**FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE**

Reliable FPGA Architecture

by

Jan Pospíšil

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
Department of Digital Design

August 2018

Supervisor:
    doc. Ing. Jan Schmidt, Ph.D.
    Department of Digital Design
    Faculty of Information Technology
    Czech Technical University in Prague
    Thákurova 9
    160 00 Prague 6
    Czech Republic

Abstract and contributions

Different types of Field Programmable Gate Arrays (FPGA) are used in many different fields of electronics. The most prevalent type is SRAM-based, which uses static RAM cells to store its configuration. The inherent drawback of this technology is its susceptibility to Single Event Effects. The Single Event Upset is the main concern, which can result not only in corrupted data being processed, but also in a major change to the design function and connections. Mitigation techniques are known to handle this issue, but their impact evaluation is not always easy. The actual impact to the reliability of a given design needs to be evaluated taking into account not only changes made to the design on the Register-Transfer Level, but also the actual implementation of the design on a given FPGA.

In our work, the main focus is on the FPGA architecture and its reliability in terms of radiation induced soft errors. We provide an overview of all the background needed to successfully handle this issue in this thesis. Later, an overview of the related works dealing with the similar topics and also connected research are presented. The method for a simulation-based evaluation of radiation induced soft errors in the SRAM-based FPGA configuration memory is proposed, an example implementation of this method on a chosen FPGA family is described, and individual steps are explained. Results of this example implementation on a set of benchmarks are presented and discussed.

In particular, the main contributions of the dissertation thesis are as follows:

1. The method for a simulation-based evaluation of radiation induced soft errors in the SRAM-based FPGA configuration memory based on parameters obtained from experiments on the real hardware is proposed.

2. The proof-of-concept toolchain for the chosen FPGA family model creation and usage through the simulation following the proposed method is implemented.

3. Individual implementation steps and intermediate data are described and test results obtained from a set of benchmarks are presented.

Keywords: FPGA, single event upset, simulation, fault model, XDL, RapidSmith

As a collaborator of Jan Pospíšil and a co-author of his papers, I agree with Jan Pospíšil's authorship of the research results, as stated in this dissertation thesis.

....
Tomáš Vaňát

Acknowledgements

First of all, I would like to express my gratitude to my dissertation thesis supervisor Jan Schmidt, to Petr Fišer and to Hana Kubátová for all the guidance and assistance during my studies and the thesis preparation.

I would also like to thank to Jozef Ferencei, Filip Křížek, and the team from the Department of Accelerators of the Nuclear Physics Institute, CAS at Řež for a possibility to join their research environment and to work on the cyclotron U-120M accelerator there, which added a practical point of view to my work. I also appreciate all the valuable insights into a physical side of radiation effects in silicon devices the provided to me.

My thanks go also to my colleague Tomáš Vaňát, with whom I did most of the measurements connected with my research topic.

Special thanks go to the staff of 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 members, for their infinite patience and care.

Dedication

To my family.

Contents

List of Figures

List of Tables

List of Listings

Abbreviations

ALT   Accelerated Life Testing

ARM   Advanced/Acorn RISC Machine

ASCII   American Standard Code for Information Interchange

ASIC   Application-Specific Integrated Circuit

ATPG   Automatic Test Pattern Generation

BJT   Bipolar Junction Transistor

BRAM   Block Random Access Memory

CAD   Computer-Aided Design

CAS   Czech Academy of Sciences

CCGA   Ceramic Column Grid Array

CCLG   Ceramic Chip Carrier Land Grid

CLB   Configurable Logic Block

CMOS   Complementary Metal Oxide Semiconductor Field Effect Transistor

CPLD   Complex Programmable Logic Device

CPU   Central Processing Unit

CQFP   Ceramic Quad Flat Package

CRAM   Configuration Random Access Memory

CRC   Cyclic Redundancy Check

CSIC   Configuration Self Integrity Check

CTU   Czech Technical University in Prague

DCI	Digitally Controlled Impedance
DRAM	Dynamic Random Access Memory
DSP	Digital Signal Processing
ECC	Error Checking and Correction
EDA	Electronic Design Automation
EDAC	Error Detection and Correction
EMP	Electromagnetic Pulse
EPP	Extensible Processing Platform
ESA	European Space Agency
FET	Field Effect Transistor
FIT	Faculty of Information Technology
FIT	Failures in Time
FPGA	Field Programmable Gate Array
GCR	Galactic Cosmic Rays
GEO	Geostationary Earth Orbit
GSR	Global Set/Reset
HDFF	Single Event Upset Hardened Flip-Flop
HDL	Hardware Description Language
IBM	International Business Machines Corporation
IC	Integrated Circuit
IOB	Input-Output Block
IP	Intellectual Property
ITS	Inner Tracking System
JTAG	Joint Test Action Group
JVM	Java Virtual Machine
LB	Logic Block
LEO	Low Earth Orbit
LET	Linear Energy Transfer
LGA	Land Grid Array

LUT	Look-Up Table
LVCMOS	Low Voltage CMOS
MAC	Multiply and Accumulate
MBU	Multi Bit Upset
MCGA	Multi-Layer Column Grid Array
MEU	Multi Event Upset
MIPS	Millions of Instructions Per Second
MOS	Metal Oxide Semiconductor
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
MQFP	Multilayer Quad Flat Pack
MTBFI	Mean Time between Functional Interrupt
MTTF	Mean Time to Failure
MUX	Multiplexer
NA	Not Available
NASA	National Aeronautics and Space Administration
NASDAQ	National Association of Securities Dealers Automated Quotations
NCD	Native Circuit Description
NPI	Nuclear Physics Institute
OTP	One Time Programmable
PIP	Programmable Interconnect Point
RAM	Random Access Memory
RHB	Radiation Hard Breakdown
RHBD	Radiation Hardening by Design
RILC	Radiation-Induced Leakage Current
RISC	Reduced Instruction Set Computer
RSB	Radiation Induced Soft Breakdown
RTL	Register-Transfer Level
RVC	Redundancy Voting Circuit
SAN	Stochastic Activity Networks

SEB	Single Event Burnout
SECDEC	Single Error Correction, Double Error Detection
SEE	Single Event Effect
SEFI	Single Event Functional Interrupt
SEGR	Single Event Gate Rupture
SEL	Single Event Latch-Up
SER	Soft-Error Rate
SESB	Single Event Snap-Back
SET	Single Event Transients
SEU	Single Event Upset
SOI	Silicon on Insulator
SPARC	Scalable Processor Architecture
SPICE	Simulation Program with Integrated Circuit Emphasis
SRAM	Static Random Access Memory
SW	Switch Box
TID	Total Ionizing Dose
TMR	Triple Module Redundancy
TSMC	Taiwan Semiconductor Manufacturing Company
UMC	United Microelectronics Corporation
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
VLSI	Very-Large-Scale integration
VPR	Versatile Place and Route
VTR	Verilog to Routing
XDL	Xilinx Design Language

CHAPTER 1

Introduction

The digital device known as Field Programmable Gate Array (FPGA) combines a reasonable computational power together with an ability to be reprogrammed. Its functionality (whole or partial) can be changed almost whenever needed, even on-line (i.e. when the circuit is under operation). The importance and use of these devices are growing and FPGAs are used not only in a "normal" environment for "common" applications. Advantages of these devices are utilized even in failure critical missions in hostile environments like in a military application or autonomous probes exploring outer space.

1.1   Motivation

FPGAs are used in many different fields of electronics. Most of FPGAs being used are SRAM-based, i.e. those using static RAM elements to store the configuration. The manufacture technology of SRAM-based FPGAs is relatively simpler than other types (e.g. flash-based or anti-fuse). This fact allows latest technology nodes to be used for manufacturing cutting-edge FPGA chips.

An inherent drawback of SRAM-based devices (not only FPGAs) is their susceptibility to Single Events Effects (SEE) generated in silicon structures by energetic charged particles passing through. SEEs do not pose a major risk for consumer electronics on the ground level, but they start to be a problem in a radiation harsh environment (e.g. a communication satellite in outer space) or in highly reliable systems (e.g. an autonomous driving unit). Latest technology SRAM-based FPGAs can be used in those critical applications only if the risk of SEE is properly mitigated.

SEEs are usually divided into two categories for CMOS devices: temporary effects and permanent (destructive) effects. The first are mainly Single Event Upsets (SEU, in other words a "bit flip"), where a value stored in the memory element is changed by a charge deposited in a circuit silicon substrate by a charged high energy particle striking the device. They are considered temporary as they do not modify the silicon device in any way and they subside once new data are written to the memory element. The latter

category contains defects like Single Event Latch-Up (SEL) or Single Event Gate Rupture (SEGR) which modifies (usually in a destructive way) the integrated circuit itself.

In an FPGA, a third category of semi-permanent defects exists. Because the actual circuit implemented in the FPGA is encoded in configuration bits stored in memory elements, the SEU defect can has a more permanent effect. The configuration memory is written only once on the device power-up and kept unchanged during a normal operation. When a bit flip occurs in the configuration memory, a defect can appear changing the implemented design, but not the FPGA itself. So the actual sensitivity of a digital design to SEUs depends not only on the given design, which is implemented on the FPGA, but also it depends on the actual implementation, i.e. which configuration bit is used to encode which function or connection of the implemented design.

On one hand, FPGA devices utilize all merits of being integrated circuits of Very-Large-Scale Integration (VLSI), but on the other hand, their proper function can be quite easily influenced by radiation. As stated in section 2.6 an error rate induced by a radiation can exceeded a non-radiation error rate by several orders of magnitude. The more critical a mission is, the more effort has to be spent on minimizing this risk. There is a need for reliable and dependable designs, whose characteristics need to be fully quantified to enable us to use all of its potential. To achieve this, tools and platforms for modeling, testing and characterizing further designs are needed. Even more, new FPGA architectures possibly built on top of this knowledge can provide designers with even a more reliable platform.

1.2   Problem Statement

Main structure used in an FPGA for proper operation is a memory element. The actual configuration of the FPGA (thus its actual function) is stored within an FPGA in a configuration memory (as described in section 2.3). If device is manufactured in the CMOS process and memory elements are of an SRAM type, the whole device is prone to a radiation and bit errors can be introduced into the configuration memory (see section 2.6.2 for more explanation). These changes can influence the device usability, changing a function or even a structure of the implemented design. Moreover, it can influence its dependability and endanger the system in which it is used.

Designer can fight above described failures. Several mitigation techniques are available to minimize the SEU impact in digital designs in general (e.g. redundancy like the Triple Modular Redundancy (TMR), self-checking, self-repairing via reconfiguration, or self-repairing codes). Those can be also used for FPGA digital designs, but actual impact of the chosen mitigation technique is locally unpredictable and there are no quantitative characteristics of this phenomena on actual devices in use, so it needs to be evaluated for a given implementation. This means, that the evaluation has to be done every time the design itself is changed, but also when a new implementation is "compiled".

There are several methods how to evaluate the SEU sensitivity of a given FPGA design. Most straightforward and realistic is an irradiation testing for the actual pair of a device and a design. The design is loaded into the FPGA on a target board and the whole setup

is exposed either to the expected environment or to an environment predictably worse (Accelerated Life Testing (ALT)) [B.1]. For this method, an operative electronic board with the given FPGA is needed, so it cannot be conducted in an early development stage. Also a special radiation source (e.g. a particle accelerator) is needed to conduct the test, which can be expensive and not readily available.

Another approach is a fault injection, where artificial bit flips are introduced into FPGA configuration by, for example, on-line modification of the configuration memory or by off-line modification of a bitstream which is then loaded into the FPGA and the test is conducted [1, 2]. The fault injecting method needs at least a basic knowledge about the internal organization of the FPGA bitstream and the configuration memory and the test itself is usually time consuming. Same as for the irradiation method, an operative electronic board with the given FPGA is need, often modified to facilitate repetitive configuration memory manipulations. But both of these techniques are unusable in a common design flow for their computational complexity, specificity and expensiveness.

Another possibility is a deep-level simulation of the implemented design on a level of transistor implementation in the CMOS technology. Prerequisites for this type of simulation are both exact knowledge of the FPGA architecture down to a transistor level and also an individual transistor radiation sensitivities. Those information are usually not available outside the FPGA manufacture which render making general models very difficult, not even impossible. Also, such simulation would be computationally very demanding.

In this thesis, we discuss the compromise between mentioned characterization methods, a combination of an ALT and a computer simulation in sort of trade-off. The FPGA device as a platform is described by a higher-level simulation model which is calibrated by a partial ALT. As a result, we will be provided with a general model of an FPGA with resolution of basic FPGA building blocks (primitives). This model will be able to provide analysis of virtually any design for predicting the SEU sensitivity. This tool can be used as a regular part of a design flow to characterize the designers output, but it can also be used for a research on advanced FPGA architectures with an emphasis to a SEU resistivity.

1.3 Goals of the Dissertation Thesis

The main goals of the dissertation thesis are as follows:

1. To study the selected FPGA architecture and both commercial and open source tools and other resources usable for its analysis.

2. To proposed a method for a simulation-based evaluation of radiation induced soft errors in the SRAM-based FPGA configuration memory based on parameters obtained from experiments on the real hardware.

