Proceedings of ICONE19
19th International Conference on Nuclear Engineering
May 16-19, 2011, Chiba, Japan
ICONE19-43169

VERIFICATION OF FPGA-BASED NPP I&C SYSTEMS: GENERAL APPROACH AND TECHNIQUES

Anton Andrashov
Centre for Safety Infrastructure-Oriented Research and Analysis
a.andrashov@csis.org.ua

Vyacheslav Kharchenko
Centre for Safety Infrastructure-Oriented Research and Analysis
v.kharchenko@khai.edu

Volodymir Sklyar
Centre for Safety Infrastructure-Oriented Research and Analysis
vvsklyar@mail.ru

Alexander Siora
Research and Production Corporation "Radiy"
marketing@radiy.com

Lubov Reva
Centre for Safety Infrastructure-Oriented Research and Analysis
l.reva@csis.org.ua

Application of FPGA technology for implementation of digital safety-critical systems turns the attention of more and more audience around the globe. This trend is fully supported by the nuclear community where all major players of industry (i.e. vendors, regulators and academia) have responded on the technology.

Though there is a list of approved nuclear applications of FPGAs, but there is still no consensus on what FPGA is and how to apply it for NPP I&C systems in a safe manner. This puts some uncertainties into a processes, products (FPGA components, tools etc.) and assessment methods used to support new technology in a real world (Kharchenkoet al., 2008).

Therefore implementation of design and V&V activities for FPGA-based I&C systems and for safety critical systems as a whole needs special attention at all the stages of their lifecycles. It is very important to identify faults and design errors at initial phases of the lifecycle (Bakhmach et al., 2009).

This paper presents a common approach and techniques for design and verification of Field Programmable Gates Arrays (FPGA)-based Instrumentation and Control (I&C) systems for Nuclear Power Plants (NPP).

Appropriate regulatory documents used for I&C systems design and verification are discussed taking into account latest international standards and guidelines. Typical design and implementation flow for FPGA electronic designs, including such activities as design description, logic synthesis, placement & routing and bitstream generation, is presented. Some safety-related features of implementation process are discussed. Corresponding technical documents, related to design and implementation processes are outlined.

An approach for verification of FPGA electronic design algorithms, used in FPGA-based I&C systems, is proposed. Testing and invariant-oriented techniques for assessment of FPGA-based I&C systems are described.

During testing, the states («input signals») that correspond to both normal mode and each condition of output signals forming are in-turn simulated at inputs of algorithms' software model, implemented in Integrated Development Environment (IDE). Appropriate changes of output states in software model of the algorithms and/or in preset checkpoints inside the model are observed using a monitor of IDE and recorded as «hardcopies» of the screen.

An example of test-based verification for FPGA-based Reactor Trip System for PWR-type nuclear reactor control logic algorithm is outlined.

Invariant-oriented technique is based on the concept of invariants, which are rules, properties, relations of I&C system that stay unchangeable (and inviolate) during development and operation with all the possible input values. Variety of invariant types includes the following: invariants that are based on features and constraints of VHDL; invariants for timing charts; invariants for diagrams of control algorithms and automaton models.

In future, to increase verification quality, we plan to use various tools for multi-version testing of FPGA electronic design algorithms and FPGA-based I&C systems as a whole. Appropriate tools reducing time to generate test cases, as well as templates for functional units of the algorithms are under development.

REFERENCES

Kharchenko, V., et al., FPGA-based NPP I&C Systems: Development and Safety Assessment, 2008, RPC Radiy, National Aerospace University KhAI, SSTC on Nuclear and Radiation Safety, 188 p.

Bakhmach, I., et al., Advanced I&C Systems for NPPS Based on FPGA Technology: European Experience, 2009, ICONE-17.

Proceedings of ICONE19
19th International Conference on Nuclear Engineering
May 16-19, 2011, Chiba, Japan

ICONE19-43169

VERIFICATION OF FPGA-BASED NPP I&C SYSTEMS: GENERAL APPROACH AND TECHNIQUES

Anton Andrashov
Centre for Safety Infrastructure-Oriented Research and Analysis
Ukraine, Kharkiv, 37 Astronomichna str.
+380503931043
a.andrashov@csis.org.ua

Vyacheslav Kharchenko
Centre for Safety Infrastructure-Oriented Research and Analysis
Ukraine, Kharkiv, 37 Astronomichna str.
+380679151989
v.kharchenko@khai.edu

Volodymir Sklyar
Centre for Safety Infrastructure-Oriented Research and Analysis
Ukraine, Kharkiv, 37 Astronomichna str.
+380999370183
vvsklyar@mail.ru

Alexander Siora
Research and Production Corporation "Radiy"
Ukraine, Kirovograd, 29 Geroyev Stalingrada str.
+380522373020
marketing@radiy.com

Lubov Reva
Centre for Safety Infrastructure-Oriented Research and Analysis
Ukraine, Kharkiv, 37 Astronomichna str.
+380664417898
l.reva@csis.org.ua

Keywords: NPP I&C systems, FPGA, standard, verification and validation, invariant-oriented assessment, reactor trip system, case study.

