SRAM-Based FPGA Design Techniques for Safety Related Systems Conforming to IEC 61508

a Survey and Analysis

Ali Hayek and Josef Börcsök
Chair of Computer Architecture and System Programming
University of Kassel
Kassel, Germany
ali.hayek@uni-kassel.de

***Abstract*—With the announcement and development of safety standards such as IEC 61508 and DO-254 standards a basis for the implementation of qualitative and quantitative analyzes in the areas of reliability and safety for electronic safety-related systems was laid. Especially with the publication of the second edition of the standard IEC 61508 standard and the introduction of new aspects such as on-chip redundancy and the use of integrated systems in such systems is becoming increasingly attractive. SRAM-based FPGAs are considered as the mainstream FPGA technology and represent an excellent platform for the development of system-on-chips due to their complexity and programming flexibility. In this paper, the implementation of FPGA-based safety-related systems according to the standard IEC 61508 is targeted. First, the advantages and challenges of FPGAs for the use in such systems are presented. Afterwards, measures and methodologies are discussed, which are required for the implementation of such systems. Finally suitable FPGA implementation of these measures is presented.**

Keywords—Safety-related systems; FPGAs; IEC 61508; on-chip redundancy

I. INTRODUCTION

Nowadays embedded System-on-Chip applications are increasingly used in several industrial control processes. Due to their capability of implementing complex digital circuits and of rapid-prototyping, reconfigurable logic devices are also increasingly used in such processes compared to a decade ago. Thanks to the rapid development of the IP-Core market (Intellectual Property), whole communication microcontroller units can be today shortly integrated in such reconfigurable devices. Furthermore, due to their flexibility of programming and testing such circuits on run-time, reconfigurable devices provide furthermore a popular platform for safety-related self-repairing systems. This paper deals with the use of SRAM-Based Field Programmable Gate Arrays (FPGA) in safety-related applications, as the most widely used reconfigurable devices. Thus, according to Moore's Law, the number of transistors that fit into a certain semiconductor area doubles every 18 months. This implies that FPGAs are getting on one hand more powerful, since the latest Xilinx FPGA generation (Virtex 7) contains more than 2 million logic cells, equivalent to 20 million ASIC gates and feature novel architectural properties [1], which make them more interesting for complex safety-related systems. On the other hand, the latter argument not necessarily counts for safety-related hardware, which, in contrast, rather sticks at the same level of complexity, or computational power for a longer period. Indeed, with the increasingly smaller structures of such FPGAs, the susceptibility to external effects increases, such as the susceptibility to single-event upsets (SEUs). For this reason, the usage of SRAM-Based FPGAs in safety critical fields require the adoption of specific reliability and fault tolerance techniques, like redundancy and reconfigurability, in order to protect their functionality against such transient effects. Against this background, this paper deals with the use of SRAM-based FPGAs in safety-related systems conforming to established safety standards, such as the IEC 61508. In our opinion, there are two important points for dealing with this:

- Safety properties of the hardware description code downloaded to the FPGA and
- Safety of the FPGA Hardware itself.

In this paper we first introduce the safety standards which are relevant for this work. Especially the standard IEC 61508 [2] and its second edition (IEC 61508 Ed. 2) [3] are explained in detail. In addition, the standard DO-254 [4] from the field of aviation is introduced briefly, since it is widely used in the United States America and has parallels to the standard IEC 61508. Afterwards, the FPGA structure and an analysis for the use in safety-related applications are presented. The heart of this research work is the analysis of the use of SRAM-Based FPGAs according to the standard IEC 61508 Ed. 2. Latter is divided into two main aspects: Modeling and coding methodologies on software and physical measures on hardware. On the one hand, the implementation of standard techniques and measures on FPGA platform is motivated and discussed. This includes techniques for increasing the reliability of such systems like on-chip redundancy and placement and routing. On the other hand, the coding methodologies of FPGA programming languages such as VHDL are discussed. In this context, we study the possibility of the use of these languages for realizing safety properties. Furthermore, coding and verification measures are discussed in this section. Finally, the implementation of the proposed techniques will be presented on Xilinx FPGA hardware using

978-1-4673-2489-2/12/$31.00 ©2012 IEEE

representative safety-related architectures and recent work results.

