Implementation of a Fault-Tolerant System Using Safety-Related Xilinx Tools Conforming to the Standard IEC 61508

Emil Gracic, Ali Hayek, Josef Börcsök
Chair for Computer Architecture and System Programming
University of Kassel
Kassel, Germany
e-mail: {emil.gracic, ali.hayek}@uni-kassel.de

***Abstract*—Designing fault-tolerant systems implicates a redundant architecture of most critical system components. Traditional techniques - DMR and TMR, require a compromise with either incapability of a failed system masking or with an extremely high complexity and voter discrepancy. This paper presents a novel, warm redundancy architecture on FPGA based on a 32-bit microcontroller, which eliminates the drawbacks of DMR and TMR contributing their benefits. In order to mask a failed system, an enhanced diagnostic is implemented comprising CPU self-tests, internal and external watchdog, clock guarding and Xilinx soft error mitigation controller. Consequently, on-chip redundancy consists of two microcontrollers operating on the warm principle, in addition to an external watchdog that is designed to mask a failed system according to the status of diagnostics and to perform the activation of the redundant one. Increased availability is achieved due to the integration of two external watchdogs. Following the certified isolation design flow of Xilinx tools a conformance with the safety standard IEC 61508 is conducted. Furthermore, novel measures for mitigation of common cause failures are investigated to meet the requirements of the standard.**

Keywords-fault-tolerant; FPGA; warm redundancy; diagnostic; IEC 61508

I. INTRODUCTION

In the last two decades, a design of fault-tolerant systems in the functional safety area was highlighted by conservative requirements, but also by conservative extensions [1]. In this context, the International Electrotechnical Commission (IEC) 61508 (a worldwide organization that incorporates all national electro-technical committees) has been significantly limited in its capacity to design redundant systems since its first edition published in 1998 [2]. Therefore, it was only possible to implement a redundant architecture by integrating two or more separated hardware components.

The second edition of IEC 61508 from 2010 has recognized this restriction and introduced new methodologies for designing integrated circuits with on-chip redundancy (Annex E) and avoiding systematical failures during the development phase (Annex F) [2]. In this way, the functional safety area has become suitable for using Field Programmable Gate Array (FPGA) as a platform for developing of the fault-tolerant systems by exploiting all of its benefits (such as re-configurability, high performance, low development costs etc.).

In 2012 the study [3] outlined observations of aspects needed to be covered in order to implement on-chip redundancy using a FPGA platform. On the other hand, the study [4] offered concrete solutions for safety aware placement and routing developed to comply with the requirements of IEC 61508. Following this trends, Xilinx certified Isolation Design Flow (IDF) in 2013 by the certification authority TÜV SÜD in Munich, Germany, incorporating aforementioned aspects. Likewise another, worldwide approved FPGA manufacturer – Altera, certified its design flow and tools by TÜV SÜD in 2014.

With respect to the requirements of IEC 61508, Xilinx published a brief analysis about reducing the risk and increasing the efficiency for certified safety applications [5]. This analysis highlights a mitigation of Common Cause Failures CCF [1], [2], [5] and addresses the following areas as the most sensitive:

- I/O Banks
- Clock
- Configuration memory
- Power supply

This type of failures requires a particular attention because they can produce a serious damage of the system and replicating of the critical system components would not be sufficient for their elimination.

Although many research works dealt with the challenges of on-chip redundancy on FPGA [3]-[6], not a single work exists at the moment, which implements the on-chip redundancy and comprises the mechanisms for mitigation of CCF.

This research work presents concrete solutions for common cause failure mitigation of an on-chip redundant system based on two ColdFire microcontroller systems. In addition, a concept that is capable to handle the issue of distributing inputs and outputs of the redundant architecture is introduced. Furthermore, the concepts of the redundancy such as Dual and Triple Modular Redundancy DMR, TMR are briefly discussed and a novel concept of warm redundancy with an enhanced diagnostic system that is capable to mask a failed system is proposed.

