Czech Technical University in Prague
Faculty of Information Technology
Department of Digital Design

Dependable design methods for programmable circuits with respect to area overhead

by

Pavel Vít

A dissertation thesis submitted to
the Faculty of Information Technology, Czech Technical University in Prague,
in partial fulfilment of the requirements for the degree of Doctor.

Dissertation degree study programme: Informatics

Prague, August 2017

Supervisor:
doc. Ing. Hana Kubátová, CSc.
Department of Digital Design
Faculty of Information Technology
Czech Technical University in Prague
Thákurova 9
160 00 Prague 6
Czech Republic

Abstract and contributions

This dissertation thesis deals with a problem in current FPGAs, which is caused by SEUs. The problem is that a particle can change a bit in the configuration memory of the FPGA and dramatically change its function. Therefore it is difficult to use these chips in mission critical applications like space programs or even railway transportation. Faults caused by SEUs are classified as soft errors, because they could be repaired by loading of a new configuration. This opportunity is used in proposed methods, only a part or a whole FPGA is reconfigured and repaired.

The probability of a SEU increases with the size of the design (used bits). Presented methods are focused to achieve the minimal area overhead. The first part describes the hardware structure of different FPGAs and applies circuits on different structures and describes FPGA utilization. This leads to a benefit in the area overhead and therefore in a higher reliability. This dissertation thesis also proposes a new method based on duplication and reconfiguration, this method increases reliability and availability of a safety system. The method is designed for modular systems. A classification of different secured circuits is described at the end of the work. Basic circuits are compared and reliability and availability is calculated using FTA.

All experiments are made on the railway station safety device, which is a system created from different blocks. Other measurements were applied on the standard set of MCNC benchmarks. Results are compared between each other and also with basic principles like TMR. The most relevant factors are reliability and availability.

Keywords:

SEU, FPGA, Reliability, Availability, Modular System, Railway Station Safety Device, Reconfiguration, FTA, duplication

Acknowledgements

I would like to thank to my dissertation thesis supervisor, Dr. Hana Kubátová. She has been a constant source of encouragement and insight during my research and helped me with problems and professional advancements.

Also I would like to thank to Dr. Petr Fišer for a valuable feedback and useful remarks related to my research.

Special thanks go to my friends from the Department of Digital Design, who maintained a pleasant and flexible environment for my research. I would like to express special thanks to the department management for providing most of the funding for my research.

Finally, my greatest thanks go to my family, for their infinite patience.

My research has also been partially supported by the Ministry of Education, Youth, and Sport of the Czech Republic under research program MSM 6840770014, by the Czech Science Foundation as project No. 102/09/1668, by the Grant Agency of the Czech Technical University in Prague, grant No. SGS 10/118/OHK3/1T/18, SGS 11/090/OHK3/1T/18, SGS12/094/OHK3/1T/18, GS13/101/OHK3/1T/18, SGS14/039/OHK3/1T/18.

Contents

List of Figures

List of Tables

Abbreviations

ASIC	Application-Specific Integrated Circuit
BRAM	Block RAM
CAN	Controller Area Network
CED	Concurrent Error Detection
CLB	Configurable Logic Block
CRC	Cyclic Redundancy Check
ECC	Error Correction Code
FIT	Failure in Time
FMEA	Failure Mode and Effect Analysis
FPGA	Field-Programmable Gate Array
FS	Fault Security
FSM	Finite-state Machine
FTA	Fault Tree Analysis
IC	Integrated Circuit
ICAP	Internal Configuration Access Port
IOB	Input and Output Block
LUT	Look-Up Table
MDS	Modified Duplex System
MTBF	Mean Time Between Failures
MTTF	Mean Time To Failure
MTTR	Mean Time To Repair
SIL	Safety Integrity Level
SRAM	Static Random Access Memory
ST	Self-Testing
THR	Tolerable Hazard Rate
TMR	Tripple Module Redundancy
TSC	Totally Self-Checking
VHDL	VHSIC Hardware Description Language
VLSI	Very-large-scale integration

Chapter 1

Introduction

This chapter describes my motivation from beginning of my diploma thesis to the finish of the dissertation thesis, nowadays problems of current Field Programmable Gate Arrays (FPGAs) and main goals of this thesis. It also describes basic properties of widely used FPGAs and their problems in mission critical applications in space and even on the ground level.

1.1 Motivation

Field-Programmable Gate Arrays are very popular and often used in all electronic applications. These chips have lots of advantages, for example a very short development time, high performance and very easy implementation techniques. They have also an ability of implementation of different circuits in one FPGA thanks to the reconfiguration. A function could be changed in a short time and it allows to use less resources and therefore achieve lower power consumption and lower price. FPGAs are ideal for circuits produced in small series. The price of chips implemented is low compared to a production of an Application-Specific Integrated Circuit (ASIC) by a factory, where the minimal quantity of chips is required and that quantity is very high.

FPGAs can be used in automotive industry, railway transportation, aviation or space programs etc. All these mentioned sectors influence human lives. These applications require the system which performs a required function with the highest reliability available. These systems must guarantee safety function, which means that it must not be dangerous in any case of a failure. Moreover there should be procedures how to repair the system in the shortest available time to avoid money or lives losses. This property is called maintainability.

These chips can contain a very universal system inside. They can be modified and the same board without change in the hardware can perform different functions. FPGAs are often used in some fast ethernet switches. The connection between two points is made in the hardware and reconnected during operation and not only by forwarding some data in a processor.

Thanks to the complexity and variability of FPGAs the railway station safety device was implemented in VHDL on our department [1]. The cooperation on the connection of this system to the model of the railway station on the Faculty of Transportation Science, Czech Technical University in Prague began with a diploma thesis [A.15], which was supervise by me. Trains and railways are in dreams of all young boys. A possibility of the work on improvement of this system started with my diploma thesis [A.14]. This system requires high availability and reliability of the service. Because of radiation sensitivity of FPGAs and their low reliability in irradiated areas the main challenge was discovered and accepted. It is evident that a good design, testing and improvement of the system is very important. This is the reason why my thesis will be focused on a design of a method how develop a reliable and fault tolerant system.

1.2 Problem Statement

Common and widely used FPGAs are based on SRAM memories, which are sensitive to the radiation. These FPGA based systems can not be easily used in safety and reliable systems because of their high sensitivity to the radiation effects such as Single Event Upsets (SEUs). It can cause changes of the content of embedded memory of Look-Up Tables (LUTs), interconnections and other configuration bits. These changes are not detectable by off-line test methods, because the circuit is in a good condition on start up. The function of the circuit must be checked on-line by a logic inside or outside of the chip during operation if the function must not be interrupted.

These changes in the memory, which represent faults and possible errors, are not permanent for low irradiation level. They are called soft errors. They can by easily corrected by reprogramming of the FPGA or only by reprogramming of the part with a fault. This observation could help us to design methods how to create a high dependable system, which is based on unreliable FPGAs. Everything is more complicated because of unknown or only partly known internal structure, which is a secret and know how of FPGA manufacturers. Also internal logic differs between all manufacturers and leads to different reliability techniques for different manufactures of FPGAs.

1.2.1 Mission Critical Applications

Mission critical applications are usually small series application with specification where the critical circuit must fulfill a sufficient reliability for a whole time of the mission. Applications like flight to the orbit, regulation of a nuclear power plant, some components of cars etc. There are two types of these missions. *Short time missions* like space programs require very high reliability in the beginning of lifetime of the circuit. This kind of a mission lasts about a few weeks. *Long time missions* are different in designers view, these applications have to guarantee a high reliability for a period of years. Especially the railway station safety device is a typical kind of a long time mission. These devices have operation life in tens of years. Even on ground level there is also a low level of radiation

which can cause a SEU during the long lifetime. These applications are challenging for a design in the FPGA.

1.3 Goals of the Dissertation Thesis

The main goal of my dissertation thesis is to find a way how to compare dependable designs to each other and use a realistic and practical calculation, which covers all potential risks (often hidden) of the whole design and are suitable for practical applications. Other goals are to simulate, implement and test a high reliable method with higher availability, which uses small area and resource overhead to keep the low power system with a high availability. Further evaluate technological differences between hardware structures of FPGAs and propose a suitable solution, how to achieve a high reliability and availability of the system, which utilizes these HW properties. And use all possible modern techniques available on current FPGAs.

The main goals are listed here:

1. Find a comparative method to evaluate the whole reliable design
2. Utilize HW properties to achieve higher reliability and availability
3. Design and simulate a high reliable system with modern techniques
4. Achieve a low area overhead in reliable applications
5. Design a suitable method for an easy practical implementation

1.4 Structure of the Dissertation Thesis

The dissertation thesis is organized into 7 chapters as follows:

1. *Introduction*: Describes the problem statement. My motivation to achieve the suggested goals and my these goals.
2. *Background and Related Work*: Introduces the reader into the theoretical background, basic methods and details of the problem statement. This chapter also surveys the current state-of-the-art.
3. *Technology Based Design*: This chapter describes my approach how to utilize all HW properties of an FPGA and achieve low area overhead and higher reliability.
4. *Upgraded MDS*: Deals with a new high reliable method, which is designed with respect to high dependability parameters.

5. *FTA Comparison*: In this chapter the reader can find details of a comparative method, which compares all aspects of a dependable system like reliability, area overhead, speed of fault recovery and safety level.

6. *Main Results*: All results are summarized in this chapter.

7. *Conclusions*: Describes all achieved goals with detailed explanation and proposes possible topics for future work.

Chapter 2

Background and Related Work

This chapter describes important technical background about FPGAs and problems in this type of circuits. Some important terms and definitions connected to the work are mentioned here. This chapter also describes the railway station safety device in details and possibilities.

2.1 FPGAs

Field Programmable Gate Arrays (FPGA) are integrated circuits, which could be programmed to perform a special function. Development of FPGAs started in the 80s of the 20th century [2]. These semiconductors are based on a regular matrix of logic blocks connected together by programmable interconnections. Main difference between FPGAs and Application Specific Integrated Circuits (ASICs) is the ability of reprogramming. ASICs can perform only a special designed task, which is given by the manufacturer and FPGAs could be programmed by the designer in his office [3]. FPGAs could be programmed once (not often used) or many times, which are used in most cases because their simplicity during development. Most of these FPGAs are based on SRAM memories. These chips could be reprogrammed also during operational time. It offers low cost for low to medium volume production and faster time to market.

FPGAs are composed from regular structure of Configurable Logic Blocks (CLBs), which contains Look up tables (LUTs), dedicated blocks like DSP, Block RAMs etc. and input and output blocks (IOBs). All these parts are connected via routing switches. You can see the block diagram in Figure 2.1, which is from [4].

2.1.1 Problems in FPGAs

As written above, most of FPGAs are based on SRAM memories, which are sensitive on radiation [5]. This radiation is not only in space or high altitudes, but also on ground level [6], where the intensity is low, but not zero. Radiation could be generated even by the package of the integrated circuit (IC). Alpha or neutron particles can cause a change

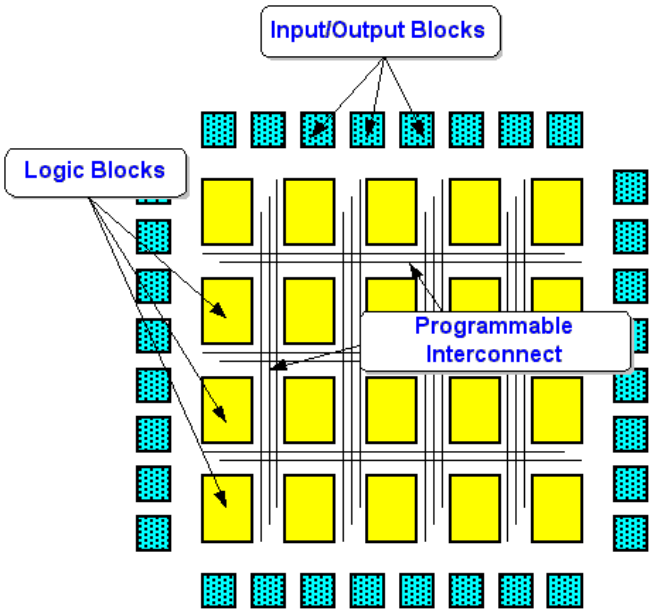

Figure 2.1: FPGA internal structure

of state in flip-flops or memory cells. This change is called a Single Event Upset (SEU) [7] and belongs into soft error, because the change will not destroy the cell. The problem is more serious because the change in configuration memory of an FPGA can cause an unintended change in functionality. The change will remain until it will be detected or corrected. The change in interconnection and function is in the Figure 2.2 from [8].

Systems with SRAM FPGAs should incorporate an error mitigation technique, which could be on system, software or hardware level. This is much more critical in high reliable or safety application such as aerospace, automotive, medical or military applications. The system malfunction could cause loses of lives or finances. Some SRAM reliability and simulation were presented in [9].

2.2 Dependability

Covers all functional properties of the system and defines their measures like RAMS.

2.2.1 RAMS

Reliability The ability of a component or a system to operate under designated set of operating conditions over a given period of time.

Availability The probability that the system will be functioning correctly at a given time.

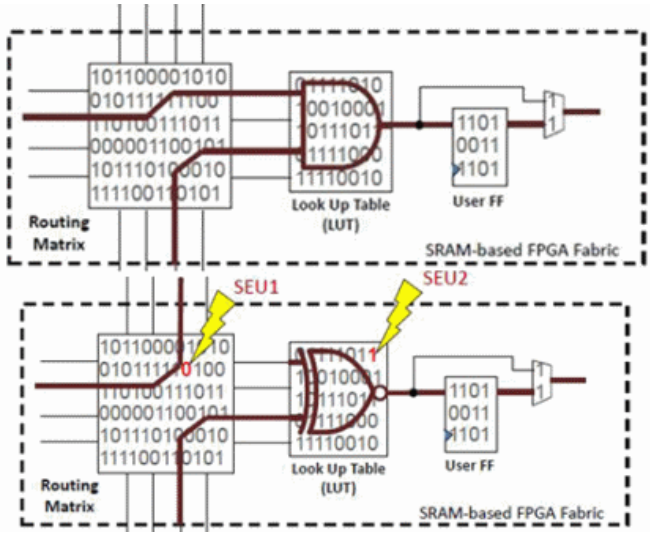

Figure 2.2: SEU in an FPGA

Maintainability The ability of a system to be maintained. Maintenance is the action taken to retain a system in, or return a system to, its designed operating condition.

Safety The property of the system that will not cause injure human or damage to the environment.

2.2.2 Faults in Time

Faults can be also classified by a time duration.

Permanent fault is a malfunction of a component to the reparation.
– For example burn of a chip

Transient fault is not permanent and the functionality is restored after some time.
– Wrong soldering for examples

Intermittent fault is a periodical malfunction of a component.
– Like overheating and cool down of a part

SEUs are not easy to classify, because the fault is corrected afters restart of the system, but during non stop operation it will not be repair itself.

2.2.3 Faults Measuring

2.2.3.1 Reliability

The average time to get the failure is *Mean Time to Failure* (MTTF). The frequency of faults can be measured by the index *Mean Time Between Failures* (MTBF).

2.2.3.2 Maintainability

The time to repair a component is called *Maen Time to Repaire* (MTTR).

2.2.3.3 Safety

Safety is achieved when a system meets applicable standards and the design process is according to these standards like EN 50 126[10].

2.2.3.4 Availability

It (A) measures the percentage of time of a device in the operating state and the lifetime. It could be calculated for long time availability applications by MTTF and MTBF according to equation 2.1

$$A = \frac{MTTF}{MTTF + MTBF} \tag{2.1}$$

2.2.4 Definitions of Terms

Fault A fault is present in the system when physical difference is observed between the good or correct system and the actual system.

Error Error is a difference between the correct output and detected output.

System Failure A system failure occurs when the system fails to perform its required function.

Hazard A hazard is a situation with a potential to people injury or damage to the environment.

Fault detection It is a process, when the system detects a fault itself.

Fault location It is a process for localization of the position of the fault.

Fault recovery It is a process of repair of the fault, which repairs the system.

2.2.5 Fault Classes

The Concurrent Error Detection (CED) techniques allow a faster detection of the soft errors. There are three basic quantitative criteria in a field of CED [11]:

- Fault Security (**FS**).
- Self-Testing (**ST**).
- Totally Self-Checking (**TSC**).

These three aspects have to be used in the on-line testing field to evaluate the level of safety of the designed or modeled system.

To get more the precise evaluation and computation of all dependability parameters according to the strict reliability requirements [10] all faults should be classified and separate into four classes, A, B, C and D [12] according their impact on a tested circuit in the FPGA.

- **Class A** – Hidden faults. These are faults that do not affect the circuit output for any allowed input vector. Faults in this class have no impact to the FS property, but if this fault can occur, a circuit cannot be ST.
- **Class B** – Detectable fault. These are faults detectable by at least one input vector. They do not produce an incorrect code word (valid code word, but incorrect one) for other input vectors. These faults have no negative impact to the FS and ST properties.
- **Class C** – Undetectable faults. These are faults that cause an incorrect code word for at least one input vector. They are not detectable by any other input vector. Faults from this class cause undetectable errors. If any fault in a circuit belongs to this class, the circuit is neither FS, nor ST.
- **Class D** – Partial faults. These are faults that cause an undetectable error for at least one vector and a detectable error for at least one another vector. Although these faults are detectable, they do not satisfy the FS property and so they are also undesirable.

2.3 Railway Station Safety Device

Even at these days the railway safety devices are in many cases constructed from relays. Relays based devices are very popular because of their good dependability parameters and easy calculation of dependability. Relay blocks correspond with structure of railway and it is easy to use them.

The FPGA implementation was developed in our department [1]. The simple railway station is shown in Figure 2.3, upper and lower parts represent the same station, only the lower part is composed of five individual blocks. These blocks are very similar in internal structure and this device is a nice example of modular system. The proposed device uses the same structure of blocks like relay schema, but it is completely rebuilt. The communication and function of these blocks is different, but function of the whole circuit is the same like in previous case. Some relay blocks were put together into one new. Railway station safety device is based on five basic blocks. Each of these blocks consist a FSM. Figure 2.3 shows a simple safety device with two rails and two rail switches. This device is an example of long time mission.

Old big relays have a lot of disadvantages. The main one is a size of circuit. Relay blocks are so big, that it takes one room for small railway station. Bigger safety device can

take even whole small building. This innovation of nowadays safety device is not so easy because the predefined dependability (safety and reliability) properties should fulfill strict railways norms [10], [13], [14]. Therefore the dependability model has to be constructed according the real failure rate, so the fault classes have to be find out and real dependability parameters have to be compute.

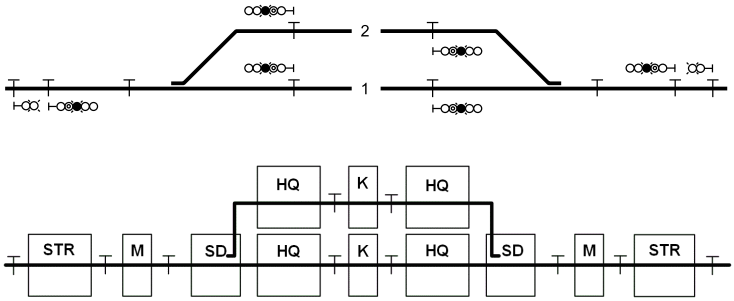

Figure 2.3: Simple railway station with FSM based blocks

2.3.1 Function of blocks:

- **STR block** represents a starting signal. This block is placed on the entrance of railway station and is control a semaphore, which shows permission for trains to go into railway station. This block can be a start point of a train path. This train path is built directly from one semaphore to the other. This block can be also the end of a train path. STR block solves a track before the home signal and signalizes whether this track is free or occupied.

