The 11th IEEE International Conference on Dependable Systems, Services and Technologies, DESSERT'2020
14-18 May, 2020, Kyiv, Ukraine

Software Fault Insertion Testing for SIL Certification of Safety PLC-Based System

Oleg Odarushchenko
Radics LLC
Kropyvnytskiy, Ukraine
o.odarushchenko@radics.tech

Oleksiy Striuk
Radics LLC
Kropyvnytskiy, Ukraine
o.striuk@radics.tech

Kostiantyn Leontiiev
RPC Radiy,
Kropyvnytskiy, Ukraine,
ksleontiev@radiy.com

Elena Odarushchenko
Information systems and technologies department
Poltava State Agrarian Academy
Poltava, Ukraine
elena.odarushchenko@gmail.com

Abstract — **The problems of the safety PLC market are analyzed. A tendency of the PLCs transition for industrial applications (for example NPP I&C) in new technologies has been identified. The advantages of FPGA technology for developing safety-based PLCs in comparison with microprocessor technology are formulated. The fault insertion testing (FIT) is considered as one of the mandatory techniques applied in the process of the certification against requirements of IEC 61508 according to safety integrity level (SIL). The concept of the HW FIT-ability is generalized for SW components of FPGA PLC-based safety-critical I&C systems (FPICS) (SW FIT-ability). Procedures, techniques and tools taking into account existing set of insertion variants for SW faults are proposed.**

Keywords— I&C systems; PLC - programmable logic controller; fault insertion testing (FIT); software FIT; FIT procedures; FIT tools

I. INTRODUCTION

At the current time, programmable logic controllers (PLCs) are widely used as the core of automated control systems for technological processes and productions. The main operating mode of the PLC is its long-term autonomous use, often in adverse environmental conditions, without serious maintenance and practically without human intervention. The main areas of PLCs application are automation of technological processes of industrial production - instrumentation and control systems (I&C); in emergency protection and alarm systems (including nuclear power plants (NPP) - PLC – Based safety-critical I&C systems (PICS).

The relay and analog electronics technology used in I&C is becoming more and more obsolete. Users of these systems replace these older systems with more modern, usually microprocessor-based equipment.

However, experience has shown that for safety applications, obtaining regulatory approval for microprocessor systems can be difficult and costly.

Microprocessor-based systems are complex and include a large number of software in the form of operating systems and other software, as well as custom software that executes user application.

In addition, microprocessors tend to become obsolete much faster than the analog equipment they replaced. Thus, users are faced with the need for repeated replacements. As a result, there is growing interest in studying other electronic technologies for use in I&C.

Field-programmable gate arrays (FPGAs) are gaining increased attention worldwide for application in NPP and I&C systems (FPGA PLC-based safety critical I&C systems (FPICS)), particularly for safety applications. Use of FPGAs has potential to reduce complexity and the associated burden of gaining regulatory approval and also provide better protection against obsolescence as compared to conventional microprocessor-based systems, which have been the technology of choice over the last two decades [1,2].

Also of interest is a combination of the advantages microprocessor and FPGA technology, partly shown in this article.

Application of the FPGAs may cause specific risks which must be decreased to an acceptable level. NPP FICs should meet requirements of the standards related to verification and validation (V&V).

Existed techniques of V&V for FPICs are based on documentation inspection, technical review, static and code analysis and functional testing, integration and validation testing and others [10]. The fault insertion testing (FIT) is one of the modern and mandatory techniques applied in process of certification to requirements of IEC 61508 [3,5,6] according with safety integrity level (SIL). The faults are inserted into the modules and configuration files of FPICS to solve the following tasks:

- to demonstrate that the implementation of PLC integrity functions is correct and complete;

978-1-7281-9957-3/20/$31.00 ©2020 IEEE

- to investigate failures where the Failure Modes, Effects and Diagnostic Analysis (FMEDA) analyst could not be sure what the impact of a specific failure might be [4];

- to increase test coverage and test quality;

- to increase diagnostic coverage for software (SW) and hardware (HW) components and their configuration;

- to test fault-tolerance procedures and means;

- to proof actual level of FPICS safety.

At present, there are not detailed regulation requirements and SIL-oriented FIT techniques for FPICs. The goal of the paper is generalization of the concept hardware FIT-ability [7÷9] for SW components FPICS (SW FIT-ability) and to propose procedures, techniques and tools taking into account existing set of insertion variants for SW faults.