A roughly background of this research work is outlined in the second section of this paper. Actual works dealing with the implementation of fault-tolerant systems according

978-1-5090-3278-5/16/$31.00 ©2016 IEEE

to the IEC 61508 are presented in Section III. The next sections deal with the essential parts of this work - the introduction of a novel redundancy concept and the realization of CCF mitigation mechanisms. Section VI evaluates the investigated fault-tolerant system in the context of functionality, fault detection degree and comparison with actual works. Finally, the last section concludes this study.

II. BACKGROUND

A. Redundancy

Redundancy, as a replication of most critical system components (or of the whole system), is a crucial aspect of designing fault-tolerant systems, because it can provide a system with the capability to continue operating even in the presence of faults.

Straightforward redundancy architectures DMR and TMR are widely used in fault-tolerant systems on FPGA. It is worth mentioning that IEC 61508 requires either a common diagnostic for all redundant system parts or an independent diagnostic for each of them. This contribution needs to be more considered for future research on FPGA, because some of the current studies [4], [5] and [7] only cover architectural issues. On the other hand, [3] doesn't offer the implementation based on IDF.

Triple Modular Redundancy is modelled on the triplication of a system. On FPGA, it results in a full redundancy of combinational and sequential logic and in inserting voters to determine which output should be proceeded as the correct one [8]. As a consequence of simplicity, based on the majority voting concept, TMR is very attractive. According to many authors the main benefit of this concept is the capability of detecting transient and permanent faults on the FPGA [8]-[10]. Lima et al. [11] briefly discuss the drawbacks of the TMR, such as area and I/O pad limitation, synchronization of voter inputs and power dissipation. This study offers an approach for improving the DMR due to a required implementation of concurrent error detection based on time redundancy. On the contrary, Ichinomiya et al. [8] criticize the concept of DMR addressing its disadvantage to mask a failed system and demonstrate the implementation of an improved TMR system.

In particular, the main disadvantage of the TMR is represented by the voting concept based on the majority principle (two vs. one). It doesn't provide a voter with the essential information based on quality and correctness (what happens, if two outputs are incorrect?). Therefore, it is quantitatively restricted on the majority score. A critical drawback is also its susceptibility to the single point of failure [9].

Another fundamental aspect of redundancy is manifested in the way how the redundant parts operate together. The following classification is given in [1]:

- Cold redundancy: the replicated system will be put in operation when the current operating system has failed.
- Warm redundancy: the replicated system runs in idle and will be put in operation when the current operating system has failed.
- Hot redundancy: the replicated system runs in parallel with the current operating system.

B. IEC 61508

This is an international safety standard with the scope to define a general, conceptual solution for design and moderation of a safety-related system during its whole lifecycle [2]. It consists of seven parts covering three main aspects in order to implement functional safety of electrical / electronic and programmable electronic systems:

- Methodologies for hardware design, verification and validation
- Methodologies for software design, verification and validation
- Documentation of each development phase

Due to its generality, the IEC 61508 serves as a fundamental concept for many industrial areas. ISO 26262 was published after the adoption of specific requirements for the automobile industry. Similarly, IEC 61511 was developed for the electrical, electronic and programmable electronic systems in the process industry.

As aforementioned, the second edition from 2010 intends to involve FPGAs and ASICs as platforms for designing fault-tolerant systems on a single chip.

Describing all parts of the norm is beyond the scope of this study. Therefore, we will briefly outline fragments of the part 2 – Annex A and E, which are crucially important from the FPGA and microcontroller perspective.

1) Annex A – Control measures for failures during operation

This annex takes into account techniques for controlling the failures that occur during the system operation and intends to estimate the diagnostic coverage from the system. For our purpose, microcontroller integration implicates a detailed consideration of the tests presented in the Table A. 1:

- Bus
- Memory management unit
- Direct memory access
- CPU: Registers, internal RAM, program counter, stack pointer
- Interrupt Handling

2) Annex E - Special architecture requirements for integrated circuits with on-chip redundancy

In addition to other fragments of part 2, this annex defines requirements for the on-chip redundant architectures. It also incorporates measures which increase (Table E.1) and decrease (Table E.2) a susceptibility to common cause failures. In order to design a fault-tolerant system on a single semiconductor substrate, the following has to be ensured [2]:

