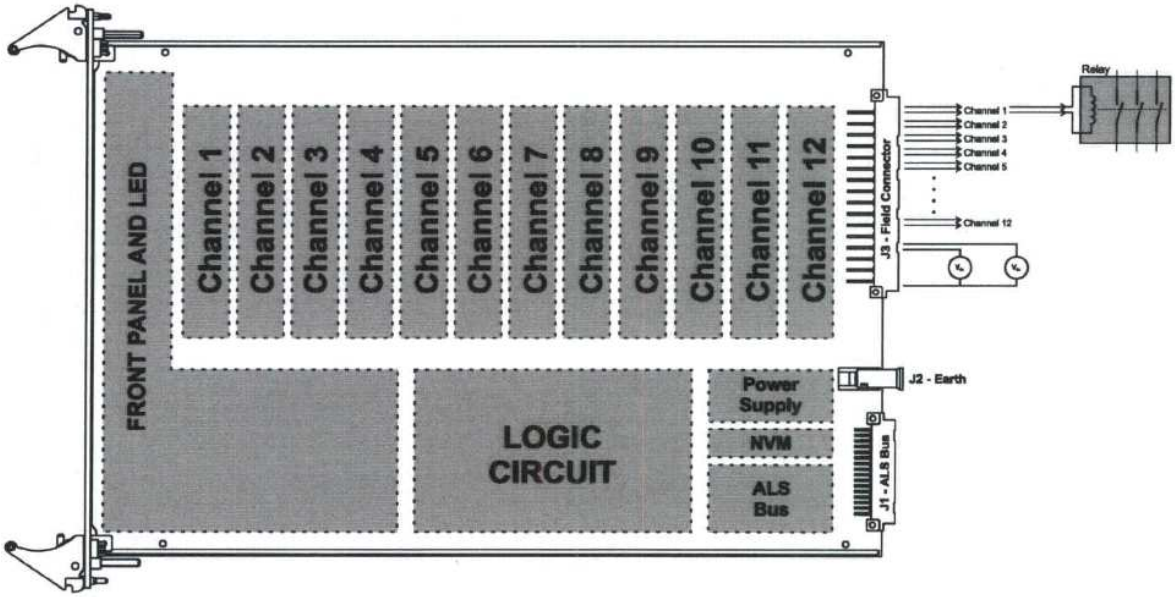

Figure 40 – Advanced Logic System relay driver output board

Source: Reference [30]

The board self-testing ensures channel output integrity. This integrity data is fed back to the CLB embedded in the RAB response packets. Additionally, it can sense continuity through the relay coil and detect a shorted or open-circuit coil.

Board is also capable of performing channel testing to ensure that the channel could change state if demanded, without affecting the state of the channel.

Analog output board

This is an 8-channel analog output board. Each output channel can be individually configured for voltage or current output operation:

Current mode: [4 to 20 mA] or [0 to 20 mA].

Voltage mode: [0 to 5 VDC], [0 to 10 VDC], [-5 to 5 VDC] or [-10 to 10 VDC].

The 8 output channels are independent, but located on a common isolation domain.

Like in the analog input boards, each channel has the following characteristics:

- A failure in one channel will not affect other channels.

- The calibration of one channel will not affect other channels.
- Each channel is independently (and individually) calibrated.
- Each channel supports out-of-range detection, as well as automatic recovery from an overload condition.
- Each channel performs a self-test to ensure integrity

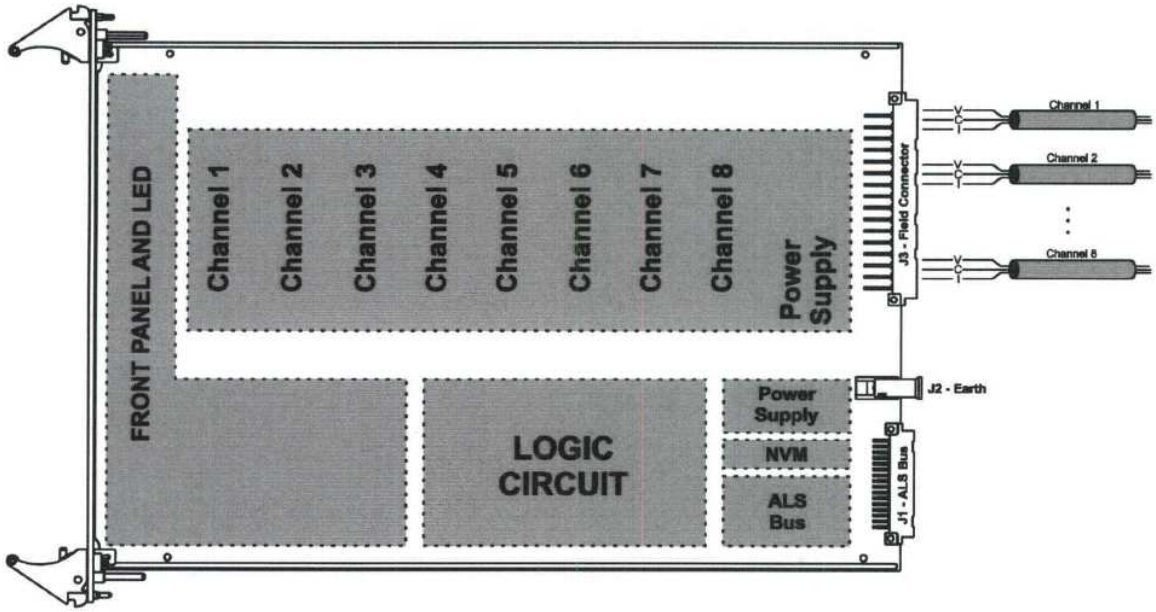

Figure 41 – Advanced Logic System voltage/current analog output board

Source: Reference [30]

COMMUNICATION BOARD

Communication board provides capabilities for reliable and secure communications. It can be used for establishing an isolated and reliable communication link between two ALS systems. It also provides capabilities for remote input/output.

The communication boards can be configured for secure one-way communication with electrical isolation, which can be used for sharing information with non-safety systems or equipment, like control systems or HMI systems.

Communication board is a slave device on the both RAB and TAB busses.

Different communication schemes can be configured:

- Repeat all RAB and TAB communication on the Comm Link (mirror port).
- Selectively collect data transmitted on the RAB and TAB and send that data at regular interval.
- Received data directly from the CLB and only send that data when commanded by the CLB.

Communication links are isolated from ALS logic and capable of withstand 1,500 Vrms. They can be configured according to different serial standards, like RS-232, RS-422 and RS-485. 8 channels per board are available.

UART packet mode is preferred when the board is used to transfer data to another ALS rack.

Each channel has a dedicated buffer to store incoming or outgoing information.

The board does not support higher level protocol features, such as automatic re-transmission in case of data error.

Once again, the communication board has a dedicated external flash device (NVM) to store specific configurations. Local copies are maintained/utilized in the FPGA. Transfers take place during card energization.

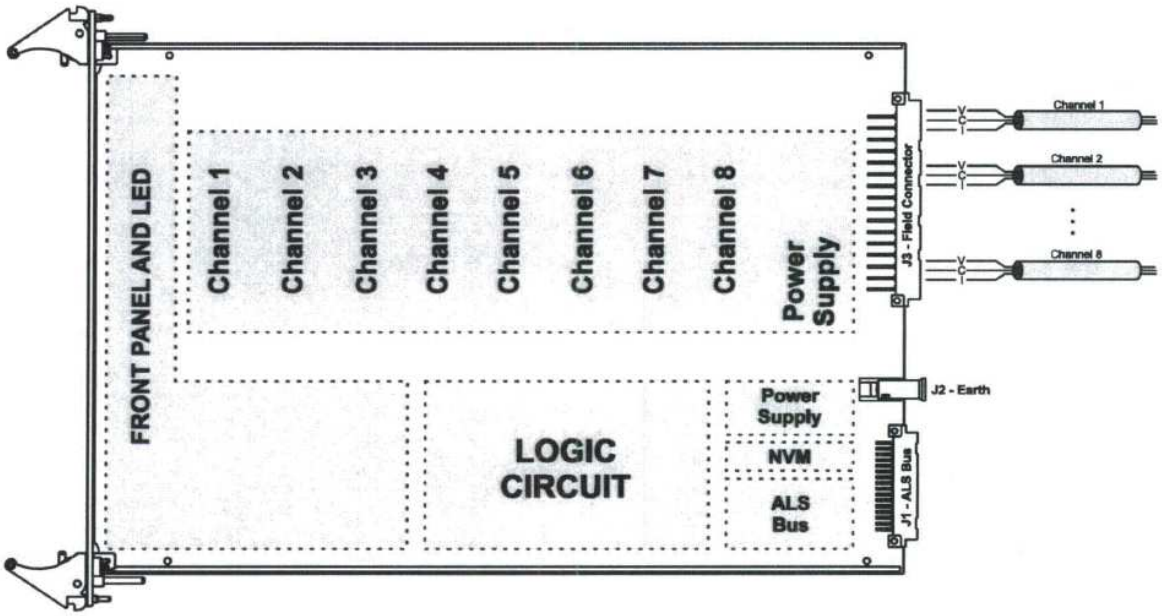

Figure 42 – Advanced Logic System communication board

Source: Reference [30]

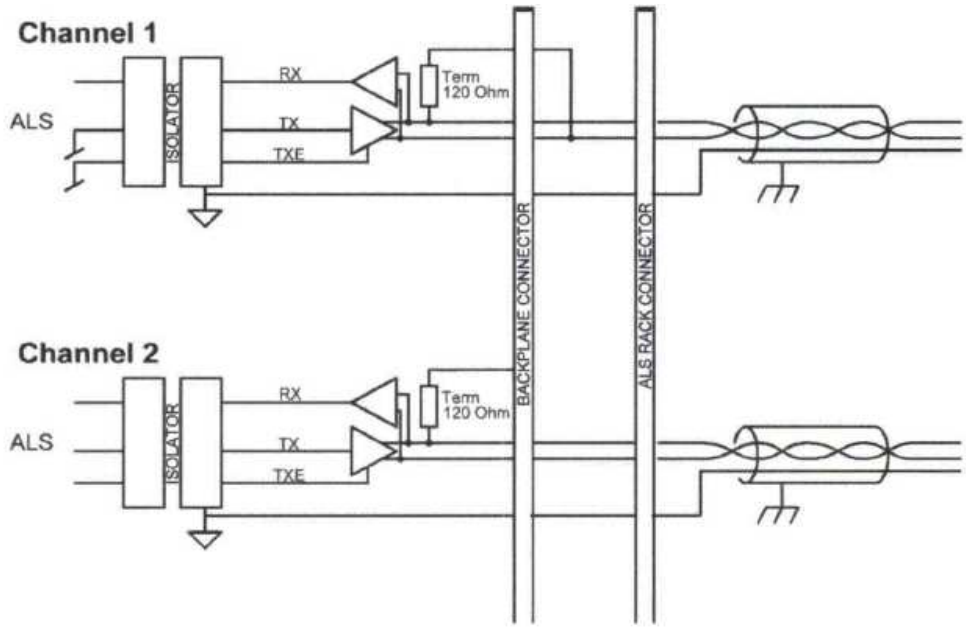

Figure 43 – Advanced Logic System communication board channel isolation detail

Source: Reference [30]

POWER SUPPLY BOARD

Power supply units (PSUs) are used to generate a regulated local supply to be used by all ALS boards in the rack. The PSU can be connected directly to a Class 1E power source. A redundant pair of PSUs is normally used to provide load sharing and redundancy.

PSUs are hot-swappable and are provided with diagnostic (normally open contact).

PSUs can also generate higher voltages (24 VDC, 48 VDC) for digital outputs or to feed dry contact inputs, eliminating the need for additional external redundant qualified power supplies.

Different powering scheme can be implemented without the use of PSUs, as can be seen in Figure 44. This is the current scheme in current Westinghouse/CSI topical report.

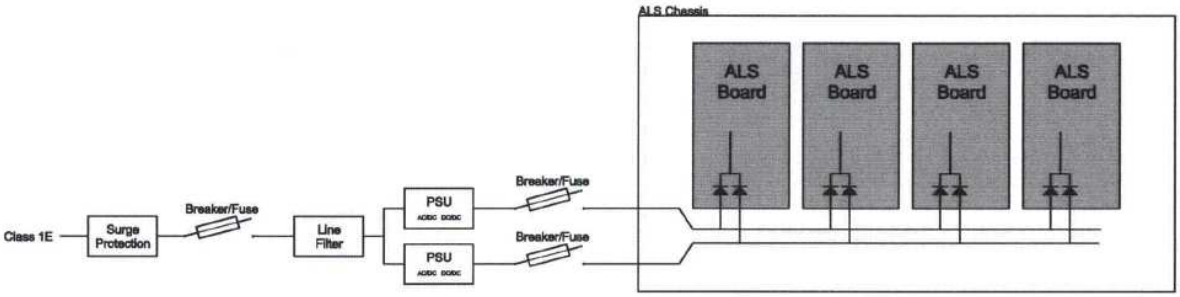

Figure 44 – Advanced Logic System power scheme without power supply units

Source: Reference [30]

DIAGNOSTIC AND TEST EQUIPMENT

ALS Service Unit or ASU could be implemented in a standard computer with ASU software application installed, providing the following when connected to STB boards:

- State information – monitoring of real-time operation, including all I/O signals.
- System and board information – detailed information about the static parameters of an ALS system, including board FPGA programming, board build information, and board set-point configurations.
- Black Box – retrieving and presenting the black box information.
- Testing – surveillance tests that can be initiated using the ASU.

As ASU only interacts with STB board, its operation is passive and non-intrusive. It cannot modify the system configuration, nor can it override any of the safety-related functionality within the system.

ASU can generate configuration reports, retrieving information from NVM chip related to the card, like part number, revision, serial number, FPGA part and revision number, etc. It also can collect data on the configuration and setpoints parameters of the card.

The ASU can be implemented also in a way that it communicates directly with the CLB when the COMM Enable keyswitch has been activated on the Control Panel.