II. SW FIT-ABILITY AND SW FIT PROCEDURE

A. SW FIT-ability

The concept *FIT-ability* was formulated as the ability of FPGA PLC-based safety critical I&C systems (FPICS) to fulfill of HW fault insertion testing [6]. Further we will talk about SW FIT –ability, i.e. ability to insert faults regarding to actual SW components. Hence, for SIL-oriented process of FPICS certification concept of SW FIT-ability may be specified as ability to fulfil of fault insertion (seeding) testing according to Failure Modes, Effects and Diagnostic Analysis (FMEDA) results.

The input data for performing SW FIT is the results of FMEDA (T_{FMEDA}), were T_{FMEDA} presents the list of type of failure which should be realized for SW Components. For SW FPICS:

$$FMEDA = \cup(SW\ Components)FMEDA \qquad (1)$$

The next task is determination of a position (a point in the SW Component) pf_i and means mf_i of insertion for all faults f_i. In general pf_i and mf_i can be specified as sets $pf_i = \{pf_{ij}\}$, $mf_i = \{mf_{ik}\}$, $j = 1,\ldots, npf_i$; $k = 1,\ldots, nmf_i$, and

$$\exists i,v,\ i \neq v:\ pf_i \cap pf_v \neq \varnothing,\ mf_i \cap mf_v \neq \varnothing.$$

The full FIT covered space of f_i (FF_CSf_i) will call a set received as a Cartesian product:

$$FF\_CSf_i = pf_i \times mf_i. \qquad (2)$$

The full FIT covered space of SW Component (FF_CSSWComp) is an aggregation of all FF_CSf_i:

$$FF\_CSSWComp = \cup\ FF\_CSf_i,\ i = 1,\ldots,F \qquad (3)$$

A dedicated FF_CSSWComp table corresponds full FIT covered space of SW Component. Trivial FIT covered space of f_i ($TFCS_i$) will call a set for which:

$$\forall pf_{ij}\ \ \exists! mf_{ij},\ Card\ TFCS_i = npf_i;$$
$$TFCS = \cup\ TFCSf_i,\ TFCS_i \subseteq FFCSf_i,$$
$$TFCS \subseteq FFCSf.$$

The implemented FIT covered space of SW Component (IFCSSWComp) is a subset of *FFCS*, $IFCSSWComp \subseteq FFCS$ (and $TFCS \subseteq IFCS$) and should be developed taking into account a restriction of *FITR* for fault insertion on different SW Components points. The set IFCSSWComp is designed with the main restriction in mind - fault insertion must be technologically feasible.

Hence,

$$IFCSSWCom = FITR \blacklozenge FFCS. \qquad (4)$$

where ♦ -operation of the FFCS set filtration by the FITR. IFCSSWComp is described by sets of *points* $Ipf_i = \{Ipf_{ij}\}$ and means $Imf_i = \{Imf_{ik}\}$, $j = 1,\ldots, Inpf_i$; $k = 1,\ldots, nmf_i$. It's understandable that $\forall Ipf_{ij} \subseteq pf_{ij}$, $\forall Imf_{ik} \subseteq mf_{ik}$. The main tasks of SW FIT are the following:

- checking the diagnostic subsystem ability of the PLC to detect errors in the configuration data and checking its ability to perform protective actions in accordance with the type of failure detected (f_{ij});
- checking the IDE on-line components ability to detect errors in the configuration data and check their ability to perform protective actions in accordance with the type of failure detected.

B. SW FIT Procedure

The SW FIT procedure consists of the following stages:

Stage_1. Analysis of the software detailed design description.. Defining a list of potential threats.

TABLE I. List of the Threats

List of threats	
Threat type	Source of threat
Applicable Data Types	The User Application Logic block provides input expected as FP32, but which are actually SI32, or vice versa. SI32 can be interpreted as FP32.
Loopbacks	The User Application Logic block can receive data from the block that must be executed after it (loop)
Elements and components of the IDE library	Inconsistency of the VHDL component description with the function description in the IDE. An error can lead to the execution of a non-specified for PLC function.
Inconsistent XML definitions (descriptions) for library components	PLC can use (mix) data between data processing channels

Stage_2. Development of FMEDA – tables.

