IAEA Nuclear Energy Series

No. NP-T-3.17

Application of Field Programmable Gate Arrays in Instrumentation and Control Systems of Nuclear Power Plants

IAEA
International Atomic Energy Agency

IAEA NUCLEAR ENERGY SERIES PUBLICATIONS

STRUCTURE OF THE IAEA NUCLEAR ENERGY SERIES

Under the terms of Articles III.A and VIII.C of its Statute, the IAEA is authorized to foster the exchange of scientific and technical information on the peaceful uses of atomic energy. The publications in the **IAEA Nuclear Energy Series** provide information in the areas of nuclear power, nuclear fuel cycle, radioactive waste management and decommissioning, and on general issues that are relevant to all of the above mentioned areas. The structure of the IAEA Nuclear Energy Series comprises three levels: **1 — Basic Principles and Objectives**; **2 — Guides**; and **3 — Technical Reports**.

The **Nuclear Energy Basic Principles** publication describes the rationale and vision for the peaceful uses of nuclear energy.

Nuclear Energy Series Objectives publications explain the expectations to be met in various areas at different stages of implementation.

Nuclear Energy Series Guides provide high level guidance on how to achieve the objectives related to the various topics and areas involving the peaceful uses of nuclear energy.

Nuclear Energy Series Technical Reports provide additional, more detailed information on activities related to the various areas dealt with in the IAEA Nuclear Energy Series.

The IAEA Nuclear Energy Series publications are coded as follows: **NG** — general; **NP** — nuclear power; **NF** — nuclear fuel; **NW** — radioactive waste management and decommissioning. In addition, the publications are available in English on the IAEA Internet site:

http://www.iaea.org/Publications/index.html

For further information, please contact the IAEA at PO Box 100, Vienna International Centre, 1400 Vienna, Austria.

All users of the IAEA Nuclear Energy Series publications are invited to inform the IAEA of experience in their use for the purpose of ensuring that they continue to meet user needs. Information may be provided via the IAEA Internet site, by post, at the address given above, or by email to Official.Mail@iaea.org.

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

The following States are Members of the International Atomic Energy Agency:

AFGHANISTAN
ALBANIA
ALGERIA
ANGOLA
ANTIGUA AND BARBUDA
ARGENTINA
ARMENIA
AUSTRALIA
AUSTRIA
AZERBAIJAN
BAHAMAS
BAHRAIN
BANGLADESH
BARBADOS
BELARUS
BELGIUM
BELIZE
BENIN
BOLIVIA, PLURINATIONAL STATE OF
BOSNIA AND HERZEGOVINA
BOTSWANA
BRAZIL
BRUNEI DARUSSALAM
BULGARIA
BURKINA FASO
BURUNDI
CAMBODIA
CAMEROON
CANADA
CENTRAL AFRICAN REPUBLIC
CHAD
CHILE
CHINA
COLOMBIA
CONGO
COSTA RICA
CÔTE D'IVOIRE
CROATIA
CUBA
CYPRUS
CZECH REPUBLIC
DEMOCRATIC REPUBLIC OF THE CONGO
DENMARK
DJIBOUTI
DOMINICA
DOMINICAN REPUBLIC
ECUADOR
EGYPT
EL SALVADOR
ERITREA
ESTONIA
ETHIOPIA
FIJI
FINLAND
FRANCE
GABON
GEORGIA
GERMANY
GHANA
GREECE
GUATEMALA
GUYANA
HAITI
HOLY SEE
HONDURAS
HUNGARY
ICELAND
INDIA
INDONESIA
IRAN, ISLAMIC REPUBLIC OF
IRAQ
IRELAND
ISRAEL
ITALY
JAMAICA
JAPAN
JORDAN
KAZAKHSTAN
KENYA
KOREA, REPUBLIC OF
KUWAIT
KYRGYZSTAN
LAO PEOPLE'S DEMOCRATIC REPUBLIC
LATVIA
LEBANON
LESOTHO
LIBERIA
LIBYA
LIECHTENSTEIN
LITHUANIA
LUXEMBOURG
MADAGASCAR
MALAWI
MALAYSIA
MALI
MALTA
MARSHALL ISLANDS
MAURITANIA
MAURITIUS
MEXICO
MONACO
MONGOLIA
MONTENEGRO
MOROCCO
MOZAMBIQUE
MYANMAR
NAMIBIA
NEPAL
NETHERLANDS
NEW ZEALAND
NICARAGUA
NIGER
NIGERIA
NORWAY
OMAN
PAKISTAN
PALAU
PANAMA
PAPUA NEW GUINEA
PARAGUAY
PERU
PHILIPPINES
POLAND
PORTUGAL
QATAR
REPUBLIC OF MOLDOVA
ROMANIA
RUSSIAN FEDERATION
RWANDA
SAN MARINO
SAUDI ARABIA
SENEGAL
SERBIA
SEYCHELLES
SIERRA LEONE
SINGAPORE
SLOVAKIA
SLOVENIA
SOUTH AFRICA
SPAIN
SRI LANKA
SUDAN
SWAZILAND
SWEDEN
SWITZERLAND
SYRIAN ARAB REPUBLIC
TAJIKISTAN
THAILAND
THE FORMER YUGOSLAV REPUBLIC OF MACEDONIA
TOGO
TRINIDAD AND TOBAGO
TUNISIA
TURKEY
UGANDA
UKRAINE
UNITED ARAB EMIRATES
UNITED KINGDOM OF GREAT BRITAIN AND NORTHERN IRELAND
UNITED REPUBLIC OF TANZANIA
UNITED STATES OF AMERICA
URUGUAY
UZBEKISTAN
VANUATU
VENEZUELA, BOLIVARIAN REPUBLIC OF
VIET NAM
YEMEN
ZAMBIA
ZIMBABWE

The Agency's Statute was approved on 23 October 1956 by the Conference on the Statute of the IAEA held at United Nations Headquarters, New York; it entered into force on 29 July 1957. The Headquarters of the Agency are situated in Vienna. Its principal objective is "to accelerate and enlarge the contribution of atomic energy to peace, health and prosperity throughout the world".

IAEA NUCLEAR ENERGY SERIES No. NP-T-3.17

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

INTERNATIONAL ATOMIC ENERGY AGENCY
VIENNA, 2016

COPYRIGHT NOTICE

Marketing and Sales Unit, Publishing Section
International Atomic Energy Agency
Vienna International Centre
PO Box 100
1400 Vienna, Austria
fax: +43 1 2600 29302
tel.: +43 1 2600 22417
email: sales.publications@iaea.org
http://www.iaea.org/books

Printed by the IAEA in Austria
January 2016
STI/PUB/1701

IAEA Library Cataloguing in Publication Data

Names: International Atomic Energy Agency.
Title: Application of field programmable gate arrays in instrumentation and control systems of nuclear power plants / International Atomic Energy Agency.
Description: Vienna : International Atomic Energy Agency, 2016. | Series: IAEA nuclear energy series, ISSN 1995–7807 ; no. NP-T-3.17 | Includes bibliographical references.
Identifiers: IAEAL 16-01018 | ISBN 978–92–0–103515–8 (paperback : alk. paper)
Subjects: LCSH: Nuclear power plants — Instruments. | Nuclear reactors — Control. | Field programmable gate arrays.
Classification: UDC 621.039.56 | STI/PUB/1701

FOREWORD

One of the IAEA's statutory objectives is to "seek to accelerate and enlarge the contribution of atomic energy to peace, health and prosperity throughout the world." One way this objective is achieved is through the publication of a range of technical series. Two of these are the IAEA Nuclear Energy Series and the IAEA Safety Standards Series.

According to Article III.A.6 of the IAEA Statute, the safety standards establish "standards of safety for protection of health and minimization of danger to life and property". The safety standards include the Safety Fundamentals, Safety Requirements and Safety Guides. These standards are written primarily in a regulatory style and are binding on the IAEA for its own programmes. The principal users are the regulatory bodies in Member States and other national authorities.

The IAEA Nuclear Energy Series comprises reports designed to encourage and assist R&D on, and application of, nuclear energy for peaceful uses. This includes practical examples to be used by owners and operators of utilities in Member States, implementing organizations, academia, and government officials, among others. This information is presented in guides, reports on technology status and advances, and best practices for peaceful uses of nuclear energy based on inputs from international experts. The IAEA Nuclear Energy Series complements the IAEA Safety Standards Series.

Operating nuclear power plants worldwide are, or will soon be, facing the need to replace or upgrade both analogue and digital instrumentation and control (I&C) systems. New plants are being designed with new I&C systems. In the case of safety and safety related systems, these new systems need to be qualified and accepted based on appropriate criteria. Regulatory approval for microprocessor based safety and safety related systems involves confirmation of evidence that the design and the verification and validation processes have ensured that the application will fulfil its requirements and its intended design function, and this can result in significant cost and time expenditure before implementation. As a result, it can be quite costly to obtain approval for small, simple systems incorporating safety functions when the systems are developed using microprocessor based technology.

Field programmable gate arrays (FPGAs) are gaining increased global attention for application in nuclear power plant I&C systems, particularly for safety and safety related applications, but also for non-safety applications. This is because, compared with microprocessor based systems, FPGA based solutions can generally be made simpler, more testable, less reliant on complex software (e.g. operating systems) and easier to qualify for safety and safety related applications. In addition, they can offer acceptable solutions to diversity requirements and new opportunities for cost effective, long term support over extended plant lifetimes.

In response to the nuclear industry's interest in the use of FPGA technology, the IAEA decided to join the efforts of the Topical Group on Field Programmable Gate Array Applications in Nuclear Power Plants (TG-FAN). It became obvious in the TG-FAN meetings that FPGA based applications were becoming a more important part of both operating and new nuclear power plants. Therefore, it was deemed important that guidance on the application of FPGAs in I&C systems for nuclear power plants be developed to help plant owners, suppliers, regulators and researchers decide whether the technology constitutes a viable option for their applications. The IAEA agreed with the need for guidance, and initiated this new IAEA Nuclear Energy Series publication.

The report was produced by a committee of international experts and advisors from numerous countries. The IAEA wishes to acknowledge the valuable assistance provided by the contributors and reviewers listed at the end of the report, especially the contribution made by J. Naser (United States of America) as the chair of the authoring group. The IAEA officer responsible for this publication was J. Eiler of the Division of Nuclear Power.

EDITORIAL NOTE

Guidance provided here, describing good practices, represents expert opinion but does not constitute recommendations made on the basis of a consensus of Member States.

This report does not address questions of responsibility, legal or otherwise, for acts or omissions on the part of any person.

Although great care has been taken to maintain the accuracy of information contained in this publication, neither the IAEA nor its Member States assume any responsibility for consequences which may arise from its use.

The use of particular designations of countries or territories does not imply any judgement by the publisher, the IAEA, as to the legal status of such countries or territories, of their authorities and institutions or of the delimitation of their boundaries.

The mention of names of specific companies or products (whether or not indicated as registered) does not imply any intention to infringe proprietary rights, nor should it be construed as an endorsement or recommendation on the part of the IAEA.

The authors are responsible for having obtained the necessary permission for the IAEA to reproduce, translate or use material from sources already protected by copyrights.

CONTENTS

1. INTRODUCTION

1.1. BACKGROUND

Most nuclear power plant instrumentation and control (I&C) systems were based on analogue technology when they were built in the 1960s and 1970s. In the 1980s and 1990s, computer based, programmable I&C systems were developed for introduction into new plants and to replace analogue technologies in existing plants. The new systems had many advantages compared to older analogue based solutions. However, the extensive functionality and resulting complexity of software made the verification and validation (V&V) of computer based I&C systems time consuming and expensive. In addition, computer based applications use microprocessors, which now have short product life cycles, and applications may become obsolete in a relatively short time due to lack of spare parts and support.

In order to address the above problems, the industry is beginning to move towards field programmable gate array (FPGA) based applications, as evidenced by implementations such as those described in Annex I to this report.

FPGA chips may be limited to simple and verifiable logic structures that can be interconnected to perform a desired application. The following are the most obvious advantages of this technology:

— Certain FPGA types cannot be configured on-line and are therefore inherently resistant to certain cybersecurity issues. This makes FPGAs attractive for safety applications, where on-line reconfigurations would be unacceptable.
— Design and V&V for FPGA based platforms for safety applications can be made much simpler and less costly than for microprocessor based ones.
— In general, FPGA vendors that provide product lines to industries requiring high reliability and long lifespans, such as aerospace, aircraft and military industries, tend to offer longer term support of the product lines than do microprocessor vendors.
— FPGA based applications are more resilient to hardware obsolescence due to the portability of the hardware description language (HDL) between different versions of FPGA chips produced by the same or different manufacturers.

FPGA technology has been applied in the satellite, military, aerospace, aircraft, transportation industries, and in other industries where the reliability of applications is of primary importance. Considering the benefits and favourable experience with the use of FPGAs in other industries, the nuclear power industry has started to introduce FPGAs in nuclear power plants for safety, safety related and non-safety applications.

In the late 2000s, experts agreed that FPGA technology was a viable option to be introduced for nuclear power plant applications. As a result, the Topical Group on Field Programmable Gate Array Applications in Nuclear Power Plants (TG-FAN) was formed to discuss issues related to the design, qualification, implementation, licensing and operation of FPGA based I&C applications in nuclear power plants.

The adoption of FPGAs for nuclear power plant I&C systems was also recommended to the IAEA by the Technical Working Group on Nuclear Power Plant Instrumentation and Control members. At the time of publication of this report, eight annual FPGA workshops, increasingly attended by plant owners, suppliers, regulators and researchers, have been held in various Member States with cooperation between the IAEA and the TG-FAN. These workshops provide specialists, users and regulators with an opportunity to share their experience in the field and understand the issues affecting the industry so that these can be taken into consideration when adopting FPGA based solutions. The intent is to continue with the events as long as the need to share the above knowledge remains.

The IAEA plays a central role in international discussions on technical applications, licensing and benefits of FPGA based solutions. This is the first IAEA report published for Member States on this new and rapidly growing technology, which can provide substantial benefits to the safety and reliability of nuclear power plants.

1.2. OBJECTIVE

The objective of this report is to describe current best practices and issues associated with the application of FPGA based solutions in nuclear power plants. The publication includes a description of the technology and current knowledge on development processes and tools. It also includes a discussion on advantages and challenges associated with the application of FPGAs, as well as licensing issues. The report is not intended to discuss the advantages and challenges of other digital solution alternatives.

The report is directed at all personnel in Member States involved in the design, manufacture, qualification, licensing, implementation and operation and maintenance (O&M) of FPGA based I&C systems and components in nuclear power plants. The following are foreseen as users of this publication:

— Nuclear power plant operators;
— Technical support organizations;
— Regulatory bodies;
— Research and development organizations;
— Manufacturers/vendors.

Guidance provided here, describing good practices, represents expert opinion but does not constitute recommendations made on the basis of a consensus of Member States.

1.3. SCOPE

This publication provides technical guidance as a basis to support the development and applications of FPGA based platforms in nuclear power plant I&C systems. It provides a description of FPGA technology, as well as the benefits and challenges experienced by the industry in the application of the technology, and general application characteristics.

In addition, the report includes a discussion on methods and tools for the development and licensing of FPGA based platforms. It provides specific information for FPGA applications in the complete or partial replacement of I&C systems. It also addresses FPGA applications in new nuclear power plants.

Various applications and plans by utilities to incorporate FPGA technology into their nuclear power plants are discussed in Annex I of this report.

1.4. STRUCTURE

This report contains six main sections. Section 1 introduces the topic and the objectives of the publication. Section 2 discusses the basics of FPGA technology, the place of FPGAs within the electronic hardware designs, and the advantages and challenges with the technology. Section 3 describes the methods and tools for the development and verification of FPGA designs. Section 4 discusses the specific licensing issues that emerge with FPGAs. Section 5 provides an overview of possible applications of FPGA technology for the replacement of I&C in existing nuclear power plants, as well as for I&C systems for new nuclear power plants. Section 6 gives a summary and the recommendations of this report.

Annex I gives some applications and planned examples with FPGA based replacement systems and new nuclear power plant designs, and Annex II introduces the typical life cycle for an FPGA platform.

Definitions of terminology in use, mostly specific to the development and application of FPGA technology, are provided in a glossary.

2. INTRODUCTION TO FIELD PROGRAMMABLE GATE ARRAY TECHNOLOGY

This section includes a description of FPGA technology, a comparison between the different variations of related products available on the market, an outline of their strengths and weaknesses, and design life cycle examples of FPGA applications. Also presented are advantages and challenges exhibited by FPGA technology compared to microprocessor technology for digital I&C system implementations.

This report addresses mainly FPGAs as one member of a family of devices that may be configured using HDL. Many of the concepts and processes explained in this report are applicable to other HDL programmed devices (HPDs), but the focus remains specifically on FPGAs. The main differences between some devices within the HDL programmable family are also covered in this section.

2.1. FIELD PROGRAMMABLE GATE ARRAYS WITHIN THE HARDWARE DESCRIPTION LANGUAGE FAMILY

HDL programmable devices are a type of large scale integrated circuit where the internal hardware architecture is configured for a specific application according to the user needs after production of the chip. The circuits are fabricated 'void' of any functionality, and are entirely configured (i.e. their logic is programmed into the device) for the given applications through the use of HDLs such as very high speed integrated circuit hardware description language (VHDL) and Verilog. VHDL and Verilog are the two standardized HDLs. HDLs are a way of representing the internal architecture and behaviour of elementary logic components within a device, as well as the links between these components.

Figure 1 illustrates where HDL programmable devices are placed compared to other electronic hardware technologies and the relative location of FPGAs within the HDL family of devices.

The figure also illustrates where FPGAs fit compared to other technologies used for similar applications.

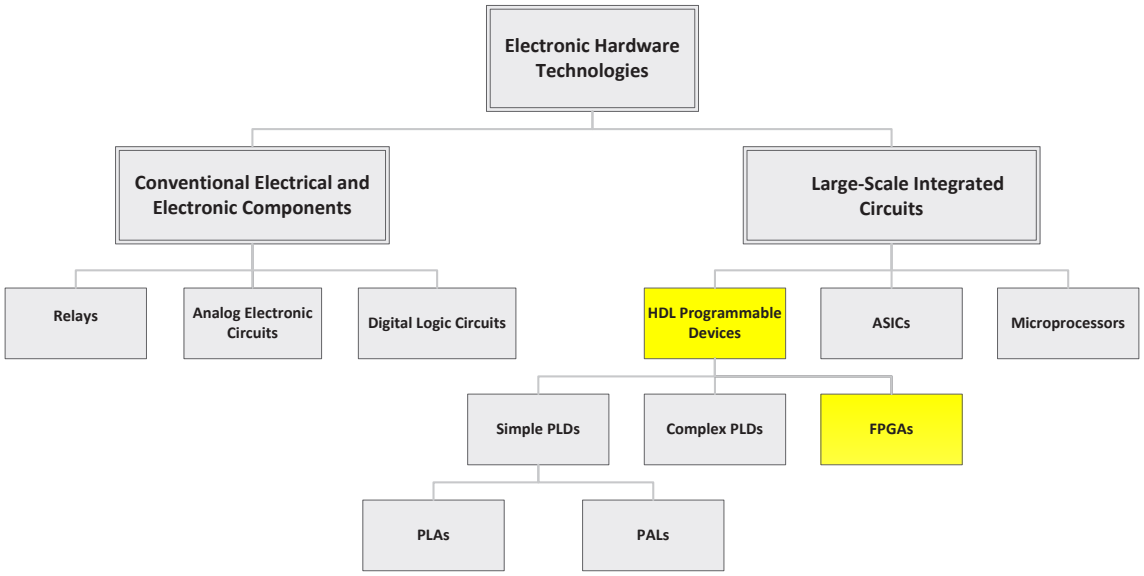


FIG. 1. Electronic hardware technologies [1]. ASIC — application specific integrated circuit; FPGA — field programmable gate array; HDL — hardware description language; PAL — programmable array logic; PLA — programmable logic array; PLD — programmable logic device.

On the left hand side of the diagram are conventional electrical and electronic components, including relays, analogue electronics and digital electronic components (e.g. transistor–transistor logic circuits). These technologies are becoming obsolete. The technologies on the right hand side of the diagram are presently the most commonly used ones for the replacement of old components and systems. With the exception of simple programmable logic devices (PLDs), these are based on the utilization of large scale integrated circuits. The following are brief descriptions of each of the items in this category:

— Microprocessor based equipment includes programmable logic controllers (PLCs) configured as single units or in distributed control system architectures.
— Application specific integrated circuits (ASICs) are custom designed and fabricated at the integrated circuit foundry for specific applications; hence, unlike FPGAs or complex programmable logic devices (CPLDs), they are not configurable after production. ASICs are designed from the outset for specific applications and therefore tend to be cost effective only when large numbers of devices are produced.
— HDL programmable devices contain arrays of logic elements that can be interconnected by users to perform functions required for specific applications. The first to become available were simple PLDs. These included programmable logic arrays and programmable array logic (PAL) devices. CPLDs evolved from PALs. They essentially combine multiple PALs on a single chip with configurable interconnections.

The difference between FPGAs and the other HDL programmable types described above is in their internal architectures. FPGA architectures allow greater functional capabilities.

HDLs use a register transfer level (RTL) design abstraction for modelling digital circuits. It is independent from the hardware implementation and represents only the flow of information and logical operations necessary for the required application.

A gate level description specific to a particular FPGA model can be derived through logic synthesis of an RTL representation. The results of logic synthesis are then transformed into a physical implementation for the target FPGA via ‘place and route’ tools (see Section 2.3.3 for more details), and eventually the configuration is loaded onto the FPGA (i.e. the FPGA is ‘programmed’).

2.2. DIFFERENCES BETWEEN HARDWARE DESCRIPTION LANGUAGE AND SOFTWARE

Differences between the development processes for software (as applicable to high level languages such as C) and HDL are as follows:

— Software is written in high level languages (source code), compiled and then assembled to generate binary code that is specific to the target microprocessor, which sequentially executes the code according to the principles of a Von Neumann architecture (data and instruction memory, arithmetic and logic processing registers, input and output channels, control units for managing interruptions and bus communications, etc.).
— HDL is also a high level language. However, instead of generating binary code to be sequentially executed, the RTL description (see Section 2.1) is translated by synthesis and place and route processes (see Section 2.3.3) into an FPGA specific bitstream, through which the chip is configured by physically connecting logic elements in compliance with the desired application. The above configuration results in parallel execution of all logically independent operations, rather than a typically serial process, as in the case of conventional microprocessor based platforms.
— The capabilities for formal methods of logic verification and systematic trajectory testing are features supported by several HDL development environments.

HDL implementations have some common features with hardware as follows:

— HDLs such as VHDL and Verilog are standardized, precise and formal languages for describing combinational and sequential logic circuits that can be realized in a physical hardware implementation on an FPGA.
— RTL code is a description of numerous programmed logic elements that may operate in parallel, without relying on operating system software to provide a link between the hardware and the application.

— The code generated during the development process from the RTL description is only used to configure the hardware resources of the FPGA (unlike binary code stored in memory and executed sequentially by a microprocessor).
— Provided simple configurations are used (e.g. no soft cores), runtime problems specific to software, such as task scheduling, microprocessor overloading, interruption sequences and exceptions, infinite loops, management of access to memory, illegal instructions, etc., are not present in HDL applications.
— Software based systems generally require an operating system. This introduces a complexity that is not directly related to the application. HDL based devices can be configured as purely hardware based, which allows a much closer equivalence between functional requirements and hardware implementation, including the possibility to simulate logical behaviour prior to burning the final application into the target device. (However, it should be noted that some leading regulators such as the US Nuclear Regulatory Commission (NRC) already require the FPGA development process to be treated as a form of 'software development process'.)

The operating principles of HDLs can be kept simple such that the realized design is fundamentally different from that of software based systems, which execute instructions sequentially.

On the other hand, there are similarities between HDL and software development processes to the extent that syntax or logic errors could result in implementation errors. Furthermore, as is the case with software, the use of syntax and type verification tools significantly reduces the risk of those systematic faults that these tools can detect — those that can be identified without any knowledge of the application.

Given that HDL implementations lead to the interconnection of hardware components, it becomes much easier to demonstrate the deterministic behaviour required for Category A safety applications (the highest safety category under the classification published by the International Electrotechnical Commission (IEC) in standard 61226 [2]). Very exhaustive systematic verification is possible with simulation tools.

Fault detection and fault isolation is another major benefit. It is much easier to predict the behaviour of a random hardware fault on a simple FPGA implementation and protect against any undesirable consequences (i.e. problems with design features) than on a software based platform.

2.3. WHAT ARE FIELD PROGRAMMABLE GATE ARRAYS?

FPGAs are integrated circuits within the broader family of HDL programmable devices, and are designed to be configured by the user, after manufacture, through the use of HDLs. FPGAs differ in their detail design among different vendors and product lines; however, they share a common basic architecture such as that illustrated in Fig. 2.

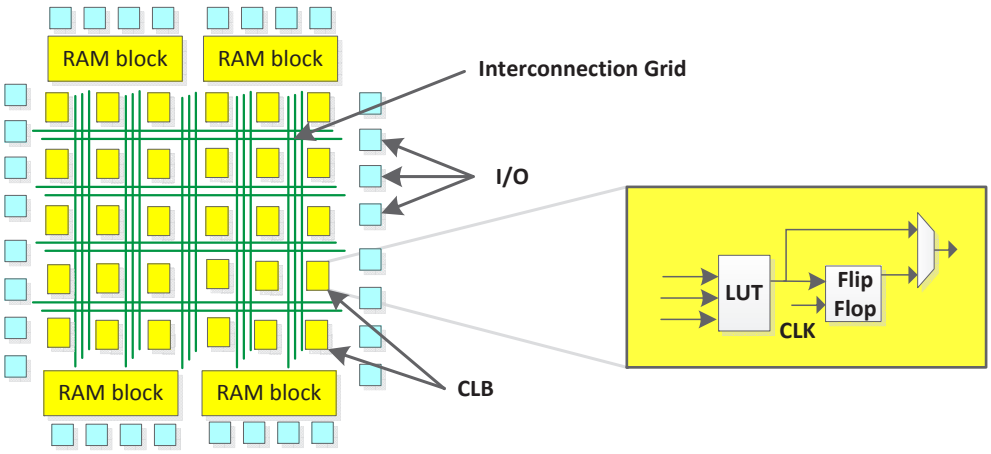


FIG. 2. Typical field programmable gate array architecture [1]. CLB: configurable logic block; Clk: clock; I/O: input/output; LUT: look-up table; RAM: random access memory.

This architecture includes:

— *A set of configurable logic blocks (CLBs)*. A CLB can be configured to implement any logic functions (e.g. AND, OR, XOR, NOT). Therefore, 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 are interconnected to generate more complex functions. The output of each CLB includes a flip-flop for synchronizing the data flow within the FPGA.
— *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 analogue to digital conversion.
— *An internal interconnection grid*. This is a set of wires to be interconnected at intersecting points when the FPGA is configured to the desired application. The above interconnections link different CLB inputs and outputs, as well as FPGA input and output blocks, in configurations representing the desired applications.
— *Application data memory*. Most FPGAs contain application dedicated memory. Some of these devices include static random access memory (SRAM), and others, when used for applications requiring fast power restart and/or resistance to single event upsets (SEUs), are equipped with non-volatile flash memory.
— *Some FPGA architectures contain additional elements to those mentioned above*. For example, some configurations include microprocessors linked to the CLBs through the interconnection grid. These elements will increase the complexity of the FPGA based systems, and their adoption should be carefully considered by designers with regard to the overall requirements of their applications.