Application specific periodic surveillance tests can be implemented to be performed through the ASU. Based on the needs of the application features may be implemented in the CLB that allows surveillance testing to be performed and/or monitored through the ASU.

The ASU is used to readout and change application setpoints and channel calibration coefficients. The CLB holds the application setpoints and according to the application, it will allow the ASU to modify these setpoints. The ASU is also used during input/output channel calibration where it is used for selecting the board and board channel to be calibrated and to change calibration coefficients based on the readings of external MTE.

ASU can only modify the safety system tunable parameters stored in NVM for which it is designed (for instance, input/output calibration coefficients, setpoints and tuning constants). It is not possible to modify the safety channel algorithm.

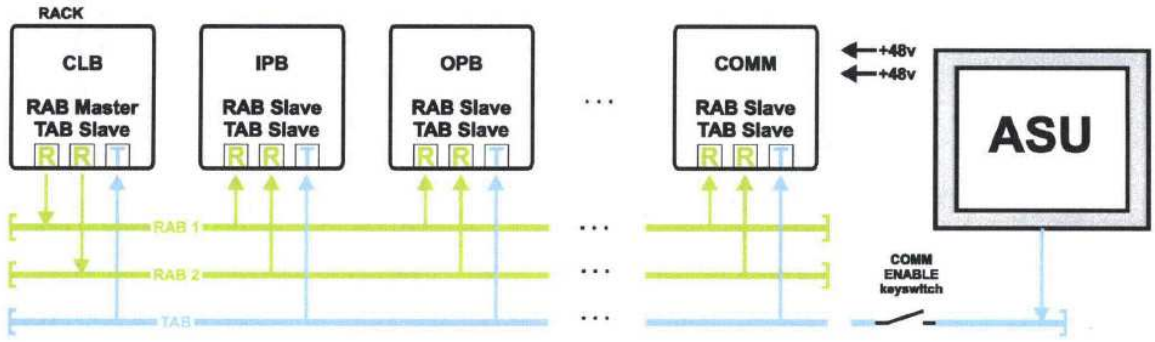

Figure 45 – Advanced Logic System ASU communication scheme without the use of STB

Source: Reference [32]

CONNECTORS

Connectors are Phoenix Contact Variocon™ style connectors, which are modular connectors. Variocon™ connector shells can accommodate between 2-5 modules where each module includes between 2 and 8 connections.

Figure 46 – Advanced Logic System connector and rack rear view

Source: Reference [32]

POWER MANAGEMENT AND GROUNDS

Power management architecture is as following:

- Power supply units (PSUs) generates 5 VDC for the rack (power bus on back plate) from external power feed. Upon entry onto the ALS board the 5V is fused, filtered and overvoltage protected. The fuse ensures that catastrophic failures on a board cannot disrupt the rack power. Filtering is done to avoid noise propagating both from the rack to the boards or in the reverse way.
- Each ALS board has a DC-DC voltage regulator and filters that generates stable voltages for the different digital domains (5, 3.3 and 1.5 VDC). 3.3 VDC domain is used for FPGA I/O ring components, while 1.5 VDC domain is used for FPGA core.

Ground domains are as follows:

- Chassis ground.
- Digital ground – used for ALS logic circuits and comes from PSUs. 3.3 and 1.5 VDC supplies are referenced to this ground.
- Isolated field ground (one or more than one) – used for field conditioning circuits in I/O boards. Inputs and output are isolated.

In current Westinghouse/CSI topical report, a different power scheme is presented. Rack power supply is a direct 48 VDC supply from qualified external power supplies.

HIGH INTEGRITY CHARACTERISTICS

ALS platform has several features to ensure high integrity of the system:

- Reliable communication scheme.

  Both RAB and TAB interfaces provide a layered failure detection scheme. This layered failure detection scheme is designed to detect any possible failure that can occur on the communication bus, the communication devices, or the communication logic circuits within the FPGA.

  Four detection schemes are responsible for detecting a communication failure:

  - Redundancy failure detection.
  - Synchronization failure detection.
  - CRC failure detection.
  - Protocol failure detection.

  Detection and reaction of redundancy and synchronization failures are instantaneous, while detection of CRC or protocol failures are not (payload data will be transferred before the CRC is calculated), but the reaction will be instantaneous (receiver will not allow the data to propagate).

  RAB is also redundant.

- Redundant logic.

  All FPGAs are implemented with redundant digital logic. This is to protect the ALS board against a type of failure which can potentially occur over time as a result of manufacturing defects, radiation damage or flash cell charge degradation. This focuses on how the redundancy is implemented internal to the FPGA and has nothing to do with other levels of redundancy such as the redundant input or outputs, or application level redundancy. Difference between the redundant circuits will cause the ALS to take appropriate action. The redundancy implementation will detect any deviation between the redundant circuits before a possible erroneous signal can propagate to the rest of the system.

- BIST (Built-In Self-Test).

  BIST is used for exercising all critical functions within a board. This is done to ensure that latent failures cannot build up in the system and make the system inoperable without the knowledge of plant personnel. BIST typically applies input stimuli on the inputs to a sub-circuit and validate the correct response on the outputs.

- Self-testing.

ALS is intended to eliminate the need for surveillance testing required by Technical Specifications[1] making use of a combination of redundancy and self-testing which automatically and transparently verifies the critical functions of system.

Nevertheless, an ALS system can be specified to support surveillance testing as needed by the application, placing the channels in bypass or partial trip, depending on Technical Specifications. This can be done by toggle switches mounted on a Control Panel in the cabinet.

Critical functions in a particular application are defined by the customer and become key requirements when specifying the application specific system.

The self-testing of the ALS platform can be divided into segments as shown on Figure 47.

- Input and output boards will determine the integrity of the external wiring and perform an application specific action in case of failure.
- Input and output boards are designed to automatically detect circuit failures between the channels and the ALS Bus. This is done with a combination of redundancy and self-test. The channels typically include a self-test circuit where the channel is disconnected from the field and tested. These tests are done in a way that the test time does not affect the response time of the system. Such self-tests occur in a timely basis and on every change of state of the input.
- ALS bus communication is protected with redundancy, CRC and timeout detectors. The failure detection mechanism on the ALS bus ensures that any one board cannot cause an unannounced and failure.
    - If the CLB fails the slave boards will detect a timeout event an all enter their fail-safe state. The redundant alarm in the STB will ensure that the failure is announced, even if the CLB is not capable of doing so.
    - If an ALS slave board fails, the CLB will detect the failure and perform a predetermined action.
- The logic function in CLB is application specific. CLBs are typically based on state machines and other basic building blocks. A self-test strategy can be designed specifically to ensure no hidden failures in the critical functions. The FPGA logic inside the CLB is protected by the standard dual-core redundancy.

[1] This would require additional licensing effort, as Technical Specification should be modified to accommodate to this new situation. Typical STS (Standard Technical Specifications) or ITS (Improved Technical Specifications) are not suitable for this elimination.

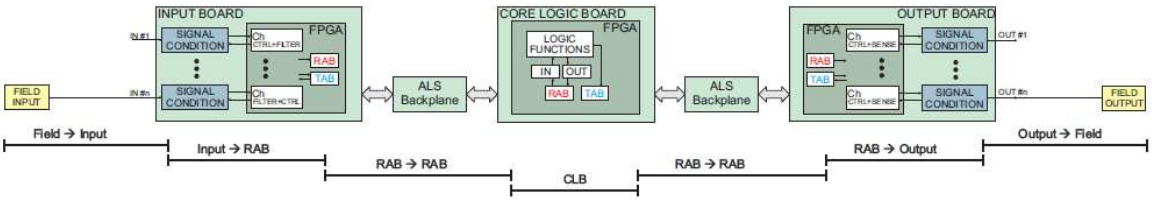

Figure 47 – Advanced Logic System segmentation for self-testing strategy

Source: Reference [32]

ALS OPERATION

ALS operation is a cyclic three phase operation that takes place each system frame:

- Field inputs sampling.

  On a given input board there will be a number of input channels each responsible for conditioning, sensing, filtering, and sampling the field inputs. In the event an input channel changes the signal conditioning circuit will detect this transition and change state. Each channel is described with state information and integrity information. If a channel fails self-test it will be marked as invalid in the integrity information. The CLB retrieves the state and integrity information during the regular polling of data from the input board.

- Logic control and voting.

  Input state and integrity information are retrieved from the input boards and are stored in the input register bank in the CLB. When all input data is present the application specific core logic circuit within the CLB will perform its logic function. Based on the current state of the system, input states and integrity information, the CLB will determine a new output state for all outputs. The application specific logic functions consist of timers, random logic gates, FSMs, 2/4-voters, etc. The decision making process is instantaneous. All system level integrity and data checks are performed during this phase of operation.

  The results of the application specific logic circuit are stored in an output register bank within the CLB FPGA and from there the information is transmitted to the output boards.

- Output.

  The output boards receive information from the CLB. The digital circuits will immediately drive the signal conditioning circuit and perform the intended output function.

SYSTEM MODES

ALS platform utilizes a system mode concept. The system mode is communicated between all boards within the rack as part of the integrity information and provides diagnostics information to the plant staff. The system mode can be during operation in one of the following four modes:

- FCO – Full Capability Operation.

  The ALS platform is operating in a normal mode of operation, and is ready to perform the intended safety function. All circuits are 100% functional and operational. Input channels are updated and evaluated and are in accordance with expected values. Output channels are controlled in the manner for which they are intended, all feed-back information is as expected, and the Core Logic is fully functional.

- RCO – Reduced Capability Operation.

  The ALS platform has detected one or more problems and indicates that it operates in reduced mode of operation. The failures have been isolated to prevent the failures from propagating through the system and causing unintended plant events. The ALS rack may perform specific actions depending on the location of failure, based on customer requirements. These actions could include:

  - Entering a partial trip condition.
  - Entering a fail-safe condition.
  - Performing a trip.
  - Providing detailed status indication (operability indication to control room).

  The system continues to perform as specified and all unaffected circuits will continue to perform their function. Input channels are updated and evaluated. Output channels are controlled in the way they were intended. The Core Logic is fully functional. The ALS platform alarm is active to show that maintenance is required.

- HALT mode indicates that a serious failure has been detected in the ALS platform and that it has entered a fail-safe state.

  In this mode the ALS is inoperable and not capable of performing the intended safety function. All operations will stop and the system enters a fail-safe state. Input boards continue as usual, except they do not respond to RAB requests. The front-panel LEDs continue to be updated. Output boards are placed in a fail-safe state, where all outputs will enter their configured fail-safe mode which can be fail-as-is or fail-as-defined. All RAB communication has seized when the ALS is in HALT.

  The HALT mode is also the power up condition of the ALS platform.

- RESET mode is a transitory state which is only entered when the CLB reset switch is toggled.

  The RESET mode informs all ALS boards that the system will attempt to enter FCO mode and resume full operation. If failures persist the ALS platform will automatically

degrade its system mode to the proper level.

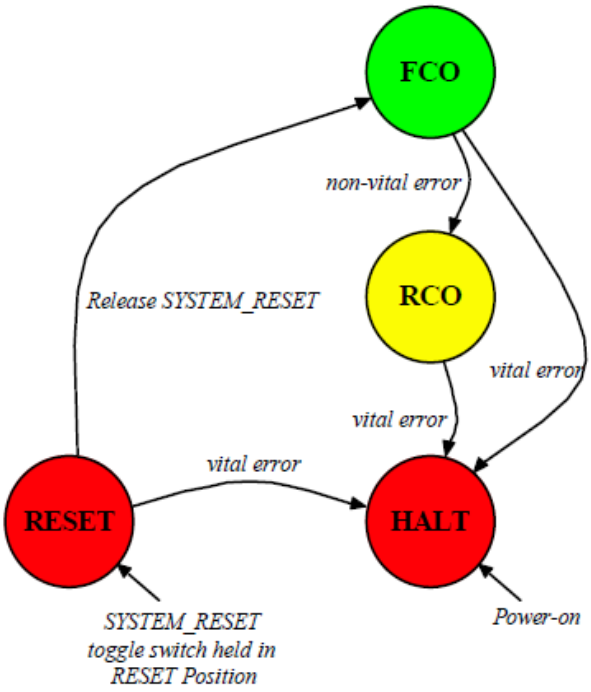

Figure 48 – Advanced Logic System modes finite state machine

Source: Reference [32]

At a lower level all ALS boards maintains a local system mode. The local system mode will ideally track the ALS system mode, but it allows the individual boards to autonomously enter RCO or HALT as soon as the failure is discovered. As soon as the detected failure has been transmitted to the CLB, the CLB will adjust the system mode accordingly. The system mode master-copy is physically located in a register within the integrity monitor module in the CLB. Local replicas of the system mode are maintained by each ALS-board within the slave integrity monitor module. System mode is communicated to all ALS boards in the system during transactions on the RAB.

System mode will initially be HALT mode when the system powers up or when the CLB board is inserted into the rack.

TESTMODE is a special mode which basically provides access to otherwise hidden and inaccessible registers in the ALS boards, without actually probing or forcing external voltages onto boards. TESTMODE is only active when explicitly enabled from an external device using special tools (ALS Test Unit or ATU). ALS boards will never enter TESTMODE when in a rack.

Appendix 3

Diablo Canyon Reactor Protection System upgrade project

Diablo Canyon NPP is replacing its existing Eagle21™ (microprocessor-based) Process Protection System (PPS)[1]. Project began some years ago, and is intended to use Tricon™ platform from Triconex (Invensys Group, recently acquired by Schneider). Tricon™ is a TMR platform that has been previously qualified and accepted by NRC through a safety evaluation report in December 2001. Solid State Protection System is not in the scope of the project [33].

Although Tricon™ platform has been already generic qualified by NRC, any specific application should be individually approved by NRC through a Licensed Amendment Request. NRC did not found enough diversity during the evaluation of the D&DiD assessment for the generic platform.

Because the primary system is microprocessor-based, and so subjected to software common-cause failure, current regulation (SRP NUREG-800 and associated BTP) requires a diverse automatic actuation circuit (DAS) for those functions with no additional means of diversity[2] .