TABLE II. FMEDA Table (Fragment)

FMEDA ID	Type of failure	Expected Detection during Normal Development Testing	Development Testing Required	Detection during Normal Use	Impact on PLC and Safety	Mitigation Action Required
RUF.01.1	Creates a command call to the wrong kind of block, but block type is valid to the UAL controller	May be blocked by the "compiler" if the argument list structure is different. Would be detected by a test of all AFB items (checking use on a schema to upload file).	Integrated test from items on a schema to the M1 tool output covering all blocks.	Subtle errors may not be detected.	This is potentially dangerous.	Use independent tool to generate an EU-readable assembly listing 1:1 with the upload file(s), and require the EU to review it for safety projects.

Stage_3. Separation of the failure types plurality into subsets (Table III).

TABLE III. Table of FF_SWCS

Faults, fsw_i	SW Components		
	SW Com_1	…..	SW Com_M
fsw_1	+		+
…			+
fsw_F	+		

Stage_4. Receiving the sets of *FF_SWCSf*$_i$, *FF_SWCS* and corresponding tables (Table IV). The feature of this stage is necessity of correct understanding and technological interpretation of FMEA-based types of failure.

TABLE IV. Table oFF_SWCS

SW $Comp_i$ subset types of failure, fsw_i	Attributes of SW fault insertion	
	Points, pf_i	Means, mf_i
fsw_1	pf_1	mf_1
…	…	…
fsw_z	pf_j	mf_F

III. INDUSTRIAL CASE

A. FPICS example

An example of FPICS is the FPGA-based platform RadICS that was developed by Research and Production Corporation Radiy. RadICS Platform is a new generation product. It is based on more than 10 years of RPC Radiy experience with digital I&C platform design, production, operation, and maintenance.

The procedures described in this section are part of the process of its SIL 3 – certification [6,10]. The platform is presented on Figure 1 a).

a) FSC chassis populate with modules

b) Logick Module (LM)

Fig.1. FPGA-based platform RadICS. a) FSC chassis populate with modules, b) Logick Module (LM)

The RadICS platform is composed of multiple types of modules, based on the use of FPGA chips as computational, processing, and system-internal control engines for each of the modules. In terms of its high level functionality and flexibility, the RadICS Platform is essentially a safety PLC, except that the internal logic is performed by FPGAs instead of microprocessors.

The typical safety-related I&C systems channel configuration, based on the RadICS Platform consists of one seismic-resistant Chassis, which contains Logic Module (LM) and up to 14 other RadICS modules (i.e., I/O and optical communication) in any combination of their types. The basic set of I/O modules types comprises Analog Input Module (AIM), Discrete Input Module (DIM), Discrete Output Module (DOM), and Analog Output Module (AOM). The Optical Communication Module (OCM) can be used to extend the system to multiple Chassis configuration. It is also possible to provide interchannel links between 2, 3, or 4 channels via fiber-optic communications directly between their LMs for coincidence voting. The RadICS Platform offers a fast response time ($\leq$ 10 milliseconds).

LM performs input modules data acquisition, execution of user application logic (UAL), drives the output modules, and processes diagnostic data from all I/O modules installed in the RadICS Platform Chassis. The I/O modules provide interfaces with other devices (e.g., detectors, sensors, actuators, signalization devices).

The seismically-resistant Chassis for RadICS Platform provides protected external interfaces to process I/Os, two independent power supply units (links), communications links, local inputs/outputs (from/to the built-in-Chassis/cabinet detectors/sensors/keys or indicators). Internal Chassis interfaces facilitate connections to the various modules that are installed within the Chassis by means of dedicated, isolated, point-to-point high-speed LVDS communication lines.

FPGA based platforms produced by RPC Radiy are used in the most critical and high-reliability demanding NPP applications such as Reactor Trip System (RTS), Engineered Safety Features Actuation System (ESFAS), and Rod Control System (RCS).

The functionality of each module is driven by the generic platform logic is developed with an FPGA-specific IDE and implemented with Hardware Description Languages (VHDL is used for RadICS platform implementation). The UAL is developed with a specialized IDE – Radiy Platform Configuration Tool (RPCT). RPCT provides means to manage large FSC-based Safety System (FSS) projects involving multiple FSC chassis that may be used to implement channelized or large complex systems. An FSS project may include many FSC chassis, in designing an application, the end-user selects the needed modules for each chassis, assigns them to a slot and then develops UAL that executes in the LM. The UAL is implemented with a graphical application-specific language that is very similar to Functional Block Diagram Language specified by IEC 61131-3. Figure 2 shows a fragment of UAL designed in RPCT.