II. Functional Safety Standards

Norms and standards for safety-related systems aren't new; MIL-STD-882 from the US Department of Defense (DoD) developed in 1963 [5], is the first standard in this area. This standard is derived from the military area. The idea was to improve the safety of weapons and to keep the risk of undesired accidental damage to people and the environment in an acceptable range. In 1998, a new paradigm was been developed with the standard IEC 61508, which has been associated with a new definition of the term "functional safety" [2]. The main innovation is that in the context of functional safety only the safety features of a system are considered. The other non safety-related functions are in accordance with the standard IEC 61508 only a part of quality management. In the following sections, the standard IEC 61508 is primarily introduced, as the safety standard applied in Europe. Furthermore, a short insight into the standard DO-254 is given. Latter is irrelevant for the current work but could be applicable in future considerations.

A. The Standard IEC 61508

The standard IEC 61508 is a standard in the area of safety technology, which was developed by the International Electrotechnical Commission (IEC), an international standards organization, and first released in 1998. It is titled "Functional safety of electrical / electronic / programmable electronic systems" (E / E / PES). The standard is also known as basic safety standard, because it is application independent, but it addresses all safety functions of a system. It is regarded as basis for further application-specific standards. The standard IEC 61508 is limited to electrical and electronic programmable electronic safety-related systems. In this context, it defines four safety-integrity-level, so-called SIL. This applies: the higher the SIL, the safer the E / E / PES. The specification of SIL provides developers, producers and customers a clear and unequivocal basis for negotiating basic aspects of safety integrity.

The standard IEC 61508 is generally seen as a basis for further standards. In this regard, the standard gives sufficient flexibility for technical respectively technological innovations. The standard is also kept consciously abstract and flexible in regard to the methods to cover the requirements on hardware and software, while the requirement is clearly defined, it leaves ample room for researchers and developers to apply own implementation ideas and makes them free of the need to comply with stringent rules. In the context of in this research work considered FPGAs, it gives for example a note on the requirements for on-chip redundancy (OCR) under point E.1 in Part 2 of the standard [3] and remembers that for example on the current state of the art FPGAs possibly don't meet the appropriate requirements. But this is only a hint, because it gives nevertheless the possibility that OCR can be targeted for FPGAs, under the condition that the corresponding requirements are satisfied. Furthermore, innovations find their way into new drafts of the standard. While OCR in the first standard edition was still unconsidered, it was contained in the following draft standards, and could thereby be taken into consideration by developers and certification authorities in terms of the standard.

B. The Standard IEC 61508 Ed. 2

The standard IEC 61508 is divided into 7 parts and provides a guide for developing safety systems. The specific implementation of the requirements is flexible. In the present work, the requirements for the development of safety-related systems based on FPGA conforming to the second edition of the standard IEC 61508 are mainly considered. In the following the main novel features in the second edition are described briefly. In the next section the applicable features for FPGAs are argued in detail:

- Clear definition of Systematic Integrity Compliance Route (Route 1_S, Route 2_S and Route 3_S)
- Clear definition of Hardware Integrity Compliance Route (Route 1_H and Route 2_H)
- New definition of Proven-in-Use terms
- New requirements for Application Specific Integrated Circuits (ASICs), which are the main category of FPGAs

C. The Standard DO-254

The standard DO-254 [4] is performed under the title "Design Assurance Guidance for Airborne Electronic Hardware" and is a standard for the development of complex electronic hardware systems in the aviation field. It was developed in April 2000 by the RTCA (Radio Technical Commission for Aeronautics) and EUROCAE (European Organization for Civil Aviation Equipment) and is today carried as a standard for the development of complex electronic hardware in the aviation field, of both by the American aviation authority FFA, as well as the European Aviation Safety Agency EASA demanded. The standard DO-254 is, like the standard IEC 61508, a safety standard, which is also application independent, but specifically refers only to the hardware development.