Diablo Canyon changed the strategy to a new FPGA-based DAS based on ALS and submitted a new D&DiD assessment to NRC. This new approach was approved by NRC in April 2011 [34] [35].

Figure 49 represents the current architecture for Diablo Canyon NPP based on Eagle21™ system.

The proposed new architecture for Diablo Canyon NPP based on new Tricon ™ system and making use of ALS platform for implementation of required DAS system is represented in Figure 50.

Figure 51 shows a simple block schematic of Tricon ™ system, in which it can be seen the TMR architecture.

An example of implementation in a particular protection channel is represented in Figure 52.

[1] Original system was based on analog Hagan 7100 system from Westinghouse and was replaced with Eagle21™ in 1994.

[2] Credit to manual operator actions could be given for certain functions, depending on the D&DiD analysis and applicable regulation. This is the reason why in the past some safety systems could be replaced using microprocessor-based systems with no further diversity.

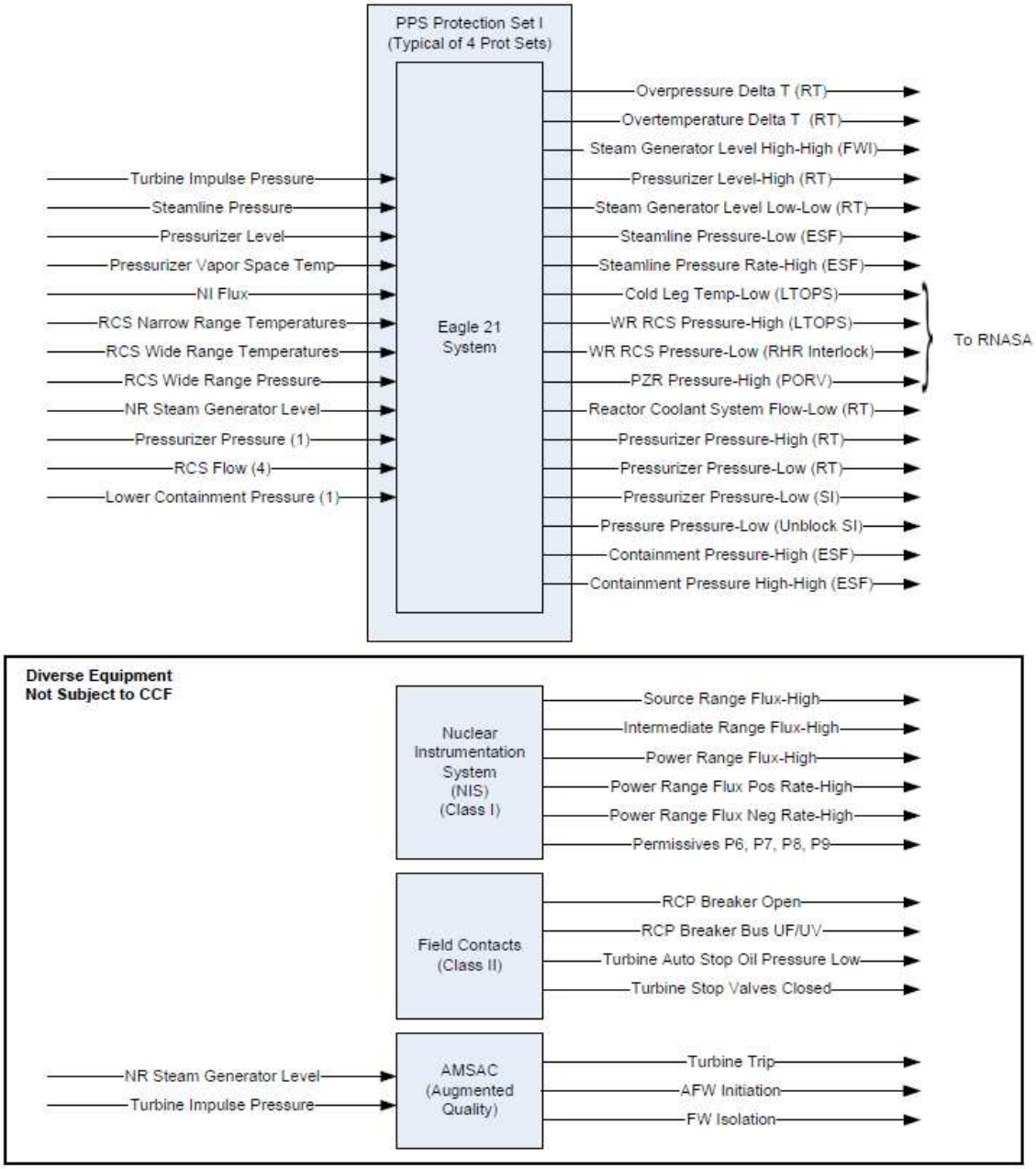

Figure 49 – Current Diablo Canyon architecture

Source: Reference [34]

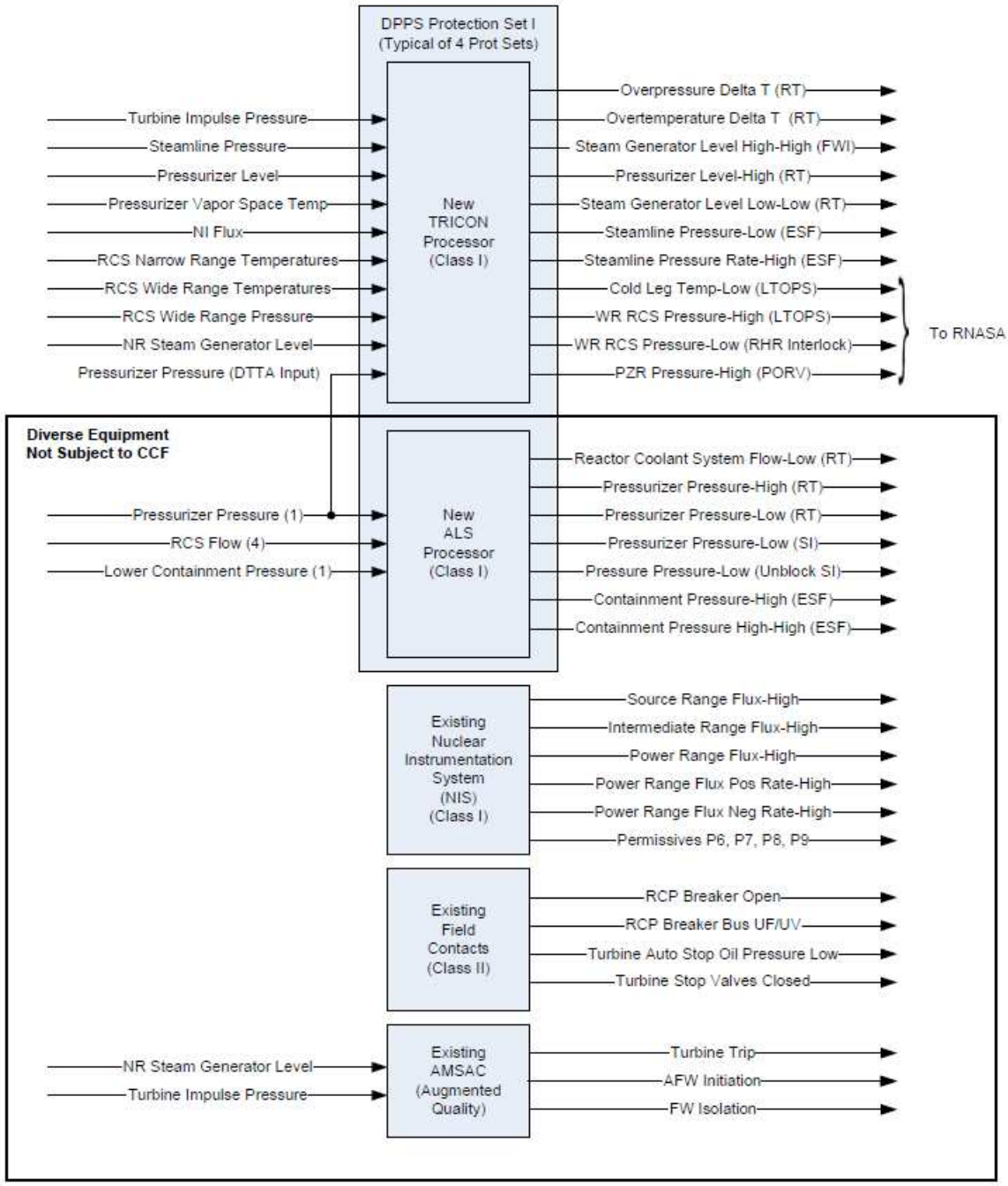

Figure 50 – Proposed Diablo Canyon new architecture after Eagle21^{TM} project replacement

Source: Reference [34]

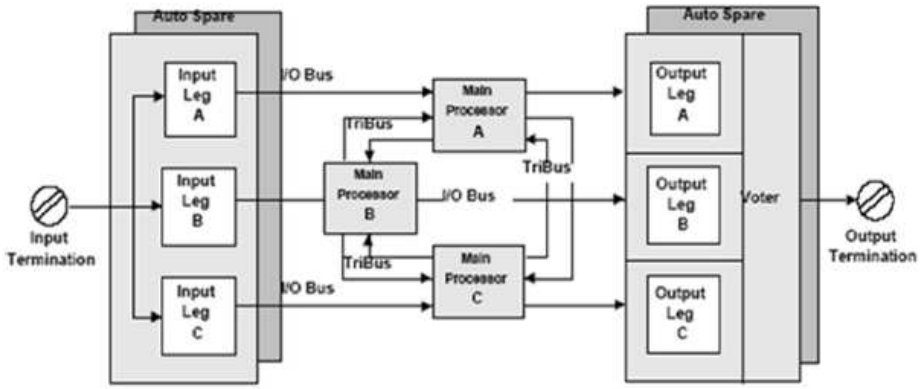

Figure 51 – Tricon™ system simple block scheme

Source: Reference [33]

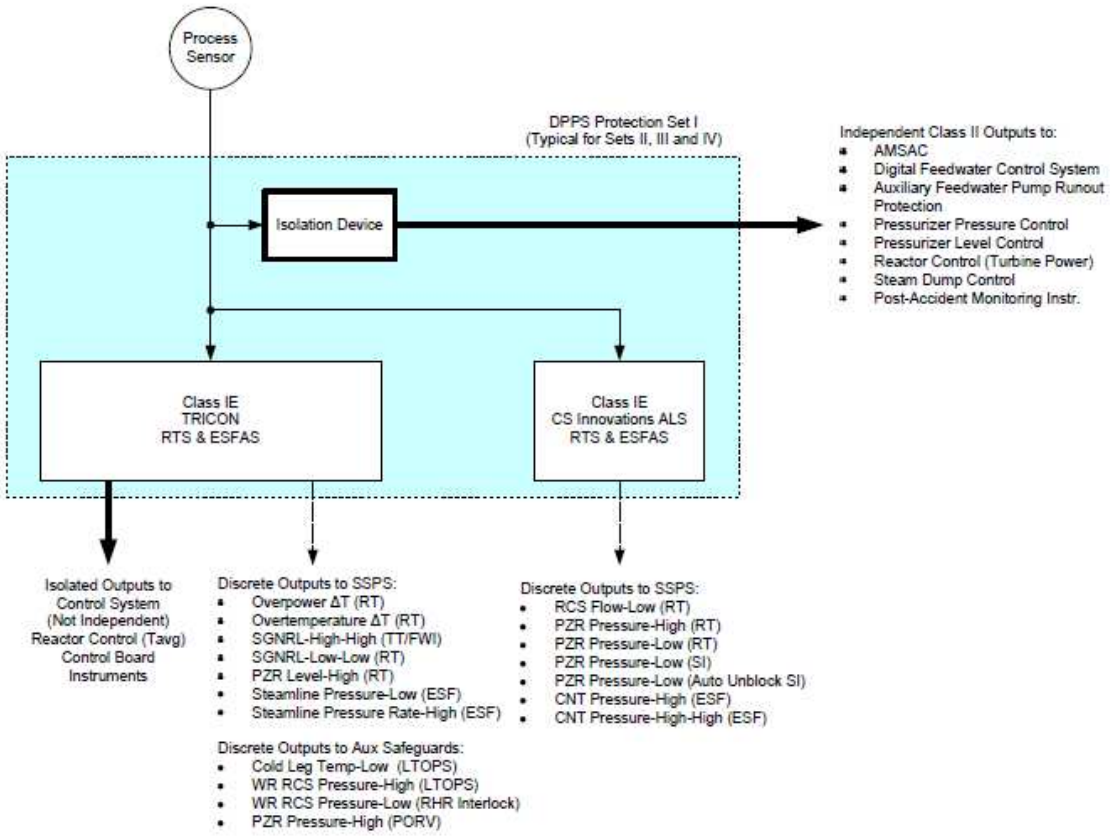

Figure 52 – Detailed implementation of a channel replacement

Source: Reference [34]

Appendix 4

Digital Control Computer Upgrade in Darlington (Canada)

The Darlington Nuclear Generation Station is a four-unit CANDU (CANadian Deuterium Uranium) reactor type plant located in Ontario (Canada). It uses a redundant pair of Digital Control Computers (DCCs) for each reactor. The DCCs are based on Digital Equipment Corporation (DEC) PDP-11/70 computers, working as a mater/standby configuration. An overall system diagram is represented in Figure 53.