ABSTRACT

This paper presents a general approach and techniques for design and verification of Field Programmable Gates Arrays (FPGA)-based Instrumentation and Control (I&C) systems for Nuclear Power Plants (NPP).

Appropriate regulatory documents used for I&C systems design, development, verification and validation (V&V) are discussed considering the latest international standards and guidelines.

Typical development and V&V processes of FPGA electronic design for FPGA-based NPP I&C systems are presented. Some safety-related features of implementation process are discussed. Corresponding development artifacts, related to design and implementation activities are outlined.

An approach to test-based verification of FPGA electronic design algorithms, used in FPGA-based reactor trip systems is proposed. The results of application of test-based techniques for assessment of FPGA electronic design algorithms for reactor trip system (RTS) produced by Research and Production Corporation (RPC) "Radiy" are presented. Some principles of invariant-oriented verification for FPGA-based safety-critical systems are outlined.

1. INTRODUCTION

1.1 Motivation

Application of FPGA technology for implementation of digital safety-critical systems turns the attention of more and more audience around the globe. This trend is fully supported by the nuclear community where all major players of industry (i.e. vendors, regulators and academia) have responded on the technology.

Though there is a list of approved nuclear applications of FPGAs, but there is still no consensus on what FPGA is and how to apply it for NPP I&C systems in a safe manner. This puts some uncertainties into a processes, products (FPGA components, tools etc.) and assessment methods used to support new technology in a real world (Kharchenkoet al., 2008).

Therefore implementation of design and V&V activities for FPGA-based I&C systems and for safety critical systems as a whole needs special attention at all the stages of their lifecycles. It is very important to identify faults and design errors at initial phases of the lifecycle (Bakhmach et al., 2009).

1.2 Goal and structure

The goal of the paper is to present some results related with the implementation of design and V&V activities for FPGA-based NPP I&C systems producing. Stated approach and technique are based on development and

licensing experience and used by RPC Radiy (Kirovograd, Ukraine).

Special attention is paid to general V&V approach and some particular testing techniques.

The paper is structured as following. Section 2 discusses regulatory basis of FPGA technology for NPP I&C systems. Section 3 describes development and V&V processes for FPGA electronic design of NPP I&C systems. Principles, criteria and methods for test-based verification of FPGA electronic design algorithms for RTS are described in Section 4. Besides, an example of test-based verification for RTS algorithm is outlined. Section 5 includes propositions concerning some principles of invariant-oriented verification for FPGA-based safety-critical systems. Finally, Section 6 concludes the paper and presents directions of future researches.

2. REGULATORY BASIS OF FPGA TECHNOLOGY FOR NPP I&C SYSTEMS

International standards form the basis containing the requirements for implementation of all basic processes for FPGA-based NPP I&C systems. The risks caused by the usage of FPGA-based NPP I&C systems are the reason why the requirements for these systems are so strict. One of the indispensable conditions for NPP I&C systems operation (licensing) is to comply with the requirements of the regulatory documents.

Following standards and guidelines should be considered when dealing with FPGA-based NPP I&C systems:

- IEC 62566 CDV «Nuclear Power Plants–Instrumentation and control important to safety–Development of HDL-programmed integrated circuits for systems performing category A functions»;
- IEC 61508: 2010 ed. 2.0 (7 parts) «Functional safety of electrical/electronic/programmable electronic safety-related systems»;
- EPRI TR 1019181 «Guidelines on the Use of Field Programmable Gate Arrays in Nuclear Power Plant I&C Systems»;
- NUREG/CR-7006 «Review Guidelines for Field-Programmable Gate Arrays in Nuclear Power Plant Safety Systems».

2.1 IEC 62566 CDV

IEC 62566 is rapidly evolving standard (3 intermediate versions have been issued during 2010, all of them had a different titles. French committee made a proposal for a standard dedicated to HDL-programmed integrated circuits (HC) in 2006. Opinion leaders on behalf of nuclear industry, regulatory bodies and academia are taking part in document development. At the moment it has a status of committee draft version (CDV) and its approval is planed in 2011.

IEC 62566 includes information regarding terms and definitions and provides general requirements on:

- dedicated development life-cycle addressing each phase of the development of HCs, including specification of requirements, design, implementation, verification, integration and validation;
- planning and other complementary activities such as modification and manufacturing;
- selection of pre-developed components. This includes micro-electronic resources (such as a blank FPGA) and HDL statements representing pre-developed blocks (IP-cores);
- design and implementation according to simplicity and determinism principles, recognized to be of first importance to achieve “fault free” implementation of category A functions;
- tools used to design, implement and verify HCs.

Moreover IEC 62566 contains language and coding rules to be followed while using HDL language.

2.2 IEC 61508: 2010 ed. 2.0

The first edition of IEC 61508 was issued during 1998-2000. New (second) edition of IEC 61508 have been approved in 2010 and for the first time it covers FPGAs. This document includes 7 parts within 594 pages. IEC 61508 is an umbrella standard which addresses functional safety as a whole and has to be allied together with domain-specific standards i.e. IEC 61513 for nuclear domain, IEC 61511 for process industry. But it is approved by the most part of nuclear community. In some countries (for example, Canada) IEC 61508 is mandatory for NPP I&C systems.