Like in the standard IEC 61508, it includes no binding guidelines for the direct implementation, but it lists conceptional guidelines for the intended certification of the whole development process. Outside the norm there are further standards, such as DO-178B [6], which deals exclusively with the software development in the aviation field. The standard specifies a complete documentation during development and takes into account the life cycle of the product. A consistent and binding implementation of the product life cycle from concept to decommissioning, as specified in the standard IEC 61508, however is not requested.

III. FPGAs in Safety-related Systems

A. FPGAs

SRAM-Based FPGAs are a special form of Programmable Logic Devices (PLDs) with the capability of implementing complex digital components in a short period of time [7]. SRAM-Based FPGAs are two dimensional arrays of complex logic blocks (CLBs) and flip-flops with electrically programmable interconnections between logic blocks and use traditionally fine-grained logic. A CLB is implemented using mutli-level logic, which gives it a more compact design compared to an implementation with two-level AND-OR logic [6]. A CLB of an FPGA can be configured in such a way that it can provide functionality of complex microcontroller units as shown later on. These logic blocks can be implemented by any logic components like: transistor pairs, combinational gates like basic NAND gates or XOR gates, n-input Lookup tables or Multiplexers. Fig. 1 shows the traditional structure of a SRAM-Based Xilinx FPGA.

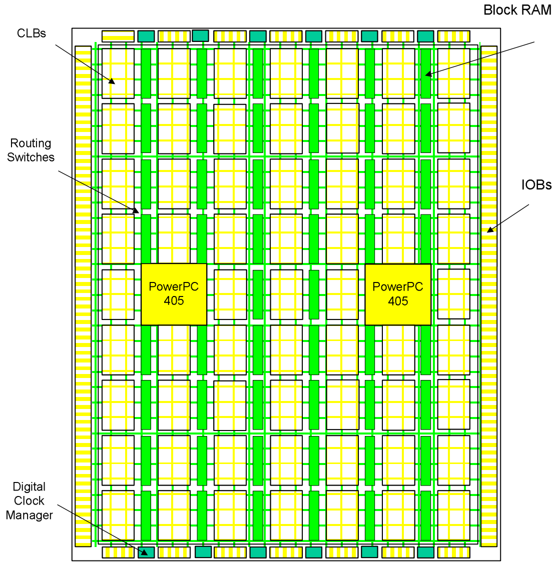

Figure 1. FPGA Architecture

B. FPGAs in Safety-related Systems

Due to the combination of their flexibility and efficiency requirements in the development of digital components, FPGAs has been used in most of the traditional embedded systems. FPGAs have gained popularity during the last decade because they provide many of the advantages from both hardware and software components. In terms of safety-related systems, FPGAs also provide a wide spectrum of properties which support developing such systems on reconfigurable devices. Primarily, an FPGA can be configured and tested in the same field and on run-time by the same developer. Due to this flexibility and the capability of implementing more than one component on a single chip, the use of FPGAs instead of more traditional hardware can drastically reduce the mean time to failure of the system. The next important point is the reconfigurability of SRAM-Based FPGAs. This is the capability of changing one or some components on run-time without changing the functionality of the whole system. In other words, in case of systematic or random failure in a FPGA, the design can be configured within some milliseconds in such a way that the faulty parts of the FPGA are not included in the placement of the components. In order to handle fault tolerance in FPGAs some techniques need to be introduced. As FPGAs are hardware devices programmed by software, they lie somewhere in between hardware and software. Therefore, when considering traditional safety evaluation techniques according to the standard IEC 61508, several techniques should be considered. These techniques can be categorized as redundancy-based, diagnosis-based and programming-based and are summarized in following points:

- Redundancy-based techniques rely on component duplication on system-level.
- Diagnosis-based techniques rely on additional processes that apply testing procedures on the design
- Programming-based techniques rely on the implementation of safety properties in the code which realize the system

IV. Coding Methodologies

In regard to the requirements of hardware and software, the FPGA development is double edged: On the one hand, FPGAs are hardware systems. On the other hand the development of FPGAs is mainly done by software configuration. Concretely, the FPGA description is usually written in so-called hardware description languages (HDL). Such a description is similar in many respects to classic programming languages. Usual representatives for HDLs are currently VHDL and Verilog. System-C is a target language to generate code for both hardware and software systems. In this work System-C is not considered, but pure hardware description languages. In this section, the methodologies according to the standard IEC 61508 Ed. 2 are presented, which arise on coding and verification level for design FPGA-based systems.

