Proceedings of the 17th International Conference on Nuclear Engineering
ICONE17
July 12-16, 2009, Brussels, Belgium

ICONE17-75270

LICENSING PRINCIPLES OF FPGA-BASED NPP I&C SYSTEMS

Alexander Siora
Research and Production Corporation “Radiy”
Kirovograd, Ukraine

Volodymyr Sklyar
State Scientific Technical Center on Nuclear and Radiation Safety
Kharkov, Ukraine

Yuriy Rozen
State Scientific Technical Center
on Nuclear and Radiation Safety
Kharkov, Ukraine

Svetlana Vinogradskaya
State Scientific Technical Center
on Nuclear and Radiation Safety
Kharkov, Ukraine

Mikhail Yastrebenetsky
State Scientific Technical Center
on Nuclear and Radiation Safety
Kharkov, Ukraine

ABSTRACT

This paper objective is to discuss licensing principles of FPGA-based NPP I&C systems. Licensing aspects of FPGA-based NPP I&C systems lay in peculiarities of FPGA-technology. The main idea for licensing performing of FPGA-based NPP I&C systems lays in consideration of FPGA-chip as hardware and FPGA-project as software. FPGA related licensing principles are discussed in details in the paper.

INTRODUCTION

Use of Field Programmable Gates Arrays (FPGA) for safety control functions performing in NPP I&C systems is a relatively novel technology which requires careful learning. An appearance of a draft of a new standard IEC 62566 “Nuclear Power Plants– Instrumentation and control important to safety– Selection and use of complex electronic components for systems performing category A functions” confirms the above thesis.

This paper objective is to discuss licensing principles of FPGA-based NPP I&C systems.

Licensing aspects of FPGA-based NPP I&C systems lay in peculiarities of FPGA-technology [1,2]. The accumulated experience permits to formulate such licensing principles for FPGA-based NPP I&C systems [3,4]:

– the main idea for licensing performing of FPGA-based NPP I&C systems lays in consideration of FPGA-chip as hardware and FPGA-project as software;

– requirements of software engineering and NPP I&C standards should be used for FPGA-based NPP I&C systems;

– development of FPGA projects is a part of NPP I&C systems life cycle corresponding to software development;

– FPGA projects are treated as a kind of software and may include:

1) graphic diagrams of digital devices in problem-oriented languages;
2) a code in hardware description language;
3) an assembler code or a code in high level language to be executed in the environment of microprocessor emulators implemented to FPGAs;

– FPGA development stages duly consider the peculiarities of accepted technologies and tools;

– each of FPGA development stages ends in verification of obtained product;

– verification of FPGA projects duly considers the peculiarities of accepted technologies and tools;

– qualification is performed for hardware, including FPGAs.

The above licensing principles are discussed in details in the paper.

STANDARDS ANALYSIS

Standards of the following organizations should be taken into account for NPP I&C systems analysis [5,6]:

– standards JTC1/SC7 “Software Engineering” of ISO/IEC (Subcommittee 7 of Joint Technical Committee for Information Technology of International Organization for Standardization and International Electrotechnical Commission);

– standards on Software Engineering of IEEE (Institute of Electrical and Electronics Engineers);

– standards on I&C systems functional safety of IEC Subcommittee 65A "System aspects";

– standards on NPP I&C systems of IEC Subcommittee 45A "Instrumentation and control of nuclear facilities";

– standards on NPP I&C systems safety of IAEA (International Atomic Energy Agency).

Logical connections between standards developers are presented in Figure 1.

IEC and IAEA standards related NPP I&C systems and its components are presented in Figure 2. This standard structure contains a Draft of the new standard of IEC 62566. The IEC 62566 will establish requirements to I&C hardware components which includes programmable electronic components (FPGA, boards etc.).

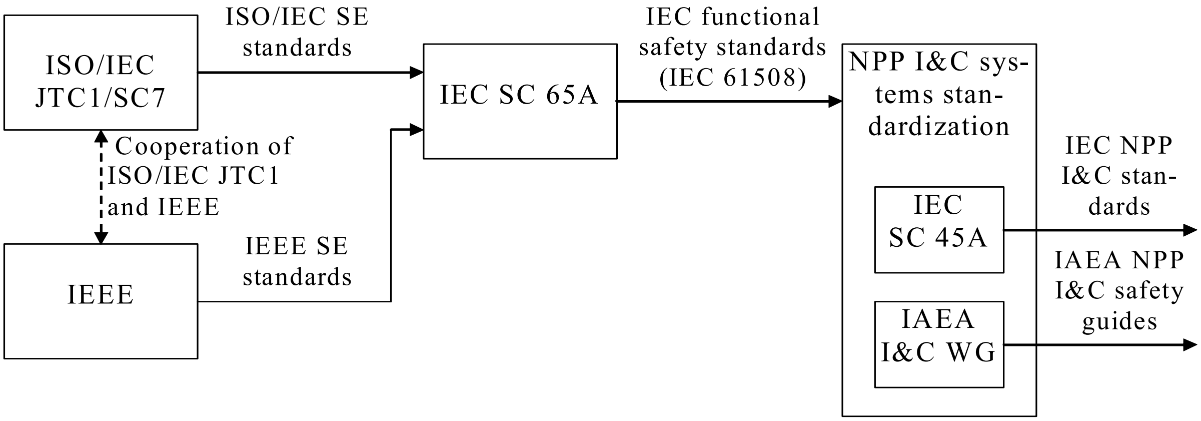
ISO/IEC JTC1/SC7
ISO/IEC SE standards
IEC SC 65A
IEC functional safety standards (IEC 61508)
NPP I&C systems standardization
Cooperation of ISO/IEC JTC1 and IEEE
IEEE
IEEE SE standards
IEC SC 45A
IAEA I&C WG
IEC NPP I&C standards
IAEA NPP I&C safety guides