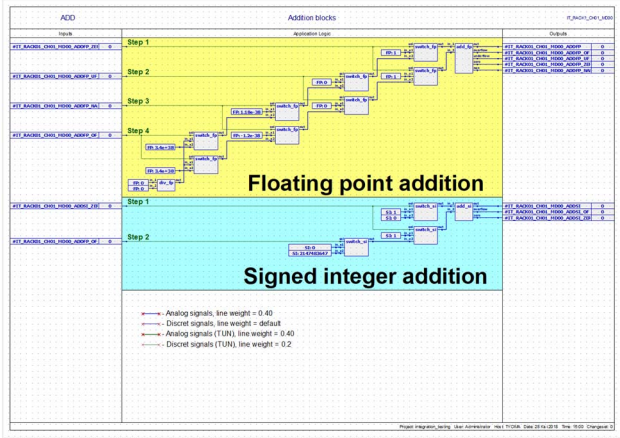


Fig. 2. A fragment of UAL designed with RPCT

Basic components of the UAL are Application Functional Blocks (AFB). Each AFB allows the user to select and use within UAL project some function: logical, mathematical, timing etc. Each "graphical" AFB in RPCT is bind to the hardware-implemented AFB component within FPGA-based LM, this component executes the specified function when FSC is in run-time. During the compilation, RPCT converts a graphical representation of the UAL into the UAL code that is uploaded to the FSC and executed during runtime. Mainly the UAL code consists of the following software instructions:

Reading data from the dedicated memory cells and putting it to a particular AFB.

- Execution of the AFB function.
- Writing results of the AFB function execution to the dedicated memory cells.

By the data structure, the UAL code is a hex byte code of Radiy's proprietary assembler for FPGA-based LM. The UAL code is a subject of SFIT described in this industrial case.

B. An Example of SW FIT Procedure

This subsection illustrates the implementation of SW FIT procedure. Suppose, the FMEA table contains a particular type of fault: fsw_1 – "Logic AFB was configured to perform a wrong function". Table V shows functions and function codes for Logic AFB.

TABLE V. Function Codes of Logic Block

AFB	Function	Function code
Logic Block	AND	1
	OR	2
	XOR	3

Assembler command "WRFBC" is used to configure a particular implementation R (number 0 for the considered example) of an AFB with number N (Logic Block has the number 1) to perform some function (see function codes of the Logic Block in Table V).

Table VI presents IF_SWCS table for the described particular fault.

TABLE VI. Example of the IF_SWCS TABLE

Subset types of failure, fsw_i	Points, Ipf_i	Means, Imf_{ij}
fsw_1 Logic AFB was configured to perform a wrong function	**Ipf_1** Function code field within assembler command	**Imf_{11}** Function code is WRONG and INVALID: - Function Code ≠ Function Code specified in the UAL project - Function Code is out of the legal range - Function Code = 4
		Imf_{12} Function code is WRONG and INVALID: - Function Code ≠ Function Code specified in the UAL project - Function Code is out of the legal range - Function Code = 0
		Imf_{13} Function code is WRONG but VALID: - Function Code ≠ Function Code that was specified in the UAL project - Function Code is within the legal range [1..3] - Function Code = 2 instead of 1 (OR instead of AND)

In order to insert the defined fault means (Imf_i) into the selected points (Ipf_i) a test designer shall perform the following:

a) Identify in the UAL data the point to insert a fault (a particular command and a particular data field in the command).
b) Change the correct data value to the wrong one.
c) Recalculate and replace checksums that protect UAL data from corruption.

Table VII shows the application of the above described steps for the considered example.

TABLE VII. Example of the FAULTS INSERTION

	Hex Code	Mnemocode
Init. command	128100020001	WRFBC LB.0, **funct_code <= 1**
Imf_{11}	AA8100020004	WRFBC LB.0, **funct_code <= 4**
Imf_{12}	D28100020000	WRFBC LB.0, **funct_code <= 0**
Imf_{13}	7A8100020002	WRFBC LB.0, **funct_code <= 2**

In order to perform the run-time testing of the design ad mitigation measures, which shall prevent the impact of the UAL errors the data that contains an inserted fault has to be uploaded to the LM. Figure 3 shows the actual results that were obtained for Imf_{11}, Imf_{12}, and Imf_{13} that were defined in Table VII. Figure 3(a) and Figure 3(b) demonstrate the front panel of LM that indicates the fault code – "#2D7 - Error of Logic Block", these states were caused by Imf_{11} and Imf_{12}. Figure 3(c) demonstrates the display of the on-line Monitor client (that is a part of the RPCT toolset) for UAL that contains the fault define with Imf_{13}. The FSC LM is not able to detect such a fault, but this fault can be identified during the UAL project testing.