3. To implemented a proof-of-concept toolchain for the chosen FPGA family model creation and usage through the simulation following the proposed method.

4. To tested this implementation on benchmarks.

Theoretical Background

A theoretical background in all major aspects with respect to this thesis is given in this chapter. The technology of digital integrated circuits and of Field Programmable Gate Arrays in particular is addressed. A connection between these integrated circuits and Single Event Effects threatening them in a harsh environment of ionizing radiation is shown. Also a particle accelerator facility used for irradiation tests which results are used in this work is mentioned.

2.1   CMOS Technology

Complementary MOS (CMOS) technology is process nowadays widely used for fabrication analog and digital integrated circuited. It used phenomenon of Field Effect Transistor (FET) and in general it is a descendant of a bipolar technology, which uses Bipolar Junction Transistors (BJT).

Basic element of CMOS is Metal-Oxide-Semiconductor field effect transistor (MOS-FET, or simply MOS). By simple description, a current flowing through this transistor is directly controlled by the voltage applied in the transistor structure. This difference from BJT (where main current is controlled by current too) offers lower power consumption (because of lack of controlling current) and higher achievable frequencies, because parasitic capacitances in the control structure doesn't need to be (de)charged by the control current.

According to a polarity of major (and only) carriers in the MOS a pMOS and an nMOS type of transistor can be distinguished (with holes and electrons as carriers respectively). The pMOS transistor utilizes n-type semiconductor substrate and nMOS utilizes p-type semiconductor substrate as a conducting path.

In typical CMOS both mentioned types of transistors are used in symmetry. This helps to creation of digital circuits in this technology as all basic logical elements (inverters, AND, OR, NAND gates and more) can be easily assembled from the same type of nMOS and pMOS transistors.

2.2 Digital Circuits

Digital circuits are composed from two types of structures: combinational and sequential. For purpose of this work term "digital circuit" refers to an integrated circuit intended to process discreet logical signals.

Combinational structures represents part of a digital circuit able to calculate logical functions. This behavior is achieved by logical gates connected to each other by conductors.

Sequential structures can be easily described as memory elements. They can hold a value for a defined period of time and are (usually) operating with rhythm of clock, i.e. one or more signals synchronizing several sequential structures together. Primary type of memory element used in a CMOS technology is **Static RAM** (SRAM), because all its structures can be manufactured in basic CMOS. Six CMOS transistors comprise single SRAM memory cell and one bit of information is here stored in closed feedback of two logical inverters. When SRAM is not enough, further technologies have to be combined together.

All above mentioned structures can be built of pMOS and nMOS transistors, which are described below.

2.3 Field Programmable Gate Arrays

As to a Field Programmable Gate Array (FPGA), we refer to a programmable digital integrated circuit of an island type, where "islands" of logic, memory elements and input/outputs blocks are hierarchically connected by a "sea" of interconnection. Actual way how both logic and interconnect is implemented differs between FPGA manufactures, but also between individual product families or series of those chips. By term "FPGA architecture", we understand one selected implementation type of a selected FPGA family or series from a selected FPGA manufacture. A brief introduction to FPGA circuits will be provided in this section.

2.3.1 FPGA Architecture

The FPGA is a complex type of programmable device, successor of CPLD. It features various resources, which can be connected together to form nearly inexhaustible amount of various digital circuits. Example of a basic FPGA architecture can be seen in Figure 2.1 where an upper right corner of device is displayed. Resources used in FPGA can be divided into four main categories: logical blocks, an interconnection matrix, supporting circuits and hard-coded parts.

Logical blocks are the heart of this programmable chip. In most common FPGA architectures, they consists mainly of one to several function generators (usually Look-Up Tables (LUT), one to several memory elements (usually D flip-flops) and a network of multiplexers and switches constituting a local interconnection (see Figure 2.2 for example).
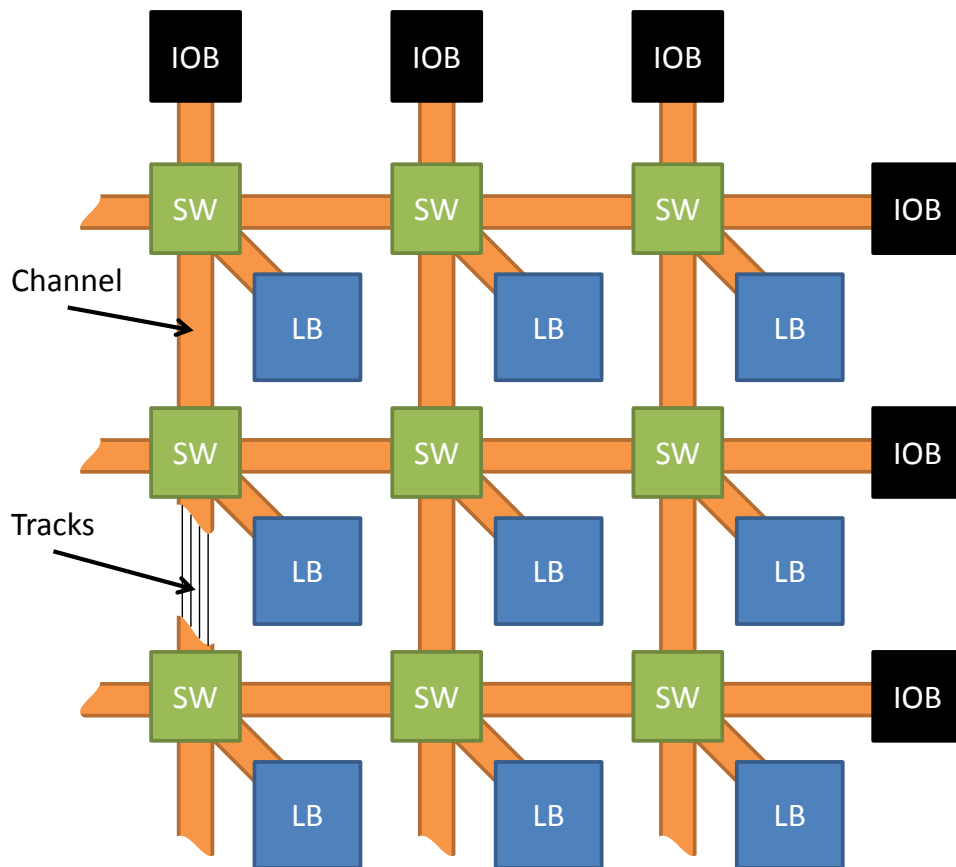
Figure 2.1: The example of an upper right part of an island type FPGA. LB denotes Logic Island (or Logic Block), SW denotes Switch Box and IOB denotes Input/Output Block. Individual elements are connected by the sea of interconnect, which is composed of individual metal wires.

Target design is cut up to smaller pieces and fitted into these logical blocks at the process of transforming a written code into a configuration of the real chip.

These parts in individual logical blocks are afterwards connected back together by programmable **interconnection matrix**, or a "sea of interconnection". This routing structure can have several hierarchies of interconnection points (referred to as "switch boxes") connected between each other and islands by metal wires.

Supporting parts of the chip include for example power circuits, blocks for digital clock synthesis, programming and testing interfaces (e.g. JTAG) or input/output blocks which are responsible for connecting the chip to its surroundings (labeled as "IOB" in Figure 2.1). The last category, **hard-coded parts**, is more often included only in bigger and complicated FPGAs. A wide range of such blocks are going from dedicated memory blocks to complicated communication and computational circuits.

High configurability of FPGAs is only possible by extensive memory use. Every possible
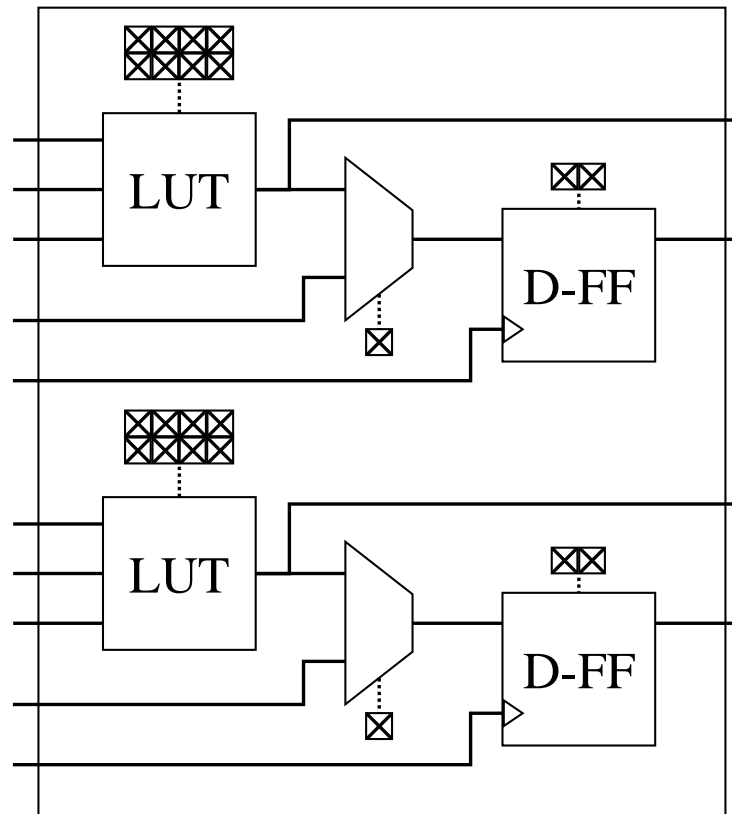
Figure 2.2: The example of an FPGA logic block (island) containing two 3-input LUTs, two D flip-flops and two multiplexers for variable internal connection. Example configuration bits are indicated as crossed boxes.

function and connection has to be defined before FPGA can operate and all these definitions are stored in memory scattered around the whole chip. Logical functions encoded into LBs are represented by Look-Up Tables (LUT) which are represented by memories and all other parts are programmed in a similar way. Every interconnection is also encoded into configuration bits stores beside every interconnection node.

And there are plenty of those connections, because all previously mentioned parts are hierarchically connected. Individual logical elements are connected together inside LB, neighboring LBs are connected together by local interconnects and all LBs with special blocks are connected together with global interconnects. This global interconnects comprise data channels (each of them containing several tracks, actual signal paths, of a different length) connected together in **switch boxes** (see Figure 2.1). Even more than 90 % of all configuration memory can be used for interconnection configuration (but not all of this memory is used in average design). Because of this huge amount of interconnecting, SEE in configuration memory can cause not only functional change, but also a structural change.

FPGA devices can be divided according to the configuration technology they are built

on. The essence of the FPGA is a programmability of its functions. Every logical block, all hard-coded blocks and every single switch in an interconnection matrix need some memory to store their configuration. Most widespread technology is **SRAM**, which offers true re-programmability and uses only standard CMOS process. On the other hand the SRAM cells consume more power in comparison with other technologies and because of its volatile character they need an external storage for a configuration bitstream. In spite of this the SRAM is the most used technology for FPGA configuration memories and it is expected to be so in the future [3]. Another memory technologies used in FPGA devices are **flash** (which need extra process to be manufactured) and **antifuses** (which are One-Time-Programmable (OTP).

A content of a configuration memory is essential for the proper function of the device. An accidental change of the configuration memory can alter the logical function of an implemented circuit, change the routing connection between individual logical blocks (inside the logical block or globally) or modify the parameters of the chip interface pins or hardcoded blocks. With configuration memory of several tens to hundreds megabits with the latest FPGAs the probability of SEE is high enough to pose a significant threat in reliable-critical applications.

2.3.2   FPGA Structures

In this part several main FPGA programmable structures will be introduced. What is inside FPGA, how it can be programmed and mostly how it can misbehave in case of radiation disturbance is important to the future explanation.

In the category of **logical block structures** we can find mainly **Look-up tables (LUT) and user memory elements**. A LUT is a structure capable of representing logical structure. Usually a LUT has n_i inputs (three to eight) and one output and it can represent any logical function of n_i variables. It is build up from $n_i \times 1$ bit memory where n_i inputs is used as an address to this memory and data output is output from this LUT. A LUT memory is written only in time of device configuration by way of bitstream. User memory elements are usually represented by configurable D flip-flops (where for example type of reset, active clock edge or reset state can be configured). Beside these two primary structures logic blocks contain few more types of supporting elements like **multiplexors** for inter-block connection or **some basic logic elements** helping to implement come more complex design parts like carry chains.

The category of **interconnecting structures** is more varied and more mysterious. Basic structures involved inside logical blocks are (at least in general) described in device datasheet, but about real implementation of interconnection system only a few information can be found. Few interconnecting structures can be named: **pass-transistors**, which can be considered simply as bi-directional switches between two segments of wires, **active tri-state buffers** which has the same function as pass-transistors, but only works in one direction, or **multiplexers**, more complex structures to provide interconnection between more than two wire segments. Besides this more various structures (like simple **buffers**) can be found in interconnecting matrix. All these structures are combined together to

create bigger structural part named switch boxes. In these structures channels (composed of individual tracks) are connected in several possible patterns (e.g. universal, disjoint, Wilton or Imran, see Figure 2.3).
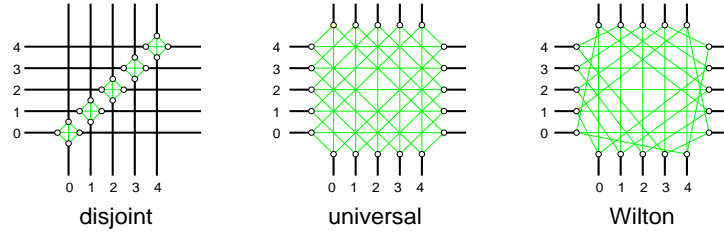
Figure 2.3: Examples of different connection patterns used in switch boxes (interconnection blocks) inside FPGA chips. (Source: [4])

2.4 Ionizing Radiation

In this section we will bring up a short overview of radiation in connection with defects in CMOS devices. Several information stated in this section has been adopted from [5].