2.3.1. Comparison between field programmable gate arrays and complex programmable logic devices

The routing capabilities in an FPGA allow for many more signal paths to be created than can be done by the crossbar routing technique used in CPLDs.

In addition, the above two technologies differ in the following:

— Because of their higher logic-to-interconnect ratio, CPLDs can generally yield a faster solution for simple applications. However, FPGAs offer much greater flexibility and larger design capacity.
— FPGAs offer the possibility to embed intellectual property (IP) cores, including microprocessors, to perform complex functions, whereas CPLDs typically do not.
— CPLD technology is based on a continuous interconnecting structure, which results in functional predictability and high performance, particularly with respect to signal propagation delays. FPGAs have segmented interconnect structures; the number of segments necessary to route signals is determined by the software tools during the place and route phase, and this affects signal propagation times. Therefore, for FPGAs, it is not possible to know signal propagation times until the design has been placed and routed.
— FPGA logic cells have finer granularity than CPLDs. As a result, for the same functionality, it takes more cells to implement a function in an FPGA than in a CPLD.

Table 1 provides a comparison of some key attributes of CPLDs and FPGAs.

TABLE 1. COMPARISON OF KEY ATTRIBUTES OF COMPLEX PROGRAMMABLE LOGIC DEVICES (CPLDs) AND FIELD PROGRAMMABLE GATE ARRAYS (FPGAs) [1]

Attribute	CPLD	FPGA
Configurable logic block	Logic array	Gate array
Density	<500 000 gates	>500 000 gates
Speed	Fast, predictable	Application and design dependent
Interconnect	Crossbar	Routing
Power consumption	High	Medium to low

2.3.2. Field programmable gate array related technologies

There are three main technologies applied for storing the configuration of the interconnection grid, CLBs and I/O blocks in an FPGA:

— *SRAM.* This is a rewritable device so it can be modified without physically replacing the FPGA component. Given that it is volatile memory, the configuration and data are not retained on loss of power. Also, a power glitch may alter the FPGA configuration (interconnection grid, CLBs and 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 electrically erasable programmable read only memory (EEPROM).* Flash and EEPROM technologies are rewritable and non-volatile. In both cases, the FPGA configuration is unaltered by power losses. Flash memory could be considered a modern version of EEPROM technology, and differences between these two types are mainly in the way they are reprogrammed (EEPROMs must be completely erased before reprogramming, whereas flash memory can be reprogrammed in blocks). The data retention capabilities of flash memories are guaranteed for a limited period of time (e.g. 10 years). This fact must be taken into consideration when designing and maintaining the system.
— *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, connections are created between logic blocks by means of heated conducting links, thus the name ‘antifuse’. The above process applies both to CLBs and to I/O block configurations. If the FPGA needs to be reconfigured, it will be necessary to physically replace the component. For this reason, antifuse devices are sometimes referred to as ‘one time programmable’ devices.

When FPGAs are programmed via external interfaces, it is important to remember that it is not the FPGA chip itself that is being configured, but a memory (in the case of SRAM, flash or EEPROM) that stores the configuration in the form of a gate level description and loads it onto the chip during power up. Only in the case of antifuse technology does the FPGA chip hardware configuration become permanent.

2.3.3. Field programmable gate array programming process

The FPGA programming process comprises four main development phases, as shown in the V-shaped diagram in Fig. 3.

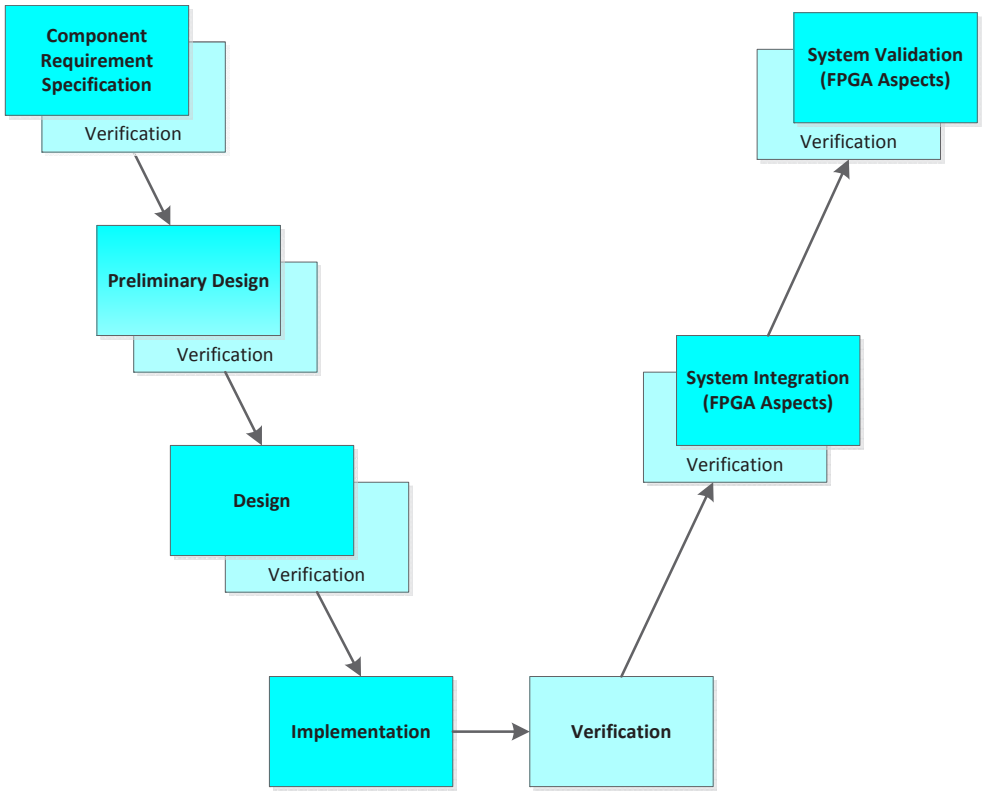


FIG. 3. V-shaped field programmable gate array (FPGA) programming model [1].

2.3.3.1. Component requirements specification

The objective of this phase is to systematically and precisely state all the requirements that apply to the final (i.e. programmed) FPGA circuit. These requirements usually result from decomposing the I&C system architectural design into components and allocating system functionality, safety/dependability and performance requirements to each component. This is a manual process; however, there are tools to help in the requirements traceability process.

2.3.3.2. 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, the decomposition into modules (application specific or predeveloped), the implementation of defensive design practices, the device families to be used, and the required library functions and IP cores. This is a manual process.

2.3.3.3. Design

The objective of the design phase is to develop a detailed description of the processing to be performed by the FPGA circuit (much like software source codes describe the processing to be performed by a microprocessor). This step is independent of the actual hardware, unless designers wish to take advantage of unique features offered by certain circuits. The design is usually expressed using HDL. FPGA based I&C platform vendors sometimes propose more application oriented languages, and tools that can translate these languages into HDL.

A design can be described at different levels of detail. The most common detailed design representation used today is the RTL. At this level, FPGA functions are described in terms of signal flows or data transfers between registers (flip-flops or other memory elements) and operations performed on those signals. Changes in registers are simultaneous and follow the ticking of a clock signal.

Testing of designs before they are configured into physical FPGAs is usually performed via simulation. RTL simulation is especially suited to the detection of logical and other functional errors. Testing for timing and other performance requirements usually requires more detailed information on the final FPGA circuit, and is performed during the implementation phase.

Formal verification techniques usually start with a formal specification of the design properties, using languages such as Property Specification Language. Formal verification tools can then systematically check whether these properties are indeed satisfied by the RTL design. Most tools can generate examples to help system designers locate errors.

Even though the design team could apply some or all of the verification techniques above, they would also typically be performed by an independent V&V team with expertise similar to that of the design team.

2.3.3.4. Implementation

The implementation phase starts upon completion of the design phase. Implementation is usually divided into two main steps: (i) synthesis and (ii) place and route. Both steps are normally supported by tools provided by the FPGA vendor.

The objective of the synthesis step is to translate the circuit independent RTL into an equivalent description expressed in terms of the resources available in the selected FPGA circuit. This circuit dependent description is called a 'netlist'.

Testing of the synthesis phase is mostly done by simulation. In addition to the netlist, logic synthesis tools generate timing files that can be used to simulate timings more accurately than is possible during the design phase. Formal verification techniques are also available to demonstrate that the netlist is consistent with the RTL.

The place and route tool calculates the best physical positions and mapping of the FPGA resources used. It aims to reduce inter CLB bottlenecks by distributing data transfers uniformly over the interconnection grid. The output of the place and route process depends on factors such as clock frequency and maximal signal set-up time, which is the maximum time available to prevent metastability. The place and route tool also generates a timing file that is more accurate than the one produced by synthesis, because it also includes timing associated with routing.

2.3.4. Field programmable gate array based system development life cycle

Whereas the life cycle described in Section 2.3.3 addresses the development of FPGA programming, the life cycle in this section determines and describes the processes, organization and documentation for the main development stages of the system for FPGA development. It should be based on applicable standards, but may be customized by taking into account the specifics of the system and FPGA applications development.

The development of an FPGA based system often uses a predeveloped platform, which is a set of components that can be combined into one or more architectures and configured to implement specific applications. The life cycle then mostly aims at designing the system architecture and programming of application logic in the FPGA circuit(s). This is generally supported by application oriented development tools and functional block libraries (see Fig. 4).

There are cases (e.g. for module replacements, where form factor constraints are important) when the development of an FPGA based system starts nearly 'from scratch'. The life cycle then needs to also include hardware design. Tools could be generic electronic design tools.

As different platforms may lead to very different system architectures, and because hardware electronic design is not within the scope of this publication, the rest of this subsection focuses mainly on the stages leading to FPGA requirements specification, with or without a platform.

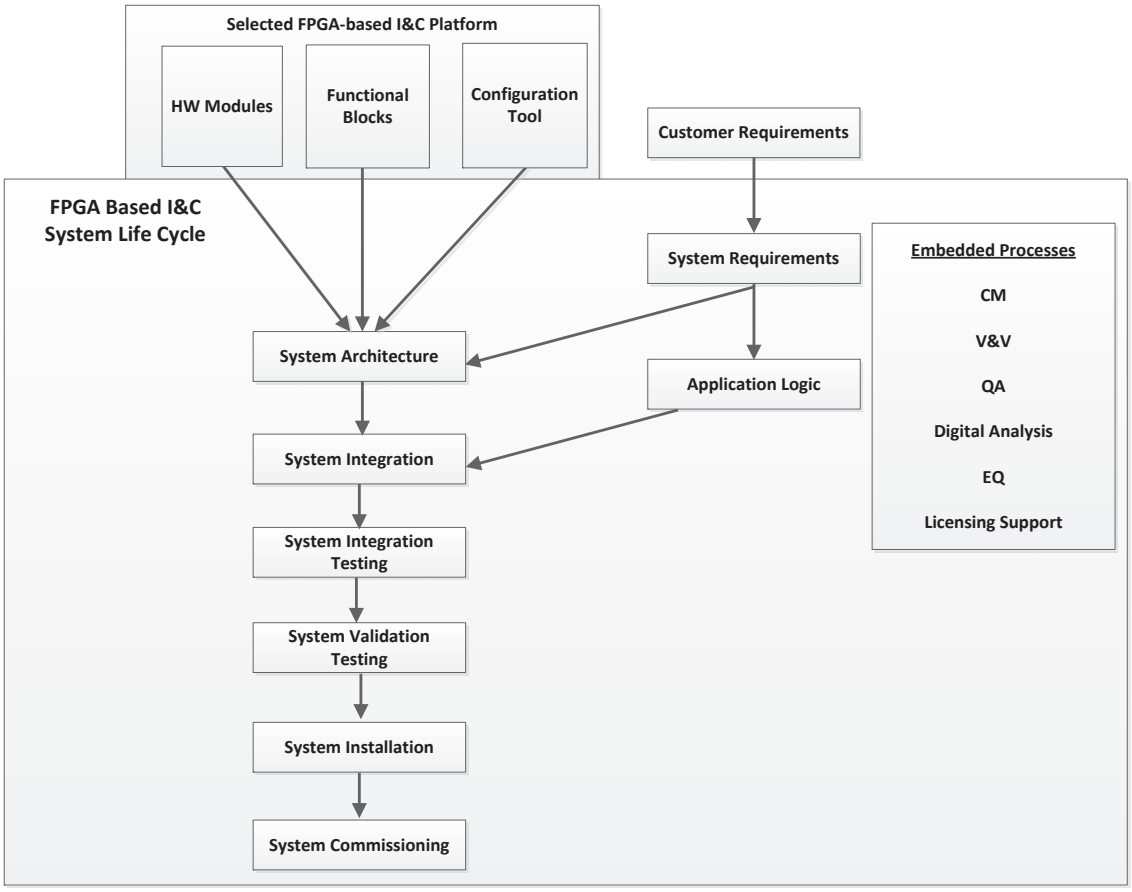


FIG. 4. Conceptual links between a field programmable gate array (FPGA) based platform and an FPGA based system life cycle. CM — configuration management; EQ — equipment qualification; HW — hardware; I&C — instrumentation and control; QA — quality assurance; V&V — verification and validation.

2.3.4.1. System requirements specification

This phase specifies, in particular:

— The boundaries of the system and interfaces with other systems;
— The functions to be performed by the system;
— The design constraints and non-functional requirements that must be satisfied;
— The interfaces with human operators;
— The environmental conditions.

How this phase is developed and reviewed is not within the scope of this report.

2.3.4.2. System architectural design

Based on system requirements and features of predeveloped items (including the platform, if any), this phase determines:

— The architecture of the system (i.e. its structure and organization into subsystems and modules, their interfaces and their interactions, and if some of the modules are FPGA based);
— The allocation of system requirements to subsystems and modules.

How this phase is developed and reviewed is not within the scope of this publication. The result of the phase is design documentation, which includes specification of:

— The system architecture;
— The subsystems and modules, in particular, the FPGA based modules;
— The internal interfaces and interactions;
— The allocation of system requirements to subsystems and modules.

2.3.4.3. Field programmable gate array requirements specification

This phase corresponds to the top left phase of Fig. 3 on the V-shaped FPGA programming life cycle. Based on system requirements and system architectural design, requirements are specified for each FPGA, including:

— Functions to be provided;
— Modes and transitions;
— I/O, interface and interaction requirements;
— Parameters that can be modified during operation;
— Performance requirements and restrictions;
— Assumptions regarding the FPGA environment;
— Fault detection and diagnostics requirements.

This phase would typically be performed by the system design team, who would have expertise in system and FPGA design.

Completeness and compliance of FPGA requirements with system requirements and system architecture are verified after the FPGA requirements specification phase. These would typically be performed by a separate team with similar expertise.

2.3.4.4. System integration and system integration testing

This phase and the following one correspond to the right hand side of the V-shaped life cycle in Fig. 3. System integration includes assembling newly developed and predeveloped components in a whole system and adjustment of all components and connections, if necessary. Verification at this phase is mainly performed using tests, which check internal and external system interfaces, as well as system functions that cannot be checked with the functional testing of the FPGA. Tests are generally performed on the basis of an integration test plan and test specification, and result in an integration test report.

2.3.4.5. System validation

Validation checks compliance of the whole system with the initial system requirements specification. It is generally performed on the basis of a validation test plan and test specification, and results in a validation test report.

2.4. GENERAL APPLICATION AREAS SUITED TO FIELD PROGRAMMABLE GATE ARRAY BASED IMPLEMENTATIONS

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, of the order of tens of milliseconds or longer. The functional and performance capabilities offered by current FPGA circuits feature from tens of thousands to millions of gates, operating with clock frequencies of the order of hundreds of megahertz, which are more than adequate to support such functions.

FPGAs are able to meet most safety and non-safety system requirements associated with I&C applications in nuclear power plants, and many of the advantages exhibited by FPGAs discussed in this publication in terms of safety applications are also applicable to non-safety applications. When these are not obvious, they will be specifically mentioned under the appropriate subsections.

The implementation of FPGA based safety and non-safety related applications in operating and new plants is expected to grow substantially. Reduced complexity and the ability to segregate safety from non-safety functions make FPGA based safety applications attractive, especially in the area of acceptance. Licensing costs and risks are reduced because inherently simpler designs can be achieved, and the safety properties can be demonstrated (in a systematically verifiable manner) to the regulator. An Electric Power Research Institute (EPRI) report [3] describes several safety applications, and notes their potential for reducing licensing risks and costs. The applications were obtained from nine operating plants and six planned new plants in different countries. Some of the applications are also described in Annex I.

The implementation of primary reactor protection functions was identified as the most important expected FPGA application in existing plants. Other applications of interest include diverse actuation systems, emergency diesel generator controls and load sequencers, and post-accident monitoring systems. The functions considered to be most suitable for FPGAs included process related discrete functions (e.g. protective logic functions, bistables, comparators, coincidence logic, priority logic) and data communication functions. FPGAs have already been adopted in some designs as a suitable diverse technology where the primary or diverse actuation would be performed via a microprocessor or other technology, thereby minimizing common mode failures.

Although most of the considered applications were related to safety, FPGA technology is also suitable for a wide range of non-safety related applications. Interest has also been expressed in using FPGAs for relatively simple human system interface (HSI) functions.

There are some functions, including a few in boiling water reactor (BWR) safety systems, that require response times as short as 20 ms, and that cannot be implemented easily with most microprocessor based PLCs. Owing to their ability to support simple hardware only based logic and their parallel logic processing capabilities, FPGA based I&C systems may be optimized for high speed performance and made suitable for applications requiring very short response times (of the order of a few milliseconds to tens of milliseconds). Examples of these could be the regional overpower protection in Canada deuterium–uranium (CANDU) reactors and BWR safety systems.

The replacement of analogue boards performing complex functions and taking substantial cabinet space with considerable intercard wiring constitutes another example for which FPGA technology could be most suitable. With an FPGA based replacement, the functions originally implemented via analogue technology could be grouped into a single FPGA card, thereby tremendously reducing the number of parts. This would reduce the number of boards and eliminate intercard wiring without the need for a multipurpose, microprocessor based machine. The result can be increased reliability, as well as reduced installation and maintenance activities. Any resulting freed slots could be used for additional functionality or, where necessary, the addition of redundant or diverse capabilities in order to further increase the system fault tolerance and reliability.

FPGAs can be used economically and reliably to ensure that systems of lower safety importance do not affect the ability of more important systems to perform their safety functions. For example, FPGA based solutions can be used to implement priority logic, which ensures that important safety signals receive priority in controlling plant equipment, so that erroneous signals from less important systems will not block or defeat a safety function.

Modern I&C architectures rely heavily on data communication between I&C systems and subsystems (e.g. between redundant channels or divisions). This could be a source of common cause failures (CCFs) propagating through communication links. Dedicated communication devices, such as filters and gateways, can be used to mitigate this risk. However, commercially available data communication devices are not normally developed with safety applications in mind, and the qualification process for them, where possible, could demand considerable effort. FPGA applications can be designed as simple interfaces that implement only what is required and do not rely on interrupts or handshaking, thereby reducing the risk of failure propagation and simplifying the safety case.

Most cybersecurity guidelines recommend that protection measures follow a defence-in-depth approach with multiple layers of defence. Qualified FPGA based gateways and filters placed between layers of defence could be part of the solution. Gateways and filters can be implemented directly into the FPGA without additional capabilities, whereas a microprocessor based implementation would also have basic software, operating systems and potential unused functionality.

Antifuse based FPGA technology is easier to qualify for harsh radiation environments. This would allow the introduction of controllers inside nuclear power plant containment, thereby reducing wiring for cases where final elements, sensors and/or actuators are located inside the containment. This is an application that would be difficult to achieve using microprocessor based technology.

Owing to their capability to emulate or replicate microprocessor functionality using soft core IP technology, FPGAs could constitute a feasible way to replace or provide a diverse alternative to microprocessor based systems. One utility has developed a soft core emulator, which has been shown to be a good replacement solution for obsolete microprocessors. In addition to being a solution to the lack of spare parts, FPGA based emulators are cost effective compared to the costs of replacing the entire system, with its associated wiring changes, procedure changes, training, etc.

FPGA technology is well suited to module for module replacements. This is a very effective method to replace obsolete modules, to remove single points of vulnerability and to provide additional functionality. This approach is cost effective compared to the cost of replacing the entire system. An example of this can be seen at utilities that are replacing circuit cards with FPGA based circuit cards.

FPGA technology is suitable for applications requiring low power, complex signal sensing and processing, and potentially strong encryption solutions, because these applications suit the strengths of FPGA technology. One example of such an application is low power sensing and wireless data transmission. This application requires that the equipment run on batteries for long periods of time, monitoring plant performance during post-accident scenarios, when normal power sources are not available.

Designers and users, of course, would have to perform the pertinent cost–benefit analysis to help them in their decision to emulate, resort to a different technology at the module level, or undertake a major system refurbishment using FPGA or another replacement technology.

It should always be remembered that when a new technology is implemented in the plant, the operating and/or maintenance procedures will need to be changed. New training related to the new technology, its operating characteristics and its maintenance requirements will be required. At the same time, a new technology creates new opportunities for improvements that should be considered when applications using the technology are designed and implemented.

Examples of FPGA applications are given in Annex I to this report.

2.5. ADVANTAGES AND CHALLENGES OF FIELD PROGRAMMABLE GATE ARRAY BASED INSTRUMENTATION AND CONTROL SYSTEMS

This section includes a brief description of the advantages of FPGA technology, mainly compared with microprocessor based solutions, and the challenges that practitioners and users should anticipate dealing with when considering adopting FPGA technology.

2.5.1. Advantages

2.5.1.1. Lower complexity

Microprocessor based systems usually include a significant amount of relatively elaborate software such as real operating systems and application programmes. Depending on the application, this results in rather complex design, V&V and licensing processes. By avoiding the need for operating systems and other software, FPGA based systems can be made simpler and easier to test and qualify for safety applications.

Some of the advantages exhibited by FPGAs can be lost or significantly reduced as greater complexity is introduced by way of microprocessor based cores, operating systems and natively embedded self-testing and diagnostics. This added complexity in the design thereby affects the reliability assessment and safety case.

Figure 5 shows a comparison of how complexity increases with functionality for conventional hardware, FPGAs and microprocessor based systems. The figure includes three different FPGA architectures: FPGAs with flat hardware logic and FPGAs with an embedded microprocessor, with and without an operating system.

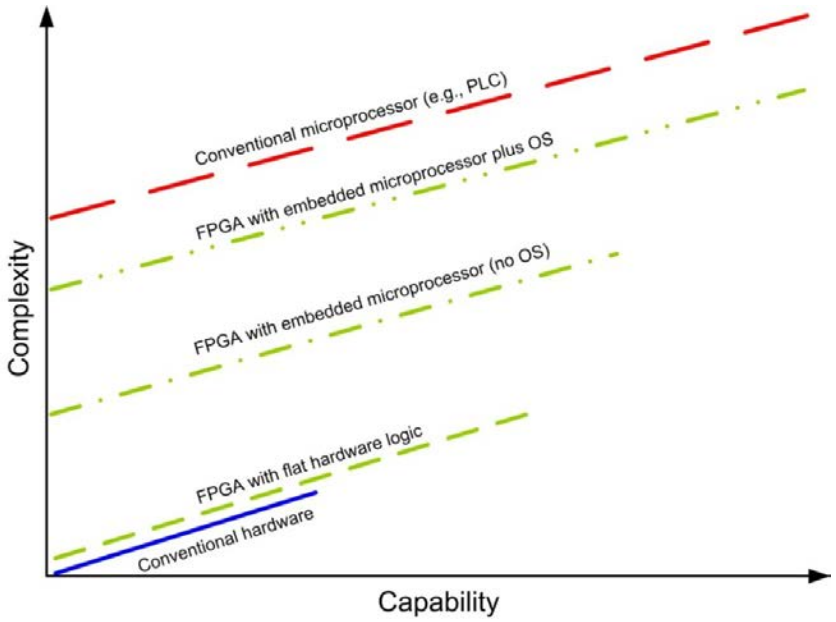


FIG. 5. Complexity versus capability for different instrumentation and control technologies [1]. FPGA — field programmable gate array; OS — operating system; PLC — programmable logic controller.

2.5.1.2. Greater application portability

Rapid obsolescence of electronic components results in comprehensive refurbishments of I&C systems during the lifetime of the nuclear power plant, and vendors can seldom guarantee full upward compatibility during the long periods between refurbishments. FPGA technology allows a significant degree of design portability, i.e. only the final steps (synthesis and place and route) are dependent on the particular FPGA circuit chosen. This mitigates their vulnerability to obsolescence.

2.5.1.3. Longer lasting availability of technical support

Portability of HDL code helps to achieve hardware independent applications. This is expected to result in available technical support throughout the lifetime of the plant. It also helps to substantially reduce the cost of porting an existing application onto a different FPGA platform that will be available in the future.

2.5.1.4. Faster response times

FPGAs can process logic calculations in a parallel fashion at high clock speeds and are therefore suitable for applications that require very short response times, including those not achievable with microprocessors.

2.5.1.5. Simpler verification and validation processes

As opposed to microprocessor based systems, which are mostly general purpose machines, FPGA based systems can be designed to include only the required functionality, and thus there are no hidden functions that might remain either untested or express themselves in unpredicted ways under certain machine states. This results in lower complexity and simpler V&V efforts.

2.5.1.6. Segregation of safety and non-safety functions

FPGAs have the ability to process separate functions independently and in parallel on the same integrated circuit (provided that these functions are logically independent and that the application does not call for the use of on-board random access memory).

Support functions, such as self-monitoring and diagnostics, and other non-safety functions may be physically separated from the main safety functions, either by ensuring that no signal flows from them to the main safety functions, or by placing them in separate FPGA chips altogether. This results in:

— Simpler design and validation of safety related applications;
— Assurance that failures of support and other non-safety functions will not cause loss of safety functions and vice versa, leading to overall higher system reliability;
— Easier determination of failure modes so that they are designed out, thereby further increasing the reliability, potentially decreasing the need for periodic testing and simplifying diagnostic functions;
— Easier methods for designers, testers and verifiers to apply more rigour to the design processes and testing of safety functions than to other functions, thus simplifying and lowering design complexity and development and testing costs.

Functional separation can be relatively simple to achieve because of the possibility of designing FPGA systems with minimal or no shared resources.

2.5.1.7. Reduced vulnerability to cybersecurity attacks

Cybersecurity in programmable systems is always a concern, regardless of the adopted technology. FPGA based implementations tend to increase the level of difficulty that would be faced by a potential attacker for the following reasons:

— They can be designed not to include high level, general purpose components that could be easily tampered with.
— Where cybersecurity is a potential concern, the FPGA designs can be based on antifuse (one time programmable) type chips. Alternatively, the system design can be such that physical access is required in order to modify the programming of the FPGAs.