Figure 1. Standards developer in areas of Software Engineering (SE) and NPP I&C systems

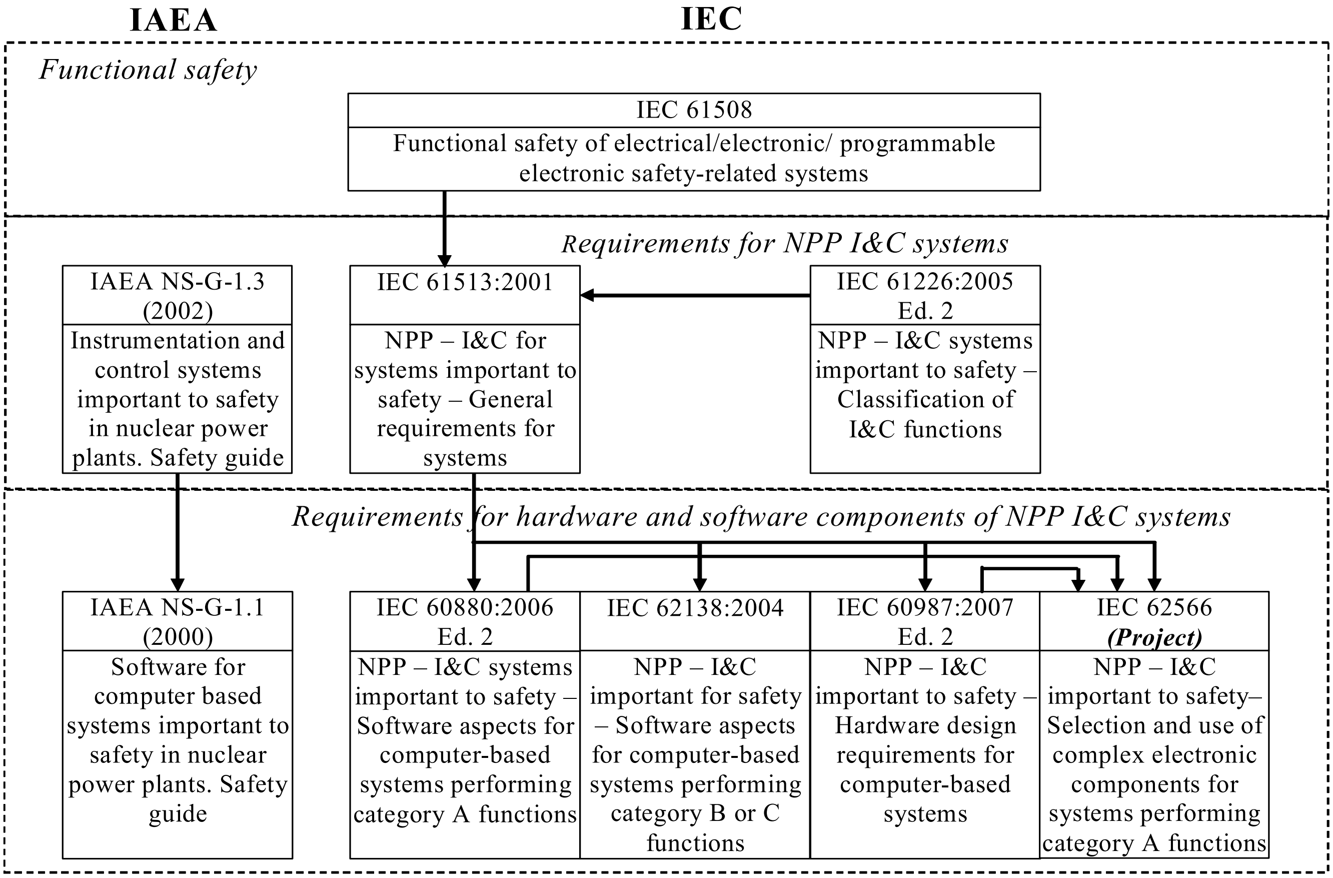
IAEA
IEC
Functional safety
IEC 61508
Functional safety of electrical/electronic/ programmable electronic safety-related systems
Requirements for NPP I&C systems
IAEA NS-G-1.3 (2002)
Instrumentation and control systems important to safety in nuclear power plants. Safety guide
IEC 61513:2001
NPP – I&C for systems important to safety – General requirements for systems
IEC 61226:2005 Ed. 2
NPP – I&C systems important to safety – Classification of I&C functions
Requirements for hardware and software components of NPP I&C systems
IAEA NS-G-1.1 (2000)
Software for computer based systems important to safety in nuclear power plants. Safety guide
IEC 60880:2006 Ed. 2
NPP – I&C systems important to safety – Software aspects for computer-based systems performing category A functions
IEC 62138:2004
NPP – I&C important for safety – Software aspects for computer-based systems performing category B or C functions
IEC 60987:2007 Ed. 2
NPP – I&C important to safety – Hardware design requirements for computer-based systems
IEC 62566 (Project)
NPP – I&C important to safety– Selection and use of complex electronic components for systems performing category A functions

