Approach for a Safe-SoC for Cyber-physical Application according to IEC 61508

Josef Börcsök
ICAS, Institute for Computer Architecture and System Programming
University of Kassel
Kassel, Germany
j.boercsoek@uni-kassel.de

Waldemar Müller
ICAS, Institute for Computer Architecture and System Programming
University of Kassel
Kassel, Germany
w.mueller@uni-kassel.de

Eike Hahn
ICAS, Institute for Computer Architecture and System Programming
University of Kassel
Kassel, Germany
eike.hahn@uni-kassel.de

Michael Schwarz
ICAS, Institute for Computer Architecture and System Programming
University of Kassel
Kassel, Germany
m.schwarz@uni-kassel.de

Mohamed Abdelawwad
ICAS, Institute for Computer Architecture and System Programming
University of Kassel
Kassel, Germany
m.abdelawwad@uni-kassel.de

***Abstract*— Using electronic systems to control complex applications has found its way into nearly all technical and industrial areas during the last four decades. Today, in addition to system size, reduced system costs, optimized energy consumption and high reliability or safety, the aspects of functional safety are increasingly in the focus of many applications. Especially concerning safe embedded cyber-physical systems, which is favored by increasing integration of components, these aspects are of central importance. This article describes a consistently safety 1oo2 SoC architecture model (a miniaturized safety system on a chip) based on a modified software comparator architecture. The design and realization of the safety SoC, according to IEC 61508, are also presented.**

Keywords*— *Functional Safety, Miniaturized Safety Systems, Safety SoC, Safety Chip Design

I. INTRODUCTION

Due to the rapidly increasing use of embedded systems in safety-related applications (vehicles, medical technology, mechanical engineering, process industry), functional safety issues are becoming more important. The reason is that the normative or legislative requirements in these areas are increased [1].

Various large semiconductor manufacturers have in the meantime recognized this major market trend, especially in the fields of autonomous driving, semi-autonomous driving and collaborative robotics. Currently, there are various architectures and approaches to implement microcontrollers or microprocessors as redundant systems for safety-related applications [2]. However, partly due to interpretable safety standards, chip designers have only followed the hardware lockstep concept. The principle can generally be described as follows. Two identical computing units are integrated, which both execute exactly the same program sequence and their clocks are synchronous. For monitoring and diagnosis, these approaches use a hardware comparator that compares the execution of each command of the redundant architectures. If a discrepancy is detected, safety measures are generally taken to reach a predefined safe state of the system (usually de-energize the system).

This architectural concept is used in a similar way in series of manufacturers of safety systems on a chip such as Texas Instruments [3], Infineon [4], Renesas [5], NXP [6], and Freescale [7]. However, in these approaches, an ESD (Electronic Shut Down) system is realized. In the many applications, however, precisely this behavior is undesirable or problematic, e.g., in autonomous driving and human-robot collaboration. In addition to these problems, chip design is partly difficult due to the complex timing requirements of the Lockstep architecture. This architecture concept was also followed for more than 14 years in the Institute of Computer Architecture and System Programming at the University of Kassel to implement several industrial projects [8, 9] up to full certification up to SIL 3 according to IEC61508 [10].

A new architectural approach was developed in the Institute of Computer Architecture and System Programming. In this approach, the same number of computing units (three processors) is used to detect malfunctions in the redundant computing unit. However, the system is not forced to be de-energized, but to continue the operation with a reduced safety integrity level or reduced functionality.

The ReSCU-V1 SoC architecture model presented in this paper is based on a standard-compliant 1oo2 architecture [11] that complies with the IEC 61508 safety standard. In the following section, the SoC architecture model of the ReSCU-V1 is described, where the comparison to existing architecture models is considered. In Section III, the peripheral units connected to the safety architecture are presented with regard to the safety-related requirements and compared with existing structures on the market. Section IV introduces the verification process and Section V gives a general overview of the physical design and the final chip layout of the ReSCU-V1.

II. ARCHITECTURE MODEL OF RESCU-V1 SOC

The relevant safety standards demand several technical requirements for "safety chip design" that shall not be underestimated. Although the basic model (Fig. 1) seem easy to implement initially, the difficulty of implementation is usually hidden in the details. In addition to the pure hardware requirements, there are also a number of requirements regarding fault tolerance, fault detection and diagnostic measures that must be met for use in a safety-related application of such an architecture (Fig. 2). Furthermore, the failure rates of the individual architecture components λ_{DU}

(dangerous undetected failures rate) and λ_{DD} (dangerous detectable failures) have to be considered.