Radiation in its general definition is transfer of energetic particles (or energy itself) through vacuum or some matter media. Basic division of radiation types are to ionizing and non-ionizing. Here we should introduced term **ionization**, which is a process when one or more electrons are liberated from atom or molecule by the collision of particle or by energy transferred into the system. It is quite similar process to **excitation**, when electrons are only shifted to the more energetic state (receiving an energy), but ionization requires more energy than excitation.

Now we can define **ionizing radiation** as a radiation with energy high enough to introduce ionization in the material through which it traverse. This ionization can be caused directly by energy of radiation (e.g. high energetic proton can tear off electron directly) or it can be done secondary. **Secondary ionization** is caused by daughter products of interaction between radiation and matter (e.g. high energetic neutron can interact with atom nuclei and through nuclear reaction emit α particle which can later cause ionization in the matter). Radiation of charged particles (such as electrons, protons or ions) causes primary ionization, while radiation of uncharged particles (like photons or neutrons) can cause secondary radiation.

2.4.1 Units

Single particle from a highly energetic radiation can cause more ionization in the material. Charge deposit in a material this way is spread along a particle path and amount of this charge in individual positions is in relation of how a particle is slowed down by the

interaction with material. This characteristic is expressed by **Linear Energy Transfer** (LET), sometimes called **Linear Stopping Power**. LET can be expressed as

$$L = \frac{\mathrm{d}E}{\mathrm{d}l} \left[\mathrm{J\,m^{-1}, eV\,m^{-1}, keV\,\mu m^{-1}} \right] \tag{2.1}$$

and it represents amount of energy $\mathrm{d}E$ deposited over given distance $\mathrm{d}l$ and its actual value depends on radiation and material types. In electronic LET is usually expressed as

$$L = \frac{1}{\rho} \frac{\mathrm{d}E}{\mathrm{d}l} \left[\mathrm{MeV\,cm^2\,mg^{-1}} \right] \tag{2.2}$$

where ρ is material density. Silicon (Si) is (in the most cases) the material to be considered in this relation. Equation 2.2 can be referred to as Mass Stopping Power too. Because energy is considered to be transferred by charged particles, LET is not used for characterizing neutron radiation.

The particle **fluence** Φ is a number of particles N passing per unit area a.

$$\Phi = \frac{\mathrm{d}N}{\mathrm{d}a} \left[\mathrm{cm^{-2}} \right] \tag{2.3}$$

The change of fluence in time is called **flux** F, expressing number of particles passing unit area per unit time.

$$F = \frac{\mathrm{d}\Phi}{\mathrm{d}t} \left[\mathrm{cm^{-2}\,s^{-1}} \right] \tag{2.4}$$

A more comprehensive overview of units and relations in this field can be found in [5].

Minimal amount of charge deposited in an actual CMOS structure which triggers a radiation defect is called **critical charge** Q_{crit}. Charge smaller than this level can be safely drained out of the structure or it can deepen the degradation of a material. In the latter case not only momentary energy deposited in the material is important, but also overall dose gradually deposited in a longer time. This accumulated amount of dose is called **Total Ionizing Dose** (TID).

The term **cross-section** in radiation related texts determines a probability of an interaction between two particles, or between a particle and matter. It is usually denoted as σ and it depends on the type and energy of an interacting particle, and also on the character of the target material. In this text, we will refer to it as either a **device cross-section** σ_{device} or a **bit cross-section** σ. The device cross-section is a sensitivity of the whole device and can be calculated from a number of SEEs ε caused by a fluence Φ:

$$\sigma_{device} = \frac{\varepsilon}{\Phi} \left[\mathrm{cm^2} \right] \tag{2.5}$$

The bit cross-section is rather a characteristic of a single bit, so it is more appropriate for a memory chips. It can be calculated as a number of SEEs ε caused by a fluence Φ per individual bit:

$$\sigma = \frac{\varepsilon}{\Phi \cdot n} \left[\mathrm{cm^2\,bit^{-1}} \right] \tag{2.6}$$

The bit cross-section can also be used to characterize a programmable integrated circuit, like an FPGA, when referring to its configuration memory or user registers.

2.4.2 Sources of Ionizing Radiation

The radiation always originates from a source from which the energy is radiated to its surroundings. Different types of sources produce different type of radiation with specific parameters and different ability to induced defects in CMOS. Basic classification can be to natural and artificial radiation.

The **artificial radiation** is generally less harming in comparison with some types of natural radiation. The only real risk lies around high energy facilities for experiments, power generation or sensing devices (used in a medical facility or in an industry). These sources feature higher probability of radiation risk even for short period of exposition, so usually TID is not a main concern in designing devices into these environments.

Another type of radiation source is a nuclear explosion, which is characterized by burst of gamma-ray followed by burst of neutrons, accompanied by Electromagnetic Pulse (EMP) generating high voltages and currents in all conductive materials.

In a case of a **natural radiation** the main risk lies outside of our atmosphere. The terrestrial, underground and air radiation is (in the most cases) not strong enough to make any serious damage to the CMOS chips. To the contrary the outer space is much more hostile environment in this aspect. Event thought radiation under our protecting atmosphere have to be considered in critical applications.

Space radiation characteristics in the vicinity of Earth depend largely on Sun activity and on the place it is considered for, because the Earth has several defense mechanisms to shield its content against this radiation. Main attackers from outer space environment are high energy protons and heavy ions. This type of radiation is called **Galactic Cosmic Rays** (GCR) and consists of $85\,\%$ of protons, $13\,\%$ of α particles and about $2\,\%$ of heavy ions (from H to Ni). Another source of radiation is the Sun, as solar flares consist mainly from high energy protons and heavy ions.

Because of shielding ability of the Earth, most of this radiation is shielded away on its surface. Part of it is deflected by Earth's magnetic field and part is trapped in so called **Van Allen Belts** (see Figure 2.4). These three belts have the following characteristic:

- Inner belt: $1600\,\mathrm{km}$ to $12\,900\,\mathrm{km}$ from Earth center, contains electrons and positive ions (mostly protons)

- Outer belt: $19\,000\,\mathrm{km}$ to $40\,000\,\mathrm{km}$ from Earth center, contains electrons with energy of few MeV

- Middle belt: situated between two previously mentioned, with electrons of energy about $4\,\mathrm{MeV}$ to $7.5\,\mathrm{MeV}$, observed for a first time at the end of 2012 [6]
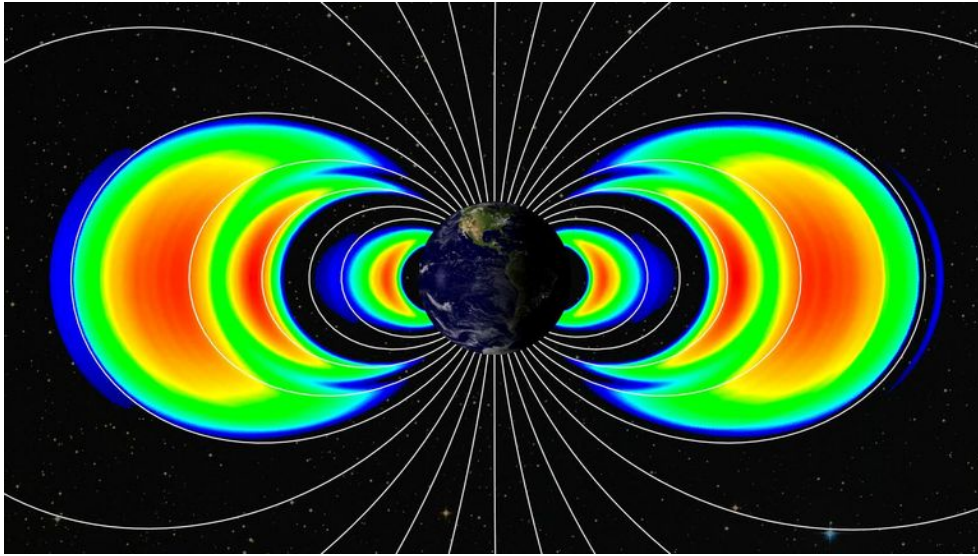
Figure 2.4: Visualization of three Van Allen Belts created using data from NASA Van Allen Probes. (Source: APL, NASA [7])

Activity in radiation belts are highly dependent on the Sun activity, solar flares bursts can increase it by a factor 10^3. These conditions have to be considered when space exploration mission leads through these belts.

Main source of **atmospheric radiation** are events outside of it. Beside natural radioisotopes, a broad spectrum of energy are received from the space, i.e. from the Sun and from the deeper space. These radiation penetrating the atmosphere are partially shielded, thus only about a 1% of it reach the sea level. As shielding are performed gradually through the whole depth of atmosphere, the intensity of radiation decrease also gradually. Side effect of this attenuation is secondary radiation. Particles of GCR interacting with particles of atmosphere create wide range of cascades of different particle types with wide range of spectrum (e.g. photons, e, hadrons or nuclei).

Most important threat of this "cosmic shower" are **neutrons**. This higher flux component of radiation has higher LET values and no charge. These both help neutrons to penetrate deeper into the atmosphere and into materials. Neutrons with energies from $1\,\text{MeV}$ up to $200\,\text{MeV}$ (or even more than GeV scale) interact in a matter via nuclear reaction and create secondary ionized particles. Examples of secondary products created by neutron interaction with silicon can be found in Table 2.1.

Not only high energy neutrons can cause trouble. **Low-energy neutrons** with energy below $1\,\text{MeV}$ cannot usually produce ions with energy high enough to disrupt device function, but this doesn't apply for reactions with Boron or Gadolinium. These two elements has extremely large cross-section with thermal neutrons with energy even less than $1\,\text{eV}$. Interacting with such a neutron high energetic ions (up to MeV) can be created. [9] For

Reaction product	Threshold energy [eV]
^{25}Mg + α	2.75
^{28}Al + p	4.0
^{27}Al + d	9.7
^{24}Mg + n + α	10.34
^{27}Al + n + p	12.0
^{26}Mg + ^{3}He	12.58
^{21}Ne + 2 α	12.99

Table 2.1: Reaction overview for the n + ^{28}Si interaction. (Adapted from [8])

example

$$^{10}\text{B} + \text{n} \longrightarrow {^7\text{Li}} + \alpha \tag{2.7}$$

with resulted energy of Lithium ion of $\approx$1 MeV and energy of α particle of 1.5 MeV. Both of the resultants can cause radiation induced defect in silicon device, but active track of these particles is only about 0.5 µm, thus only thin layer of ^{10}B near silicon substrate poses threat.

Another type of particles created in atmosphere from GCR are **muons**. They are created via weak interaction as a decay product of pions created from protons collision with atmosphere particles. Lifetime of muons is 2.2 µs and are only secondary charged particles which can reach sea level (and even penetrate deeply underground). Muons interact extremely few with matter (excepted at low energies by direct ionization). Details on this type of radiation can be found in [10].

Most significant portion of GCR are **protons**. They interact with silicon as neutrons do, but at sea level protons are hundreds time less numerous, so their impact to electronics is negligible.

Radiation from natural isotopes (or derived by other ways from non-cosmic sources) can be called **ground radiation**. Here several different emitters exists emitting α or β particles and gamma-ray. Radiation from β and gamma emitters are not able to deposit enough energy to disrupt device operation, thus only α emitters present a reliability concern in microelectronics. These can be a naturally radioactive material or material that contains residual traces of radioactive impurities. [10]

The $\boldsymbol{\alpha}$ **particle** is a double ionized Helium (two neutrons and two protons). They are emitted from the nuclear decay (e.g. of uranium or thorium) and have a rather small penetration depth. For a 10 MeV α particle the penetration depth in silicon is less than 100 µm. Because of this, α particles from outside the package are not a concern. Nowadays techniques leading to highly purified materials are enough matured to minimize α emitters in manufacturing process.

More detailed description of each ionizing radiation environments can be found in [11].

2.5 Dependability Terms

The basic term is **defect**, which means an actual physical problem, a trouble caused by some unwanted event (e.g. an ionizing radiation in our case). An example can be a wire interrupted by the electro migration or (as in our case) an open MOS transistor as a consequence of a charged particle passage.

The logical model of a defect is called a **fault** and can be expressed (and calculated with) independently of a technology. In our case an open MOS transistor can be translated into a "short" fault if it is a part of a memory element which is responsible for controlling the pass-transistor on a signal path.

A fault can be propagated into an **error** which can be expressed as a change in data or change of a state of a digital design. When such error occurs in a controlling or otherwise critical part of a design a **failure** can occur, i.e. the situation in which the behavior of the digital design is wrong, possibly dangerous. Not every defect has to become an error and not every error has to produce failure.

