7th International Conference on Modelling and Simulation in Nuclear Science and Engineering (7ICMSNSE)
Ottawa Marriott Hotel, Ottawa, Ontario, Canada, October 18-21, 2015

7ICMSNSE-96

Modelling of Field Programmable Gate Array Based Nuclear Power Plant Safety Systems Part I: Failure Mode and Effects Analysis

P. McNelles[1], Z. C. Zeng[1], G. Renganathan[1]
[1] Canadian Nuclear Safety Commission, Ottawa, Ontario, Canada
Phillip.Mcnelles@cnsc-ccsn.gc.ca, Zhaochang.Zeng@cnsc-ccsn.gc.ca,
Guna.Renganathan@cnsc-ccsn.gc.ca,

Abstract

Field Programmable Gate Arrays (FPGAs) are programmable hardware that can be used to perform instrumentation and control functions. The potential use of FPGAs in Nuclear Power Plant safety systems requires that FPGA-based systems must be functionally safe and reliable. To accomplish this, a Failure Mode and Effects Analysis (FMEA) was performed, to uncover the potential failure modes, their causes and their effects. In addition, methods to avoid the failure modes, or mitigate and/or control their effects were recommended. Furthermore, these analysis results were used as the guidelines for review for FPGA-based safety systems. This paper discusses the details of the analysis performed and the results of the study including providing recommendations for future regulatory reviews of FPGA-based safety systems.

Keywords: FPGA, FMEA, Instrumentation and Control.

1. Introduction

The design and construction of Nuclear Power Plants (NPPs) considered the technology that was available at the time. NPPs that were designed and constructed outside of the last 10 years did not incorporate Field Programmable Gate Array (FPGA) technology into the Instrumentation and Control (I&C) systems. For example, the NPPs in Canada were constructed from 1971-1992, however at that time FPGA technology was not well developed, or did not gain momentum to get introduced into those I&C systems. In more recent years, FPGAs have garnered interest worldwide for NPP I&C system applications. New NPP designs have introduced FPGAs, and as part of ageing management and refurbishment, and it is expected that the FPGAs will be introduced into the I&C systems of the operating plants. FPGA implementations have taken place in the VVER reactors in Ukraine, Boiling Water Reactors (BWR) and Advanced Boiling Water Reactors (ABWR) in Japan, Pressurized Water Reactors (PWR) in France, and platforms have been developed to work with the AP1000 reactor. Additionally, during the refurbishment of the Embalse plant in Argentina, FPGA components were introduced into I&C systems [1,2].

Before being introduced into NPP safety systems, the FPGA-based systems are expected to comply with fundamental safety objectives and principles, and the safety and reliability of the FPGA-based systems must be established. The publically available regulatory guide that is specific to FPGA-based safety systems was published by the United States Nuclear Regulatory Commission (USNRC), known

as NUREG/CR-7006 "*Review Guidelines for Field-Programmable Gate Arrays in Nuclear Power Plant Safety Systems*" [3]. This document is largely based on the report from Oak Ridge National Laboratory (ORNL), known as ORNL/TM-2009/20 "*Review Guidelines for FPGAs in NPP Safety Systems*" [4]. However, neither of those documents established specific regulatory requirements for FPGA-based systems. CNSC staff considers the development of regulatory guidelines regarding FPGA-based systems to be important and essential.

The introduction of FPGA-based systems into nuclear power plant I&C systems has potential benefits, but also introduces the potential of new failure modes that may affect the operation and margin of safety. These potential new failure modes must be identified, and method(s) to avoid or mitigate them must be established, before FPGAs can be noted to be useable in Canadian NPPs. In an attempt to develop and establish the regulatory requirements for the FPGAs, a Failure Modes Effect and Analysis (FMEA) was performed on FPGAs, using internationally available literature and activities performed on this topic. This paper discusses the results of the work and the recommendations provided for the FPGA design, and the regulatory aspects that needs to be considered while integrating FPGAs into the NPP I&C systems.

The outline of this paper is as follows. Section 2 discusses background information about FPGA technology and programming, and Section 3 discusses the FMEA and the purpose for performing one on FPGAs. Section 4 details the results from the FMEA (identified failures modes, causes, effects and mitigation techniques) as well as the different failure categories and failure types that were considered. Section 5 then discusses conclusions and recommendations for FPGA-based I&C systems, and Section 6 presents the brief summary of the work performed for this paper.

2. FPGA Background

FPGAs are a form of Integrated Circuit (IC) that is programmed by the end user after they have been manufactured. They can be used to perform I&C functions that are performed by other technologies, such as PLCs, microprocessors, Application Specific Integrated Circuits (ASICs) and analog circuits. Although FPGAs use digital logic, they do not require the use of software or operating systems on the chip, setting them apart from most digital logic devices. Due to this, FPGAs are referred to as a form of Programmable Logic Device (PLD) or Programmable Digital Device (PDD) [1,2]. FPGAs have traditionally been programmed with Hardware Description Language (HDL), with the most popular types being VHDL (IEEE 1076-1993) and Verilog (IEEE 1364-2005). In more recent years, tools have been developed to extend other languages, such as Python or MATLAB, to HDL. The use of HDLs means that FPGAs can also be categorized as HDL Programmed Devices (HPD).

There are certain characteristics of FPGAs that created the interest in their use for NPP safety systems. FPGAs have a fast response time, and do not run software or operating systems [1,2]. FPGAs can also be reprogrammed (depending on the type of FPGA used), resulting in reduced obsolescence when compared to other technologies, like ASICs or analog circuits [6]. However, in order for FPGAs to be used in nuclear plant safety systems, they still must be qualified. Additionally, although FPGAs do not run software on the chip, they are configured using software packages (tools), therefore both hardware and software (logic) faults could affect system safety. Moreover, as with all semiconductor technologies, FPGAs can be vulnerable to ionizing radiation, in the form of both destructive and non-destructive interactions with radiation, commonly known as Single Event Effects (SEEs) [1, 3, 4].