IEC 61508 is based on SIL1-SIL4 conceptions for specifying the safety integrity requirements of the safety functions. Each SIL level has corresponding quantitative requirements to probabilistic failure performance indexes. Functional safety in terms of IEC 61508 could be expressed as automatic safety function shall perform the intended function correctly or the system shall fail in a predictable (safe) manner.

IEC 61508 has the following structure:

- Part 1: General requirements;
- Part 2: Requirements for electrical/electronic/programmable electronic safety-related systems;
- Part 3: Software requirements;
- Part 4: Definitions and abbreviations;
- Part 5: Examples of methods for the determination of safety integrity levels;
- Part 6: Guidelines on the application of IEC 61508-2 and IEC 61508-3;
- Part 7: Overview of techniques and measures.

Parts 1-4 are normative meaning the requirements must be met for compliance with the standard. Parts 5-7 are informative meaning that they provide examples, guidelines, techniques and measures but do not mandate the use of any specific guidelines, techniques or measures to be in compliance.

The normative parts of the standard have thousands of requirements, i.e., sentences including the term “shall” or “must” which need to be correctly addressed for compliance with the standard. Broadly speaking, these requirements fall into one of two groups which relate directly to the two fundamental concepts of IEC 61508.

One group of requirements covers the design lifecycle process. This is intended to provide a sufficient level of measures against “systematic” or software failures of the system, i.e. “fault avoidance”.

Another group of requirements covers the probabilistic analysis of all the hardware involved in any safety function.

This is intended to provide a sufficient level of integrity against "random" or hardware failures of the system.

2.3 EPRI TR 1019181

EPRI TR 1019181 was issued in December 2009. EPRI TR 1019181 was developed by international team of experts leaded by Joseph Naser. This report is intended to help the end users (i.e. utilities, system integrators) to reduce technical and regulatory risks and to reduce costs for digital I&C retrofits and new plant designs. EPRI TR 1019181 consists of 8 sections and 2 appendixes within 213 pages.

Section 1 outlines introduction to the report including objectives of the report, approach used by authors to conduct the research, etc.

Sections 2 and 3 of EPRI TR 1019181 focused on the FPGA technology. Section 2 begins with a primer on FPGAs presenting following information:

- FPGAs architecture and technology;
- CPLDs architecture and technology;
- comparison of FPGAs and CPLDs;
- FPGA programming process;
- tools;
- pre-developed hardware designs, etc.

Section 3 discusses advantages and limitations of FPGAs as compared to other technologies.

Section 4 provides guidelines for designing an FPGA application from a utility viewpoint. The following design aspects are considered:

- types of NPP applications that are the most suitable for the use of FPGAs;
- planning a modification involving FPGA technology;
- acceptance qualification tests and analyses;
- licensing, etc.

Section 5 discusses FPGA design and implementation processes in details subject to FPGA application designer's point of view. It includes guidance on all major items relevant to design and verification of FPGA applications.

Section 6 describes how regulators have approached the review of FPGA-based systems as well as important elements of the safety case based on past licensing reviews. The approach of US NRC to review of FPGA-based NPPs I&C applications is highlighted as example. Given information and guidance focused on FPGA specific aspects of regulatory reviews. Items such as equipment qualification, for example, are not specifically addressed as these were not found to have any particular differences owing to use of FPGAs versus other technologies.

Section 7 provides a summary and conclusion regarding the differences between FPGAs and other electronic technologies in terms of key attributes important to NPP I&C applications.

Section 8 includes references, glossary of terms and a list of acronyms used in the report.

Appendix A provides extensive list of examples of FPGAs application in NPP I&C systems from the nuclear industry worldwide:

- example A1 Wolf Creek plant – USA main steam and feedwater isolation system replacement;
- example A2 Darlington plant – Canada digital control computer upgrade;
- example A3 Advanced boiling water reactor plant – Japan power range neutron monitoring system;
- example A4 Kozloduy units 5 and 6 – Bulgaria modernization of engineered safety features actuation systems;
- example A5 EDF 900 Mw series – France rod control system – slave logic units.

Appendix B to this report provides detailed examples and case studies on insights and lessons learned from previous applications of FPGAs in non-nuclear domains.

2.4 NUREG/CR-7006

NUREG/CR-7006 was published in February 2010. It was prepared by Oak Ridge National Laboratory and University of Tennessee on the request of US NRC. NUREG/CR-7006 is aimed on helping NRC staff to conduct a review of FPGA-based NPP I&C systems by representing a compilation of safe FPGAs design practices. Another, vendor-oriented goal of this guide is to provide a criterion for establishing of specific design and V&V processes related to FPGAs which could be accepted by the regulator during licensing. NUREG/CR-7006 consists of 5 chapters and 2 appendixes within 94 pages.

Chapter 1 describes background of the research and document structure.

Chapter 2 discusses the safety-critical issues related to the hardware of FPGA design for NPP I&C systems. FPGA hardware design practices presented in the chapter are divided into two groups: the board-level design practices and FPGA internal logic design practices.

Chapter 3 provides information on design entry methods for FPGA based NPP I&C systems in terms of application of HDL languages. It outlines best practice (code examples) of FPGA electronic design's components implementation using VHDL and Verilog languages. The examples cover the most commonly used HDL structures such as state machines, multiplexers, decoders, read-only memory, random access memory, first-in-first-out memory, etc.