For a quantitative description of above mentioned terms we will introduce a **Soft-Error Rate** (SER) and a **Failures in Time** (FIT). The SER expresses a rate of soft-errors (for soft-error see below) occurring in a unit time on average. The FIT represents a rate of failures in a unit time. The FIT of a device is often represented in one billion (10^9) of device hours in an operation.

2.6 Radiation Effects in CMOS

In this section we will present a basic overview of different mechanisms induced by high energy radiation in CMOS integrated circuits. Some information mentioned in this section are adopted from [9] and [8].

An accidental change of configuration memory bit without a reason is highly improbable. But there are reasons for such changes. One of possibilities is radiation. Accelerated charged particles can affect silicon structures by few different ways. It can alter memory state saved inside the device or it can influence signals travelling through the chip. According to the environment the device is used in and the probability of radiation induced event the device (or the whole system) has to be hardened. Methods of hardening in this case include wide range of changes from manufacturing technology modification, through changes in a device architecture and ending with modifications in the software level of the system.

CMOS structures are continually scaling to smaller and smaller size. Technology nodes used nowadays already moved from tens of nanometers to ones of nanometers. Besides positives, scaling brings negatives too and one of these negatives is increasing sensibility to radiation. Structures used shrinks down; also charge which is necessary to proper operation shrinks down. Nowadays charge required for operation is only about $1\,\mathrm{fC}$, which equals to only about 6242 charge carriers (electrons or holes). To generate one electron-hole pair energy of $3.6\,\mathrm{eV}$ needs to be deposited into a silicon, thus it is quite easy to disrupt device

operation by relatively small amount of external charge deposited by ion passing through a device, because individual particle can deposit from 1 to several hundreds of fC.

Most charge-sensitive part of CMOS device is reverse biased junction. In field of memory elements while SEE used to be a threat for Dynamic RAM (DRAM) in the past, it is not a main concern nowadays. Sensitivity of DRAM memory elements decreased in time due to change of DRAM fabrication technology form planar capacitors with large junction area to contemporary 3D capacitors. These new structures have significantly increased critical charge Q_{crit} which is needed to disrupt memory operation. On the other hand SRAM memories follows scaling sensitization and their radiation immunity gets lower and lower with shrinking dimensions, increasing voltage (i.e. voltage decreasing slowly than the scaling do) and decreasing Q_{crit}.

Still less sensitive than SRAM memory is logic made out of CMOS, but its sensitivity is increasing same way as for SRAM. For sub-micron nodes radiation sensitivity of logic starts reaching the level of SRAM sensitivity and are far above sensitivity of SRAM memories equipped with some type of automatic error detection and correction mechanisms. Because of increasing operation frequency and thus decreasing period time, even a short SET glitch can easily propagate through logic gates and can be finally latched.

Radiation in CMOS can leads to several phenomena from parametric shift to complete device failure.

2.6.1   Radiation Induced Defects Classification

Basic division of radiation induced effect can be to **permanent defects** which remains in the device for long time and are unrecoverable (or it is hard to recover them), **intermittent defects** which are present in the device for long time in a latent form and their consequences appear only time to time and **transient defects** which stands only to the first reset of a device or some type of correction (e.g. bitstream reprogramming).

Division described below was partially adopted from [12].

2.6.1.1   Permanent and Intermittent Defects

Permanent defects are results of material (or structure) degradation caused by high energy radiation, frequently degradation of dielectric. These defects can affect device parameters (static and dynamic) or even the device function. They are result of TID as their effect gradually rises with accumulating irradiation dose and they remain after irradiation and even after circuit is powered off or reprogrammed. These types of defects significantly reduce the operational lifetime of a device, which can be a concern for long term projects like space missions.

For example dielectric (typically in MOS transistor gate or capacitor isolation) can suffer from trapped charge generated by electron-hole decomposition ignited by radiation. Some of electron-hole pairs are recombined, but majority of the charge (mostly positive charge in SiO_2, but negative charge can be also trapped in some type of dielectric, as mentioned in [13]) remains in the structure and slowly changes the device parameters.

Example of this type of defect can be **Radiation-Induced Leakage Current** (RILC). This is a non-destructive trap-assisted tunneling current increasing oxide leakage current. Charged particles (X-rays, gamma-rays or electrons) creates electron-hole pairs along their path and due to physical properties of materials used in CMOS process this charge creates positive layer in the Si-SiO_2 interface region. This effect induced changes in device characteristics as a threshold voltage, carrier mobility, noise immunity or dynamic timing parameters. It can be also called surface effect [14].

The voltage shift caused by this effect is stated to be in an order of few millivolts at a dose of 1 Mrad or more. Same holds for a conventional SiO_2 and also for an alternative gate dielectrics [12]. Alternative gate dielectric discussed in this paper also exhibits excellent resistance to failure resulting from Single Event Gate Rupture (SEGR). According to authors, SEGR is not expected to be a significant concern for the deployment of high-κ[1] materials in space. Nevertheless, because ion exposure is known to affect the long term reliability of ultra-thin oxides, there is a possibility that space electronics with high-κ gate insulators could experience reduced operational lifetimes.

Another example can be **Radiation Induced Soft Breakdown** (RSB) which increases oxide leakage current, thus is significant problem in low-power application as it increases basal power consumption.

2.6.1.2   Transient Defects

Transient defects are sort of temporary effects which can completely fade out after end of irradiation or when a circuit is powered down or reprogrammed. Commonly they are called **Single Event Effects** (SEE). They are caused by high energy protons, neutrons, α-particles or heavy ions. When high energy particle strikes the CMOS circuit, dense plasma of electron-hole pairs is generated along particle path (as described below). In case this path goes through sensitive part of the CMOS structure, several type of SEE can be initiated. SEEs are primary concern of all radiation induced defects for terrestrial commercial devices. First SEE have been observed and described in 1975 [15]. It was inside an electronic on the board of telecommunication satellite (even though it was not in CMOS yet). SEEs are called to be volume effects.

According to [8] way of generation SEE can be described as follows: Charged particle (heavy ion, α-particle, low energy proton, low energy muon and a secondary particle produced by neutron) deposit electrical charge along their path in sensitive volume. Three successive steps follow as seen in Figure 2.5: a) the charge is deposited along particle path with sub-micron radius and high carrier concentration; b) carriers quickly drift towards depletion region, high current and voltage transients are presented and funnel shape distortion of potential extending depletion region (in the direction of originating particle path), duration of this step is about 1 ns; c) slower evanescence of carriers as they are collected, recombined or diffused away the junction region. The last step takes hundreds of ns. Current to time plot on the right of Figure 2.5 shows mentioned steps.

---

[1]High-κ here means high dielectric constant material substitution of SiO_2.
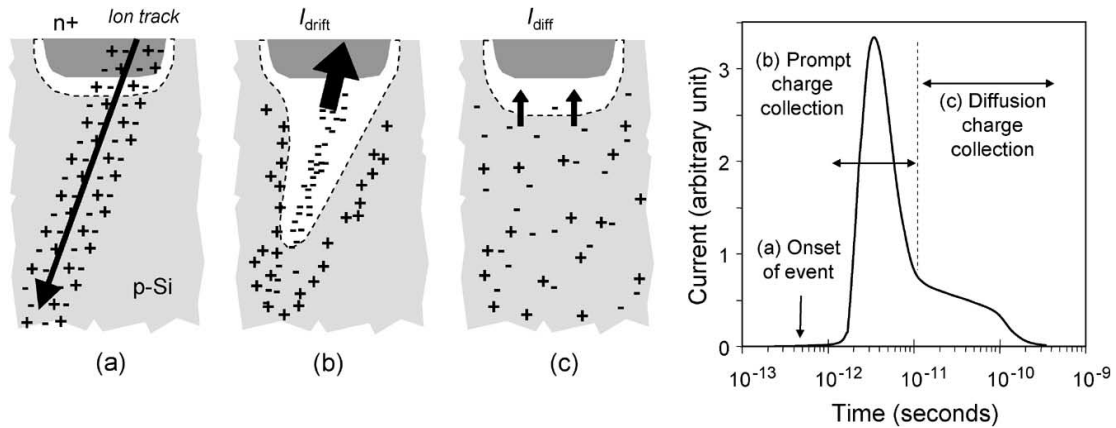
Figure 2.5: Different stages of the SEE in a CMOS device are shown. A charge is first generated by an energetic charged particle passing through a reverse-biased junction (frame a). This charge is then collected in two phases: a prompt one (frame b) and a slower diffusion phase (frame c). The shape of a resultant current pulse is shown in the last frame. (Source: [8])

SEE can be further divided into **soft-errors** and **hard-errors**. The only difference is in a type of a consequence such a SEE will cause. While soft-errors can cause no permanent damage (and thus their impact is correctable), hard-errors can lead to a permanent damage to the circuit.

Soft-Errors   While typical FIT for non-SEE failures can be 50 FIT to 200 FIT, SER can easily exceeded 50 000 FIT/chip (which is one error per 2 years).

Main soft-error is called **Single Event Upset** (SEU) and denotes memory change (or bit-flip) when logical value saved in a memory element is inverted. SEU is similar to **Single Event Transient** (SET) with the difference of a place where a charged particle strikes the system. In case of SEU the affected place is inside a (SRAM) memory element and this event causes not only temporary glitch, but lasting change of a stored value. This change of the logical value saved in a memory element can affect all structures of the FPGA which utilize a memory. When a SEU occurs inside the user memory, "only" the user data are affected, but the design keeps its function. The function of the design is not preserved in case of a configuration memory cell is hit. In case it is used for the design indeed. Logic function of the logical cell itself can be affected when configuration memory of logical block is struck or in case a memory of routing switch is strike, which has more global impact to the function of the device and can cause total loss of function. SEUs can affect supporting parts or hard-coded block as well with the same output.

The SET is a glitch caused by radiation. It only leads to a SEU if propagated through a circuit with the right timing (see Figure 2.6). It is a temporary change of the logical value in the combinational logic or on the connecting segments. When the induced peak is strong enough and come in the wrong time, it can cause unexpected behavior.
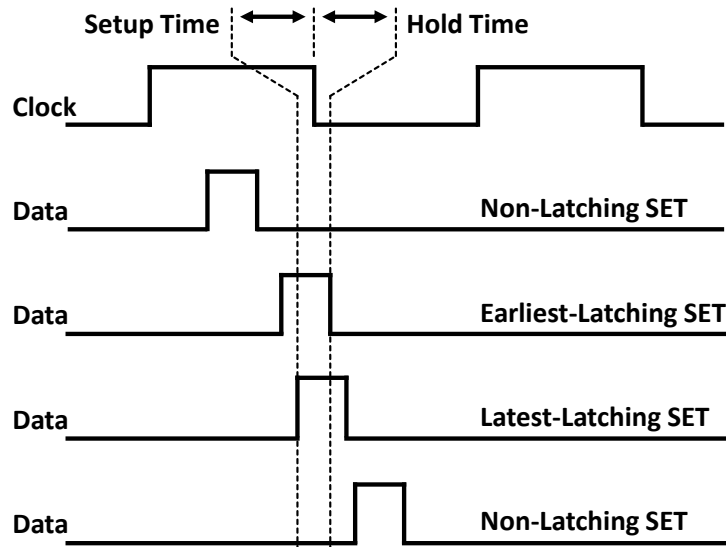
Figure 2.6: Different situations of relationship between a SET pulse arriving to a memory element and a clock signal driving the same element. The SET pulse is latched only if arrived in a certain period of time around the active clock edge. (adopted from [16])

When a critical part of the circuit is affected, **Single Events Function Interrupt** (SEFI) can occur. In this case, not only some data are affected, but rather whole functionality of the device is distorted.

Hard-Errors Most important hard-error in digital CMOS circuits is **Single Event Latch-Up** (SEL). In a CMOS pair of nMOS and pMOS transistor a parasitic bipolar structure can be found between well and substrate. When a SEL occurs a short circuit is introduced by this structure. It is a self-augmenting phenomenon and power cycle (a reset) has to be performed to get rid of this problem. It can be destructive, because the short circuit is usually created directly between power rail and a ground.

Another hard error is **Single Event Gate Rupture** (SEGR). This occurs mainly in condition of high electric field on dielectric (e.g. write or clear state in non-volatile memories). Temporary conductive path caused by high energy particle pass-through equals charges accumulated on the opposite sides of dielectric and if those charges are high enough, high current will flow causing high temperature and dielectric melt-down. A failure (i.e. a short circuit between the drain and the gate) can arise immediately or at a later time (with a continually rising gate current — RILC) [17].

Among hard-error several more phenomena can be classified. For example **Single Event Snap-Back** (SESB) in nMOS structures, **Radiation Hard Breakdown** (RHB) or **Single Event Burnout** (SEB) in power transistors. The SEB is characterized by a short between the drain and the gate, and the drain and the source caused by an avalanche breakdown of the parasitic NPN bipolar transistor inherent in the MOSFET structure [18].