The main technologies used in modern FPGA chips are Antifuse, Flash and SRAM (Static Random Access Memory). Antifuse and Flash FPGAs are said to be "non-volatile", as they do not lose their configuration when the power is turned off, as opposed to "volatile" SRAM FPGAs, which must be reconfigured whenever there is a power cycle (on/off). Antifuse FPGAs are also known as One-Time Programmable (OTP), since they can only be programmed once (cannot be reprogrammed), while Flash and SRAM FPGAs are able to be reprogrammed.

3. Failure Mode and Effects Analysis (FMEA)

FMEA are a commonly used method in reliability and safety analysis, and are often performed at the start of a reliability/safety analysis program. FMEAs become an important part of that program. The FMEA was performed on the FPGAs to identify data regarding the failure modes. This includes identifying the potential failure modes, their causes, their potential effects on FPGA-based systems, as well as providing information on how to eliminate or mitigate/control those failure modes. Additionally, the FMEA can identify the effects of latent design errors, and/or determine if the Single Fault Criteria is established [22]. To obtain the data required for this FMEA, an extensive literature review was performed, taking into account a wide variety of information from the international community. This documentation included the aforementioned reports from the US NRC and ORNL, reports from VTT, EPRI and the OECD-NEA, and standards from IEC, IEEE and CSA. Documentation, such as reports and white papers from FPGA manufacturers (Xilinx, Altera and Microsemi) were considered, as was research published in scientific journals and conferences. The data from the literature review was further processed during this research (as seen in Section 4), to provide a more detailed analysis of the failure mode information

The FMEA resulted in a list of 126 possible failure modes, based on the literature review. These failure modes were then broken down in categories based on the point in the FPGA-based system lifecycle where the failures occur, as well as the Failure Types, as discussed in Section 4. The FMEA was also used to document the potential cause(s) of each failure mode, its potential effect(s) on an overall FPGA-based system, and identify the methods to eliminate or mitigate those failure modes. Recommendations are noted from the study of the literature available for these failure modes. These include recommendations for regulatory review of the FPGA-based system(s). The FMEA results from this work provides additional information for use in future modelling and reliability analysis of FPGA-based systems, to further analyze the failures and the methods to mitigate those failures and to develop regulatory review guidelines to implement FPGA-based safety systems.

4. FMEA Results

This section discusses the FMEA results, including the breakdown of the different failure categories and the sets of general failure modes (as discussed in Section 5.2.3 "*Failure Mode Determination"* of IEC 60812) [5]. This section presents the failure modes, effects, mitigations and includes a breakdown of FPGA Failure Types and the "When and Why" Matrix. It should be noted that this research did not consider a specific system. Instead, the FMEA focused on general FPGA-based systems, and only considered the effects of failures at the board level. The effects mentioned in this paper are the effects that the failure modes would have on the system elements that are under consideration. These are referred to as "Local Effects" in Section 5.2.5.2 "*System Initiation, Operation, Control and Maintenance"* of IEC 60812.

4.1 Failure Categories

The results of the FMEA were modelled using two main categories; "Design" and "Operation". The "Design" category consists of two groups "Design Defects" and "Manufacturer Defects", while "Operation" is broken down into "Environmental", "Stress/Aging", and "Maintenance Induced (Human Factors)". These terms are explained in more detail below, and a visual representation is given in Figure 1. These groups are again divided into smaller subsections, based on their effects on the system and the remedy actions to eliminate, mitigate and control these potential failure modes.

Design: Modelled failures with the logic or the chip itself that occur in the design or fabrication stage.

Design Defects: Failures due to problems with the FPGA logic and/or the system hardware.

Manufacturer Defects: Defects in the physical FPGA chip from the manufacturing process.

Operation: Modelled failures that occur while the FPGA is in operation inside the nuclear power plant.

Environmental: Failures that occur due to the environment that the FPGA is operating in. Possible failures would include radiation-induced failures such as SEEs.

Aging/Stress: Failures that occur due to the aging effects experienced by semiconductors, as well as thermal and/or mechanical stress on the FPGA.

Maintenance Induced (Human Factors): Failures that occur during maintenance, such as tampering with the FPGA, that is either intentional or unintentional.