2.5.1.8. Stronger technology of choice when required to comply with diversity requirements

FPGA technology could constitute a viable option for diversity between primary and redundant safety functions. Redundant microprocessor and FPGA based systems must, as a minimum, meet the diversity criteria in NUREG/CR-6303 [4] as follows:

— Design diversity (different technologies and architecture);
— Equipment diversity;
— Functional diversity (different ways to achieve the same result);
— Software diversity (different programming languages, design methodologies and software architecture).

2.5.1.9. Greater cost effectiveness

The following main factors contribute to making FPGA technology an attractive option from a cost perspective:

— Lower design complexity and simpler V&V processes result in shorter development and change implementation times.
— Simpler and more effective safety justification and/or dependability assessment can be a very significant contributor to cost reduction.
— The ability to carry out module for module replacements can, in some cases, eliminate the need for a full system replacement, thus reducing the cost. The same can be said for component and part replacements.
— Additional functionality on a module, such as self-monitoring and self-diagnostics, can increase the reliability and reduce surveillance and testing costs.

— The design capability to integrate in one integrated circuit what originally required multiple boards in the current digital system architecture reduces the number of hardware components and associated interconnections, which can have a favourable impact on cost and reliability.
— The intrinsic robustness to cyber-attacks results in simpler measures to protect against malicious tampering with the system.
— The portability of applications results in significant cost savings by avoiding the need to maintain and store large quantities of obsolete equipment to ensure availability of parts throughout the lifetime of the plant.
— FPGAs 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. Furthermore, the heat load for FPGAs is lower, potentially reducing cooling system requirements and cost.
— Implementations designed to improve the reliability of existing systems are much easier and less costly. For example, a printed circuit card, module, rack or entire system replacement could be designed for increased reliability by eliminating single points of vulnerability and adding suitable redundancy while maintaining the same footprint as the equipment to be replaced. This can lead to the reduced likelihood of power reductions and plant outages.
— While maintaining the same footprint, FPGA technology lends itself to the addition of capabilities to reduce O&M costs. For example, self-monitoring and self-diagnostic functions can be segregated from the application functions. These diagnostic and self-test capabilities help reduce and simplify surveillance and periodic testing, thereby reducing plant operating costs.

2.5.1.10. Suitability for a wide range of applications

FPGAs are well suited for systems, components and parts. This allows more cost effective replacements, when applicable, than full system replacements.

In addition to replacement of major systems or the design of I&C solutions for new or existing nuclear power plants, FPGA technology could be adopted to implement changes to a variety of applications, such as ‘form, fit and function’ (FFF) replacement of electronic components, reverse engineering and emulation of microprocessor based applications.

2.5.1.11. Other advantages

Other advantages of FPGA technology include:

— Flexibility to accommodate design changes;
— Considerable market support (parts and service);
— Considerable diagnostic capabilities (including built-in test features);
— Reduced footprint and cabling costs;
— Possibility of simulations and optimization of functionality;
— Potential use to easily prototype solutions that could later be implemented in different technologies.

2.5.2. Challenges

2.5.2.1. Relatively few current applications in the nuclear power industry

Although FPGAs have been widely used in various other industries and in consumer products for decades, they are still very new in nuclear power plants. There is not a wealth of nuclear power industry operating experience on which to build a reliable, meaningful database for gaining lessons learned and good practices, or to support the bases for acceptance or reliability numbers. It is possible to find many FPGAs and CPLDs in digital I&C systems already in operation in nuclear power plants, but they are typically buried in the design of electronic circuit boards and do not play as prominent a role as the microprocessor and its software in a PLC based system. This poses challenges and risks in the safety analyses and licensing efforts for utilities and designers.

There is currently only one standard, published by the International Electrotechnical Commission (IEC), that provides guidance and requirements for FPGA based solutions for the nuclear power industry (IEC 62566 [5]); to date, this standard has not been adopted by most regulatory bodies.

2.5.2.2. Limited availability of products and tools

Currently, there is only a limited number of FPGA based I&C platforms and products available and ready to use for nuclear power plant applications. Although sound development methods, languages and tools are available on the market, they have not reached the level of user friendliness and acceptance by developers reached by their microprocessor based counterparts, such as for PLCs. There are also questions about the transparency of these tools.

2.5.2.3. Proprietary intellectual property cores are less transparent

Vendors offer libraries of IP cores to end users as a means of faster development and a way of securing continued business. An IP core, which is a reusable unit of logic, cell design or chip layout design, can complicate the acceptance process when it is used in the application. As the IP core represents the IP of the vendor, its internal behaviour is often not available for the customer to use for interactions with the regulator. In addition, the customer has no control over changing the IP core.

2.5.2.4. Decreased access to internal signals for monitoring, testing and troubleshooting

Compared to conventional electronics or microprocessor based solutions, an FPGA based solution can result in less observability and less access to signals within the functional logic. Therefore, analysis should be performed during the design phase to be able to provide access to important signals for activities such as monitoring, testing and troubleshooting (see Section 3.1.8). In many cases, this is accomplished through the use of a test access port in the FPGA chip, but there are no guarantees that signals whose accessibility was missed during the design phase will be accessible to users through this access port. This requires extra effort on the part of the designer and solid communications with those who will be responsible for maintaining the system during plant operation.

2.5.2.5. Accommodating graphic and complex human system interface functions

FPGAs are very well suited for data processing functions involving the mathematical calculations and image processing associated with control room displays and other HSIs. However, without the use of soft core central processing unit (CPU) emulation, they are not the best solution for very complex processing such as that required for graphical interfaces, menu systems and windows based interfaces that allow selection of different means of information display, soft controls, alarm filtering and management, and procedure management.

In addition, complex HSIs tend to require more frequent modifications than control algorithms. For such complex functions, these types of modifications are presently easier to implement in software.

2.5.2.6. Need for specialized expertise (hardware and software) on the design team

Modern FPGA tools provide the capability to develop applications from workstations by connecting functional blocks into an integrated design, simulating the system functionality, verifying and validating the design, and implementing it in the target FPGA hardware. The design can also be changed and reverified relatively easily using the provided toolset. As the tools have become easier to use, properly trained specialists can use them to design and implement applications. FPGA designs are currently represented at the RTL as ‘code’ written in an HDL, and the design implementation is accomplished through the 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, and the design process is quite similar to software design processes, including associated V&V activities performed at successive stages of design development. Similar to conventional software based systems, the design team needs to have special coding expertise and an understanding of the 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 designing an FPGA based application essentially means designing the internal configuration of an integrated circuit. There are hardware design issues that must be addressed properly in order to ensure a reliable and safe initial design, and to ensure that reliability and safety are maintained when any changes are made to the design. Examples of hardware design issues include ground bounce, occurring as multiple outputs change state simultaneously, timing glitches, which can occur if propagation delays internal to the circuit are not handled correctly, problems with multiple clock domains causing timing issues, etc.

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.

3. METHODS AND TOOLS FOR DEVELOPMENT AND VERIFICATION

3.1. DESIGN GUIDELINES

This section provides design guidelines that are relevant for FPGA based solutions. The guidelines may be applied during implementation of the life cycle for the FPGA based platforms and systems presented in Section 2.3.4 above. These recommendations concern mostly the design of the FPGA circuits that are important to the correct implementation of the I&C functions important to safety.

3.1.1. Requirements specifications

Producing appropriate requirements specifications is a key issue for all highly critical systems (FPGA based or not), as any error will most likely be propagated into the design and implementation of the actual system. In particular, requirements specifications need to be complete, unambiguous, consistent and functionally validated.

Requirements specifications for FPGA based systems present few or no fundamental differences with other types of systems (e.g. section 5 of IAEA NS-G-1.1 [6]). However, because FPGAs are often considered for module for module replacements, a few specific guidelines might be worthy of mention:

— As equipment installed decades ago may lack precise documentation, particularly at the module level, reverse engineering may be necessary to specify the new module requirements.
— In some cases, it is worthwhile that the reverse engineering also determines ‘why’ the time performance and/or the behaviour of the legacy module are the way they are. In some cases, they are so just as consequences of the initial design and implementation, not because it was a functional requirement.
— More accurate, faster replacement modules are not always beneficial, as they might, for example, react to noise spikes that would have been rightly filtered out and ignored by the legacy modules.
— Module for module does not necessarily mean FFF. Improvements could be provided in terms of self-monitoring, fault signalling, fault tolerance, testability and reliability.
— Lessons learned from operating and maintaining the legacy equipment could be taken into consideration when specifying the replacement modules.

The requirements applicable to the FPGA circuit itself need to address topics that include:

— Functional and timing requirements, which are the ‘raison d’être’ (reason for existence) of the module;
— Electrical and logical interfaces;
— Environmental conditions and corresponding qualification requirements;
— Quality of service requirements, covering subjects such as reliability, failure modes that are preferred or that are to be avoided, fault tolerance, applicable standards and regulatory requirements;
— O&M requirements, including observability of internal states, testability, modification of application parameters and failure signalling;

— Other constraints, such as portability of application designs, acceptable technologies, constraints related to diversity requirements, and design provisions, if any, to accommodate future changes.

3.1.2. Synchronous design

In an FPGA design, a synchronous block is a design component in which the internal registers and outputs are modified simultaneously, at times defined by a clock signal. A synchronous circuit is a circuit in which the blocks 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'.

Synchronous design is one means to achieve deterministic behaviour. It also helps to provide modularity and clarity of the design. It is highly recommended, particularly for safety applications and for applications critical to plant performance, that FPGA designs be made synchronous whenever possible. When different clock domains really need to exchange data with one another, appropriate measures should be applied. The main reason is that so called 'clock domain crossings', where data are transmitted from one clock domain to another, could lead to non-deterministic behaviour and therefore require extreme care.

If asynchronous design is used, an analysis of all paths should be performed to demonstrate that the outputs comply with the requirements specification and that no adverse glitches or metastability will result.

More information and guidance on synchronous design are provided in EPRI TR-1019181 [1], EPRI TR-1022983 [3], IEC 62566 [5] and NUREG/CR-7006 [7].

3.1.3. Predeveloped designs or hardware description language modules

Predeveloped design components for logic functions or general purpose interfaces can be integrated into the complete design. Such designs can come in different forms, as discussed below.

Native blocks represent the pre-existing resources in the FPGA chip, for example, an OR gate or a more complex function such as a multiplier or a serial transmission controller. The FPGA design configures and connects the native blocks to provide the required function(s). The integrated electronic design environment provided by the circuit vendor can also automatically translate particular high level functions into a lower level description, relying on native blocks. Design information on native blocks is often limited. This is not an issue with simple functions (such as OR or an LUT). For more complex functions, verification limited to black box testing could raise licensing issues for applications important to safety. Therefore, it might be preferable to avoid complex native blocks for these applications.

IPs are design components that can be obtained independently from the FPGA circuit. There are two main types of IPs:

— Soft IP cores are provided in a format independent of any FPGA circuit, usually in an assessable format (e.g. RTL) that allows extensive verification.
— Hard IP cores are provided in a format that allows their use only with a given circuit or circuit family. They are often offered by the circuit vendor, or by IP vendors who want to protect their IP. Thus, they are provided in a format that limits verification to black box testing.

As for native blocks, IPs with limited transparency may be difficult to verify and maintain in applications important to safety. Use of such IPs in applications important to safety should be avoided; otherwise, additional software and hardware verification of the IP core itself needs to be performed [7]. A common approach is to consider a basic HDL library of 'safety logic elements' that is formally and systematically verified and reused across multiple projects.

3.1.4. Design/coding rules

It is generally a good practice to apply predefined design and coding rules in the development of complex systems. As such rules need to take into consideration many specific issues such as the coding language, the design

method, the tools, the predeveloped components or the nature of the system to be developed, it is not the objective of this publication to provide specific rules. However, a few topics to be addressed are listed below:

— *Naming conventions*. As complex designs or codes can introduce large numbers of object names, the objective of naming conventions is to help readers in remembering what each name refers to.
— *Presentation conventions*. The objective here is to facilitate legibility by applying text or graphical layout rules.
— *Avoidance of language constructs/method features*. Languages and methods may contain constructs and features that increase their expressive power but that come at the expense of proper structure or that could be misunderstood by many designers.
— *Complexity limits, testability and verifiability*. Rules may specify complexity limits in pieces of code or design artefacts, or may specify ways to facilitate the use of verification tools.
— *Documentation and commenting.* The objective here is to provide additional information on the intentions of the designers, or explanations on non-obvious design choices.
— *Use of predeveloped components*. Rules may put restrictions on, or require, the use of such components, or specify how such components may be used.
— *Tool independence and portability*. Rules may be specified to ensure that design artefacts and code do not unnecessarily depend on specific tools.
— *Design/coding templates*. To enhance legibility and facilitate verification and future modification, templates addressing specific design issues may be provided.

The correct application of the rules needs to be verified and enforced.

3.1.5. Fault tolerance and self-monitoring

FPGA designs raise a specific reliability issue compared to microprocessor based designs. The normal functioning of a microprocessor tends to use a very significant part (if not all) of its electronic circuitry. Thus, a random hardware error that could affect the behaviour of the microprocessor is unlikely to remain undetected until a demand situation occurs. In the case of FPGAs, however, if no appropriate design measures are taken, parts of the electronic circuitry might remain inactive until a demand situation occurs. As a result, a random hardware error could remain undetected and cause a failure on demand.

To limit the likelihood of such events, several approaches (discussed in the rest of this subsection) might be considered, for example:

— Internal redundancy;
— Active self-monitoring;
— Passive self-monitoring;
— Self-healing.

Internal redundancy and cross checking may be used to detect incorrect processing due to random errors in a single channel. There are several ways to implement on-chip redundancy (see Fig. 6).

Some electronic development environments have a redundancy option to triplicate each elementary circuit with a two out of three vote. This allows the chip to tolerate multiple hardware errors, as long as no triplicated elementary circuit is affected by more than one error. However, if voting discrepancies are not signalled, errors could accumulate, unknown to operators and maintenance staff (even after periodic testing), until a real failure occurs. Another issue that may need to be taken into consideration is the fact that this approach utilizes a significant amount of chip resources. In addition, the voting logic itself could be subject to failure (‘quis custodiet ipsos custodes?’ (who watches the watchmen?)). Lastly, it might be worthwhile to verify that the triplication has been performed correctly by the development environment.

Another way to implement on-chip redundancy is to explicitly include, in the design, multiple copies of the circuitry implementing the required functions, with the associated voting logic. This would usually be done at a higher functional level such as is the case for automatic use of triple redundancy. It might be necessary to prevent automatic ‘optimization’ by the electronic development environment because this may result in the removal of

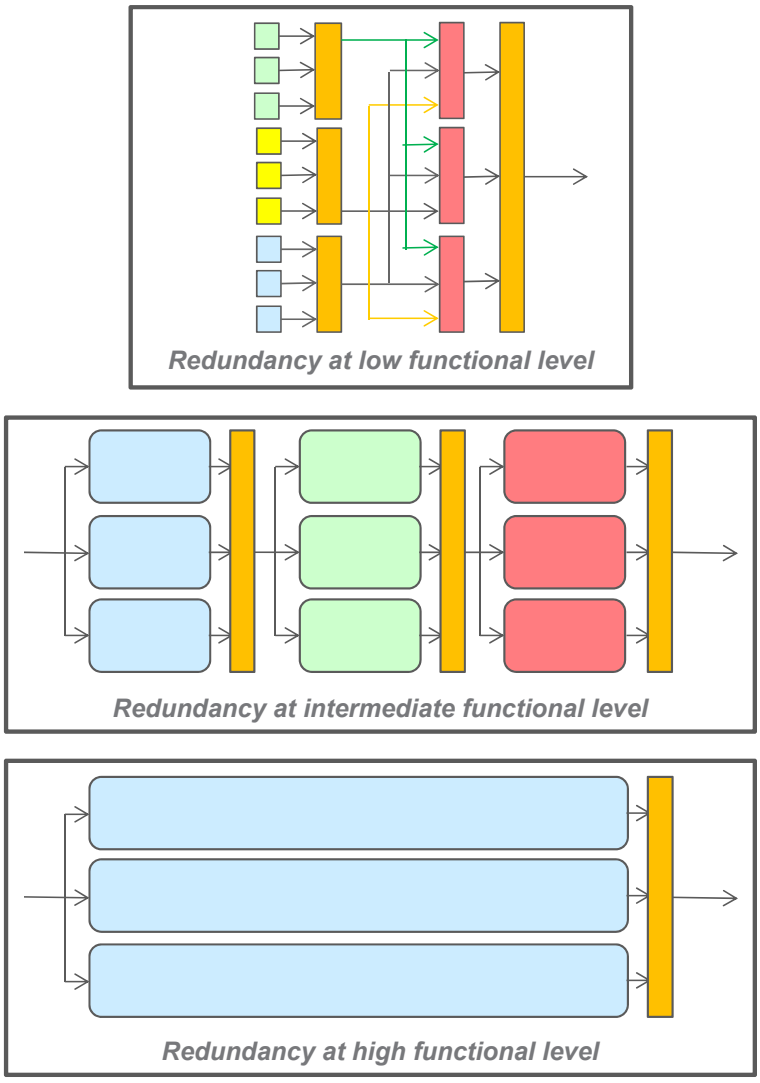


FIG. 6. Various internal redundancy schemes at different functional levels.

redundant circuits. This type of redundancy is usually less tolerant to multiple errors, but as the voting logic is determined by the application designer, voting discrepancies can be signalled immediately and systematically. The application designer can also choose the number of redundant copies that are needed, based on a probabilistic analysis, for example. Finally, the application designer can usually instruct the place and route tools to place each redundant copy in a separate area of the chip in order to limit the risk of a single radiation event affecting multiple copies.

It should be noted that on-chip redundancy does not necessarily eliminate the need for redundancy at higher levels in the I&C architecture. (As off-chip redundancy is not specific to FPGA designs, it is not further discussed here.)

Active self-monitoring follows a very different approach, where the FPGA chip (or parts of it) alternates between two operation modes:

— Normal mode, where the chip provides the required outputs at the designated pins;
— Test mode, where field inputs are substituted by test inputs, and corresponding outputs are masked (i.e. not provided at the chip pins) and checked against expected values.

One benefit of this approach is that a random hardware error that could affect outputs is likely to be detected and signalled very rapidly, whereas in the case of redundancy, it might be detected only in a demand situation. The drawback is that it can add significant complexity to the design. It could also put some time performance

constraints on the design and the FPGA circuit. The effects of hardware errors affecting the active self-monitoring circuitry itself (the size of which could be significant) need to be assessed, as they might, in the case of inadequate design, lead to failure on demand.

Contrary to active self-monitoring, passive self-monitoring does not disrupt the operation of the circuit. It includes techniques such as:

— Runtime checking of design assertions and/or assumptions, which may be used against both random and systematic errors;
— Parity checks or cyclic redundancy checks (CRCs), which may detect random data corruption;
— Range and plausibility checks (of the values of key signals, key registers, internal states, etc.), which may be used to detect data that are obviously incorrect.

These techniques have a more limited scope than internal redundancy and active self-monitoring, but they can generally be implemented by so called 'observers', i.e. parts of the circuit design that are electrically prevented from interfering with the main functions of the circuit in case they are themselves affected by random or systematic errors.

Whereas the above mentioned measures provide some protection against permanent and transient random faults, self-healing provides protection mainly against transient random faults, i.e. faults that would disappear if the circuit is reinitialized. Its main principle is to ensure that the design has no long term memory, so that any transient fault that does not cause an immediate failure would be rapidly 'forgotten' by the normal operation of the circuit.

The need for such measures should be determined based on several factors, including:

— The circuit basic failure rate, which is often provided by the circuit vendor in terms of failure in time (FIT). Not all circuits are equivalent, and FIT rates vary significantly.
— The reliability targets required of the system and its FPGAs.
— The periodic testing intervals.

3.1.6. Reliability analysis

Various failure rate factors may need to be estimated, including the following:

— Probability of failure on demand due to a design error;
— Probability of failure on demand due to a random hardware error;
— Frequency of spurious actions due to a design error;
— Frequency of spurious actions due to a random hardware error.

Principles for estimating failure rates due to random hardware errors could be determined and justified based on:

— The FIT rates for the elementary blocks within the FPGA circuit. It might be worthwhile to distinguish between transient and permanent faults, between faults affecting a single elementary block and those affecting multiple neighbouring blocks, between different types of elementary blocks (registers, logic blocks, I/O and external interface blocks, internal and external interconnection blocks, memory holding the FPGA programming, etc.). It is also worthwhile to distinguish between the failure modes of the blocks.
— The effects of faults in each elementary block. Depending on which elementary block(s) is/are affected, on the failure mode(s) of the block(s) and on the FPGA design itself, the detection and signalling of the error could be immediate, or delayed until a periodic test or a demand situation occurs. If immediate, and in a redundant configuration, there may be no functional consequences or spurious actions. If delayed, there could be functional consequences or failure on demand.

Particular considerations need to be taken into account for cases such as the automatic triplication of elementary circuits, where several non-signalled faults need to accumulate before there is a spurious action, failure on demand or detection by periodic testing.

Failure rates due to design errors are notoriously difficult to determine and justify for high quality, highly reliable digital systems and devices, and no generally accepted method is currently available. As these factors cannot usually be ignored, analysts may follow approaches such as:

— Conformance to international standards, for example, IEC 61508 [8];
— Reliability growth;
— Operating experience;
— Fault injection;
— Statistical testing [9].

Each approach has its own limitations, but might provide useful insights in estimating failure rates due to design errors. Additional guidance, including guidance on estimating the rates of CCF due to design errors, may be obtained from EPRI TR-1021077 [10].

3.1.7. Diversity

FPGA based solutions are sometimes used to provide diversity within an overall I&C architecture, in particular, with respect to microprocessor based PLCs. However, when performing a diversity analysis for this type of application, it might be worthwhile to clarify points such as:

— Does the PLC platform contain FPGAs (e.g. in I/O or data communication boards) that are identical to, or of the same family as, the FPGA design being considered for diversification?
— If the FPGA being considered for diversification embeds a microprocessor, can a design error in that embedded microprocessor or its software adversely affect the safety or safety related functions assigned to the FPGA based system?

Diversity is sometimes provided within the FPGA based system itself. As it may have a cost in terms of complexity and risk of spurious actuation, the need for such diversity should be determined taking into consideration the diversity that already exists between systems. Different approaches can be taken for internal diversity, such as:

— Different FPGA circuits providing either identical functions, or different functions serving the same overall purposes. The degree of difference between circuits could be assessed taking into consideration factors such as the designs of the naked, unprogrammed circuits, including:
  • The natively integrated blocks;
  • Their electronic development environments;
  • The circuit programming technologies (e.g. SRAM, flash or antifuse);
  • The basic circuit technologies (e.g. the width of electrical connections).
— Identical FPGA circuits providing either identical functions implemented differently, or different functions serving the same overall purposes.
— A single FPGA circuit providing either identical functions implemented differently, or different functions serving the same overall purposes. In this case, it is worthwhile to ensure that the functions do not interfere with one another to the point that the failure of one could cause the failure of the other. In addition, separating the functions in separate areas of the chip helps to limit the risk of a single radiation event affecting the two functions concurrently.

Design diversity should be associated with appropriate segregation and independence, so that the failure of one function due to a design error will not adversely affect the diverse implementation of that function. This is particularly important for the case of design diversity implemented within a single FPGA circuit, where independence of the two implementations is more difficult to demonstrate than in the case where two separate and diverse circuits are used (EPRI TR-1022983 [3]).

3.1.8. Testability and observability

During development and V&V, simulation techniques may be used to determine, with reasonable accuracy, the behaviour of the FPGA design and the evolution of internal signals and states in different functional conditions.

On-line observation of internal signals and states may be used for monitoring and diagnostics. Off-line observation may be used for testing or troubleshooting during application development. Off-line observation may also be used to ensure that the manufactured and configured chip complies with the verified and validated design, and, in particular, that all specified logic blocks are present, connected and operating correctly.

The design should identify which signals and states need to be observed and then take appropriate measures to make them logically accessible, for example, through the Joint Test Action Group (JTAG) interface. JTAG is the commonly used name in the Institute of Electrical and Electronics Engineers (IEEE) standard 1149.1-2013 [11]. Although it was originally designed for testing printed circuit board (PCB) assemblies, today JTAG is also used for accessing internal elements of integrated circuits and to provide design for testability capability. Measures should be taken to prevent unauthorized interference with the circuit through the test and observation ports (including, but not limited, to JTAG).

3.1.9. Cybersecurity

Cybersecurity has become a critical issue, and extensive guidance is being developed. This section addresses only what is specific to FPGAs.

FPGA based I&C systems can be designed to be less vulnerable to cyber-attacks than microprocessor based I&C systems. This is possible because:

— Particularly when using FPGA circuits with only very simple native blocks and simple, soft IP cores, any capability not necessary to the application can be eliminated from the design. The soft IP cores used could also be verified to contain no hidden and no unnecessary capabilities. Thus, contrary to many microprocessor based designs where predeveloped software tends to be significantly more complex and difficult to analyse, malicious attackers will find many fewer capabilities that they could ‘hijack’.
— The system and its modules may be designed so that any reprogramming of the FPGAs would require physical access to the system. This would displace the cybersecurity issue to physical security, which needs to be ensured anyway.

This does not exempt FPGA based system designers and operators from following the general cybersecurity recommendations regarding design, manufacturing, installation on-site and O&M.

3.1.10. Obsolescence management

Similar to most I&C systems and devices, the lifetime of FPGA based I&C systems and devices is unlikely to cover the 60 year (or more) lifetime of the nuclear power plant itself. However, two main approaches may be used to reduce the issues raised by obsolescence. The first one aims at ensuring a long lifetime for FPGA based I&C systems and devices. The second one aims at facilitating their replacement when they do become obsolete.

Extended lifetime mainly relies on careful design choices. Some aspects are as follows:

— Durability in operation of modern electronic circuits is, in part, determined by feature size; the smaller the feature, the more sensitive the device is to ageing mechanisms such as electromigration (where the metal atoms of the interconnecting wires are physically displaced by the high intensity electric currents). The mass market electronics industry aims at ever decreasing feature sizes (currently, a few tens of nanometres) in order to improve speed, density and functional capability to the point where the expected lifetime of a modern electronic chip is only a few years. Fortunately, a number of FPGA vendors have chosen to target industrial and military markets, where high performance and reduced costs are secondary to robustness, durability in operation and long commercial lifetime. These vendors typically have products with feature sizes in excess of a hundred nanometres.

— Durability in operation of modern electronic circuits might be reduced by adverse operating conditions such as high temperature, electric overvoltage or physical manipulation (e.g. for maintenance or periodic testing).
— The FPGA programming technology can also have an impact on obsolescence. Non-reprogrammable technologies (such as antifuse) require a physical replacement of the FPGA chip if programming needs to be modified; this needs to be taken into consideration when estimating the necessary amounts of spares.
— Obsolescence contracts may be established with circuit vendors so that the user is alerted sufficiently in advance should the vendor decide to discontinue its product, or make any significant change in the manufacturing.
— The FPGA design may embed fault tolerance and diagnostic measures, allowing the circuit to remain in operation, even with a few hardware faults.