19

2.6.2 Single Event Upset in FPGA

One of the phenomenon caused by radiation in a CMOS device is Single event upset (SEU). This event is started by charge deposited by ionizing radiation in a junction of CMOS transistor, which is a part of memory element structure. When a specific ionization is made in a specific time, the state of the memory element can be changed, i.e. the stored value (logical `one` or `zero`) is flipped (logically negated). Impacts of this bit-flip vary with memory element surroundings. When affected memory bit is a part of memory array, user data can be influenced. On the other hand when this memory element serves as a configuration for logic portion of a design or as a state holder for some processing unit, behavior of whole design can be affected.

In addition to SEU, Multi Event Upset (MEU) (also called Multi Bit Upset (MBU)) can be seen. This term usually refers to multiple events caused by one charged particle.

Single (and multiple) event effect has an important role in FPGA devices. Because FPGA utilizes a lot of memory elements not only for storing data, but mainly for storing configuration of the device, SEU has to be taken into account. FPGA devices need a new set of fault models than what is common for Application-Specific Integrated Circuits (ASIC), e.g. stuck-at, open, short... This difference originates in an additional hierarchy level added – the configuration memory which, when affected by SEU, can change the circuit definition. The actual fault model associated with the configuration memory SEU depends on a given FPGA architecture.

According to FPGA structures mentioned in section 2.3.2 following faults can appear in FPGA. In user memory primary manifestation of SEU can be observed causing a bit-flips in data, a temporary data corruption which can be mitigated for example by using TMR. In the configuration memory of logical blocks, SEU can alter function of an implemented design and thus generate rather complex fault usually impacting function and connection of the design being implemented in a given FPGA. For example, the $n-$input LUT is most often implemented as 2^n-bit memory where a truth table of the implemented function is stored. A SEU in this structure can change a value stored in one of those bits modifying the stored truth table, thus modifying the function being implemented. This fault then manifest itself only if appropriate LUT inputs are activated.

Which fault introduces a SEU in interconnection structures depends on actual type of element SEU appear in. One of the faults SEU can introduce in pass-transistors (a passive connection of two segments of wires) is **stuck-open**. This fault can occur when both wire segments are driven by the same signal and should be connected. When SEU alters controlling bit the signal is disconnected and its destination becomes un-driven (so called dangling). Another type of fault on this element is **short**. A short fault occurs when both segments are driven, but not with the same signal and thus don't have to be connected, and bit-flip causes pass-transistor to close the circuit of these two signals. Last type of fault on this passive connecting element is **antenna** which will be described later.

On active switches (as tri-state buffers) SEU can introduce same faults as on pass-transistor, but direction and extent is limited as signal can traverse active element only one way. Similar situation can occur on multiplexers used for signal routing. With no

buffer drivers introduced before multiplexers inputs short and antenna faults can appear. Moreover stuck-open and **signal exchange** (i.e. connecting another signal as a driver to a destination of an original signal) can be result of SEU even if buffers are introduced. Which fault can occur in which condition also depends on multiplexer encoding type (e.g. binary or one-hot). And even more complex fault can be created depending on the actual FPGA architecture, like complicated wired-AND or wired-OR function between several distinct nets [19, 20].

Antenna is special type of fault which is hardly to predict and even to detect. When an inactive segment is connected to an active one, significant delay can be introduced into signal path. Technology used is not ideal, thus resistance in form of open MOS transistors (up to several kΩ) and capacity of paths (up to hundreds fF) need to be taken into account when computing delay of routed design, because they form R-C elements. When SEU connects unused segment to active signal path, delay of this signal can be increased and (in extreme situation) timing of whole design can be affected. The state of unused elements in the routing matrix have to be considered too. While open switches connecting unused wires have no influence on the routed design, when a SEU connects such widely spread network of antennae to the active signal path, a timing can be greatly affected.

2.6.3 Radiation Impact Mitigation

There are several ways to improve the resistance of the CMOS device against radiation-induced SEE. The first step is made **in manufacturing process**. Right selection of technology together with proper modifications of technology and rules of layout can avoid some risks in general. For example presence of SEL can be avoided by breaking parasitic double BJT structure by isolating P and N complementary MOS from each other or by placing the chip on an insulator, a technology called Silicon on Insulator (SOI). Also good choice of materials used for manufacturing the devices, for packing or bonding is as important as their purification, which mainly eliminates α-emitters. As mentioned in [8], the flux of α particles nowadays reaches values of $0.001\,\mathrm{h}^{-1}\,\mathrm{cm}^{-2}$ which is equivalent to about one uranium or thorium part per 10 billion. This is five orders of magnitude less than in the beginning of CMOS technology and it helps to lower the α-induced SEE to around 20 % of all SEE. Also boron reduction can mitigate α particle as described in section 2.4.2. Atoms of ^{10}B near substrate can be completely replace by ^{11}B which has abundance about 80 % within boron and has nearly million time smaller cross-section and (with slow neutrons) produces gamma rays, which are much less damaging.

The next step is taken **in time of designing the FPGA architecture**. The way how memory blocks are connected and organized can for example avoid SEE in them, or at least it can help with later error recovery. The way how switches in interconnection matrix are arranged can lower the severity of routing error caused by SEE. Or the way how the configuration data are cared for can reduce the configuration SEE: a CRC in the configuration data, a bitstream scrubbing, etc.

With the FPGA on mind the next step is usually taken **in time of implementing a user design**. Everything what technology and architecture cannot offer has to be

implemented in user design. This includes CRC for the data being transferred, processed and stored, ECC for the memory modules, TMR for the functional blocks and so forth. In the time of development of FPGAs many of these features have been taken from the design implementation stage and incorporated into the FPGA architecture itself.

The last step is **device usage**, or installation. Where radiation cannot be eliminated completely, proper shielding have to be observed. Not all radiation can be shielded out (primarily high energy neutrons cannot be easily shielded as a 1 m thick concrete wall can shield just about 60 % of that radiation [21]), but its impact can be lowered for example by aluminum shielding on satellites or Lunar regolith in case of future Moon base (both the Lunar regolith and aluminum have by the way comparable shielding effects, as stated in [22]).

The last chance of getting really hardened system **when all of previously mentioned is not enough** is moving from pure basic SRAM FPGA to non-SRAM FPGA (flash or antifuses) or directly to an ASIC.

2.6.4 Accelerated Life Test

An Accelerated Life Test (ALT) is a method of quantitative characterizing. It is based on accelerating the life of a tested subject by intensification of its environment conditions. Tests can be conducted under higher pressure, elevated temperature or (as in our case) under heavier radiation. Using this method, a device gains the same amount of radiation as it would receive during years of operation, in a few minutes. If the appropriate energy spectrum of particles is used, then this method gives very accurate results. The problem is that it is very expensive and not easily accessible.

2.7 Cyclotron U-120M at NPI, Řež

The isochronous cyclotron U-120M is a particle accelerator operated by the Department of Accelerators of the Nuclear Physics Institute at Řež (see Figure 2.7) [23]. It is used for both research activities (nuclear astrophysics, a fast neutron flux generation, nuclear data measurements and irradiation campaigns to name few) and a medical radionuclides production. This cyclotron was originally commissioned in 1977 and several times upgraded until now [24, 25]. It can deliver an accelerated beam of protons with energy ranging from 6 to 37 MeV and a current up to 10^{14} proton cm^{-2} s^{-1}.

We had a possibility to use this facility within our collaboration with Tomáš Vaňát from CTU, FIT, Jozef Ferencei from NPI, CAS, and Filip Křížek from NPI, CAS. The primary objective of this collaboration was the radiation hardness studies for the ALICE Inner Tracking System (ITS) upgrade project, but also several irradiation campaigns link to the topic of this thesis were conducted [A.1, A.2].

Because the cyclotron facility was not used for a programmable circuit irradiation before, a new beam instrumentation and partially also beam control instruments had to be developed, as described in [26]. The nominal cyclotron output current is far too high for

Figure 2.7: A photo of the cyclotron U-120M at NPI, Řež. The main magnet core is visible in blue and its two pole windings are visible in red with a vacuum chamber in between. The vacuum chamber itself is not visible well, as the view is obstructed by a cyclotron technology and beam instrumentation equipment. An extraction beam pipe is visible on the left green support with several correction magnet mounted. (Source: [23])

SEU irradiation studies, so new methods of the cyclotron operation and also for measuring beam intensity and controlling its actual energy was developed. After this had been carried out, the cyclotron output proton flux can be lowered to $\approx 10^3 \, \text{proton} \, \text{cm}^{-2} \, \text{s}^{-1}$ which can be comfortably used to introduce SER in order of few events/s.

2.8   Xilinx Design Language

The Xilinx Design Language (XDL) is an ASCII variant of the Xilinx's proprietary Native Circuit Description (NCD) format used by the Xilinx ISE development toolchain [27, 28]. The XDL describes mapping of user logic to FPGA primitives, assignments of those primitives to physical locations in the FPGA chip (i.e. design placement) and also assignments of design nets (signals) to individual physical routing resources (i.e. routing of the design). As the lowest granularity of the XDL in terms of logic resources is FPGA primitive, it doesn't exactly describe the logic function itself. This function is easy to guess for some elements (e.g. LUTs, memory elements, multiplexers...), but quite hard to estimate for others (e.g. hard blocks of multipliers, PLLs or high speed transceivers).

Xilinx ISE toolchain offers a command line utility `xdl` which can perform bidirectional translation between XDL and NCD. The same utility also offers a possibility to export a primitive-level description of a given FPGA chip. The main usage foreseen of this feature was to facilitate user extensions of the official and otherwise quite closed FPGA design flow,

as shown in Figure 2.8. An FPGA design being processed by the Xilinx ISE toolchain can be exported to XDL in several various stages. Directly after the mapping stage, a mapped but unplaced and unrouted design netlist can be obtained. Another exports from place and route stage offer both placed or placed and fully routed design representation to be accessed.
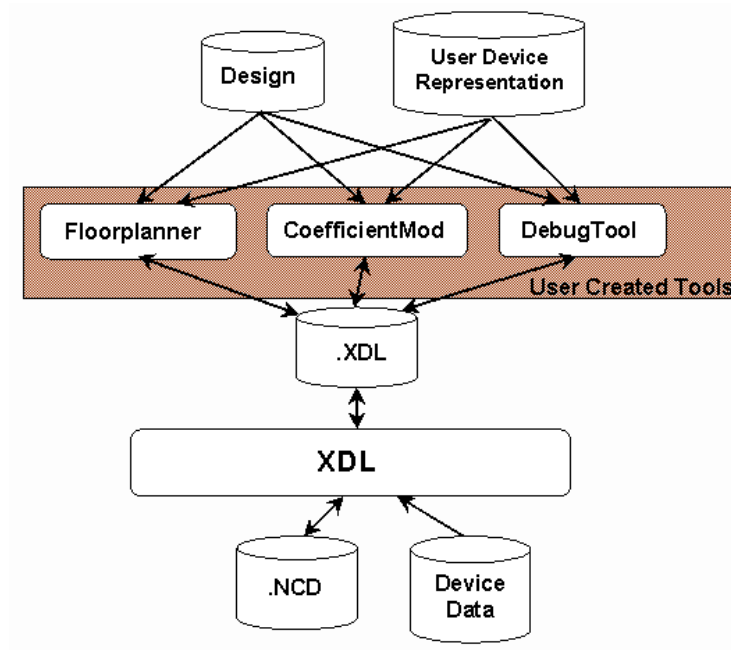

Figure 2.8: A foreseen usage of the Xilinx ISE command line utility `xdl` is shown. Besides the possibility to export primitive-level description of an FPGA chip, it offers also an NCD export to an ASCII XDL format and an import back to the NCD. (Source: [27])

Previous Results and Related Work

An overview of previous and related work in the field is presented in this chapter. The current state of the art in the field of radiation hardened FPGA is shown first and devices manufactured with emphasis to the radiation resistance are described. Various irradiation tests are described both done by our team and other researchers. Several frameworks and tools connected with the FPGA design flow are described and also previous work in the field of fault models simulation is presented.

3.1 Radiation Tolerant FPGAs on the Market

Main goal of this section is to provide the reader with the up-to-date overview of FPGA devices specially manufactured to withstand hostile environment with focus on radiation hardening (so called "space-grade" or "rad-hard" series of FPGAs). Other FPGA devices not manufactured in this category but usable for such an application will be discussed too. More over this text will provide very brief introduction to the radiation hardness issues, their sources, impacts and common means of eliminating used by the FPGA manufactures.

In this section we will present main companies producing FPGA devices with the focus on those variations of chips specified as somehow hardened against radiation. Most of the data presented here are gained from the official presentations of mentioned companies and publicly accessible technical texts. This part of research was originally done at beginning of 2013. Those information which significantly changed from that time were updated in 2018.

3.1.1 Xilinx

All information in this section comes from Xilinx website [29], FPGA family overviews, user guides, packaging and pinout specifications and other product documentation available.

Xilinx, Inc. is one of the biggest semiconductor company focused on programmable logical devices. It was founded in 1984 and now is the leader in this field with nearly 50 percent market share and 2.36 B\$ in revenues in year 2017. Their silicon products can

be divided into four groups: configuration memories, CPLDs, EPP[1] and mainly FPGAs. FPGAs are divided into families, which goes through "generations". The last generation contains two families: Kintex UltraScale+ and Virtex UltraScale+ pure FPGA devices, Zynq UltraScale+ MPSoC with integrated ARM processor cores and Zynq UltraScale+ RFSoC with integrated analog-to-digital and digital-to-analogue converters. This generation is built on 16 nm FinFET technology.