Figure 2. IEC and IAEA standards structure for NPP I&C systems

NPP I&C system V-shape life cycle, as it is defined in IAEA NS-G-1.1 Safety Guide, is presented in Figure 3. Every stage of software development is terminated by verification of the obtained product. An integrated system should be validated against initial requirements. If FPGAs is used as programmable components, software development stages should be replaced by FPGA-project development stages.

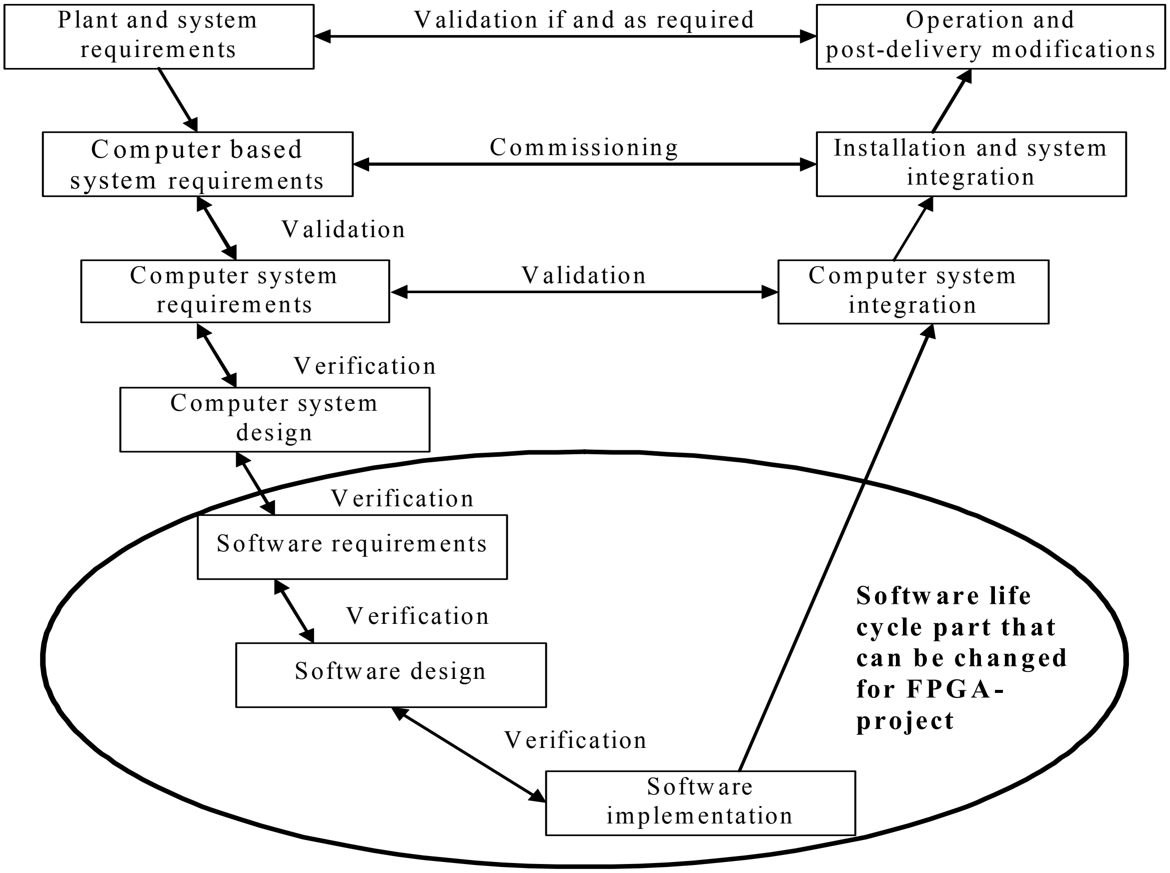
Plant and system requirements
Validation if and as required
Operation and post-delivery modifications
Computer based system requirements
Commissioning
Installation and system integration
Validation
Computer system requirements
Validation
Computer system integration
Verification
Computer system design
Verification
Software requirements
Verification
Software design
Verification
Software implementation
Software life cycle part that can be changed for FPGA-project

Figure 3. V-shape life cycle of NPP I&C systems

LIFE CYCLE OF FPGA-PROJECTS

Our analysis of software life cycle showed that for FPGA it should study the section of life cycle beginning from specification of a computer I&C system and up to integration of a computer I&C system. Such actions may be performed in parallel to software development.

Besides, verification after each stage of development is obligatory for both FPGA-project and software. Software verification is a process aimed to confirm software compliance to defined requirements by way of versatile tests and obtaining verifiable proofs [7,8].