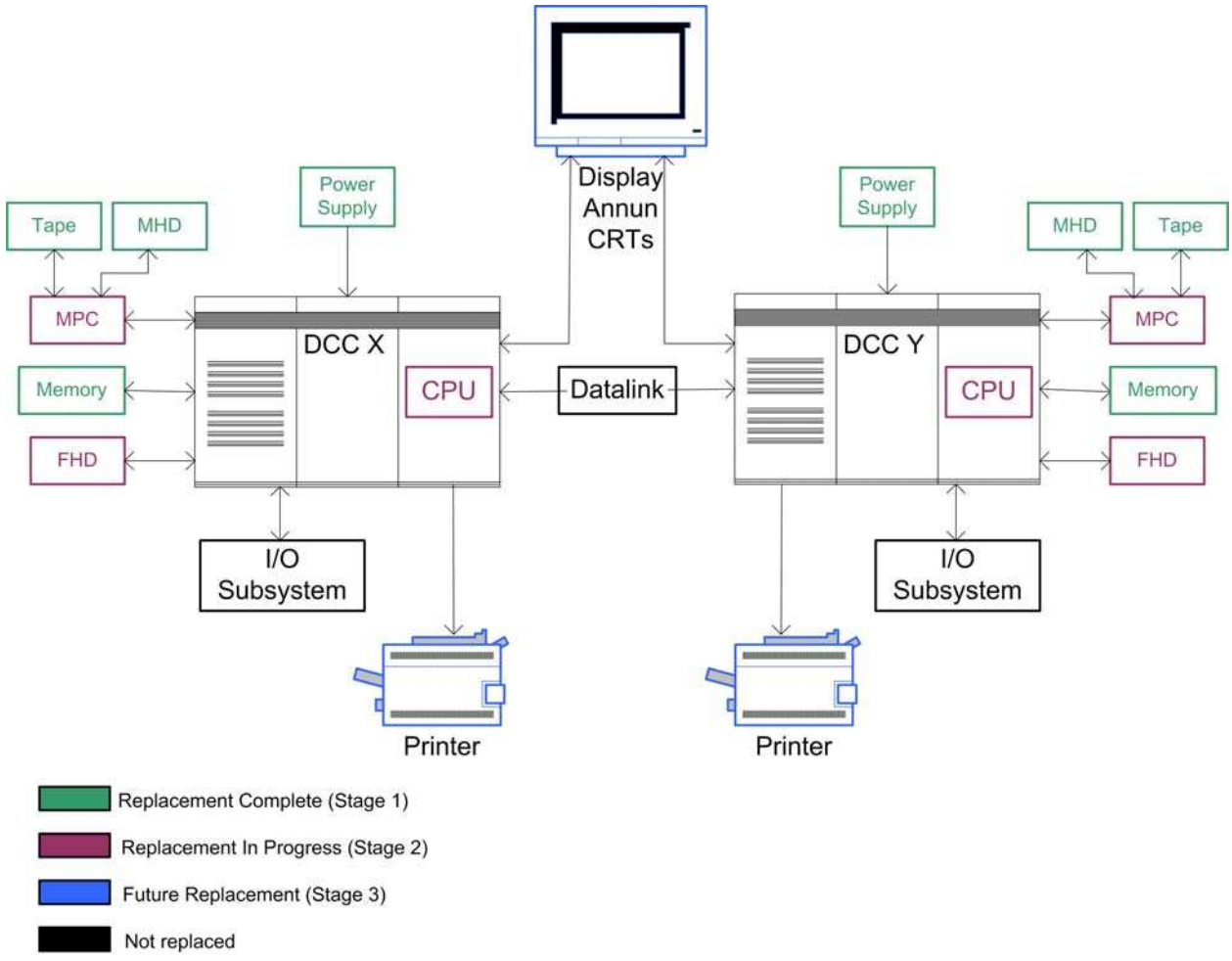

Figure 53 – Darlington overall control system architecture

Source: Reference [36]

The DCCs perform reactor control functions such as reactor power regulation, steam generator pressure control, and deaerator level control. They also provide alarm annunciation and data display via an operator interface. Each system consists of two independent digital computers, with each being capable of taking control of the unit in a master/standby arrangement. The two computers communicate with each other via a computer-to-computer data link.

The DCCs include a self-checking subsystem. Faults in either software or hardware are detected by a combination of internal hardware and software self-checking plus external watchdog timers. When a fault is detected in one computer that is acting as master, a failover to the standby computer is carried out.

Darlington first unit began commercial operation in 1990. By the end of the 90s, obsolescence issues started becoming a serious problem, so a replacement project was put in place. Main objectives of the project included performing the installation on-line, keep I/O subsystem and reduce at a minimum software changes.

The operator, Ontario Power Generation (OPG), has also been used FPGAs in the control computer replacements at Pickering station, and is even considering using FPGAs in a reactor protection system application in the next years [37].

A first stage in the modernization project addressed items that were most unreliable and carried a high maintenance burden. These items were memories, power supplies, moving head disks (MHDs) and magnetic tape drives. The last two items were replaced with new, custom-designed FPGA-based equipment. The company had experience with FPGAs, as a Unibus interface card for testing purposes had been already developed, making use of SRAM type FPGA technology from Altera.

The RM03 moving head disk and the magnetic tape drive were replaced using the same FPGA-based technology. The Flexible RM03 Emulated Disk (FRED) has no moving parts and uses solid state memory, thus increasing reliability, reducing maintenance and power requirements, increasing memory space and reducing operator burden associated with processing magnetic tapes. High-speed data extraction functionality to an external historian was also added.

Second stage of the replacement project addresses obsolescence of the following items: fixed head disks (FHDs), Massbus Peripheral Controllers (MPCs), Central Processing Units (CPUs), Operator Console Panels (OCPs, which is a relatively simple HSI) and Unibus backplanes. For this stage, the plant decided to contract the original systems integrator for the DCC systems, L-3 Communications MAPPS (formerly CAE), and Quickware Engineering and Design (QED).

The replacement solution for the CPU makes use of a PDP-11/70 emulator implemented in an FPGA, developed by QED, thus allowing the existing control software to be maintained. The plant has previous experience with QED’s emulators, like in the Fuel Handling System.

The Unibus wire-wrapped backplanes are replaced with a new PCB-based 4-slot unit.

Three additional computer systems that are being replaced include the Sequence of Events Monitor (SEM), the Common Processes (CP) computers and Fuel Handling (FH) computers. These also use PDP-11/70 CPUs and will be replaced with the FPGA-based emulator solution. The strategy is to install the upgrade on the less complex computers first, so the sequence is SEM first, then FH and CP, and finally the DCCs. The first DCC replacement on each unit will be done during a unit outage so the new hardware systems can be tested in controlling mode during reactor startup. The second DCC on each unit will be installed at power.

Systems involved in the replacement project are safety class 2 or 3, based on IEC 61226.

FPGA technology used is SRAM-based Virtex 5 family (XC5VLX 30/50/110) from Xilinx. Language used is VHDL. IP cores are used from QED Engineering for the PDP-11 emulator.

Canadian standard CSA Z299.2-85 is applied for the overall design and quality assurance process, and Canadian standard CSA Q396.1.1-89 is applied for software quality assurance.

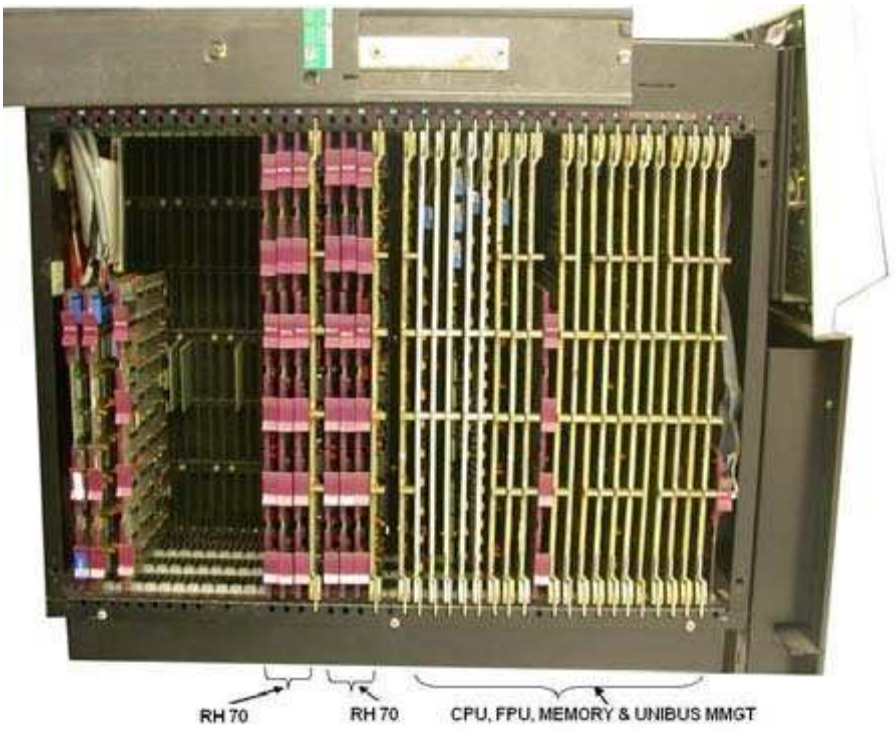

Figure 54 – Front view of PDP-11 CPU chassis

Source: Reference [36]

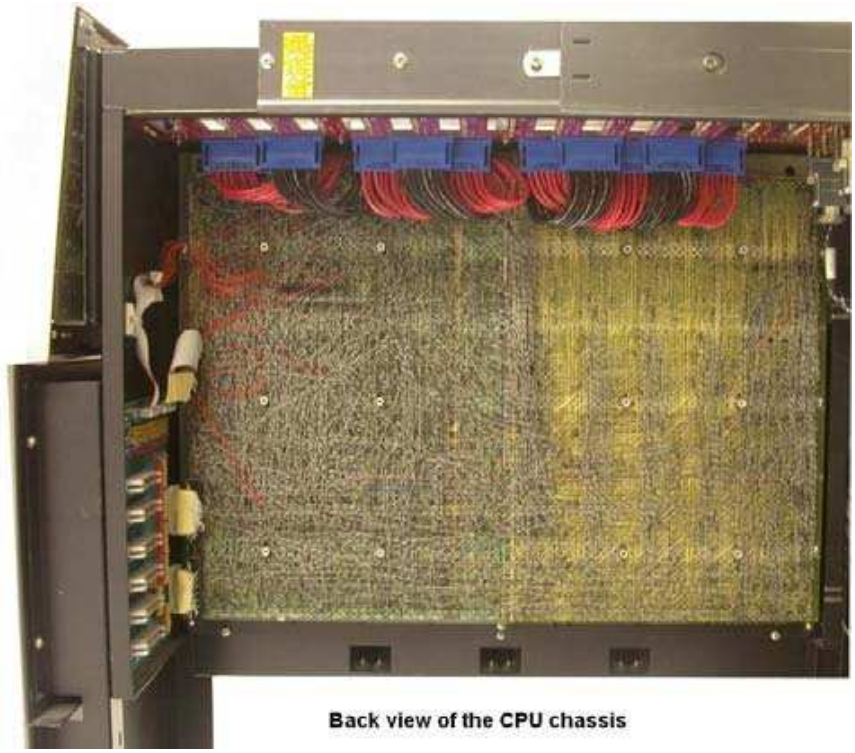

Figure 55 – Rear view of CPU backplane

Source: Reference [36]

Figure 56 – PDP-11 emulator board

Source: Reference [36]

EDAC is widely used. Circuit boards containing FPGAs have been designed with a number of features that support off-line and on-line diagnosis. For example, the emulator supports plant's on-line hardware diagnostic check program and provides the required diagnostic registers. Logic analyzer ports are provided on the boards, and there are a number of status indicators

and error LEDs. Errors that are detected are logged into non-volatile memory.

For each component of the replacement DCC system two design reviews are conducted: a preliminary design review (PDR) and a critical design review (CDR). The PDR is conducted after the initial high-level design of the component is completed to ensure that it is ready to proceed into detailed design. The CDR is performed after the detailed design is completed in order to identify any potential design defects before the system design is finalized.

Although FMEA was not formally adopted as a design review tool, individual reviewers follow steps similar to those used in an FMEA when they conduct the design reviews – they identify potential system failures, analyse their effects, and suggest countermeasures if needed. This practice has helped identify important failure modes. For example, in the PDR for the operator console panel (OCP) it was discovered that a failure of the OCP could lead to an automatic shutdown of the new system, reducing DCC availability and potentially causing a reactor shutdown. The OCP design was modified such that failures in the OCP will not cause a system shutdown.

Some considerations for the validation testing strategy are:

- The tests focus on functional system-level testing, rather than low-level testing, using an existing high-fidelity simulator that includes models of all plant processes and supports closed-loop testing of the new DCCs.
- Test results will be compared to those from running the same tests on a real PDP-11/70 (baseline results).
- All validation testing will be repeated on the final delivered system and after any subsequent modifications (full regression testing).

  NOTE: Since the software is unchanged (other than the self-check program), test data and data from the existing running computer systems can be used to validate the operation of the emulators.

Goal Structuring Notation is being used to ensure that all safety aspects are implemented properly.

A traceability study is performed to link individual test cases to the system requirements. The objective is to verify that every system requirement is tested, with an ultimate goal of demonstrating that the final product is at least as reliable as the original PDP-11/70 computer.

OPG entered into a 25-year maintenance contract with the supplier of the DCC replacement equipment. The agreement requires that the supplier maintain spares and, if required, come up with substitutions, including reverse engineering to develop replacements if necessary. The supplier has a similar agreement with the Canadian Navy for this type of equipment.

The repair strategy for the FPGA-based equipment is to replace any failed printed circuit cards. OPG does not expect to have capability to troubleshoot and repair the cards, other than the possible exception of changing out components that are easily replaced, such as

capacitors. With modern surface-mount technologies used with FPGAs today, specialized equipment is required to remove an FPGA chip and install a new one. OPG does not intend to maintain this capability, at least in the near term.

To help ensure long-term support, OPG's typical practice is to request that sufficient design documentation, test records, tools, and development and test environments be provided with new FPGA-based equipment so that the utility would have the capability to make design changes or develop replacements should this ever become necessary. If the supplier is not willing to provide this, then OPG typically asks that it be put in escrow so that if the supplier goes out of business or discontinues support for the product, the utility will have the ability to take it in-house or provide it to an alternate supplier to obtain the needed support.