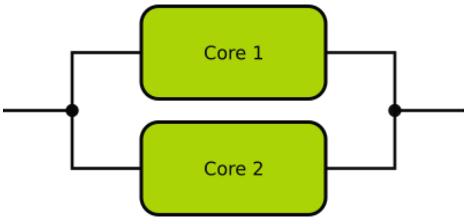

Fig. 1: Principle redundant structure of a 1oo2 processing unit.

Some of the measures mentioned above cannot be realized by hardware measures alone; therefore, the normative requirements of the development of safety-related software also come into play here. The basic approach to SoC redundancy is necessary to achieve the hardware fault tolerance required by the standards. However, in addition to the actual redundancy of the processing units, the peripherals' consideration must also be taken into account to achieve the overall safety of the SoC. A further important aspect of the redundant structure is the requirement from safety standards for freedom of interference among both channels.

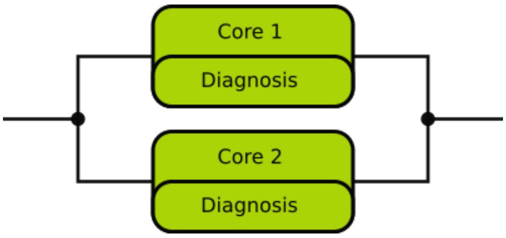

Fig. 2: Principle redundant structure of the processing unit with diagnostic measures.

A double redundant system has only one ß-factor, as shown in Fig. 3. A 2oo3 system is shown on the right-hand side as an example, which is often discretely constructed and shows four ß-factors for a triple-redundant system. The factor $ß_0$ refers to failures that affect all three channels simultaneously. The other three factors $ß_1$, $ß_2$ and $ß_3$ indicate the sensitivity that failures occur in two of the three channels due to a common cause.

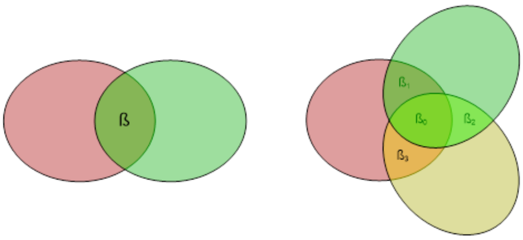

Fig. 3: Possible common cause failures for a double and triple-redundant system.

Mathematically, the calculation of the average probability of failure on demand (PFD_{avg}) of an SoC architecture 1oo2 according to the safety standard IEC61508 is formulated as [10]:

$$PFD_{avg} = 2(1-\beta)\lambda_{DU}[(1-\beta)\lambda_{DU} + (1-\beta_D)\lambda_{DD} + \lambda_{SD}]t'_{CE}t'_{GE} + \beta_D\lambda_{DD}MTTR + \beta\lambda_{DU}\left(\frac{T_2}{2} + MTTR\right) \quad (1)$$

Since calculations with several ß-factors are complicated and the ß-factors are often only based on expert analyses and assumptions, all ß-factors are combined to one ß-factor. In the further course of the work, the ß-factor refers to the common cause failures that simultaneously affect all system channels. Due to these circumstances, a safety consideration for dangerously occurring failures of 1oo2 redundant architecture is now possible using the FTA, as shown in Fig. 4.

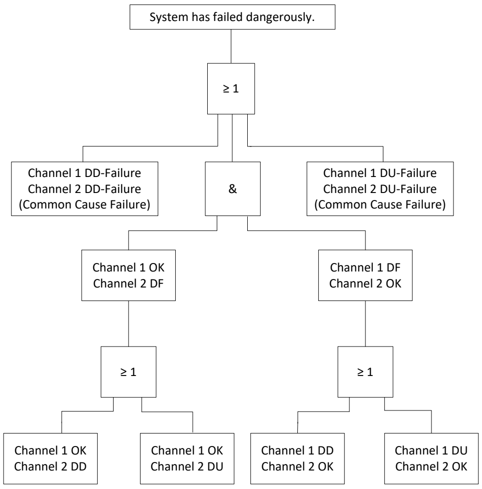

Fig. 4: General FTA consideration of a 1oo2 SoC architecture.

For the realized SoC of the ReSCU-V1, an IP from IPextreme is used, i.e., ColdfireV2. The structural design of the central processing units is realized according to Fig. 2 by means of a 1oo2 redundancy concept. However, to achieve the highest possible synchronicity, both processing units are supplied by a common (externally monitored) clock source, as shown in Fig. 5. Both architecture branches have their own memory, which does not affect the other redundant unit. The separate external memories allow each processor on demand to execute different programs.