Chapter 4 identifies FPGA-specific design life cycle and the appropriate design flow through the life cycle. The design flow includes a number of sequential design steps starting from design requirements accompanied with their corresponding V&V steps. Every step is described in detail and necessary output documentation is specified. Selection of FPGA components and use of design tools are also discussed in this chapter.

Chapter 5 presents conclusions postulated main attributes and areas of concerns related to FPGA-based NPP I&C development and licensing.

Appendixes provide the list of references concerning FPGA-related published materials (including standards, guides, papers, etc.) and their summary.

3. DEVELOPMENT AND V&V PROCESSES OF FPGA ELECTRONIC DESIGN FOR FPGA BASED NPPs I&C SYSTEMS

3.1 Scheme of development and V&V processes

The main processes of FPGA-based I&C systems lifecycle are development and V&V processes. This section will

discuss the main steps and features of FPGA electronic design development and V&V.

The development process consists of design and implementation phases. The FPGA electronic design integration is a final phase of development process.

V&V process supports each step of development process (the outputs of the step) with appropriate checks and analysis.

The development and V&V processes of FPGA electronic design for FPGA-based NPPs I&C systems are shown at Fig. 1.

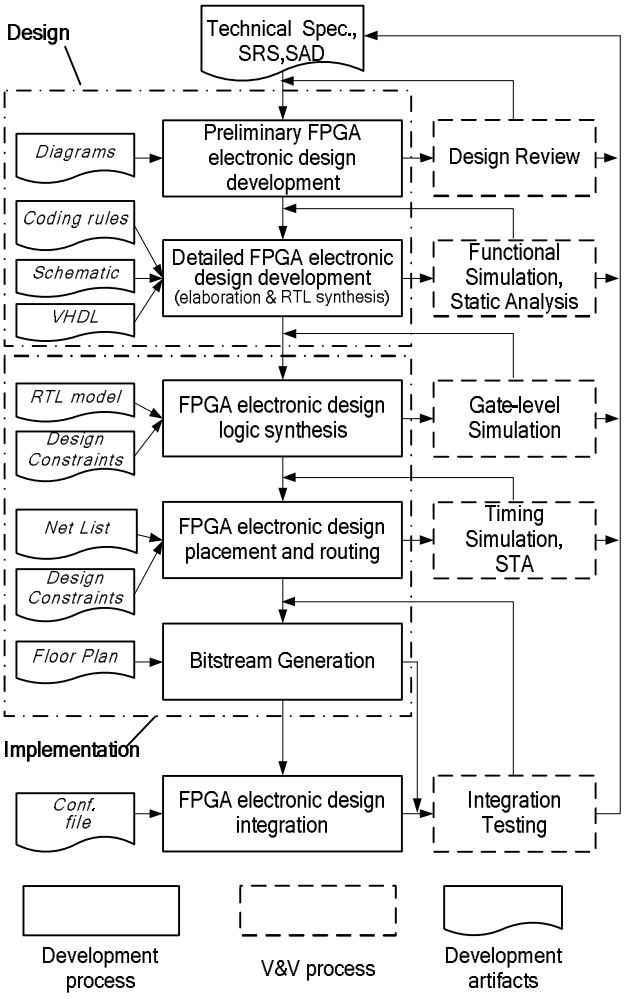

Fig. 1 FPGA electronic design development and V&V processes

Before the starting of FPGA electronic design development the following artifacts shall be prepared and reviewed :

- Technical Requirement Specification (Technical Spec.) – should describe all customer, commercial goals and technical goals for the system development;
- Safety Requirements Specification (SRS) - should identify all safety-related requirements of the system, including requirements from appropriate nuclear standards and basic safety standards such as IEC 61508, and from the specific safety requirements for the intended application(s);
- System Architecture Description (SAD) - should provide an overview of the system and should allocate the requirements to the different subsystems (e.g., hardware, software, and FPGA).

All this artifacts shall be reviewed and approved before getting down to the subsystem (component) level. Later they shall be verified after subsystems integration.

3.2 Design development

The initial and the most critical phase of the development process is design development, which includes preliminary or architectural design and detailed design.

Preliminary design should define all functional blocks, their interfaces and other information required for the next phase. At this step such important criteria as reliability, design traceability and design verifiability should be taken into account. Textual or/and graphical description (diagrams) of design partitioning and other design requirements is the result of preliminary development. Upon completion of this design activity the design review should be performed. The result of the review may require a different design partitioning or correction of the initial requirements.

Detailed design refines preliminary design into FPGA electronic design behavior description. Detailed design should implement the functionality of the FPGA electronic design. The design entry typically used to implement detailed design are HDL coding (VHDL, Verilog) and schematic representation. Detailed design is finalized by elaboration of FPGA electronic design components (assembles all the files that compose the design, and performs some checks) and RTL model synthesis. In RTL model, a circuit's behavior is defined in terms of the flow of signals (or data transfer) between hardware registers, and the logical operations performed on those signals.