Appendix 5

Power Range Neutron Monitoring System in Advanced Boiling Water Reactor in Japan

Toshiba is one of the primary suppliers of I&C systems to Tokyo Electric Power Company (TEPCO) for the utility's boiling water reactor (BWR) plants in Japan. TEPCO and Toshiba began the process of evaluating and designing digital software-based I&C systems in the mid-80s, starting with radioactive waste processing systems, which is a non-safety system. In 1996 they began with safety-related system implementations. Digital power range neutron monitoring system (PRNMS) was the first one to be developed. Afterwards, even human-system interfaces (HSIs) have been moved to digital. Toshiba's digital platform for new reactors and upgraded system is TOSMAP^{TM}, which is a micro-processor based platform.

In the 80's and 90's the digital systems implemented by Toshiba were microprocessor-based. Some programmable logic devices were used for signal processing in these systems, but the primary functions were CPU-based. However, in the 2003 timeframe, Toshiba began using FPGAs for the primary logic functions within digital radiation monitoring systems. Again, they started with non-safety systems, moving into safety-related radiation monitoring systems using FPGAs in 2004. After gaining experience with radiation monitoring systems, Toshiba developed FPGA-based PRNM systems, with the first installation at an Advanced Boiling Water Reactor (ABWR) plant for a Japanese utility in 2007. Toshiba is using FPGAs in other safety-related systems, including the reactor protection system.

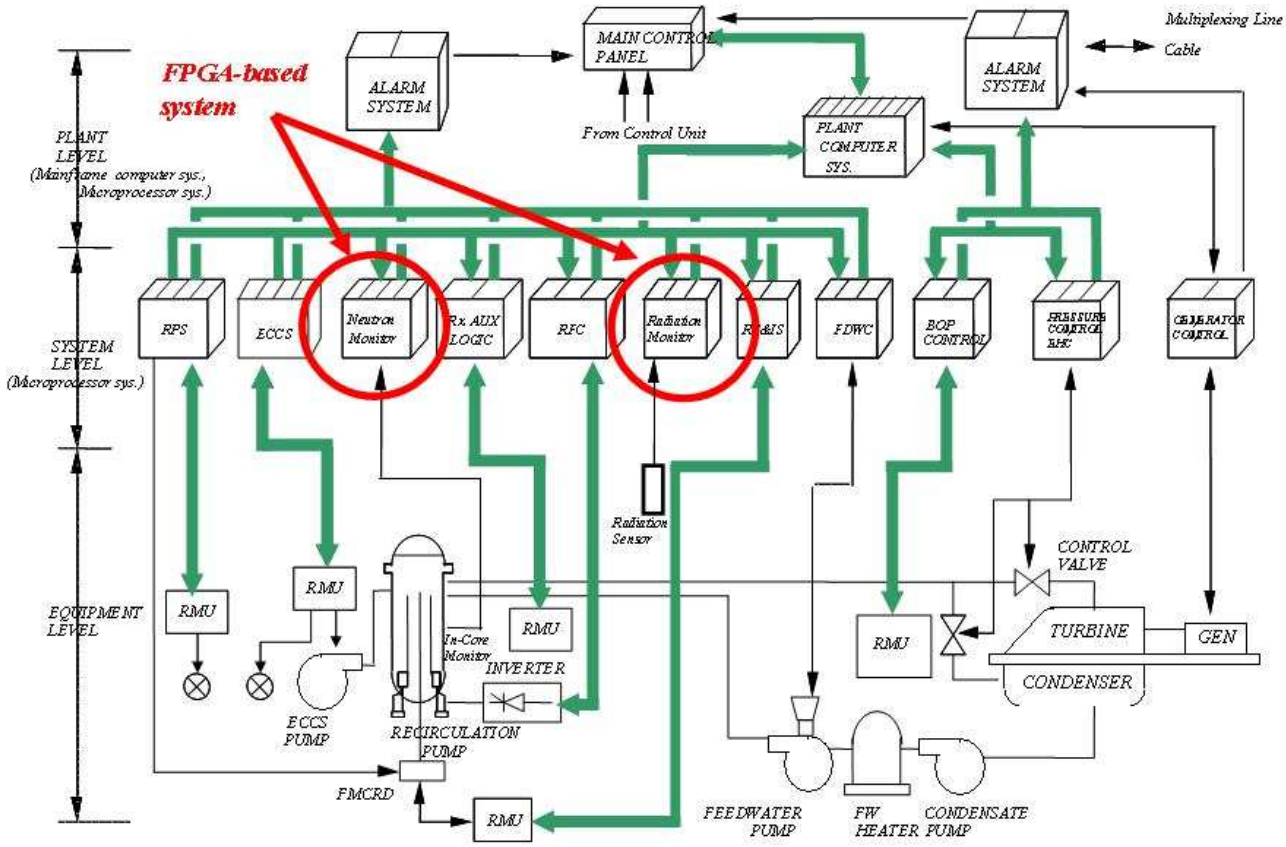

Figure 57 – Instrumentation&Control and Information Systems architecture of an ABWR

Source: Reference [38]

Operating experience with the FPGA-based radiation monitoring and neutron monitoring systems has been very good. No serious problems have been encountered. The only known reported problem was caused by a hardware defect. The printed circuit board was replaced and no further problems have been experienced.

The power range neutron monitoring system receives signals from neutron flux detectors in the reactor to measure local neutron flux in the reactor core, calculate overall reactor flux, process these values and provide signals to the reactor protection system, plant process computer, and other plant systems as required. It includes local power range monitor (LPRM) units and average power range monitor (APRM) units, as well as a flow unit and a rod block monitor (RBM). FPGAs are used in the LPRM and APRM units, as well as in the RBM. The LPRM and APRM modules (circuit cards) are mounted in a chassis (Figure 59).

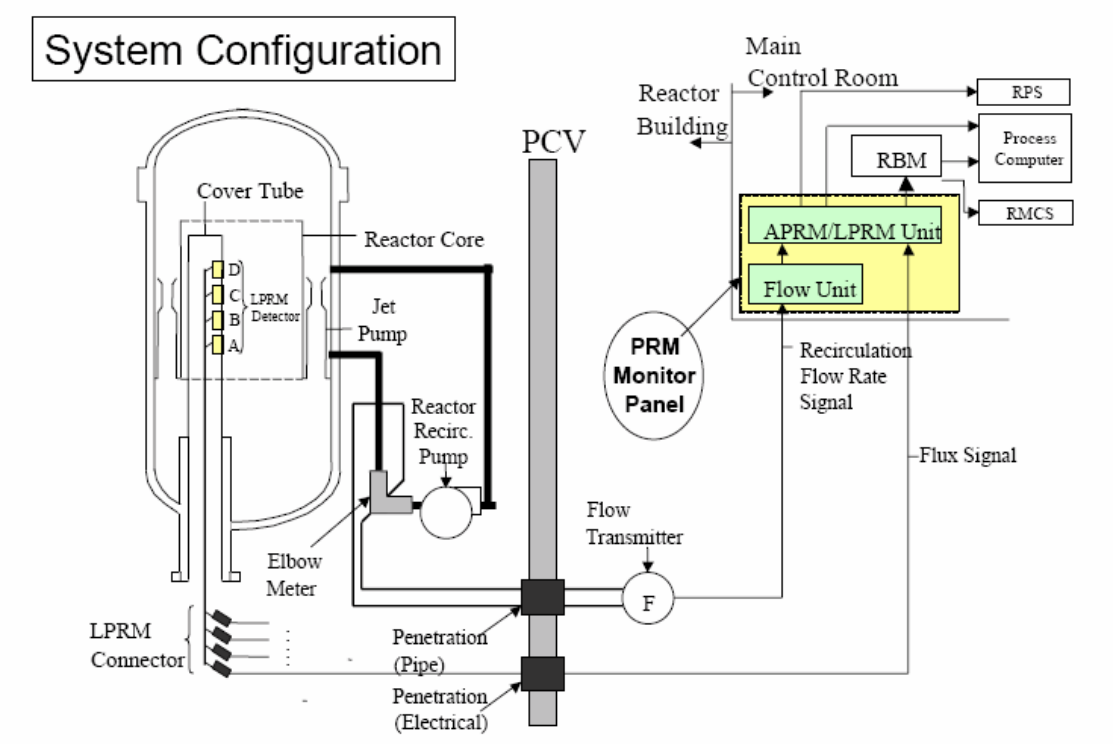

Figure 58 - Power Range Neutron Monitoring System overview

Source: Reference [38]

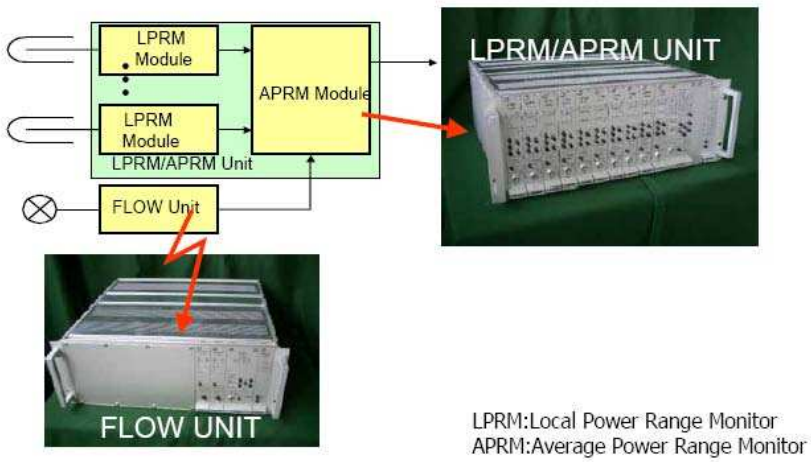

Figure 59 – Power Range Neutron Monitoring System hardware

Source: Reference [38]

Some key drivers for using FPGA technology in this project are:

- Non-volatile, non-rewriteable programming of the FPGA (anti-fuse technology), which prevents loss of configuration or unplanned configuration changes in service.
- High speed operation as compared to CPU-based systems.
- Longer product life cycle and longer-term support of the product, especially when compared to support offered by microprocessor vendors.

- Improved testability and verification at lower cost compared to CPU-based applications (no operating system used).
- Reduced drift as compared to analog equipment.
- FPGAs can be programmed after shipment from semiconductor foundries, making them suitable for low-volume applications such as nuclear power plants, as compared to ASICs.
- Compatibility with conventional analog-based systems, minimizing cost and implementation time for upgrades.

In general, Toshiba considers that FPGAs are well suited to applications for which the functionality is relatively simple and does not change frequently. Such functionality can include simple Boolean logic, process control algorithms (e.g., proportional-integral-derivative control), and human-machine interfaces that are relatively simple and stable (e.g., equipment front panel HMI for calibration and setpoint entry). A more complex, full-featured HMI (e.g., an information display system with windowing, or a plant computer system) is most likely not suited to FPGA implementation, and is thus better suited to a microprocessor or other CPU-based application.

For safety-related applications such as the PRNM, Toshiba has chosen 300 nm technology non-rewriteable FPGAs from Actel (antifuse type A54SX72A (72K gates) and A54SX32A (32K gates)). They are not radiation-hardened, as will be installed in a mild environment (main control room).

In the LPRM and APRM modules, FPGAs perform functions including digital filters, comparators, calculations, bistables, input interfaces, front panel HMI functions, and diagnostic functions. The Actel FPGAs that are used can process only digital logic, so all analog signal processing and analog-to-digital conversion is done in separate devices, so it is not a SoC implementation.

The Toshiba design approach is based on the concept of the “functional element” (FE), defined as the minimum logical unit in an FPGA application, built from FPGA cells. Toshiba recognized that 100% exhaustive testing of an entire application for a nuclear power plant is not practical. Each FE’s logic function is designed to be sufficiently simple so that it can be verified through exhaustive, full pattern testing. Each FPGA is programmed using combinations of verified FEs.

FEs implement common functions such as addition, multiplication, comparison, and data communication. The process for development of the FEs includes the following steps:

- Define and document the FE requirements.
- Prepare a detailed design document, or FE specification.
- Prepare an FE test procedure.
- Generate an FE requirements traceability matrix (RTM).
- Write the HDL source code (in VHDL o Verilog).

- Verify that the FE works correctly using simulation.
- Map the FE to the FPGA and program the FPGA.
- Perform FE validation.

Toshiba has developed a library of validated FEs that can be applied to multiple applications.

The programming process followed by Toshiba for implementing FPGA logic includes the following steps:

- Design: define the functional and detailed design for each FPGA on each module.
- Implementation: directly write HDL code that implements the required logic, using VHDL or Verilog. A diagram is then generated from the HDL and verified against the application requirements.
- Logic synthesis: using a synthesizer, transform the HDL code into a netlist.
- Place&route: map the logical structures of the netlist onto macrocells, interconnections, and input/output pins, retaining the structure of the logic and producing a fuse map.
- Embed: use a device programmer to embed the logic into the FPGA chip.

An important aspect of Toshibas's process is that, when the logic is compiled (synthesis and place&route), the tools used are not allowed to optimize the connections. This helps in the V&V process as all FEs has been fully tested individually.

Another important aspect is that the system doesn't incorporate any internal diversity, based on its simplicity and the almost full testing during V&V process.

Some of the tools used by Toshiba include the following:

- Text editor to develop and edit HDL source code from Actel.
- Synplify® tool (by Synplicity®) to synthesize logic from the HDL of the main application code and for each FE used.
- Tools provided with the Actel integrated development environment (IDE):
    - NetList Viewer.
    - Place & route tool.
    - Static timing analyser.
    - Package layout tool.
    - Power consumption analysis tool.
- ModelSim®, which is a simulation software for HDL (by Mentor Graphics).
- Silicon Sculptor II, which is the Actel's device programmer (by BP Microsystems).

- Pinport, which is an interface between digital hardware and the ModelSim simulation environment (running on a PC), used to test a new FPGA hardware prototype by use of vector patterns (by SynaptiCAD Sales).