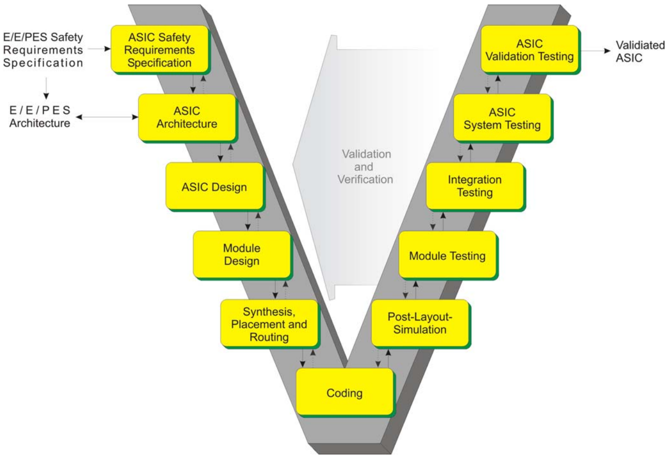

Figure 2. V Model conforming to IEC 61508

A. Safety-related Design Cycle

To develop safety-related systems on FPGA level, the standard IEC 61508 recommends an approach based on the V-model shown in Fig. 2. This is due to the fact that FPGA system development is not only a hardware development, but also a software development. In this context, requirements of both part 2 and part 3 of the standard IEC 61508 are considered for the used HDL code. This is especially in view of avoiding systematic faults important and useful. For this, appendix C in part 3 offers a guidance for quantifying the systematic safety integrity. More general requirements for safety-related FPGA design include:

- Clear, unambiguous, testable requirements;
- Traceable safety requirements specifications;
- Detailed hardware and software specifications for the used components; among others: Interfaces, Performance and response times;
- Requirements on systematic safety integrity:
  - Avoiding systematic faults according to standard IEC 61508-2 and -3 (Route 1_S),
  - Using of proven-in-use terms (Route 2_S),
  - Only software: requirements of the standard IEC 61508-3 (Route 3_S);
- Requirements on hardware safety integrity: to determine Route 1_H or Route 2_H;
- Systematic safety integrity: systematic ability of the elements of the safety functions, architecture-related restriction of max. SIL.

The three routes to avoid systematic errors mentioned above can be interpreted as described below.

For proven-in-use term the residual number of systematic error is assumed to be small enough. Proven-in-use terms are defined as those terms which were used long enough in similar projects. This primarily means that the field experience with the used terms should be conforming to the targeted SIL. Terms which are even already certified for the intended SIL can surely meet the requirements for systematic safety integrity following the Route 2_S. Route 1_S refers to hardware; in our case to the FPGA chip itself and its substrate, layout and manufacturing process, but also the HDL code. Although the latter also has software characteristics, it also applies for HDL code the requirements of the standard IEC 61508 Part 2. Route 3_S is reserved for the software running on the developed FPGA system, e. g. operating system and driver software.

In any case, measures and methodologies for avoiding systematic faults, and thus for increasing systematic safety integrity are treated in Appendix F of Part 2 of the standard. The main part is represented in tabular form, wherein some measures and methodologies for respective SIL are recommended or not recommended. Considering an FPGA design as software, Part 3 of the standard introduces different requirements for software.