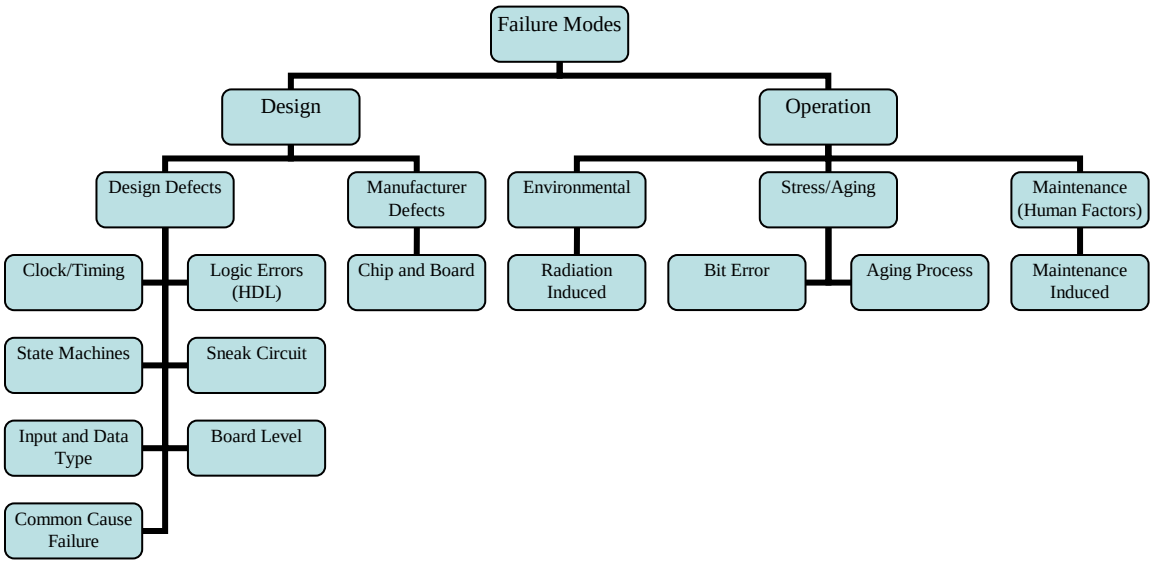

Figure 1: Failure Category Diagram

This categorization was done in this paper for the ease of discussing the remedy actions, and was not present in the literature that was surveyed. For example, for the failure modes considered as "Design Defects", efforts should be made to eliminate/mitigate those failure modes during the design state. In the case of residual failure modes, Built-In Self-Test (BIST) should be incorporated into the design to detect errors that occur during operation. This categorization will also be used to construct a "When and Why" matrix, as seen in Section 4.4.

4.2 Sets of Failure Modes

IEC 60812 describes the different failure groups as a "Failure Effects Summary", so that terminology was used in this report. IEC 60812 defines "failure effect" to be the "consequence of a failure mode in terms of the operation, function or status of the item" [5]. The categories mentioned in Section 4.1 were then broken down into several "Sets of Failure Modes", based on their similar causes and/or failure effects. Each set that was identified in the FMEA includes a description of the set, and methods to eliminate or mitigate those failures. This section is organized with the description of each set of failure modes, followed by the discussion of the avoidance and/or mitigation methods.

4.2.1 Design Defects

Design defects considered in this section focus on logic failures, as well as potential hardware "Faults and Failures to be assumed when quantifying the effect of random hardware failures or to be taken into account in the derivation of safe failure fraction", as specified in IEC 61508-2, Table A.1 that were deemed to be relevant to FPGA-based systems [6].

Clock/Timing Failure Modes

Proper timing is critical for FPGA systems to function as intended, so any design or logic errors that affect the clock, upset the system timing, or in general create timing errors can cause the system to behave incorrectly. Therefore, all potential failures due to the timing/clock behavior must be controlled. There were 20 potential failures that were identified that relate to clock/timing properties.

Many of the failure modes in this section were due to the use of asynchronous (not tied to the clock) signals, and could be remedied by using synchronous (tied to the clock) signals instead. If the input signal is asynchronous, synchronizer chains (double or triple registers) or FIFOs (First In First Out) can be used to synchronize that signal. Timing errors can be identified and eliminated during the design phase through the inclusion of proper timing constraints, Static Timing Analysis (STA), and by performing timing (gate-level) simulations. The gate level simulations are used to expand upon STA (limited when analyzing asynchronous signals and multi-cycle clock paths), verify reset sequence and initialization, power estimation, detect metastability or glitches, and verify the proper timing execution [24].

Logic Error (HDL Programming) Failure Modes

These failure modes refer to failures due to errors in the actual HDL code used to program the FPGA, not an error in the design specifications. The FPGA is programmed by the end user, meaning any programming errors or deficiencies could lead to unforeseen logic errors in the final system(s) [3,4]. Such issues include synthesizer problems, calculation errors, difficulties with simulation, and additional asynchronous behavior. It was found that there were 14 potential failure modes in this category.

The failure modes in this section were due to programming errors, and can be avoided by following HDL programming standards and industry best practices [8,23]. Additionally Section 3.1 "*Reliability*" of NUREG-7006 provides a summary of the methods to avoid these programming errors [3].

State Machine Failure Modes

State Machines, or Finite State Machines (FSM) are a mathematical computational model that is used to design software programs and sequential logic circuits. FSMs have seen extensive use in FPGA-

based I&C systems, meaning any failures in the state machine could cause the system to hang, suppressing the outputs and/or cause the erratic activation of other system elements. The FMEA uncovered 8 potential failure modes for (FPGA) state machines.

IEC 61508-7, Section B.2.3.2 *"Finite State Machines/State Transition Diagrams"* provides general guidance for state machines, while IEC 612566, Section 8.4.6 *"Finite State Machines"* provides requirements for FPGA-based state machines. State machines should be analyzed and tested to ensure that they conform to those standards. Reset signals can also be used, to force the state machine into a state that is already analyzed (such as the starting state), in case errors/hang-ups/deadlock occurs [7,8].

Sneak Circuit Failure Modes

When considering the FPGA logic, the sneak circuits are a design error that could either cause the intended output not to be generated, or cause an unintended output to be generated. Due to the possible system failure, sneak circuits must be analyzed. In this case, it is the only failure mode considered. It should be noted, however, that sneak circuits could be caused by either hardware design errors or software design errors. Sneak circuits are a form of latent design error.

Sneak circuits should be eliminated where possible. Even for small, combinational FPGA-based system designs with a limited number of I/O, and where 100% testability can be obtained, that testing may not guarantee that the system is sneak circuit free [3,4]. To remedy this, Sneak Circuit Analysis (SCA) should be performed to locate and eliminate sneak circuits [9,10].

Input and Data Type Failure Modes

This section includes information about possible input or data type errors. The inputs can overflow and/or become stuck, causing inaccurate data to be propagated through the system. The use of different data types (such as fixed-point or floating-point packages), which currently are not well-supported by many vendor tools (e.g. VHDL-2008 is not as universally supported as VHDL-1993). There were 6 failure modes identified in this set. At the time of publication, the current standards for HDL code include IEEE-1076 (VHDL), IEEE-1364 (Verilog), and IEEE-1800 (SystemVerilog).