Development of FPGA-based digital device consists of such stages (see Figure 4):

– development of signal formation algorithm block-diagrams;

– development of signal formation algorithm program models in design environment which is determined depending on type and/or manufacturer of FPGA realization environment applied;

– integration of signal formation algorithm program models (development of digital device integrated program model) into design environment;

– implementation (loading) of integrated digital device program model to FPGA.

The key term here is «signal formation algorithm block-diagram» implying a certain functionally finite project module presented in the form of a graphic diagram or a listing in hardware description language (HDL). The result of each step is a new product, the final result being a FPGA with implemented logic structure. At each step the developed product must be verified. The procedures of FPGA-based I&C system development and verification are shown in Figure 4. The following is a description of FPGA-based I&C system LC stages.

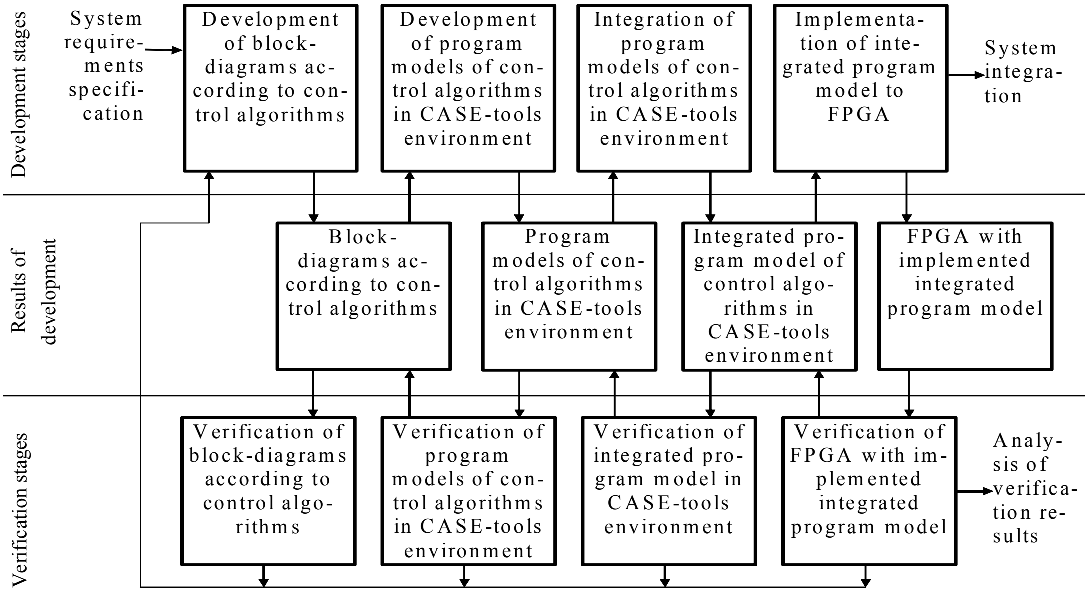
Development stages
Results of development
Verification stages
System requirements specification
Development of block-diagrams according to control algorithms
Development of program models of control algorithms in CASE-tools environment
Integration of program models of control algorithms in CASE-tools environment
Implementation of integrated program model to FPGA
System integration
Block-diagrams according to control algorithms
Program models of control algorithms in CASE-tools environment
Integrated program model of control algorithms in CASE-tools environment
FPGA with implemented integrated program model
Verification of block-diagrams according to control algorithms
Verification of program models of control algorithms in CASE-tools environment
Verification of integrated program model in CASE-tools environment
Verification of FPGA with implemented integrated program model
Analysis of verification results

Figure 4. A part of I&C system life cycle concerning FPGA-project

Development of signal formation algorithm block-diagrams. Signal formation algorithm block-diagrams are developed as a direct preparation to development of a digital device block-diagram in CASE-tools (design) environment. Initial data are:

– System Requirement Specification (SRS) (functional general and non-functional general safety requirements) with due consideration of function distribution between software and hardware;

– process engineering requirements that may be formulated by customer as an addition to SRS requirements.

Initial information in requirements may be both in verbal form and in the form of formalized (problem oriented) languages describing function algorithms.

Signal formation algorithm block-diagrams are developed in the form as close as possible to scheme presentation in FPGA design environment and, as a consequence, should take into consideration the peculiarities of tools applied. For complicated digital devices one of critical issues is structure division into functional modules and formation of series and/or parallel tiers of such modules.

In case the requirements to functioning algorithms are presented in verbal form, at that stage such works may be consecutively performed:

– development of description of functioning algorithms in a formalized language;

– development of signal formation algorithm block-diagrams as adapted to current tools.

Development of signal formation algorithm block-diagram program models in CASE-tools environment. The initial data at that stage are signal formation algorithm block-diagrams. Development of signal formation algorithm block-diagram program models in design environment is performed using specialized CASE-tools comprising a library of typical functional elements and blocks.

In the course of development signal formation algorithms and FPGA logic structure are presented in the form of visualized conditional graphic images (block-diagrams). It should be noted that the development of signal formation algorithm block-diagram program models and FPGA program model is similar to the process of software product development in a problem oriented program language using specialized tools.