- The redundant components are physically, electrically and thermally separated by constituting them as independent channels/blocks with own inputs and outputs
- The sufficient distance between the boundaries of redundant components is guaranteed
- At least one CCF decreasing method is implemented
- CCF related to power supply can be avoided

- Diagnostic coverage of each separated block responds to a minimum of 60%

The susceptibility to common cause failures should be mitigated due to an implementation of methods linked to the concepts of diversity, testing, monitoring and temperature control. This system characteristic is referred to an estimation of the β–factor (βic) regarding the following conveniences [2]:

- A basic β_{ic} responds to 33 %
- Implementation of methods from table E. 1 increases and those from E. 2 decreases βic-factor
- Final estimated β_{ic} may not exceed 25%

C. ColdFire Microcontroller Version 2

The CFV2SPP is an integrated microcontroller subsystem that includes the ColdFire Core Version 2 (V2) and a set of peripherals from Freescale's Standard Product Platform (SPP) [12]. It is suitable for an investigation of fault detection measures as a consequence of its architecture which permits direct access to signals of the internal processor states. The structure of CFV2SPP is depicted in the Fig. 1.

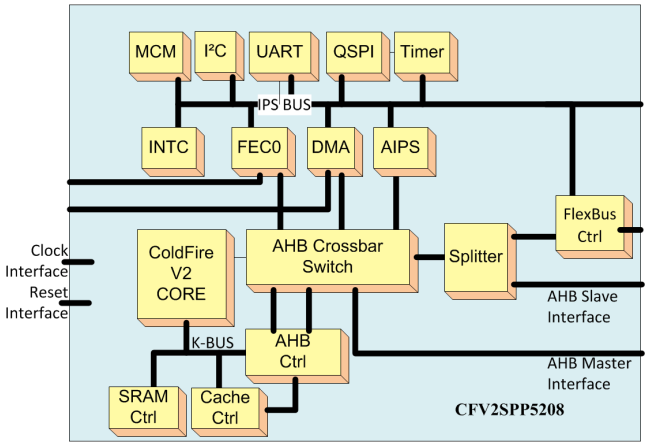

Figure 1. ColdFire microcontroller V2 with SPP

III. Related Works

Isolation Design Flow from Xilinx provides a certified concept and tools for the practical isolation of redundant components so that the two main requirements of the IEC 61508 – sufficient distance between boundaries of redundant parts and a physical, thermal and electrical separation – can be achieved effectively [6], [13], [14].

A. Xilinx Dual-Lockstep MicroBlaze Processor System

A safety architecture consisting of two Microblaze processors, comparator logic and various peripherals (I/O interfaces, external memory etc.) is presented in Hallet et al. [5]. All mentioned components are separated using IDF and verified with the Isolation Verification Tool (IVT). Article [15] demonstrates its implementation on Spartan6 FPGA.

According to [5] the system is able to detect single errors (transient and permanent) on the CPU and the internal memory. In contrary, it is not capable to detect external errors (flash memory, SDRAM etc.). As a consequence, a common cause failure may occur and cause a failure of both CPUs. Furthermore, Dual-Lockstep can be interpreted only as a fail-silent node incapable to operate correctly or to stop operating after a failure is registered [5].

B. Coarse Grained Mixed-Signal TMR Architecture

Girardey et al. focused in study [16] on an approach of the implementation of a safety-related system considering the requirements of IEC 61508. In addition, Corradi [7] demonstrates its adoption for IDF.

This system is based on the TMR concept. Three redundant components (channels) - Microblaze microcontroller, Digital Signal Processing module (DSP) and Analog Measuring module are operating in parallel dealing with the same task [16]. The results will be transferred to the voter, which selects a correct output respectively to the majority score [16].

The benefits of the proposed TMR system are diversity in system components as well as in platforms. The voter is capable to reconfigure the platforms dynamically if permanent faults do occur. Furthermore, the authors argued conclusively that the beta factor β_{ic} has been reduced to tolerable 25%.

On the other hand, the voter fails to eliminate the issue of single point of failures and more critically to eliminate the drawback of the majority concept (as discussed in Section II). While this study covers some areas of CCF, an intolerable gap related to clocking and power supply is still remaining.