Although no IP-core is employed, Toshiba uses a software life-cycle approach for FPGA development, so some of the standards followed includes IEEE Std. 603 (endorsed by 10CFR50.55a), IEEE Std. 7-4.3.2 (endorsed by previous versions of RG 1.152), IEEE Std. 1012 (endorsed by RG 1.168) and EPRI TR-107330 (Generic requirements specification for qualifying a commercially available PLC for safety related applications in NPPs).

Traditional qualification, including environmental (vibration, temperature and humidity), seismic, EMC, surge, EFT, ESD and isolation tests, are performed following corresponding regulation.

The use of FPGA chips that are widely used in other industries (e.g., military, aerospace, aircraft) increases the likelihood that the vendor will continue to support them in the long term. The current chip vendor made a commitment to the US Department of Defense (DoD) to support the chips for 30 years. They are now 15 years into this, so it is expected that there will be at least 15 more years of support. The strict use of standard HDL programming should allow a level of portability to other chips as well, if necessary.

Regarding licensing, TEPCO worked with the Japanese regulator to gain approval of the PRNM installation for an Advanced Boiling Water Reactor plant. At that time, FPGA-based systems were viewed by the regulator primarily as hardware-based systems. Previous experience and testability of the system were key factors in gaining approval.

In Japan, the use of software development and V&V processes similar to those called out in IEEE 7-4.3.2 was considered supplementary to the experience and testing of the hardware. However, Toshiba is providing this system for the new South Texas Project Units 3&4, and offers it in general for US BWRs, so approval by the US NRC will be required. There, use of a software development type process will be more central to gaining licensing approval.

Toshiba initially sought approval for both the platform and the PRNM application through submittal of topical reports to the NRC. They submitted a generic topical report for review and approval of the platform in March of 2008. This report describes Toshiba's quality assurance program, and design, development, review, test, qualification and manufacturing processes for FPGA-based safety systems. It also describes how the generic qualification process is implemented for a specific system/application. The NRC's Office of Nuclear Reactor Regulation (NRR) had the lead on this review.

In October of 2008 Toshiba completed the PRNM system qualification testing and submitted to NRC a corresponding system topical report. This report describes Toshiba's application of the generic topical report methodology in conjunction with NRC guidance for qualification of a specific FPGA-based system – the PRNM system. It documents the system specification, results of FPGA logic development and V&V, and hardware qualification activities. It also documents the EPRI TR-107330 Requirements Compliance and Traceability Matrix (RCTM),

and the application guide prepared by Toshiba. EPRI reports based on the generic and system topical reports were then issued by Toshiba in April 2009 in accordance with EPRI guidance.

In about this same timeframe, the NRC's Office of New Reactors (NRO) was beginning the review of Toshiba's FPGA-based system designs for the new builds at South Texas Project (STP) Units 3 & 4. As a result, Toshiba asked NRC to stop review of the topical reports, at least until the STP application has completed its review by NRO.

Last information available from NRC is that Toshiba has gone on with licensing its FPGA-based platform [39] [40].

There is no specific requirement in Japan for diversity to address CCFs, and the PRNM system doesn't include any internal diversity. External diversity can be provided to address any residual CCF concerns, consistent with the overall D&DiD strategy chosen for the plant. For South Texas Project Units 3 & 4, the Advanced Boiling Water Reactor currently includes specific measures and equipment that provide the required D&DiD for the safety equipment, external to those systems.

THIS PAGE INTENTIONALLY LEFT BLANK.

Appendix 6

Modernization of Engineered Safety Features Actuation System in Kozloduy Units 5&6 nuclear power plant in Bulgaria

Research and Production Corporation (RPC, also known as Radiy) is a manufacturer and supplier of instrumentation and control systems for nuclear power plants in Eastern Europe. Its I&C systems are licensed for safety applications in Ukraine and Bulgaria, and also certified in Russia.

First commissioning of Radiy's current FPGA-based platform in a nuclear power plant took place in 2003. In six years since that time, 46 applications of Radiy's systems have been installed in 17 VVER (Russian PWR) nuclear reactors in Ukraine and Bulgaria.

The Radiy FPGA-based platform has been applied to Reactor Trip Systems, Engineered Safety Features Actuation System, Reactor Power Control and Limitation System and Rod Control System.

Radiy has provided solutions that incorporate diversity to protect against common cause failure, like for the RTS in Ukrainian plants. The diversity consists of two separate and diverse sets of equipment, each capable of implementing the RTS functions, using FPGAs from different vendors and being different technologies, providing separate design teams who employed different software languages and different toolsets. Each set can be taken out of service for testing or surveillance, while the other continues to provide full RTS functionality, like with the typical old-fashion discrete-type RTS.

The Radiy platform architecture has two levels, with the lower one being safety (category A per IEC 61226), and comprised of a number of cabinets, each containing a set of standard modules that perform the primary safety functions. These modules include processing of input signals from sensors or manual actions, performing voting logic or control algorithms to command output signals and driving signals to the final elements or actuators. They also incorporate diagnostic modules to provide both locally and remote information.

Figure 63 represents a simple ESFAS implementation with Radiy platform. Input signal processing is performed in Normalizing Converters Cabinets (NCC), which send the processed signals to the Signal Forming Cabinets (SFC). Remote Control Cabinets (RCC) control the actuators based on the outputs from the automatic control logic or remote control commands from the main control room. Cross Output Cabinets (COC) control separate, non-ESFAS actuators. A Signalling Cabinet (SC) provides annunciation signals to the control room. The RCCs, COC and SC all receive signals from the three (in this case, but it depends on the particular reactor) control channels (from the SFCs) and perform the two-out-of-three majority

voting logic. In the case of Kozloduy NPP, complete ESFAS is comprised of about 60 cabinets in total. Kozloduy ESFAS is not redundant.

The upper level is a computer cabinet that contains industrial workstations. Software running on these workstations performs functions such as receiving and displaying the process and diagnostic information coming from the lower-level cabinets, providing the HSI for maintenance and engineering people, and archiving of the process and system diagnostic information. The functions performed by the upper level computers do not impact the safety function and are classified as categories B and C per IEC 61226.

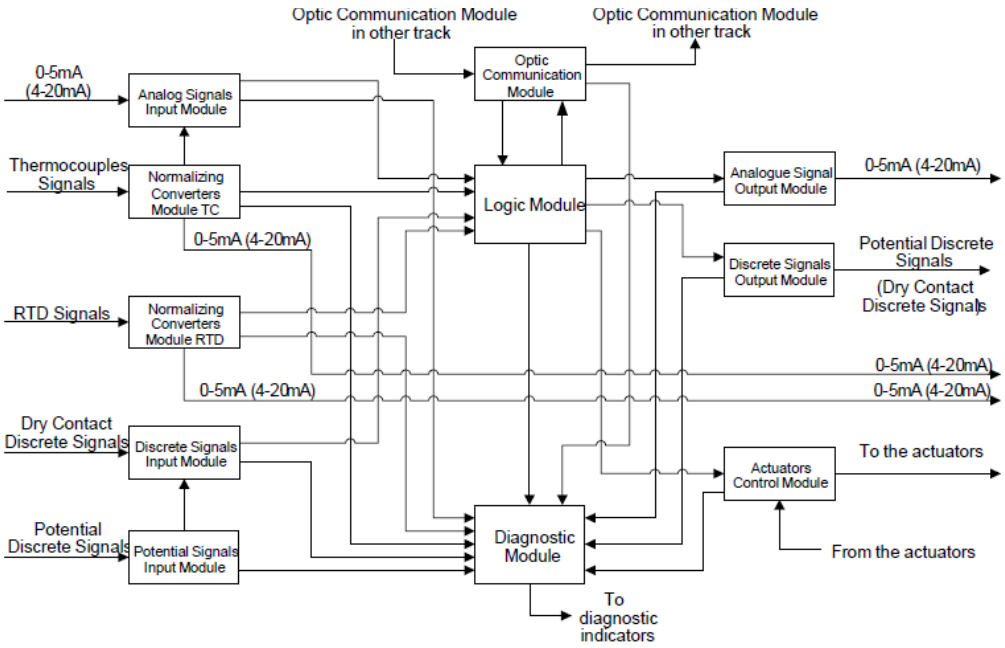

Figure 60 – Radiy platform simple architecture representation

Source: Reference [41]

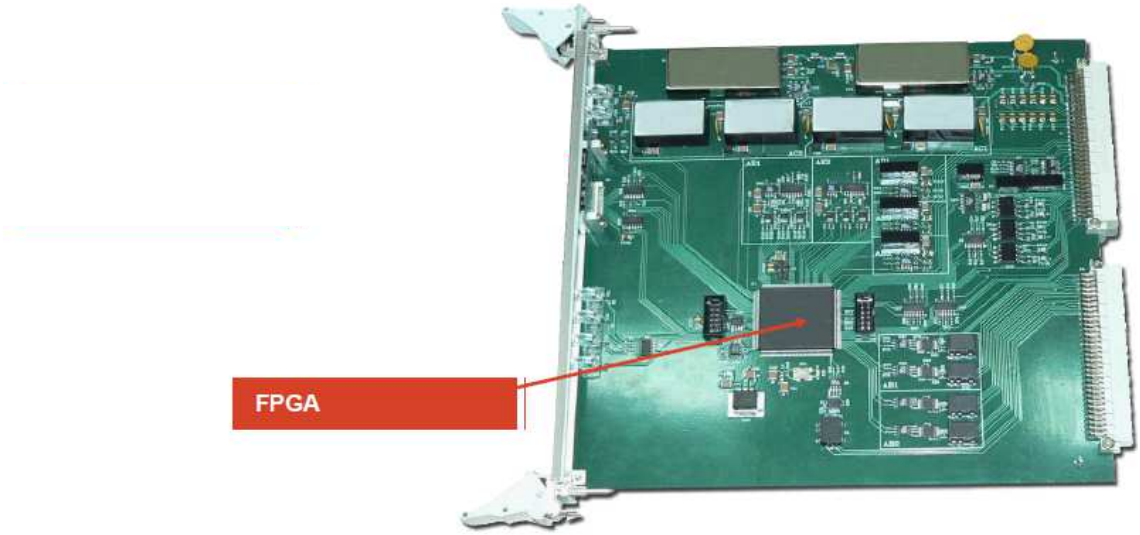

Figure 61 – Example of Radiy module

Source: Reference [41]

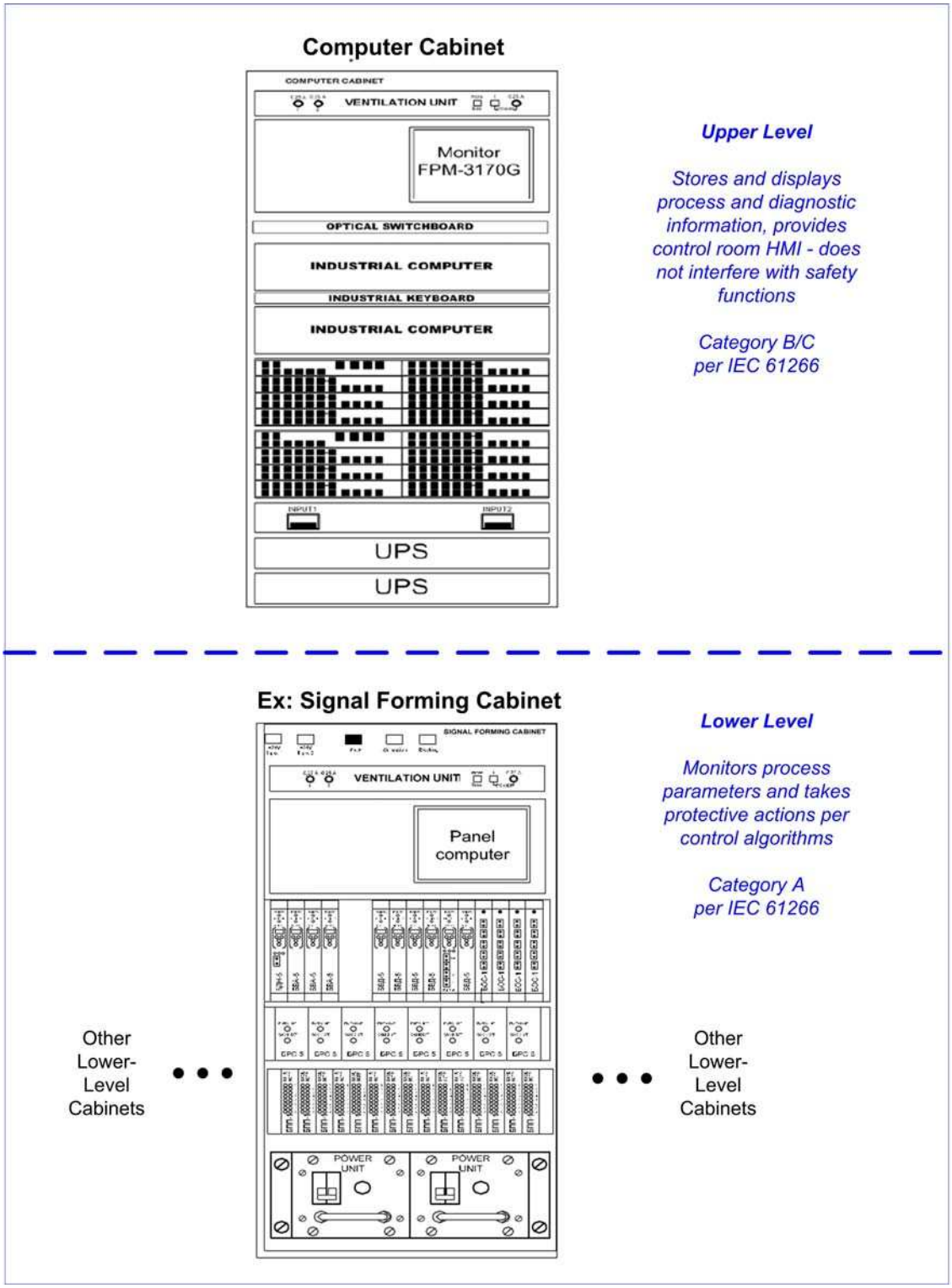

Figure 62 – Radiy's platform cabinet layout

Source: Reference [42]