An alternative to direct diagram drawing may be FPGA structure description in a HDL, such as Verilog or VHDL.

HDL is a formalized record that may be used at all development stages. System function is defined as transformation of input values into output values, operation time in this transform being prescribed in explicit form. General FPGA structure is prescribed by a list of connected components – functional blocks that realize signal formation algorithm block-diagram program models.

At this stage some library modules lacking in standard tools library must be created. Such library modules in FPGA structure are called IP-Cores (Intellectual Property Cores) or IP-functions. Such modules are universal and reliably repetitive, from the one side, and capable of parametric adjustment to a particular project, from the other side. Repeated application of IP-Cores permits to reduce labor costs and design period of digital devices, ensuring their high reliability.

Integration of signal formation algorithm block-diagram program models in CASE-tools environment. At that stage signal formation algorithm block-diagram program models as developed at a previous stage in CASE-tools environment are integrated. An important issue in this is establishment of connections and sequences between developed functional blocks (signal formation algorithm block-diagram program models), including input and output signal formers, in strict conformity with the developed signal formation algorithm block-diagrams.

As well as at the previous stage, integration may be performed both in the form of graphic diagrams and by programming in hardware describing language.

The result of this stage is a finite digital device program model ready to be loaded to FPGA.

Implementation of integrated program model to FPGA. The developed digital device logic structure program model is implemented by adjustment of connections between FPGA logic cells using the appropriate interface equipment (JTAG interface) connected to an instrumental PC. Interface equipment for adjustment of connections between crystal logic cells is selected in accordance with the type and/or manufacturer of components applied.

Thus, this step is a transfer from software realization of digital device to its final hardware realization. The product of this implementation is an FPGA-based digital device that performs certain functions within I&C system.

VERIFICATION OF FPGA-PROJECTS

The conformity between verification stages of FPGA-projects, tasks performed at those stages and methods of task performance is explained in Table 1.

Let us give a short characteristic of FPGA-projects verification methods.

Documentation technical review method applied to assess completeness and correctness of algorithm block-diagrams. Signal formation algorithm block-diagrams are results (products) of a corresponding FPGA-project development stage. The completeness of signal formation algorithm block-diagrams is assessed by comparison between the lists of developed algorithm block-diagrams to signal formation conditions according to SRS agreed with customer. Conformity criterion is coincidence between the list of developed algorithm block-diagrams and signal formation conditions according to SRS.

Correctness of signal formation algorithm block-diagrams is assessed by correctness, unambiguous treatment and preparation quality of the developed block-diagrams. In the course of analysis the following must be confirmed:

– a separate algorithm block-diagram is presented for each signal formation condition;

– each block-diagrams comprises a strict signal formation condition formulation in accordance with SRS;

Table 1. The conformity between verification stages of FPGA-projects, tasks performed and methods of task performance

Verification stage	Verification task	Verification method
Development of signal formation algorithm block-diagrams	Completeness and correctness assessment of signal formation algorithm block-diagrams	Documentation technical review
	Conformity assessment to SRS	Traceability analysis
	Structuredness assessment of algorithm block-diagrams	Complexity assessment
Development of signal formation algorithm block-diagram program models in CASE-tools environment	Testing of algorithm block-diagram program models	Functional and timing simulation in CASE-tools environment
	Completeness of tests assessment	Walk-through of documentation
	Conformity assessment to algorithm block-diagrams	Traceability analysis
Integration of signal formation algorithm block-diagram program models in CASE-tools environment	Testing of digital device program model	Functional and timing simulation in CASE-tools environment
	Completeness of tests assessment	Walk-through of documentation
	Conformity assessment to program models	Traceability analysis
Implementation of digital device program model to FPGA	Testing of FPGA with implemented program model	Blackbox functional testing
	Completeness of tests assessment	Walk-through of documentation
	Conformity assessment to integrated program model	Traceability analysis

– each block-diagram is performed under established form as a connection of typical structural elements selected from a prescribed standard set;

– inputs and outputs of each structural element are clearly and unambiguously identified in accordance with established rules;

– identifiers and names are specified for all input signals, alteration limits being also specified for continuous signals;

– identifiers, names and destinations are specified for all output signals in the scheme;

– set-points determining signal formation conditions and return to normal operation are specified in the scheme with necessary accuracy;

– necessary timing characteristics (backoffs under alterations of input signals, signal formation delays, signal issuance time before automatic de-energization, etc.) are specified in the scheme with necessary accuracy.

Conformity criterion is meeting all the above requirements to algorithm block-diagrams preparation.

Method of functional and timing simulation in CASE-tools environment. Functional and timing simulation in CASE-tools environment implies testing of each of signal formation algorithm block-diagram program models in design environment as well as testing of the integrated program model.

Under functional simulation conditions ("input signals") corresponding to normal operation and to each of signal formation conditions are consecutively imitated at test inputs. Altered states of program model outputs and/or of established control points within the program model as caused by such effects are observed at instrumental screen and registered as "hard copies" from the screen. Under timing simulation consecutive state alterations at one of program model inputs are imitated in turns and altered states of test outputs ("timing diagrams") and/or of established control points within the program model are monitored.