B. *Coding Methodologies*

Considering an FPGA design as software, Part 3 of the standard introduces different requirements for software that are applied in HDL code. The most important in our view are listed below:

- Modularity
- Other methods to reduce code complexity;
- Programming conforming to following aspects:
  - functionality,
  - exchange of information between elements,
  - timing behavior,
  - timing constraints,
  - concurrency,
  - data structures,
  - Design-related assumptions and dependencies,
  - Exception Handling (on HDL-level: Wiring of interrupt control lines),
  - preconditions, invariants, results / post conditions,
  - comments;
- Ability to represent the design at multiple levels (structurally, functionally) - this is generally satisfied with HDL,
- Intelligibility.
- Testability (on verification and validation level).

In this context, the standard also requires the determination of suitable coding rules and naming conventions. But these are not specified, it is left to developers to define in advance useful guidelines.

Finally, for verification issues HDL tests are especially targeted. To illuminate this topic is beyond the scope of this document. For more information about this and about requirements on HDL code in general, see our related work in [9] and [10].

V. Implementation Methodologies

After the coding methodologies have been introduced in the last section, this section deals with the technical implementation of safety-related systems on FPGA hardware. In this context, the term “on-chip Redundancy” is introduced in detail. Furthermore, the requirements and implementation methodologies of OCR on FPGA are presented. Additionally, the handling with common cause faults is argued briefly. Finally an example for FPGA-based safety architecture is presented in the last section.

A. *On-chip Redundancy*

On-chip redundancy (OCR) is defined as a multiple (redundant) component implementation on a single chip. Hereby is generally not specified whether these components are active or passive redundant components. In Fig. 4 an example of a double on-chip redundancy is shown.

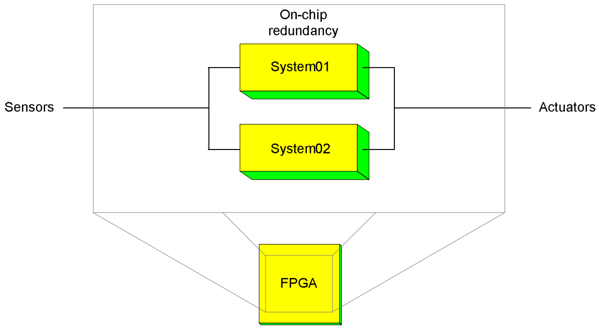

Figure 3. On-chip Redundancy on FPGA

For the purposes of functional safety one usually considers channels; over the entire loop from sensors to control logic and actuators. In this regard, OCR could be used in order to implement redundant control logic or even the whole loop without using multiple chips. In case auf FPGAs, the simplest example is the 1oo2 architecture (one out of two); therefore a single FPGA could be used to implement two processors channels and its needed diagnosis components. Architecture-related requirements for integrated circuits (ICs) in general with OCR are described in Appendix E in Part 2 of the standard IEC 61508. At a glance, it is noted that the requirements apply to purely digital ICs with common substrate. Furthermore, there are currently no requirements for ICs with a mixed design of digital and analog parts, so-called mixed-mode ICs, or even purely analog ICs. It is also noted that the standard in terms of OCR and in favor of safety is driving a more conservative course. For this reason, the maximum SIL is limited to SIL 3. Nevertheless, the most important requirements on OCR are the following:

- Restriction to SIL 3,
- No systematic features upgrading by combination,
- Consideration of random faults by temperature increasing,
- Physical channel separation by formation of separated blocks with "sufficient" distance to avoid short-circuits, for instance by electron migration and crosstalk,
- Short circuits and crosstalk between adjacent lines of different blocks must not lead to failure of a safety function,
- Measures to avoid faults caused by faulty power supply; e.g. Noise, crosstalk, high currents caused by short circuits,
- Connecting the substrate to ground, independent of the design process, for example n-well or p-well CMOS,

Due to the above requirements OCR for FPGAs, PLDs and other target platforms may not be feasible. But this is subject of current research, such as in [8].

From practical view, some of the requirements can be covered by concrete simulation runs for the target FPGA design, such as temperature propagation at maximum clock frequency. Other requirements, such as avoiding cross talk, can be covered by applying concrete formal assessments for the routing. For other requirements, such as noise impacts and the migration of soft errors, sufficient probability models or statistical experience results can be applied. For the minimum distance required between physical blocks experience values depending on FPGA type can be took into account. In any case, all these measures and methodologies have to be evaluated and fixed in agreements with the suitable certification authority, .e.g. TUV in Germany.