Xilinx has "space-grade" FPGAs in its portfolio in the Virtex families and currently is supporting two families of this specification. First family is Virtex-4QV, which was introduced in 2008, and the second family is Virtex-5QV introduced in the second half of 2010. One of the first Virtex space-grade devices was used in NASA's Mars rovers.

Virtex-4QV is SRAM-based FPGA manufactured on an UMC 90 nm copper CMOS process technology. It offers 400 MHz performance at 1.2 V core voltage and is delivered in three variants: LX (optimized for high-performance logic), SX (optimized for signal processing) and FX (optimized for signal processing, with embedded hard-coded processor[2]).

According to Xilinx tests devices from this family has guaranteed TID of 300 krad (per method 1019 [30] with dose rate ≈50.0 rad/sec). Increased SEL immunity is achieved by thin epitaxial layer in the wafer manufacturing process. This modification assure tolerance of LET value more than $125\,\mathrm{MeV\,cm^2/mg}$ (per method 1020 [30]), measured as heavy ion LET with fluency exceeding 10^7 particles/cm^2. SEFI is determined to be typically 1.5×10^{-6} upsets/device/day for typical GEO 36 000 km. Part of the integrated memory (namely BRAM) has support for parity bits and some devices (LX type) are equipped with built-in ECC. Virtex-4QV is more radiation tolerant than radiation hardened. It offers the same amount of resources as commercial versions of Virtex-4.

Virtex-4QV chips were packed by surface-mount-compatible ceramic flip-chip column grid array (CF) technology based on IBM technology CCGA (see Figure 3.1). This technology improves thermal cycle reliability and radiation tolerance. It uses high-temperature solder columns as interconnections to the board combined with a high-density, non-hermetic multi-layer ceramic substrate. Top is covered by silicon carbide heat-spreader. From 2014 these chips are shipped in Land Grid Array package type (without columns) [31] because IBM has shut down their line for these packages.

To utilize all offered radiation tolerance attributes, user design TMR is required as proposed by Xilinx. This can be accomplished by TMRTool [33] software supplied by the vendor (it supports Virtex-4QV and Virtex-5QV only).

Virtex-5QV is the next generation of Virtex-4QV family (see Figure 3.2). Is it manufactured on 65 nm CMOS copper process technology with 12 layers of metal and triple-oxide technology to reduce static power consumption. It's performance is little bit higher than Virtex-4QV: 450 MHz with core voltage of 1.0 V. Only one device is offered in this family (XQR5VFX130). It has only around 40 % of resources in comparison to biggest commercial Virtex-5 (20 480 vs. 51 840 CLB slices). This indicates more structures are added

---

[1]EPP stands for Extensible Processing Platform, a ASIC-FPGA co-design with an ARM processing unit and a programmable logical structure.

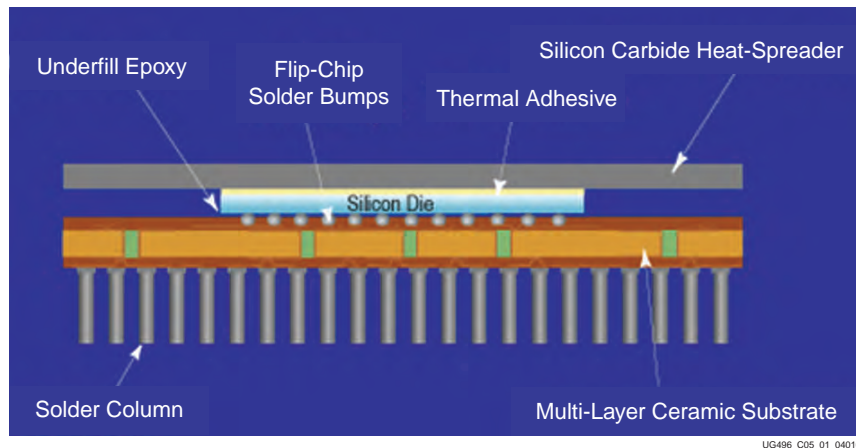[2]Dual PowerPC 405 RISC Cores operating at frequency up to 350 MHz.

Figure 3.1: CF package construction as it was used for Xilinx Virtex-4QV FPGA chips until 2014. (Source: [32])

to the chip to improve its radiation resistance parameters. Contrary to its predecessor, Virtex5-QV is more radiation hardened than radiation tolerant.

This device is made up by RHBD technology, i.e. its radiation hardness is achieved by the design. It has guaranteed TID of 1 Mrad (per method 1019 [30] with dose rate 300 rad/sec). Immunity to SEL is achieved same way as in Virtex-4QV (by a thin epitaxial layer in the wafer manufacturing process), ensuring tolerance of LET value more than 125 MeV cm^2/mg (per method 1020 [30], measured as heavy ion LET with fluency exceeding 10^8 particles/cm^2).

Figure 3.2: Xilinx Virtex-5QV FPGA chip. (Source: [29])

Used RHBD consists of several type of hardening techniques in this chip. All registers are implemented with dual-node latches, both in a configuration memory and user-land registers (in CLBs and IOBs). SEU rate for configuration memory at GEO orbit is stated to

be 3.8×10^{-10} errors/bit/day (with 35 Mbit of configuration memory). Data-paths across the chip are protected by filters against SET, providing filtration of glitches up to 800 ps. Supporting circuits (as configuration controller, JTAG controller or DCI [34]) are hardened (besides techniques previously mentioned) by TMR and independent and redundant error detection and correction circuits. This bring SEFI rate for this circuit to values about 2.7×10^{-7} interrupts/device/day, which is $\approx 10\,000$ years of MTTF. Error detection and correction mechanisms are also incorporated in block memories.

Virtex-5QV chips are packed by the same technology as Virtex-4QV.

3.1.2 Intel

All information in this section comes from former Altera website (not available any more) and Intel's Programmable Solutions Group website [35], FPGA series overviews, application notes and other product documentation available.

The original Altera Corporation has been found in 1983 and was producing a wide range of CPLDs, SRAM FPGAs and ASICs. The company was acquired by Intel Corporation in 2015 and all programmable portfolio moved to its Programmable Solutions Group division. It is the second biggest player on the FPGA market with its share about 40 % and 1.9 B\$ in revenues in year 2017 (only the Programmable Solutions Group division). Intel's silicon products can be divided into generations which go through series of their products. The latest generation is generation 10 and contain devices manufactured with different processes ranging from 60 nm down to 14 nm.

The main FPGA series in the last generation produced by Intel are Stratix 10, Arria 10, Cyclon 10 and MAX 10. The **Stratix 10** series manufactured with Intel's own 14 nm Tri-Gate process consists of the most powerful chips able to run at maximum frequency[3] 1 GHz. **Arria 10** devices are fabricated with TSMC's[4] 20 nm technology and represent compromise between performance and power consumption (and price). They are supposed to be used in mid-range projects and their maximum fabric frequency is 644 MHz clocks with core voltage 1.2 V. The next series of Intel's FPGA is **Cyclon 10** with two sub-series being fabricated with 20 nm technology and also power optimized 60 nm. Its frequency limit is 644 MHz and 500 MHz respectively with core voltage 1.2 V and 1.8 V respectively. It is recommended for the high-volume and cost-sensitive applications with lowest maximal performance (from previously mentioned) and also lowest power consumptions. The last series is Max 10, which was previously (in older generations) categorized as CPLD. Now in FPGA category, it is manufactured with TSMC's 55 nm, with maximum frequency 450 MHz and core voltage of 3 V.

Intel does not produce any FPGA devices dedicated as "space-grade", but on the other hand all of their latest chips are stated not to be prone to SEL and have several more different protection mechanisms incorporated.

---

[3]All the frequency limits stated in this paragraph are the limits of the clock network inside the chip's fabric.

[4]Taiwan Semiconductor Manufacturing Company Limited

All of the FPGA series have built-in **error detection and correction mechanism for configuration memory** by means of CRC. The configuration memory is protected both in the start-up configuration process and later during the use of the device. Configuration memory is divided into frames and CRC hashes are computed for each frame individually. The configuration process is secured by 16 bit CRC hashes precomputed in the time of building the configuration data (at the end of a design flow). In the device these precomputed hashes are checked on the fly during the configuration process. In case an error is detected, the configuration process is halted and the status is reported through dedicated pin. After successful configuration a new 32 bit CRC hash is computed for each frame and stored beside the configuration data in the device. In time of regular operation of the device the configuration memory is scanned all the time and CRC hashes are verified.

This process can detect up to 5 bit change in one frame of the configuration memory and it can correct up to 2 adjacent bits errors. In Stratix V devices (second latest generation in 2018) this protection provides 99.999 999 976 7 % of error detection. Since this mechanism is hardcoded into the FPGA, there is no performance impact on the user design. Moreover, structures of this mechanism inside the device are not susceptible to soft-errors according to Intel's research. FIT can be decreased by ignoring "don't care" configuration bits in Stratix devices. This is done by soft-logic function incorporated into user design and sensitivity map stored outside the chip. Every time CRC error is detected sensitivity map is consulted whether the error occurred in used part of the configuration memory or not. More over the Stratix V supports partial reconfiguration, encryption and compression of the configuration bitstream.

Next protection mechanism offered by Intel in their FPGAs is **integrated Error Checking and Correction (ECC) for user memory**. Memory blocks have built-in ECC which are able to detect up to three adjacent bits error and correct error of up to 2 adjacent bits. In some chips and for some memory configurations this support is only partial and additional soft-logic have to be provided in user design.

Soft-errors in configuration memory represent vast majority of SEU observed through testing. This is caused by the counts of memory bits utilized for the configuration memory and for the rest of the chip. Most memory bits are used for the configuration memory and the next bigger consumer of the memory bits is user memory, which is equipped by an integrated ECC. Beside this two memories (configuration memory and user memory blocks) chips contain only few more SRAM cells for user logic registers and I/O registers. None of these represents any severe risk in terms of SEU because **user logic registers** have smaller neutron cross-section (and thus require higher critical charge) and **I/O register** are in low count and uses higher voltage, which both make them more resistant against SEU. According to Intel statements, no upsets has been observed within the registers during SEU testing on lower density FPGAs. The overall MTBFI is stated to be of hundreds of years, even for very large, high-density FPGAs.

For those who need more than previously stated detection and correction mechanisms, Altera (at that time) used to offer an **ASIC migration process**, which was called Hard-Copy. HardCopy chips were pin-to-pin compatible with selected FPGA chips and had similar functionality inside. This program offered an easy migration with a result of nearly

SEU immune chip. The HardCopy variant was available only for selected chips and is no longer available (latest supported family was 40 nm Stratix IV).

3.1.3 Microsemi

All information in this section comes from Microsemi websites [36] and all their official documents available through these sites.

The Microsemi Corporation has been found in 1960 and now is producing wide range of semiconductor devices in area of aerospace, defense, security, enterprise, communications, industrial and alternative energy markets. In November 2010 Microsemi has acquired an Actel Corporation which brings to the company portfolio a non-volatile (antifuse and flash based) programmable logic solutions. Microsemi Corporation has been acquired by Microchip Technology Incorporated (NASDAQ: MCHP) at beginning of 2018, but still keeps its branding at the time of writing this text.

Nowadays Microsemi offers several families of flash-based FPGA devices (ProASIC-PLUS, IGLOO2, IGLOO, ProASIC3, Fusion and PolarFire) and also antifuse-based chips (SX-A, Axcelerator (AX), eX and MX). Some of them have also a "radiation-tolerant" variant, namely RTAX-S/SL, RTAX-DSP, RTSX-SU, and RT ProASIC3. These families form a "Radiation-Tolerant FPGA" group of the Microsemi portfolio. All of these chips utilize RHBD. Microsemi expanded its "Radiation-Tolerant FPGA" portfolio in 2015 adding RTG4 family of flash-based FPGAs.

The **RTAX** and **RTSX** families are antifuse representatives of radiation tolerant FP-GAs. Both families have their commercial ancestors (Axcelerator and SX-A) and are hardened by design, by the similar way. In this paragraph we will introduce common features of these devices. Modified CMOS process is enriched by metal-to-metal antifuse technology. These antifuses are made from amorphous silicon and are situated between two uppermost metal layers. The radiation hardening is implemented by design in several ways. Clock lines are wider and have stronger clock drivers. Registers (in logical and I/O block) are hardened by TMR in the silicon, which eliminates need of use a TMR in user design. As shown in Figure 3.3 the TMR scheme used is triple latches with separated voters and common lines for clock, data and other signals [37]. The sensitivity of this architecture to a SEU depends on the roll and tilt of the device against the radiation according to tests. At some angles the probability of striking more than one flip-flop in TMR at a time with charged particle is higher. Configuration memory is stated not to be prone to any SEE because of antifuse technology used[5]. It also allows zero boot time, because configuration data storage is nonvolatile.