IV. Implementation of a Novel Warm Redundancy Concept on FPGA

Contrasting the benefits and drawbacks of DMR and TMR the main challenge is to investigate a new concept, which incorporates only the benefits. One common criticism of DMR is the inability to mask a failed system – detection of permanent faults.

A. Masking of the Failed System

The CFV2SPP microcontroller provides direct access to internal status signals, which indicate whether the CPU operates correctly (pst_data and mfrz_b signals [12]). Using this advantage it is possible to identify random faults (transient and permanent) due to an implementation of intensive CPU self-tests, which directly affects internal status signals. Consequently, the failed CPU can be masked. In order to improve a fault masking concept and to increase the system reliability, we conducted a detailed diagnostic of the system by investigating:

- Internal watchdog timer capable to signalize a failure state of the CPU demonstrated in output reset signal ipg_swt_reset_b [12]
- Dynamic heartbeat signal generation based on internal DMA Timer and its external guarding
- External watchdog
- Soft Error Mitigation (SEM) Controller for the detection and correction of faults in FPGA configuration memory

B. *Self-Tests according to Annex A of the IEC 61508*

Implementation of all required segments from the Annex A would implicate the design of a safety-related operating system. This aspect is out of scope of this study. Therefore, we are going to perform enhanced CPU tests in order to increase the reliability of the aforementioned CPU status signals and fault masking capability.

An implementation approach of self-tests for microcontrollers has been analyzed in [17] taking into account the norm requirements. In the context of the CPU tests, it has to be extended since the register and arithmetic logic unit tests are the only covered. Therefore, we contribute the following tests:

- All addressing modes (nominal-actual value comparison, condition code test when possible)
- All CPU registers (walking-bit test)
- All data movement instructions (nominal-actual value comparison, condition code test when possible)
- All integer arithmetic instructions (nominal-actual value comparison, condition code test when possible)
- All logical instructions (nominal-actual value comparison, condition code test when possible)
- All shift instructions (nominal-actual value comparison, condition code test when possible)
- Bit manipulation instructions (nominal-actual value comparison, condition code test when possible)
- Program control instructions (nominal-actual value comparison, condition code test when possible)
- System control instructions (nominal-actual value comparison, condition code test when possible)
- All instructions of Enhanced Multiply-Accumulate Unit (nominal-actual value comparison, condition code test when possible)
- All registers of Miscellaneous Control Module in order to ensure correct configuration of internal watchdog
- All registers of DMA Timer in order to ensure its correct configuration
- Galpat test for internal SRAM

C. *External Watchdog*

This module is implemented in hardware description language (HDL). It involves CPU status, reset signal of internal watchdog timer, status of the SEM Controller and a heartbeat signal of DMA Timer.

In order to avoid a situation where the CPU hangs and permanently sends the same signal, we perform mutual signal generation of the DMA timer. In this case the heartbeat signal will be firstly configured to run four times slower than the system clock in period of 5ms. After this time a reconfiguration of heartbeat signal occurs (it is eight times slower than the system clock). This configuration runs also 5ms.

The watchdog provides an external, periodical signal which enables the counter based on the heartbeat signal in order to implement its capturing and validation each 5ms, but also a clock guarding since the DMA timer signal is based on the system clock. To keep it simple, only five flip-flops are used to shift the counter value and to control it. Two flip-flops are needed to generate status information immediately before and at the end of the timing window.

A contributed signal obtained from CPU status (improved by self-tests), SEM Controller, internal and external watchdog timer ensures a fault masking of the main and activation of the redundant CFV2SPP system.

D. *Hardware Design*

Comparing cold, warm and hot redundancy concept, it can be assumed that the hot one is not suitable for CCF avoidance, because both systems operate with the common clock. In this constellation, implementation of fault masking measures is not applicable since the parallel running systems are checking each other permanently through a comparator. On the other hand, cold redundancy leads to a critical consideration of a time required for activation and startup of a redundant system. Following this argumentation, the researchers chose a warm concept, where both systems are going to be initialized at the same time, but only the main system runs in application mode.