In order to mitigate these failure modes, the input range should be properly defined before implementation, to avoid overflow altogether. However, in case overflow does occur, it must be detected, and there should be alerts that are sent to the operators to warn of errors. Resolution and Resize errors can occur when using Fixed-Point mathematics (a data type that has a fixed number of digits before and after the decimal point, as opposed to the Floating-Point math that is commonly used in modern computers), which is common in FPGAs. The proper resolution for all values in the system (especially critical parameters such as setpoints) should be carefully calculated beforehand. The potential failures that could be caused by the newer fixed-point and floating-point data type packages (such as those in VHDL-2008) can be avoided by using the better supported standards (such as VHDL-1993) instead of the newer packages [24].

Board Level Failure Modes

These represent general, high-level failure modes (potentially representing fault tree top events), which require consideration under IEC 61508-2 [6]. Complying with that standard will assist in achieving a system where the hardware will function reliably. This section includes a subset of the IEC 61508-2, Table A.1 items that are relevant to FPGA-based systems, which accounts for 9 assumed failure modes.

These failure modes were taken from the standard IEC 61508-2, Table A.1. Additional information on those failure modes is presented in Tables A.2 to A.14 of that standard. These represent general failure modes, that any electrical/electronic safety system could encounter, and must be eliminated (or controlled). The methods for testing, detection, eliminating and controlling these failures can be found in IEC 61508 [6].

FPGA Chip and Board Failure Modes

These are failures that can occur in the manufacturing/fabrication process of the physical FPGA chip (hardware), and could result in damaged pins, or impurities that cause corrosion. The failure modes of the chips itself must be known, to ensure that the chips used in the FPGA system are reliable. There were 9 total failure modes identified for this group.

To mitigate failures due to the FPGA chip and board, the chips should be inspected and tested for any defects and/or impurities that would cause failures and/or accelerate the aging process. The chips should be inspected for cleanliness (such as water or foreign particles), that could lead to corrosion or ion mobility failures. The chips should be tested to check for any damaged pins, and the supply voltage/current should be tested to eliminate power-up or power pin decoupling errors [3,4,13]. Additionally, Bent Pin Analysis (BPA) (also called Cable Failure Matrix Analysis (CFMA)) should be performed, to determine the potential hazards of bent/damaged pins, and the mitigation required to control those hazards.

Common Cause Failures (CCF)

Common Cause Failures (CCF) are a serious issue in reliability engineering, as seemingly redundant systems can fail due to a single initiating event, which removes the underlying assumption that all failure modes are independent. Therefore, one cannot only consider independent and random failures, as CCFs must be mitigated, in order to have a reliable design. There is only 1 failure mode in this set (CCF itself), and it is a form of latent design error that is assumed to exist in the design and cannot be found through testing. CCF was included in the “Design Defect” section, as the system logic has the potential to be to cause of a CCF, however hardware or environmental factors may also cause CCFs, as stated below.

Conventional CCF causes are [5]:

- Design (Software/Logic)
- Manufacturing (Component Flaws)
- Environmental (Temperature, Electrical Interference)
- Human Factors (Maintenance Actions)

FMEAs have limited use when analyzing CCFs, however it can be used to study all the possible causes that could trigger a CCF. Traditionally, a combination of different methods is used to mitigate CCFs. These include using functional diversity and defense-in-depth, system modelling, component analysis, and the physical separation of components. If the system is a very simple, asynchronous design, it is possible to obtain 100% test coverage, which would eliminate any latent design (logic) error. However, that would not remove the vulnerability to CCF due to other causes. Additional standards from IEC (60880, 61566, 61802 Section 6.1), IEEE (7.4.3.2), and the CNSC (REGDOC 2.5.2) should be considered when designing systems that are resilient to common cause failures [5,7,11,12,21].

4.2.2 Stress/Aging Failures

These failures are due to the aging process. As with any technology, aging effects will eventually render the chip unusable. The failures due to the aging process must be mitigated, as these failures modes cannot be completely eliminated during the design phase of the FPGA-based system.

Bit Error Failure Modes

These failures are almost exclusively due to the continuous reconfiguration of SRAM FPGAs. Eventually, the multiple reconfiguration cycles will result in bit failures. Bit errors, coupling faults, stuck bits, and data degradation were all modeled under the umbrella term of "Bit Error Failures", as they all affect the state of the bits ("0" or "1"). It was seen that there were 9 total failure modes in this category [14]. However, configuration errors are still possible in Antifuse and Flash FPGAs, and non-volatile memory can degrade over time, so 2 of those failure modes could still apply to non-volatile FPGAs.

Bit Error Failure Modes can largely be avoided by using Antifuse (OTP) FPGAs. The Antifuse FPGAs will eventually lose data, however the Mean Time To Failure (MTTF) can be calculated to predict when that failure will happen. Regardless of the FPGA technology (SRAM, Flash or Antifuse), the configured system should then be tested thoroughly, to verify that it performs correctly and matches the simulations, to ensure no bit errors have occurred, or detect the failure(s) if any did occur. If a bit error occurs with an OTP FPGA, then the FPGA would have to be replaced, however the Flash or SRAM FPGA could be reconfigured in the event of a configuration or routing error. However, the SRAM chip must be re-configured every time there is a power cycle, meaning that it must be tested thoroughly every time, whereas Flash and OTP chips are only configured once and retain their configuration through power on/off cycles. The constant re-configuring that occurs with SRAM FPGAs is what causes it to be susceptible to configuration failures, while OTP and Flash FPGAs are more resilient.

Aging Process Failure Modes

Failure modes due to aging cannot be eliminated during the design and implementation part of the life-cycle, and as such must be mitigated. Many of these failures have dependencies on temperature, electric properties (electric field, voltage, or current) or the material properties. There were 15 total aging failure modes found during the FMEA research.