Facilitated replacement also relies on particular design choices such as:

— *Appropriate documentation of requirements, design, interfaces and implementation*. FPGA based systems and FPGA designs are no different to any other system on this point.
— *Portability of FPGA applications*. The FPGA design should preferably be such that should the FPGA circuit need to be replaced due to obsolescence, the design down to the RTL (circuit independent level) could be kept, and only the implementation phase (circuit dependent level) and hardware qualification would need to be redone. In this regard, it is usually preferable to choose circuits that do not embed native blocks that are too specific and that would be unlikely to be found in circuits in the future.

3.1.11. Field programmable gate array selection

In the selection of an FPGA circuit, the following criteria should be considered:

— *Functionality*. The circuit should provide the resources required by the system design.
— *Programming technology (SRAM, flash, EEPROM and antifuse)*. This determines how programme modifications are introduced in the system; it can also influence the lifetime of the chip programming, i.e. the time duration after which the chip should be reprogrammed to ensure that it has not been spuriously modified by radiation events.
— *Feature size.* This has a significant impact on chip durability (see Section 3.1.10). In addition, smaller feature sizes often lead to increased sensitivity to radiation events, with a single event affecting multiple blocks.
— *Failure rates*. As noted in Section 3.1.5, vendors can generally provide the FIT rates of their circuit families, including under different radiation conditions.
— *Complexity*. Some FPGA circuits have natively embedded features, such as configurable microprocessor cores, adjustable cache sizes, multipliers, dividers, floating point operations, accelerators or analogue to digital convertors. Although some may be useful, they may also raise concerns with regard to safety and obsolescence. For applications important to safety, the safety assessment of such features should be taken into consideration (EPRI TR-1019181 [1]). During the selection process, it should be ensured that sufficient information on the FPGA chip is available regarding design, verification, manufacturing and operating experience, in order to perform the assessment.
— *Software tools*. FPGA circuit vendors provide software tools that contain specific information on their products (e.g. on native blocks or timings). The capabilities, extent, quality and ease of use of these tools constitute key factors when choosing a circuit.
— *Circuit vendor*. Confidence in the vendor's ability to collect operating experience and to address design, manufacturing, support, quality, cybersecurity and long term planning issues should be considered. It should also be noted that the definitions of a long term timescale are different for an FPGA circuit vendor and an I&C system. It is about 5 years for the vendor, and about 30 years for typical plant systems. This difference highlights the necessity to establish a strategy for covering risks related to the discontinuation of the chip manufacturing and toolset support relevant to the I&C system. This might be an agreement with the vendor for long term support of the selected FPGA circuit.

— *Operating experience*. Analysis of operating experience may provide additional arguments for the FPGA circuit selection and acceptance process, especially when the type of application and the operating environment are similar. The problem with operating experience is that, sometimes, it may be difficult to obtain enough information and reliable data to perform an accurate evaluation.

3.1.12. Complexity

Designers should use the opportunity offered by FPGAs to make their designs as simple and modular as reasonably possible. Various means are at the designers' disposal, some of which have already been mentioned in the preceding sections:

— Choose circuits with only simple native blocks.
— Choose only simple soft IP cores (note: a common practice is to limit use of soft cores or only allow the use of precertified and very simple soft cores for Category A functions).
— Design independent functions implemented in the same chip to also be electrically independent. In particular, self-monitoring functions could be implemented in the form of electrically independent observers.
— Avoid putting too many functions in the same chip (unless it provides a clear design benefit).
— Avoid over-designing fault tolerance and self-monitoring functions.

3.2. VERIFICATION AND VALIDATION

3.2.1. Environment of the field programmable gate array circuit

All the V&V techniques described below are applied to particular representations (RTL, synthesis results, place and route results) of the design to be verified. The techniques need the same level of representation of the environment that will be provided to the FPGA circuit, for example, in terms of inputs, clock signals, loads and power supplies. Any assumptions made at the interface between the circuit and its environment need to be stated and/or represented, including (but not limited to) electrical/electronic aspects, clock skews and electrical loads.

3.2.2. Simulation

Simulation is the most commonly used V&V technique for FPGA design. Its objective is to determine the behaviour of a design (or part of it) on a workstation that emulates the hardware circuit. It can be performed at various levels of detail and accuracy, and at various stages of the life cycle. Simulation of the RTL can typically be performed during the logical design phase and is mostly useful for detecting logical errors. Detection of timing errors usually needs more detailed information on the hardware circuit and is typically performed at various stages of the implementation phase, such as synthesis, and then place and route, which is often also done by simulation.

Most (if not all) FPGA circuit vendors provide powerful integrated electronic design environments that feature advanced simulation capabilities and embed detailed timing information for their circuits. Indeed, they consider such environments to be a necessary complement without which their circuits could not be used effectively.

There are several significant benefits of simulation, such as:

— It can be used early in the life cycle, even before the actual hardware circuit is selected.
— It allows states and signals internal to the design to be observed and/or controlled. In particular, it allows verification techniques such as fault injection, which are extremely difficult to implement on an actual hardware circuit, where internal signals and states may not be observable unless measures are taken in the design to make them accessible.
— It can take into account the variability, in particular, in timings, of actual hardware circuits, and determine behaviours in best and worst case timings.

There are also some limitations of simulation, such as:

— It is a testing technique, and, as such, it allows verification of the FPGA design only for the cases that are being tested. Therefore, it needs to be used in association with functional and/or structural test coverage criteria or other justification approaches to ensure that testing has been sufficient. Depending on the rigour of these criteria, simulation might be very time consuming.
— The correctness and accuracy of timings are dependent on adequate specifications of the best and worst case timings for the hardware circuit being considered.

3.2.3. Test coverage

The rigour of the test coverage criteria to be applied is usually determined based on the importance (to safety or to production) of the functions implemented. Tools are available to help the V&V staff generate test cases to satisfy given coverage criteria. Other tools are available to help the V&V staff analyse simulation results, in particular to determine whether they are consistent with expected results, and whether they are different from previous results (in the case of regression testing).

FPGA designs are sometimes sufficiently simple that a 100% testing approach can be contemplated. However, there are multiple interpretations of what 100% testing means. One of them is testing all possible combinations of input values. If the FPGA design is not stateless, then the internal state registers and their possible values need to be treated like, and in combination with, the inputs in the determination of the test cases.

Various arguments, in particular logical separation arguments, may be taken into consideration in order to reduce the number of test cases. For example, if the FPGA design implements two functions that cannot electrically interfere with one another, then one might argue that they can be 100% tested separately.

It is apparent from the above discussion that in order to justify a 100% testing approach, certain structural properties of the FPGA design need to be justified, such as systematic identification or lack of internal state registers, or logical and electric separation of functions, possibly using code inspection or formal verification.

Note that the 100% testing approach mostly addresses the functional aspects and does not necessarily cover 100% of the purely electrical/electronic aspects, which then need to be dealt with separately.

3.2.4. Formal verification

The objective of formal verification is to ascertain that the FPGA design has a claimed property, based on systematic (and often mathematically based) reasoning. Various techniques are available, and the selection of technique(s) to be applied will usually depend on the property that is claimed. Examples of such properties are:

— Functional and timing properties, which express constraints on the outputs (or on internal signals and states) in relationship with the inputs and other signals provided to the FPGA circuit by its environment. The electronics industry has developed languages such as the Property Specification Language (IEEE 1850-2010 [12]) or SystemVerilog assertions (IEEE 1800-2009 [13]) to specify functional and timing properties.
— Structural properties, which result from specific measures important to the safety justification of the design. For example, when the FPGA design includes main functions (which are the raison d'être of the circuit) and ancillary functions (such as self-monitoring functions), it is good practice to design the ancillary functions so that they do not adversely interfere with the main functions.
— Integrity properties, which claim freedom from particular types of so called 'intrinsic errors'. (An intrinsic error is an error that can be recognized without any knowledge of the functional and timing requirements.)
— Equivalence properties, which claim that a representation A of the design is equivalent to another representation B for a given functional, timing, structural or integrity property. Typically, the property has been verified on B, and A is an implementation of B.

A number of powerful formal verification methods, tools and environments have been developed for the benefit of the electronics industry, where design and manufacturing costs and time-to-market constraints are such that formal verification is a significant competitive advantage. Methods include:

— *Model checking*. This systematically explores all possible states, and verifies that the property to be ascertained is always satisfied. One of its advantages is that if this is not the case, the method can provide a counterexample that could help the designer to understand why, where and when the violation occurs. Its main limit is that combinatorial explosion may overwhelm even a major computational infrastructure.
— *Static timing analysis*. This computes the expected timing of combinatorial logic, without requiring simulation.
— *Design rules checking*. This verifies that specified rules have been adhered to.
— *Assertion based formal verification*. This checks that an assertion, i.e. a statement about a required specific functional characteristic or property, is indeed true for the design. Assertions may be expressed on the design's external interfaces, or on particular aspects of the design's internal behaviour.

Formal verification has several benefits, such as:

— Contrary to testing and simulation, formal verification is a systematic approach. If correctly implemented, conclusive and successful, it ascertains that the property is true for all (not just for some specific) cases.
— Formal verification can address properties that cannot be verified, or are difficult to verify, by testing or simulation.

It has also a number of limitations, such as:

— Even though their typical scale and complexity are significantly lower than those of the rest of the electronics industry, with current tools, FPGA designs to be used in nuclear power plants cannot always be formally verified in a single step using a single tool. Thus, a verification strategy needs to be developed to take into account the specifics of each design. This often requires a high level of expertise in the use and limitations of the tools, in addition to a good knowledge of the design to be verified.
— As for simulation, the formal verification tools operating on the implementation level of the FPGA design are dependent on an adequate specification of the best and worst case timings for the hardware circuit being considered. In addition, formal verification tools are highly complex and cutting edge technology, and their pedigree is often difficult to justify.
— Formal verification applies to a representation of the FPGA circuit, not to the FPGA circuit itself in its environment. Therefore, testing with the real hardware is still necessary (see Section 3.2.5).

When the top property claimed is a functional and timing property, and when the formal verification tool capabilities are exceeded due to the complexity of the FPGA design, the verification strategy can sometimes be based on the time honoured 'divide et impera' (divide and conquer) principle, where the design is partitioned into smaller, simpler components, each having well defined interfaces and interactions (structural property). Some of these components contribute to the implementation of the function, others do not. For the former, the strategy would be: (a) assuming that each component correctly performs its own function(s), to verify that the whole correctly implements the top function(s) (functional and timing properties); (b) to verify that each component correctly performs its own function(s) (functional and timing properties). For the latter, the strategy would be to verify that these components do not interfere with the components implementing the top function (structural property). Finally, the strategy would entail verifying that the final netlist resulting from the last implementation stage is equivalent, for all these properties, with the design representation used in the previous steps.

The capabilities of formal verification tools may be exceeded for reasons other than design complexity, for example, due to the variability of the interactions of the FPGA circuit with its environment (as is the case for a microprocessor emulator), or due to the comparatively long times needed to perform the top function (as is the case for a sequencing function spanning several minutes or more).

In some cases, the formal verification of functional properties needs to overcome an additional difficulty related to how functional requirements are stated. A typical example is the computation of floating point mathematical functions such as the sine function. In reality, FPGAs (and all other digital devices and systems) do not compute sine functions, but use an approximation based on a finite power series and approximate coefficients. In this case, the verification needs to be performed in two steps: (a) verify (non-formally) that the algorithm used is appropriate for the function to be performed and (b) verify (formally) that the design correctly implements the algorithm.

It is apparent from the above discussion that formal verification can sometimes be a complicated and error prone endeavour. Therefore, it is preferable that the verification is well documented (including all assumptions made) so that it can be scrutinized, and if necessary repeated, by an independent party, much like a mathematical proof.

3.2.5. Field programmable gate array hardware testing

Neither simulation nor formal verification can address all electric/electronic aspects. There is also a need to verify that the physical programming of the actual FPGA chip has indeed been performed correctly. Therefore, testing still needs to be performed with the real FPGA hardware, in its real electric/electronic environment. Testing may concentrate on what has not been fully verified by the previously applied V&V techniques, but it may be limited by the reduced ability to control input signals and observe internal states.

Hardware tools, in addition to software tools, are often needed in order to inject, collect and analyse electric signals.

3.2.6. Failure analysis

Failure analysis at the level of an FPGA circuit may address different issues, such as:

— Verification that when it fails, the FPGA circuit does so in the specified failure modes, particularly when some failure modes can have very undesirable effects;
— Verification that the design measures taken and claimed to prevent or mitigate specific failure modes or failure mechanisms have been correctly implemented;
— Assessment of the effectiveness of the measures taken for fault tolerance and self-monitoring (see Section 3.1.5).

The behaviour of the circuit in the presence of random faults affecting the hardware, either transiently or permanently, can be determined by a combination of simulation and fault injection. Code and design inspection or formal verification may also be useful.

3.2.7. Gradation of verification and validation measures

For FPGA designs intended to support functions important to safety, developers are expected to apply rigorous V&V measures to both the I&C platform and the application specific parts of the system. Examples of such measures include independent V&V, use of V&V tools that are independent from those used for the design, simulation and testing to specified coverage criteria at each main step of the life cycle, and formal verification. The rigour and nature of the measures may depend on national regulatory requirements and practices.

For FPGA designs that are not important to safety but that are important to plant performance, developers might decide to apply all or part of these measures, considering that in some cases, the financial cost of a failure might far exceed the cost of the measures.

3.3. TOOLS

Software tools play an important role in the overall development process of FPGA based applications. They may increase the integrity of the FPGA electronic design and have many benefits in design and V&V. The FPGA specific tools are divided into three main categories:

— *Application development tools*. These are used to develop and verify the logic design of an FPGA, and can, to a large extent, be independent from the choice of FPGA circuit. One of their main outputs is a logical design in an RTL format. Application oriented tools may facilitate this work by allowing the logic design to be expressed in terms related to the application domain. FPGA based platforms often include such tools, together with predeveloped functional block libraries.
— *Implementation tools*. These perform the synthesis and place and route, and the low level, time accurate simulation and analyses. As they are dependent on the FPGA circuit chosen and contain circuit specific information and characteristics, they are usually supplied by the FPGA vendor.
— *Independent verification tools*. For applications important to safety or to plant performance, one can perform independent V&V and use different verification methods and tools, such as static analysis and formal verification. For such applications, availability of independent verification tools should be evaluated (EPRI TR-1019181 [1]).

3.3.1. Tool quality

The software tools used for the development and V&V of FPGA designs are usually highly complex. In addition, they need to evolve at a rapid pace in order to meet the ever changing and increasing demands of the electronics industry. Therefore, there is a risk that they could generate incorrect designs or mislead V&V staff. Several approaches may be used in order to reduce the likelihood of undesirable effects, such as:

— Verifying that the candidate tools are developed according to suitable software quality assurance. Certification to appropriate standards by credible bodies could facilitate this task.
— Verifying that the candidate tools have already been used successfully by a number of other users.
— Designing the V&V plan to also detect incorrect results from generation tools.
— Checking that the candidate V&V tools detect known errors injected in test cases.
— Analysis of the error reports for the candidate tools. The fact that the tool vendors maintain report lists, make them available to their users and have traceable corrective actions is usually a favourable factor. In addition, the tool users may put in place specific measures to circumvent the reported errors in order to avoid them or mitigate their adverse effects.

3.3.2. Tool integration

Upstream life cycle activities, such as requirements specification, system level design, electronic board design or HDL code generation from high level, and graphical design specifications often need to rely on external tools not provided by circuit vendors. This is also usually the case for transverse life cycle activities such as configuration management, traceability management and documentation management. However, it is highly desirable that appropriate interfaces and interactions are set and maintained between these tools and the integrated electronic development environment in order to ensure appropriate coordination and to limit human errors.

The issue of compatibility between tools used to implement some support activities for the development of the FPGA design, such as configuration and change management, requirements traceability, PCB development, etc., should be considered. During the design process and after the final product is released, tools should be maintained under configuration control and change control.

In addition, when the V&V plan relies on verification tools that are not integrated in the vendor's integrated electronic development environment, it is important to check that these tools use the necessary information in the right format and from the right source. For example, particular types of verification, most notably those relying on information specific to the hardware circuit such as timings, rely on information provided in the form of annotations in the generated synthesis or place and route.

Unfortunately, integrated toolsets do not always export all necessary information, thereby possibly limiting the use of independent tools.

3.3.3. Tool cybersecurity

FPGA circuits and applications can be designed to provide strong cybersecurity defences (see Section 3.1.9). However, the design, implementation and V&V tools could be vulnerable to cyber-attacks and thus need appropriate vulnerability analysis and protection measures. Thus, it is recommended to take measures to ensure that throughout the life cycle and operational lifetime of the FPGA based system, the toolset is not compromised in ways that could introduce errors or malicious code in the FPGA design, implementation or physical programming, or miss errors that could otherwise be detected.

3.3.4. Tool life cycles

The usually long operational lifetimes of I&C systems in nuclear power plants mean that measures need to be taken in order to ensure that the necessary tools remain available and operable. This includes (but is not limited to) measures concerning the hardware infrastructure that is necessary to execute the software tools, tool documentation, training, support, and version and configuration management.

As most of these tools are subject to commercial competition pressures, they are subject to frequent changes. The decision to use a newer version needs to take into consideration several, sometimes contradictory, factors such as:

— The need for support from the tool vendors, as they often tend to provide limited support (if any) for older versions;
— The compatibility of the new tool versions with FPGA designs in operation in the plant;
— The quality and credibility of the new tool versions.

4. LICENSING

As with any system or component in nuclear power plants, the licensing of FPGA based digital systems or digital systems using PLDs is an important part of the design, development, implementation and use. The introduction of FPGAs into nuclear power plants has not been without challenges. Although their use in non-safety applications has been fairly straightforward, the licensing effort associated with the use of FPGAs in safety systems has been more problematic.

Some licensees and vendors treated FPGAs as purely hardwired logic in early discussions with national regulatory bodies. This was not acceptable due to the very different design processes, tools and ranges of applications of the two technologies. These different views, as well as the lack of specific regulations and regulatory guidance for FPGA based systems for nuclear power plant applications in the past, has slowed the implementation of FPGAs into safety systems.

As a result of the above, the development of standard IEC 62566 [5] was initiated in 2007. The development of this standard involved the active participation of regulators, utilities, I&C designers and platform providers from many countries, including, but not limited to, Canada, Finland, France Germany, Japan, Republic of Korea, Ukraine, United Kingdom and United States of America. These Member States provided valuable input about the scientific grounds, the technical issues and the licensing aspects of FPGA technology. IEC 62566 [5] was approved in 2011 and published in 2012. Many national regulatory agencies are now reviewing the standard for use in the regulatory review of FPGA based digital safety systems.

This section covers the licensing of FPGA based systems. While aspects related to the development process (life cycle, principle of independent V&V, etc.) that have been successfully used to support software based systems translate well to FPGA based systems, aspects related to the products themselves (e.g. how they are designed) are different.

Licensing of FPGA based systems includes two aspects. The first is environmental qualification of the components and systems for nuclear applications. This includes electromagnetic and radiofrequency interference, system environmental testing for the location where they will be used, component ageing, etc. The second aspect of licensing is the evaluation of functional correctness and completeness, which includes all evidence needed to demonstrate that the system will perform as required under all normal, abnormal and accident conditions. Licensing activities are generally quite different from country to country, and will only be discussed here in general terms.

4.1. ENVIRONMENTAL QUALIFICATION

The national regulatory body will need to review the environmental qualification of FPGA based components and systems to determine whether the equipment will be able to operate within the specified environment over the design lifetime of the system. This includes both the normal and the worst conceivable abnormal operating conditions expected in the given equipment location, and all events the equipment is credited to mitigate.

The equipment is tested by subjecting it to a wide range of parameters, such as temperature, humidity, seismic load, vibration, radiation and electromagnetic compatibility/interference (EMC)/(EMI). The documentation provided by the licensee to the national regulatory body (e.g. equipment qualification (EQ) requirements, test plans, methodologies and test reports) should include sufficient information to support the claim that the proposed digital I&C system is adequately robust to perform its safety function within its design basis under normal and abnormal environmental conditions.

While type testing is the preferred method for microprocessor based systems, it may not be adequate for FPGA based systems, because their susceptibility to EMC/EMI, temperature, humidity and pressure depends on their final configuration.

After an FPGA reconfiguration, licensees should demonstrate that prior EQ remains valid by testing, inspection, analysis or a combination of the above.

When considering an FPGA device's timing, for example, it should be recognized that its performance varies based on the device's environmental conditions and its configuration. In other words, for a given FPGA programme, its timing characteristics under high temperature, low input voltage and high humidity will vary from those under low temperature, high input voltage and low humidity. Therefore, an FPGA programming change can subtly alter timing, and the FPGA's timing can subtly change under the range of environmental conditions that the safety functions must remain operable for; hence, qualification that includes FPGA programming changes made using documented processes under an acceptable quality assurance programme should include an analysis that determines whether or not the prior qualification remains valid and bounding.

For pure hardware only changes, the demonstration that a prior EQ remains valid and bounding is sometimes referred to as 'extension analysis'. For software changes, 'regression analysis' is a normal part of most change processes, and this analysis typically results in targeted retesting. Software regression testing rarely includes repeating portions of EQ such as seismic or environmental qualification. For FPGA based systems, design changes and maintenance may change the versions from those used during EQ. If the licensees or vendor use type testing for environmental qualification, additional testing and/or analysis will be needed to address the possible differences in the response of the FPGA based systems in different configurations to different environmental conditions. This might require requalification or incremental qualification after reprogramming the FPGAs.

Documented processes should include analysis appropriate to the FPGA technology, and recognize that the analysis results could lead to targeted retests similar to regression testing performed for computer software changes.

4.2. FUNCTIONAL DEMONSTRATION

4.2.1. General

Functional demonstration to support licensing of digital safety systems for use in nuclear power plants varies from country to country. However, most countries and regulators are using IEC 62566 [5] for general

guidance on issues related to V&V, overall qualification and acceptance of FPGA based systems and designs. Safety demonstration can take the form of a review of FPGA based modules, platforms and functional block libraries, and other IP core development and testing. Certification is based on a particular national or international standard and/or relates to dedication of predeveloped components or systems for use in a particular country by a national regulatory body. The most important principles of any safety demonstration are associated with what is needed to prepare a case to the national regulatory body that the component or system is safe for use. Functional demonstrations in most cases will provide some aspects of the information and analysis needed to satisfy the full set of requirements in a particular country. In the case of qualification of components or systems, this is usually done based on functions that are available for use on the platform, but not on nuclear power plant specific applications. This kind of functional demonstration can be useful for reducing licensing uncertainty, but it is only as valuable as the amount of information provided to the national regulatory body for review. In some cases, these reviews end in certification of only the basic architectures and building blocks of the platforms, but not of specific systems design. Vendors, licensees and regulatory bodies must understand what is being reviewed and what value will be provided before the effort is put into these reviews, to ensure expectations are met.

Effective implementation of any technology in the nuclear power industry is highly dependent on the success of both the development and use of national regulations and regulatory guidance. Technically complete and accurate regulations and regulatory guidance, and the effective use of these by national regulatory bodies and licensees, are key to the effective introduction of any new technology, particularly one as potentially complex as FPGAs.

Most national regulations and regulatory guidance are based on accepted national or international standards. When well established national or international standards exist, it is usually more effective for national regulatory bodies to incorporate the standards into the regulations or regulatory guidance. However, as has been discussed previously, FPGAs and similar CPLDs are only now beginning to be used extensively in the nuclear industry, and national and international standards, such as IEC 62566 [5], have only recently been introduced. These standards are not yet generally referenced in national regulatory guidance.

Although there are a number of industry guidance documents and standards for FPGAs in use in fields other than nuclear power, these standards have not been adopted for implementation in the nuclear power industry. The current situation is that some countries are now using IEC 62566 [5] as a basis for the development of FPGA applications and as best practice in the safety demonstration, while other countries are using existing regulations, regulatory guidance and standards originally written for non-FPGA based digital systems for the licensing of FPGAs. Although this has been possible, it has produced a number of challenges for vendors and licensees.

One of the challenging aspects associated with licensing these systems is that the technology (at least in nuclear power plant applications) is still relatively new, and the terminology used in regulatory guidance and standards has not yet become standardized.

Section 2.1 and the Glossary provide the definitions for this publication; however, it is important to understand that national and international standards may use different definitions (if these terms are defined at all). For example, IEC 62566 [5] uses a different set of definitions from this publication, and it is important to understand that the licensing process does not depend on names such as ‘FPGA’, ‘CPLD’, ‘macros’, ‘IP cores’, etc., but rather on the underlying concepts. These issues and their impact on the technical requirements have been widely discussed by the international community during the development of IEC 62566 [5].

Standard IEC 62566 [5] defines an HPD as an integrated circuit configured for nuclear power plant I&C systems, using HDLs and related software tools. HPDs are typically based on blank FPGAs, PLDs or similar microelectronic technologies. HPDs can be developed using ‘predeveloped blocks’ (PDBs) with tools used to implement the requirements in an assembly of microelectronic resources.

A ‘block’ is defined in IEC 62566 [5] as one of the parts that makes up a design; a block may be subdivided into other blocks. IEC 62566 [5] also defines a ‘native block’ as follows:

> “A Block which represents a pre-existing resource in the integrated circuit, e.g. an OR gate or a more complex block such as a multiplier or a serial transmission controller. By programming the HPD, the Native Blocks are configured and connected to provide the required function”.

A PDB is defined in IEC 62566 [4] as follows:

> "Predeveloped functional block usable in an HDL description … PDBs are typically provided as libraries, macros or Intellectual Property cores. They are used in the development of a HPD and incorporated in this HPD. … A PDB may need significant work before incorporation in a HPD, e.g. synthesizing an electronic circuit from the HDL statements, mapping the notional components of this circuit on the hardware structures of the physical integrated circuit and routing the interconnections."

This section will continue to use the definitions in Section 2.1 and the Glossary.

Many of the unique characteristics of FPGAs, which have been discussed in earlier sections of this report, provide challenges to licensing. The fact that FPGA technology is considered to be less complex than microprocessor based systems should not lead to an automatic assumption that it is inherently safer from a licensing viewpoint, because different licensing challenges are posed.

FPGA devices are fundamentally logical function designs implemented in hardware. As more functions are added to any digital system, greater attention is needed for the development process, whether or not it is a software development process or a process used to develop an FPGA application.

FPGA design methodologies and design tools have progressed more rapidly than evaluation tools and standards, and many I&C and process engineers may not fully appreciate the need to complete detailed functional reviews to understand and document the safety of FPGA based systems.

The use of automated design and analysis tools can improve system reliability. However, system safety could also be affected by design and requirements errors. Additionally, there may be an over-reliance on design and V&V tools. This could result in a significant challenge to safety if, owing to improper use of such tools, optimization adds unintended functionality to the intended design. This is why a detailed evaluation of design and V&V tools is needed.

Despite the challenges discussed above, there have been a number of successful applications of systems using FPGAs in the nuclear power industry. Some of these systems have been developed and implemented in countries where the national regulatory bodies use specific guidance, and some have been developed and implemented in countries that use safety case based licensing. Regardless of the regulatory review methods, the information needed for the safety demonstration is similar. The remainder of this section will discuss some of the unique aspects of licensing these systems.