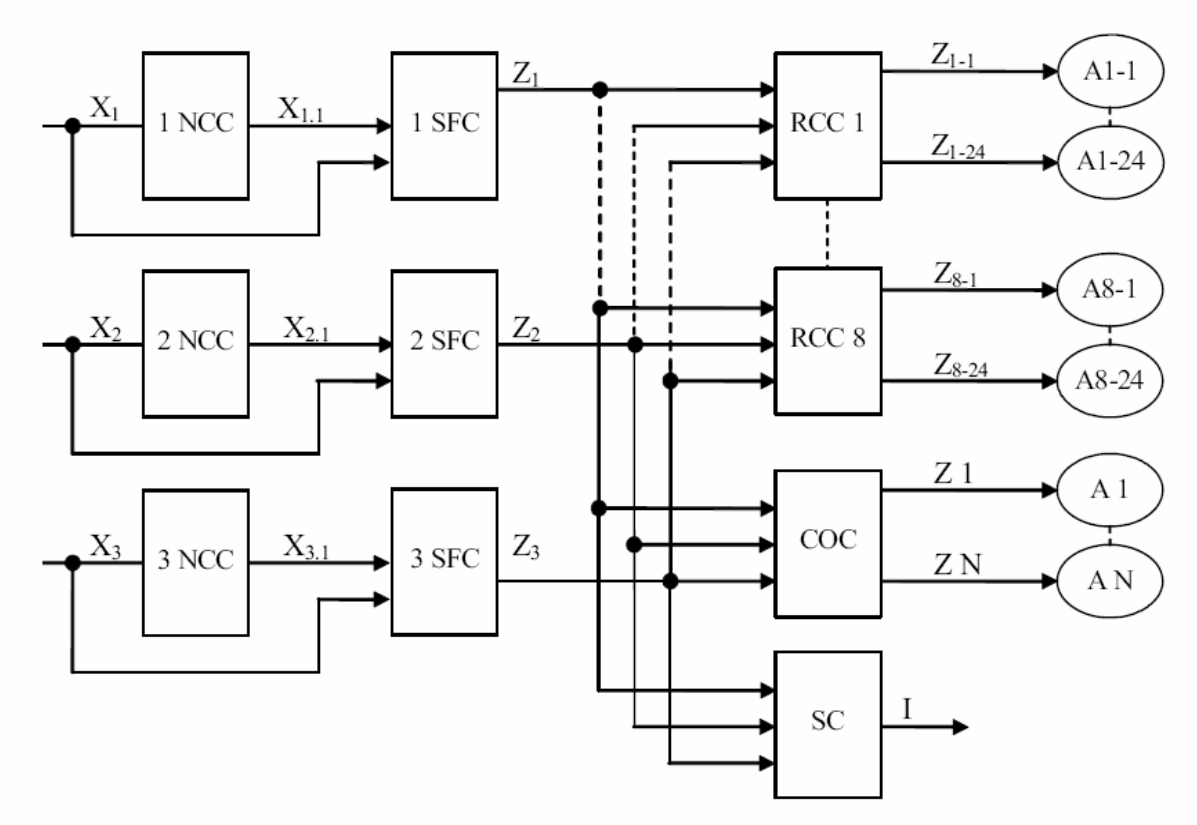

Figure 63 – Simple ESFAS implementation with Radiy

Source: Reference [42]

Around 13,000 FPGA integrated circuits have been in operation in Radiy's systems for a combined operating time of at least 20,000 years, with no serious problems so far and no common cause failures.

With 17 VVER plants now using Radiy's platform for various safety system applications, the company is turning its attention toward applying the platform to other types of reactors. In the US, Radiy has initiated preliminary discussion with NRC regarding review and approval of the platform. The Company expects to submit a topical report to NRC.

In 2008, a major modernization program was put in place for Engineered Safety Features Actuation System in Kozloduy NPP, Units 5 and 6, in Bulgaria. These reactors are VVER-1000. This project represents the largest nuclear plant I&C modernization in Europe to use FPGA-based equipment.

The first ESFAS replacement was completed in September 2008, when one of the three redundant ESFAS systems for Kozloduy Unit 6 was replaced with the Radiy platform. The installation was accomplished during a refuelling outage with duration of less than one month. The other two redundancies were replaced in September and in October of 2009 for Kozloduy Unit 6, with a replacement time of only 15 days. Unit 5 installation has also been completed in the fall of 2010.

Figure 64 – Engineered Safety Features Actuation System of Kozloduy NPP

Source: Reference [42]

Kozloduy ESFAS system performs the following functions:

- Automatic actuation of safeguards equipment when parameters exceed pre-defined limits.
- Automatic control of actuators in accordance with process control algorithms.
- Remote control of actuators based on manual actions from the control room.
- Transmission of discrete signals to other systems.
- Diagnostic, information and servicing functions.

Different levels of software are used within the platform:

- Flat hardware logic in the FPGAs, implemented as part of the FPGA electronic design, using configurable logic blocks and interconnections, with no microprocessors or run-time software, used to implement all safety functions and related communications.
- Microprocessor emulators in the FPGAs to perform auxiliary functions such as diagnostics, data reception and transmission for non-safety use, without any operating system.
- Panel Computer, a PC with a standard operating system and software that implements relatively simple front panel HSI functions such as status and data display and for entering setpoint values into the channel.

- Upper-level industrial workstations running conventional Windows® and providing the control room HSI, information display and archiving.

Data communication between the setpoints processor and Panel Computer is one-way (from setpoints processor to Panel Computer) during operation. For writing setpoint values into the setpoints processor, the ESFAS must be put out of service. Data communications between the diagnostic processor and diagnostics module is also one-way (from diagnostic processor to diagnostics module).

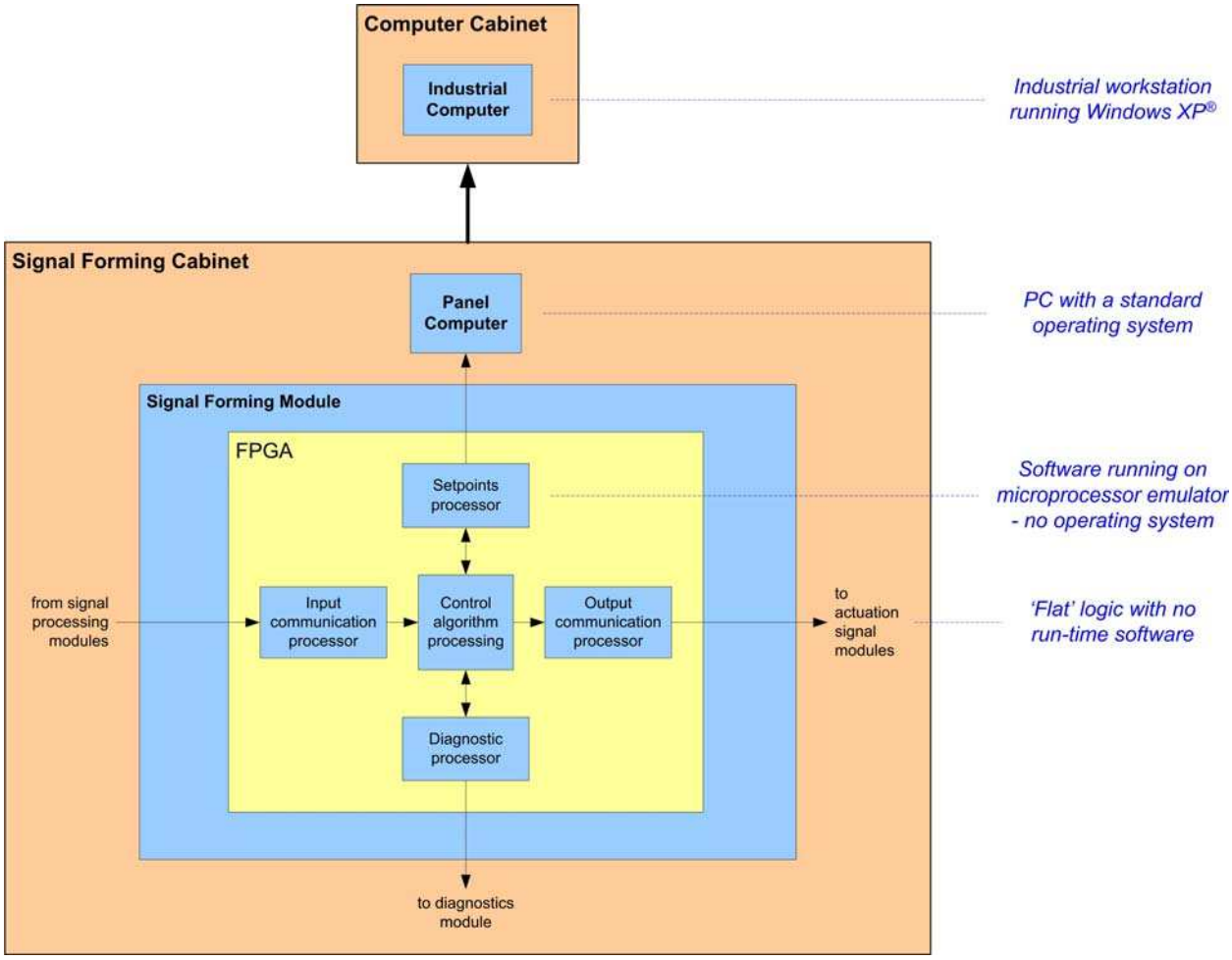

Figure 65 – Digital system architecture with different levels of software involvement (example representing Signal Forming Cabinet)

Source: Reference [42]

Radiy considers that FPGAs have a number of advantages for safety system applications, including:

- The development and verification processes are simplified as compared to what is required for microprocessor-based systems.
- Control and other safety-critical functions can be implemented in the form of flat hardware logic without requiring any run-time software.

- Parallelism inherent in control algorithm execution when the algorithms are implemented in flat hardware logic. This leads to determinism in terms of latency.
- Lower energy consumption and heat load.
- Ability to implement greater levels of redundancy and diversity in a practical way.
- Ability to reconfigure the logic more easily than with other hardware solutions.
- Improved diagnostics are possible as compared to other hardware solutions.

For safety-related applications such as the ESFAS, Radiy has chosen SRAM-technology in the primary channels (Altera Cyclone™). The diverse channels are built using flash technology from a separate vendor (Actel ProASIC3™). Both the SRAM and flash technology are rewriteable, providing advantages during development and modification. Precautions are taken to address the susceptibility of SRAM and flash ICs to single event upsets and to protect against cybersecurity threats. This strategy provides high level of diversity, as it introduces diversity of the hardware ICs, the technology (flash versus SRAM), and the associated toolsets.

The FPGAs used by Radiy are not radiation-hardened, as the equipment typically is installed in a mild environment. However, radiation exposure withstand tests have been performed for the ESFAS equipment.

In general, Radiy considers that SRAM and flash FPGAs are well-suited to I&C applications when appropriate defense barriers are implemented to prevent inadvertent or unauthorized modifications. Radiy has seen limitations in some of the toolsets for anti-fuse FPGAs, and has noted past issues regarding long-term reliability of anti-fuse connections. Also, Radiy has not found the anti-fuse chips to have a sufficient number of logic cells to implement the extensive diagnostics and self-testing included in their platform.

FPGAs are treated as programmable components and Radiy uses a software life-cycle approach to develop both the logic and the code to program the FPGA. Three types of design development and representation approaches are used for the FPGA designs:

- Graphical functional diagrams developed in a computer-aided design (CAD) environment, which includes the use of standard libraries plus additional, custom-developed libraries.
- Software model of the hardware design using a hardware description language (HDL).
- Software code developed to run in a microprocessor emulator environment, where the emulator is implemented in the FPGA as a separate functional core.

The process followed by Radiy for implementing FPGA logic is as follows:

- Develop functional block diagrams for the control algorithms: the block diagrams are developed in a CAD environment and are based on the system requirements specification, the distribution of functions between hardware and software, and other pertinent application requirements. For complicated applications, the algorithms are

broken down to functional modules, with series and/or parallel tiers of the modules to implement the needed functionality. The diagrams are developed in a form that is as close as possible to the presentation scheme used in the particular FPGA integrated development environment. For some functions such as mathematical calculations, where it is not practical or useful to develop block diagrams, HDL is developed directly without diagrams.

- Develop HDL program models: HDL representations of the control algorithms are developed and simulated within the IDE. VHDL language is used.
- Integrate the HDL program models: the HDL models are integrated using the IDE. This includes integration of IP cores with the developed models.
- Implement the integrated program in the FPGA: the integrated program model is implemented in the FPGA using a PC connected to the chip JTAG interface.

As Radiy uses a software life cycle approach to develop FPGA-based systems, the following standards are used, but adapting them for the peculiarities of FPGAs:

- IAEA NS-G-1.1. Software for computer based systems important to safety in nuclear power plants.
- IAEA NS-G-1.2. Safety assessment and verification for nuclear power plants.
- IAEA NS-G-1.3. Instrumentation and control systems important to safety in nuclear power plants.
- IEC 60880. Nuclear power plants – instrumentation and control systems important to safety – Software aspects for computer based systems performing category A functions.
- IEC 61513. Nuclear power plants – instrumentation and control for systems important to safety – general requirements for systems.
- ISO/IEC 12207. Information technology – Software life cycle processes.
- ISO/IEC 9126-1. Software engineering –Product quality – Part 1: Quality Model.
- IEC 61226. Nuclear power plants – Instrumentation and control systems important to safety – Classification of instrumentation and control functions.
- IEC 62138. Nuclear power plants – Instrumentation and control for systems important to safety – Software aspects for computer-based systems performing category B or C functions.
- IEC 62340. Nuclear power plants – Instrumentation and control systems important to safety – Requirements for coping with common cause failure (CCF).
- IEEE 730. Software quality assurance plans.
- IEEE 828. Software configuration management plans.