Finally, it is important to mention that fulfilling concrete measures and methods to cover the above requirements depends heavily form the target application and the target FPGA. In a failure mode and effects analysis (FMEA) and arrangement with the certification authority these aspects should be sufficiently considered.

B. Commom-Cause Faults

In addition to the previously considered methodologies, it is important to consider faults which have common cause, so-called common-cause faults, for the system evaluation. This is described in detail in the standard and will be touched upon only briefly here. For FPGAs with OCR a base-beta factor β_{IC} of 33% is assumed. By applying additional measures according to the tables given in the standard IEC 61508 this factor may increase or decrease. Thus the resulting beta coefficient of the FPGA is: $\beta_{FPGA} = \beta_{IC} + \Sigma$ modification. This shall not be higher than 25%. The beta coefficient affects directly the average probability of failure on demand (PFD_{avg}) of the FPGA, which is used for the SIL classification. More information on this can be found in the standard IEC 61508 [3]. In this context, the following aspects are to be considered:

- Recognizing uncontrollable faults - by diagnostic units, online tests, proof tests - needs to reach or holding the safe condition,
- For each channel and each singular executed monitoring component a diagnostic coverage (DC) of at least 60% should be achieved,
- Only diversely implemented (also differently designed) channels may monitor each other and thus improve as a watchdog the safe failure fraction (SFF) and DC,
- Homogeneous channels may only act as watchdogs for other channels if high SFF and DC has been already sufficiently reached,
- Tests regarding electromagnetic compatibility (EMC) with additional safety margin should neither impair the FPGA functionality neither destroy it
- Unsymmetrical wiring should be avoided as much as possible.

C. Safety-aware FPGA Implementation

This section describes the implementation of the measures and methodologies presented in the previous sections illustrated by a case example. In the context of a previous research work, the implementation of a redundant 1oo2 (one out of two) safety architecture with on-chip diagnostic has been presented in [8]. In Fig. 4 the block diagram of this architecture is shown. In this diagram, the single channels and diagnosis unit as well as the implemented measures according to section V A and B are mentioned. The physical separation and the establishment of guard zone are realized by using the Xilinx specific FPGA design tool “PlanAhead”. Each channel is placed in a separated power domain and has its own power supply pins. The routing between the channels is effected by the use of special Xilinx routing lines. The implementation on FPGA hardware is shown in Fig. 5.

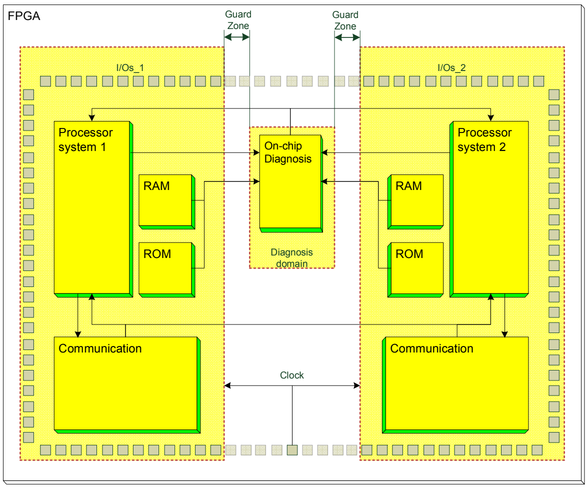

Figure 4. Safety-aware FPGA Implementation

Another aspect which will be addressed in the near future is offered through the presentation of the new Xilinx Virtex 7 series [1]. Those consist of a plurality of die on a substrate. This separation existing on FPGA can improve the separation of the safety systems implemented on the FPGA, in which the redundant channels can be placed on different silicon die blocks [11]. This feasibility of this implementation is part of an ongoing research, whose results will be published in a further paper.