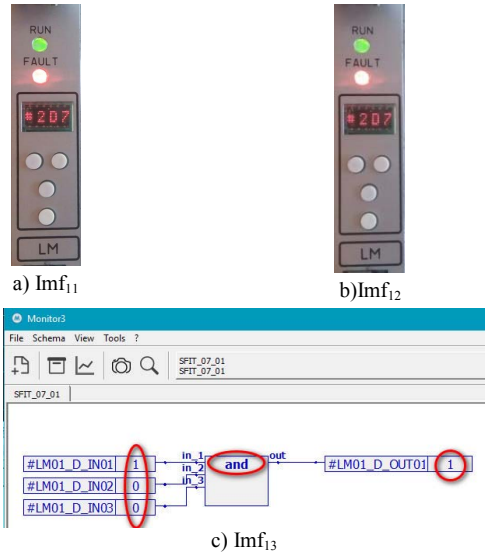
To insert various types of faults into the UAL code, a dedicated test support tool was developed by Radiy's V&V Team. The main functions of this tool are:

a) Decoding UAL commands in accordance with assembler of FPGA-based LM;

b) Recalculation of checksums for UAL commands, data frames, and UAL files that were edited during fault insertion;

c) Converting 32 bits string (in format FP32) to a decimal float value and backward.

During SIL 3 – certification of the RadICS platform up to 150 means (Imf_{ij}) for different types of faults were identified. FSC LM successfully passed testing against all the identified means.



a) Imf_{11} b)Imf_{12}

c) Imf_{13}

Fig. 3. Symptoms of the inserted faults

CONCLUSIONS

The tendency to modernization of NPP I&C systems based on the FPGA technologies is determined. The modified Software Fault Insertion Testing procedure and tools have been developed and described by RPC Radiy team. These procedure and tools used during verification and validation stages of development, testing and SIL-3 certification of safety-related I&C systems based-on RadICS Platform.

REFERENCES

[1] EPRI TR-1022983, Recommended Approaches and Design Criteria for Application of Field Programmable Gate Arrays in Nuclear Power Plant Instrumentation and Control Systems, Final Report, June 2011.

[2] EPRI TR-1019181, Guidelines on the Use of Field Programmable Gate Arrays (FPGAs) in Nuclear Power Plant I&C System, December 2009.

[3] IEC 61508:2010. Functional safety of electrical/electronic/programmable electronic safety-related systems [Text]. – Published. 2010 – 04. – IEC Standards, 2010. – 594 p.

[4] W. Goble, "The exida FMEDA Process - Accurate Failure Data for the Process Industries", exida blog, 2015.

[5] Hsueh, M., Tsai, T., Iyer. R., 1997, "Fault Injection Techniques and Tools", IEEE Computer, Vol. 30(4), pp. 75-82.

[6] Leveugle., R., 2000, "Fault Injection in VHDL Descriptions and Emulations", Proceedings of DFT'2000 Conference, October, pp. 414-419.

[7] Odarushchenko, O., Kharchenko, V., Sklyar, V, Ivasuyk, A. Fault-Injection Testing: FIT-Ability. Proceedings of East-West Design&Test Symposium (EWDTS"2013).- p.188-192.

[8] V. Kharchenko, V. Sklyar, O. Odarushchenko, O. Ivasyuk, Fault Insertion Testing of FPGA-based NPP I&C Systems: SIL Certification Issues, Proceedings of the 22nd International Conference on Nuclear Engineering (ICONE-22), Prague, Czech Republic, July 7-11, 2014, Paper ID: ICONE22-31163.

[9] Babeshko, E., Kharchenko, V., Odarushchenko, O., Sklyar, V. Toward automated FMEDA for complex electronic products. Proceedings of the International Conference on Information and Digital Technologies 2015.-p.17-22.