The redundant system operates in idle by periodical performing of self-tests. Furthermore, it waits for a signal of the external watchdog, which affects its transfer into the application mode. This concept is presented in the Fig. 3.

Xilinx Spartan6 SLX FGG900 2 FPGA is used as a hardware platform for the implementation of the warm redundant approach. Partial conformance to the requirements of the IEC 61508 can be achieved following the IDF respectively to [13].

Fig. 2 illustrates a basic concept of the fault masking related to the main system. When integrating a redundant system, two primary issues occur: How to perform activation of application mode and how to distribute input and output pads?

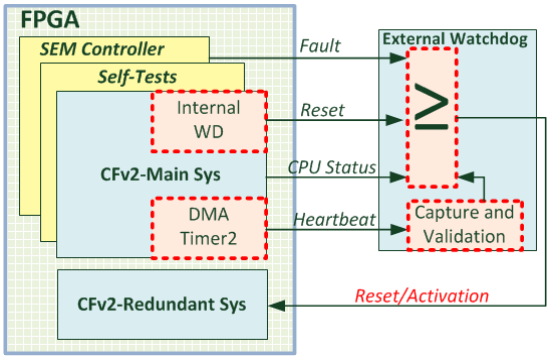

Figure 2. Basic cold redundant system with enchanced diagnostic and external watchdog

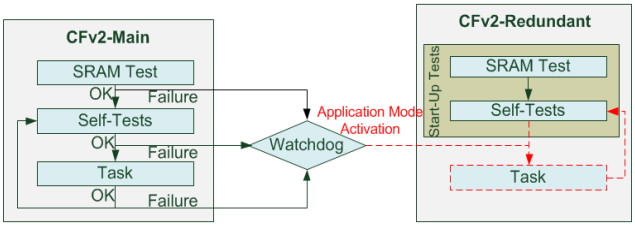

Figure 3. Implementation of warm redundancy concept

A redundant system is implemented to provide two operation modes – startup and application. By default, it operates in startup mode as long as the watchdog performs an activation of the application mode, as mentioned before. This feature is based on simple mode multiplexer within the system.

Direct pad assignment is not allowed regarding the requirements of the norm, as discussed in Section II. Xilinx IDF incorporates this aspect and defines a method of trusting routing for safe signal transfer between separated blocks based on Look-Up-Table (LUT) [13], [14]. Considering this constellation, two additional, separated blocks are introduced – Distributed Inputs to provide both systems with the global inputs and Safe Multiplexer to grant the systems an access to the global outputs. Since this aspect requires a particular attention, the researchers have implemented this module as a self-testing circuit with following characteristics:

- Redundant architecture
- Main multiplexer receives normal outputs from the CFv2 systems
- Redundant multiplexer receives inverted outputs
- Only a main multiplexer is connected to the pads
- Outputs from main and redundant multiplexer are connected to the parity checker, whose functionality matches to a XOR gate
- Enable signal of both multiplexer is controlled by external watchdog

To increase the availability of the system, two external watchdogs are integrated. The system is implemented using PlanAhead Tool of Xilinx and it consists of overall four separated blocks composing the on-chip redundancy.

On-chip redundancy validation is performed with Xilinx IVT. It results in the fulfilling of the first two requirements of the Annex E, as presented in Section II.

V. CCF Mitigation

To make the proposed system not only fault-tolerant but also conformed to the norm, mitigation mechanisms for all CCF sources outlined in the first section are investigated here.

A. I/O Banks

This source of the CCF is automatically eliminated following the IDF guidelines. Separated/isolated blocks composing on-chip redundancy may not own I/O-s from the same bank.

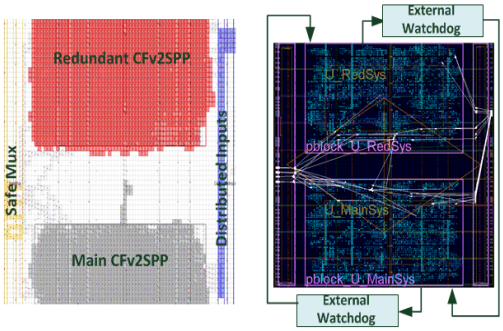

Figure 4. System implementation and validation with Xilinx tools

Furthermore, IO-Block, which connects the pad signal with the internal FPGA resources, has to be instantiated or inferred inside the isolated block [13], [14].