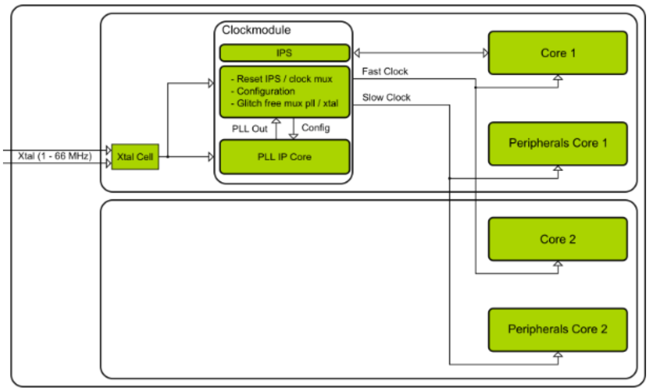

Fig. 5: Clock generation of the ReSCU-V1.

In contrast to the typical implementation of a hardware comparator, which strictly compares temporary states of the two processing units, the comparison of the states here is performed by a software comparator. For this purpose, both processing units can communicate via a communication interface, as shown in Fig. 6. At the same time, the basic functionality of the respective redundant system is indicated via separate "life signals".

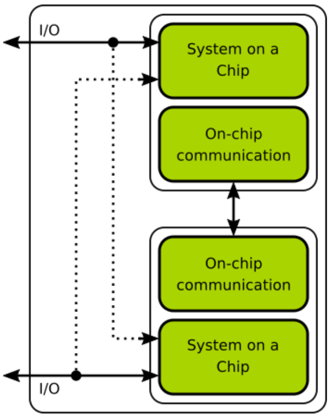

Fig. 6: The fully redundant architecture of ReSCU-V1.

The actual comparison then takes place separately in each unit, where the deviations from determined results are also transmitted via the communication interface. This not only allows a good-bad statement about the results of individual processing units but also enable assumptions on the state of the safe system. Thus, it is possible to detect and shut down sub-sections that are not functioning correctly by means of implemented diagnostic measures, but not to shut down the entire system completely. Depending on the required safety level, this is accompanied by a degradation of the safety level, which can be temporarily accepted in many applications due to the possible given functional requirements, e.g., to enable emergency running functionalities.

III. PERIPHERAL ARCHITECTURE

In almost all approaches for functionally safe SoCs on the market today, little or no special attention or architectural measures are paid to the peripheral units. However, this is where the most significant problems are to be expected from a safety perspective. Studies from the past have shown that problems can be seen, especially in signal acquisition and signal validation or signal output.

Based on this finding, an entirely consistent 1oo2 architecture approach is selected to design the functionally safe SoC ReSCU-V1 presented here, which is also implemented consistently for the digital inputs, outputs and GPIO (chip-side), as shown in Fig. 7. Besides, timer/counter modules are implemented redundantly so that frequency measurements, as well as the safe generation of PWM signals, are possible with the ReSCU-V1. This approach ensures a fully redundant safety system, according to IEC 61508.

To be able to use such a safe SoC in a cyber-physical environment, a communication structure that is as versatile as possible is required. With the presented SoC ReSCU-V1, the entire peripheral modules, consisting of standard communication interfaces such as I2C, SPI, UART as well as Ethernet, are implemented redundantly in the system. Here, too, the approach followed is that each processing unit has all the peripheral modules contained in the system.

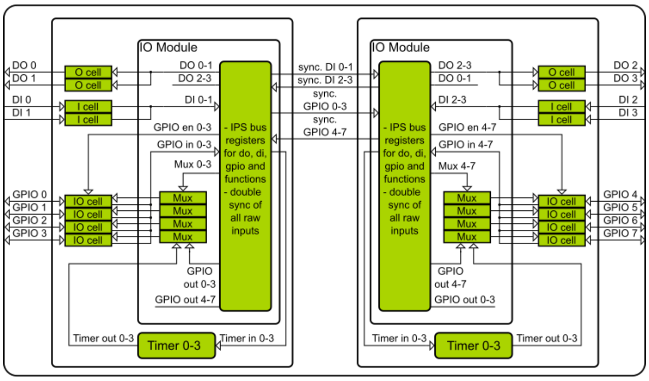

Fig. 7: Redundant I/O and GPIO architecture of ReSCU-V1.