RTAX-S/SL and **RTAX-DSP** devices are intended to be used in high-altitude or space applications. They are manufactured on 150 nm CMOS antifuse technology with 7 metal layers and with 1.5 V core voltage they are able to run on up to 350 MHz. These

---

[5]Some radiation-induced antifuse failures have been noticed during tests [38], but according to Microsemi never in any space-flight data. The antifuse rupture cases have been reported at the temperature of 125 °C and with increased supply voltage to 110 % of nominal. The LET threshold in this case was approximately 80 MeV cm^2/mg.
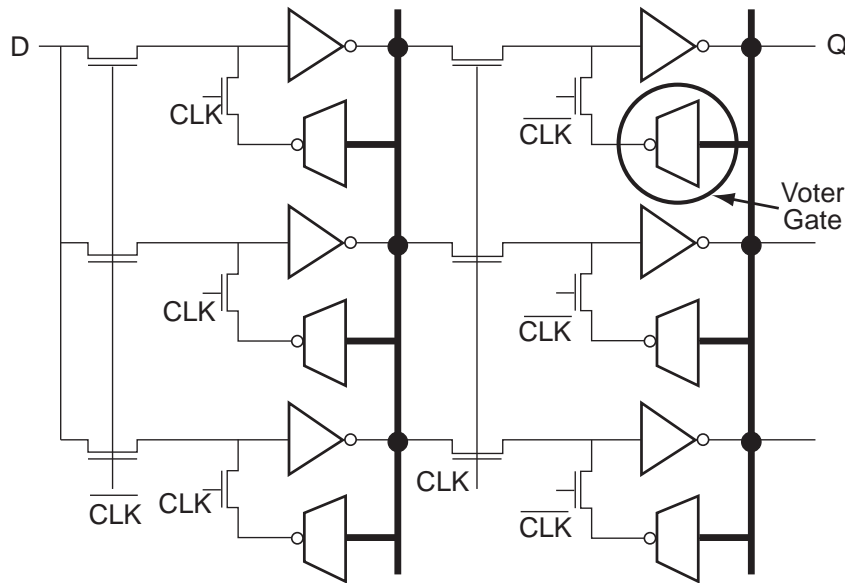
Figure 3.3: The Actel SEU-hardened D flip-flop used in their RTAX and RTSX FPGA chips. (source: [39])

chips offer up to 4 millions of equivalent gates and up to 840 user I/O pins. According to Microsemi official documents these devices can withstand TID up to 200 or 300 krad[6]. No SEL has been found in LET up to $117\,\text{MeV}\,\text{cm}^2/\text{mg}$. For the memory blocks, Microsemi offers an Error Detection and Correction (EDAC) feature with integrates an SRAM scrubber for a long term data storage. Both are in a form of a soft logic (an IP core). These features can detect up to two bit error and correct single bit error and offer SRAM upset rate lower than 10^{-10} Errors/Bit/Day. RTAX devices are divided into two groups, already mentioned **RTAX-S/SL** and **RTAX-DSP** which feature added math blocks. These DSP blocks are also SEE-hardened by design (both against SEU and SET) and offer standard MAC functionality. Packages used are CQFP, CCGA and LGA.

The **RTSX-SU** is manufactured on 250 nm CMOS antifuse technology with 3 to 4 metal layers. On the 2.5 V core voltage the system is capable of running at most at 230 MHz. The biggest device offers 108 thousands equivalent gates and up to 360 user I/O pins. The devices are stated to be immune to TID up to 100 krad and to SEL with LET more than $104\,\text{MeV}\,\text{cm}^2/\text{mg}$. The registers immunity to SEU are stated to be more than $40\,\text{MeV}\,\text{cm}^2/\text{mg}$, which leads to SEU rate in the worst-case geosynchronous orbit to less than 10^{-10} upsets/bit/day. These devices are packed in CQFP, CCGA, LGA or CCLG packages.

The **RT ProASIC3** family is one of two non-antifuse radiation tolerant FPGA family from Microsemi. It utilizes the flash technology for storing the configuration data. Flash

---

[6]The limit 200 krad is mentioned as parametric limit (all parameters are preserved) and limit 300 krad is mentioned as functional limit.

memory cells are directly incorporated into logical layout of the device and directly drive the configuration elements. These FPGAs are produced on 130 nm LVCMOS technology with 7 metal layers. On the core voltage of 1.5 V the maximum performance is 350 MHz. According to Microsemi tests these devices are SEL immune to LET up to 68 MeV cm^2/mg. The calculated SEL error-rates in LEO orbits are 1.93×10^{-12} SEL/FPGA/day for A3PE600 (the smaller device) and 2.71×10^{-10} SEL/FPGA/day for A3PE3000L (the bigger device). Up to LET of 96 MeV cm^2/mg there have been no SEU in the configuration memory during laboratory tests with heavy ions. SEUs in user flip-flops have been observed from LET of 6 MeV cm^2/mg and in user SRAM memory from LET of 1 MeV cm^2/mg, but both values appears to be frequency-dependent. SET have been observed on global clock network at LET of 4 MeV cm^2/mg and on I/O banks at LET of 7 MeV cm^2/mg. The SEU characteristics of programming circuitry are not known yet, so the programming on orbit is not recommended at this time. The radiation performance is stated to be as follows: devices are functional even after 100 krad, but the 10 % propagation delay increase is observed between 25 and 30 krad measured with dose rate 5 krad/min. With the dose rate less than 1 rad/min (which is more representative of the space environment) the 10 % threshold of propagation delay have been reach at a TID level of 40 krad. For TID effects, the primary issue is the radiation-induced charge loss in the floating gates. Chips are packed in hermetically-sealed, ceramic packages available as CQFP, CCGA or LGA.

The **RTG4** is the newest radiation-hardened non-volatile FPGA family from Microsemi with up to 151 824 registers and TID of more than 100 krad. It is manufactured with the 65 nm process with a claimed SEL immunity for the LET >103 MeV cm^2/mg. The flash technology used for the configuration memory is stated to be immune to SEUs for the LET >103 MeV cm^2/mg. User data memory registers are hardened by design with an integrated TMR and embedded SRAM memory blocks have SECDED protection. User registers are claimed to be SEU immune for the LET >37 MeV cm^2/mg with a SEU rate in GEO (for a solar minimum) $<10 \times 10^{-12}$ errors/bit $-$ day.

3.1.4 Microchip

All information in this section comes from Atmel Corporation website (not available any more) and Microchip website [40], FPGA IC overviews, application notes and other product documentation available.

The original Atmel Corporation has been founded in 1984 and was producing FPGA chips from 1993 (in this year Atmel acquired Concurrent Logic and thus gain their FPGA portfolio). The Atmel company was acquired by Microchip Technology Inc. in 2016. Atmel had two SRAM FPGA devices (the AT40 and the newer ATF280), one hybrid chip ATF697FF and one not-yet-released chip (ATFS450) in the category of radiation-hardened programmable circuits at 2013. In 2018, only AT40K family and ATF697FF are stated on the Microchip web site as available. Microchip also offers an automatic FPGA retargeting service transforming FPGA design into an ASIC.

These devices are radiation hardened by design, with built-in SEE protection. This protection incorporates SEE hardened cells for configuration and user memory, user reg-

Features	AT40	ATF280	ATFS450
Technology	0.35 µm CMOS	0.18 µm CMOS	0.15 µm SOI
Equivalent gates	46K	280K	450K
User memory	18 kbits	112.5 kbits	180.5 kbits
System speed	20 MHz	50 MHz	70 MHz
Number of cells★	2304	14 400	23 104
Availability	Available	Available	NA yet

★ two 3-LUT (or one 4-LUT) and one HDFF

Table 3.1: Overview of Atmel rad-hardened FPGAs from 2013

isters, I/O blocks and JTAG block. According to Atmel usage of these hardened cells remove the need of TMR in user design. Configuration stream for both types is protected by checksum in time of load. List of key feature can be found in Table 3.1.

The **AT40** chips are intended for low gate and low power applications. They are produced with rad-hard 350 nm single-poly, 4-metal CMOS process and has less than 1 Mbit of configuration bitstream. It is stated, that no SEL should be present below a LET threshold of 70 MeV cm^2/mg and also chips have been tested up to a TID of 300 krad (Si) (per method 1019 [30]) without any traces of degradation. Supply voltage for this type of FPGA is 3.3 V and package type is MQFP with either 160 or 256 pins (with 129 or 233 user I/O pins respectively).

Newer **ATF280** type was manufactured in 180 nm rad-hard CMOS process and had 4 Mbits of configuration bitstream. It was claimed to be resistive to TID higher than 300 krad with SEL threshold higher than 80 MeV cm^2/mg. The low sensitivity of configuration memory to SEU results in an upset rate lower than 10^{-6} errors/device/day in the worst orbit conditions. Core voltage for this type of FPGA spanned from 1.65 to 1.95 V and the package used was either MQFP or MCGA with up to 472 pins (with up to 308 user I/O pins). The ATF280 had one feature added concerning the configuration memory, a Configuration Self Integrity Check (CSIC). The CSIC is a mechanism of online checking of the configuration memory integrity.

Moreover, Microchip offers the **hybrid chip ATF697FF** (see Figure 3.4) originally developed by Atmel in 2012. It is a rad-hard CPU with embedded ATF280 SRAM-based FPGA. The processor implemented is the European Space Agency (ESA) SPARC V8 LEON2 fault tolerant model also known as AT697F for the Microchip standalone chip. The configurable unit offers a distributed SEU hardened static RAM which saves logic resources for memory implementation. Here comes key feature of the ATF697FF chip:

- Microchip 180 nm rad-hard AT58KRHA CMOS technology

- 32-bit SPARC V8

- 90 MIPS (SYSCLK = 100 MHz)

○ consumption 0.8 mW/MIPS

○ operation voltage 1.8 V (logic), 3.3 V (buffers)

○ TID at least 300 krad

○ SEL immunity 95 MeV cm^2/mg @ 125 °C
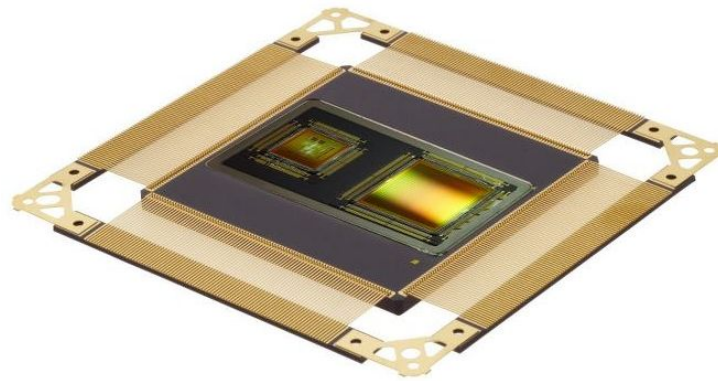
○ ATF280 FPGA incorporated

Figure 3.4: The Microchip ATF697FF with an FPGA chip on the right and a LEON2 processor on the left. (Source: [41])

3.1.5 Achronix Semiconductor

Achronix Semiconductor Corporation has been found in 2004 and is private fabless corporation designing programmable digital circuits. Achronix is now manufacturing and selling Speedster22i FPGAs based on the Intel's 22 nm technology with 3D Tri-gate transistors and Speedcore eFPGA based on TSMC 16 nm FinFER technology. Beside this commercial devices, Achronix together with BAE Systems[7] announced [42] to develop high-density and high-performance radiation hardened asynchronous FPGA named RadRunner. This chip should be based on BAE's radiation hardened 150 nm epitaxial bulk CMOS technology, called RH15. The logic on this technology should be using Achronix's picoPIPE technology together with a Redundancy Voting Circuit (RVC) methodology to protect the user circuits from single event effects.

The picoPIPE is an asynchronous digital technology offered by Achronix and used in their commercial chips too. It lacks synchronous registers and small clouds of logic elements are surrounded by asynchronous connecting elements which guard data transitions through the acknowledge signals. The new RVC technology brings two module redundancy

---

[7]BAE Systems is a British company focusing on military and defense.

followed by local voting mechanism blocking the data propagation in case of malfunction of preceding logical blocks.

A prototypes of the intended architecture has already been tested, according to the paper. Results show that no SEL and SEFI have been observed up to a temperature of 74 °C and a LET up to 55 MeV cm²/mg. No mention about the RadRunner product can be found neither on the Achronix website nor on the BAE website in 2018, though.

3.1.6 Summary

In previous text we have shown current state of the art in the field of radiation hardened FPGA. Main producers of FPGA circuits have been mentioned and their devices manufactured with emphasis to the radiation resistance have been described. In this section we will conclude current possibilities of radiation hardening which is used by leading semiconductor companies.

There are several stages in which the radiation hardness can be achieved. Multiple approaches can be combined together to form more resilient device, but most of these techniques have their own drawback. Every step of radiation hardening has an impact on performance.

In the manufacturing process usually old node is used because its technology is well known and bigger the node, lesser the radiation sensitivity. But this approach goes back in time and cause rising power consumption and lowering the maximal frequently. Every special modification of the technology process causes growth of price. Technology nodes used in this field of integrated circuits goes from 350 nm down to 65 nm, while today's most advancing chips are manufactured on 22 nm node and heading towards 14 nm. The technology can be modified by added flash or antifuse process; can use an insulator as a base or only a layout can be modified. The **packing** plays important role too. Pure and properly chosen material and technology of packing can avoid secondary generated radiation and thus lower the total dose absorbed.