- **M block** monitors a position of a train and is connected to two blocks on its sides. When the train path is through this block, it must get a signal of position from one neighboring block, than the train must occupy this block and then the neighboring block on the other side. The train must not change its direction and must not skip over one block. In any other case, an error is signalized.

- **SD block** looks like a rail switch and it monitors position of switch and also the right position of a train. This block can generate signal for semaphore on other block in the case, when more rail switches are connected together.

- **HQ block** controls a semaphore, which signalize permission to leave railway station. This block is typically used like an end point of a train path. In case, when a train leaves the station, this block is used as a start of the train path.

- **K block** monitors a position of train. In contrast to block M, the train can stop here and change its direction and go back.

All of these blocks are connected to some railway sensors, lights and switches like in Figure 2.4. These signals are often bidirectional.

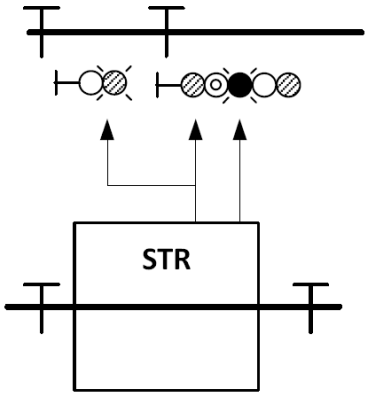

Figure 2.4: Connection of STR block to the railway

2.3.2 My Diploma thesis

My diploma thesis[A.14] was based on the VHDL railway station safety device composed from blocks [1]. These block should correspond to the structure the railway, therefore I designed a VHDL generator. It can create an interconnection between these blocks in VHDL. The structure of the railway is defined in a XML file, which was generated by a Java application. It was developed as a part of the thesis. The whole flow of the design is as follows: Create a XLM file in Java generator according to real appearance of the railway station, generate a VHDL top module by the generator from the XML file, use VHDL file in project with all railway blocks. Thanks to this procedure the user is possible to create any railway station safety device.

2.3.3 Railway model connection

I supervised a following diploma thesis [A.15], which was focused on connection between FPGA board and model of the railway station. In figure 2.5 there is a part of the railway station model created on the Faculty of Transportation Science, where the connection to the safety device is made by a CAN bus. The thesis designed a converter to it. Also the FPGA board has a VGA output and displays statuses of blocks on the monitor and allows some modifications by PS/2 keyboard.

Figure 2.5: Model of the railway station

2.3.4 Railway station graphic generator

Another work connected to this railway station safety device was a graphical editor of the proposed XML file from my diploma thesis. This work was supervised by me and it extended the Java generator of the XML file. Thanks to this bachelor thesis [A.16] it is very easy to create any railway station safety device. It speeds up any new design.

2.4 Norms

Railway application standards are applicable for safety electronics systems. All new railway systems must by created according to these norms.

- **EN 50 126** Railway applications - The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS) [10]
  – is about RAMS, definitions of hazards and risk and the process flow

- **EN 50 128** Railway applications - Communications, signalling and processing systems - Software for railway control and protection systems [13]
  – this norm is about software, roles and their relationship and V model

- **EN 50 129** Railway applications - Communication, signalling and processing systems - Safety-related electronic systems for signalling [14]
  – is more about hardware and Tolerable Hazard Rate (THR) and SIL metrics

2.5 Related Work

The related work is described in this section and specific topics will be described in specific chapters. There are some basic methods, how to increase reliability or availability of a system.

2.5.1 Basic Hardware Redundancy

The basic method how o increase reliability is to increase the number of devices. Overall the probability of fault is higher because of higher number of devices, but the availability of the system is higher, because some devices are still in operational mode. In some cases for correct function more than one device must be operable. In this section the basic principles will be mentioned with their related work and advantages and disadvantages.

A fault tolerant method could be also made on a low level by a special architecture of Configurable Logic Blocks [15].

2.5.1.1 Totally Self-Checking Circuit

The TSC circuit is composed of small blocks, where each block satisfies the TSC property. The structure of the compound design satisfying the TSC property is shown in Figure 2.6. Six places where an error is observable for this compound design has assumed. It is assumed, for simplicity, that an error occurring in the check bit generator will be observable at the parity nets (number 1), and an error occurring in the original circuit will be observable at the primary outputs (number 5). The checker in block N will detect an error if it occurs in the net number 1, 2, 4 or 5. If an error occurs in the net number 3 or 6, it will be detected in the next checker (N+1). The method used to satisfy the TSC property for the compound design is described in greater detail in [16]. Not every small block (in the compound design) satisfies the TSC property to 100%. The TSC property depends on the FS and ST properties, which are also not satisfied to 100%. For availability computations, we find the block with the lowest FS property value in the compound design.

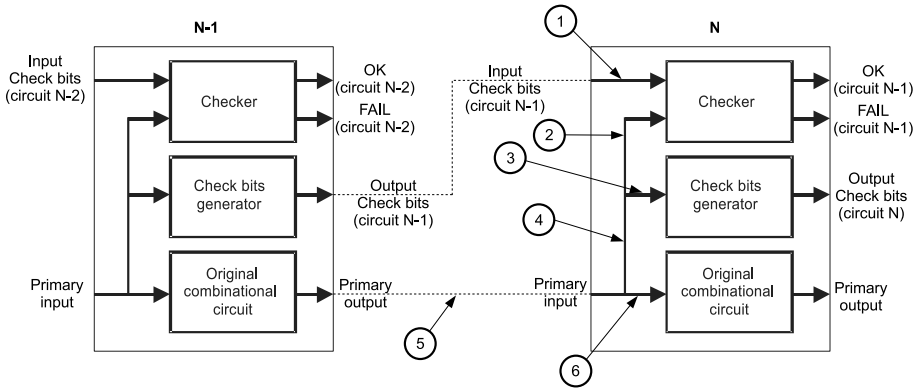

Figure 2.6: The structure of compound system corresponding the TSC property

These three aspects have to be used in the on-line testing field to evaluate the level of the safety of the designed or modeled system. FS and ST parameters are calculated according to equations 2.2 and 2.3.

$$FS = \frac{B}{A + B + C + D} \cdot 100\ [\%] \tag{2.2}$$

$$ST = \frac{B + D}{A + B + C + D} \cdot 100\ [\%] \tag{2.3}$$

Calculations of dependability parameters according to mentioned classes[12]. Testing by test vectors of any circuits returns numbers of errors according to mentioned classes.

2.5.1.2 Duplication

Duplication means system, where two parts perform similar function. The result of these two parts should be same, but the internal realization could be different. This difference could be use because of minimize of design errors. There are also two independent inputs, two comparators and two outputs. All parts must be doubled if not it becomes a single point of failure. The duplication with comparison is in Figure 2.7, where two FPGAs with original function and comparator are presented.

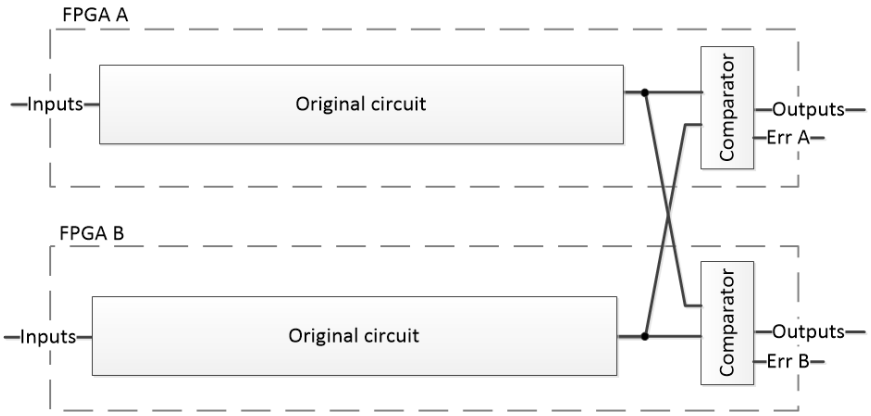

Figure 2.7: Basic duplication system

Duplication has only one big disadvantage, if there is a fault in one FPGA, the comparator can see only different signals and can not decide which FPGA is faulty.

2.5.1.3 TMR

Tripple Module Redundancy (TMR) is a circuit with three similar circuits, which perform the same function. It could be in one FPGA or in three FPGAs. It depends on current implementetion This triplication has an advantage, in comparison with duplication, in the fact that during a fault of one FPGA the voter knows where the error is. The voter can

ignore the error and the system can still operate in dual channel mode. The voter is a circuit, which decides according to majority, which FPGA does not operate properly. The voter is much more complicated in comparison with a comparator.

There are two different implementation of TMR because connections of voters take a lot of wires and input/output pins. The easier way is to use only one voter and connect all FPGAs into this single device. This could lead to a single point of failure and when a fault is in the voter, the system is down. This model is in Figure 2.8. Application of one voter is in [17].

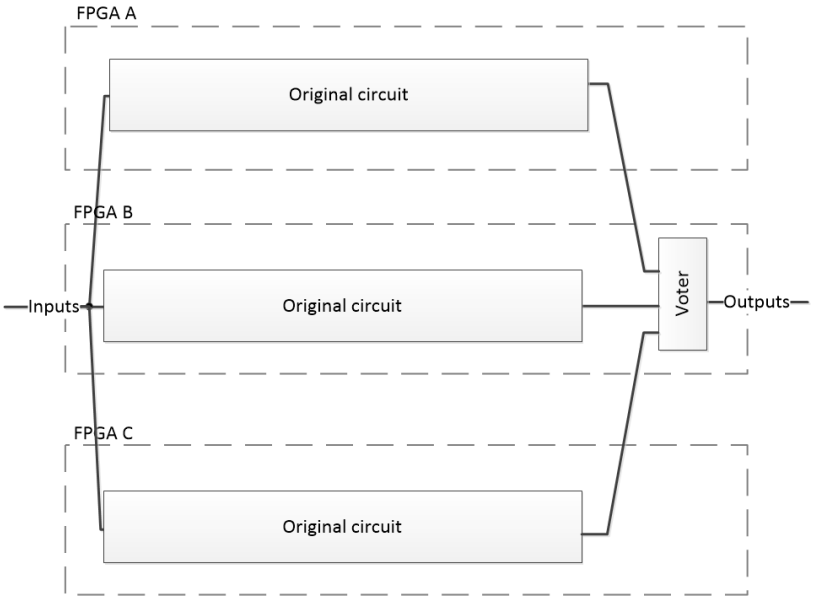

Figure 2.8: Basic schema of TMR with one voter

Other type of implementation uses three voters. It guarantees a reliable solution because if the error is in one voter, the rest of voters can operate and no error is propagated. The model with 3 voter is in Figure 2.9. Different types of voters and their application is described in [18].

TMR could be done on only a part of the system. A selective TMR was presented here [19]. Also a TMR combined with partial reconfiguration to be fault tolerant is in [20].

It is not necessary to create a whole design according to the TMR technique, for some application only a small part could be triplicated [21]. It can have soma advantages in low area and shorter time to repair.

2.5.2 Modified Duplex System

Modified Duplex System (MDS) architecture [22] uses two instances (instead of mostly used TMR architecture like e.g. [23]) of design that may be not fault tolerant. The purpose of MDS architecture is to achieve the whole circuit including all checkers and comparators to be fault tolerant. The MDS block diagram is shown in Figure 2.10.

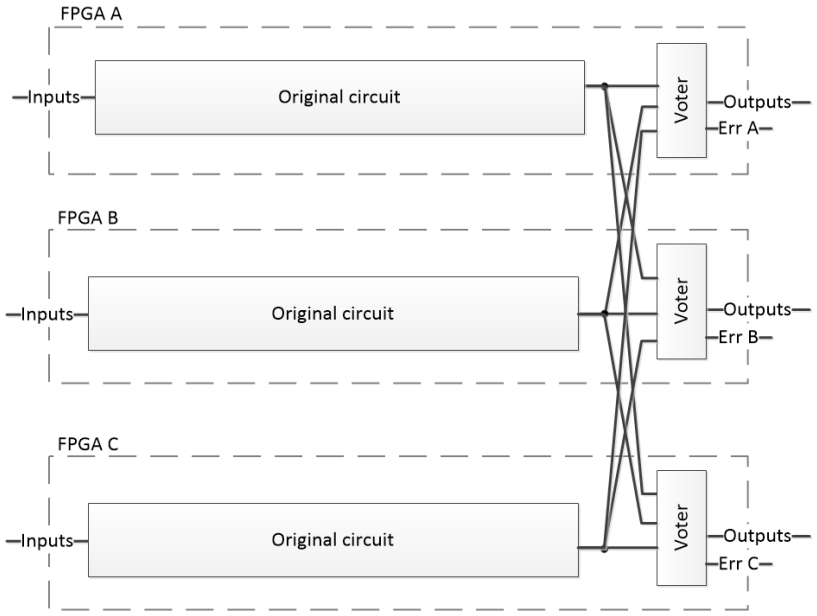

Figure 2.9: Basic schema of TMR with three voters

If an error (caused by a SEU) is not detected inside the system by some TSC block, it is detected by comparators. The error detected by comparators triggers initiate the reconfiguration of both blocks (outputs from blocks are different, but the source of the error cannot be determined). But this full reconfiguration is a time demanding process and can cause synchronization problems and therefore leads to decrease of the whole system availability.

2.5.3 SEU simulation

There are some different techniques, how to simulate a SEUs in the design. Real ionization by some radiation is possible, but has same disadvantages like low probability of the SEU in a specific place [24]. Some approaches could be based on a combination and simulation of injection of a fault[25]. Or only a SW injection into the bitstream could by used for verification of applied methods [26].

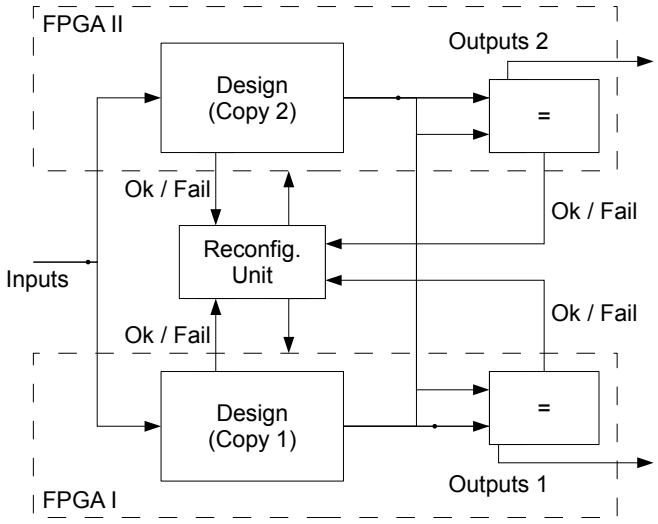

Figure 2.10: The block scheme of Modified Duplex System

Chapter 3

Technology based design

This chapter describes details of technological processes and structures of FPGAs. It deals about advantages and disadvantages of Spartan 3E, which is manufactured in 90 nm technological process and uses LUT-4 and Virtex V, which is made by 65 nm process with LUT-6. These knowledges are compared on railway station safety device and also on standard benchmarks.

Second part describes the decomposition of counter according to technological process to increase reliability of the system. All simulations are mainly focused on counters used in the railway station safety device.

The main idea in these chapter is to fit the circuit on the technology structure or vice versa.

3.1 Internal Structure

3.1.1 Block decomposition

The railway station safety device has a regular structure and sequences of blocks are similar. Each of five basic blocks consists an FSM and some of them a combinational logic and a counter. These parts of each block were divided into separate VHDL files due to usage of a different technique to increase dependability. Table 3.1 shows the content of blocks.

	STR	**M**	**SD**	**HQ**	**K**
FSM	present	present	present	present	present
logic	present		present	present	
counter	present			present	

Table 3.1: Railway station safety device blocks content

You can see in Figure 3.1 the original and new block, in this case it is e.g. STR block and HQ block. Other blocks have few parts, the number of parts you can see in Table 3.1.

Designer can use for every block or every part different method to increase dependability. Designer should find out the best combination of methods to get the highest reliability. In the other way it is possible to find out the lowest area overhead for predefined dependability.

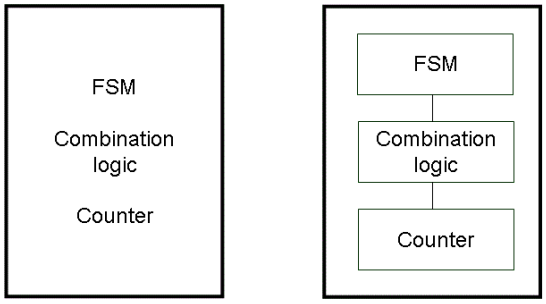

Figure 3.1: Block division into three independent parts

3.1.2 FPGA Structure

FPGAs have the regular structure of lookup tables (LUTs), which can perform any logic function. These basic blocks have typically 4 or 6 inputs and one output. In Table 3.2, you can see my research of commercial devices and its type of basic blocks. We want to also compare LUT-4 and LUT-6 structures, because of different numbers of stuck-at faults and decide which structure is better for concrete type of counter.

3.1.3 Technology processes

FPGAs are fabricated by different CMOS process. In Table 3.2 column 2, you can see the name of used technology based on [27], [28], [29], [30], [31], [32], [33]. SRAM cells are sensitive to radiation, especially of high energy neutrons. A memory cell, which was made in smaller dimensions, needs less energy to change its state. The cell is smaller and takes less area of the chip. This reduces the probability of a collision with a particle.

In the Rosetta [34] experiment Xilinx measured λ for 90 nm and 65 nm technology. Process of 90 nm has a probability of a one fault in 1 Mbit of configuration memory $2.46 * 10^{-7} h^{-1}$. The 65 nm process is much better and its probability is only $1.51 * 10^{-7} h^{-1}$. This leads to higher reliability of designs based on 65 nm process, moreover the final circuit should uses more resources.

The Stuck-at fault model describes an input or an output value change caused by a short. The typical value for the Stuck-at fault model is 0 for a low voltage level and 1 for a high voltage level. In a Stuck-at fault model, the fault can manifest either at primary inputs or at a primary output as the Stuck-at 0 or the Stuck-at 1.

Table 3.2: Technology process of FPGAs

Product	Technology process	Basic blocks
Xilinx Spartan 3	90 nm	LUT-4
Xilinx Spartan 6	45 nm	LUT-6
Xilinx Virtex 4	90 nm	LUT-4
Xilinx Virtex 5	65 nm	LUT-6
Xilinx Virtex 6	40 nm	LUT-6
Altera Stratix	130 nm	LUT-4
Altera Stratix ii	90 nm	LUT-6

You can see in Figure 3.4, that LUT-4 has 4 places of possible faults on inputs and 1 on the output, totally 5. There is LUT-6, on the right side, which has only 7 places of possible faults.