IV. VERIFICATION

A standard-compliant and approvable procedure is essential for verification and validation. For the functional verification of the developed and implemented safe SoC, it is first simulated at register transfer (RT) level and later at gate level. In addition, functional tests are performed on FPGA basis.

For each implemented module, a code review is performed first. Thereby the compliance with the previously defined coding rules is of utmost importance. These are based on the rules specified by the DO-254 standard. Afterward, a testbench based white box test is performed on RT level. With the help of appropriate tools, the modules' test coverage of greater than 99% is achieved. To achieve conformity with the standard, the test is performed by an independent, competent person who is not involved in the implementation. The goal is to perform the tests independently and to avoid systematic failures.

After the intensive testing of all modules, they are integrated into the SoC design. After successful integration, appropriate tests are performed. Test programs are developed to test the memory and peripheral modules for their functionality. These test programs are placed in simulation models of flash memories, attached to the the system only for simulation in test bench, and also executed from there. The goal is to examine the correct adaptation of the modules to the bus provided by the CPU. The test programs are triggered or checked by signals generated in the testbench and connected to the peripherals.

Finally, the system is simulated again at the gate level after the Place and Route (P&R). For this purpose, the test programs already implemented for the integration test are reused and simulated again. The difference to the integration test on RT level is that the specific time behavior of the system, which is influenced by the P&R, is used for the simulation. This simulation shows if the system keeps the given time behavior or if timing requirements are violated. Since this simulation is very computationally intensive, care must be taken to optimize the test programs' size.

V. PHYSICAL REALIZATION

The presented safe SoC ReSCU-V1 is realized with a 180 nm CMOS process on a chip area of 5x5 mm. The measures for avoiding common cause failures for the integrated circuits with on-chip redundancy as specified in the IEC 61508, Part 2 Annex E standard were considered.

In addition to the defined functionality, integrated circuits have to meet additional requirements in order to be used in safety-related applications. For the redundant processing channels or units, freedom from interference must be guaranteed so that the redundant processing channels cannot influence each other. This leads to the requirement that hardware failures in one processing channel cannot influence the other processing channel, and thus, no crosstalk between signal lines may occur. This inevitably leads to a spatial separation of the processing channels on the chip, but also to the requirement for separate power domains, as shown in Fig. 8.

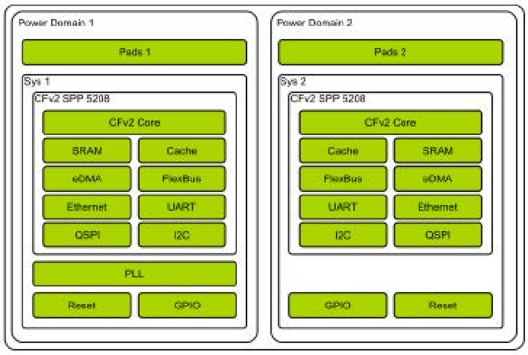

Fig. 8: Separated power domains of the ReSCU-V1.

In order to guarantee freedom from interference, IEC 61508 requires that the distance must be sufficient to avoid short circuits or crosstalk between the channels. The safety factor should be between 10 and 50. For the 180 nm process used, the specified minimum distance between the metal layers on Metal-1 is 0.56 µm and on Metal-6 is 1.6 µm. It is assumed that this distance could be used as a basis for calculating the distance. With a safety factor of 50, this results in a distance of 80 µm. This was rounded up to 100 µm from a conservative safety point of view.

As a further measure, care is taken to ensure that the IO cells and the routing to them are also placed separately. The IO cells of both channels also have their own power domain. CUT cells are placed between the two IO power domains, which further reduces the possible interference between the channels and thus simultaneously follows the design rules.

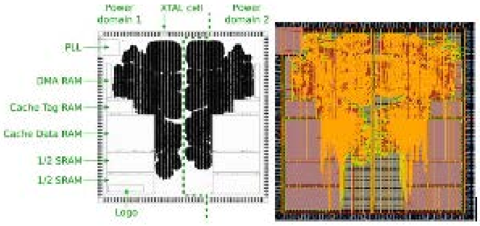

Fig. 9: Physical layout and floor plan of the ReSCU-V1.

The integrated memory blocks are also placed as far apart as possible to avoid common cause failures. After successful floor planning, the design is placed and optimized according to the timing constraints. Thus, the design is optimized for a clock speed of 120 MHz and should function stably with this clock over the entire temperature range from -40°C to +125°C.

The physical layout and the resulting floor plan with routing is shown in Fig. 9 while the ReSCU-V1 silicon die and in CPGA package is shown in Fig. 10. The symmetrical, spatial division of the two processing units can be seen. Only the macroblock of the PLL is singular in design.