Tests performed should imply:

– testing of new functional blocks (IP-Cores), arranged from typical functional elements in CASE-tools environment;

– testing of algorithm block-diagram program models arranged from typical functional elements and new functional blocks in design environment;

– testing of integrated algorithm block-diagram program model arranged from signal formation algorithm block-diagram program models in CASE-tools environment.

New functional blocks are tested directly in CASE-tools environment. As the functional blocks are invariant relative to input data, after verification they may be included into the library of CASE-tools applied and find multiple usages during development of FPGA-projects. Tests for algorithm program models are developed on the basis of signal formation algorithm block-diagrams verified at the previous stage.

Tests for integrated program model are developed on the basis of signal formation algorithm program models verified at the previous stage.

The developed tests and testing results must be presented in FPGA-project verification documents in the form of tables and timing diagrams. In timing diagrams the imitated input states (input signals) of program models in CASE-tools environment, their alterations and altered states of each of outputs should be specified.

Criterion of success is a conclusion that test results correspond to expected results.

Walk-through method of documentation viewing used to assess testing completeness. Walk-through of documentation is a kind of inspection of documents correctness, completeness and consistency. We shall mark the peculiarities of one of the key stages – analysis of testing complete coverage of algorithm program models in the process of FPGA-project verification. Such analysis is performed by comparison between the list of qualitatively different combinations of input states and/or of their alterations that cause altered output states and the list of program model input-output states that are imitated and monitored in the course of testing.

Criteria of conformity in this are:

– presence and completeness of tests for all new functional blocks composed from typical functional elements;

– presence and completeness of tests for all FPGA-project algorithm program models;

– presence and completeness of tests for integrated program model;

– presence and completeness of tests for final FPGA with implemented program model.

Blackbox functional testing method. Functional testing, named also blackbox testing, consists in experimental checking of functions performed by a programmable component with implemented program model to define their conformity to system requirements, signal formation algorithm schemes and user documentation [9].

Traceability analysis. This is done to ensure that input requirements of a certain process are exhaustively considered by analysis of their connections to output results as well as all requirements have been defined and brought through the life cycle of development, i.e. from requirement analysis up to final testing.

Traceability analysis includes identification of input requirements and confirmation of the fact that they have been considered by way of inspection of destination documents. For instance, the analysis may inspect translation of system requirements documentation into FPGA-project requirements documentation, or that of FPGA-project requirements documentation into digital device characteristic specification, or that of system requirements documentation into tests, etc. If necessary, traceability analysis may include a requirements confirmation step to ensure that actual requirements, but not simply sections of input documentation, have been traced. The results of analysis must show whether all requirements have been duly considered. For this analysis usually traceability matrices are used comprising comparison between input requirements and the elements of output results.

In the course of FPGA-projects verification traceability analysis is applied to ensure tracing or establishment of connections:

– between SRS and signal formation algorithm block-diagrams;

– between signal formation algorithm block-diagrams and their program models in CASE-tools environment;

– between signal formation algorithm block-diagrams and integrated program model in CASE-tools environment;

– between FPGA-projects integrated program model in CASE-tools environment and FPGA with implemented logic structure.

Conformity of signal formation algorithm block-diagrams to the initial data of SRS is assessed for each block-diagram separately by comparing:

– logic conditions of signal formation and return to normal operation, the latter being defined by this scheme, to conditions established in specification;

– numerical values of set points and timing characteristics that define conditions of signal formation and return to normal operation to their values established in specification;

– identifiers and names of input and output signals and alteration limits of continuous input signals as specified in the scheme to specification data.

Conformity criterion for this verification stage is coincidence of logic conditions, numerical values of set points and timing characteristics, identifiers, names and alteration limits of signals as defined from signal formation algorithm schemes to initial data established in specification.

Conformity of algorithm program models developed in CASE-tools environment to signal formation algorithm block-diagrams is assessed by way of comparison:

– of identifiers, names and alteration limits of input signals;

– of identifiers, names and formation logic conditions of output signals;

– of connection topologies between structural elements, numerical values of set points and timing characteristics specified in algorithm block-diagrams and “hard copies” from screen that diagrammatically reflect the developed algorithm program models.

Conformity criteria for this stage are:

– usage in algorithm program models of only those elements that are included into typical functional elements library of CASE-tools environment applied;

– presence and completeness of tests for all new functional blocks composed from typical functional elements;

– presence and completeness of tests for all algorithm program models;

– positive testing results of all new functional blocks and algorithm program models in CASE-tools environment applied;

– equivalence of developed algorithm program models and protective signal formation algorithm block-diagrams as verified at previous stage.

– absence of any input, output signals and/or set points in FPGA-project program model for which inputs and/or outputs exist in none of algorithm program models.

Conformity of FPGA with implemented program model to this program model in CASE-tools environment is assessed by comparison of signal formation conditions and timing characteristics as obtained by testing to logic conditions, numerical values of set points and timing characteristics specified in algorithm block-diagrams and “hard copies” from screen that diagrammatically reflect the developed FPGA logic structure program model.

Conformity criteria for this verification stage are:

– successful implementation of FPGA-project that was verified at previous stage into FPGA-chip;

– equivalence of output signal formation conditions and timing characteristics as obtained by testing to logic conditions, numerical values of set points and timing characteristics of FPGA-project that was verified at previous stage.