- IEEE 1012. Software verification and validation.
- IEEE 1028. Software reviews and audits.
- IEEE 1074. Developing a software project life cycle process.
- IEEE 1228. Software safety plans.

Radiy has used IEEE standards in software engineering because, first, these standards are very close to the IEC and IAEA standards, and second, the IEEE standards give a more detailed description of life cycle processes than do the IEC and IAEA standards.

The approach specified in IEC 60880 was used for acceptance of the tools as suitable for use in Class A systems. This includes evaluation of the quality system of the tool developer and assessment of the quality of libraries and components provided with the toolset.

Design and development of components is performed in RPC Radiy. Verification and validation is performed by an independent organization, the Safety Infrastructure-Oriented Research and Analysis Center. This center provides managerial, financial, and technical independence from Radiy design organizations.

The Radiy platform had previously gained regulatory approval in Ukraine, based on Ukrainian regulatory document NP 306.5.02/3.035-2000 “Nuclear and Radiation Safety Requirements to Instrumentation and Control Systems Important to Nuclear Power Plant Safety.” That document is harmonized to the requirements of IAEA and IEC standards.

In Bulgaria, high-level requirements of Bulgarian nuclear regulatory agency were used as the basis for licensing the system for Kozloduy. Since the high-level regulatory documents are also harmonized to IAEA and IEC standards, these were again used as the basis for gaining approval of the platform for Kozloduy. The licensing approach was to consider the FPGA chip as hardware and the IDE as software.

Qualification testing included environmental, seismic, EMC, surge, EFT, ESD and isolation tests.

Additionally Radiy is certified according to ISO 9001 and its Quality Management System comply with 10CFR50 Ap. B and ASME NQA-1 requirements. Radiy is also an approved supplier of AECL since October 2010.

THIS PAGE INTENTIONALLY LEFT BLANK.

Appendix 7

Rolls Royce Civil Nuclear experience on the development of field programmable gate array solutions

Rolls-Royce Company has provided systems for military and civil sector in United Kingdom (UK) and abroad for many years, being a subcontractor of the Royal Navy and the Royal Air Force. In the civil sector, Rolls Royce Civil Nuclear (RRCN) has provided systems and equipment to the nuclear civil sector, being very active in the English and French industry. In the case of PLDs for nuclear reactor control and protection, RRCN has over fifteen years of experience.

Rolls-Royce started PLD application to nuclear business in the early '90s. Working with Independent Nuclear Safety Assessor (INSA), Rolls-Royce received endorsement of the process by INSA in 2000. The process was then applied to the design of a complete reactor control and instrumentation suite.

Rolls-Royce acts as the main supplier and has a contract with the real PLD supplier, Ultra Electronics Ltd. Rolls-Royce developed the overall design and verification process and mandated its use by Ultra Electronics.

Portability is one of the key features of the Rolls-Royce designs. One of the primary drivers for using FPGAs versus other technologies was protection against obsolescence. Additionally, due to the regulatory environment in the UK, for safety-critical systems (category A functions per IEC 61226) the company avoid the use of microprocessors and associated run-time software.

RRCN designs for safety-critical applications are deterministic and implemented in flat hardware logic. The designs are synchronous and based on a fixed operating cycle, and make use of self-testing by an independent on-line test system.

When Rolls-Royce began their development of PLD-based solutions, CPLDs were found to be dependent on vendor-supplied tools for design implementation. On the other side, anti-fuse FPGAs were considered to be too expensive. Because of this, RRCN decided to use SRAM FPGAs for pilot projects, and this same technology was retained for the first real designs that became commercially available.

As manufacturers developed smaller feature sizes, concerns over susceptibility of volatile configurations to background radiation and associated SEUs started becoming a problem, so finally RRCN moved to flash-based FPGAs. Actel ProASIC-Plus^{TM} family is currently used. The company has also some experience in using anti-fuse technology for certain applications.

The design requirements are partitioned to develop specifications for the analog and the digital portions of the design. Further partitioning results in digital specifications for each module.

VHDL coding guidelines developed by RRCN and its subcontractor, Ultra Electronics, are used to minimize problems with the RTL design and design verification. This includes synchronous design using a single clock, not using technology-specific constructs or directives, not using IP-cores in safety systems, types of port declarations to use or avoid, etc.

RRCN exhaustively test the functionality contained in every module and relies on a Code Coverage Analysis Tool as an objective measurement of the exhaustiveness of RTL simulation, checking both the code and the test bench test vectors for completeness. The following guidelines are applied:

- 100% code coverage should be sought in each of the following categories:
    - Statements: every executable statement is examined for execution counts and those not having at least one execution are flagged.
    - Branches: possible outcomes of IF and CASE blocks are examined for execution counts with those not having at least one execution are flagged.
    - Conditions: any sub-conditions of a code branch (e.g., IF, AND, CASE, WHEN statements) are examined as to fulfilment and those not having at least one true outcome are flagged.
    - Paths: any un-traversed path is flagged.
    - Triggering: checks whether a single signal or a combination of signals is responsible for trigger execution, and any candidate signal that has not had their own individual opportunity to trigger the process or statement is flagged.
- The code should be successively refined until the goal of 100% coverage of the above categories is met.

The complete RTL design code is passed through a second, “shadow” synthesis tool. This tool is chosen from a different vendor’s toolset and is targeted to a different technology, so that the shadow synthesis tool is independent of the primary synthesis tool. The two independently derived solutions are then exhaustively compared using a Formal Equivalence Checking (FEC) tool that is independent from both synthesis tools, confirming the complete functional equivalence of the two implementations. This provides confidence that the RTL definition is complete and the synthesis tools have not inserted default logic or eliminates redundancies.

Nevertheless, synthesis tools depend on data from timing libraries, and errors or inaccuracies in those libraries could produce erroneous results from timing analysis. Estimates from synthesis tools are only an approximate guide and cannot predict signal routing delays with precision. Use of hardware-in-the-loop test is a good alternative to use to alleviate this issue.

Nevertheless, FEC has some limitations. There are two main types of FECs:

- Sequential equivalence checker: this one determines the potential states of a design and proves that their sequence is the same in the compared design representations for

all input sequences. These checkers require large amount of computing capacity to analyse large designs because the number of states is 2^n, where n is the number of flip-flops.

- Structural equivalence checkers: this one identifies the storage elements and examines the logic between them for Boolean equivalence. These checkers require a complete mapping of each storage element to the exact counterpart design, and the ability to do this can be compromised by the optimizations performed by synthesis tools.

RRCN requires that the supplier, Ultra Electronics, ensures independence between the design and verification activities, using two different teams for these activities. In addition, RRCN acts as a third independent reviewer.

Some key attributes in the process are:

- Partitioning of the design into modules that can be coded in HDL and tested in the simulation environment with full coverage.
- Independent verification and validation to ensure that the design code matches the specifications.
- Formal equivalence checking to ensure that bugs in the implementation tools are detected, being the synthesis tools and FEC tools diverse.
- Traditional testing of a physical prototype is still performed and has the same importance as in the traditional design process. This introduces the concept of “hardware-in-the-loop” testing, in which the stimulus from the VHDL test bench, presently used for RTL verification testing, can be applied in real time to the actual programmed FPGA, hosted on a card plugged into a development workstation. The real-time electrical responses are fed back and automatically compared with the simulation responses. Additionally, use of random test vectors to test for the absence of unwanted behaviours can be employed.

Another important issue is the device package. Industry has moved from hermetically sealed components, such as ceramic pin grid array, to plastic surface mount technologies. RRCN avoids the use of ball-grid array packaged devices because of the difficulty of inspecting the solder joints. Quad flat pack is the only acceptable choice among the packaging options currently offered by its FPGA vendor.

THIS PAGE INTENTIONALLY LEFT BLANK.

Appendix 8

Modernization of slave logic units of Rod Control System in Electricité de France 900 MW series reactors in France

EDF owns and operates a fleet of 58 nuclear power units in France. These units are designed and standardized into three series:

- The 900 Megawatt series (34 units).
- The 1,300 Megawatt series (20 units).
- The N4 series (4 units).

The French regulation requires that every decade, each unit is thoroughly inspected and verified. (ten-yearly inspection). Since the corresponding outage is significantly longer than the other, normal outages, EDF generally uses this opportunity to also perform significant system upgrades. In particular, in the framework of the 3rd ten-yearly outage of the 900 MW series, it has decided to replace the aged and obsolete analog I&C of the Rod Control System (RCS) of the series (RGL system). The replacement consists of a number of obsolete modules with electronic circuit boards based on up-to-date technologies. One of these circuit boards, the slave logic unit, contains an FPGA.

One of the required functions consists of generating the sequencing signals for the three coils that control the position of the control rods. This function has strict timing requirements (1 millisecond). As this timing requirement is very difficult to satisfy with microprocessor-based designs, this is the most important argument for the use of FPGA in this modernization project. Additionally, this system is not safety classified.

The upgrade is implemented by Rolls Royce Civil Nuclear (formerly Data Systems&Solutions, or DS&S). As a designer and manufacturer of nuclear safety I&C platforms and a longtime supplier of safety systems (in particular, RRCN has provided EDF with all the digital reactor protection I&C systems in use in the 900 MW, 1,300 MW and N4 series, or SPINLINE^{TM} series), RRCN has a long and extensive experience in electronic design and use of programmable electronic devices.

Project began in 2005 with the issue of the user requirement specification, first commissioning started in 2009, and expected date for end of deployment in the reactor series is 2019.

FPGA employed is Actel A3P1000 (1M gates, 130 nm technology, 350 MHz), which is flash-based technology.

The FPGA is interfaced to the power units generating the currents necessary to energize the coils, to a local conventional HSI, and to a PLC-based control and diagnostic unit (based on the Quantum™ platform from Schneider Electric, in a redundant architecture).

The main functions performed by the FPGA are:

- Generation of the control signals for the three coils that control the movement of the rods in order to perform a one-step lift or down, or hold the rods in their current positions (binary outputs).
- Check that the requested movements have actually been performed, in particular through checking coil currents (analog inputs).
- Interface with the local HSI which can be used to manually control the rod positions (binary I/Os).
- Interface with the PLC-based control and diagnostic unit (binary I/Os and serial Modbus communication).

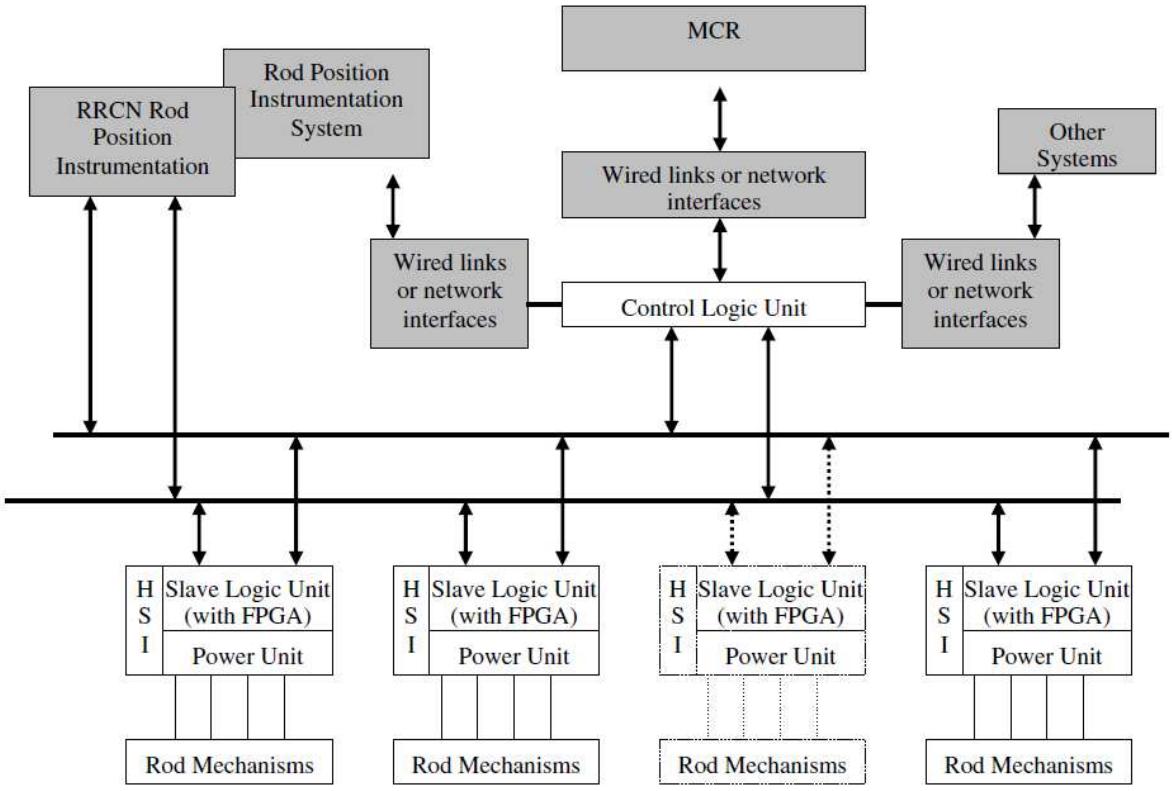

Figure 66 – Rod Control System architecture

Source: Reference [10]

RRCN applied the same FPGA design process it has used in previous projects, starting with the traceability between the overall user requirements and the requirements specification of the Slave Logic Unit (SLU), and then between the SLU requirements and the FPGA requirements.

The FPGA requirements are specified in a dedicated Requirements Document in natural language and logic schematics. In addition, some algorithms are specified in pseudo-code.

Language used is VHDL. Design is synchronous. Simulation tools included ModelSim™ from Mentor Graphics at the RTL level and place&route.

Only internal RRCN standards were used. This was possible as Rod Control System is not a safety system.

Upon failures the outputs of the SLU are put into a predetermine state. This can lead to a hold state (using double gripper scheme) or to dropping the group of rods, depending on the type of failure. It should be taken into account that system safe state is rods dropped, or reactor trip.

No other fault tolerance measures were implemented in the FPGA for availability (no redundancy was implemented at the FPGA level). Such measures were taken at the system level.