4.2.2. Acceptance process for predeveloped resources

Licensing of predeveloped resources, whatever their commercial names may be, can usually be accomplished through the procedures for predeveloped software available to national regulatory bodies.

FPGA based systems are developed in the context of nuclear I&C projects. This development uses pre-existing resources, such as native blocks that correspond to the hardware resources actually present on the unprogrammed electronic circuit; examples of these include basic gates, memory cells, LUTs, switches and other electronic functions and PDBs, which are designed using HDL descriptions. These may be available as HDL code, synthesized for the chosen technology and then placed and routed on the chosen circuit or presynthesized. Typical names for the above entities are 'libraries', 'macros', 'IP cores', etc.

The predeveloped resources are incorporated into the final product and directly participate in its function; thus, the licensing process should consider them as strictly as the rest of the design, and a formalized acceptance process, based on technical evidence and the development processes, is needed.

In IEC 62566 [5], requirements about such an acceptance process have been formalized. For countries that do not use a specific FPGA standard, modified versions of acceptance criteria for predeveloped software are typically used. In many countries, the need to demonstrate the correct behaviour of each predeveloped resource used for all conditions to which it may be submitted in the considered context makes it mandatory for licensees to submit all documentation and evidence of proper design in order to obtain acceptance for safety applications; thus, the I&C designer should consider very carefully the range of macros, IP cores, etc., to be used in the design.

Care should be taken to ensure the information needed to support this process is available before the effort is undertaken. Most countries require significant information on design, development, testing, implementation, use history and reported error history of the predeveloped resource before licensing can be completed. Discussions

with vendors as to the availability of this information to the national regulatory bodies, before the review starts, are important to ensuring success.

Demonstration to particular safety levels, such as safety integrity levels, is common in some countries, but less common in others. Care should be taken in assuming, when reviewing a component to a particular level for a particular application in a particular country, that the certification has the same meaning and is used as part of the regulatory structure of another country. The effort to license an FPGA based component or system in one country will not necessarily be of value in another country's regulatory process. The legal implications of licensing may have significant differences in different countries.

4.2.3. Development life cycle

FPGAs are used for applications that are typically sufficiently 'complex' (i.e. number of inputs, outputs, processing modes, paths, etc.) to prevent verification based only on complete testing of the final product, i.e. development assurance is also needed. Thus, the development of each FPGA should follow a strict life cycle to organize activities including requirement capture, design and implementation, integration and validation, together with verification and test activities.

IEC 62566 [5] requires, for class 1 systems, that each FPGA application should be developed within a specific life cycle. The life cycle should identify distinct phases with well defined and consistent inputs, objectives, activities, outputs and criteria to decide whether the outputs are acceptable or not.

The outputs of each phase should be reviewed by a team that is independent from the design team and which has the appropriate knowledge and means. A verification plan should be established prior to starting verification activities. The plan should document the techniques and tools to be used, as well as the verification activities and criteria.

As mentioned previously, software standards such as IEC 60880 [14], IEEE 7-4.3.2-2003 [15] and IEEE 12207-2008 [16] have also been successfully translated to support FPGA life cycle review.

4.2.4. Design requirements

As their name suggests, HDLs are languages that were initially developed to describe hardware, either from a structural or from a behavioural point of view. In the context of FPGA development, they are used not only to describe the requested behaviour, but also to produce a solution that implements the behaviour (logic synthesis, place and route).

Although the different versions of HDLs, such as VHDL and Verilog, are standardized, this is not necessarily the case for the tools used to build an electronic application that implements the described behaviour. In fact, some features of the HDLs are not synthesizable, and some others are synthesizable with a given toolset but not with another, or may lead to different interpretations by different tools. Thus, the language used in a project should, in general, be restricted to a subset of the general HDLs, in order to guarantee that the high level descriptions of the design are synthesizable with standardized libraries and that the behaviour observed by simulation of the design matches the behaviour of the final circuit. Detailed requirements regarding this aspect have been formalized in IEC 62566 [5].

More generally, design and coding rules should be established and followed to ensure a stable and reliable design. Standards such as IEC 62566 [5] provide detailed requirements in areas including the use of dedicated resources such as clock trees and power rails, naming rules, the avoidance of functions that could lead to differences between simulated and synthesized behaviours, the proper way to create delays, power management, initialization of registers, management of non-functional configurations, etc.

To ensure the deterministic behaviour of the final circuit and to make the best use of design and verification tools, the design should include requirements about synchronous design. Where asynchronous features are used, a documented analysis of each path should demonstrate the correct behaviour for all possible cases and the absence of adverse behaviour.

Additionally, all of the documentation requirements that would apply to a software based system would also apply to an FPGA technology, including documentation of all design decisions such that an independent team may know and be able to assess them.

All configurations of development tools should also be documented. In particular, according to IEC 62566 [5], all parameters that influence the operation of tools (such as 'constraint files' of the synthesis tools) should be documented and specifically verified by the independent team, because they are actually part of the design. IEC 62566 [5] documentation requirements are quite similar to those found in the conventional software based systems standards. However, this standard [5] makes it clear that care needs to be taken to ensure that appropriate changes are made to account for the unique aspects of FPGAs and HDLs.

4.2.5. Analysis and verification

Different types of verifications are needed at different phases of the design. As discussed in IEC 62566 [5], the adequacy of the design should be verified for consistency and completeness with respect to the requirements. In addition to reviews and walkthroughs, the HDL description of the FPGA should be verified by simulation or other appropriate analysis, with proper justifications of the test cases and their coverage, to confirm that the logic's behaviour is as required.

In further steps of the design (synthesis, place and route), circuit information becomes available (propagation delays through gates and lines, capacitive and resistive loads, etc.) and should be taken into account in the verification, to confirm that it does not modify the logic behaviour. For example, maximum and minimum propagation delays should be tested to verify different aspects of the design (such as fulfilment of set-up times and hold times).

More generally, post-route analysis should confirm the compliance of the design with the technology rules defined by the providers of the unprogrammed circuit and the tools. IEC 62566 [5] provides detailed requirements to address these needs and to verify specific aspects such as intended redundancies, which may be unduly removed by the tools (or the opposite, where redundancies introduced by the tools may introduce failure modes that are not visible at high levels).

4.2.6. Integration and validation

There is often no clear separation or well defined boundaries between the integration of a given FPGA and the system integration. Therefore, the integration of an FPGA should be considered to be part of the system integration. Similarly, the validation of the FPGA should be considered to be part of the system validation.

Therefore, integration and validation should basically fulfil the requirements of system level standards such as IEC 61513 [17] or IEEE 603 [18], with additional requirements to address the specific needs of FPGAs. This is the approach taken by IEC 62566 [5] (as well as IEC 60880 [14] or IEEE 7-4.3.2-2003 [15] for software). IEC 62671 [19] should be consulted when integrating predeveloped FPGA hardware modules into an overall system design.

4.2.7. Modification

Relicensing of FPGA based components or systems has not yet been needed. However, national regulatory bodies normally only license the system or platform that they reviewed, not the subsequent revisions of the systems or platforms.

If FPGA based systems are to be revised to include new libraries, components and tools, the licence will become invalid and the system will require relicensing. This is an issue being faced by microprocessor based systems and platforms in several countries.

Currently, microprocessor based systems and platforms are being relicensed to address changes that have been made. This will likely be the case for FPGA based platforms, which will also need to be relicensed. If possible, the first licensing review of any FPGA based platform should include discussions between the vendors and the national regulatory bodies to establish what changes and modifications to FPGA based platforms will require relicensing and how other changes will be viewed from a regulatory standpoint.

In addition, standards provide guidance for the modification of either the requirements (or design) or of the predeveloped items such as the electronic circuit or libraries. For example, IEC 62566 [5] warns that a new version of the electronic circuit (e.g. 'same' part with reduced die size and/or increased speed) may very well perform differently in some cases, even if it is claimed to be 'compatible', and thus the acceptance process should be

performed again. As with software based systems modifications, the amount and kind of information that will be needed to support the relicensing will depend on the extent of the revision of the FPGA based platform.

4.3. REGULATORY PERSPECTIVES ON FIELD PROGRAMMABLE GATE ARRAY TECHNOLOGY, LICENSING AND STANDARDS

From a safety perspective, it is difficult to assess the correctness of FPGA devices without extensive documentation, tools and review procedures. The purpose of any safety review is to determine, to some level of confidence, that the system will perform as expected, complete its intended functions, and not introduce failures in other systems or components.

Reviews of systems and components typically include analysis of the design, review of the design process and testing of the final systems or components. For digital systems, there has been a higher level of emphasis on the review of the design process because the nature of software makes complete analysis and testing problematic. Therefore, all aspects of these systems should be addressed during safety reviews.

National regulatory bodies should review vendor information about FPGA design processes, including software design tools and development methodologies (similar to those currently used for software reviews). Consideration of specific device design information (over and above the system level documentation) for the system under review, such as requirements and design specifications, data sheets, user manuals, programmer manuals and so forth, is also needed to fully understand the system under review.

Additionally, regulators should review device failure mode information, including any mitigation of fault tolerant designs and workaround design changes. This information is needed for the assessment of the FPGA based system reliability and failure analysis. The national regulatory bodies need to be able to conclude that the FPGA based system failure modes have been adequately identified, that vulnerabilities have been mitigated and that the reliability has been bounded.

As the reliability of FPGA based systems is dependent on both the hardware reliability (at board level and internal to the FPGA chip) and logical component (software) correctness, it can be very challenging to predict. Most national regulators do not require formal quantification of I&C system reliability; however, all require qualitative assurance that adequate safety is achieved.

4.3.1. Successful licensing

Although there is currently no detailed regulatory guidance for FPGA based systems in many countries, a number of FPGA based digital safety systems have been successfully licensed and implemented in nuclear power plants. In most cases, these systems have been licensed using current regulations and regulatory guidance that reference standards that were not specifically developed to support FPGA based digital safety systems.

Successful licensing of these systems depends on a number of factors. Of paramount importance is the need for the licensee and the national regulatory body to have good agreement on the requirements and how they will be interpreted for the FPGA based system being licensed. In the current regulations for most countries and in some national and international standards, there is a lack of consistency on the definitions of what constitutes an FPGA or a similar CPLD. From these most basic issues to more detailed performance and analysis methods, there has sometimes been a significant lack of agreement on how to treat FPGAs in regulatory reviews, how to evaluate predeveloped functions, how to assess diversity, etc.

The publication of IEC 62566 [5] in 2012 has gone a long way towards setting the expectations for vendors, licensees and regulators for safety applications. It is imperative that vendors, licensees and regulators view the regulations and regulatory guidance in the same way.

In some countries, developing this understanding is part of the regulatory process. In others, methods need to be found to reach this level of understanding. In the United States of America for example, a relatively new process has been implemented as part of a new NRC interim staff guidance [20] that provides a method by which the NRC staff and the licensee can meet to discuss how a given system will be evaluated and what documentation would be needed to support the regulatory review. In other countries, the process of developing a safety case already includes detailed discussions between the regulator and the licensee.

Inconsistencies in expectations in how the regulations and regulatory guidance are to be applied are particularly challenging when, as in the case of FPGAs, guidance that was not specifically developed for the given new technology is applied to that technology. When this is the case, regulators must ensure that vendors and licensees understand the regulations in a consistent way. One extreme example of this challenge is that in some early cases, vendors believed FPGAs could be evaluated as strictly hardware systems, without the need to review life cycle development processes. In some cases, what is needed is better education of the licensees and/or regulators; in others cases, there needs to be more detailed regulatory guidance developed to ensure that the analysis and information provided to the regulatory bodies are sufficient for a thorough regulatory review. Again, the publication of IEC 62566 [5] has improved the consistency of expectations, where it is used.

Success in licensing these systems depends on early agreement on what needs to be provided and how the systems will be evaluated. Although it may be apparent that FPGA based components and systems can be tested like hardware, the amount and type of testing will be significantly different from those applicable to less complex analogue or digital systems that will likely be replaced. Although logical extensions to the current regulatory guidance are possible, it is important that these discussions happen early and often.

Other challenges to successful licensing of FPGA based components and systems stem from the fact that FPGAs are not yet widely used in the nuclear industry. Also of particular interest is the use of FPGAs in systems requiring diversity. Regulating how diverse an FPGA based system can be from a software based system or another FPGA based system is a matter of how to demonstrate acceptable diversity using FPGA based systems.

As more FPGA based components and systems undergo regulatory reviews, the regulatory acceptability will become clearer. In addition, the adoption of FPGA specific standards like IEC 62566 [5] by more regulators will make the review process more straightforward. For the present, current standards and guidance have proved to be sufficient to complete licensing reviews.

4.3.2. Application of existing software based guidance for field programmable gate array licensing

Prior to the issuance of IEC 62566 [5], many national regulators (among others, the NRC) applied standards developed for software based systems (such as PLCs) to review and license FPGA based I&C systems in nuclear power plants. Currently, with the issuance of IEC 62566 [5], many regulators are referencing this standard as technical guidance for licensing.

Other existing standards, such as IEC 61508 [8], IEC 60880 [14], IEEE 7-4.3.2-2003 [15], IEEE 603 [18] and IEC 62138 [21] in the United States of America, have been referenced and interpreted to address the software like development of FPGA based systems, but FPGAs are more properly analysed as having both hardware and software characteristics, and these standards do not include FPGA specific attributes such as FPGA hardware design practices, FPGA design entry methods or FPGA design tools.

In some cases, it is easier to demonstrate that FPGAs meet certain criteria in these standards, such as non-interference of different functions, compared to similar analysis for microprocessor based systems. In other examples, the use of these standards has been challenging because they do not cater to the unique aspects of the FPGA development process, including the implementation of HDL codes onto the FPGAs. In addition, these standards do not provide extensive guidance on the use of predeveloped tools, which are used somewhat differently in FPGA development.

This situation has changed slightly with the introduction of IEC 62566 [5], an FPGA specific standard that now includes FPGA dedicated requirements. Additionally, the software specific standards and review guidance documents do not tailor the design life cycle and V&V process to account for the specific characteristics of FPGAs. IEC 62566 [5] was developed to address these in nuclear power plants. However, this standard has not yet been widely accepted by national regulatory bodies.

Additionally, there are several significant licensing issues that are not addressed by IEC 62566 [5]. One is the need for guidance on use of FPGAs in diversity solutions for I&C systems. This was not addressed in IEC 62566 [5] because system analysis is part of the scope of the system standard IEC 61513 [17] and is not within the scope of IEC 62566 [5].

However, IEC 61513 [17] does not have any specific guidance on how to credit the differences between FPGA and microprocessor based systems. The present situation reflects the lack of consensus on this matter: some countries accept two different software components as means of diversification, some do not, but accept software

and HDL as a means of diversification, while some do not accept anything less than pure hardwired solutions as diversification.

Another challenge is the need for guidance on the acceptability of the uses of 'complete' or 100% tests in validation. This has been a significant issue in licensing these systems because using the strict definition of 100% testing, even for the simplest of FPGAs, requires testing of all combinations of inputs and internal states. In practice, to make 100% tests possible, the design must be simplified to reduce possible combinations. This is a recommendation of many standards including IEC 62566 [5]. To date, there has been no consensus between national regulatory bodies or in the technical community on how to define 100% testing so as to meet regulatory requirements in a way that it will allow a practical approach to system design.

As national regulatory bodies develop their own guidance (or endorse existing standards such as IEC 62566 [5]) and gain more experience, the effectiveness and predictability of licensing will improve. Despite the challenges discussed above, there have been a number of successful applications of safety systems using FPGAs in the nuclear power industry. Some of these systems have been developed and implemented in countries where the national regulatory bodies use specific guidance (such as the NRC), and some have been developed and implemented in countries that use safety case based licensing.

4.3.3. Standards for field programmable gate arrays

There are a number of standards that have been published for the design, development and implementation of FPGAs. As discussed above, IEC 62566 [5] is a standard for the development of FPGAs (HDL programmed integrated circuits in the standards terminology) for use in systems performing Category A functions or safety functions in nuclear power plants, depending on which regulatory convention it is being applied to. There are also a number of other standards such as those covering the use of FPGAs in other industries such as aerospace, and others used by manufacturers, such as VHDL coding standards.

IEC 62566 [5] is a second level IEC standard in the IEC 61513 [17] series. IEC 61513 [17] provides guidance applicable to I&C at the system level. It is supplemented by guidance for hardware (IEC 60987 [22]) and software (IEC 60880 [14] and IEC 62138 [21]), and now IEC 62566 [5] for FPGAs and similar devices. IEC 62566 [5] complements IEC 60987 [22], which deals with the generic issues of hardware design of computer based systems and refers to IEC 60880 [14] when addressing issues identical to those related to software development. IEC 62566 [5] provides an approach to requirements for design, implementation, verification, integration and validation of FPGAs and similar devices. It also provides procedures for the modification and configuration control of FPGA based systems, as well as the selection and application of tools used to develop FPGAs.

Other standards that have been used to develop FPGA based systems for nuclear power plant applications today include RTCA/DO-254 [23], which is an aerospace standard that primarily describes processes and does not describe technical aspects that are specific to FPGAs. This standard was used to develop early FPGA based products by one of the current vendors of FPGA based systems for nuclear power plants. RTCA/DO-254 [23] provides a means to show Federal Aviation Administration compliance in the design of complex electronic hardware in airborne systems including FPGAs and similar devices. Although no regulatory body endorses RTCA/DO-254 [23], it should be remembered that non-nuclear standards for FPGA development predate nuclear standards, and many vendors have developed processes in compliance with these standards.

Standards in common use by FPGA manufacturers include VHDL coding standards, as well as standards for interfaces of FPGA based boards. IEEE 1076-2008 [24] defines VHDL. It was originally developed under contract with the US Air Force in the early 1980s. The language has undergone numerous revisions, and has a variety of substandards associated with it that support or extend it in important ways. IEEE 1364-2005 [25] is also available as a standard for Verilog coding.

Over the last few years, the FPGA industry has also introduced several standards that define electromechanical interfaces between FPGA based boards and other boards. These standards facilitate interoperability, flexibility and reuse, enabling host systems to be easily adapted for new applications and interface protocols without the expense of a new FPGA board design. Some of the standards place a greater emphasis on performance, while others focus on simplicity and cost. An example of these standards is the American National Standards Institute ANSI/VITA-57-1 [26], which defines how FPGA mezzanine cards should interface with FPGA motherboards.

4.3.4. Development standards for mixed systems (field programmable gate arrays and microprocessors)

Current regulatory guidance and standards, such as IEEE 7-4.3.2-2003 [15] and IEC 61513 [17], do not address FPGAs or similar devices and microprocessors in the same component or system. This is because, in part, these higher level standards attempt to provide guidance that is not specific to particular technologies (microprocessors, PLCs or FPGAs).

However, when system designers and users attempt to take credit from a regulatory stand point for specific attributes of certain technologies, these more general standards might not be sufficient. For example, IEEE 7-4.3.2-2003 [15] provides guidance on communications between and within channels, and includes information on what precautions need to be taken and what the acceptance criteria for various communication architectures are. However, it does not distinguish between communication from FPGAs to FPGAs or from FPGAs to microprocessors, and it does not provide guidance on the potential hazards that need to be avoided. This is also the case for software and general hardware guidance such as IEC 60880 [14] and IEC 60987 [22].

In some ways, this is understandable because the system level documents are not intended to provide this level of guidance, and the more specific standards only discuss specific technology. In the case of IEC standards, mixed systems are addressed by the combination of IEC 61513 [17] (system level), IEC 60880 [14]/IEC 62138 [21] (software aspects) and IEC 62566 [5] (HPDs such as FPGAs).

Current standards also do not address the differences between versions of mixed systems, such as an FPGA with a hard core processor on the same chip, an FPGA without a processor interacting with an FPGA with a processor, and an FPGA chip interacting with a processor chip.

Even without this level of detail in the current regulations, regulatory guidance or standards, there have been a number of FPGA based applications successfully licensed using current, more general standards. Although special issues associated with diversity, communications and deterministic behaviour could arise from the use of mixed systems, the need for specific guidance on digital components or systems with both FPGAs and microprocessors performing significant aspects of the safety function or functions is not yet to the point that this issue will drive new or updated standards or regulatory guidance. For the present, licensees should use current higher level guidance and standards.

4.3.5. Licensing of field programmable gate array platforms

One of the most significant advantages of a design using FPGA technology is the possible reduction in overall complexity for simple functions, compared to software based systems. For these simple functions, a ‘direct implementation’ with FPGAs may be more straightforward and therefore easier to license than using a software based system that usually requires potentially complicated system software.

Structured design of FPGA systems may use a set of PDBs (for application functions, e.g. thresholds, voters, etc., and for system functions, e.g. self-supervision, transmissions) and hardware components (electronic boards, power supplies, cabinets, etc.) that constitute a platform.

Thus, the concept of ‘FPGA platform’ is fully relevant and should be taken into account in the licensing process. It is up to the platform vendor and/or licensee to determine what they want licensed: a full system or a platform alone, and then a system based on it.

A typical approach could be the licensing of the platform including the demonstration of logical properties, which would ease the licensing of future applications; a valuable property is the independent behaviour of the platform and of any possible application embedded in it. This means that:

— It is demonstrated that the behaviour of the platform cannot be influenced by any possible application, and thus the platform can be licensed, independently of any application.
— It is demonstrated that the logic behaviour of the application is not adversely influenced by the platform, and thus the licensing of the application may concentrate on its functional behaviour.

4.3.6. Development and verification tools for field programmable gate array based systems

The use of software tools can increase the integrity and reliability of the life cycle development process, and as a result, the quality of the final digital safety system. Tools are used for numerous purposes, including checking for adherence to design rules and standards, to support configuration control and for automated testing.

In some cases, tools can become necessary because a specific development methodology requires their use. This is generally the case for FPGA based digital safety systems. However, tools having insufficient reliability or quality may introduce faults or fail to find them.

The use of criteria and processes for tool selection is generally a design decision that would normally not be reviewed by the national regulatory bodies; however, the limits of applicability of all tools should be identified and well documented.

The regulatory bodies will review the tools and their output to ensure that they are not used outside their design limits. The tools used in the development life cycle for safety systems, including FPGAs, in nuclear power plants must be verified and assessed to a level that is consistent with the potential of the tool to introduce faults into the final safety system.

The use of technical evaluations of tool vendors may be an acceptable qualification method, provided that the corresponding documentation is available and that the regulatory body has the ability to examine the quality records of tool vendors. The tools need to have sufficient reliability and quality to ensure that they do not jeopardize the reliability and quality of the safety system that will be used in the nuclear power plant.

A tool can affect the FPGA by introducing errors or by failing to detect a fault. The level of verification and assessment required for a tool may also depend on the type of tool and whether the output of the tool can be fully verified and validated. FPGA tools include:

— Transformation tools such as HDL code analysers, synthesizers, routers and those that transform a text or a diagram at one level of abstraction into another;
— V&V tools such as simulators, timing analysers, test coverage monitors, testing tools, equivalence checkers and model checkers;
— Infrastructure tools such as those supporting the development;
— Configuration control tools such as version control tools.

Because of the extensive use of tools in all phases of the development of FPGA based digital safety systems, they are closely reviewed by most national regulatory bodies.

The guidance in current general digital systems standards such as the HPD standard IEC 62566 [5], IEC 60880 [14], IEEE 7-4.3.2-2003 [15] and similar standards provides only very general criteria for the review and acceptance of particular tools in digital system development.

Most national regulatory bodies require that all tools that will be used to perform safety functions or that are used in the development process for hardware, software or firmware to perform a safety function be treated as safety software (method II in Fig. 7). This can be a significant regulatory burden for predeveloped software tools used in any digital system development cycle. Tool developers typically do not provide detailed evidence of their configuration management for their tools or the level of error tracking and quality assurance needed to support and maintain their tools as safety grade software for nuclear power plant applications.

The alternative method that has been found acceptable to most regulatory bodies is to independently review the input and output of tools as part of the development process quality assurance programme to ensure that the tool is functioning properly, and that it has not introduced any possible new fault or failure mode into the final digital system or failed to detect an error if that is part of the function of the tool. Owing to the challenges with method II, this method (method I in Fig. 7) is more commonly used at the current time.

When tools are used for specific, well bounded parts of the development process, method I has proved to be effective.

The current tool qualification guidelines provided in IEC 62566 [5] provide a good translation of general guidelines for tool qualification found in IEC 60880 [14], IEEE 7-4.3.2-2003 [15] and similar standards to more specific aspects of tool qualification for FPGA based safety systems.

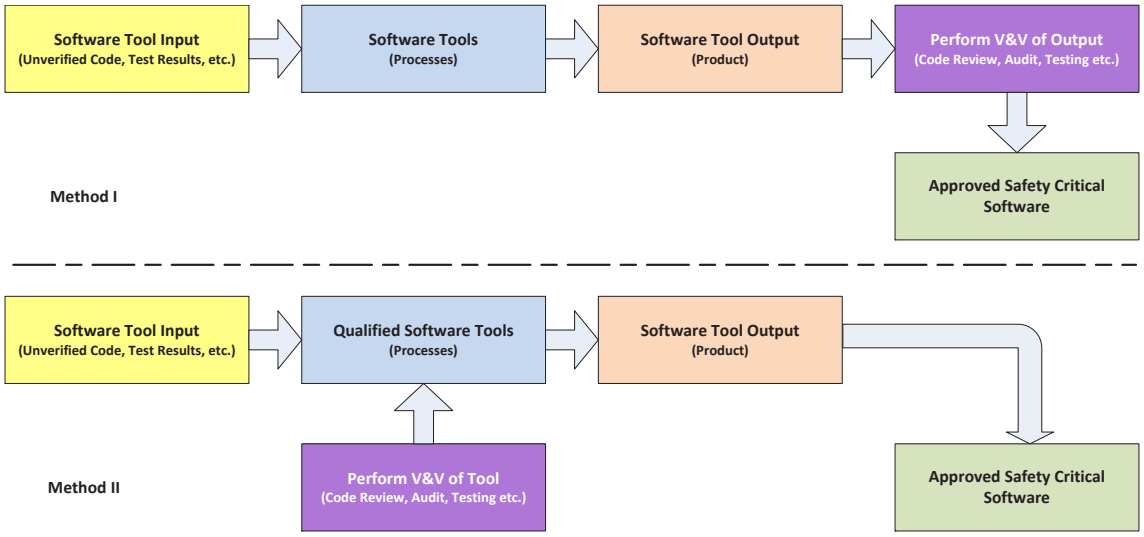


FIG. 7. Two possible methods that have been found acceptable for approval of tools. V&V — verification and validation.

As discussed in IEC 62566 [5], the tool qualification strategy should be developed, and all of the tools used should be qualified in accordance with that strategy. The strategy should consider the reliability and quality requirements of the tools and the type and use of tools in the development life cycle. For tools to be qualified, information to be provided to the regulatory body includes:

— An analysis of the tool development process and vendor tool history;
— Tool documentation to allow verification of tool output;
— Testing or validation of the tool;
— Evaluation of the tool over a period of use;
— Documented procedures and/or guidelines for tool use;
— Feedback of experience with tool use.