Aging process failures cannot be eliminated from the FPGA. Statistical methods, such as thermal aging calculations/test and MTTF calculations can be used to estimate when failures will occur, and the MTTF data should be reflected in the maintenance program. The use of (cold) system redundancy could be included, to ensure the system will function properly if one component/function fails. Built-In Self-Test (BIST) should be implemented, to test for signs of aging process failures, and to indicate to maintenance personnel that the aging process has become hazardous, and the FPGA is starting to fail. However, aging failures could also affect the BIST, so periodic testing should be performed at scheduled maintenance intervals. The BIST features should be isolated (separate from the safety features), so that the BIST functions do not interfere with the primary safety functions of the system.
The periodic tests should include tests for clock period, temperature, power use/dissipation and short-circuit currents, as those parameters can indicate that the chip is starting to degrade and may need to be replaced. The aging process failures will eventually require the FPGA to be replaced [3,4,13,15].

4.2.3 Environmental Induced Failures

Radiation Induced Failure Modes

This set discusses failures caused by radiation interactions with the semiconductor materials of the FPGA. Many of these failures are part of a group known as Single Event Effects (SEE), however there are multiple events as well. Radiation failures can either cause destructive (hard) or nondestructive (soft) errors. There were 16 individual failure modes that were identified for radiation interactions. It should be noted that this set does not consider general environmental factors (e.g. temperature, humidity, etc.), that are part of the general environmental qualification. However, guidelines for environmental qualification can be found in RegDoc 2.5.2 and CSA N290.13 [12,25].

Radiation Induced failure modes can be eliminated by using Antifuse or Flash-based FPGAs, as these materials are resistant to radiation interference. Additionally, the use of redundancy, such as Double Modular Redundancy (DMR) or Triple Modular Redundancy (TMR) can be implemented to mitigate radiation induced failure modes. Various error detection and correction codes have also been developed, to check for and correct errors such as those caused by SEEs [3,4,16,17].

4.2.4 Maintenance Induced (Human Factors) Failures

Maintenance Induced Failures

Maintenance failures are due to human (personnel) issues, and are the result of either unintended actions, such as accidents or neglect, or intended actions, such as the purposeful reprogramming of the FPGA. It was seen that there are 6 potential failure modes that fit into this section.

The unauthorized or accidental reprogramming/erasure can be eliminated through the use of administrative control. Additionally, the use of OTP FPGAs (or disable reprogramming capability on FLASH FPGAs), is recommended, as the program cannot be altered. If SRAM FPGAs are employed, then the contents of the SRAM cannot be altered. Additional methods to control access include encryption, such as the Advanced Encryption Standard (AES), restricting/disabling the Joint Test Action Group (JTAG) access port, and restricting access of personnel to the FPGA itself. The FPGA-based system should also inform control room personnel if attempts are made to access the FPGA bitstream, attempt to reprogram the FPGA, or access any data on the FPGA, while the system is in operation. In terms of Electrostatic Discharge (ESD), it is a concern that ESD could damage the FPGA chip during operation, especially during maintenance activities. The issue of ESD can be reduced by ensuring that the system can withstand ESD according to the industry standards (the ESD test requirements can be found in IEC 62003, with guidance for performing these tests found in IEC 61000-4-1), and by following proper maintenance procedures.

4.3 FPGA Failure Types

To gain more understanding from the FMEA results, additional “Failure Types” were added during this research to perform a more in-depth analysis of the failure modes, as seen in Table 1.

Table 1: FPGA Failure Types

Failure Type	Definition
A	Hardware (Degradation or General) Failure

B	Logic (“Programming”) Failure
C	Radiation-Induced Failure

Table 1 includes the three main types of failure modes that the FPGA can experience. The “Hardware” failures include the physical degradation or destruction of the chip caused by wear effects, or general hardware effects such as those mentioned in IEC 61508-2 [6], as discussed in 4.2.2 and 4.2.1, respectively. The “Logic” failures are due to faults introduced in the FPGA by logic design, and can include data overflow or state machine errors, as seen in Section 4.2.1. “Radiation” failures are due to radiation interactions with the FPGA, and therefore include radiation failures such as SEU, as shown in 4.2.3. It should be noted that all the “Radiation” failures are also hardware failures (the radiation strike will cause upsets and/or damage to the chip hardware), so all Type C failures could also be Type A failures. Similarly, many of the “Logic” failures, such as timing failures or data loss can be caused by “Hardware” failures, or could indirectly cause “Hardware” failures themselves. However, for this research, the three Failure Types were made separate, in order better describe and categorize the failure modes. The results of the Failure Types were then graphed as shown in Figure 2.