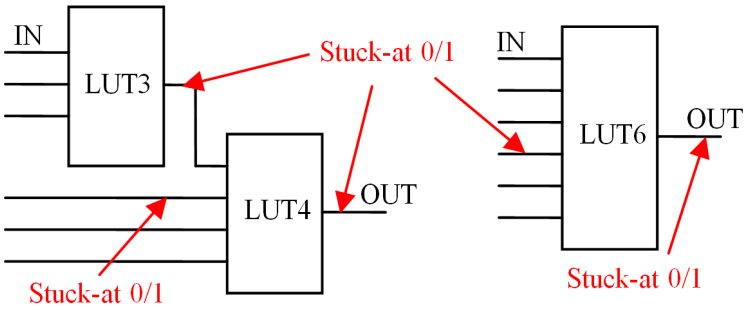

Figure 3.2: LUT-4 and LUT-6 counters

Both circuits can perform the same function. Left circuit made of LUT-4 and LUT-3 has together 9 places of possible faults. The structure of FPGA has only LUT-4 and LUT-3 is made by deactivating of one input, but there can be another fault, which can cause an error. Than the circuit has totally 10 places of possible faults. The circuit on the right side of Figure 3.4 has only 7 places of possible faults and no other.

Simulations in Xilinx ISE shows that logic functions up to 36 inputs, which has number of inputs divided by 6 without reminder, are made of n + 1 LUT-6. The maximum of possible faults is $7 * n + 7$. Functions made of LUT-4 have in these cases worst results. The number of possible fault is specific for each number of inputs and is higher then for LUT-6. Another analysis in Synplicity Synplify shows that the counter with width of 6 bits uses 6 times LUT-6, each for 1 bit output. When we use LUT-4 technology, the synthesis tool makes the counter by using 5 times LUT-4, 4 times LUT-3 and one LUT-2. The LUT-6 version has $42(6 * 7)$ possible faults and LUT-4 has $39(5 * 5 + 4 * 3 + 3)$, but in calculation with only LUT-4, the inactive inputs could be in fault state and then the number of possibly faults will increase to $50(10 * 5)$.

Table 3.3: Stuck at faults for railway station blocks

	Spartan 3E		Virtex V			
Benchmark	**LUT-4**	**Faults**	**LUT-6**	**Faults**	**Difference**	**Faults**
STR_automaton	90	450	62	434	-16	-3.56%
STR_counter	78	390	73	511	121	31.03%
STR_logic	6	30	6	42	12	40.00%
STR_total		**870**		**987**		13.45%
HQ_automaton	94	470	67	469	-1	-0.21%
HQ_counter	78	390	73	511	121	31.03%
HQ_logic	6	30	6	42	12	40.00%
HQ_total		**890**		**1022**		14.83%
K_automaton	83	**415**	56	**392**	-23	-5.54%
M_automaton	88	**440**	60	**420**	-20	-4.55%
SD_automaton	196	980	252	1764	784	80.00%
SD_logic	25	125	15	105	-20	-16.00%
SD_total		**1105**		**1869**		69.14%

3.1.4 Stuck at fault in railway interlocking system

All railway station blocks were synthesized in Xilinx ISE 13.1 twice. First time for Spartan 3E, which has 90nm technology process and LUT-4 and than for Virtex V, which uses LUT-6 and 65 nm process. Each block is composed from different types of LUTs. It differs only in number of inputs, but the hardware structure is same for all types in one device. All these LUTs were summarized and multiplied by 5 for Spartan 3E and multiplied by 7 for Virtex V. It gives the number of possible faults. Than the difference and percentage of it ware calculated. All results are mentioned in Table 3.3.

In average the number of faults is bigger by 19.22%, which gives the advantage to LUT-4 in Spartan 3E. But some blocks are much better in LUT-6, for example SD_logic is 16% better ank K an M blocks are about 5%. In opposite side there are block like SD_automaton and STR_logic, which are not suitable for LUT-6 FPGA.

3.1.5 Comparison of 2 railway stations

From basic blocks of the railway station, it is possible to create any different railway station. In this example the real number of faults will be calculated for whole system. The basic railway station with 2 rails from Figure 2.3 will be compared to a terminal railway station with 4 terminal rails. In this setup there are more switches of rails. Block schematic is in Firure 3.3.

In Table 3.4 you can compare number of blocks and than the number of possible faults in each system. For these systems the number of faults is higher for Virtex V with LUT-6. For the basic 2 rails version it increases in about 23,9%, in the other railway station, the situation is similar and number of faults increased in about 24,4%. From these results, you

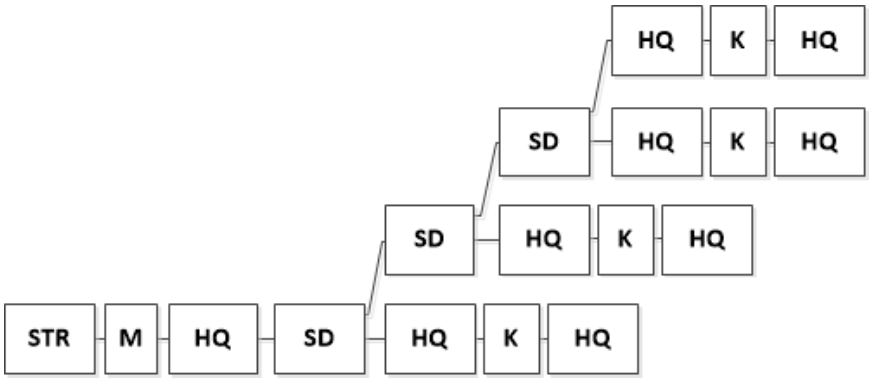

Figure 3.3: Railway station with 4 terminal rails

Table 3.4: Comparison of railway stations

	Basic railway			4 terminals rail		
	Qty	**Spartan 3E**	**Virtex V**	**Qty**	**Spartan 3E**	**Virtex V**
STR	2	1740	1974	1	870	987
HQ	4	3560	4080	9	8010	9198
K	2	830	784	4	1660	1568
M	2	880	840	1	440	420
SD	2	2210	3738	3	3315	5607
		9220	**11424**		**14295**	**17780**
		100%	**123.9%**		**100%**	**124.4%**

can see that both of these two railway stations are not very suitable for LUT-6 (Virtex V), but it has lower resource utilization in Spartan 3E, which is based on LUT-4.

3.1.6 Benchmarks

The same process was applied on standard MCNC benchmarks [35] as on blocks of the railway station. All benchmarks were also synthesized in Xilinx ISE, numbers of LUTs were summarized and multiplied. Results are compared in Table 3.5.

From differences in column 6 or percentage of faults, these benchmarks are more suitable for LUT-6 structure, in average the number of faults was decreased by 13,62%.

3.1.7 Results

We can compare 2 railway station safety devices and some benchmarks using stuck-at fault model across technology process of the FPGA. Easy functions like STL_logic or newapla benchmark are not suitable for LUT-6 FPGA, because the reduction of LUTs is not high enough and in general the number of fault increases. On the other side some complex

Table 3.5: Stuck at faults for standard benchmarks

	Spartan 3E		Virtex V			
Benchmark	**LUT-4**	**Faults**	**LUT-6**	**Faults**	**Difference**	**Faults**
apla	95	475	40	280	195	-41.05%
br1	52	260	32	224	- 36	-13.85%
br2	37	185	25	174	-10	-5.41%
dk17	40	200	27	189	-11	-5.50%
dk27	21	105	14	98	-7	-6.67%
dk48	49	245	37	259	14	5.71%
ex1010	848	4240	487	3409	-831	-19.60%
f51m	19	95	10	70	-25	-26.32%
gary	168	840	115	805	-35	-4.17%
mp2d	31	155	19	133	-22	-14.19%
newapla	16	80	13	91	11	13.75%
newcpla1	37	185	23	161	-24	-12.97%
newcpla2	24	120	11	77	-43	-35.83%
p82	25	125	14	98	-27	-21.60%
sex	19	95	14	98	3	3.16%
sqr6	21	105	10	70	-35	-33.33%

functions like apla, newapla2 or f51m are very suitable for LUT-6, where the change of technology can reduce resources and increase reliability and also reduces the price of the hardware. It is not significant that a small function is not suitable and big is suitable. It really depends on the specific function, which could be synthesized more successfully on some technology process. The evaluation which FPGA should be used in the system must be done for all systems. The advantage is it can increase reliability and safe money.

In the other hand, there is a fact that 90 nm technology process is more sensitive to SEUs according to Rosetta experiment [34]. But the internal structure is not known in details and is not possible to compare designs without knowledge of how many bits from configuration memory is used. Also it is not possible to calculate that one LUT uses same resources in both FPGA types.

3.2 Decomposition of counters

The main topic of this experiment is to decide if it is more reliable to use one counter or two with half width. This work started on counters, because it is a part of the railway station safety device. Results of this experiment are designed for SRAM based FPGAs, which are commonly used for rapid prototyping and sometime for small series of products.

Counters were converted from sequential logic into combinational logic. Inputs of the logic represent the internal value of counter, output value is the incremented input by one. It helps for a better simulation in stuck-at fault model. We can use D flip-flops to convert

combinational logic back to sequential and connect outputs to inputs through them. The value will be incremented by every clock edge.

3.2.1 Process flow

Simulation process starts at subscribing combinational logic in PLA format. Then there is created a predictor in PLA format to generate a parity bit. This is the simple security method for all of the combinational logic. I generated test vectors to check parity, it is saved in .tst file. These two PLA files are minimized by ESPRESSO[36] and then converted by BOOM[37] to VHDL. Then I performed independent synthesis in Synplicity Synplify for both files.

Then a predictor was created in the mentioned PLA format to generate a parity bit. This is the simplest security method for all of any combinational logic. I have generated test vectors to check the parity. It was saved in .tst file. These two PLA files were minimized by ESPRESSO and then converted by BOOM into VHDL file. Then I performed an independent synthesis in Synplicity Synplify for both files. There ware created EDIF files. The original and the predictor were joined together. After joining, the design was tested by test vectors files and statistic for each fault class was performed. The outputs from this method were statistic numbers of A, B, C and D classes faults. These numbers are important for calculation of FS property of the design. This is the basic CED model, which is shown in Figure 2.6.

Calculation of reliability for these two independent counters in serial connection is well known. The probability of counter A in good condition is $P(A)$ and B is $P(B)$. Typical probability of good working serial connection is in equation 3.1. $P(\neg A)$ is probability of fault on counter A, this calculation is in equation 3.2.

$$P(A \cap B) = P(A) \cdot P(B) \tag{3.1}$$

$$P(\neg A) = 1 - P(A)) \tag{3.2}$$

But SEU changes this calculation, because fault on both devices is not possible. There could be only one change in memory and it is very important. In Figure 3.4 you can see diagram of probability sets.

We have to calculate it by different way, because $A \cap B$ is empty set. Only the following combinations are possible: only A works, only B works, A and B work together. This is the main thought of calculations. We have to calculate probability by using three mentioned sets. The equation 3.3 represents the right calculation probability of fault in circuit affected by SEU.

$$P(A \cap B) = \frac{P(A) \cdot P(B)}{P(A) + P(B) - P(A) \cdot P(B)} \tag{3.3}$$

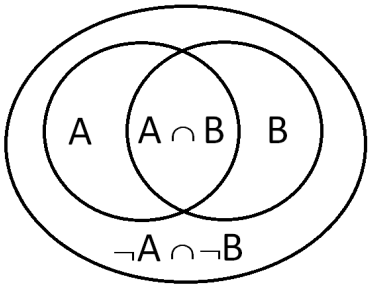

Figure 3.4: Probability of the serial connection

3.2.2 Simulation process

The simulation process took a lot of time. The size of PLA files is big and it grows exponentially. A counter of width 16 bits has about 2 MB, but 24 bits width counter has even 830 MB. Creation of VHDL files and their synthesis process was slow and sometimes was not finished. The first thought, about dividing 32 bits and more counters, was not possible and maximum possible counter has width 16 bits.

I have tried to maximize fault security parameter to 100%. It is calculated using equation 2.2, all of parameters in equation are numbers of fault in class. In Table 3.6 you can see measured data about original counters. The increasing width of bits increase number of possible stuck at faults. The number of undetected faults is rising and we can see the descending tendency of FS property. In this case, FS property for divided counter into 2 with half width is lower than FS property for one big counter. I expected it for small counters, but my technique of measuring was not able to work with wider counters. For 16 bits counter it is better to use two 8 bits counters and I think it will have rising tendency and it will be better to divide counters wider than 16 bits. This is my current target. Table 3.6 shows in last column that the number of possible faults in two counters is lower than in one. It is little bit strange, because FS property is lower.

The same process will be tested with Gray code counters. This easy code has two advantages. The first is that it changes only one bit in code word at time and the second is a parity generator. Change of one bit means that it regularly changes count of the ones and parity is regularly changed between 0 and 1. The parity generator is designed only from XOR operators.

You can see in Table 3.7 that FS property is at 100% up to 4 bits width counter. These small counters are totally self checking (TSC). Gray code has for almost all counters better results than original counters, this is right way of improvement. Design of 6 and 8 bits counters will be better with 2 small counters in series, because FS property will be at 100% like at small counters. Only 16 bits counter has very small FS property and there it will be definitely better to use two or more counters in series.

To improve our results, counters in Gray code will be better.

Table 3.6: Original counters

Width [bits]	FS [%]	Number of possible faults	FS of 2 counters [%]	Number of faults in 2 counters together
2	92.86	28	-	-
3	91.3	46	-	-
4	90.63	64	86.67	56
5	84.44	90	-	-
6	85.25	122	83.99	92
8	83.52	182	82.87	128
10	83.19	226	73.07	180
12	81.57	434	74.29	244
16	68.67	498	71.7	364

Table 3.7: Gray code counters

Width [bits]	FS [%]	Number of possible faults	FS of 2 counters [%]	Number of faults in 2 counters together
2	100	24	-	-
3	100	50	-	-
4	100	74	100	48
5	94.44	100	-	-
6	94.2	138	100	100
8	92.08	202	100	148
10	89.05	274	89.47	200
12	90.05	382	89.04	276
16	31.33	498	85.32	404

3.2.3 Results

Experiments described here show advantages and disadvantages of decomposition of counters. My expectations about the results were right. It looks like that there is a limit of the counter width and behind this limit it will be better to divide the counter into 2 smaller ones. The experiment with Gray code counters was very helpful. I get 100% coverage of faults and I have a way to get all counters safe. These experiments are very important for innovation of the railway station safety device. However it is useful for every circuit based on SRAM FPGA and contains counter.

3.3 Chapter summary

In this chapter there were described two different approaches, which are focused to increase reliability using the technology as its benefit and internal structure of the FPGA. First parts was dedicated to LUT-4 and LUT-6 topic. There was decided how to select a suitable FPGA for a final design. Two different structures of the railway station safety device were compared and utilization of the FPGA was calculated. Also standard MCNC benchmarks were compared in same way. From the results some designs are more suitable for LUT-4 and some for LUT-6.

In the second part there are compared counters, which are decomposed into two smaller counters. Also counter designed as a Gray code counters are compared and divided into two. Better results are for smaller counters in Gray code.

As a result it is better to use small counters with Gray code and decide which technology is better for the whole design after the whole circuit is designed, sometimes it has a dramatic influence on the number of used resources for example in the presented railway station safety device you can see the benefit in about 24%.

Chapter 4

Upgraded MDS

This chapter describes a new proposed method to increase reliability and availability of a safety system. The method is described in details with whole reparation flow. The system is compared with TMR and MDS. Reliability and availability markov models are attached and calculated.

4.1 Used principles

This method is based on Modified Duplex System and uses techniques like partial dynamic reconfiguration and combine it with Concurrent Error Detection techniques. The man goal of this method is to increase reliability and availability with minimal area overhead and minimal I/Os required. This method is capable to secure any modular circuit.

4.1.1 Modified Duplex System

Description of MDS is in Figure 4.1 in section 2.5.2. The main principle is to use 2 FPGAs and implement the same function in both. The main function is extended by a final comparator, which can signalize a difference in the output. It triggers an external unit to perform a complete restart of both channels with loading of the new failure free configuration.

In Figure 4.1 you can see the MDS with railway station safety design inside. This model will be upgraded in next chapters.

4.1.2 Xilinx Macro

Xilinx also designed its own SEU controler macro[38], which can read and calculate CRC of all parts of a FPGA and if the CRC does not match, it performs a reconfiguration to this part. It also allow to insert an artificial SEU into a running FPGA. This macro uses 174 Slices and one BRAM, which takes 585 configuration bits. Together it is 20 549,2 bits

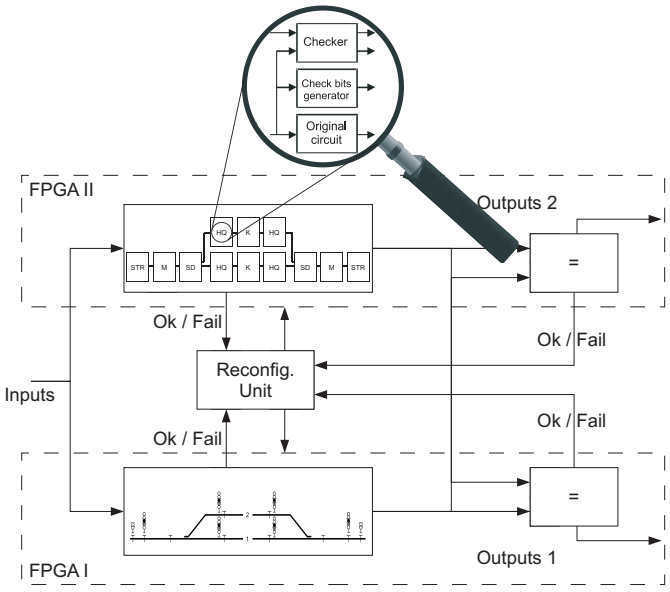

Figure 4.1: The block scheme of Modified Duplex System

which corresponds to FIT number of 33,9. This macro is not used in this system, but the approach is very similar and uses similar principles.

The reconfiguration is made through Internal Configuration Access Port (ICAP), because it is the fastest bus dedicated to it. It has a maximum speed of 3200 Mbit per second.

4.1.3 Reconfiguration

It is a process when the bitstream is loaded into the FPGA and the function is modified according to the new configuration. Loading of a whole bitstream into the FPGA will rewrite all old configurations even with a SUE. Modern FPGAs are able to perform a Partial reconfiguration, which allows to reprogram only a part of the FPGA, while the rest is in operational mode. This technique could be used for two reasons. One is to change the function during operation or reprograme the same function when a SEU is detected.

Reconfigurable partition (RP) is a dedicated part of an FPGA, where different reconfigurable modules (RM) could be loaded. Reconfigurable modules could be stored in an external memory and loaded in time when the specific function is needed. In this case the RM is only one for one RP, because I will repair only a SEU and make no change to the original function. All details about partial reconfiguration are mentioned in Xilinx documentation [39].

The static reconfiguration unit is an external device, which is made of any radiation tolerant parts, which has reliability is better than $\lambda = 10^{-10}$. It could be made easily from passive components or an easy micro controller.