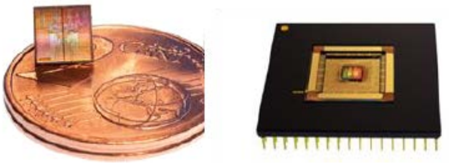

Fig. 10: ReSCU-V1 silicon die and in CPGA package.

VI. CONCLUSION

In summary, it can be stated that the presented safety SoC ReSCU-V1 meets the requirements of a fully redundant safety system. The structure was realized by two processing units based on 32-Bit ColdFireV2 microprocessors with redundant peripheral devices. The necessary physical separation of the two units was realized by two power domains, whereby a thermal and electrical (no galvanic) separation of the domains is achieved. The design of the SoC ReSCU-V1 allows applications in safety applications in mechanical engineering (EN ISO 13849), automotive engineering (ISO 26262) and railroad engineering (CENELEC 5012x).

An IO module for redundant DI, DO and GPIO with alternative functions were designed and implemented especially for the SoC ReSCU-V1. The partly configurable input/output concept offers high flexibility with regard to a flexible and easy to realize the adaptation of possible tasks. Compared to conventional structures in SoC's it offers great advantages in the safe connection of digital input and output signals.

Various interfaces are available for communication and offer a wide range of communication options. This makes the ReSCU-V1 the ideal safety core in cyber-physical, embedded and Safe-IoT applications.

FreeRTOS and a Linux operating system were adapted as test platforms to demonstrate the flexibility of the presented SoC. In this context, a programming platform according to IEC 61131-2 was adapted in cooperation with LogiCals in order to be able to perform simple programming in compliance with the standard. In the meantime, various demonstration applications have been realized with the SoC ReSCU-V1, such as a safety-related control unit for trailer couplings according to ISO 26262 [12].

In the next step, further development of the SoC ReSCU-V1 will be carried out. Here, special emphasis will be put on the possibility of optimizing and extending the I/O range.

REFERENCES

[1] J. Börcsök, *Electronic safety systems: Hardware concepts, models, and calculations*. Heidelberg: Hüthig, 2004.

[2] A. Hayek and J. Börcsök, “Safety-Related ASIC-Design in Terms of the Standard IEC 61508,” in *The Third International Conference on Performance, Safety and Robustness in Complex Systems and Applications (PESARO)*, Venice, Italy, 2013.

[3] Texas Instruments, *Safety Manual for TMS570LS31x and TMS570LS21x Hercules™ and ARM® Safety Critical Microcontrollers: User´s Guide*.

[4] Infineon Technologies AG, *Highly Integrated and Performance Optimized 32-bit Microcontrollers for Automotive and Industrial Applications*. Neubiberg.

[5] Renesas Electronics, *Microcontroller series for innovative SIL3/ASILD chassis applications*.

[6] NXP, *MPC5744P Data Sheet*. Accessed: Nov. 19 2020. [Online]. Available: https://www.nxp.com/docs/en/data-sheet/MPC5744P.pdf

[7] Freescale Semiconductor, *Qorrivva MPC5643L Microcontroller: Data Sheet: Advance Information*.

[8] A. Hayek, B. Machmur, M. Schreiber, J. Börcsök, S. Gölz, and M. Epp, “HICore1: “Safety on a chip” turnkey solution for industrial control,” in *2014 IEEE 25th International Conference on Application-Specific Systems, Architectures and Processors*, 2014, pp. 74–75.

[9] A. Hayek and J. Börcsök, “On-chip safety system for embedded control applications,” in *MELECON 2014 - 2014 17th IEEE Mediterranean Electrotechnical Conference*, 2014, pp. 315–319.

[10] *IEC 61508-6:2010 - Functional safety of electrical/electronic/programmable electronic safety related systems - Part 6: Guidelines on the application of IEC 61508-2 and IEC 61508-3*, IEC, 2010.

[11] J. Börcsök, *Functional Safety: Basic Principles of Safety-related Systems*, 1st ed. Heidelberg, Neckar: Hüthig Verlag, 2006

[12] Universität Kassel - ICAS, *Sicheres Steuergerät für Anhängerkupplungen nach ISO 26262 bis ASIL-B*. [Online]. Available: http://www.uni-kassel.de/eecs/fachgebiete/icas/forschung/automobiltechnik/sicheres-steuergeraet-fuer-anhaengerkupplungen.html (accessed: Nov. 23 2020).