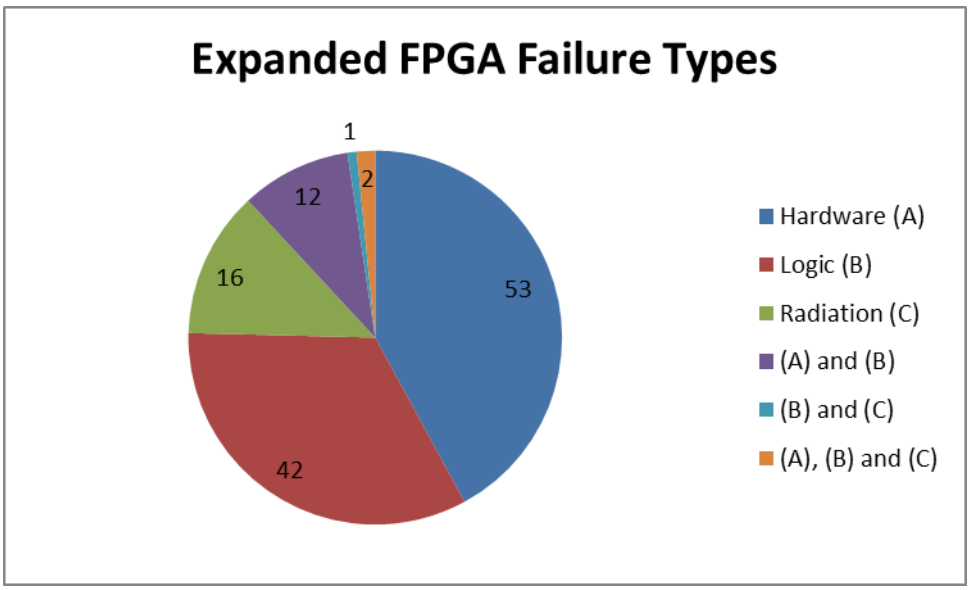

Figure 2: Expanded FPGA Failure Types

It was seen that 53 modes are only Hardware failures, 42 are only Logic failures, while 16 are only Radiation failures. In addition, it was found that 12 of the faults could be due to either Hardware and Logic, 1 fault was due to Radiation and Logic, and 2 faults could be caused by any of the 3 Failure Types (this included the Common Cause Failure (Section 4.2.1). When this graph is examined, the total number of failures adds to 126. The large number of hardware failures is partly due to the failure modes included in the Joint Electron Device Engineering Council (JEDEC) document JEP-122G, which describes general semiconductor failures, which can occur in the various semiconductor devices [13]. Additionally, all forms of wear and degradation will show up as hardware failures, as well as any non-specific (general) hardware fault. Logic faults are the second most represented, as there are a large number of potential programming issues that should be avoided, although some of them are easy to avoid (certain programming methods should not be used, but no other work is needed). Radiation effects are the least represented, due to the finite number of known radiation interactions. It should be

noted that in cases with multiple Failure Types, it does not necessarily mean that all of the Failure Types could cause the failure, but that there is at least some dependence on those types.

4.4 When and Why Matrix for FPGA Failure Modes

Another way to categorize the failure modes is through the use of a "When and Why" matrix, which was created using the results of the FMEA that was performed. The "When" refers to the stage in the FPGA-based system lifecycle that the failure occurs, and the "Why" refers to the failure category (failure modes), as discussed in Section 4.1. In this case, the "When" was broken down into two categories; "Design/Fabrication" and "Operation", while the "Why" was divided into five groups; Manufacturer Defects, Design Defects, Environmental, Stress/Aging, and Maintenance (Human Factors). The definitions of the categories and groups are given in Section 4.1. The "When and Why" matrix was then graphed, with the results shown in Figure 3. The defects (chip and logic/design defects), were all included in the design stage, with 10 and 63 failure modes, respectively. The operation stage then included the environmental failure modes (18), stress/aging (31), and maintenance (human factors) (7). As in previous cases, there is some overlap, due to double counting certain modes, as they can be included in multiple categories, causing the total count to be 129. The design stage is seen to have the most, as all logic errors are included in that section, as well as several hardware faults that are due to defects with the chip in the design process. These failure modes encompass both the FPGA and the board. In Figure 3, the blue and red colours both indicate "When" on the graph. Blue represents the "Design" stage, and Red the "Operation" stage.

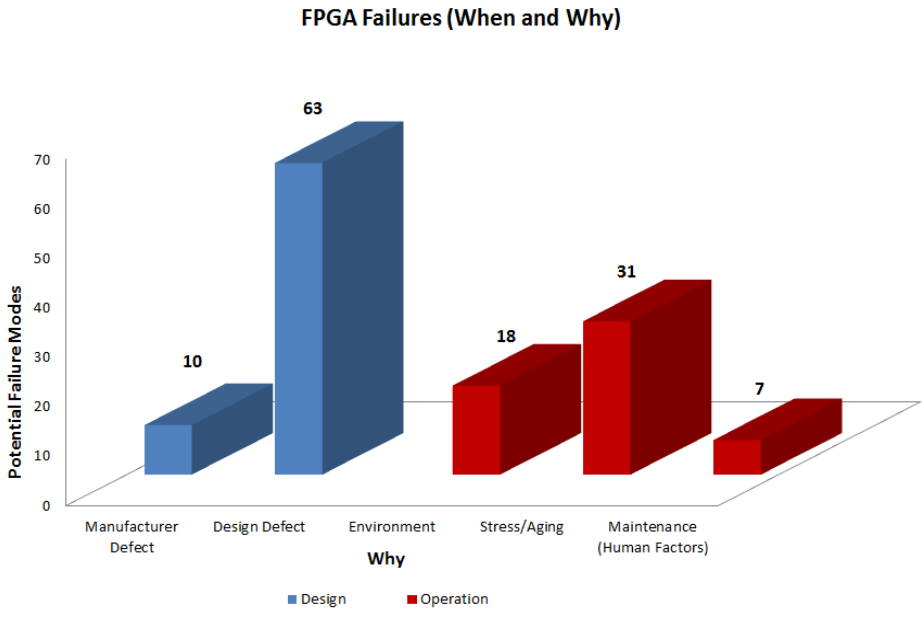

Figure 3: When and Why Results for FPGA Failure Modes

The modelling of the failure modes using the "When and Why" matrix uncovered two important points. The first point being that the majority of the failures occur during the Design/Fabrication stage (73 failure modes vs 56 in "Operation"). This shows that the most of the failures are related to the design and implementation of the FPGA-based system, and these failure modes can be eliminated before the system is implemented (except for the latent design errors due to sneak circuits and CCF). The "Environment Induced" failure modes can also be eliminated in the design phase, as

discussed in Section 4.2.3. The second point is that there are still 31 potential failure modes due to "Stress/Aging" effects, which are residual errors and cannot be eliminated. However, these failure modes can be revealed through the use of BIST and/or periodic testing. These tests would be an important feature for mitigating the failure modes due to the aging process, to identify when the FPGA-based system is starting to fail.

5. Conclusions and Recommendations

The application of FPGAs to NPP I&C systems comes with potential benefits, however it also introduces potential new failure modes that may affect the operation and margin of safety. The data and results of the FMEA have produced several insights into the FPGA technology in general and its applicability to the safety systems in the nuclear domain. A list of conclusions and recommendations that were derived from this research are given in this section.

1.) This research compiled, modelled and studied a comprehensive list of failure modes and concluded that effects of these potential failure modes could be avoided, controlled and contained (See Section 4.2).