One of the safety related feature connected with detailed design development is coding rules application. 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 final state machines; - naming convention; - coding style, etc.

Coding rules shall require application of HDL languages to its specification, i.e. to comply with appropriate standards (for example, IEEE 1076). Vendor-specific coding guidance should be also considered.

Coding rules shall be documented for each specific project in form of quality procedure. Also they shall be kept under configuration management control.

To verify detailed design outputs following techniques may be used:

- functional simulation;
- static analysis.

Functional simulation also referred as behavior simulation is used to verify the behavior 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 analysis aims for detection of coding rules violations. It can be implemented using special lint tools, for example Alint by Aldec. Such tolls allow performing in automated mode a source code check to detect deviations from coding

rules. It is possible to develop a library for a lint tool containing project-specific coding rules and apply it when necessary.

3.3 Implementation and integration

Implementation is the next important phase, which comprises logic synthesis, placement and routing and bit stream generation. The appropriate procedures of V&V are connected with each activity of implementation phase.

The first step of FPGA electronic design implementation is FPGA electronic design logic synthesis. During logic synthesis the synthesizer transforms RTL model of FPGA electronic design into gate\cell level scheme (Net List). Most synthesizers generate FPGA-independent schematic representation of the RTL 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. EDIF) or graphical files.

The synthesizer may apply different kinds of optimizations which could be defined in terms of design constraints. Design constraints basically could affect the following attributes of FPGA electronic design:

- logic synthesis; - timing characteristics; - pins assignment and adjustment; - topology of FPGA electronic design 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 electronic design by adding them to a project configuration. The syntaxes used for defining design constraints 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.

It looks appropriate for each specific project (group of projects for I&C platform), explicitly, in the way of quality procedure defining how to handle the design constraints. To support the development of such procedures vendor's guides and recommendations (Altium, 2008; Actel, 2010) could be used, but comprehensive understanding of each constraint effect by design team is required.

The verification process at this level consists of gate-level simulation, which is technology-dependent in contrast to functional simulation. But the input stimuli and expected outputs should be the same. Any differences must be justified. During gate-level simulation timing characteristics for FPGA electronic design are based on assumed gate and routing delays since the design has not yet been placed or routed.

After the logic synthesis FPGA electronic design placement and routing is carried out. It is tool-driven process that determines where registers and gates determined in Net List will be placed within an FPGA chip. This process also determines the connection paths between design elements. The resulting design connectivity is defined by the Floor Plan.

The place and route tool also generates a timing file that is more accurate than the one produced by synthesis, since it also includes timing associated with routing. Design constraints should be considered for placement and routing as well as for logic synthesis.

Generated Floor Plan can be verified by timing simulation. The input stimuli are the same as for two previous types of simulation. But timing simulation is important because of its exact timing analysis. In addition, information such as fan-out and delays for each connection path can be derived using the static timing analysis (STA). Thus, the purpose of this verification procedure is to define that all timing requirements are met.

The last step of the implementation phase is bitstream generation. The output of this phase is the configuration file, which can be implemented into FPGA chip. It contains all data required to configure FPGA chip.

The verification of configuration file is conducted after FPGA electronic design integration into FPGA chip.

During FPGA electronic design integration phase the configuration files, which have been derived at the previous step, are being downloaded to FPGA chip. Special hardware such as configuration interfaces (i.e. JTAG) are required to download configuration file into FPGA chip. Some FPGA chips and appropriate tools provide automatic checking of integration correctness.

Integration testing is indented for justification that FPGA electronic design inside the chip performs to its specification and system architecture.

During integration testing FPGA chip with integrated FPGA electronic design is on the board for which it was developed. The inputs of the board are connected to a special testbench which feeds them with input signals in accordance with testing stimuli. The outputs of the board are connected to data acquisition system which collects the response of the board on input stimuli. Output signals (response) are analyzed in accordance with pass\fail criteria.

4. TEST-BASED VERIFICATION OF REACTOR TRIP SYSTEM

4.1 The FPGA-based reactor trip system

RPC "Radiy" has designed, developed and installed more than 20 RTSs for VVER-type reactor (see Fig. 2) using FPGA-based platform RADIY^{TM}. Nuclear reactor trip systems are the most important for safety of NPP I&C systems (Yastrebenetsky, 2004).

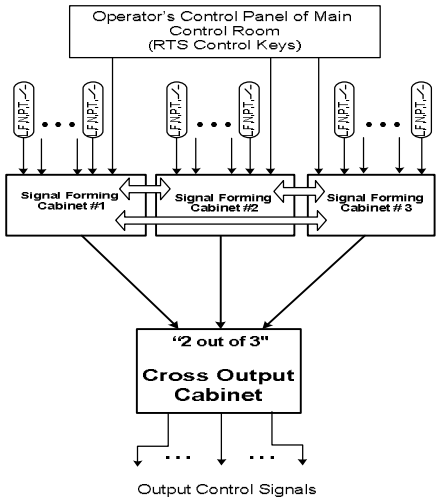

Fig. 2 A Structure of FPGA-based reactor trip system

The FPGA-based reactor trip system consists of three signal forming cabinets which receives and processes independent channelized signals from plant instrumentation and provides output signals to voting logic in the cross output cabinet.