4.2 Basic Scheme

The system was developed during the evolution of the railway station safety system in our department [40]. This system is modular and based on different types of blocks. This method reduces recovery time, because it uses partial reconfiguration more often than the whole FPGA reconfiguration, which is used only in critical situations.

Availability of the whole system increases thanks to a short time of partial reconfiguration, which rewrites only a part of the FPGA. System designed in this way uses less area overhead compared to other methods like TMR or N-module redundancy (NMR).

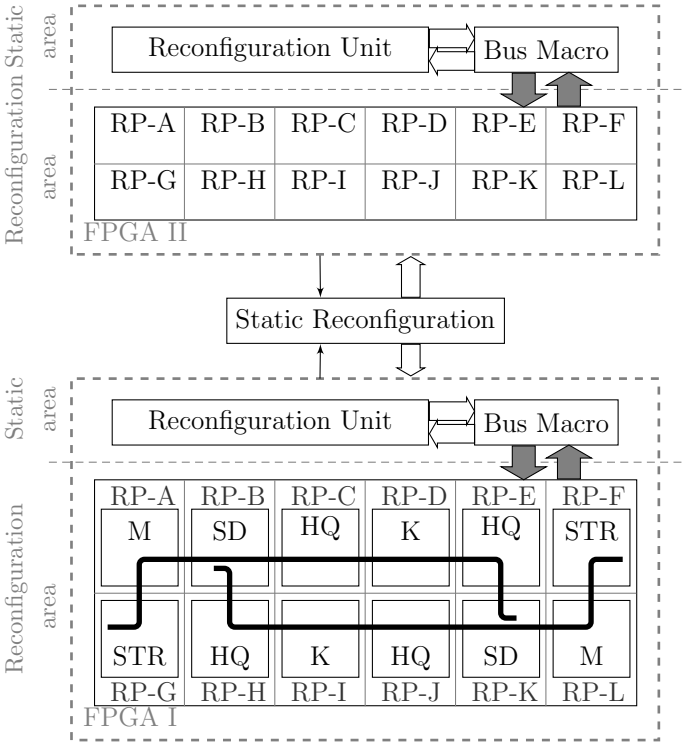

Figure 4.2: Upgraded MDS Architecture

In Figure 4.2, you can see my proposed system. It uses two boards, each with one FPGA, where the same design is loaded. Also the boards could be identical. It simplifies the systems design, reduces time of development and safes money. There are two main parts (Reconfiguration area and Static area) in each FPGA.

Reconfiguration area is a part of FPGAs which we divided into several Reconfiguration Partitions (RP). The number of RP depends on used application and their size depends on the specific architecture of an FPGA. In one RP, there is also a comparator derived from MDS. One set of RMs is prepared for both FPGAs, where each RM belongs to pertinent RP.

Static area is composed of two parts. The Reconfiguration Unit is constructed by FSM, which controls the status of each TSC block in the reconfiguration area. The Bus Macro is a bridge between reconfiguration and static areas and is here present for compatibility with older FPGAs.

Static reconfiguration is the control logic which performs reconfiguration of the whole FPGA (one or both in the same time). The reconfiguration is initiated by checkers from Reconfiguration Units and Comparators.

You can see in Figure 2.10 that each FPGA in MDS is composed of a design and a comparator. These parts are divided into blocks and placed into Reconfiguration Area in UMDS (bottom part of each FPGA shown on Figure 4.2). UMDS uses more simple Static reconfiguration unit than the MDS. It allows to use a high reliable Static reconfiguration unit. The unit is placed between FPGA boards and is capable to start whole reconfiguration. This device contains blocks, which are all designed as TSC ones. TSC schema is used for all parts of the design. PDR is performed when an error is detected by ECC by Xininx or by TSC.

The top part of the design is innovated and it improves reliability by performing partial reconfiguration of faulty part when it is needed.

4.3 Fault Recovery Flow

An error can occur in every part of an UMDS and change the functionality some block. This method achieves 100% of fault cover as described below.

4.3.1 Each FPGA

Description of the flow of one FPGA is mentioned in Figure 4.3. In the picture you can see the function of two main parts: TCS and ECC. The TCS schema allowes to detect B and C faults in the circuit. If a fault is detected, it initializes a reconfiguration of a part or whole FPGA. It depends where the fault was detected. If the fault is not detected by TSC, it means that the fault is in unused area or in used area, but it belongs to classes A, D and some of C. The ECC checker also initiates the reconfiguration (partial or full) in the same way like TSC.

4.3.2 Whole system

The overview of the flow of the whole system is in Figure 4.4. When an error is in the static area, the Static Reconfiguration unit performs reconfiguration of the whole FPGA, where the error was detected. When an error is in Reconfiguration area, it could be in the secured design or in the comparator. Errors in secured design are detected by checkers. An error in the comparator is detected by Static reconfiguration unit or checkers. Static reconfiguration unit reconfigures both FPGAs.

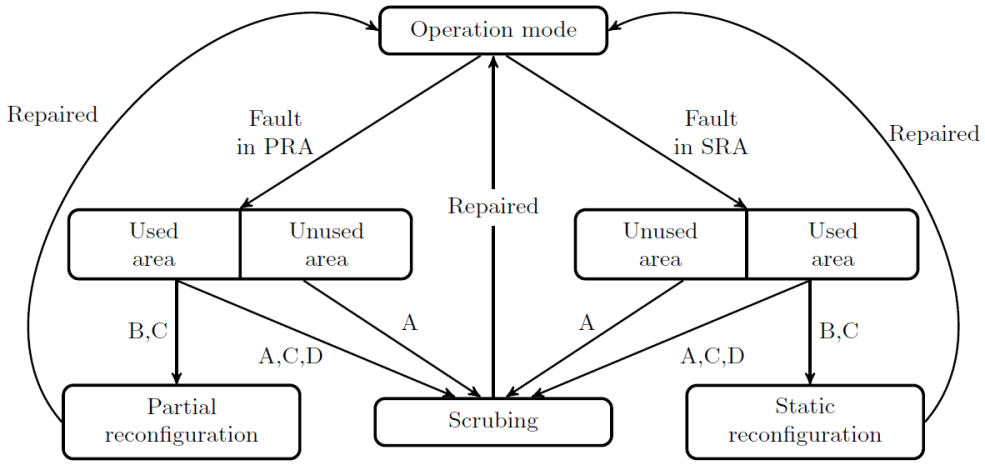

Figure 4.3: Behavioral fault model in 1 FPGA

When an error is detected by some checker, then Reconfiguration unit reconfigures only this RM. For example RP-A part detects the error and RM-A is loaded into RP-A, where the broken block is placed. Other blocks in different parts (RP-B, RP-C, etc.) are able to work at this time.

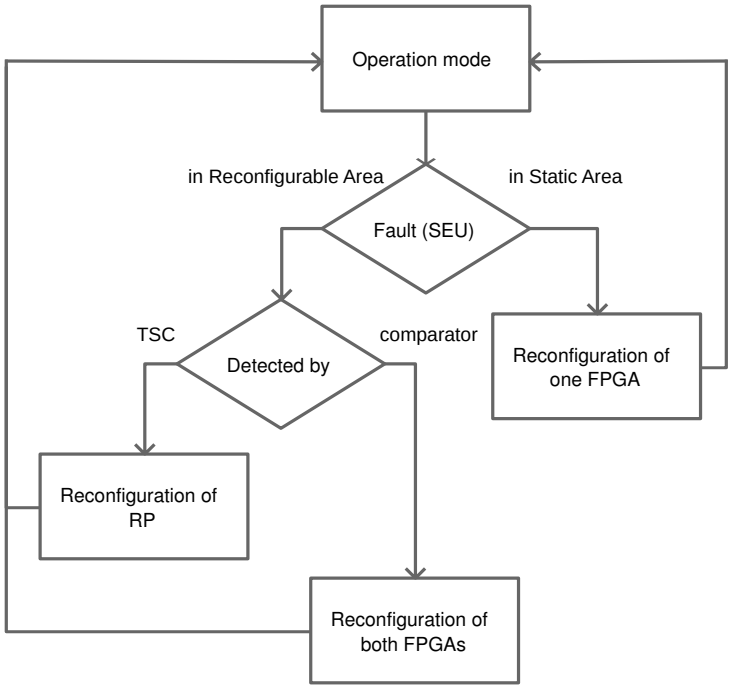

Figure 4.4: The block diagram of the whole system recovery flow

4.4 Area overhead comparison

In this section the area overhead will be compared with TMR and the original duplex system based on TSC (MDS). If we compare these systems, we have to compare all resources, not only used logic like LUTs or Slices.

In basis we can take two variants into account. The first variant is to implement everything in one FPGA and the other is to divide design into two or more FPGAs.

4.4.1 TMR or MDS in one FPGA

- TMR must contain three same logics. If we use only one voter, the weak point will be in the split of the signal. A fault at this point could not be detected.

- TMR contains three same logics with three voters, if we use right TMR in Figure 2.9. Therefore we need three times as much as input wires and output wires and each voter has connected output from each original logic. So used area (logics and routes) is so big and free IOBs rapidly decline.

- MDS contains two same logics and two voters or one (we can chose). In this case we can use one input wires only for one logic plus their corresponding parity input, the same idea for an output.

UMDS could be also implemented in one FPGA, but it will reduce reconfiguration units and lost independence. But the area and IOB usage will be lower than TMR from the basic principle. According to MDS it will have similar area overhead but Availability parameter will be much better.

4.4.2 TMR on three FPGAs and MDS in two FPGAs

- TMR - each FPGA contains one logic with one voter as in Figure 2.9. In this case one input wires and one output wires are used, but each voter has connected two output wires from remaining FPGAs and one output wire is connected from inside. So used area is so big and free IOBs rapidly decline.

- MDS - FPGAs contain one logic and one voter. One input wires and one output wires are used like in TMR, but corresponding parity wire is added for possible output error detection. For voters we need one output wires from second FPGA and one output wires to second FPGA.

UMDS uses only 2 FPGAs, which can safe a lot of resources in whole architecture from sensors to actuators. In comparison with MDS my application is much faster in correction of faults.

4.4.3 Overhead results

You can see that TMR uses more resources and area overhead can be bigger more than three times used logic of one original circuit. The probability of a fault also increases with the area, therefore our method is better in this parameter. Our method is able to detect all possible SEUs in the FPGA. From behavior diagrams you can see the fault coverage and procedure which will be done if the SEU is detected. Our system could be involved only if there will be more than one SEU on a specific place. CRC check with TSC can not allow this situation.

4.5 Synchronization after reconfiguration

When a SEU is detected in RP and partial reconfiguration is applied on this RP, it is important to synchronize whole FPGA with the second one. Otherwise the comparators will detect an error. Whole design is a long pipeline and partial reconfiguration has another advantage here, therefore we approach to this problem in similar way like in pipeline processor. We freeze executing commands in RPs in front of this reconfigured partition, because they work correctly. This RP must be synchronized with the same RP in the second FPGA. Blocks behind this RP must be also synchronized, but at average it is only half of the design.

4.6 Safety calculation

Safety calculations were measured and modeled together with our colleague Martin Kohlik.

4.6.1 MDS

The model shown in Figure 4.5 is used to calculate the failure distribution function of the MDS system.

$Fault\_Free$ is the functional/fault-free state of the system. The fault rate of the first fault is 2λ, because the first fault can affect any of the two instances. The system is in the *Latent* state when it contains a fault that has not been detected yet.

The fault detection rate is labeled as δ. If a fault is detected by the parity checkers, the system will be locked in the Fix_FPGA state. The probability of detecting a fault by parity checkers is labeled as p_{Det}. If a fault is not detected by the parity checkers, the system will be locked in the Fix_All state.

The arc leading from *Latent* to $Hazard$ expresses the probability that a second fault affects the unaffected instance before the first fault is detected.

The system locked in the Fix_FPGA state waits until the repair is finished (repair rate μ – one instance is repaired). The system locked in the Fix_All state also waits until the repair is finished (repair rate μ – both instances are repaired simultaneously). The system is not functional in these states, but the safety is not violated.

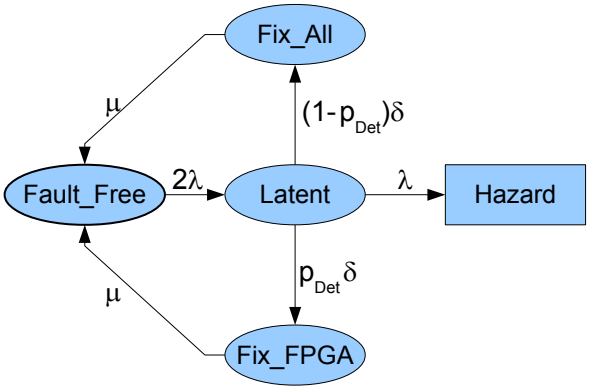

Figure 4.5: Dependability model of the Modified duplex system used to calculate the failure distribution function.

The probability of detection of a fault, the fault rate, and the self-test rate of the system form the following parameters values.

- $\mu = 10^3\,[h^{-1}]$ – the repair rate of the instance
- $\lambda = 10^{-4}\,[h^{-1}]$ – the fault rate
- $\delta = 10^{-1}\,[h^{-1}]$ – the fault detection rate
- $p_{Det} = 0.95$ – the probability of detecting a fault by the parity checkers

4.6.2 UMDS

The model shown in Figure 4.6 is used to calculate the failure distribution function of the Upgraded MDS system.

$Fault\_Free$ is the functional/fault-free state of the system. The first fault can affect the static part of the instance or the reconfigurable blocks. The fault rate of the first fault affecting the static part is $2\lambda_{Stat}$, because the first fault can affect any of the two instances. The system is in the Lat_Stat state when it contains a fault in the static part that has not been detected yet. The fault rate of the first fault affecting reconfigurable blocks is $2\lambda_{Rec}$. The system is in the Lat_Rec state when it contains a fault in the reconfigurable block that has not been detected yet.

The fault detection rate is labeled as δ. If a fault is detected by the parity checkers, the system will be locked in the Fix_Rec state. The probability of detecting a fault by parity checkers is labeled as p_{Det}. If a fault is not detected by the parity checkers, the system will be locked in the Fix_All state. A fault in the static part is always detected. When a fault is detected in the static part, the system will be locked in the Fix_FPGA state.

The arc leading from Lat_Stat to $Hazard$ expresses the probability that a second fault affects the static part of the unaffected instance before the first fault is detected. The

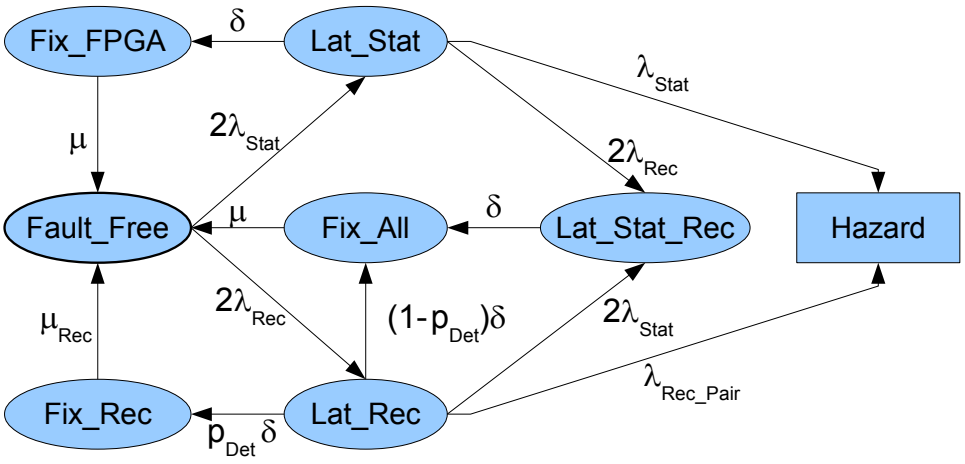

Figure 4.6: Dependability model of the Upgraded modified duplex system used to calculate the failure distribution function.

arc leading from Lat_Rec to $Hazard$ expresses the probability that a second fault affects the paired reconfigurable block of the unaffected instance (the same block as the already affected one, but in the second unaffected instance) before the first fault is detected.

The system locked in the Fix_FPGA state waits until the repair is finished (repair rate μ – one instance is repaired). The system locked in the Fix_All state also waits until the repair is finished (repair rate μ – both instances are repaired simultaneously). The system is not functional in these states, but the safety is not violated. The system locked in the Fix_Rec state waits until the repair is finished (repair rate μ_{Rec} – one reconfigurable block is repaired). The system is fully functional during this repair.

The probability of detection of a fault, the fault rate, and the self-test rate of the block form the following parameters values.

- $\mu = 10^3\,[h^{-1}]$ – the repair rate of the instance
- $\mu_{Rec} = 10 \times 10^3\,[h^{-1}]$ – the repair rate of the reconfigurable block (10 blocks form the instance)
- $\lambda_{Stat} = 0.03 \times 10^{-4}\,[h^{-1}]$ – the fault rate of the static part
- $\lambda_{Rec} = 0.97 \times 10^{-4}\,[h^{-1}]$ – the fault rate of the reconfigurable blocks
- $\lambda_{Rec\_Pair} = (0.97 \times 10^{-4})/10\,[h^{-1}]$ – the fault rate of the paired reconfigurable block
- $\delta = 10^{-1}\,[h^{-1}]$ – the fault detection rate
- $p_{Det} = 0.95$ – the probability of detecting a fault by the parity checkers

4.7 Availability

The models used to calculate steady-state availability of the system assume a single fault only. This assumption is valid, if the double-fault rate calculated from the models shown in the previous section 2.5.2 is low enough to be neglected. All states, rates and values are similar to the models shown in the previous section 4.2. We assume that the MDS is fully functional in *Fault_Free* state only, but the Upgraded MDS is also fully functional during the reconfiguration of the reconfigurable block (*Fix_Rec* state).

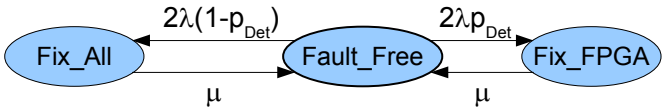

Figure 4.7: Dependability model of the Modified duplex system used to calculate the steady-state availability.

The model used to calculate steady-state availability of the MDS system is shown in Figure 4.7, the model of the Upgraded MDS is shown in Figure 4.8.

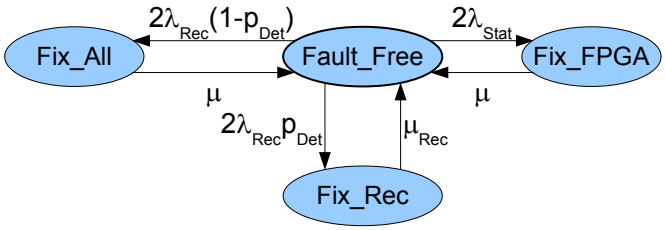

Figure 4.8: Dependability model of the Upgraded modified duplex system used to calculate the steady-state availability.

4.7.1 Results

The plot in Figure 4.9 shows the comparison of the failure distribution functions of the MDS and Upgraded MDS systems. The horizontal axis of the plot represents the time of operation measured in hours, the vertical axis represents the failure distribution function values. The dashed line represents the failure distribution function of the MDS, the thick line represents failure distribution function of the Upgraded MDS system.