    From this study, it is concluded that there are no fundamental technical barriers preventing FPGA-based systems from being used in Canadian NPP I&C systems, including safety systems. This is provided that all identified failure modes are accounted for in that system.

2.) Some of the potential failure modes for the FPGAs seen in this study are not specific to FPGAs, but are "inherited failure modes". These include failures with the logic (programming) and/or hardware, as discussed in Section 4.2.1. It also includes failure modes common to all semiconductor devices, such as aging process failures and SEEs, as seen in Sections 4.2.2 and 4.2.3, respectively. One potential inherited failure due to programming (software) errors is that of latent design errors, which may manifest themselves in the form of sneak circuits or CCF after implementation, as they could not be detected during testing.

    It is recommended that the effects of latent design errors (CCF and sneak circuits) be considered when designing and implementing the FPGA-based system(s). CCF should be mitigated through the use of diversity and defence-in-depth.

3.) The failure modes identified from this study could occur at different stages in the FPGA-based system life-cycle, designated as the "Design", and "Operation" groups (See Section 4.2). One of the benefits of this categorization is to provide oversight to facilitate the avoidance and control of the identified failure modes.

    It is recommended that the failure modes introduced in the "Design" stage be avoided by following industry standards and best practices, and rigid engineering processes. Failure modes that occur in the "Operation" stage should be detected and revealed using BIST and periodic testing at proper maintenance intervals. The necessary testing and maintenance principles should be accommodated in the design.

4.) The FPGA-based system development is very similar to the hardware design, however, the configuration of the logic for each FPGA chip is done using HDL, and therefore the design process also shares aspects of software-based system design.

All code attributes should be verified, however in the event that obtaining 100% testability is impractical it is recommended that particular attention is given to certain cases. These include boundary and corner cases, logic branches inside process statements, sensitivity lists, and to identify any unused code paths.

5.) The final product of an FPGA-based system is a form of digital hardware, so the clock and timing behavior is more critical to the correct system operation than in software-based systems. A large number of potential failure modes related to clock/timing aspects of the FPGA-based system were identified (See Section 4.2). Therefore verification and validation should demonstrate that that the logic is correct, as well as demonstrating that the clock/timing behavior is correct.

It is recommended that timing analysis and timing simulations should be performed on the FPGA-based systems, and the implemented system should also be tested, to verify the timing behavior.

6.) The use of non-standard HDL languages and/or extensions to the HDL languages has the potential to introduce failure modes into the design or logic. The extensions/packages (seen in Section 4.2.1) may not be fully supported by design software, which could lead to failures with the simulation and/or configuration of the FPGA logic. This also reduces portability of the code, as every chip and design software may not support all the extensions.

It is recommended that FPGAs should be programmed using HDL that conforms to the IEEE standards (IEEE-1076 for VHDL, IEEE-1364 for Verilog, and IEEE-1800 for SystemVerilog), and that extensions not part of those standards are avoided.

7.) FPGA design tools may include options (potentially by default) to automatically optimize the HDL code when synthesizing. A concern is that the optimized circuits may be different from the intended circuits described by the HDL code, such as removing intended redundancy and making verification more difficult. In addition, the behavior when testing the FPGA implementation may also produce different results than the intended HDL code, which would further complicate system verification. Optimization also has the potential to introduce errors into the design, such as causing interference between circuits that should be isolated.

It is recommended that the design tools used for HDL programming and synthesis do not make use of any optimization features that may be available in those tools.

8.) The failure mode information gained from this FMEA can be applied to FPGA-based systems for modelling and analysis.

It is recommended that future work will involve using the FMEA results as part of a test system, to perform research into dynamic methodologies and comparisons of traditional and dynamic hazard analysis methods, for potential use in the V&V process.

6. Lessons Learned from International Implementations

FPGAs have seen use in a variety of systems (including safety systems) in various countries in recent years. This section discusses some of the lessons learned during the licensing process for these systems, as well as information on some of the best-known FPGA platforms. Guides/standards that are specific to FPGAs are also listed, as they could be considered during the design process.

Several FPGA-based platforms exist for safety-related systems. Some of the most well-known platforms are available from Radiy, Westinghouse and Lockheed–Martin. The Radiy platform is known as "Radiy Digital Control (RadICS)" platform, and has seen use in NPPs in Ukraine and Bulgaria, as well as the Embalse plant in Argentina. To date, over 90 systems have been installed, and the RadICS platform has been certified as SIL 3 by Exida [25]. Westinghouse supplies the Advanced Logic System (ALS) platform (after acquiring CS Innovations), which has seen use in the Wolf Creek NPP, and could see use in the Diablo Canyon NPP as well as the AP1000 reactors. Much of the information on the ALS FPGA development has not been made public, however the FPGA logic has been assigned as software integrity level 4 [25,26]. The State Nuclear Power Automation Systems Engineering company (SNPAS) in China has partnered with Lockheed-Martin, to develop the NuPAC system for use in safety systems. Currently, none of the NuPAC systems have seen operational use, however the system is being reviewed by the US NRC for generic approval [25].

The licensing of the ALS by the NRC provides some information on what was required for the approval process, and some of these factors are discussed in Section 5 of this paper. The NRC reviewed the FPGA-based system in the same way as a microprocessor based system, as the NRC considered the HDL programming to be vulnerable to the same issues as traditional software code. Additionally, the NRC identified the following factors as "helpful" in the licensing process [25]:

- Functional Simplicity of the FPGA
- No microprocessor or embedded memory
- No inter-channel communication
- Separation of non-safety and safety features

The NRC also stressed the importance of design diversity, and identified several methods to improve system diversity [25]:

- Use of multiple (different) programming languages
- Different synthesis directives and/or FPGA devices
- Multiple (diverse) implementations using redundant channels
- Separation of the diverse implementations into independent channels

It was seen in the United Kingdom that concerns about the lack of diversity in the ALS Diverse Actuation System (DAS) led to Westinghouse switching the design from FPGAs to conventional electronics [25].