Thus, FPGA-project traceability analysis method for each verification stage includes such actions:

– analysis of verification stage input data presentation and separation of component classes (signals, communication lines, nodes, functional blocks, etc.);

– detailed analysis of components in each class;

– filling of traceability matrix with input data by systematization of components in each class;

– analysis of verification stage output data presentation and separation of component classes;

– conformity analysis between input and output data and filling of traceability matrix with output data by comparison of each of output data component and input data components;

– analysis of final traceability matrix, formulations of conclusions and recommendations;

– overpatching and correction of final product in case any discrepancies are found between input data and output result of development stage.

Let us prepare a formal description of FPGA-project traceability analysis.

FPGA-project is developed and verified in 4 stages, with traceability analysis performed for each stage. Assume FPGA realizing N signal processing algorithms. For each algorithm S classes of omponents exist that belong to FPGA algorithm as well as L input components A_{ij} of FPGA algorithm and M output components B_{ij} of FPGA algorithm that belong to i-th class. Our analysis shows that component classes of FPGA algorithms are equivalent for each development class. Between all input and output algorithm component we must ascertain whether equivalence conformity is met or not. In this context traceability analysis would be successful if each of input elements corresponds to one or more output elements and each of output elements corresponds to one or more input elements: $(\forall A_{ij} : \exists \{B_{ij}\}, A_{ij} \Leftrightarrow \{B_{ij}\}) \vee (\forall B_{ij} : \exists \{A_{ij}\}, B_{ij} \Leftrightarrow \{A_{ij}\})$.

In case the above condition is not met, input and output data of a development stage are not intertraceable and stage results mist be corrected.

Traceability matrix includes four columns:

– column of FPGA algorithm input components – its elements are input components A_{ij}, i = 1,...,S, j = 1,...,L;

– column of FPGA algorithm output components – its elements are output components B_{ij}, i = 1,...,S, j = 1,...,M;

– column of traceability results – elements are conclusions of traceability between FPGA algorithm input and output components; conclusion data take binary values "meeting" ($A_{ij} \Leftrightarrow B_{ij}$ met) or "not meeting" ($A_{ij} \Leftrightarrow B_{ij}$ not met);

– column of comments – additional data on FPGA algorithm components development and verification.

Complexity assessment. One of basic FPGA-projects characteristics affecting their reliability is complexity. FPGA-project complexity metrics may be applied to access the critical scope of signal formation algorithms and integrated program model, above which the probability of bringing errors drastically increases. Complexity assessment includes [10]:

– analysis of problem oriented language in which the algorithms have been developed, separation of operator and operand classes;

– prescription of weights from the point of view of complexity for operator and operand classes;

– establishment of limit value for integral complexity metric above which the probability of bringing errors into FPGA-projects drastically increases;

– direct complexity measurement including count of the number of operators and operands for each class and determination of integral complexity metric value;

– analysis of obtained complexity metric values, formulation of conclusions and recommendations;

– breaking into modules for those algorithms where complexity metric exceeds its limit value.

QUALIFICATION OF FPGA-BASED HARDWARE

Qualification tests of FPGA-based hardware in accordance with IEC standards requirements include [3,4]:

– climatic and radiation qualification (see Table 2);

– mechanical and seismic qualification (see Table 3);

– electromagnetic compatibility qualification (see Table 4).

Results of qualification tests confirmed FPGA-based hardware compliance with IEC safety requirements.

CONCLUSIONS

The main idea for licensing performing of FPGA-based NPP I&C systems lays in consideration of FPGA-chip as hardware and FPGA-project as software

Requirements of software engineering and NPP I&C standards were used for FPGA-based NPP I&C systems.

FPGA-based I&C systems life cycle has V-shape type. Every stage of software development is terminated by verification of the obtained product. Software development stages are replaced by the following specific FPGA-project development stages:

– development of signal formation algorithm block-diagrams;

– development of signal formation algorithm program models in design environment which is determined depending on type and/or manufacturer of FPGA realization environment applied;

– integration of signal formation algorithm program models (development of digital device integrated program model) into design environment;

– implementation (loading) of integrated digital device program model to FPGA.

The following verification method are appropriate for FPGA-projects:

– documentation technical review;

– traceability analysis;

– walk-through of documentation;

– complexity assessment;

– functional and timing simulation in CASE-tools environment;

– blackbox functional testing.

The following type of qualification tests were used for FPGA-based hardware:

– climatic and radiation qualification;

– mechanical and seismic qualification;

– electromagnetic compatibility qualification.

FPGA-based I&C safety platform "Radiy" by Corporation Radiy (Ukraine) is licensed for NPP application in Ukraine and in Bulgaria.

The following I&C system on the base of platform "Radiy" are now in operation at Ukrainian and Bulgarian NPPs:

– Reactor Trip Systems;

– Reactor Power Control and Limitation System;

– Engineering Safety Feature Actuation System;

– Control Rods Actuation System.