The Upgraded MDS system increases the time of operation before the failure probability reaches critical value ca. 10 times (e.g. MDS system reaches failure probability 0.01 in ca. 50,000 hours, but Upgraded MDS in ca. 530,000 hours). The ratio of these two times depends linearly on the number of the reconfigurable blocks (assuming that all other values do not depend on the number of the blocks).

The steady-state availability of the system is also increased. The probability that the system is not available is decreased from ca. $2 \times 10^{-7}[h^{-1}]$ to ca. $1.5 \times 10^{-8}[h^{-1}]$. The ratio of these two values does not depend on the number of the reconfigurable blocks.

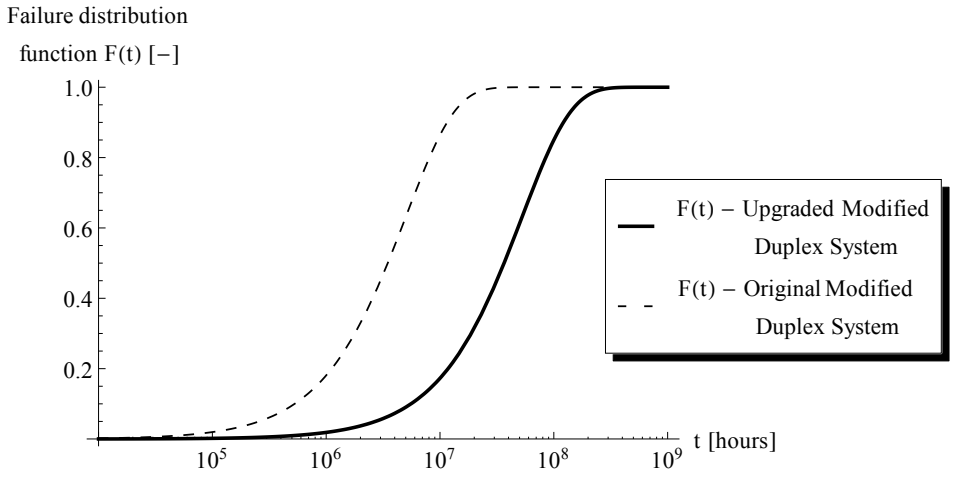

Figure 4.9: Comparison of failure distribution functions of Modified duplex system and Upgraded modified duplex system.

4.8 Summary

In this chapter the new proposed method was described. It uses two FPGAs, TSC blocks and partial reconfiguration. It is based on MDS, which is composed from the original circuit and a checker. The proposed method increase availability and safety. The partial reconfiguration allows faster restore from a fault state to the operational. This method is suitable for modular systems and was presented on the railway station safety device. The area overhead of the whole system is much smaller than TMR and uses less IOBs.

Chapter 5

FTA Comparison

This chapter describes and summarizes methods for FPGAs to improve availability and reliability parameters using partial dynamic reconfiguration and compares them together. The main goal is to increase availability and keep reliability and overhead on acceptable level. Presented methods are suitable for designer to and keep very short time to market with good parameters of the system. All methods are based on programmable hardware (FPGAs) which are sensitive to transient faults like a Single Event Upsets. I combine basic known principles with modern FPGA reconfiguration and evaluate safe using Fault Tree Analysis. Our new methods are developed for industrial and practical FPGA applications like dual channel railway applications and are proposed to reach the minimal area overhead for the low-power design.

5.1 Introduction

The goal is to find a method, how to compare reliable system, compare them and evaluate results. Applications used in space missions or public transport need to satisfy safety standards to avoid tragic consequences. We propose techniques and methodologies how to minimize faults with minimal effort on the designer. Some techniques can restore original function in a very short time thanks to the reconfiguration. Some critical applications must not be interrupted by any faults and tests, we will present also some of these techniques for application without safety state. A radiation tolerant device based on N-modular technique is presented in [41] and comparison of other fault tolerant techniques is in [42].

5.1.1 Safety systems

A safety system is specific where an error can cause an injury to people or a death. The risk analysis must be done during requirement definition phase. The risk is a combination of two elements: Probability and consequence. According to severity of the risk the value of Tolerable Hazard Rate is set in an analytic phase of the system life cycle. In Table 5.1 there is specified according to which Safety Integrity Level must be the design created.

Table 5.1: SIL table

Tolerable Hazard Rate per hour	**Safety Integrity Level**
$10^{-9} \leq THR \leq 10^{-8}$	4
$10^{-8} \leq THR \leq 10^{-7}$	3
$10^{-7} \leq THR \leq 10^{-6}$	2
$10^{-6} \leq THR \leq 10^{-5}$	1

Together with other techniques a Failure Mode and Effect Analysis must be done.

5.1.2 Safety standards

Developers have access to many standards, but most of them are not addressed to FPGAs. The railway standards are described in CENELEC EN 50128 [13], CENELEC EN 50129 [14] and CENELEC EN50126 [10].

Only some specific regulations are written in [43] which are suitable only for aerospace. In other fields there are usually norms for software and hardware and both must be satisfied. All standards agree on V-model, which should be used in safety design in FPGA. This model has a benefit in testing. Each development phase is tested for corresponded requirements.

5.2 Technical Background

FPGAs are a specific category of chips on the edge between one single device and group of many configurable blocks connected by switches and wires on a single chip.

The chip is composed from array of Logic Blocks connected together by programmable routing structure. Logic blocks are basic elements which can compute an easy function, but together they can create a complex function thanks to switches on routing wires.

A bitstream defines all these elements (configuration of logic blocks and switches). In SRAM based FPGAs this bitstream must be stored in non volatile memory and must be loaded on start up. But an advantage is that the configuration could be loaded infinitely even during operation of a part of an FPGA.

Although the radiation is more intensive in space, but some errors can occur even at ground level with lower probability.

5.2.1 Virtex 5

Our simulations and test were performed on a FPGA Xilinx Virtex 5 (XC5VLX20T), which has 6 251 200 configuration bits (length of the bitstream). This device has 3 762

configuration frames, each frame is defined by 41 words and each word has 32 bits. Details of the frame are visible in Figure 5.1 from [44].

According to the documentation [44] there is 36 Frames which configure 20 Configuration Logic Blocks (CLB). One CLB is composed from 2 Slices. Not all words are used for configuration of Slices. Only frames 26-29 and 32-35, but other frames like 0-25 configures interconnection of switches and other things. Together we can easy calculate that 20 CLBs (40 Slices) are configured by 47 232 bits, therefore 1 Slice needs about 1180,8 bits.

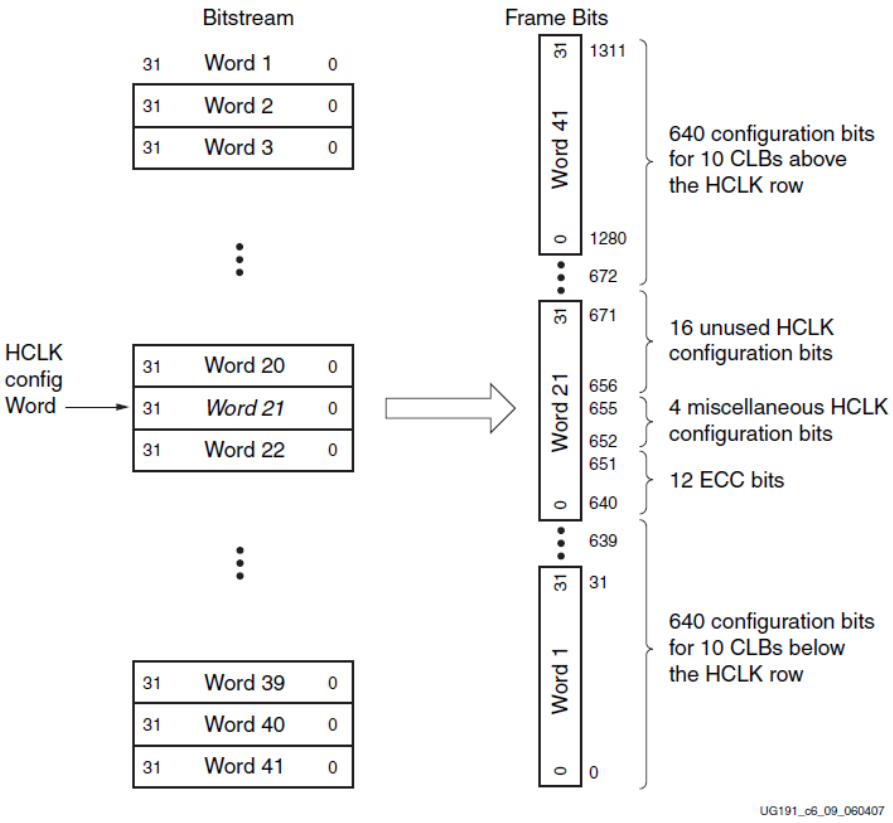

Figure 5.1: Configuration bits in a frame

5.2.2 SEU probability

According to latest Device Reliability Report [45] a Virtex 5 FPGA has 165 Failures in Time (FIT) per Mbit. One Slice has 0,0011808 Mbit which means that after an easy calculation according to equation 5.1 one slice has $\lambda = 1,95 * 10^{-10}$.

$$\lambda = \frac{FIT}{10^9}[h^{-1}] \tag{5.1}$$

5.2.3 Evaluation techniques

5.2.3.1 Reliability

Well known technique for evaluation of probability of a failure is the Fault tree analysis (FTA). It uses top down approach, where the Top event is a failure of the system. Identification of logical dependencies must be correctly done. The FTA is not only about logical connection, but in calculations the mean down time (MDT) is used to evaluate the probability.

This technique is usually used together with Failure Mode and Effect Analysis (FMEA), but this analysis goes deep inside all functions of all blocks and is not important for our methodologies.

Some faults in the circuit could influence output immediately. Some can be observed after a long time, when the circuit switches into a specific state. We assume the worst case, which means that the system is all the time with an undetectable fault until the status is changed and fault is presented. In all calculation I use standard life time of railway equipment which is 25 years. This time is called Mean Down Time (MDT) and is 219000 hours (25 years). This influence the reliability parameter of the system. If the system will be tested, it can reduce MDT and find a fault earlier and repair it.

Each element is synthesized and place and routed into the Virtex V (XC5VLX20T). After place and route the number of slices is taken from the report and used to calculate FIT and λ for each function block according to description in the section 5.2.1.

The whole system is not only about one FPGA, there are other parts like power supplies, sensors, actuators, cables and connectors etc. Together the whole system will not achieve calculated FIT. Therefore the FPGA has to achieve much higher dependability than is required by SIL levels. Target FIT number depends on the whole system and FTA of this whole system.

5.2.3.2 Availability

Availability could be improved by fast correction of a fault. It could be done by different techniques like Partial Dynamic Reconfiguration. Time of the correction is calculated to compare designed methodologies. Reconfigurable partitions are corrected by rewriting of the bitstream, it is done through ICAP, which uses 32 bit bus with clock speed of 100 MHz. It can rewrite 3200 Mbit/s in standard mode. Moreover there are some techniques, how to increase speed of ICAP [46], but it needs some logic and increases area overhead which is not suitable for this application.

5.3 Failure mitigation techniques

There are a lot of different approaches and methodologies like [26] Also some tools are offered by the manufacturer of an FPGA [38]. Generally all these techniques should be divided into two different approaches:

- Modification of a function
  – parity, decomposition etc.
- Addition of another circuit
  – reconfiguration unit, voters, checkers etc.

These approaches could be also combined together like [47]. All calculations in this paper will correspond to the railway station safety device described above. This paper is focused on a combination of both groups. It has same benefits for developers like easy implementation, they can select a suitable method for specific level of reliability and availability with shorter time to market.

Whole device is composed of different modular system blocks. Each part of a block has to be secured, because a SEU can occur. A change of one bit leads to a modification of the circuit function, often drastically. That causes unpredictable behavior in practical applications, for example the control device can change signals to green in all traffic lights of a crossroad.

5.3.1 Reconfiguration

Partial Dynamic Reconfiguration (PDR) of the FPGAs will be applied because a part of the circuit can be changed without disturbing of a rest of the functional FPGA. PDR will be applied to increase reliability and availability parameters. Reconfigurable modules will be rewritten to repair their transient faults. One big block or few small blocks will be placed in one Reconfiguration Module (RM).

Basic reconfiguration unit, which can read error statuses of comparators or voters is composed from 127 Slices and is able to perform partial reconfiguration of a failure RP. It will increase availability of the system, but slightly decrease reliability in some occasions. The Xilinx Macro uses more resources (174 Slices + 1 BRAM).

5.4 Case study

Basic railway station safety device which is divided into blocks according to Figure 2.3 can achieve even SIL 2. There are all blocks in FTA connected into OR logic function. It means that any failure should cause an error. In sum the system has $\lambda = 1.168 * 10^{-7}$ and FIT 119.9, which corresponds to SIL 2. This basic schema does not allow user to know if the output is in an error state or not.

As is written above SIL 2 of the function in FPGA could not lead to whole system on level SIL 2. Therefore main function in the FPGA must be secured to higher level. In next sections there are mentioned developed methods which are based on common known principles and combined with partial dynamic reconfiguration.

All presented method are designed as a dual channel logic, which is widely used in practical applications. Most of railway system use two same channels of the system from the beginning (a sensor) to the end (an actuator). Safety of this design is partly transfered

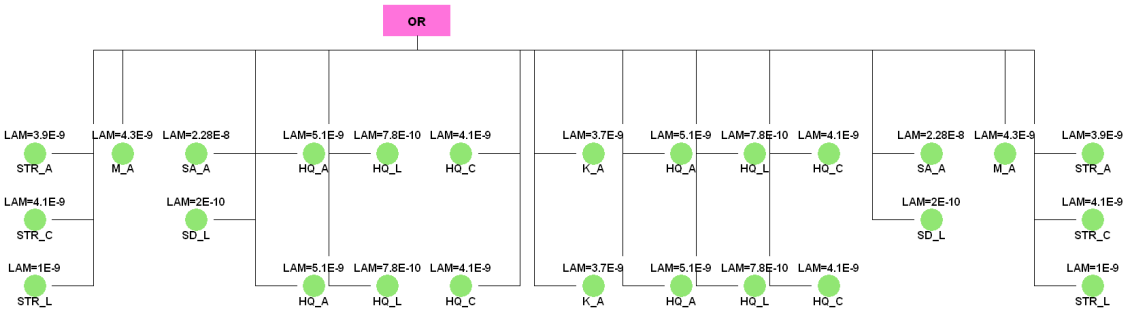

Figure 5.2: FTA of the basic decomposed railway

to the dual channel logic, where the same output is the only correct allowed status. For example two relays must be switched on together, if one is not switched the function is not performed.

5.4.1 One channel techniques

The list of one channel techniques represents only a few examples of basic methods, because the main disadvantage can not be eliminated in one channel design. The weakest point is the output comparator or voter in all cases and variants, which is the single point of failure. More over the only one input could be faulty even outside of the system and all secured devices will work correctly with faulty data.

5.4.1.1 Duplication - one channel

The safety circuit is duplicated as is shown in the Figure 5.3. Outputs of both same functions are connected to an easy comparator. This comparator can only compare the information and if there is a difference, it will generate error signal, which is connected into reconfiguration unit. It uses only 3 slices for all outputs. This unit will reconfigure both function, because it does not know, where the problem is. If the error will not disappear the comparator block is also reconfigured.

Here is the list of advantages and disadvantages:

+ very easy design
+ low area overhead
− error in comparator can cause wrong output
− comparator does not know in which circuit is the error
− an error in the Reconfiguration unit can cause a malfunction of the whole system

FTA was calculated according to Figure 5.4 and results are in the Table 5.2.

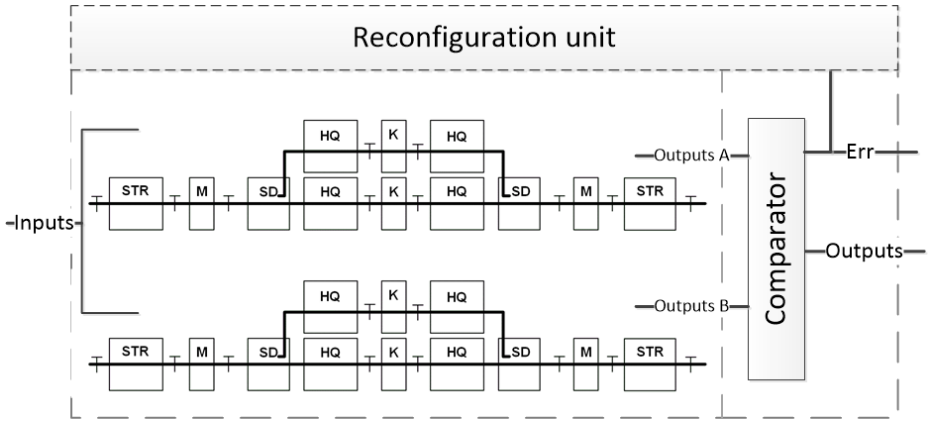

Figure 5.3: Block diagram of the duplication in one FPGA

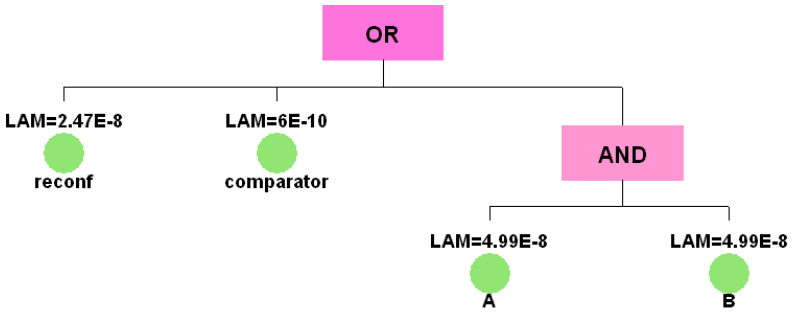

Figure 5.4: FTA of the duplication in one FPGA

5.4.1.2 Triplication - one channel

This method uses most of resources of the FPGA, because the circuit is 3 times in one FPGA. Also the voter is not easy in comparison with a comparator. The voter is quite complicated device and uses 40 slices and has 7,79 FIT. The main advantage of this system is, that it can work during an error in one circuit. Reconfiguration can be performed and no delay or disturbance of the system will be visible. Weak point of this design is the voter, which uses more resources and a fault of this circuit can lead to an error. Block diagram is shown in the Figure 5.7.