A number of standards/guides could be considered when developing an FPGA-based safety system. Many of these documents are common to nuclear power plants, digital I&C systems or software

systems, however several of them were identified as FPGA-specific documents. These include IEC 61508 (contains some FPGA-specific information), IEC 62566, NUREG/CR-7006, EPRI-TR 1019181 and EPRI-TR 1022983 [25].

7. Summary

The FMEA was performed to identify the potential board-level failure modes for FPGA-based systems, find their causes, their potential effects on an I&C system, and to identify ways to eliminate or mitigate those failures. There are regulatory guide(s), technical reports, standards and scientific papers that consider FPGA failure modes, however none of those documents individually contained a comprehensive list of the potential FPGA failure modes. This paper sought to identify and compile a list of failure modes and provide a discussion of those failure modes that was as detailed as possible. The "Failure Types" for each failure modes were considered, to perform more in-depth modelling of their causes. The failure modes were broken into categories based on the stage of the life-cycle that the failures occur, and the overall cause of those failure modes, which were then graphed in a "When and Why" matrix. Lastly, conclusions and recommendations for FPGA-based NPP I&C systems were presented based on the results from the FMEA and failure mode modelling.

8. References

[1] Valtion Teknillinen Tutkimuskeskus (VTT), "The current state of FPGA technology in the nuclear domain". Vuorimiehentie, Finland, 2011.

[2] McNelles, P. and Lu, L., "A Review of the Current State of FPGA Systems in Nuclear Instrumentation and Control". Proceedings of the 21st International Conference on Nuclear Engineering, Chengdu, July 30-August 2, 2013. ICONE.

[3] United States Nuclear Regulatory Commission (U.S. NRC), NUREG-7006, "Review Guidelines for Field Programmeable Gate Arrays in Nuclear Power Plant Safety Systems", Washington D.C., 2010.

[4] Bobrek, M., & Bouldin, D., "Review Guidelines for FPGAs in NPP Safety Systems", Oak Ridge, Tennessee, 2010.

[5] International Electrotechnical Commission (IEC), 60812, *Analysis techniques for system reliability- Procedures for failure mode and effects analysis (FMEA),* Geneva, 2006.

[6] IEC, 61508-2, "Functional Safety of electrical/electronic/programmable electronic safety related systems- Part 2: Requirements for electrical/electronic/programmable electronic safety-related systems", Geneva, 2010.

[7] IEC, 61508-7, "Functional Safety of electrical/electronic/programmable electronic safety related systems- Part 7: Overview of techniques and measures", Geneva, 2010.

[8] IEC, 62566, "Development of HDL-programmed integrated circuits for systems performing category A functions", Geneva, 2012.

[9] European Space Agency (ESA), ECSS-Q-40-04A Part 1, "Sneak Analysis- Part 1: Method and Procedure", Noordwijk, The Netherlands, 1997.

[10] ESA, ECSS-Q-40-04A Part 2, "Sneak Analysis – Part 2: Clue list", Noordwijk, The Netherlands, 1997.

[11] IEEE, 7.4.3.2, "Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Station", New York, 2010.

[12] Canadian Nuclear Safety Commission (CNSC), REGDOC 2.5.2, "Design of Reactor Facilities: Nuclear Power Plants", Ottawa, Canada, 2014.

[13] JEDEC Solid State Technology Association, JEP-122G, "Failure Mechanizms and Models for Semiconductor Devices", Arlington, Virginia, 2011.

[14] "March LR: A Test for Realistic Linked Faults". Proceedings of the 14th VLSI Test Symposium, Princeton, 1996. van der Goor, A. J., Gaydadjiev, G. N., Yarmolik, V. N., & Mikitjuk, V. G. pp. 272-280. IEEE.

[15] National Aeronautics and Space Administration (NASA), JPL 08-5, "Microelectronics Reliability: Physics-of-Failure Based Modeling and Lifetime Evaluation". Pasadena, California, 2008.

[16] Sturesson, F., "Single Event Effects (SEE) Mechanism and Effects", Space Radiation and its Effects on IEEE Components, European Space Agency, 2009.

[17] Mutuel, L. H., "Appreciating the Effectiveness of Single Event Effect Mitigation Techniques". Proceedings of the 33rd Digital Avionics Systems Conference, Colorado Springs, 2014, SB1-11, IEEE.

[18] Xilinx, WP433, "Understanding and Mitigating System-Level ESD and EOS Events in Xilinx 7 Series Device", San Jose, California, 2013.

[19] Electric Power Research Institute (EPRI), TR-1019181, "Guidelines on the Use of Field Programmeable Gate Arrays (FPGAs) in Nuclear Power Plant I&C System". Palo Alto, California, 2009.

[20] EPRI, TR-1022983, "Recommended Approaches and Design Criteria for Application of Field Programmable Gate Arrays in Nuclear Power Plant Instrumentation and Control Systems", Palo Alto, California, 2011.

[21] IEC, 60880 , "Nuclear power plants - Instrumentation and control systems important to safety - Software aspects for computer-based systems performing category A functions", Geneva, 2006.

[22] U.S. NRC, Topical Report 6002-00301, "Advanced Logic System Topical Report", Washington D.C., 2010.

[23] Radio Technical Commission for Aeronautics (RTCA) Incorporated, DO-254, "Design Assurance Guidance For Airborne Electronic Hardware", Washington, D.C., 2000.

[24] Digital Instrumentation and Controls Working Group (DICWG), DICWG-05, "Common Position on the Treatment of Hardware Description Language (HDL) Programmed Devices for use in Nuclear Safety Systems", Issy-les-Moulineaux, 2013.

[25] Menon, Catherine and Guerra, Sofia, "Field Programmable Gate Arrays in Safety Related Instrumentation and Control Applications", Energiforsk, Sweden, 2015.

[26] ALS V&V Plan, Westinghouse non-proprietary document 6002-00003-NP, Rev 7, NRC document ML 12332A274, October 2012.