Logic modules in the signal forming cabinets communicate with operator interface and alarming systems in main control room and emergency control room. Status and diagnostic information is sent by diagnostic module in signal forming cabinets to engineering workstation.

There is a standard set of signal forming algorithms (control logic) for reactor trip systems. Various modifications (completion and deletion of elements, updating of various parameters) of the algorithms occur during development of such systems; the modifications are defined by particular requirements specification.

However, in whole, the basis of a system is stable. Thus, it is important to determine the methods and techniques of verification for signal forming algorithms in a trip system that would confirm their compliance with the specification. These techniques shall consider some features of reactor trip systems, specifically, their standard operation algorithms, data capacity, etc.

4.2 Principles of FPGA electronic design algorithms verification

Testing is one of the basic verification methods. During verification of FPGA electronic design testing is implemented via functional, gate-level and timing simulation of the design in integrated design environment (Munden, 2005).

During the testing, the input stimuli («input signals»), corresponding to both normal mode and each condition of output signals conditioning, should be in-turn simulated at inputs of algorithms' model, implemented in integrated design environment. Appropriate changes in output states of software model of the algorithms and/or in preset checkpoints inside the model are observed using a monitor of IDE and recorded as «hardcopies» of the screen or in a dedicated file.

Testing of algorithms should be implemented in compliance with some set of rules, known as principles of testing. These principles are based on the following regulations:

- initial set of algorithms being classified using complexity criterion and divided into appropriate groups;
- for testing purposes, a complex of standard methods for testing being formed subject to different nature and complexity level of the groups;
- the main criterion for testing is test coverage criterion; the key component of each testing plan is a method that assure the compliance with the coverage criterion;
- each algorithm requires a policy for testing, i.e. it is necessary to define a set and the order for different test methods application; it allows to achieve hundred-per-cent or acceptable (maximum possible from the practical point of view) coverage;
- during development of testing plan it is necessary to explicitly mark the elements to be tested (components and their connections) that can be implemented during system validation only; a list of such elements should be considered during development of validation plans.

4.3 Classification of RTS algorithms

The following principles are the basis of classification:

- availability of memory elements in the algorithm (M attribute);
- complexity level (cascades) of the algorithm (C attribute).

The first principle divides the algorithms into combinatorial algorithms, i.e. that do not include memory elements, and algorithms with memory elements; the second principle divides the algorithms into few-cascade algorithms (attribute 1) and multi-cascade algorithms (attribute 2).

Few-cascade algorithms contain standard functional elements of middle complexity (comparators, majority elements, etc.) that form single cascade. Second cascade can be formed by inverters only. Such algorithms can be tested by testing of the correctness for standard function execution. Multi-cascade algorithms implement two and more cascades formed by the standard functional elements, or cascades formed by both standard functional elements and logic elements (AND, OR, etc.).

It is also possible to separate another group formed by some algorithms that contain delay elements if their application effects on time shift in output signals only, and does not effect on output logic.

Thus, every algorithm has one from the following attributes: C1 – combinatorial few-cascade algorithm; C2 – combinatorial multi-cascade algorithm; M1 – sequential few-cascade algorithm; M2 – sequential multi-cascade algorithm; C2D – conventionally combinatorial algorithm with delay element.

In general, RTS includes 34 algorithms. Table 1 presents a classification of RTS algorithms in accordance with the specified classification criteria.

Table 1. Classification of RTS algorithms

Attr.		C1	C2	C2D	M1	M2
Algorithms structure	A1	**V**				
	A2	**V**				
	A3		**V**			
	A4					**V**
	A5		**V**			
	A6				**V**	
	A7			**V**		
	A8		**V**			
	A9		**V**			
	A10					**V**
	A11		**V**			
	A12			**V**		
	A13		**V**			
	A14		**V**			
	A15				**V**	
	A16		**V**			
	A17				**V**	
	A18		**V**			
	A19			**V**		
	A20	**V**				
	A21	**V**				
	A22	**V**				
	A23	**V**				

Attr.		C1	C2	C2D	M1	M2
	A24		**V**			
	A25					**V**
	A26		**V**			
	A27	**V**				
	A28	**V**				
	A29		**V**			
	A30					
	A31	**V**				**V**
	A32	**V**				
	A33	**V**				
	A34	**V**				

4.4 Testing criteria and policy

Testing plan and test cases development is based on the coverage criterion. In addition, we need to consider time required for testing (time criterion). Coverage criterion is the main.

The assurance of compliance between the testing plan and coverage criterion is implemented by development of a test suites set, along with the explanation of hundred-per-cent or acceptable, subject to the algorithm being tested. Time criterion is complementary. During development of test suites, it is necessary to select the test variant that can assure required test coverage along with the less number of tests.

The best variant can be selected using coverage tables; such tables are applied in a classical approach of fault function tables and its modifications.

Testing policy defines a general approach to testing and includes the following elements.

1. Development of testing plan on the basis of object representation that is based on the algorithm analysis approach:- functional (testing of algorithm implementation subject to the specification);- circuit (testing of availability for each algorithm's element); - combined (the policy can vary inside the testing plan scope).