+ no interruption during reconfiguration

+ voter knows which circuit is faulty

− area overhead is quite high

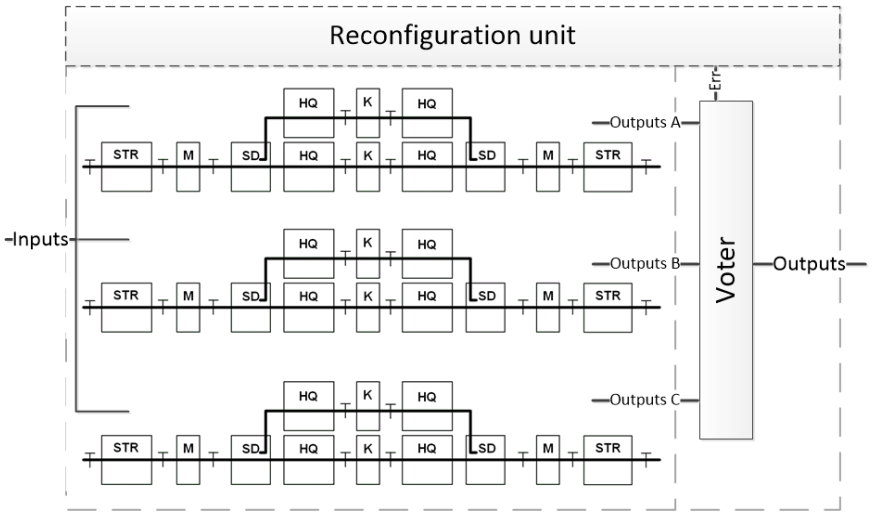

Figure 5.5: Block diagram of the triplication in one FPGA

– the voter is a weak point and error is not detectable

– an error in the Reconfiguration unit can cause a malfunction of the whole system

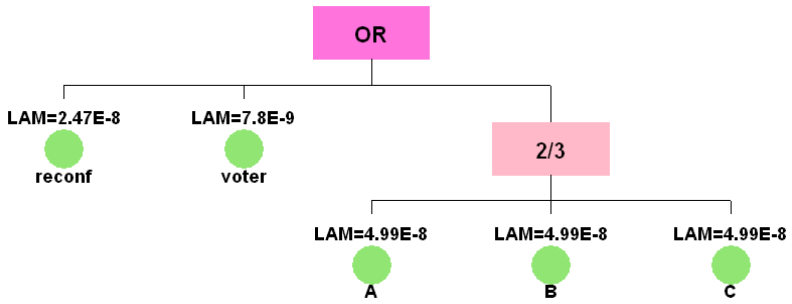

Figure 5.6: FTA of triplication in one FPGA

5.4.1.3 Triplication - three channels

This method uses most of resources in sum. There are 3 voter, 3 reconfirubation logics and 3 crcuits. But the function in onr FPGA is similar to the original function. Because each votes has inputs from all 3 circuits, it uses high amount of I/Os of each FPGA. In some cases it requires much bigger FPGA only because of the number of pins. In other cases, it is even not possible to connect everything. In case of some decomposition the number of required I/O pins increases and the problem is more important. The voter itself

is similar to the voter in previous solution. The main advantage of this system is, that it can work during an error in one FPGA. Reconfiguration can be performed and no delay or disturbance of the system will be visible even if the error will be in the reconfiguration unit of one FPGA. Block diagram is shown in the Figure 5.7.

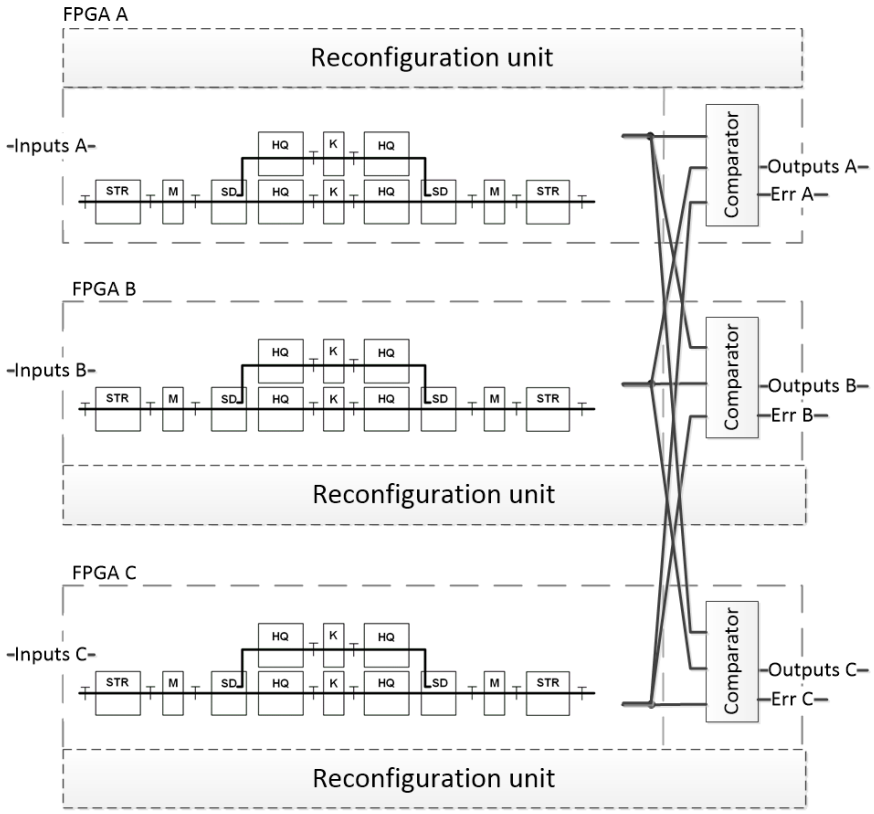

Figure 5.7: Block diagram of the triplication in three FPGAs

+ no interruption during reconfiguration

+ each voter knows which FPGA is faulty

∘ it is important to have 3 channel logic from input to output

− area overhead is very high

− number of I/Os is very high

Table 5.2: Summary of results of first methods

Method	Original	Duplication	TMR	3xFPGA
Size [Slices]	614	1358	2009	2343
Area overhead [%]	100	221.17	327.19	381,6
Number of FPGAs [-]	1	1	1	3
Dual channel logic	no	no	no	yes, 3
Lambda $[h^{-1}]$	$1.17 * 10^{-7}$	$3.11 * 10^{-8}$	$4.5 * 10^{-8}$	$1.83 * 10^{-8}$
Reconfiguration time $[\mu s]$	-	452,47	226.57	299.19
Safety Integrity Level [-]	SIL2	SIL3	SIL3	SIL3

5.4.2 Duplication in one FPGA

The safety circuit is duplicated in one FPGA as is shown in the Figure 5.8 and also inputs and comparators are duplicated. This design is suitable for basic dual channel logic and is enhanced in next sections. This design is very easy for the designer. Comparators are connected only to outputs. If there is a fault in one circuit it will be detected by a comparator and the reconfiguration unit should rewrite both circuits, because there is no information which channel is not correct. Hazard of this solution is in the reconfiguration unit, which is a weak point. The error in it can cause a malfunction of whole FPGA.

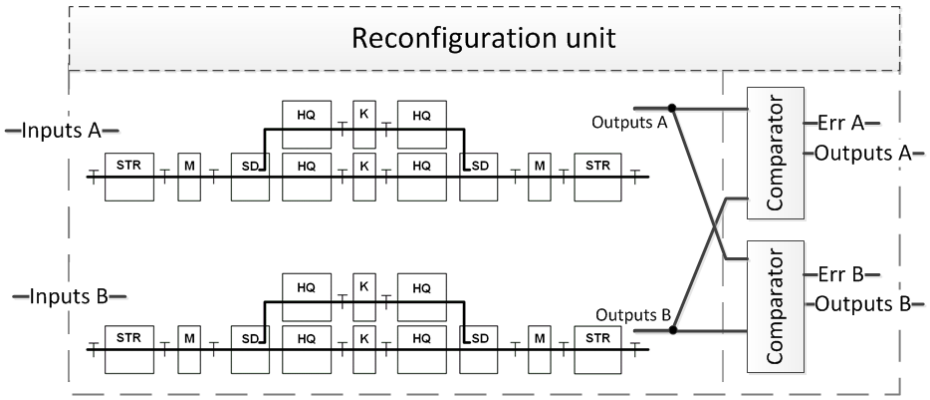

Figure 5.8: Block diagram of the duplication in one FPGA

\+ easy design

\+ low area overhead

\+ error in comparator can cause wrong output of one channel (not an issue)

– comparator does not know in which circuit is the error

– an error in the Reconfiguration unit can cause a malfunction of the whole system

In Figure 5.9 you can see the block diagram of the FTA.

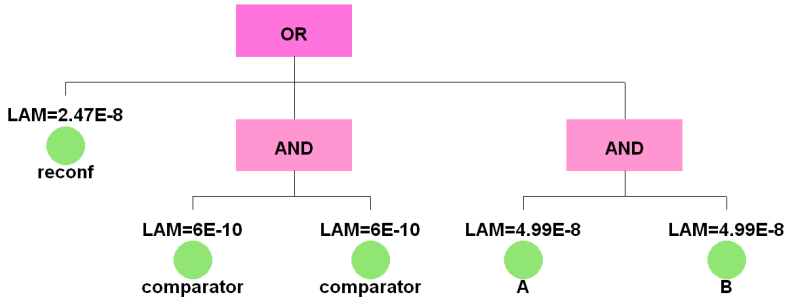

Figure 5.9: FTA of the duplication in one FPGA

5.4.2.1 Division into 4 parts

The design is now divided into four big parts. This solution is suitable for design which are not easy to divide or separate and it does not increase dramatically the work of the designer. Each part is placed in one reconfiguration module with a comparator (RM1A, RM2A,...), which is connected to a previous part from same channel and previous part from the second channel. The comparator has an error output to the reconfiguration unit, which can reconfigure only the previous reconfiguration module. It can rapidly safe time of the reconfiguration. But also in this design the comparator does not know in which channel is the error, therefore the reconfiguration must be performed on both channels. In Figure 5.10 is a block diagram.

\+ faster reconfiguration of a faulty block

\+ area overhead is still on a low level

○ comparators use some resources

○ reliability is acceptable

– an error in the Reconfiguration unit can cause a malfunction of the whole system

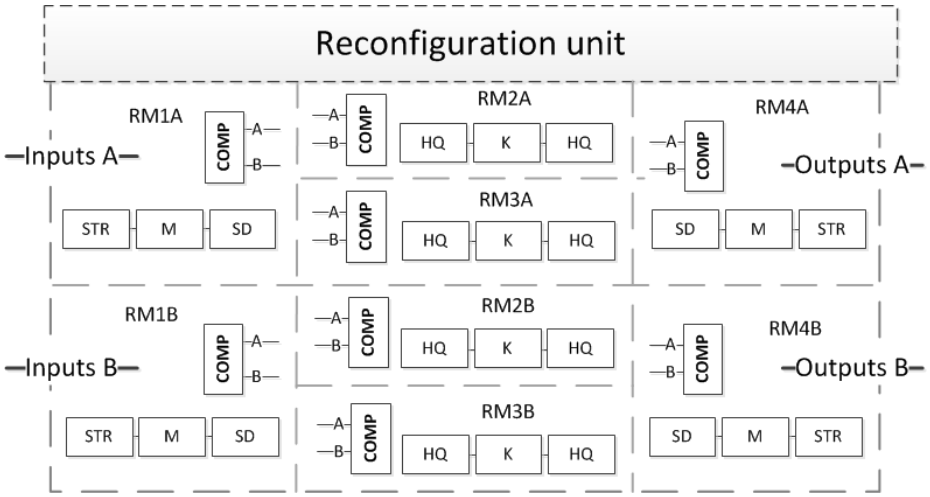

Figure 5.10: Division into 4 parts

5.4.2.2 Division into functional blocks

The design is now divided into basic functional blocks as mentioned above. Each block is placed in one reconfiguration module with a comparator, which is connected in similar way as in previous method. It brings smaller reconfiguration modules but more comparators. Together with the higher number of blocks also the number of compared outputs is increasing. You can see in Figure 5.11 suggested solution. Also here the reconfiguration must be performed on both channels.

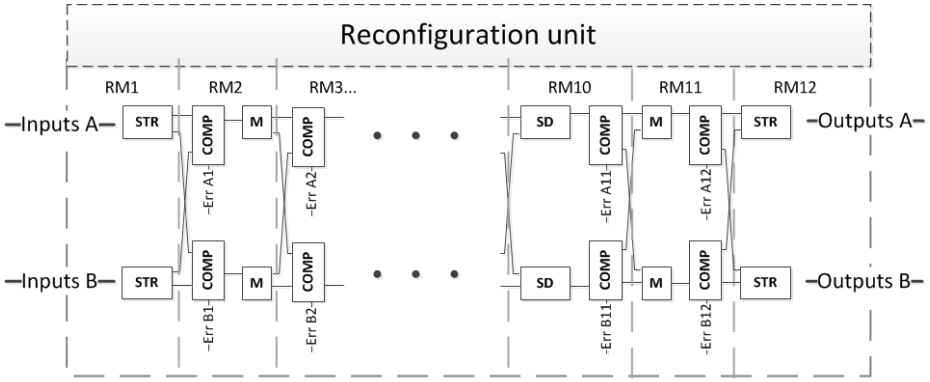

Figure 5.11: Division into functional blocks

+ fast reconfiguration, block are small

∘ area overhead is growing

Table 5.3: Summary of results for duplication

Method	**Original**	**Duplication**	**4 groups**	**Blocks**	**Sub blocks**
Size [Slices]	614	1361	1371	1511	1563
Area overhead [%]	100	221.66	223.29	246.1	254.56
Number of FPGAs [-]	1	1	1	1	1
Dual channel logic	no	yes	yes	yes	yes
Lambda [h^{-1}]	$1.17 * 10^{-7}$	$3.18 * 10^{-8}$	$3.73 * 10^{-8}$	$4.67 * 10^{-8}$	$5.43 * 10^{-8}$
Reconfiguration time [μs]	-	453.13	137.27	87.08	73.8
Safety Integrity Level [-]	SIL2	SIL3	SIL3	SIL3	SIL3

- ○ reliability is still acceptable
- – suitable for specific designs
- – an error in the Reconfiguration unit can cause a malfunction of the whole system

5.4.2.3 Decomposition into sub blocks

The design is decomposed into submodules which was mentioned in Table 3.1. It upgrades the previous method. These submodules are really small and uses only tens of Slices of the FPGA. Therefore the reconfiguration is very fast. Number of comparators is much higher because all connections between blocks must be monitored. The block diagram of this decomposition is complicated and is not shown here.

- \+ very fast reconfiguration of faulty block
- \+ area overhead bigger
- – reliability decreasing
- – needs more work to decompose design
- – an error in the Reconfiguration unit can cause a malfunction of the whole system

5.4.3 Two FPGAs

Dual channel architecture is often used in railway application, but sometimes it is required to have 2 completely independent devices. Therefore the overhead and price of the second FPGA is not so big issue for designer. The circuit is placed in two different FPGAs which can guarantee physically independence. Usually the connection between these two independent channels is made by “safety resistors”, which are usually big resistors (in

MELF package) and designer has to guarantee maximum voltage and power on the resistor. The design is shown in Figure 5.12 and extended only by an easy comparator (similar as is in the chapter 5.4.2). This design has a benefit in 2 independent reconfiguration units, which can work at the same time.

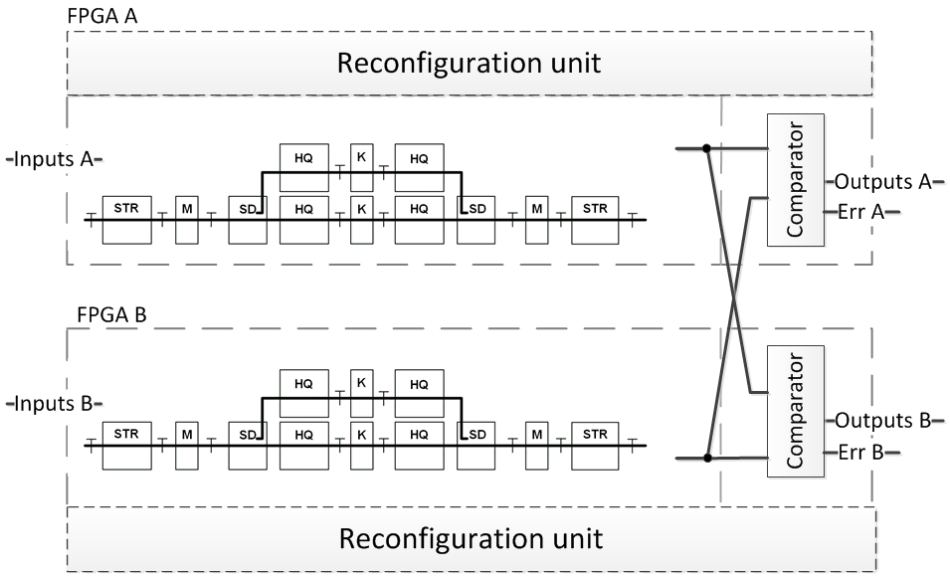

Figure 5.12: Block diagram of two FPGAs

+ same design in both FPGAs
+ physical independence of both channels
+ higher reliability (SIL 4)
+ low area overhead if 2 FPGAs are required
+ faster reconfiguration (1 reconfiguration unit per FPGA)
− an error in the Reconfiguration unit can cause a malfunction of one channel

The FTA was calculated according to Figure 5.13.

5.4.3.1 Division into 4 parts

The system is divided into 4 main functional blocks like in Figure 5.10, but in each FPGA is one channel. You can find benefits of these two approaches in this design. Main benefits are: 2 times faster reconfiguration and independent designs. The main issue is that the design is less reliable and is only in SIL 3 class.

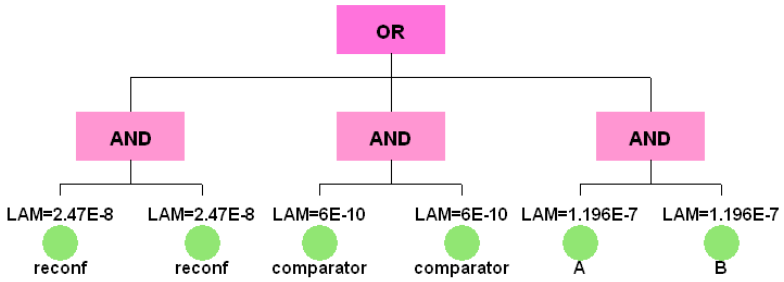

Figure 5.13: FTA of two FPGAs

5.4.3.2 Division into functional blocks

Whole system is separated into functional blocks and a comparator is added to each block. It is together placed in one RP, which is shown in Figure 5.14, where all RPs are separated by a dashed line. This design is still SIL3, reconfiguration time is about 30 % faster than division into 4 parts.

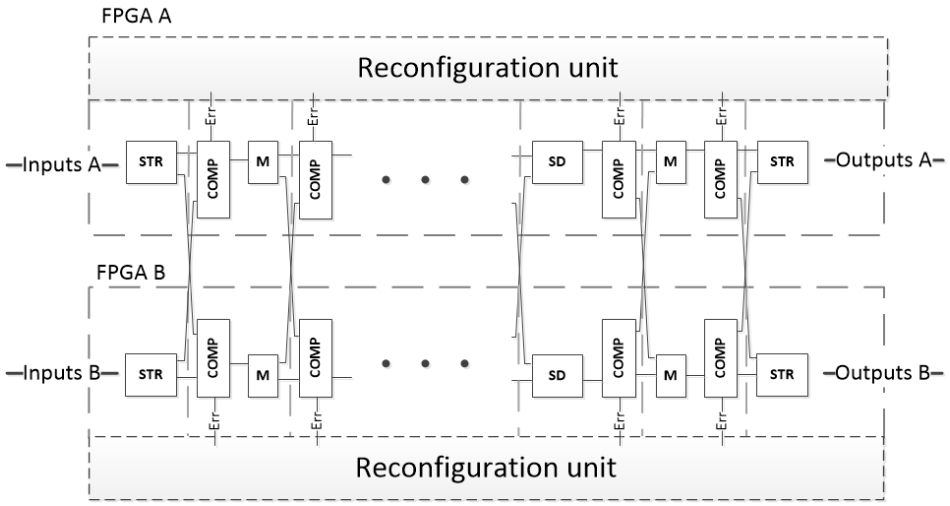

Figure 5.14: Division into functional blocks in two FPGAs

5.4.3.3 Decomposition into sub blocks

This decomposition into sub blocks improve the speed of reconfiguration to the higher level, but together with it the reliability parameter decreases. Reconfiguration speed is almost 6 times faster than the basic duplication into 2 FPGAs and almost twice faster than division into 4 parts.