This information is similar to the information that would be provided for the qualification of any predeveloped software. The quality and reliability requirements of a tool should be determined by considering:

— The consequences of a fault in the tool;
— The possibility that a tool could cause or induce faults in the FPGA that implements the safety function;
— The relationship to other tools or processes that should mitigate the consequences of a fault in the tool.

FPGA based platform qualification/certification would usually include some regulatory evaluation of tool sets used to support development of applications using a particular platform.

Although there are significantly different levels of platform qualification/certification in various countries, electronic board level tools, library functions and most other development tools are generally included in these platform reviews. In the case of FPGA platform qualification/certification, this may be the most important aspect of these application neutral reviews.

Of particularly high interest to regulators in the review of tools, be it life cycle development, board or application level libraries or analysis tools, is the possibility to introduce faults, particularly those leading to common mode failures, into the FPGA based system. Great care must be taken to avoid introducing potential faults into FPGA based systems through tools because of the possibility of introducing similar errors to redundant or diverse parts of a safety component or system. Information provided to national regulatory bodies should include analysis that focuses on this aspect of tool qualification/certification.

4.3.7. Diversity (field programmable gate array–field programmable gate array and field programmable gate array–microprocessor) and 100% testing

The need for diversity review of I&C systems derives from the inability to predict, through testing and analytical methods, all of the failure modes for state machines.

Analysis can be used to predict the regions over which an analogue system exhibits continuous performance. Analyses of analogue system designs using methods based on first principles and tests can establish a reasonable expectation of continuous performance and proven capabilities over substantial ranges of input conditions. It is this aspect of analogue systems, not their 'simplicity', that distinguishes them from digital systems. In fact, analogue systems are not always simpler than microprocessor or FPGA based systems.

In microprocessor or FPGA based I&C systems, requirements, specifications, code, data, data transmission and hardware may be common to redundant divisions and/or functions. Although this commonality is the basis for many of the advantages of digital systems, it also raises a key concern: a design using shared data or code has the potential for CCF, defeating the redundancy achieved by the hardware architecture.

The issue of CCFs in digital systems has been widely known in the digital system research community for more than 30 years. It is because of this concern that most national regulatory bodies require software CCF evaluations of digital systems.

FPGAs have licensing issues similar to those of microprocessors, but there may be a major difference between the technologies. The software in a microprocessor based component can consist not only of the application logic itself, which is often very simple (e.g. comparison of a single physical parameter to a threshold), but also of much more complex functions needed to schedule this logic within the computer, to manage communication channels and networks, to perform self-supervision and diagnostics, etc. As these components run on the same microprocessor, they share internal resources such as processor registers, stacks and memory; thus, it is very difficult to demonstrate their independence.

Therefore, the demonstration of correctness of even a simple logic channel implementation can be very challenging because it is 'merged' with more complex parts. On the other hand, the parallel nature of processing within an FPGA allows the designer to implement these functions into different parts of the circuit, or even in physically different circuits that are guaranteed by hardware to communicate only in one way and do not share computing structures such as stacks or memory. Thus, independence between the core logic and the infrastructure (communications, diagnostics, etc.) can be easier to justify. This would make the demonstration of simple logic channels much easier than in the case of a software implementation. On the other hand, the ability to demonstrate the independence and adequacy of FPGA development tools is typically more challenging than for microprocessor based systems.

FPGAs are generally viewed by regulatory bodies as diverse from software based systems, and provide a higher level of diversity due to the different nature of their components. To determine the level of diversity for FPGA based I&C systems, it is necessary to analyse the system and specify types and subtypes of diversity according to a classification scheme. One such basic scheme is described in NUREG/CR-7007 [27], and can be used to review different FPGA technology attributes. Attributes of combinations of FPGAs and microprocessors for main and diverse systems should be reviewed to:

— Determine local diversity metrics for specified diversity types and subtypes;
— Calculate integrated diversity metrics, taking into account local metric values, and weights corresponding to diversity types and subtypes;
— Compare calculated integrated diversity metrics with acceptable values and make decisions regarding the assessed system.

An example of determination of an acceptable value is described in NUREG/CR-7007 [27]. In general, values of local and integrated metrics should be normalized.

The method for determining what level of diversity is needed and the criteria for assessing it for an I&C system has been determined by the NRC (BTP-7-19 [28]), but generally, the existence of CCFs in software must be assumed to be credible. This is true for both software and FPGA based systems (in the FPGA case, because of the susceptibility of associated tools to CCFs).

One method to avoid the proposed analysis is for a simple system to be completely (100%) tested, i.e. including all possible states, to eliminate the concern that there are potentially unanalysed failure modes. However, this approach has not been successfully used because of disagreements between national licensing bodies and vendors on what constitutes 100% testing.

Multiple interpretations of what is necessary for 100% testing include testing of all possible combinations of input and output values. But if the device is not stateless (i.e. if it retains some internal memory of the past), then the internal state registers and their possible values need to be treated in combination with the inputs in the determination of the test cases. This is also true of external conditions (i.e. initial or boundary conditions on the device). This approach requires that all possible states of the device must be knowable and known, so that the test metrics can be developed and demonstrated to be complete. The above approach to 100% testing would include some method of demonstration that all states have been tested.

Some national regulatory bodies use a definition of 100% testing that requires that every possible combination of inputs, outputs, initial conditions, internal and external states, and every signal path, is tested and found to produce only correct responses. Or if it is not possible to test all internal and external states, then every possible combination of inputs and every possible sequence of potential inputs are tested and found to produce a correct response.

There are some experts who argue that logical separation should be taken into consideration in order to reduce the number of required test cases. For example, if the FPGA design implements two functions that cannot electrically or logically interfere with one another, then one might argue that they can be tested separately and thus reduce the number of combinations needed. An equivalence class argument (similar to what is done in fault injection testing) could also be put forward to reduce the testing requirements.

Regardless of what definition is used, this method of resolving diversity will only become practical when there is agreement between the I&C system vendor and the national regulatory body as to the exact definition, before the licensing process has begun.

4.3.8. Reliability claims for field programmable gate array based systems

FPGA based systems share, with software based systems, the fact that their reliability depends on two factors: hardware reliability, which is typically assessed quantitatively using information available from failure data or failure analysis methods, and some assessment of the correctness of the logic (sometimes referred to as software reliability or conditional reliability due to software). The correctness of the logic would include coverage of all possible sources of errors in requirements, design and implementation. This matter is generally assessed qualitatively, based on the consensus of experts; typically, the experts must agree on the fact that using the processes and design requirements of the appropriate standards would provide sufficient confidence that potential errors in the logic will not degrade the required system reliability. For example, in the context of IEC standards, using a combination of IEC 62566 [5] (FPGAs), IEC 60880 [14] (software), IEC 61513 [17] (system) and IEC 60987 [22] (general hardware) would provide confidence in the correctness of class 1 systems.

Some regulators require that a quantitative reliability number be provided as part of the licensing process. In the case of FPGAs, the methods and data needed are similar. The major new challenge associated with quantitative estimation of the effect of software on reliability for FPGAs is the need to assess non-real time software tools as part of the ‘likelihood of correctness’ review.

Although there are methods recommended in the ‘software reliability’ literature, there is no consensus on them, and none have been applied to the licensing of FPGA based systems to date.

4.3.9. Field programmable gate array specific redundancy and fault tolerance

Tolerance to faults (either intrinsic due to destruction of an element, or due to external influences such as SEU) can be increased by hardware level features such as ‘triple modular redundancy’ (transistors are triplicated in such a way that an analogue two out of three vote is performed in each redundant group) or built-in CRC checks.

However, in the context of nuclear safety systems, redundancies are required to fulfil the single failure criterion, and thus they should be physically separated and electrically isolated.

Therefore, the hardware level features mentioned above, which are internal to an integrated circuit, cannot discharge this regulatory requirement. However, they are frequently used to increase system availability or reliability.

4.3.10. Field programmable gate array verification after modification

For safety systems, the principle is that all phases impacted by a modification have to be performed again. This is explicitly required by most standards, and as discussed in Section 4.2.7, in many cases, will require relicensing by the national regulatory body.

Requirements for relicensing vary from country to country, but the use of FPGA based systems should not significantly alter the application of these requirements for relicensing of modified systems.

Reverification implies that the impacted documents have to be updated and verified again. Verifying the modification of the logic itself (at HDL level) is a classical problem of functional impact analysis, which is not specific to FPGAs.

More specific is the fact that a small change in the logic (e.g. using an OR function rather than an AND, in some equations) may lead to a different set of equations after logic simplification, and this different set of equations may result in a completely different implementation due to specialization and availability of hardware resources on the chip.

Thus, the implementation of verification, and especially the post-route analyses, may be affected much more than the extent of the logic modification would suggest (for details on the objectives and requirements of verification, including post-route analyses, see IEC 62566 [5]).

4.3.11. Documentation

The purpose of a regulatory review of any component or system is to assess whether or not the component or system will perform its intended function. Evaluations of components or systems are performed against established technical criteria in order to ensure that they will perform their safety functions and comply with regulations, so that public health and safety will be protected. It is not intended that the review or audit activities by the national regulatory body include an evaluation of all aspects of the design and implementation of I&C systems.

The review's scope and associated information need to comply with the regulations of the particular country. Reviews should assess the adequacy with which the component or system meets the regulations. To support this review, the national regulatory body needs access to system design, development processes and associated V&V analyses.

For the review of digital systems (based on either FPGA or software based systems), additional information to assess the acceptability and correct implementation of life cycle activities, as well as information needed to support detailed reviews of the architecture, are needed to determine if regulatory requirements such as redundancy, independence, deterministic behaviour and diversity are satisfied.

Each national regulatory body will have its own list of documents required throughout each phase of the development life cycle. Although these lists are fairly well understood, and the information needed to complete a regulatory review for a digital system is well defined, most lists were not developed with FPGAs in mind. The primary challenge with developing a list of needed documents for regulatory review of FPGAs is to ensure that all aspects of the design processes that can affect the safe performance of the component or system are included.

A typical list of documentation needed to support a regulatory review can be found in IEC 62566 [5] and NRC Interim Staff Guidance 6 (ISG-6) [20]. Components and systems that use FPGAs will have a list of documentation similar to that for other digital systems.

Generally, the types of documents used in the design and implementation of FPGAs will map well to the documentation for software based systems, and will include information necessary to document the implemented functions, interfaces and I/O protocols, as well as descriptions of control registers, etc. The documentation needs to describe all modes of operation and transitions, including power on and reset.

Documentation requirements need to consider design, integration, V&V and support activities. Configuration management for FPGAs should be similar to that for software based systems.

The configuration management needs to support the regulatory review, as well as to ensure that the V&V is supported. The need to support these basic development processes is similar for software based systems and FPGAs, because the latter rely on software to support their configuration.

As discussed in Section 4.2.2, extra care should be taken to ensure that the information needed to support the review of predeveloped resources is available before the start of licensing is undertaken. Regulators in most countries require significant documentation of design, development, testing, implementation, use and reported error history of predeveloped resources before licensing can be completed. Demonstration of predeveloped resources is frequently one of the most challenging aspects of project documentation.

4.3.12. Gradation of lower class systems

Gradation of requirements for lower class systems can be accomplished through the use of available guidance for gradation of microprocessor based systems.

Many countries do not use lower class systems in their regulatory structure. In these countries, gradation would not be as important; however, some of the countries that do not use IEC standards use other categories such as 'important to safety' or 'risk important'. For most of these cases, requirements are defined for special classes, at least at a high level. Countries that use IEC standards and treat class B and C systems as part of their regulatory structure need some method for qualifying class B and C systems. In this sense, FPGAs have the same licensing issues as microprocessors.

As discussed above, most countries use more general standards and regulatory guidance for the licensing of FPGA based components and systems, but the more FPGA specific standard IEC 62566 [5] provides very useful information.

As the life cycle for development of FPGAs is similar to that of software, the most effective method for licensing would be to use the gradation that has been previously developed as part of IEC standards for lower class software systems, even if the systems were developed differently. The gradation would include the design process and qualification of the FPGA electronic design, particularly in the areas of quality assurance, independent V&V and configuration management, but not hardware testing and qualification aspects of FPGA regulatory reviews. For lower category systems, justification of the use of IP cores may be easier, and may be similar to the situation in which predeveloped software components are used in a software based system.

4.3.13. Reduction in variations in standards and country regulations

As discussed earlier in this section, it is critically important for the vendor, licensee and national regulatory body to agree on the regulations, regulatory guidance and standards that will be applied to the licensing of any digital system. Prior to the issuance of IEC 62566 [5], this was a significant challenge for FPGA based systems, because most countries had not yet adopted specific regulations or regulatory guidance for FPGA based systems.

As international and national standards are updated and new standards developed, there is an effort to reduce the variations in the national and international standards. This is generally true for all I&C standards, and is being worked on by both the standard developing organizations, such as the IEC and the IEEE, and other programmes, such as the Multinational Design Evaluation Program (MDEP).

In the case of harmonization of FPGA standards and regulatory guidance, there is some possibility that this effort will be more successful, because detailed technical positions have not yet been firmly established. The international community (IAEA, Nuclear Energy Agency, IEC, MDEP, etc.) will continue its efforts to harmonize the standards and use all of the available resources and pathways to do so.

One suggestion has been to have more direct interactions between regulators as they prepare to endorse new standards and/or write new regulations. From the standpoint of reducing the uncertainty in licensing, the provision of more guidance on the documentation needed and the reduction of variances in the areas that are reviewed and referenced in the design, development and testing of FPGA based systems will be an improvement.

4.3.14. Simplification of regulatory requirements and structure

As more experience in the development, implementation, regulation and use of FPGA based systems is gained, it will be possible to update and streamline some of the current regulations and guidance. As more FPGA specific guidance is developed and tested, it should be possible to replace much of the overlapping regulatory structure (covering both FPGAs and software based systems). This should provide greater consistency and efficiency in licensing reviews for the national regulators and improved regulatory predictability for vendors and licensees.

5. FIELD PROGRAMMABLE GATE ARRAY BASED REPLACEMENT SYSTEMS AND NEW NUCLEAR POWER PLANT DESIGNS

This section provides an overview of possible applications of FPGA technology for the replacement of I&C modules or systems in existing nuclear power plants, as well as for I&C system designs for new nuclear power plants. Even though the approaches to be followed are similar to those used for the application of microprocessor based technology, designers and users must take into account the specific characteristics of FPGAs when considering adopting this technology.

As discussed in Section 2, FPGA technology can provide all required nuclear power plant I&C functions while exhibiting several advantages over other digital technologies. These will be discussed in the following subsections dealing with module replacement, system level replacement or a combination of these.

5.1. REPLACEMENTS AND UPGRADES IN EXISTING PLANTS

Plans call for many nuclear power plants to remain in operation for longer periods of time than originally anticipated, which has resulted in the need to replace I&C components.

Replacement projects are undertaken because systems or some of their components are obsolete, unreliable and/or require frequent maintenance. Replacement may also be undertaken in order to add or change functionality to improve the safety and reliability of the system. For example, changes may need to be performed in order to eliminate single point vulnerabilities in the system.

This section provides a discussion on replacement of I&C systems or parts thereof using FPGA technology, which is particularly suitable for this purpose. (Additional details can be found in Refs [1, 3].)

5.1.1. One for one module replacements or upgrades

Replacements or upgrades at the module level may result from three main causes:

— For existing nuclear power plants, module level replacement may be undertaken to address obsolescence issues.
— In some cases, this may result from the need to upgrade or improve the functionality of the system.
— As an alternative to replacement, users or designers may choose to modify the existing modules to meet new technical requirements.

As for all projects, the first step, once the need for changes is identified, is the requirements specification. Such requirements might already exist, or may have to be reverse engineered. After all requirements are defined, the new FPGA based modules are developed using the life cycle model described in Annex II.

If the overall I&C design was already FPGA based, the additional functionality could be implemented by reconfiguring the FPGA. However, there might be instances in which the additional functional, performance or reliability requirements may dictate that the additional functionality be segregated from the existing one. Examples of this could be avoidance of CCFs or faster response time requirements.

FPGA technology offers the following advantages:

— If the I&C system is FPGA based and the change comprises stringent timing and no segregation requirements, it could be incorporated into an existing FPGA chip, and there would be no need to add another module. This would greatly simplify the change, given that no additional hardware and space would be required.
— If segregation or performance requirements are such that designers have to implement changes via additional modules, FPGA technology presents the following advantages over other technologies:
  • It tends to occupy less space than a CPU based application, which is especially important in existing plants due to their limited space availability;
  • Owing to their faster response times, the use of FPGAs makes it easier to meet timing requirements;
  • Functional segregation is easier to achieve in FPGA based systems.

One reason for the addition of modules could be a decision to add redundancy to a safety related function. Adoption of FPGA technology for the additional modules, where the existing equipment is not FPGA based, could be advantageous because it could provide added diversity as well as redundancy, while also providing an FFF solution that would simplify the interfaces with the rest of the systems and the installation.

Where the solution to functionality changes involves the replacement of existing modules or the upgrade of existing ones (third case above), it is worthwhile considering the possibility of using FPGA technology to design circuits that are FFF compatible with the existing ones in order to minimize the need for changes to interfaces and simplify installation.

For additional advantages and challenges associated with FPGA technology, see Section 2.5.

5.1.2. Multiple module replacement

Instead of a module for module replacement, there may be a need to reduce the number of modules in a system by grouping the functionality from multiple obsolete modules into fewer replacement modules. This would result in more availability of space for additional devices, lower power requirements, fewer spare modules, improved reliability and simpler interfaces.

Depending on the individual cases, the functional performance and safety requirements definition may be as simple as the sum of all the requirements for each of the original modules, or there may be a need to change or add additional functionality based on operating experience or licensing requirements.

When grouping functions from multiple to single modules, designers must take into consideration common faults, failure modes and propagation of faults, as well as associated licensing implications. Cost–benefit and reliability analyses might show that some of the additional space made available by the functional grouping could be used as means to provide additional redundancy and/or diversity to achieve a net increase in the reliability of the system.

5.1.3. Replacement of entire systems

There could be several reasons why users may decide to replace entire I&C systems in operating stations, some of which include:

— Platform obsolescence;
— Low reliability;
— New licence requirements (e.g. diversity requirements on redundant systems or improvement of the HSI);
— The need for additional or better functionality;
— The need for additional space.

Solutions to one or more of the above problems could be implemented using different technologies. FPGA technology provides effective means of addressing each of the above cases as follows:

— *Platform obsolescence*. All electronic components are vulnerable to obsolescence, and FPGAs are no exception. However, the higher portability of their applications makes FPGA based platforms less vulnerable to obsolescence.
— *Low reliability*. FPGA based platforms could be custom designed and certified based on widely used industry standards for nuclear power plant safety I&C applications. Additionally, they do not include operating systems or other general purpose resources, and avoid unnecessary functionality that could deteriorate their reliability. In addition, as in microprocessor based platforms, diagnostics and self-checks can be added in order to improve the availability of the system.
— *New licence requirements*. FPGA technology constitutes an effective means of diversification as a backup for microprocessor based systems. However, it is limited in its capability to provide a flexible graphical user interface or other complex HSIs. Therefore, for cases where complex HSIs are required, suitable means of interfacing between the two systems would have to be provided, with a suitable means for preventing fault propagation from the HSI to the FPGA system.
— *The need for additional or better functionality*. The same arguments as those provided for the problem of low reliability above apply here. In addition, FPGAs, owing to their parallel processing capabilities and lack of operating systems, allow applications with faster response times, which are not always achievable with other technologies.
— *The need for additional space*. By itself, this seldom constitutes a reason for system replacement, but in many cases, it is a desirable outcome of the refurbishment project. FPGAs provide an opportunity for significant space economy by grouping functionalities originally distributed among various modules, resulting in a smaller footprint. The reader is referred to Section 5.1.2. for considerations associated with the grouping of functions.

5.2. FIELD PROGRAMMABLE GATE ARRAY BASED INSTRUMENTATION AND CONTROL SYSTEMS AND DEVICES FOR A NEW NUCLEAR POWER PLANT DESIGN

Design requirements for I&C systems in nuclear power plants are derived from probabilistic and deterministic safety analyses and regulatory requirements, including diversity and defence-in-depth requirements. The processes for deriving I&C system requirements from high level requirements are closely monitored and controlled. These higher level requirements could result in additional specific applications for which FPGA technology could be of particular interest, such as dedicated interfaces, priority logic, communication gateways or filters.

Strategies adopted by designers to meet the above requirements could vary; however, in addition to compliance with functional, performance and safety requirements, a robust design should include, as a minimum, the following features:

— Adequate margins;
— System expandability;
— Resilience to obsolescence;
— Compliance with modern standards;
— Amenability to the introduction of diagnostics and function separation;
— Ability to interface with other technology based platforms;
— Ease of removal and installation;
— Cabling and connection minimization;
— Wide support by users and suppliers;
— Ease of prototyping and optimizing applications;
— Ease of maintenance and operation;
— Availability of skilled designers and maintainers;
— Availability of design and troubleshooting tools;

— Compatibility of equipment footprint with available space;
— Relatively simple designs and V&V processes.

FPGA technology provides all of the above capabilities, in varying degrees, to designers. As such, it would be possible for utilities to include FPGA platforms as a basis or a redundant/diverse alternative for I&C systems in new nuclear power plants.

Most challenges are presently found in the lack of sufficiently skilled designers and maintainers and the tools necessary to generate applications and maintain equipment, particularly for utilities that prefer to use their own staff to carry out the above activities. However, an increase of available expertise is expected as the industry continues to gain experience and confidence in the utilization of FPGAs in nuclear plant applications.

6. SUMMARY

Over the lifetime of a nuclear power plant, conditions that can impact its I&C systems change. Plants experience new and changing requirements and commitments, ageing and obsolescence of systems and equipment, increasing goals to improve performance and reduce O&M costs, and the emergence of new technologies. Plant life extensions further amplify these changes due to the increased lifetime of the plant. Therefore, utilities need to select I&C strategies that will allow them to continue to operate effectively under these changing conditions. New plants should be designed to accommodate expected changing conditions over their lifetimes.

Historically, the nuclear power industry has tended to replace obsolete analogue equipment with microprocessor based technology. In the process, it has addressed many of the shortcomings of the old technology and provided opportunities for improving performance and reliability, as well as reducing the complexity of I&C systems. However, some safety related microprocessor based solutions have had licensing challenges, and the industry has been actively trying to overcome these by searching for alternative technologies.

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, of the order of tens of milliseconds or longer. The functional and performance capabilities offered by current FPGA circuits meet most safety and non-safety requirements associated with I&C applications in nuclear power plants.

For appropriate applications, FPGAs are a viable technology that allows the full range of replacement options and new systems. Given that FPGAs are usually simpler systems than microprocessor based solutions, with easier separation of functions, it is perceived that licensing will be easier. FPGAs offer solutions to some of the needs linked to I&C related problems such as diverse actuation, more complex priority logic, and secure data communication filters and gateways.

FPGAs are an industrially mature technology. There are FPGA chip suppliers catering to the needs of industrial applications, including the aerospace and military aviation industries. These suppliers provide high reliability chips, long term support, and suitable development and V&V tools. There also exists significant operating experience in other industrial sectors.

FPGA solutions could be implemented as replacements at the module level in order to upgrade system functionality and performance, as the addition of redundant and/or diverse equipment to meet safety function availability requirements, as full system replacements to achieve all or part of the above objectives, or as the addition of new systems.

FPGA based nuclear power plant applications have advantages over other technologies and intrinsic challenges. This report has identified the most relevant ones.

An increased number of FPGA based applications can be expected as nuclear operators and regulators become more familiar with the advantages of the technology. In addition to the current typical FPGA applications, such as providing computer emulation, interfaces between systems, replacement of obsolete modules and diversity against CCFs, the technology is expected to be applicable to large scale replacement of I&C systems in modernization projects, as well as providing complete I&C systems (safety and non-safety) in new nuclear power plant designs.

Once the system requirements have been clearly, thoroughly and unambiguously defined, users and practitioners are encouraged to become sufficiently familiar with the advantages and challenges of FPGA technology (and any other technologies under consideration) to determine the degree to which it fits their technical needs. They should also take into consideration licensing issues, the development life cycle of FPGA technology and associated standards, as referred to in this report and other related publications, in order to be able to support their technical and business cases and thus arrive at a sound decision as to whether FPGAs should be the technology of choice for their applications.

FPGAs may, if used properly, implement functions in a more straightforward and testable manner than current software based systems, and thus reduce technical barriers to licensing. The development of specific guidance for FPGA based nuclear safety systems, such as IEC 62566 [5], adds confidence that challenges will be properly addressed by utilities, I&C developers and regulators.

REFERENCES

[1] ELECTRIC POWER RESEARCH INSTITUTE, Guidelines on the Use of Field Programmable Gate Arrays (FPGAs) in Nuclear Power Plant I&C Systems, Rep. EPRI TR-1019181, EPRI, Palo Alto, CA (2009).

[2] INTERNATIONAL ELECTROTECHNICAL COMMISSION, Nuclear Power Plants — Instrumentation and Control Important to Safety — Classification of Instrumentation and Control Functions, IEC Standard 61226, IEC, Geneva (2009).

[3] ELECTRIC POWER RESEARCH INSTITUTE, Recommended Approaches and Design Criteria for Application of Field Programmable Gate Arrays (FPGAs) in Nuclear Power Plant I&C Systems, Rep. EPRI TR-1022983, EPRI, Palo Alto, CA (2011).

[4] PRECKSHOT, G.G., Method for Performing Diversity and Defence-in-Depth Analyses of Reactor Protection Systems, Rep. NUREG/CR-6303, Lawrence Livermore National Laboratory, CA (1994).

[5] INTERNATIONAL ELECTROTECHNICAL COMMISSION, Nuclear Power Plants — Instrumentation and Control Important to Safety — Development of HDL-Programmed Integrated Circuits for Systems Performing Category A Functions, IEC Standard 62566, IEC, Geneva (2012).

[6] INTERNATIONAL ATOMIC ENERGY AGENCY, Software for Computer Based Systems Important to Safety in Nuclear Power Plants, IAEA Safety Standards Series No. NS-G-1.1, IAEA, Vienna (2000).

[7] BOBREK, M., BOULDIN, D., HOLCOMB, D.E., KILLOUGH, S.M., SMITH, S.F., WARD, C., WOOD, R.T., Review Guidelines for FPGAs in NPP Safety Systems, Rep. NUREG/CR-7006, Office of Nuclear Regulatory Research, Washington, DC (2010).

[8] INTERNATIONAL ELECTROTECHNICAL COMMISSION, Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems, IEC Standard 61508, IEC, Geneva (2010).

[9] MAY, J., HUGHES, G., ZHU, H., "Statistical software testing and test adequacy", Testing Safety-Related Software (GARDINER, S., Ed.), Springer-Verlag, London (1999) 155–170.

[10] ELECTRIC POWER RESEARCH INSTITUTE, Estimating Failure Rates in Highly Reliable Digital Systems, Rep. EPRI TR-1021077, EPRI, Palo Alto, CA (2010).

[11] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, IEEE Standard for Test Access Port and Boundary-Scan Architecture, IEEE Standard 1149.1-2013, IEEE, New York (2013).