B. Clock

Common system clock is a very sensitive CCF source. In particular, it is not sufficient and adequate to define mitigation in this context by simple clock duplication as reported in [5]. Therefore, a detailed analysis of the clocking management on Spartan6 FPGA is conducted to gain a deeper understanding of capability for CCF mitigation. Spartan6 SLX FGG900 consists of 24 clock regions. The global clock network will be constructed due to 16 global clock multiplexers (BUFGEMUX). Eight multiplexers belong to the sixth clock region and the others to the seventh.

To enhance the clock autonomy of the main and redundant system, two separated clock sources are implemented as illustrated in Fig. 5. Furthermore, the clock signals from the both systems are connected to the diverse BUFGMUX of the sixth and seventh clock region.

C. Configuration Memory

Soft Error Mitigation (SEM) controller is an IP Core from Xilinx which can detect and correct errors registered in the configuration memory [10]. As built-in module for each separated/isolated module it increases the degree of CCF mitigation, facilitates the system monitoring and provides a partial dynamic reconfiguration of the failed system.

D. Power Supply

To mitigate CCF based on power supply it is not essentially important to integrate dual power supply, as established in [5].

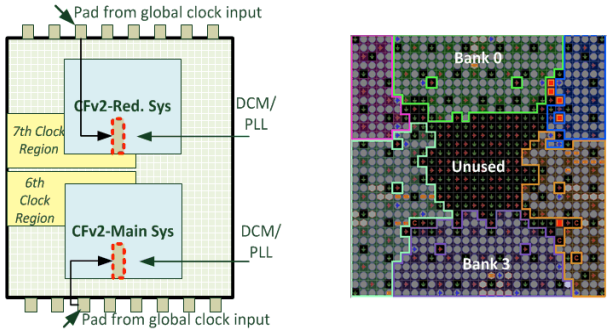

Figure 5. Clock and power supply CCF mitigation

Following IDF, the designer has a complete overview and a major influence on placement and routing allows him the elimination of common power supply sources. As Fig. 5 demonstrates, main CFv2SPP system is placed in the area of bank 0, while the redundant CFv2SPP occupies the area of bank 3. Between them is a gap of unused pins. The common cause failures are eliminated due to a conservative assignment only of those power supply pins, which belong to appropriate bank (0 for the main and 3 for the redundant system). Respectively, the unused area is extended on other banks with intention to avoid CCF for Distributed Inputs and Safe Multiplexer.

VI. EVALUATION

The impact of additional modules, Distributed Inputs and Safe Multiplexer, is examined as the first analysis. Detailed software tests over UART and Debug interface were performed without impact on the system timing and the complete architecture was successfully simulated with Cadence tools.

A diagnostic coverage DC (measure for fault detection) can be estimated in two ways – by performing a fault injection methods or by an integration of control measures for failures during the operation (Annex A).

Due to the implementation of self-tests, which conforms to the Annex A, DC for the CPUs can be estimated between 90-99%. As mentioned in hardware design section, self-tests will be executed immediately before the actual task, so that the processor and RAM faults can be detected at the right time.

Referring to the Safety Multiplexer, DC can be also assumed between 90-99% since it is redundant, comparable and testable.

Respectively to Annex E, an estimation of beta factor is performed as follows: 33% (basic value) + 2% (internal connections between blocks) – 6% (Separated power supply) – 4% (physical isolation and separation) = acceptable 25% [2].

Generally, the proposed system intends to reach safety integrity level SIL 2. From the IOS 26262 perspective, it corresponds with automotive SIL B and C.