Table 5.4: Summary of results for 2 FPGAs

Method	**Original**	**2 FPGAs**	**4 groups**	**Blocks**	**Sub blocks**
Size [Slices]	614	1488	1514	1690	1770
Area overhead [%]	100	121.18	123.29	137.62	144.14
Overhead total[%]		242.35	246.58	275.24	288.27
Number of FPGAs [-]	1	2	2	2	2
Dual channel logic	no	yes	yes	yes	yes
Lambda $[h^{-1}]$	$1.17 * 10^{-7}$	$6.22 * 10^{-9}$	$1.28 * 10^{-8}$	$2.23 * 10^{-8}$	$2.99 * 10^{-8}$
Reconfiguration time $[\mu s]$	-	226.57	71.96	47.23	38.75
Safety Integrity Level [-]	SIL2	SIL4	SIL3	SIL3	SIL3

5.4.4 Two FPGAs and duplication in both

This version in Figure 5.15 extends the architecture described in section 5.4.3 by version in section 5.4.2. The safety function is duplicated in both FPGAs. First comparator compares internal circuits, second comparator compare internal data with data from the second channel. If there is an error in one circuit it is detected by first comparator. The FPGA reconfigures both circuits inside, but the other FPGA is without an error. In this case both PFGAs have a good output, because the faulty FPGA takes output from the good one. And location of the error is easier. If there is an error in second comparator, outputs are different - it is a safe state and rest of dual channel logic knows it. Reconfiguration of both FPGAs can be started externally. This solution uses much more resources, but has most advantages. Speed of the reconfiguration is similar to duplication in one FPGA. You can find the FTA in Figure 5.16.

+ same design in both FPGAs
+ physical independence of both channels
+ easy comparators inside
+ one fault tolerant
+ an error in the Reconfiguration unit can not cause a malfunction of the system
∘ slightly more comparators
− slower reconfiguration
− big area overhead

In Figure 5.17 you can see the flow of the system. Where no state is a total failure. λ is a probability of a fault in reconfigurable partition, λR is a probability of a fault in reconfiguration unit and λ_C is a probability of a fault in comparators.

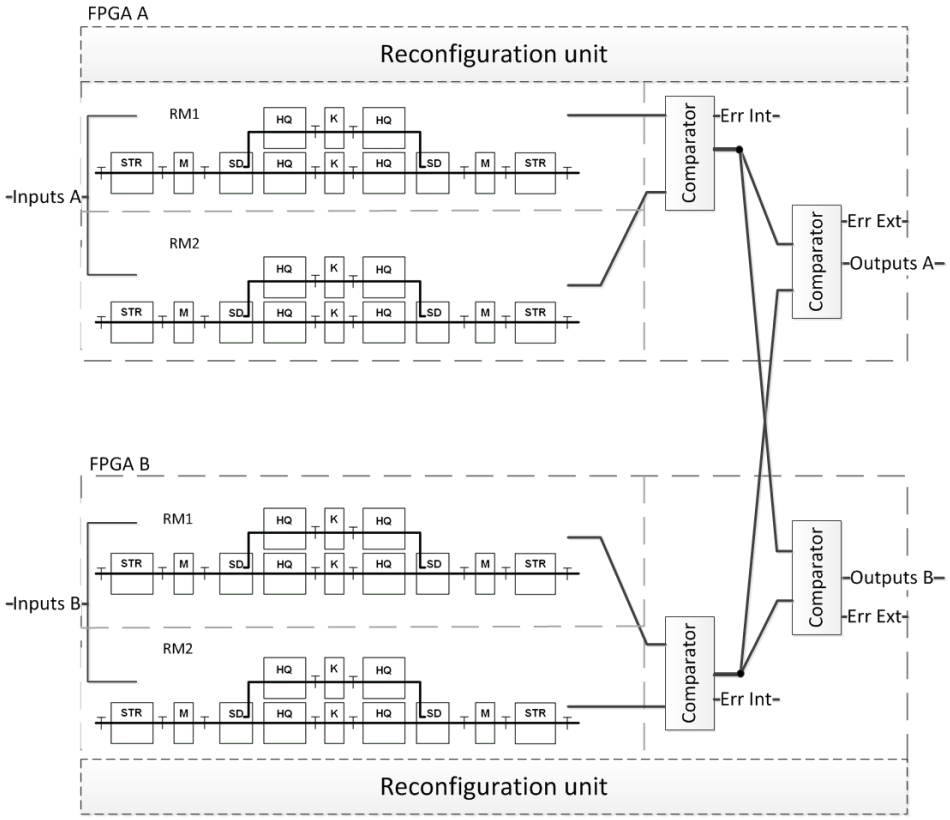

Figure 5.15: Block diagram of the duplication in both FPGAs

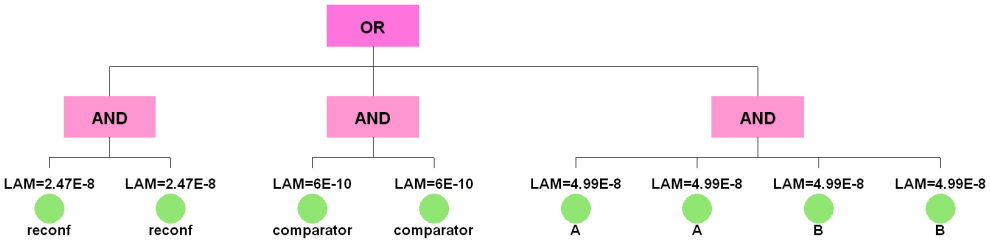

Figure 5.16: The FTA of duplication in two FPGAs

5.4.4.1 Division into 4 parts

This solution combines previously used approaches of 2 FPGAs and division into 4 blocks. This brings very low number of comparators and has about 3 times faster reconfiguration. Reliability of SIL 4 still remain.

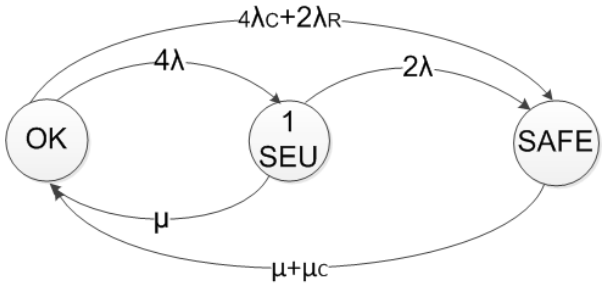

Figure 5.17: FTA of duplication in two FPGAs

Table 5.5: Summary of results for duplication in 2 FPGAs

Method	**Original**	**2x Duplic.**	**4 groups**	**Blocks**	**Sub blocks**
Size [Slices]	614	2734	2774	3126	3286
Area overhead [%]	100	222.64	225.89	254.56	267.59
Overhead total[%]		445.27	451.79	509.12	535.18
Number of FPGAs [-]	1	2	2	2	2
Dual channel logic	no	yes	yes	yes	yes
Lambda $[h^{-1}]$	$1.17 * 10^{-7}$	$2.65 * 10^{-10}$	$2.82 * 10^{-10}$	$3.13 * 10^{-10}$	$4.27 * 10^{-10}$
Reconfiguration time $[\mu s]$	-	453,13	143.91	94.46	77.49
Safety Integrity Level [-]	SIL2	SIL4	SIL4	SIL4	SIL4

5.4.4.2 Division into functional blocks

This solution is very optimal for this design, because complexity of the division into functional blocks is not very high and reliability decreases only slightly, which could be acceptable. In other hand the reconfiguration time decreased rapidly. This version is still SIL4.

5.4.4.3 Decomposition into sub blocks

This is the most complicated solution with very high number of comparators in both FPGAs, but it has very low reconfiguration times and acceptable reliability on SIL 4.

5.4.5 Standard benchmarks

A set of standard benchmarks was synthesized in Xilinx ISE. Their size in slices was used to calculate the average size of the bitstream and a probability of these small circuits was calculated. Almost all of them are SIL 4, but with some additional logic it will need some security technique. Also it is important to know if the output is in failure mode. These

Table 5.6: Size in Slices and lambda for benchmarks

Benchmark	Slices	kbits	Lambda $[h^{-1}]$
apla	14	16.14	$2.73 * 10^{-9}$
br1	12	13.84	$2.34 * 10^{-9}$
br2	8	9.23	$1.56 * 10^{-9}$
dk17	11	12.68	$2.14 * 10^{-9}$
dk27	7	8.07	$1.36 * 10^{-9}$
dk48	19	21.91	$3.7 * 10^{-9}$
ex1010	188	216.79	$3.66 * 10^{-8}$
f51m	5	5.77	$9.74 * 10^{-10}$
gary	42	48.43	$8.18 * 10^{-9}$
mp2d	11	12.68	$2.14 * 10^{-9}$
newapla	7	8.07	$1.36 * 10^{-9}$
newcpla1	8	9.23	$1.56 * 10^{-9}$
newcpla2	5	5.77	$9.74 * 10^{-10}$
p82	7	8.07	$1.36 * 10^{-9}$
sex	9	10.3	$1.75 * 10^{-9}$
sqr6	8	9.23	$1.56 * 10^{-9}$

calculations are in Table 5.6. This table shows the relation between area and reliability and also shows that these benchmarks are very small in comparison to real circuits like railway station safety device.

5.5 Results

You can find all results at the end of each part from duplication in one FPGA are in Table 5.3. All methods used in two FPGAs are in Table 5.4 and combination of both are in Table 5.5. All tables includes the original function and are comparable each other. From our results you can see that the big overhead can not increase reliability and a need of fast reconfiguration increase resources. Comparison of the one channel circuits and TMR in one or three FPGAs is in Table 5.2.

According to simulations and measurement in tables, you can see that for higher availability and lower reconfiguration times the design needs more resources. It forced reliability on the opposite side. With more resources the probability of an error increase. If SIL 4 is required, it is not possible to satisfy it with one FPGA, a designer has to choice a slow reconfiguration and a single circuit in each FPGA or duplication in both FPGAs and a faster reconfiguration. If SIL 3 is enought it could be satisfied with one or 2 FPGAs.

All these methods could be combined together to achieve an optimal level of safety, availability and difficulty. This chapter described basic methods and compared them in speed of reconfiguration, reliability and resources. According to these results it does not make sense to create a huge overhead.

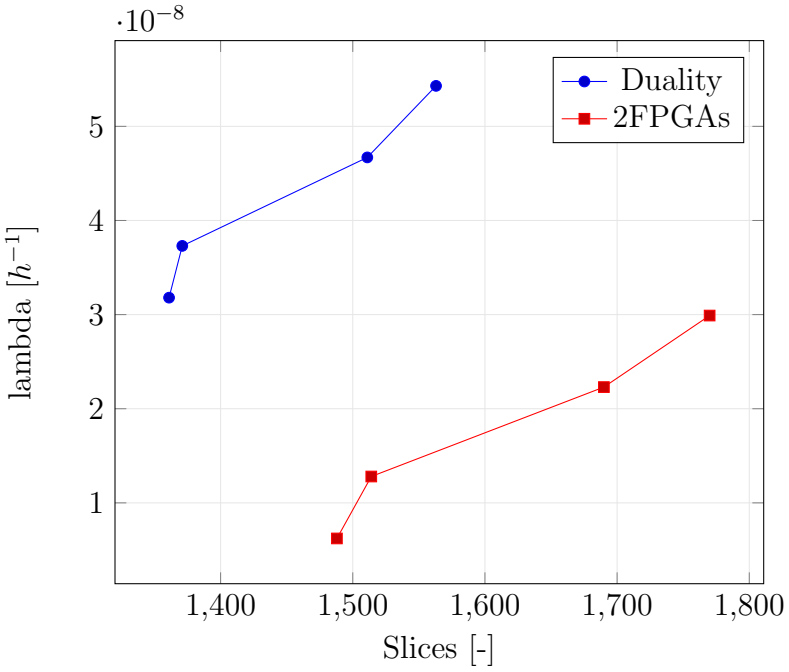

Figure 5.18: Comparison of reliability vs. slices (duplication and 2 FPGAs)

A nice overview of decreasing of the reliability you can see in Figures 5.18 and 5.19. On the X axis there is the number of slices which are used and on the Y axis is a value of calculated reliability parameter. In both plots the worse value is higher (higher lambda is lower reliability).

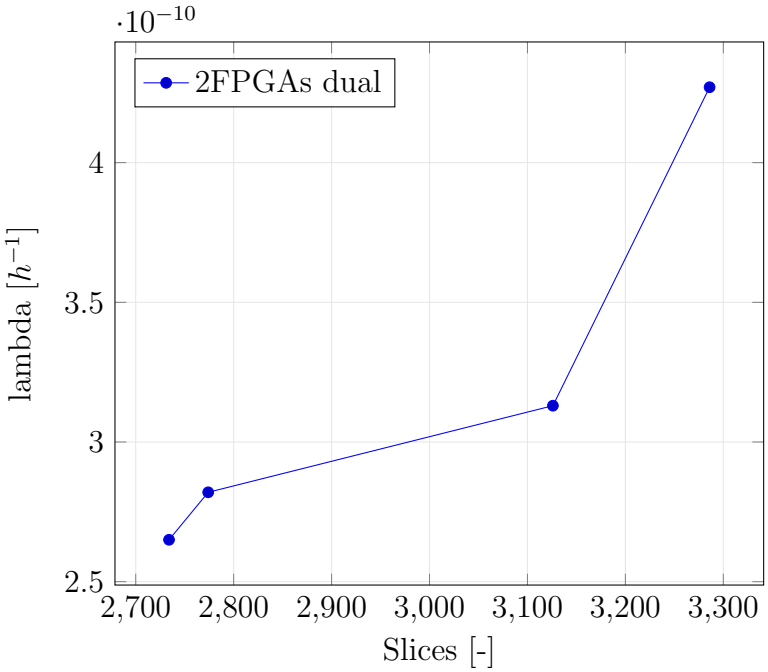

Figure 5.19: Comparison of reliability vs. slices (duplication in 2 FPGAs)

Chapter 6

Main Results

6.1 Technology based design

The main researched information is that the railway station safety device uses less resources in LUT-4 structures. It is quite interesting, because standard MCNC benchmarks show different tendency in average. Some of benchmarks are also worse if they are mapped in LUT-6. It leads to an evident observation, that modern devices based on LUT-6 are not the best solution for all circuits. This result must be taken into account, because it could dramaticly influence reliability, availability and other properties like speed and price.

This behavior is caused by a specific internal functions and connections of this railway circuits. Even in the railway circuit, you can observe that some blocks could be mapped better in LUT-6. But over all blocks the LUT-4 structure is better. There were compared two different railway situations, which show similar results and the advantage of LUT-4 vs. LUT- 6 in the number of LUTs is about 24% more for LUT-6.

The result for the reliable system is to focus at first on the most suitable hardware architecture and than start with other reliability improvements. In some cases it will not be necessary.

The second result from this chapter was an observation and simulation of counters. There are two tasks difference between basic binary counter and Gray code counter and between wide counter and two serial counter was tested on number of faults and FS property. Gray code counter shows higher FS property up to 12 bit width. Probability of the reliability of two counters in series is better for two counter in single fault model, which the FPGA is. Therefore as a result it is better to use 2 or more Gray code counters in series rather than one.

6.2 UMDS

In this section the new proposed method to increase reliability and availability is described. This method upgrades the original MDS and solves its disadvantages like low availability.

The method is based on two FPGAs with the same design, which is easy for a designer. Each FPGA is divided into two parts. The secured circuit is placed in the reconfiguration area and the control logic is in the static area. This logic controls error signals from TSC blocks and from checkers.

The Upgraded MDS has a very high reliability thanks to a dual channel logic and mechanisms inside like TCS blocks and partial dynamic reconfiguration. The system has higher availability in comparison with MDS 4.9. A failure distribution function was calculated by a Markov model.

The proposed method is suitable for modular systems, where each module is placed into one RP, secured by TSC and a SEU will be detected and rewritten only in this RP, if the SEU will be in a different part like reconfiguration unit, the whole FPGA will be rewritten and repaired.

6.3 FTA comparison

Information researched in the last chapter describe the relationship between area overhead, reliability and availability. Some basic methods were presented and combined together like a real design. All duplex and triplex systems were compared in speed of reconfiguration and resource usage. FTA was used to evaluate the reliability from used size of configuration memory. That was calculated from the number of Slices of the circuit. Availability was calculated by size of the biggest reconfigurable unit and a maximal speed of the ICAP interface.

All parts of the circuit like reconfigurable unit or additional comparators and voters were considered in the calculations. They are often not included and results look awesome. These parts increase the area overhead and utilize more parts of the FPGA, which can be hit by a particle which causes SEU. From all tables in this chapter is visible, that reliability decreases when more resources are used. Also in Figure 5.18 and Figure 5.19 you can see decreasing tendency of the reliability. The only possibility how to increase reliability with more resources is a clever idea and smart connection of blocks or FPGAs, but in comparison with other methods the overhead is only about 225%. Easy additional circuits will not increase reliability, you can see that TMR uses much more resources and results are not appropriate.

The result of this chapter is that a very low granularity can not increase reliability, it can only make better availability for the price of the reliability. The dual FPGA logic shows best results and fulfill railway safety standard. Easy security by comparators and duality in two FPGAs can reach SIL4 and satisfy norms. For a small railway station it is possible to use a very low granularity and get the benefit of fast reconfiguration while the whole system is still SIL4. For some bigger railway stations it will not be possible and a compromise in the granularity must be done to achieve SIL4.

Chapter 7

Conclusions

7.1 Summary

The goals of this dissertation thesis were described in the first chapter. In the next section the introduction into technical background and the problem statement were mentioned. Other chapters deal with my designed methods, solutions, proposals and calculations, which were verified on models or emulated in software mostly on the railway station safety device. In the sixth chapter all results were summarized and described.

A part of contributions of this thesis are proposals for hardware developers, who want to achieve a low area and resource overhead in the FPGA. The railway safety device described in VHDL from relay logic has special functions inside and is not in average suitable for FPGAs with LUT-6 structure like Virtex V and newer. These circuits use less resources in LUT-4 FPGAs. Used resources are the key for a low power design and also a high reliable design, because more hardware is used more faults can happen. This statement is not only about SEUs, but also about hardware faults in general. Next techniques in chapter three describe how to minimize counters and how to select a suitable hardware for the design. Counters in standard binary arithmetics uses more resources and are more sensitive. Their parameters FS and ST are not good as similar counters in Gray code.