2. Defining the approach of specified coverage assurance subject to the task dimension (element capacity). This element of the policy is implemented by application of boundary testing methods, checkerboard methods, and others.

3. Defining the order for application of different coverage assurance methods subject to:

- peculiarities of testing;
- restrictions related with testing of some elements and functions during testing, as well as substantiated transferring of some functions to validation phase.

4.5 Methods and techniques for testing

The following are the main methods and techniques used during testing.

1. Single complex task can be reduced to a set of simple tasks in a case of multi-cascade algorithms. Such method can be implemented by:

- marking of iterative fragments (subdiagrams);
- decomposition of algorithmic diagrams into subdiagrams to represent the algorithm in aggregative form with further testing subject to already tested subdiagrams.

In this case testing includes two phases: testing of subdiagrams and testing of the whole algorithm subject to subdiagrams.

2. Marking and preliminary testing of complex functional units.

3. Application of methods, related with the technique of test impacts forming, which assure required test coverage subject to high dimension of the task (unit capacity, number of impact combinations).

3.1 Method of boundary values (consists of marking of key groups for functional units input values, which correspond to a set of equivalent states of the units).

3.2 Checkerboard method (based on application of two types of input impacts: 010101…01 and 101010…10).

4. Exhaustive search for all the possible combinations of input values.

4.6 Practical application of the testing technique

Further we present an example of the proposed testing principles. Fig. 3 depicts a standard algorithm of reactor trip system of VVER-type nuclear power reactor.

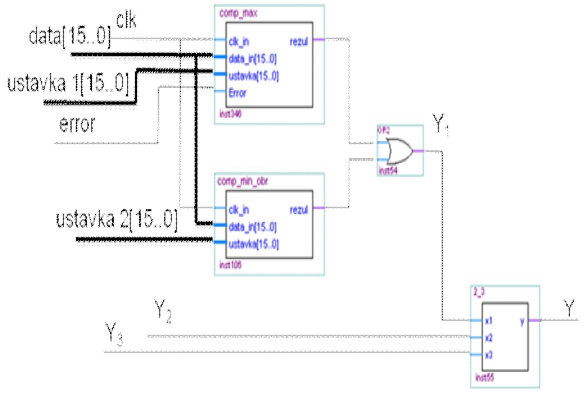

Fig. 3 A standard algorithm for reactor trip system

Given algorithm is of C2 group, since it does not contain memory elements and includes two cascades of functional elements. The algorithm forms a signal of reactor trip when the reactor core pressure exceeds 17,65 MPa, i.e. it monitors the range of acceptable values for one technological process variable of nuclear reactor (pressure) using «2 out of 3» majority voting.

Testing of the algorithm can be implemented using the method of boundary values for comp_min_obr and comp_max elements, as well as for exhaustive search for «2 out of 3» majority element. Thus, test coverage criterion in this case is exhaustion of 100% input values combinations based on the method of boundary values.

According to the principles of testing we should check this algorithm using the next groups of input stimuli:

1. Forming of "0" at comp_min_obr comparator output, testing of signal propagation from comp_max:

- Operation within inert zone;
- Actuation when setpoint value is exceeded;
- Transition to zero state when the value is below the setpoint;
- Actuation on error signal;

- Out of range.

2. Forming of «0» at comp_max comparator output, testing of signal propagation from comp_min_obr:

- Operation when setpoint value is exceeded (or equal);
- Operation when setpoint value is not reached.
- Out of range.

Table 2 presents the part of test cases for testing the algorithm.

Table 2. Test cases

Input signals						Output	
data	ustavka1	error	ustavka2	Y2	Y3	Y1	Y
Testing of signal propagation from comp_min_obr							
Operation when setpoint value is exceeded (or equal)							
10699	44040	1	10696	0	0	0	0
10698	44040	1	10696	0	1	0	0
10697	44040	1	10696	1	0	0	0
10696	44040	1	10696	1	1	0	1
Operation when setpoint value is not reached							
10695	44040	1	10696	0	0	1	0
10694	44040	1	10696	0	1	1	1
10693	44040	1	10696	1	0	1	1
10692	44040	1	10696	1	1	1	1

The results of functional simulation using ModelSim tool are presented in Fig. 4. Given diagram proves that the algorithm, which is implemented into design tool, meets specification requirements.

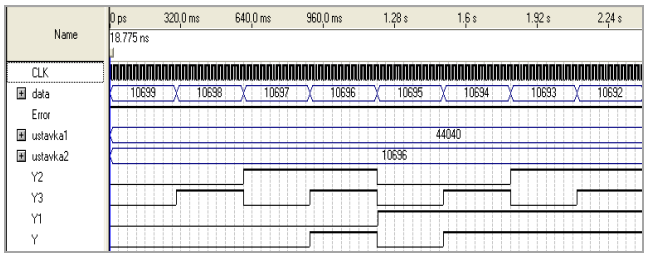

Fig. 4 Testing of signal propagation from comp_min_obr

5. INVARIANT-BASED VERIFICATION

In addition to testing of FPGA electronic design algorithms, it is possible to verify them using invariant-oriented approach. Such approach is a basis for independent verification and validation (IV&V). Requirement to IV&V is one from key requirements during development of nuclear reactor trip systems. Invariant-oriented IV&V includes three stages.