[12] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, IEEE Standard for Property Specification Language (PSL), IEEE Standard 1850-2010, IEEE, New York (2010).

[13] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, IEEE Standard for SystemVerilog — Unified Hardware Design, Specification, and Verification Language, IEEE Standard 1800-2009, IEEE, New York (2009).

[14] INTERNATIONAL ELECTROTECHNICAL COMMISSION, Nuclear Power Plants – Instrumentation and Control Systems Important to Safety — Software Aspects for Computer Based Systems Performing Category A Functions, IEC Standard 60880, IEC, Geneva (2006).

[15] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations, IEEE Standard 7-4.3.2-2003, IEEE, New York (2003).

[16] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, Systems and Software Engineering — Software Life Cycle Processes, IEEE Standard 12207-2008, IEEE, New York (2008).

[17] INTERNATIONAL ELECTROTECHNICAL COMMISSION, Nuclear Power Plants — Instrumentation and Control Systems Important to Safety — General Requirements for Systems, IEC Standard 61513, IEC, Geneva (2003).

[18] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, IEEE Standard Criteria for Safety Systems for Nuclear Power Generating Stations, IEEE Standard 603, IEEE, New York (1998).

[19] INTERNATIONAL ELECTROTECHNICAL COMMISSION, Nuclear Power Plants — Instrumentation and Control Systems Important to Safety — Selection and Use of Industrial Digital Devices of Limited Functionality, IEC Standard 62671, IEC, Geneva (2013).

[20] NUCLEAR REGULATORY COMMISSION, Licensing Process Interim Staff Guidance, US NRC Interim Staff Guidance DI&C-ISG-06, NRC, Washington, DC (2011).

[21] INTERNATIONAL ELECTROTECHNICAL COMMISSION, Nuclear Power Plants — Instrumentation and Control Systems Important to Safety – Software Aspects for Computer Based Systems Performing Category B or C Functions, IEC Standard 62138, IEC, Geneva (2004).

[22] INTERNATIONAL ELECTROTECHNICAL COMMISSION, Nuclear Power Plants — Instrumentation and Control Important to Safety — Hardware Design Requirements for Computer-Based Systems, IEC Standard 60987, IEC, Geneva (2007).

[23] FEDERAL AVIATION ADMINISTRATION, Design Assurance Guidance for Airborne Electronic Hardware, Document RTCA/DO-254, US FAA, Washington, DC (2005).

[24] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, IEEE Standard for VHDL Analog and Mixed-Signal Extensions — Packages for Multiple Energy Domain Support, IEEE Standard 1076-2008, IEEE, New York (2011).

[25] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, IEEE Standard Verilog Hardware Description Language, IEEE Standard 1364-2005, IEEE, New York (2005).
[26] AMERICAN NATIONAL STANDARDS INSTITUTE, FPGA Mezzanine Card (FMC) Standard, ANSI-VITA-57-1, ANSI, Washington, DC (2008).
[27] WOOD, R.T., BELLES, R., CETINER, M.S., HOLCOMB, D.E., KORSAH, K., LOEBL, A.S., MAYS, G.T., MUHLHEIM, M.D., MULLENS, J.A., POORE, W.P., III, QUAILS, A.L., WILSON, T.L., JR., WATERMAN, M.E., Diversity Strategies for Nuclear Power Plant Instrumentation and Control Systems, NUREG/CR-7007, Office of Nuclear Regulatory Research, Washington, DC (2010).
[28] NUCLEAR REGULATORY COMMISSION, Standard Review Plan for the Review of Safety Analysis Reports for Nuclear Power Plants: Chapter 7, Instrumentation and Controls, NUREG-0800, BTP 7-19, NRC, Washington, DC (2003).

Annex I

SPECIFIC APPLICATION EXAMPLES AND EXPERIENCE

Field programmable gate array (FPGA) systems and components have been installed in many nuclear power plants worldwide, and many more projects utilizing FPGAs are expected to start in the future. This section summarizes some of these projects.

I–1. FIELD PROGRAMMABLE GATE ARRAY BASED SYSTEMS INSTALLED IN OPERATING PLANTS

Argentina

As a form, fit and function type replacement, the existing special safety systems annunciation system for the Embalse nuclear power plant in Cordoba is being replaced by using FPGA technology. In addition, the signal processing units of the new safety shutdown system No. 2 (SDS2) main heat transport pump trip function are being developed for the Embalse nuclear power plant. Both systems are supplied by RPC Radiy (an FPGA based system designer and manufacturer) and will be delivered by CANDU Energy (formerly known as Atomic Energy of Canada Limited).

Bulgaria

In Bulgaria, RPC Radiy has installed six FPGA based Category A engineered safety feature actuation systems (ESFASs) in Units 5 and 6 of the Kozloduy nuclear power plant [I–1], with each unit having three ESFASs. Installation and commissioning were performed between 2008 and 2010, with each system requiring an installation time of 18–29 d.

Canada

An FPGA based emulator for the obsolete PDP-11 computers used in several non-safety systems in CANDU plants has been implemented. The FPGA based emulator has been used in fuel handling systems for over 10 years. An updated version of the emulator has also been prepared for use in digital control computers (DCCs), which provide reactor control functions. Other peripheral devices associated with the DCCs also are being replaced with FPGA based solutions.

In addition, FPGAs are being used as the basis for the replacement of the display system controller circuit card. The components on the existing circuit card are obsolete and are affected by ageing issues.

Czech Republic

FPGAs are used in the non-programmable logic (NPL) part of the instrumentation and control (I&C) systems at Units 1 and 2 of the Temelin nuclear power plant. The NPL provides the priority logic arbitration between the microprocessor based primary reactor protection system (PRPS) and the diverse protection system (DPS), and the fixed wire component control signals to determine the safety actuations. The NPL also implements the function of ensuring that loads go to a safe state in the event that there is a discrepancy between the PRPS and the DPS. Another function of NPL is the implementation of the safety diesel load sequencer, including the sequencer automatic test.

France

In 2009, EDF initiated the FPGA based replacement of obsolete electronic modules comprising the rod control system (RCS) and the reactor in-core measurement system in its 900 MW series of plants (both are non-safety systems). FPGAs are used as part of the new systems, performing control and interface functions. RCSs are being

replaced with the new FPGA based systems in all 34 units of the 900 MW series over the 10 year period from 2009 to 2019.

Japan

Toshiba has supplied the following FPGA based safety and non-safety systems to operating Japanese nuclear power plants: startup range and power range neutron monitoring systems for boiling water reactors (BWRs), and radiation monitoring systems for both BWRs and pressurized water reactors (PWRs).

In addition to developing the above FPGA based monitoring systems for the advanced boiling water reactor (ABWR) design, Toshiba has developed reactor trip and isolation systems for ABWR type plants.

Republic of Korea

Complex programmable logic devices (CPLDs) are used in digital safety systems of operating nuclear power plants in the Republic of Korea to perform functions such as system initialization, bus interface, control of input/output signal transfers, memory control and peripheral channel control.

FPGAs are used in performing self-diagnostic functions. FPGAs are also foreseen to perform component control functions for the engineered safety features in new APR-1400 plants under construction in the Republic of Korea.

Sweden

FPGAs are used in the component interface module (CIM) of the replacement safety system in Unit 2 of the Ringhals nuclear power plant. The CIM acts as the interface between the primary safety actuation system (microprocessor based) and the actuated plant equipment, and also responds to signals from the independent diverse actuation system (DAS), as well as operator commands. The CIM incorporates a priority logic that ensures the appropriate signals are passed to each component, or in the event of a conflict, the components are placed into a safe state.

Ukraine

In Ukraine, FPGA based systems have been installed in safety applications in all nuclear power plant sites, using the RPC Radiy platform. In the past 11 years, 30 reactor trip systems (RTSs), 10 reactor power control and limitation systems, 1 rod control system, 18 engineered safety feature actuation systems, and 6 nuclear and conventional island control systems have been installed in various power units of the Zaporozhye nuclear power plant, the South Ukraine nuclear power plant, the Rovno nuclear power plant and the Khmelnitski nuclear power plant [I–1].

In the RTS, a primary system and a diverse system have been employed, both FPGA based but using different FPGA chips from different vendors to implement the RTS application, providing diversity and defence in depth against potential common cause failures.

United States of America

An FPGA based control system was installed in 2009 to replace obsolete equipment in the main steam and feedwater isolation system at the Wolf Creek Generating Station. This system was based on the advanced logic system (ALS) platform developed by Westinghouse/CS Innovations. The application was approved by the US Nuclear Regulatory Commission (NRC) and is now in operation [I–2].

In addition, FPGAs are being used as the basis for the replacement of the timing modules used in emergency diesel generators. The components on the existing circuit cards are obsolete and affected by ageing issues.

I–2. POTENTIAL FUTURE APPLICATIONS

I–2.1. Potential applications in operating plants

Canada

Ontario Power Generation continues to assess high maintenance and obsolescence prone applications for possible FPGA based replacement. Even though the DCCs have been in use since the early days of CANDU plants, they are now experiencing obsolescence and must be replaced with a reliable current technology, of which FPGAs, given their emulation and other desirable capabilities, could be a strong candidate. The same can be stated about other electronic components in CANDU nuclear power plants, including the shutdown computers.

France

EDF is currently developing a CPLD based solution to upgrade the rotational speed measurement system for the primary pumps of their 900 MW series plants. In addition, EDF has developed and formally verified an FPGA based emulator for the Motorola 6800 microprocessor, which is currently used to perform a number of I&C functions in the 1300 MW series of plants, including safety critical reactor protection functions. Use of the emulator would allow EDF to overcome problems with obsolescence of the Motorola 6800 processor, while retaining the existing, qualified software that implements the reactor protection functions.

Republic of Korea

The programmable logic controller based DPSs will be replaced with FPGA based logic controllers in eight power units in the Yonggwang nuclear power plant and the Ulchin nuclear power plant by the end of 2015. After successful replacement of the DPSs, the installation of FPGA based primary protection systems will be considered.

United States of America

The operator plans to use the FPGA platform ALS from Westinghouse/CS Innovations in the Diablo Canyon nuclear power plant to replace the digital reactor protection system (RPS) and ESFASs. The application is currently under review by the NRC [I–2].

I–2.2. Field programmable gate array applications in new plant instrumentation and control designs

Canada

CANDU Energy is actively collaborating with RPC Radiy to develop and pilot an engineering process suitable for creating FPGA applications for safety critical functions for the new enhanced CANDU-6 reactor (EC-6). The pilot project is focusing on development of an FPGA application for the safety shutdown system No. 1 (SDS1) and the emergency core cooling system of the EC-6 design.

China

In China, the China Nuclear Power Engineering Company is evaluating the use of FPGA based safety systems in the new, advanced version of their earlier CP1000 design. The new design, ACP1000, is a 1000 MW PWR unit designed by China National Nuclear Corporation. FPGA based solutions are considered for the RPSs, DASs, ESFASs and the post-accident monitoring system [I–3].

The China Techenergy Company is developing an FPGA based platform, FitRel, to be used in the design of the DAS of Units 5 and 6 of the Yangjiang nuclear power plant, which are under construction following the CPR-1000 design [I–4].

State Nuclear Power Automation System Engineering Company and Lockheed Martin Corporation have been developing a safety I&C platform, NuPAC, and the NuPAC based RPS for the China advanced pressurized water reactor (CAP-1400). The two companies are jointly pursuing generic approval from the NRC for the NuPAC platform and approval from the Chinese National Nuclear Safety Administration for the specific application of NuPAC in the CAP series of nuclear power plants [I–5].

Triconex of Invensys Process Systems (now Schneider Electric) is also employing FPGAs for the priority logic modules in new plants being built in China.

United States of America

In Westinghouse's AP1000 design, FPGA based solutions developed by CS Innovations are used to implement the safety related CIM system, interfacing between field components and the protection and safety monitoring system (PMS) and the plant control system (PLS). FPGAs also provide prioritization of commands from the safety related PMS and the non-safety related PLS. The non-safety related DAS of AP1000 provides a diverse backup to the RPS, and it is implemented by using the CS Innovations FPGA based ALS platform [I–2].

In Areva's US EPR design, the priority actuation and control system (PACS) is implemented by using programmable logic devices (PLDs). PACS provides prioritization of commands from safety related systems and non-safety related systems, such as the DAS. Programmable electronic devices, such as FPGAs or PLDs, are considered as options for implementing DAS [I–2].

In the Combined License Application for South Texas Project, Units 3 and 4, it was indicated that the ABWR design's neutron monitoring system and the reactor trip and isolation systems will be implemented based on the I&C platform developed by Toshiba using non-rewritable antifuse FPGAs [I–2].

In Mitsubishi's US advanced pressurized water reactor design, FPGAs are used in certain modules in the MELTAC digital platform. These modules are the control network interface module, the bus master module and the power interface module [I–2].

In the Hitachi-GE economic simplified BWR design, a PLD based independent control platform (ICP) is used to implement the following functions in the accident mitigation systems: vacuum breaker isolation function, anticipated transient without scram and standby liquid control, and high pressure control rod drive isolation bypass function. The ICP platform is different from the microprocessor based platforms used in the RTS and ESFAS [I–2].

REFERENCES TO ANNEX I

[I–1] BAKHMACH, I., KHARCHENKO, V., SIORA, A., SKLYAR, V., ANDRASHOV, A., "Experience of I&C systems modernization using FPGA technology", paper presented at 7th Int. Topical Mtg on Nuclear Plant Instrumentation, Control and Human Machine Interface Technologies, Las Vegas, NV, 2010.

[I–2] ZHAO, J.Y., "Overview and regulatory summary of FPGA/PLD based applications in U.S. nuclear industry", paper presented at 5th Int. Workshop on Applications of FPGAs in Nuclear Power Plants, Beijing, 2012.

[I–3] RIGANG, C., "Evaluation of FPGA based safety system application in ACP1000", paper presented at 5th Int. Workshop on Applications of FPGAs in Nuclear Power Plants, Beijing, 2012.

[I–4] SHI, G., "FitRel platform based on FPGA technology", paper presented at 5th Int. Workshop on Applications of FPGAs in Nuclear Power Plants, Beijing, 2012.

[I–5] QIU, S., "Collaborative development of FPGA based safety platform and reactor protection system", paper presented at 5th Int. Workshop on Applications of FPGAs in Nuclear Power Plants, Beijing, 2012.

Annex II

TYPICAL LIFE CYCLE FOR A FIELD PROGRAMMABLE GATE ARRAY PLATFORM

A field programmable gate array (FPGA) based platform life cycle should consider the life cycle of each component of the platform. Figure II–1 and Table II–1 present a typical life cycle model of an FPGA based platform, including hardware components, functional blocks and FPGA electronic design. This model is based on IEC 61508 [II–1].

Text cont. on p. 69

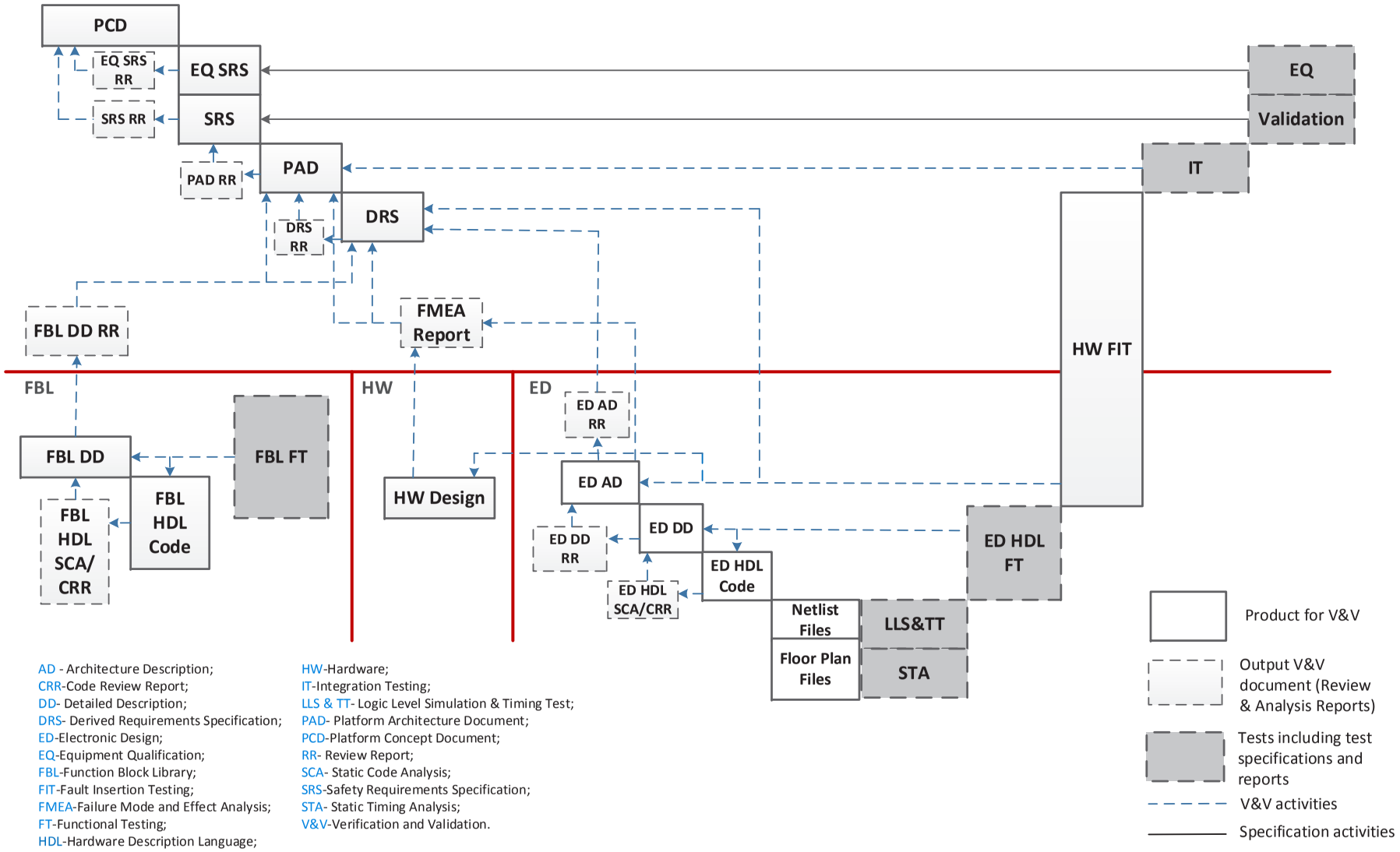


FIG. II–1. Example of life cycle for a field programmable gate array based instrumentation and control platform.

TABLE II–1. EXAMPLE OF LIFE CYCLE FOR A FIELD PROGRAMMABLE GATE ARRAY BASED INSTRUMENTATION AND CONTROL PLATFORM

Field programmable gate array (FPGA) based instrumentation and control platform life cycle phases	Phase description	Phase results
Requirements identification	This phase identifies the safety needs of the target or market industries and clients. The platform concept decisions are market driven, based on what functionality the vendor decides it is confident it can implement. Functionality required for the platform is defined in a black box way.	Platform concept document
Safety requirements specification (SRS)	This phase starts the formal engineering work of designing the platform. The black box requirements from the requirements identification phase are carefully specified in order to be realizable, verifiable and traceable. All safety related requirements of the platform should be documented, including requirements from applicable standards and from the specific safety requirements for the intended application(s). Equipment qualification (EQ) requirements should be specified during this phase. Validation is based on these requirements.	SRS EQ SRS
SRS review	SRS and EQ SRS should be reviewed for completeness, consistency and testability.	SRS review report EQ SRS review report
Validation planning and validation	This phase is executed in two parts: planning and testing. The planning part generates the validation test plan and the test specification from the SRS. The testing part produces the validation test report near the end of the platform development. Typical testing methods are: — Functional and black box testing to demonstrate requirements coverage; — Interface testing together with boundary value analysis and equivalence classes; — Performance/stress testing; — Error guessing. As part of the validation phase, EQ is to be implemented. The EQ test plan and specification describe the planning part of the EQ from the EQ SRS. The testing part produces the EQ test report near the end of the implementation phase.	Validation test plan Validation test specification Validation test report EQ test plan EQ test specification EQ test report

TABLE II–1. EXAMPLE OF LIFE CYCLE FOR A FIELD PROGRAMMABLE GATE ARRAY BASED INSTRUMENTATION AND CONTROL PLATFORM (cont.)

Field programmable gate array (FPGA) based instrumentation and control platform life cycle phases	Phase description	Phase results
Platform architecture design	This phase implements the requirements from the SRS. While the SRS is a black box activity, platform architecture design is white box, and defines the internal structure of the platform, providing design solutions for the requirements, and allocating the parts of the solution to the various major components of the platform. Associated with these major components (and possibly any repeatedly used minor modules), the platform architecture design identifies and assigns: — FPGA safety functional requirements; — FPGA safety integrity requirements, required to implement the safety concept. These requirements may be stated in the platform architecture design itself, or in an extension derived requirements specification (DRS) document. Platform architecture design should describe the platform with drawings related to semiformal methods, and drawings should accompany text where they clarify the design. All interfaces between the different components should be defined. Platform architecture design should consider: — Requirements coverage; — Error detection and reaction. Error detection design and testing should consider: — How to test different kinds of logic (e.g. algorithms, encoding of key data); — Where to locate the testing logic (same chip, separate location of same chip, separate chip or chip on a separate module); — The use of diverse chips, or similar chips on the same or separate boards; — Where to use these techniques in the product.	Platform architecture design Derived requirements specification
Platform architecture design review	Platform architecture design and DRS should be reviewed for completeness, consistency and testability.	Platform architecture design review report DRS review report

TABLE II–1. EXAMPLE OF LIFE CYCLE FOR A FIELD PROGRAMMABLE GATE ARRAY BASED INSTRUMENTATION AND CONTROL PLATFORM (cont.)

Field programmable gate array (FPGA) based instrumentation and control platform life cycle phases	Phase description	Phase results
Hardware (HW) design	In this phase, the architecture requirements for HW from the platform architecture design are applied to design and build the chassis and to design and populate the printed circuit boards (PCBs) with their appropriate power supplies, watchdogs, logic modules, input/output (I/O) devices (e.g. analogue to digital converters), communications devices (e.g. transceivers), the FPGAs themselves, etc., and produce the required final documentation for the hardware. The HW design should be precisely described by circuit descriptions, circuit diagrams, layouts, part lists, etc. It is assumed that in the FPGA electronic design (ED) phase, the FPGA logic will be developed to perform self-testing of hardware components. Some use of a temporary FPGA logic tool to exercise the PCB components to confirm correctness of the layout might be involved.	HW modules circuit descriptions HW modules circuit diagrams HW modules PCB layouts HW modules part lists HW modules mechanical and assembly drawings
Failure modes and effects analysis (FMEA) for HW design	To validate HW design reliability, analysis techniques, e.g. FMEA, might be used. As mentioned above, to perform self-testing of HW components, some FPGA logic may be involved; therefore, FMEA might be deferred to the FPGA ED architecture phase.	FMEA report

TABLE II–1. EXAMPLE OF LIFE CYCLE FOR A FIELD PROGRAMMABLE GATE ARRAY BASED INSTRUMENTATION AND CONTROL PLATFORM (cont.)

Field programmable gate array (FPGA) based instrumentation and control platform life cycle phases	Phase description	Phase results
FPGA electronic architecture design	In the platform architecture design phase, functionality is allocated to the various components to form an integrated architecture for the FPGA ED. Typically, this phase defines the functionality of the major functional blocks within the FPGA design and particularly their interconnection and interaction with other blocks, both within the FPGA and with external interfaces. Features, such as diagnostics and self-checks, that are necessary to check the correct functionality of the design are described in this phase. The FPGA ED architecture definition should include: — All the ED functional blocks; — The interfaces between the blocks; — Performance, e.g. definition of device operating clock frequency; — External I/O constraints (speed, voltage, separation). The following safety related features should be considered during this phase: — Synchronous versus asynchronous design; — Logical and physical separation into subsystems; — Layering of subsystems with decoupled buses; — Loose coupling of subsystems to avoid propagation of timing problems and faults; — HW redundancy of separated modules with checkers to address random HW failures (hard and soft errors); — Information redundancy, such as error correcting code, protected communication protocols; — On-line protection mechanisms: ● Memory protection for all bus masters including direct memory access; ● Built-in self-tests (BISTs) at startup for redundant units; ● Cyclic BISTs of non-redundant units.	FPGA ED architecture description
FPGA ED architecture review	FPGA ED architecture description should be reviewed for completeness, consistency and testability.	FPGA ED architecture description review report

TABLE II–1. EXAMPLE OF LIFE CYCLE FOR A FIELD PROGRAMMABLE GATE ARRAY BASED INSTRUMENTATION AND CONTROL PLATFORM (cont.)

Field programmable gate array (FPGA) based instrumentation and control platform life cycle phases	Phase description	Phase results
Functional block library (FBL) design	During this phase, each block specified in the FPGA ED architecture phase is designed. The FBL may include platform specific blocks (transceivers, diagnostic elements), as well as application specific blocks (logical, mathematical, time functions, etc.). The FBL design description may be at the level of specifying state machine functions, mathematical functions, detailed module I/O definitions, etc. The specific considerations relating to FPGA design for this phase might be: — Random access memory usage and arrangement; — Clocking resource (phase locked loops, routing) and arrangements; — Module I/O connectivity, bus types.	FBL design description
FBL design review	FBL design description should be reviewed for completeness, consistency and testability.	FBL design description review report
FBL coding	During this phase, the FBL design description is translated into a synthesizable design description, which typically takes the form of a very high speed integrated circuit hardware description language (VHDL) or Verilog description of the circuit functions and typically uses a standard text editor for design entry. Schematic design entry methods might also be used. One of the safety related features connected with FBL coding is application of coding rules. Such rules may support different safety aspects of the design, for example: — Avoidance of asynchronous logic; — Support of error detection and correction mechanisms; — Clocks; — Loops; — Description of finite state machines; — Naming conventions; — Coding style, etc. Coding rules should require application of hardware description languages (HDLs) to its specification, i.e. to comply with appropriate standards (e.g. IEEE 1076-2008 [II–2] for VHDL). FPGA vendor specific coding guidance should also be considered. Coding rules should be clearly documented and kept under strict configuration management control.	FBL hardware description language (HDL) code (synthesizable design files)

TABLE II–1. EXAMPLE OF LIFE CYCLE FOR A FIELD PROGRAMMABLE GATE ARRAY BASED INSTRUMENTATION AND CONTROL PLATFORM (cont.)