In the next section the new proposal method based on the Modified Duplex System was presented. This method uses TSC blocks, CRC checksum and partial dynamic reconfiguration to achieve higher availability of the whole system. Upgraded MSD uses two FPGAs and is suitable for modular systems, which could be divided into blocks and placed into reconfigurable partitions. The railway station safety device fits on these parameters of UMDS, it is composed from 5 different block and could be again separated into sub blocks like counters, combinational logic and FSMs.

The last part of my dissertation thesis evaluated common used techniques, combine them together to create a complex set of systems and compared them together in parameters like reliability, area overhead and speed of reconfiguration, which is connected to availability. As the input to calculations the measurement from Xilinx were taken in account. These measurements were made during long time tests of very large number of

their FPGAs made by different technology processes, structures and all f them were placed in different locations over the world. The result shows the average number of failures in time for 1 Mbit of the configuration memory of each FPGA. Because of some available information about internal structure from the documentation of Virtex V, it was possible to calculate the average number of bits for each circuit. Thanks to these results reliability parameters were calculated with FTA technique and appropriate Safety Integrity Level was selected. Also the speed of reconfiguration was calculated and therefore the availability could be compared. All these result are comparable together in tables and figures of the fifth chapter. From these results the connection between area overhead and reliability is visible. Even with enormous amount of logic it is not possible to achieve highest reliability, more important is good idea and right design.

7.2 Contributions of the Dissertation Thesis

The contributions of my dissertation thesis are summarized in the following list:

- I designed a method, how to compare reliable designs. It uses FTA, which is based on the data from the size of the circuit in the specific FPGA. The size is in slices, because the amount of configuration memory can be directly calculated. The final reliability of the whole system can be calculated from the structure and the reliability of each part of the circuit.

- Blocks of the railway station safety device and standard benchmarks were synthesized into different HW structures (LUT-4 and LUT-6). My goal was to utilize less resources. The railway blocks are more suitable for LUT-4. Benchmarks are in average more suitable for LUT-6. Other modern devices must be checked in which technology they will use less resources and have higher reliability. These are the proposals for developers how to select a suitable hardware and easily increase reliability.

- A new UMDS method was designed, behavioral models of fault restoration were presented, a safety calculation and availability models were presented. A comparison with TMR was described and UMDS uses less resources. UMDS has higher reliability in comparison with MDS.

- Basic methods based on comparison were evaluated and compared each other in size and reliability. From these results circuits which uses more resources (for example to increase availability) are less reliable because the additional area is also sensitive on SEU. It is better to use two FPGAs than create a duplication inside, it also has a benefit in totally independent channels.

- An easy design for practical use is a system with duplication, it does not require special skills and from results it has a very good reliability. Easy duplication in dual channel logic has the best results. It can be separated into small parts and achieve

high reliability with good reliability on SIL4. In this system it is easy to calculate reliability and availability.

7.3 Future Work

In the future I would like to continue with implementation of the railway safety device and connect secured system to the model of the railway station. I would like to supervise diploma theses and develop some new more reliable methods.Also I would like to make some real measurement of my approaches in radiation environment described in [24].

These gained information will be used in my other future work not only in safety systems. I would like to continue with a dream work of young boys – with a railway and everything connected to trains.

Bibliography

[1] Martin Zatřepalek, P., Ing. Kubalik Pavel. Zabezpecovaci zarizeni pro zeleznicni stanici zalozene na FPGA. *Diploma thesis*, may 2008.

[2] Moore, A. *FPGAs for dummies*. John Wiley & Sons, Inc., second edition, 2017, ISBN 978-1-119-39047-3.

[3] What is the Difference Between an FPGA and an ASIC, Xilinx. `https://www.xilinx.com/video/hardware/what-is-the-difference-between-an-fpga-and-an-asic.html`, accessed: 2017-08-15.

[4] The FPGA Site. `http://www.fpga-site.com/`, accessed: 2017-08-15.

[5] Felix, J. A.; Dodd, P. E.; Shaneyfelt, M. R.; et al. Radiation Response and Variability of Advanced Commercial Foundry Technologies. *IEEE Transactions on Nuclear Science*, volume 53, no. 6, Dec 2006: pp. 3187–3194, ISSN 0018-9499, doi:10.1109/TNS.2006.886041.

[6] Normand, E. Single event upset at ground level. *IEEE Transactions on Nuclear Science*, volume 43, no. 6, Dec 1996: pp. 2742–2750, ISSN 0018-9499, doi:10.1109/23.556861.

[7] Baumann, R. C. Radiation-induced soft errors in advanced semiconductor technologies. *IEEE Transactions on Device and Materials Reliability*, volume 5, no. 3, Sept 2005: pp. 305–316, ISSN 1530-4388, doi:10.1109/TDMR.2005.853449.

[8] Joe Mallett, S. High-Reliability FPGA Designs. `http://www.eetimes.com/author.asp?section_id=36&doc_id=1324287`, 10 2014.

[9] Gyepes, G.; Arbet, D.; Brenkus, J.; et al. Application of IDDT test towards increasing SRAM reliability in nanometer technologies. In *IEEE 15th International Symposium on Design and Diagnostics of Electronic Circuits & Systems, DDECS 2012, Tallinn, Estonia, April 18-20, 2012*, 2012, pp. 167–170, doi:10.1109/DDECS.2012.6219046. Available from: `https://doi.org/10.1109/DDECS.2012.6219046`

[10] EN 50126 - Railway applications - The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS). Technical report, European Commitee for Electrotechnical Standardization, 1999.

[11] Pradhan, D. K. (editor). *Fault-tolerant Computer System Design*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996, ISBN 0-13-057887-8.

[12] Kubalik, P.; Fiser, P.; Kubatova, H. Fault tolerant system design method based on self-checking circuits. In *12th IEEE International On-Line Testing Symposium (IOLTS'06)*, 2006, ISSN 1942-9398, pp. 2 pp.–, doi:10.1109/IOLTS.2006.37.

[13] EN 50128 - Railway applications - Communication, signalling and processing systems. Software for railway control and protection systems). Technical report, European Commitee for Electrotechnical Standardization, 2011.

[14] EN 50128 - Railway applications - Communication, signalling and processing systems. Safety related electronic systems for signalling). Technical report, European Commitee for Electrotechnical Standardization, 2003.

[15] Basheer Ahmed, C. B.; Pillement, S.; Piestrak, S. J. Fault-aware configurable logic block for reliable reconfigurable FPGAs. In *IEEE International Symposium on Circuits & Systems*, ISCAS 2015, Lisbonne, Portugal, May 2015, p. p.2206. Available from: `https://hal.archives-ouvertes.fr/hal-01104069`

[16] Dobias, R.; Kubalik, P.; Kubatova, H. Dependability computations for fault-tolerant system based on FPGA. In *2005 12th IEEE International Conference on Electronics, Circuits and Systems*, Dec 2005, pp. 1–4, doi:10.1109/ICECS.2005.4633533.

[17] Shen, Z.; Feng, C.; Gao, S.; et al. Study on FPGA SEU mitigation for readout electronics of DAMPE BGO calorimeter. In *2014 19th IEEE-NPSS Real Time Conference*, May 2014, pp. 1–1, doi:10.1109/RTC.2014.7097551.

[18] Danilov, I. A.; Gorbunov, M. S.; Antonov, A. A. SET Tolerance of 65 nm CMOS Majority Voters: A Comparative Study. *IEEE Transactions on Nuclear Science*, volume 61, no. 4, Aug 2014: pp. 1597–1602, ISSN 0018-9499, doi:10.1109/TNS.2014.2311297.

[19] Samudrala, P. K.; Ramos, J.; Katkoori, S. Selective triple Modular redundancy (STMR) based single-event upset (SEU) tolerant synthesis for FPGAs. *IEEE Transactions on Nuclear Science*, volume 51, no. 5, Oct 2004: pp. 2957–2969, ISSN 0018-9499, doi:10.1109/TNS.2004.834955.

[20] Ichinomiya, Y.; Tanoue, S.; Amagasaki, M.; et al. Improving the Robustness of a Softcore Processor against SEUs by Using TMR and Partial Reconfiguration. In *2010 18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines*, May 2010, pp. 47–54, doi:10.1109/FCCM.2010.16.

[21] Niknahad, M.; Sander, O.; Becker, J. FGTMR - Fine grain redundancy method for reconfigurable architectures under high failure rates. In *The 16th North-East Asia Symposium on Nano, Information Technology and Reliability*, Oct 2011, pp. 186–191, doi:10.1109/NASNIT.2011.6111144.

[22] Kubalik, P.; Dobias, R.; Kubatova, H. Dependable Design for FPGA Based on Duplex System and Reconfiguration. In *9th EUROMICRO Conference on Digital System Design (DSD'06)*, 2006, pp. 139–145, doi:10.1109/DSD.2006.38.

[23] Lanuzza, M.; Zicari, P.; Frustaci, F.; et al. Exploiting Self-Reconfiguration Capability to Improve SRAM-based FPGA Robustness in Space and Avionics Applications. *ACM Trans. Reconfigurable Technol. Syst.*, volume 4, no. 1, Dec. 2010: pp. 8:1–8:22, ISSN 1936-7406, doi:10.1145/1857927.1857935. Available from: `http://doi.acm.org/10.1145/1857927.1857935`

[24] Vanat, T.; Pospisil, J.; Krizek, F.; et al. A System for Radiation Testing and Physical Fault Injection into the FPGAs and Other Electronics. In *2015 Euromicro Conference on Digital System Design*, Aug 2015, pp. 205–210, doi:10.1109/DSD.2015.98.

[25] Ceschia, M.; Violante, M.; Reorda, M. S.; et al. Identification and classification of single-event upsets in the configuration memory of SRAM-based FPGAs. *IEEE Transactions on Nuclear Science*, volume 50, no. 6, Dec 2003: pp. 2088–2094, ISSN 0018-9499, doi:10.1109/TNS.2003.821411.

[26] Straka, M.; Miculka, L.; Kastil, J.; et al. Test platform for fault tolerant systems design properties verification. In *2012 IEEE 15th International Symposium on Design and Diagnostics of Electronic Circuits Systems (DDECS)*, April 2012, pp. 336–341, doi:10.1109/DDECS.2012.6219084.

[27] "Virtex-4 Family Overview", DS112 (v3.1). Technical report, Xilinx, August 30, 2010. Available from: `http://www.xilinx.com/support/documentation/data_sheets/ds112.pdf`

[28] "Virtex-5 Family Overview", DS100 (v5.1). Technical report, Xilinx, August 21, 2015. Available from: `http://www.xilinx.com/support/documentation/data_sheets/ds100.pdf`

[29] "Virtex-6 Family Overview", DS150 (v2.5). Technical report, Xilinx, August 20, 2015. Available from: `http://www.xilinx.com/support/documentation/data_sheets/ds150.pdf`

[30] "Spartan-3 FPGA Family Data Sheet", DS099. Technical report, Xilinx, June 27, 2013. Available from: `http://www.xilinx.com/support/documentation/data_sheets/ds099.pdf`

[31] "Spartan-6 Family Overview", DS160 (v2.0). Technical report, Xilinx, October 25, 2011. Available from: http://www.xilinx.com/support/documentation/data_sheets/ds160.pdf

[32] "Stratix Device Handbook, Volume 1", v3.2. Technical report, Altera, July 2005. Available from: http://www.altera.com/literature/hb/stx/stratix_section_1_vol_1.pdf

[33] "Stratix II Device Handbook, Volume 1". Technical report, Altera, May 2007. Available from: http://www.altera.com/literature/hb/stx2/stx2_sii5v1_01.pdf

[34] Lesea, A.; Drimer, S.; Fabula, J. J.; et al. The rosetta experiment: atmospheric soft error rate testing in differing technology FPGAs. *IEEE Transactions on Device and Materials Reliability*, volume 5, no. 3, Sept 2005: pp. 317–328, ISSN 1530-4388, doi: 10.1109/TDMR.2005.854207.

[35] Yang, S. Logic Synthesis and Optimization Benchmarks. Dec. 1988.

[36] R. K. Brayton, e. a. *"Logic Minimization Algorithms for VLSI Synthesis"*. Kluwer Academic Publishers, 1984.

[37] Fiser P., H. J. *"BOOM a heuristic Boolean minimizer"*. IEEE/ACM International Conference on, vol., no., pp.439-442, 2001.

[38] Chapman, K. "SEU Strategies for Virtex-5 Devices", XAPP864 (v2.0). Technical report, Boston, MA, April 1, 2010.

[39] Xilinx. "Partial Reconfiguration User Guide", UG702 (v14.1). Technical report, Boston, MA, April 24, 2012.

[40] Borecky, J.; Kubalik, P.; Kubatova, H. Reliable Railway Station System Based on Regular Structure Implemented in FPGA. In *2009 12th Euromicro Conference on Digital System Design, Architectures, Methods and Tools*, Aug 2009, pp. 348–354, doi:10.1109/DSD.2009.210.

[41] Tarrillo, J.; Kastensmidt, F. L.; Rech, P.; et al. Neutron Cross-Section of N-Modular Redundancy Technique in SRAM-Based FPGAs. *IEEE Transactions on Nuclear Science*, volume 61, no. 4, Aug 2014: pp. 1558–1566, ISSN 0018-9499, doi: 10.1109/TNS.2014.2343259.

[42] Morgan, K. S.; McMurtrey, D. L.; Pratt, B. H.; et al. A Comparison of TMR With Alternative Fault-Tolerant Design Techniques for FPGAs. *IEEE Transactions on Nuclear Science*, volume 54, no. 6, Dec 2007: pp. 2065–2072, ISSN 0018-9499, doi:10.1109/TNS.2007.910871.

[43] ECSS-Q-ST-60-02C ASIC and FPGA development. Technical report, European Cooperation for Space Standardization, 2008. Available from: http://ecss.nl/standard/ecss-q-st-60-02c-asic-and-fpga-development/

[44] Xilinx. “Virtex-5 FPGA Configuration User Guide”, UG191 (v3.12). Technical report, Xilinx, May 8, 2017.

[45] Xilinx. “Device Reliability Report”, UG116 (v10.6.1). Technical report, Xilinx, July 1, 2017.

[46] Hansen, S. G.; Koch, D.; Torresen, J. High Speed Partial Run-Time Reconfiguration Using Enhanced ICAP Hard Macro. In *2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum*, May 2011, ISSN 1530-2075, pp. 174–180, doi:10.1109/IPDPS.2011.139.

[47] Kamanu, E.; Reddy, P.; Hsu, K.; et al. A new architecture for single-event detection and reconfiguration of SRAM-based FPGAs. In *High Assurance Systems Engineering Symposium, 2007. HASE '07. 10th IEEE*, Nov 2007, ISSN 1530-2059, pp. 291–298, doi:10.1109/HASE.2007.68.

Reviewed Publications of the Author Relevant to the Thesis

[A.1] P. Vít, H. Kubátová "High Availability and Reliability Dual Channel Systems Based on Reconfiguration". In *submited to Reconfig 2017,* 17^{th} *International Conference on Reconfigurable Computing and FPGAs*, Cancun, Mexico: December, 2017, ISBN: 978-1-5090-2816-0.

[A.2] J. Borecký, M. Kohlík, P. Vít, H. Kubátová "Enhanced Duplication Method with TMR-Like Masking Abilities". In *DSD 2016,* 19^{th} *Euromicro Conference on Digital System Design*, Limassol, Cyprus: August, 2016, ISBN: 978-1-5090-2816-0.

[A.3] J. Borecký, P. Vít, H. Kubátová "Fault Recovery Method with High Availability for Practical Applications". In *MEMICS 2014,* 9^{th} *Doctoral Workshop on Mathematical and Engineering Methods in Computer Science*, Telc, Czech Republic, October, 2014, ISBN: ISBN 978-80-214-5022-6.

[A.4] P. Vít, J. Borecký, M. Kohlík, H. Kubátová "Fault Tolerant Duplex System with High Availability for Practical Applications". In *DSD 2014,* 17^{th} *Euromicro Conference on Digital System Design*, Verona, Italy: August, 2014, ISBN: 978-1-4799-5793-4.

[A.5] J. Borecký, P. Vít, H. Kubátová "Fault Recovery Method of Modular Systems based on Reconfigurations". In *PESW 2014,* 2^{th} *Prague Embedded Systems Workshop*, Prague, Czech Republic: June, 2014,

[A.6] J. Borecký, P. Vít, H. Kubátová "Fault Recovery Method of Modular Systems based on Reconfigurations". In *DUOC 2014, Design with Uncertainty Opportunities and Challenges*, York, United Kingdom: March, 2014,

[A.7] Vít, P.; Kubátová, H. "Dependability structures testing device based on real model of a railway station". In *POSTER 2012 - 16th International Student Conference on Electrical Engineering*, Prague, Czech Republic: CTU, Faculty of Electrical Engineering, 2012, ISBN: 978-80-01-05043-9.

[A.8] J. Borecký, P. Vít, H. Kubátová "Self Repair Architectures Based on Partial Dynamic and Static Reconfiguration". In *MEMICS 2011, 7th Doctoral Workshop on Mathematical and Engineering Methods in Computer Science*, Lednice, Czech Republic, October, 2011, ISBN: ISBN 978-80-214-4305-1.

[A.9] Vít, P.; Kubátová, H. "Increasing Dependability by Fitting Circuit on FPGA". In *Počítačové architektury & diagnostika*, Bratislava, Slovakia: STU v Bratislave, 2011, ISBN: 978-80-227-3552-0.

[A.10] Vít, P.; Kubátová, H. "Impact of FPGA Technology Process on Depandability of Counters". In *Proceedings of the Work in Progress Session - DSD 2011*, Oulu, Finland: University of Oulu, 2011, ISBN: 978-3-902457-30-1.

[A.11] Vít, P.; Kubátová, H. "Using Decomposition to Create Fault Secure Counters of the Railway Station Safety Device". In *POSTER 2011 - 15th International Student Conference on Electrical Engineering*, Prague, Czech Republic: CTU, Faculty of Electrical Engineering, 2011, ISBN: 978-80-01-04806-1.

[A.12] Vít, P.; Kubátová, H. Návrh obodů s volitelnou úrovní spolehlivosti na bázi FPGAIncreasing Dependability by Fitting Circuit on FPGA". In *Počítačové architektury & diagnostika*, Ceskovice, Czech Republic: STU v Bratislave, 2010, ISBN: 978-80-214-4140-8 .

Remaining Publications of the Author Relevant to the Thesis

[A.13] P. Vít "Design Methodology for Dependable Modular Systems". In *Ph.D. Minimum Thesis, Faculty of Information Technology*, Prague, Czech Republic: 2017,

[A.14] Pavel Vít, Jaroslav Borecký "Nástroj pro generování zabezpečovacího zařízení pro železnici". In *Diploma thesis, Faculty of Electrical Engineering*, Prague, Czech Republic: June 2010,

Supervised Publications

[A.15] Pavlištík, L., Vit P. "Periferní komunikace zabezpečovacího zařízení pro železnici". In *Diploma thesis, Faculty of Electrical Engineering*, Prague, Czech Republic: 2011,

[A.16] Kulovaný, J., Vit P. "Gracký editor zabezpečovacího zařízení pro železnici". In *Bachelor thesis, Faculty of Electrical Engineering*, Prague, Czech Republic: 2011,