TABLE I. COMPARSION BETWEEN THE PROPOSED SYSTEM AND RELATED WORKS

CCF Mitigation	**System**		
	Mixed-signal TMR	***Xilinx Dual-Lock***	***CFv2SPP***
I/O	✓	✓	✓
Clock			✓
Config. memory	✓		✓
Power supp.			✓

VII. CONCLUSION

By the implementation of enhanced diagnostic on a ColdFire microcontroller system and the investigation of the novel, warm redundancy architecture, this research work introduced a technique for the elimination of a traditional DMR drawback - incapability to mask a failed system. Due to a contribution of CCF mitigation measures and Xilinx IDF based design the system conforms to the IEC 61508. Further research will be conducted to extend the testing measures related to the other fragments of the Annex A and E in order to comply with all norm requirements for SIL2.

REFERENCES

[1] J. Börcsök, "Functional safety: basic principles of safety-related systems," Hüthig, Heidelberg, 2007.

[2] International Electrotechnical Commission, IEC/EN 61508: "International standard 61508: Functional safety, safety related systems", Second Edition, Geneva, 2010.

[3] A. Hayek, J. Börcsök, "SRAM-based FPGA design techniques for safety related systems conforming to IEC 61508 a survey and analysis," 2nd International Conference on Advances in Computational Tools for Engineering Applications (ACTEA), IEEE Conference Publications, pp. 319-324, Beirut, December 2012.

[4] R. Girardey, M. Hübner, J. Becker, "Safety Aware Place and Route for On-Chip Redundancy in Safety Critical Applications", Computer Society Annual Symposium on VLSI, IEEE Conference Publications, pp. 74-79, Lixouri, July 2010.

[5] E. Hallett, G. Corradi, S. McNeil, "Xilinx Reduces Risk and Increases Efficiency for IEC61508 and ISO26262 Certified Safety Applications", Xilinx White Paper, April 2015.

[6] J. D. Corbett, "The Xilinx Isolation Design Flow for Fault-Tolerant Systems," Xilinx White Paper, October 2103.

[7] G. Corradi, R. Girardey, J. Becker, "Xilinx Tools facilitate development of FPGA Applications for IEC61508", NASA/ESA Conference on Adaptive Hardware and Systems, IEEE Conference Publications, pp. 54-61, Erlangen, June 2012.

[8] Y. Ichinomiya, S. Tanoue, M. Amagasaki, M. Iida, M. Kuga, T. Sueyoshi, "Improving the Robustness of a Softcore Processor against SEUs by using TMR and Partial Reconfiguration", 18th International Symposium on Field-Programmable Custom Computing Machines, IEEE Conference Publications, pp. 47-54, Charlotte, May 2010.

[9] J. Anwer, M. Platzner, S. Meisner, "FPGA Redundancy Configurations: An Automated Design Space Exploration", 28th International Parallel & Distributed Processing Symposium Workshops, IEEE Conference Publications, pp. 275-280, Phoenix, May 2014.

[10] Xilinx, Inc., "LogiCORE IP Soft Error Mitigation Controller v3.1", Product Specification, October 2011.

[11] F. Lima, L. Carro, R. Reis, "Designing Fault Tolerant Systems into SRAM-based FPGAs", Design Automation Conference, IEEE Proceedings, pp. 650-655, June 2003.

[12] IPextreme®, "CFV2SPP5208 Integration Guide", December 2007.

[13] S. McNeil, "Developing Secure Designs with the Spartan-6 Family Using the Isolation Design Flow", Xilinx Application Note, June 2013.

[14] J. D. Corbett, "Isolation Verification Tool (IVT) Software User Manual", Xilinx User Manual, December 2013.

[15] T. Hardcastle, "Spartan-6 FPGA Dual-Lockstep MicroBlaze Processor with Isolation Design Flow", Xilinx Application Note, July 2012.

[16] R. Girardey, M. Hübner, J. Becker, "Dynamic Reconfigurable Mixed-Signal Architecture for Safety Critical Applications", International Conference on Field Programmable Logic and Applications, IEEE Conference Publications, pp. 503-506, Prague, September 2009.

[17] T. Tamandl, P. Preininger, "Online Self Tests for Microcontrollers in Safety Related Systems", 5th International Conference on Industrial Informatics, IEEE Conference Publications, pp. 137-142, Vienna, June 2007.