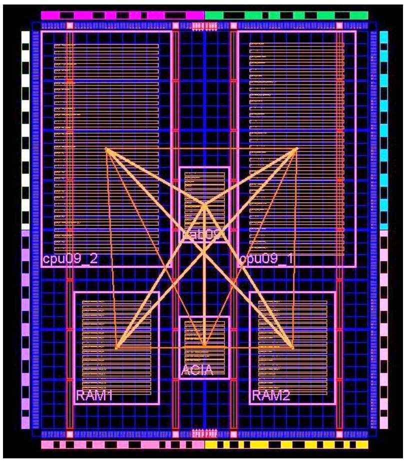

Figure 5. Safety-aware FPGA Implementation with PlanAhead

VI. Conclusion and Future Work

General requirements for the development of safety-related FPGA-based systems conforming to the standard IEC 61508 second Edition were presented in this paper. These include the development model for the implementation phase (V-model); requirements for the HDL code that both the requirements of Part 2 in the sense of an FPGA design but must also meet the requirements of Part 3 in the capacity of software. The same applies to tests on HDL level (so-called testbenches and verification models).

The determination of the maximum safety integrity level, which consists of the hardware and the systematic safety integrity of the individual elements, has been introduced. The possible routes to each – Route 1_S-3_S for the systematic and Route 1_H and Route 2_H for hardware safety integrity - were discussed briefly. It was noted that the upgrading of the systematic safety integrity by grouping for OCR is not allowed. Also noted was that the quantification of systematic safety integrity may not be clearly possible. For this purpose, methodologies and measures to avoid systematic faults were presented and recommendations for the FPGA implementation for the targeted SIL were made.

In a similar context, the handling with common cause faults was introduced. Techniques and measures to improve the beta factor were presented and discussed in terms of the standard IEC 61508.

Finally, the implementation of the key requirements mentioned above on FPGA platform was illustrated with an example of a safety-related 1oo2-architecture. Particular attention was paid to the separation channel by power domains and guard zone.

Future research work will be on one hand the targeting of novel FPGA architectures such as Xilinx Virtex 7. On the other hand, the calculation and evaluation of FPGA systems for safety-related applications according to the standard IEC 61508 will be targeted.

References

[1] Xilinx Inc., "DS180: 7 Series FPGA Overview," Advance Product Specification, published by Xilinx Inc., 2012

[2] International Electrotechnical Commission, IEC/EN 61508: International standard 61508 functional safety: safety related systems, Geneva; 2005

[3] International Electrotechnical Commission, IEC/EN 61508: International standard 61508 functional safety: safety related systems: Second Edition, Geneva; 2010

[4] RTCA Inc., "DO-254: Design Assurance Guidance for Airborne Electronic Hardware," Washington D.C., USA , 2000

[5] Department of Defense, "MIL-STD-882: Standard Practice for System Safety," Washington D.C., USA, 2000

[6] RTCA Inc., "DO-178: Software Considerations in Airborne Systems and Equipment Certification," Washington D.C., USA , 1992

[7] S. Kilts, "Advanced FPGA Design: Architecture, Implementation, and Optimization", Wiley & Sons, Wiley-IEEE Press, August 2007

[8] J. Boercsoek, A. Hayek, B. Machmur, M. Umar, "Design and Implementation of an IP-based Safety-related Architecture on FPGA", XXII International Symposium on Information, Communication and Automation Technologies (ICAT), Sarajevo, Bosnia and Herzegovina, 2009, IEEE Conference Publications, pp.1-6

[9] A. Hayek, M. Schreiber, J. Boercsoek, "Basic VHDL tests conforming to IEC 61508", Seventh International Conference on Networked Sensing Systems (INSS), Kassel, Germany, 2010, IEEE Conference Publications, pp.41-44

[10] A. Hayek, M. Schreiber, J. Boercsoek, "Sophisticated VHDL testing conforming to IEC 61508", 24th International Congress on Condition Monitoring and Diagniostics Engineering (COMADEM), Stavanger, Norway, 2011

[11] P. Dorsey, "Xilinx Stacked Silicon Interconnect Technology Delivers Breakthrough FPGA Capacity, Bandwith, and Power Efficiency," White Paper, W380, Xilinx Inc., 2011