5.1 Profiling and checking of requirements

A goal of this stage is in forming of regulatory profile and testing of requirements to the structure and documentation composition of a system and FPGA electronic design (Andrashov et. al., 2010a):

- analysis of system documentation composition;
- analysis of FPGA electronic design documents structure;
- forming of regulatory profile for the design (specification for both regulatory (non-functional) and functional requirements);
- completion of documentation analysis checklist.

The documentation consists of: documents that establish requirements to both I&C system and engineering process; documents that establish requirements to FPGA electronic design; FPGA electronic design architecture, description for FPGA electronic design; VHDL and C code (when soft-processors are used); documents concerning FPGA chips with implemented electronic design; documents concerning integrated I&C system.

It is possible to apply two different environments to develop the checklists: spreadsheet or database. To transform design regulatory profile into convenient form for coverage assessment, it is necessary to identify attributes and metrics that help to assess the degree of met requirements for FPGA design.

5.2 FPGA design analysis and identification of a set of invariants

As a result of such analysis, the project can be divided into separate parts (components); it is the initial information for model formulation: non-functional requirements; supervised algorithms inside FPGA design (VHDL code); PID-regulators inside FPGA design; soft-processor emulator inside FPGA design (IP core).

Every part requires analysis to identify a set of invariants. Invariants are invariable identifiers and relations that implemented independently of input data. Invariants can be divided into two groups that are sensitive to different components of the design: universal invariants, i.e. independent from the design; invariants that are specific for the considered design (Andrashov et. al., 2010b).

Variety of invariant types includes the following: invariants that are based on features and constraints of HDL; invariants for timing charts; invariants for diagrams of control algorithms and automaton models; invariants for soft-processor (Nios), semantic invariants, etc.

Analysis of invariants for FPGA design is a subject of independent research. For example, invariants that are based on timing charts can be described by a tuple of operations number for memory elements per automaton cycles of the design.

This tuple is an invariant of FPGA design, since if it is correctly implemented and the system operates properly, then the value of vector elements or their compression to scalar should be unchanged.

Moreover, if there is a branching in the algorithm (timing charts) then the vector can be generalized to automaton graph with the nodes corresponding to line sections and weighed by particular vectors.

Invariants of control algorithm diagrams are based on interpretation of specification's functional elements and tests of tolerance. Automaton invariants are based on regularities of algorithm diagrams, transition tables and of H-algebra (elements: a set of imbedded algorithms or and operations of extension, compression).

5.3 FPGA design assessment using invariants

During invariant-oriented verification it is necessary to identify profile of faults to measure test coverage and validity factors with each invariant to identify potentially

unsafe parts of the code. For this purpose it is possible to use special test-beds.

For each design «defects-invariants» matrix is developed and task of coverage is solved. After this, FPGA design is tested on invariant groups using the tools that implement procedures of static analysis and invariant-based assessing.

Thus, the final verification report is formed based on the results of testing and invariants implementation.

6. CONCLUSIONS AND FUTURE WORK

This paper provides an overview of international standards and guidance related to FPGA-based NPP I&C systems.

It describes development and V&V processes used by RPC Radiy for FPGA electronic design of NPP I&C systems with respect to some safety related features such as coding rules and design constrains.

Principles and a set of standard methods and techniques of test-based verification for FPGA electronic design algorithms used in reactor trip systems are discussed. Testing policies, test coverage criteria, and methods, which allow assuring the highest possible test coverage, are outlined.

Invariant-oriented approach is the base technique for independent V&V implementation and allows reducing costs on verification, assuring required coverage and validity of assessment. It is an example of cost-effective technique of verification for safety-critical systems, in particular, FPGA-based nuclear reactor trip systems.

In future, to increase verification quality, we plan to use various tools for multi-version testing of FPGA electronic design algorithms and FPGA-based I&C systems as a whole.

Appropriate tools reducing time to generate test cases, as well as templates for functional units of the algorithms are under development.

REFERENCES

Kharchenko, V., et al., FPGA-based NPP I&C Systems: Development and Safety Assessment, 2008, RPC Radiy, National Aerospace University KhAI, SSTC on Nuclear and Radiation Safety, 188 p.

Bakhmach, I., et al., Advanced I&C Systems for NPPS Based on FPGA Technology: European Experience, 2009, ICONE-17.

Altium. Technical Reference TR0103 (v2.0), 2008.

Actel. Design Constraints User's Guide (v.9), 2010.

Yastrebenetsky, M., et al., NPP I&Cs: Problems of Safety, 2004, Technika, Kyiv. (translated in USA by NRC, 2007).

Munden, R., ASIC and FPGA Verification: A Guide to Component Modeling, 2005. Morgan Kaufmann, 336 p.

Andrashov, A., et. al., Safety Case-Oriented Assessment of Critical Software: Several Principals and Elements of Techniques, 2010a, Wroclaw, OWPW, pp 11-25.

Andrashov, A., et. al., Verification of FPGA Electronic Designs for Nuclear Reactor Trip Systems: Test- and Invariant-Based Methods, 2010b, Proc. of IEEE East-West Design & Test Symposium, St. Petersburg, pp 92-97.