The main benefits from FPGA application are the following:

– realization control and other safety-critical functions in the form of "hard" logic (FPGA with implemented project has own hardware for each algorithm), without software;

– reducing time necessary for software verification;

– performing parallel processing of all control algorithms within cycles, and providing determined temporal characteristics due to parallel operation of control algorithms etc.

Table 2. Working and testing values of climatic and radiation impacts for FPGA-based hardware

Name and unit of measurement	Working environment impact value	Testing environment impact value
Temperature, °C lower, not more upper, not less	15 30	10 50
Speed of temperature change, °C/hour not less	–	5
Relative humidity, % lower, not more upper, not less	10 95 (at 30°C)	5 (at 15°C during 2 hours) 100 (at 50°C during 2 hours)
Barometric pressure, kPa lower, not more upper, not less	84 108	84 108
Absorbed dose rate, Gr/hour, not less	10^{-4}	10^{-4}

Table 3. Testing values of seismic impacts for FPGA-based hardware

Impact direction	Acceleration amplitude, m/s^2, not less, at frequency of									
	0,5 Hz	1 Hz	2 Hz	3 Hz	4 Hz	5 Hz	6 Hz	10 Hz	15 Hz	30 Hz
X,Y	17,4	28,8	37,6	24,8	12,8	11,4	11,4	11,4	11,4	6,0
Z	8,8	15,6	21,0	19,4	17,6	17,6	16,0	10,0	5,0	3,6

Table 4. Hardness degree of electromagnetic compatibility testing for FPGA-based hardware

Interference type	Standard	Hardness degree
Electrostatic discharge	IEC 61000-4-2:2001. Electromagnetic compatibility (EMC) – Part 4: Testing and measurement techniques – Section 2: Electrostatic discharge immunity test	4
Radiated, radio-frequency, electromagnetic field	IEC 61000-4-3:2001. Electromagnetic compatibility (EMC) – Part 4: Testing and measurement techniques – Section 3: Radiated, radio-frequency, electromagnetic field immunity	3
Fast transient and burst (nanosecond noise spikes)	IEC 61000-4-4:2001. Electromagnetic compatibility (EMC) – Part 4: Testing and measurement techniques – Section 4: Electrical fast transient / burst immunity test	4
Surge (microsecond noise spikes)	IEC 61000-4-5:2001. Electromagnetic compatibility (EMC) – Part 4: Testing and measurement techniques – Section 5: Surge immunity test	4
Conducted disturbances, induced by radio-frequency fields	IEC 61000-4-6:2003. Electromagnetic compatibility (EMC) – Part 4: Testing and measurement techniques – Section 6: Immunity to conducted disturbances, induced by radio-frequency fields	3
Power frequency magnetic field	IEC 61000-4-8:2001. Electromagnetic compatibility (EMC) – Part 4: Testing and measurement techniques – Section 8: Power frequency magnetic field immunity test	5
Pulse magnetic field	IEC 61000-4-9:2001. Electromagnetic compatibility (EMC) – Part 4: Testing and measurement techniques – Section 9: immunity test	5
Damped oscillatory magnetic field	IEC 61000-4-10:2001. Electromagnetic compatibility (EMC) – Part 4: Testing and measurement techniques – Section 10: Damped oscillatory magnetic field immunity	4
Voltage dips, short interruptions and voltage variations	IEC 61000-4-11:2001. Electromagnetic compatibility (EMC) – Part 4: Testing and measurement techniques – Section 11: Voltage dips, short interruptions and voltage variations immunity tests	4

REFERENCES

1. J. Lach, W. Mangione-Smith, M. Potkonjak, Enhanced FPGA Reliability through Efficient Run-Time Fault Reconfiguration, IEEE Trans. on Reliability, 49, pp. 296-304 (2000).
2. G. Doerre, D. Lackey, The IBM ASIC/SoC Methodology – A Recipe for First-time Success, IBM Journal of Research and Development, 6, pp. 649-660 (2002).
3. M.A. Yastrebenetsky (edit), Safety of Nuclear Power Plants: Instrumentation and Control Systems, Technika, Kyiv & Ukraine (2004).
4. V. Kharchenko, V. Sklyar (edits), FPGA-based NPP Instrumentation and Control Systems: Development and Safety Assessment., RPC "Radiy", National Aerospace University "KhAI", State STC on Nuclear and Radiation Safety, Kharkiv & Kirovograd & Ukraine (2008).
5. N. Leveson, Safeware: System Safety and Computers, Addison-Wesley (1995).
6. Preckshot G., Method for Performing Diversity and Defense-in-Depth Analysis of Reactor Protection Systems. NUREG/CR-6303, LLNL, Livermore, USA (1994).
7. M.R. Lyu Handbook of Software Reliability Engineering. McGraw-Hill Company, 1996. 805 p.
8. V. Kharchenko, Multi-version Systems: Models, Reliability, Design Technologies, Proceeding of 10th ESREL Conference, Munich, Germany, Vol. 1, pp. 73-77 (1999).
9. L. Pullum, Software Fault Tolerance Techniques and Implementation, Artech House Computing Library (2001).
10. A. Avizienis, J.-C. Laprie, B. Randell, C. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing", IEEE Transactions on Dependable and Secure Computing, vol.1, pp.11-33 (2004).