Field programmable gate array (FPGA) based instrumentation and control platform life cycle phases	Phase description	Phase results
FBL functional testing and static code analysis	This phase involves verification of FBL HDL code. To verify FBL HDL code, the following techniques may be used: — Functional simulation; — Static code analysis. Functional simulation, also referred to as behaviour simulation, is used to verify the behaviour of HDL code. Functional simulation is implemented using corresponding tools. Simulation coverage criteria, including positive (requirements coverage) and negative (error coverage) testing stimuli, should be identified and followed taking into account appropriate regulatory documents. Static code analysis is performed via automated tools in order to detect deviations from coding rules in the source code. See Section 3.2 of the main text of this report for more details on verification and validation (V&V) techniques.	FBL functional test plan FBL functional test specification FBL functional test report FBL static code analysis report
FPGA ED	During this phase, platform related functional blocks produced during the FBL coding phase are integrated together based on the FPGA ED architecture. All FPGA ED components should be specified and described by: — Functional view; — Interfaces; — Static data flow or object flow view; — Dynamic/event view; — Failure view (list failure modes and fail-safe state).	FPGA ED detailed description
FPGA ED review	FPGA ED detailed description should be reviewed for completeness, consistency and testability.	FPGA ED detailed description review report
FPGA ED coding	During this phase, the FPGA ED detailed description is translated into a synthesizable design description, to create higher level functions and ultimately the top level FPGA design. Synthesizable descriptions take the form of HDL type of circuit functions and typically use a standard text or graphic editor for design entry. Coding rules should be followed during FPGA ED coding.	FPGA ED HDL code (synthesizable design files)

TABLE II–1. EXAMPLE OF LIFE CYCLE FOR A FIELD PROGRAMMABLE GATE ARRAY BASED INSTRUMENTATION AND CONTROL PLATFORM (cont.)

Field programmable gate array (FPGA) based instrumentation and control platform life cycle phases	Phase description	Phase results
FPGA ED functional testing and static code analysis	This phase involves verification of FPGA ED HDL code. To verify FPGA ED HDL code, the following techniques may be used: — Functional simulation; — Static code analysis. See Section 3.2 of the main text of this report for more details on V&V techniques.	FPGA ED functional test plan FPGA ED functional test specification FPGA ED functional test report FPGA ED static code analysis report
FPGA ED implementation	The implementation phase consists of logic synthesis, placement and routing, and bitstream generation. The first step of FPGA ED implementation is FPGA ED logic synthesis. During logic synthesis, the synthesizer transforms the HDL code of the FPGA ED into the gate/cell level scheme (netlist). Most synthesizers generate FPGA independent schematic representation of the HDL code model, as well as the FPGA specific schematic representation. For FPGA specific schematic representation, gates/cells may be different. The result of logic synthesis is textual (e.g. ED interchange format) or graphical files. The synthesizer may apply different kinds of optimizations, which could be defined in terms of design constraints. Design constraints could basically affect the following attributes of FPGA ED: — Timing characteristics; — Pin assignment and adjustment; — Topology of FPGA ED in the FPGA chip. Design constraints are typically defined in constraint files, using the constraint editor. One or more constraint files can be mapped to FPGA ED by adding them to a project configuration. The syntaxes used for defining design constraint files are usually vendor specific. Special attention should be paid by the design and V&V teams to provide hard evidence of correctness and consistency of design constraints. After the logic synthesis, FPGA ED placement and routing is carried out. This is a tool driven process that determines where registers and gates described in the netlist will be placed within the FPGA chip. This process also determines the connection paths between design elements. The resulting design connectivity is defined by the floor plan, which is a schematic representation of the location of the different logic entities in the FPGA chip. The place and route tool also generates a timing file that is more accurate than the one produced by synthesis, because it also includes timing associated with routing. Design constraints should be considered for placement and routing, as well as for logic synthesis. The last step of the implementation phase is bitstream generation. The output of this phase is the configuration file, which can be implemented into the FPGA chip. It contains all data required to configure the FPGA chip.	FPGA ED netlist FPGA ED floor plan FPGA ED bitstream

TABLE II–1. EXAMPLE OF LIFE CYCLE FOR A FIELD PROGRAMMABLE GATE ARRAY BASED INSTRUMENTATION AND CONTROL PLATFORM (cont.)

Field programmable gate array (FPGA) based instrumentation and control platform life cycle phases	Phase description	Phase results
FPGA ED logic level simulation, timing simulation and static timing analysis	This phase involves verification of the FPGA ED netlist and floor plan. The following techniques might be applied: — Logic level simulation; — Post-layout timing simulation; — Static timing analysis. See Section 3.2 of the main text of this report for more details on V&V techniques.	FPGA ED logic level simulation and timing test plan FPGA ED logic level simulation and timing test specification FPGA ED logic level simulation and timing test report FPGA ED static timing analysis (STA) test plan FPGA ED STA test specification FPGA ED STA test report
FPGA ED integration	During the FPGA ED integration phase, the configuration files, which were derived at the previous step, are downloaded to the FPGA chip of the corresponding HW module. Special HW such as configuration interfaces (i.e. the Joint Test Action Group) is required to download the configuration file into the FPGA chip. Some FPGA chips and appropriate tools provide automatic checking of integration correctness.	HW modules with integrated FPGA ED
Integration testing	Integration testing phases are conducted at two stages for the ED. At stage 1, negative (i.e. fault insertion) tests applied to minimal hardware configurations may be performed. Negative integration testing (i.e. fault insertion) should confirm FMEA results for each HW module and provide credit for self-diagnostic coverage. At stage 2, positive integration tests involving all HW module types take place. Positive integration testing should confirm that HW modules with integrated ED together comply with the platform architecture design.	Fault insertion test plan Fault insertion test specification Platform integration test plan Platform integration test specification Fault insertion test report Platform integration test report
Validation	See validation planning and validation.	Validated platform

REFERENCES TO ANNEX II

[II–1] INTERNATIONAL ELECTROTECHNICAL COMMISSION, Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems, IEC Standard 61508, IEC, Geneva (2010).

[II–2] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, IEEE Standard for VHDL Analog and Mixed-Signal Extensions — Packages for Multiple Energy Domain Support, IEEE Standard 1076-2008, IEEE, New York (2011).

GLOSSARY

100% testing. There is no globally agreed definition for 100% testing. The following is one example of a definition. A system can be said to have been 100% tested if every possible combination of inputs, outputs, initial conditions and internal and external states, and every signal path is tested and found to produce only correct responses. Or if it is not possible to test all internal and external states, then every possible combination of inputs and every possible sequence of potential inputs is tested and found to produce only correct responses.

antifuse. A technology for storing the programming or configuration of the interconnects, configurable logic blocks (CLBs) and input/output (I/O) blocks in a programmable logic device (PLD) such as a field programmable gate array (FPGA) or complex programmable logic device (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 antifuse.

application specific integrated circuit (ASIC). An integrated circuit customized for a specific use and configured by means of a mask at the factory.

assertion. A statement in a design that is claimed or assumed to be true. For example, an assertion may be used to state that the results of a sine function are in the range [−1, +1]. Assertions usually help in understanding a design, or help design verification. An assertion violation at runtime means that something is wrong.

authorization. The granting, by a regulatory body or other governmental body, of written permission for an operator to perform specified activities. Authorization could include, for example, licensing, certification or registration. The term authorization is also sometimes used to describe the document granting such permission. Authorization is normally a more formal process than approval.

bitstream. A contiguous sequence of bits (binary digits), representing a stream of data, serially transmitted continuously over a communications path. It is frequently used to describe the configuration data to be loaded into an FPGA.

certification. A confirmation of certain characteristics of a component, system or part. This confirmation is often, but not always, provided by some form of external review, education, assessment or audit.

combinatorial logic. In digital circuit theory, a concept in which two or more input states define one or more output states, where the resulting state or states are related by defined rules that are independent of previous states.

complex programmable logic device (CPLD). A PLD that contains a number of ‘macro cells’ that are essentially the same as programmable array logic (PAL), and the means to interconnect them. A CPLD is sometimes referred to as a ‘super-PAL’.

crossbar. A mechanism for connecting input wires and output wires in PLDs; for example, an $n \times m$ crossbar connects n different input wires to m output wires.

electrically erasable programmable read only memory (EEPROM). A type of solid state storage containing non-volatile memory that can be erased and reprogrammed. Erasure of data is done electrically.

electronic design block. A functional block used in electronic design.

fault injection. A technique for improving the coverage of a test by introducing faults to test code paths, in particular, error handling code paths, which might otherwise rarely be followed. It is often used with stress testing and is widely considered to be an important part of developing robust software.

field programmable gate array (FPGA). An integrated circuit that can be programmed in the field by the instrumentation and control (I&C) manufacturer. It includes programmable logic blocks (combinatorial and sequential), programmable interconnections between them and programmable blocks for inputs and/or outputs. The function is then defined by the I&C designer, not by the circuit manufacturer.

finite state machine. An abstract model of a machine that has a primitive internal memory and a behaviour composed of a finite number of states and transitions between those states based on inputs, and output actions on other parts of the design. The behaviour of a finite state machine can be represented in a state transition diagram. Within a given state, for each input combination, 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.

flash. A type of solid state storage containing non-volatile memory that can be erased and reprogrammed. Flash is similar to EEPROM, but its memory is erased in larger blocks.

flat logic or flat hardware logic. Logic that is implemented directly in a circuit's electronic design using simple, configurable native logic blocks and interconnections, and not using any complex native blocks (such as microprocessors with their runtime software).

flip-flop. A bistable 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. As flip-flops are implemented electronically, they require power and ground connections.

floor plan. A schematic representation of tentative placement of the major functional blocks for an integrated circuit. In modern, electronic design processes, floor plans are created during the place and route stage.

form, fit and function (FFF). A description of an item's identifying characteristics. If the specifications, or criteria, for FFF of a particular item are met, then the item may generally be considered interchangeable with other items with the same requirements.

functional block. A visual way to represent functions. Each functional block contains the inputs, outputs, processes, requirements and constraints of a given function. The position of the block on a functional block diagram in relation to other blocks displays how the functional block interacts with other blocks, and in what order functions can be performed by the controller.

functional block library. A library of functional blocks.

gate array. An array made up of 'basic cells', each containing a number of transistors and resistors. Interconnecting pathways are used to create the desired functionality.

hard intellectual property (IP) core. An 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 and route process as would be required for a soft core. These are necessarily circuit technology specific.

hardware description language (HDL). A language that allows one to formally describe the functions and/or the structure of an electronic component for documentation, simulation, analysis or synthesis.

hardware description language programmed device (HPD). A class of electronic circuits that are configured using HDLs. These devices include FPGAs, PALs, programmable logic arrays and PLDs.

intellectual property (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 of faster development and a way of securing continued business.

licence. A legal document issued by the regulatory body granting authorization to perform specified activities related to a facility or activity. A licence is a product of the authorization process (or licensing process).

licensee. A holder of a current licence. The licensee is the person or organization with overall responsibility for a facility or activity (the responsible legal person).

logic array. An array made up of 'logic cells'. Interconnecting pathways are used to create the desired functionality.

logic synthesis. A process by which an abstract form of desired circuit behaviour, typically a register transfer level (RTL), is turned into a design implementation in terms of the resources (logic gates and other native blocks) of an actual hardware circuit. Common examples of this process include synthesis of HDLs, including very high speed integrated circuit hardware description language (VHDL) and Verilog. Some tools can generate bitstreams for PLDs such as PALs or FPGAs, while others target the creation of ASICs. Logic synthesis is one aspect of electronic design automation.

look-up table (LUT). An electronic design block that replaces runtime computation with a simpler array indexing operation. The savings in terms of processing time can be significant, as retrieving a value from memory is often faster than undergoing an 'expensive' computation or I/O operation.

microprocessor. A multipurpose, programmable device that accepts digital data as input, processes them according to instructions stored in its memory, and provides results as output. It incorporates the functions of a central processing unit (CPU) on a single chip. The semiconductor manufacturing process is now able to put multiple CPU cores onto a single chip.

native block. A block of circuitry that is hardwired into the FPGA circuit and represents a resource that can be used to create the desired application. Native blocks include the array of relatively simple CLBs that are interconnected by the application programming, and other blocks, such as LUTs that are predeveloped and embedded in the circuit to perform specific functions (e.g. a commonly used data communication interface), and can be used if needed as part of the application.

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

place and 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 array logic (PAL). A type of simple PLD that consists of a programmable AND plane followed by a fixed OR plane.

programmable logic array. A type of simple PLD that consists of two levels of logic: an AND plane and an OR plane, both of which are programmable.

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 ASIC, which is manufactured to perform a specific application and cannot be changed after manufacture.

register transfer level (RTL). In synchronous digital circuits, a description of signal flow between registers (flip-flops) and the combinatorial logic functions of the gates through which signals flow.

safety integrity level. Commonly referred to as SIL, this is a relative level of risk reduction provided by a safety function, or a specified target level of risk reduction. The requirements for a given safety integrity level are not consistent among all of the functional safety standards. In the European functional safety standards based on the International Electrotechnical Commission (IEC) 61508 standard[1], four safety integrity levels are defined, with safety integrity level 4 being the most dependable, and safety integrity level 1 being the least dependable. A safety integrity level is determined based on a number of quantitative factors in combination with qualitative factors such as the development process and life cycle management.

sequential logic. In digital circuit theory, this is a type of logic circuit whose output depends not only on the present value of its input signals but also on the past history of its inputs.

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 as it is a 'soft error', effecting a state change in the logic node or memory bit, but not permanently damaging the circuit.

single point of failure or single point of vulnerability. A potential risk posed by a flaw in the design, implementation or configuration of a circuit or system in which one fault or malfunction causes an entire system to stop operating correctly. It is undesirable in any system with the goal of high availability and/or reliability.

soft intellectual property (IP) core. An IP core that is in the form of a netlist or HDL. A soft IP core requires verification of a function following implementation (synthesis and/or place and route), unlike a hard IP core.

stress testing. A software testing activity that determines the robustness of software by testing beyond the limits of normal operation. Stress testing is particularly important for 'mission critical' software, but is used for all types of software. Stress tests commonly put a greater emphasis on robustness, availability and error handling under a heavy load, than on what would be considered correct behaviour under normal circumstances.

state machine. See finite state machine.

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 RTL — for example, in an 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.

---

[1] INTERNATIONAL ELECTROTECHNICAL COMMISSION, Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems, IEC Standard 61508, IEC, Geneva (2010).

toolset. A ‘package’ or ‘set’ of tools used in electronic design of a PLD. Circuit vendors typically provide toolsets that are 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 as to support design and verification and validation.

Verilog. An HDL used to model electronic systems at the RTL. Verilog is described in Institute of Electrical and Electronics Engineers (IEEE) standard 1364-2005[2].

very high speed integrated circuit hardware description language (VHDL). A language used to model electronic systems at the RTL. VHDL is described in IEEE standard 1076-2008[3].

[2] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, IEEE Standard Verilog Hardware Description Language, IEEE Standard 1364-2005, IEEE, New York (2005).

[3] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, IEEE Standard VHDL Language Reference Manual, IEEE Standard 1076-2008, IEEE, New York (2008).

ABBREVIATIONS

ABWR	advanced boiling water reactor
ALS	advanced logic system
ASIC	application specific integrated circuit
BWR	boiling water reactor
CCF	common cause failure
CIM	component interface module
CLB	configurable logic block
CPLD	complex programmable logic device
CPU	central processing unit
CRC	cyclic redundancy check
DAS	diverse actuation system
DCC	digital control computer
DPS	diverse protection system
EEPROM	electrically erasable programmable read only memory
EMC	electromagnetic compatibility
EMI	electromagnetic interference
EPRI	Electric Power Research Institute
EQ	equipment qualification
ESFAS	engineered safety feature actuation system
FFF	form, fit and function
FIT	failure in time
FPGA	field programmable gate array
HDL	hardware description language
HPD	hardware description language programmed device
HSI	human system interface
I&C	instrumentation and control
ICP	independent control platform
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
I/O	input/output
IP	intellectual property
JTAG	Joint Test Action Group
LUT	look-up table
MDEP	Multinational Design Evaluation Program
NPL	non-programmable logic
NRC	Nuclear Regulatory Commission
O&M	operation and maintenance
PACS	priority actuation and control system
PAL	programmable array logic
PCB	printed circuit board
PDB	predeveloped block
PLC	programmable logic controller
PLD	programmable logic device
PLS	plant control system
PMS	protection and safety monitoring system
PRPS	primary reactor protection system
PWR	pressurized water reactor
RCS	rod control system
RPS	reactor protection system
RTL	register transfer level
RTS	reactor trip system

SEU	single event upset
SRAM	static random access memory
TG-FAN	Topical Group on Field Programmable Gate Array Applications in Nuclear Power Plants
V&V	verification and validation
VHDL	very high speed integrated circuit hardware description language

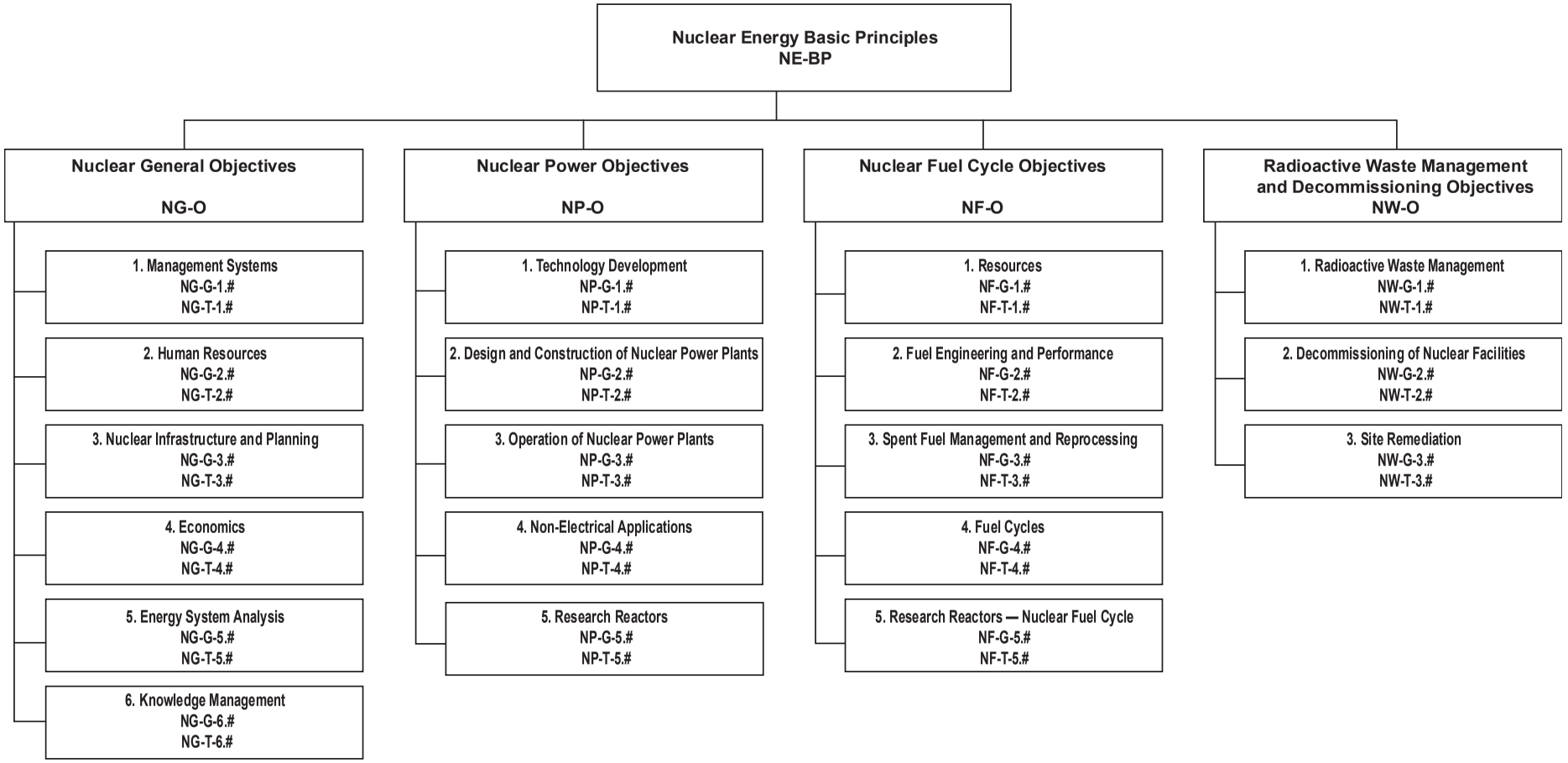
CONTRIBUTORS TO DRAFTING AND REVIEW

Andrashov, A.	Research and Production Corporation Radiy, Ukraine
Arndt, S.	Nuclear Regulatory Commission, United States of America
De Grosbois, J.	International Atomic Energy Agency
Eiler, J.	International Atomic Energy Agency
Gassino, J.	Institute for Radiological Protection and Nuclear Safety, France
Glockler, O.	SunPort SA, Switzerland
Hai, Z.	State Nuclear Power Automation System Engineering Company, China
Naser, J.	Electric Power Research Institute, United States of America
Nguyen, T.	Électricité de France, France
Russomanno, S.	Global Nuclear Solutions Inc., Canada
Seaman, S.	Westinghouse Electric Company, United States of America

Consultants Meetings

Vienna, Austria: 11–14 February 2013, 17–21 March 2014

Structure of the IAEA Nuclear Energy Series



Key

BP: Basic Principles
O: Objectives
G: Guides
T: Technical Reports
Nos 1-6: Topic designations
#: Guide or Report number (1, 2, 3, 4, etc.)

Examples

NG-G-3.1: Nuclear General (**NG**), Guide, Nuclear Infrastructure and Planning (topic **3**), **#1**
NP-T-5.4: Nuclear Power (**NP**), Report (**T**), Research Reactors (topic **5**), **#4**
NF-T-3.6: Nuclear Fuel (**NF**), Report (**T**), Spent Fuel Management and Reprocessing (topic **3**), **#6**
NW-G-1.1: Radioactive Waste Management and Decommissioning (**NW**), Guide, Radioactive Waste (topic **1**), **#1**

IAEA
International Atomic Energy Agency

No. 24

ORDERING LOCALLY

In the following countries, IAEA priced publications may be purchased from the sources listed below or from major local booksellers.
Orders for unpriced publications should be made directly to the IAEA. The contact details are given at the end of this list.

BELGIUM

Jean de Lannoy
Avenue du Roi 202, 1190 Brussels, BELGIUM
Telephone: +32 2 5384 308 • Fax: +32 2 5380 841
Email: jean.de.lannoy@euronet.be • Web site: http://www.jean-de-lannoy.be

CANADA

Renouf Publishing Co. Ltd.
22-1010 Polytek Street, Ottawa, ON K1J 9J1, CANADA
Telephone: +1 613 745 2665 • Fax: +1 643 745 7660
Email: order@renoufbooks.com • Web site: http://www.renoufbooks.com

Bernan Associates
4501 Forbes Blvd., Suite 200, Lanham, MD 20706-4391, USA
Telephone: +1 800 865 3457 • Fax: +1 800 865 3450
Email: orders@bernan.com • Web site: http://www.bernan.com

CZECH REPUBLIC

Suweco CZ, s.r.o.
SESTUPNÁ 153/11, 162 00 Prague 6, CZECH REPUBLIC
Telephone: +420 242 459 205 • Fax: +420 284 821 646
Email: nakup@suweco.cz • Web site: http://www.suweco.cz

FRANCE

Form-Edit
5 rue Janssen, PO Box 25, 75921 Paris CEDEX, FRANCE
Telephone: +33 1 42 01 49 49 • Fax: +33 1 42 01 90 90
Email: fabien.boucard@formedit.fr • Web site: http://www.formedit.fr

Lavoisier SAS
14 rue de Provigny, 94236 Cachan CEDEX, FRANCE
Telephone: +33 1 47 40 67 00 • Fax: +33 1 47 40 67 02
Email: livres@lavoisier.fr • Web site: http://www.lavoisier.fr

L'Appel du livre
99 rue de Charonne, 75011 Paris, FRANCE
Telephone: +33 1 43 07 43 43 • Fax: +33 1 43 07 50 80
Email: livres@appeldulivre.fr • Web site: http://www.appeldulivre.fr

GERMANY

Goethe Buchhandlung Teubig GmbH
Schweitzer Fachinformationen
Willstätterstrasse 15, 40549 Düsseldorf, GERMANY
Telephone: +49 (0) 211 49 874 015 • Fax: +49 (0) 211 49 874 28
Email: s.dehaan@schweitzer-online.de • Web site: http://www.goethebuch.de

HUNGARY

Librotrade Ltd., Book Import

Pesti ut 237. 1173 Budapest, HUNGARY
Telephone: +36 1 254-0-269 • Fax: +36 1 254-0-274
Email: books@librotrade.hu • Web site: http://www.librotrade.hu

INDIA

Allied Publishers

1st Floor, Dubash House, 15, J.N. Heredi Marg, Ballard Estate, Mumbai 400001, INDIA
Telephone: +91 22 4212 6930/31/69 • Fax: +91 22 2261 7928
Email: alliedpl@vsnl.com • Web site: http://www.alliedpublishers.com

Bookwell

3/79 Nirankari, Delhi 110009, INDIA
Telephone: +91 11 2760 1283/4536
Email: bkwell@nde.vsnl.net.in • Web site: http://www.bookwellindia.com

ITALY

Libreria Scientifica "AEIOU"

Via Vincenzo Maria Coronelli 6, 20146 Milan, ITALY
Telephone: +39 02 48 95 45 52 • Fax: +39 02 48 95 45 48
Email: info@libreriaaeiou.eu • Web site: http://www.libreriaaeiou.eu

JAPAN

Maruzen Co., Ltd.

1-9-18 Kaigan, Minato-ku, Tokyo 105-0022, JAPAN
Telephone: +81 3 6367 6047 • Fax: +81 3 6367 6160
Email: journal@maruzen.co.jp • Web site: http://maruzen.co.jp

RUSSIAN FEDERATION

Scientific and Engineering Centre for Nuclear and Radiation Safety

107140, Moscow, Malaya Krasnoselskaya st. 2/8, bld. 5, RUSSIAN FEDERATION
Telephone: +7 499 264 00 03 • Fax: +7 499 264 28 59
Email: secnrs@secnrs.ru • Web site: http://www.secnrs.ru

UNITED KINGDOM

The Stationery Office Ltd. (TSO)

St. Crispins House, Duke Street, Norwich, NR3 1PD, UNITED KINGDOM
Telephone: +44 (0) 333 202 5070
Email: customer.services@tso.co.uk • Web site: http://www.tso.co.uk

UNITED STATES OF AMERICA

Bernan Associates

4501 Forbes Blvd., Suite 200, Lanham, MD 20706-4391, USA
Telephone: +1 800 865 3457 • Fax: +1 800 865 3450
Email: orders@bernan.com • Web site: http://www.bernan.com

Renouf Publishing Co. Ltd.

812 Proctor Avenue, Ogdensburg, NY 13669-2205, USA
Telephone: +1 888 551 7470 • Fax: +1 888 551 7471
Email: orders@renoufbooks.com • Web site: http://www.renoufbooks.com

Orders for both priced and unpriced publications may be addressed directly to:

IAEA Publishing Section, Marketing and Sales Unit
International Atomic Energy Agency
Vienna International Centre, PO Box 100, 1400 Vienna, Austria
Telephone: +43 1 2600 22529 or 22530 • Fax: +43 1 2600 29302
Email: sales.publications@iaea.org • Web site: http://www.iaea.org/books

15-45971

INTERNATIONAL ATOMIC ENERGY AGENCY
VIENNA
ISBN 978–92–0–103515–8
ISSN 1995–7807