Next step is taken in **designing FPGA architecture**. What building blocks the device can offer and how much of these resources are available can have direct impact on the radiation sensitivity. More complex structures are more prone to radiations and bigger amount of configuration memory increase the probability of functionally failure of the device. On the other hand checking and correcting structures added to the architecture lower the final error-rate. Most of these features were originally implemented in user design and their hard-coded transformation into silicon now helps to reduce errors. In addition to that, a hard-coded logic should be less prone to radiation, because it lacks most of the volatile memory elements — the configuration memory. Among these features we can find ECC memories, CRC codes for configuration memory or TMR for user registers.

Described technologies used today for the radiation hardening are at the edge of usability in radiation environment like outer space or high energy experiments. We have to use the best what technology can offer us and build architecture which guarantee the specified level of radiation hardness. There are some aspects which the technology can avoid (for example SEL is not a problem for most of the application), but some of them still persist

and have to be compensated by architecture. The susceptibility of SRAM memory cells to the radiation is still the issue. And here are open area for innovation in the architecture, new ways how to piece building blocks together to deal with the unreliability of the weak link in the technology chain. Beside this there are also possibilities of enhancing the technology itself indeed.

3.2 Irradiation Tests

In this section radiation experiments meant to collect data for the model calibration are presented. These experimental data were obtained through the collaboration with Tomáš Vaňát from CTU, FIT working on his research project "Physical Fault Injection and Monitoring Methods for Programmable Devices" [26, B.2].

Several irradiation tests were done on the cyclotron U-120M at NPI, Řež (see section 2.7 for details). A proton beam was used to irradiate a Xilinx Spartan-3 FPGA (part number XC3S200-4-FT256) made in $90\,\mathrm{nm}$ CMOS process. The tested FPGA was loaded with a 4-bit wide circular shift register with inserted combinational logic between individual register stages shown in Figure 3.5. The same design was also analyzed by our model, see Chapter 6 for more details.
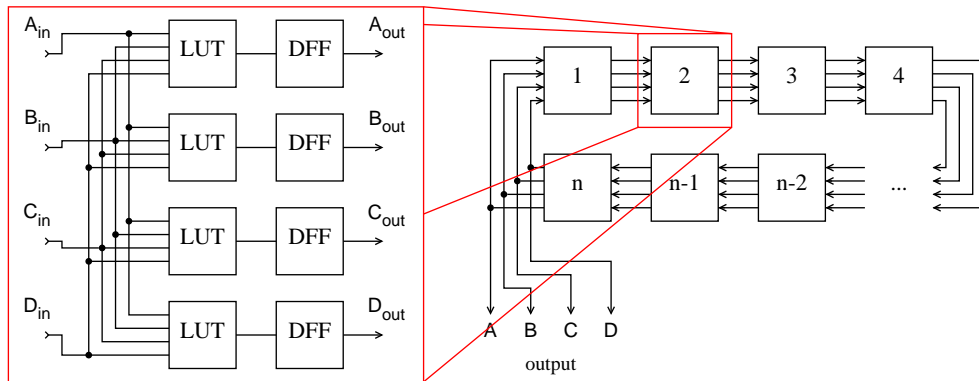

Figure 3.5: The FPGA design used for irradiation tests is visible on the right. The design is self-initializing (after the reset), so it has only outputs. Each sub-block (shown on the left) contains four 4-LUTs and four user memory registers. The design is specially crafted for the given FPGA architecture (Xilinx Spartan-3), so it uses all SLICE LUTs and registers available on the chip. (Source: [26])

As a result from these tests, data mentioned in Table 3.2 were obtained. Those can be compared with data presented in Xilinx Device Reliability Report [43], where a neutron cross-section for Spartan-3 configuration memory is stated to be $2.4 \times 10^{-14}\,\mathrm{cm^2\,bit^{-1}}$ $\pm\,18\,\%$. In another research [44], neutron cross-section for the same chip is stated to be[8] $2.81 \times 10^{-14}\,\mathrm{cm^2\,bit^{-1}}$.

---

[8]Assuming that the data shown in Table 1 in the referenced publication is a per bit cross-section.

Beam type	Energy [MeV]	Cross-section $[10^{-14}\,\mathrm{cm^2\,bit^{-1}}]$
p^+	31.04 ± 0.31	2.03 ± 0.16
p^+	29.10 ± 0.34	1.66 ± 0.23
p^+	27.05 ± 0.35	1.62 ± 0.14
p^+	24.88 ± 0.38	1.64 ± 0.18
p^+	22.53 ± 0.40	1.56 ± 0.19
p^+	17.12 ± 0.45	0.91 ± 0.12
p^+	13.84 ± 0.50	0.59 ± 0.06
p^+	9.74 ± 0.56	0.35 ± 0.11
n^0	14	0.50 ± 0.08

Table 3.2: Xilinx Spartan-3 FPGA configuration memory cross-sections for different energies of accelerated proton and neutron beams. (Source: Figure 5.5 from [26])

FPGA resource	Configuration bits	Normalized cross-section	
		$1 \to 0$	$0 \to 1$
LUTs	61 440	1.00	1.29
MUXs	61 440	0.25	0.82
Slice configuration	61 440	0.61	1.08
Decoded PIP	245 760	0.38	0.90
Non-decoded PIP	153 600	0.46	0.81
User memory (BRAM)	225 024	0.84	0.93

Table 3.3: Relative Xilinx Spartan-3 FPGA configuration memory alpha-induced cross-sections for different sub-categories of the configuration memory. Values are displayed for $1 \to 0$ and $0 \to 1$ transitions separately and they are normalized to the LUT $1 \to 0$ cross-section. (Source: Table I from [45])

3.3 Other Irradiation Results

A different approach was chosen by authors in [45] where same Spartan-3 FPGA chip with etched package was irradiated by alpha particles. Authors doesn't state absolute cross-section values (which would nevertheless not be easily comparable to proton or neutron induced ones), but relative sensitivity of different types of configuration memory is mentioned (see Table 3.3).

3.4 Non-commercial Tools

Several non-commercial frameworks and tools connected with the FPGA design flow and used in this thesis are described in this section. They are usually academic and/or open-source projects, often primary used for FPGA architecture studies.

3.4.1 Verilog to Routing

One of the more complex frameworks is the Verilog to Routing project (VTR) [46, 47]. This open source academic CAD suite is intended to help with both proposing new architecture of programmable circuits and developing EDA tools for such chips. This tool allows user to create custom FPGA architecture and perform timing-driven packing, placement and routing on it. It cannot be used straight for design synthesis into commercial FPGAs as none of these have architecture open enough. But in case our custom FPGA architecture (even virtual) can be provided, these tools can do all from Verilog language to design finally routed on the given architecture. First part of VTR is the **ODIN II** Elaboration and synthesis tool for Verilog language, than the **ABC** provides logic synthesis and FPGA technology mapping and finally the **VPR** (Versatile Place and Route) provides packing, placement and routing. The latest official release of VTR in version 8.0 from October 2017 can be downloaded from the project web page [48]. Development source codes are accessible on the project GitHub page [49].

3.4.2 RapidSmith

The RapidSmith is a Java framework for low-level manipulation with Xilinx FPGAs [50]. It can interface with the official Xilinx ISE FPGA flow tool set via XDL files [28]. A complete definition of internal structures is obtained for a selected FPGA down to the level of individual building elements (e.g. LUT, multiplexers, memory elements...) and routing wires. User can import either mapped, only placed or fully routed design from the ISE into the RapidSmith, make any changes in the placement and/or routing and export the design again into the XDL format, which can be later imported back to the ISE tool set. The framework also offers functions for bitstream manipulation.

3.4.3 HOPE

The HOPE is a fault simulator supporting stuck-at fault models simulation in synchronous sequential circuits [51]. Parallel version of the single fault propagation technique is implemented. An input format of a digital circuit to be analyzed is the BENCH, format used for ISCAS'85 and ISCAS'89 benchmark circuits [52]. Input test vectors can be supplied in a separate file or a built-in random test pattern generator can be used. If no any additional information is provided, all possible fault in the circuit are simulated. This behavior can be constrained by suppling a list of selected fault as an additional program input. In this

case, only supplied faults are simulated. The main output from the HOPE simulation is a list of detected and undetected faults for the given input test vectors.

3.5 Models and Simulations

A need of different fault models for analysis of FPGA devices is widely known. Different fault models were published tackling both logic FPGA part and its routing [53, 54, 55, 56]. The detailed architecture of commercially available FPGAs is rarely available and if so, only a view in a certain level of abstraction is available and not the detailed transistor level description. The reasonably precise fault model of the whole FPGA is thus difficult to obtain in most cases. Particular fault models can usually be composed by ways of a bitstream analysis and architecture estimation, often to be confirmed by an irradiation testing or a fault injection campaign [19].

Several works on SEU simulation or modeling in CMOS ASIC have been published. In [57] SPICE-like simulator called PARA has been introduced. This simulator is based on switching-level simulation algorithm aimed on degradations in electrical characteristics. Another SPICE-like software called SITA have been introduced in [58]. This software accomplishes probabilistic device-level simulation to analyze SEU induced errors propagation. Another probabilistic description of error propagation in complex circuits formulated and solved as a set of linear equations has been published in [59]. Here results have been compared with experimental data obtained from real chip exposed to high energy laser.

Several works were also published targeting fault model simulation. In [60] for example, authors present a probabilistic error propagation model on general FPGA fault models. The error rate of each node is evaluated and the whole circuit error probability is calculated based on the error rate propagation.

Also SEU simulation in FPGA is mentioned in several papers, for example in [61] there is a hardware-software co-simulation presented. Sensitivity of various designs to configuration SEU has been tested with SLACC-1V (see [62]) computer acceleration card. This platform includes two Xilinx Virtex XCV1000 devices in which the tested design has been implemented in parallel (same for both devices, a tested design on one and a golden design in another). In one of the Virtex device a bitstream has been modified during the simulation. This has been done in bit-by-bit basis to simulate all possible configurations SEU by brute-force. For the whole chip one complete simulation last for nearly 27 minutes. No inner FPGA architecture has been considered and only chip-level characteristics has been analyzed. This work has been further developed in [63] where TMR design modification has been evaluated and in [64] where results has been compared with real radiation tests.

Another method is published in [56] and subsequently in [65], where authors presented an analysis tool based on behavioral simulation able to calculate an average predicted error rate of a single SEU. A double-method of design simulation and device testing have been proposed for characterizing SEU in configuration memory and SEFI these SEU can induced. In proposed method so called "physical layer" (FPGA device) and "application layer" (the

implemented design) is analyzed independently. For purpose of proofing this method, an irradiation test (ALT) of selected designs have been performed on an ion accelerator in this work. Radiation testing have been conducted on Xilinx FPGA Virtex XCV300PQ240-4 basically in two ways to characterize this platform. In a first part of this work device have been irradiated until SEFI has appeared. Than bitstream have been read from the device and connection between SEU caught inside the device have been concluded. In the second part empty device (with no design loaded) have been irradiated and whole bitstream have been read periodically to get the average "device cross-section" (i.e. probability of SEU caught). Software simulation have been done to evaluate actual design by way of ad-hoc developed simulation tools. This simulation perform analyses only for SEFI induced by SEU (i.e. it cares only when implemented design stopped function properly). Two designs have been used in presented work: a sequential one (partial processor core) and a combinational one (set of multipliers). This is accomplished by deep analysis of bitstream, structure of which has been known. Actual analysis has utilized Xilinx design flow tools, especially `NCD2VHDL` (tools to convert placed and routed design back to VHDL description). In conclusions following equation has been proposed:

$$\sigma_{PREDICTED} = \epsilon_{PREDICTED} \cdot \sigma_{fpga} \qquad (3.1)$$

where $\epsilon_{PREDICTED}$ stands for calculated design error rate (SEFI probability computed by simulation), σ_{fpga} stands for "device cross-section" (obtained by radiation test) and $\sigma_{PREDICTED}$ results in predicted error rate for given design on design platform. As a result, LUT structures has been identified as most sensitive part of a bitstream and "short" type defect in routing matrix has been identified as most threating for SEFI arose from interconnection SEU.

This work was continued in [66] where a static analyzer tool was presented to inspect an FPGA design and verify its immunity against SEU in the configuration memory. Routing resource fault models obtained by a bitstream analysis and architecture estimation were considered and the soft error impact on the TMR mitigation technique was evaluated. The presented static analyzer tool is later referenced to as STAR.

FPGA routing fault models were considered also in [20] and their static analysis was used to help develop the Reliability-Oriented Place and Route Algorithm (RoRA). This algorithm can place and route a design (not taking into account timing constraints) in such a way, that no corruption of the selected mitigation technique (TMR in this case) can be caused by a single soft error. Results of this algorithm were tested by the fault injection campaign on the Xilinx Spartan-II FPGA.

Another simulation approach was presented in [67] where a general fault model simulation environment based on Stochastic Activity Networks (SAN) was introduced. The work was extended in [68, 69] to include general fault models of the logic portion of an FPGA in order to calculate observability for individual soft error defects in logic structures.

All previously mentioned works were combined in [70, 71]. The Electrical Effects Static Analyzer (E^2STAR) was introduced as an enhanced version of the STAR tool. Overall simulation environment was presented comprising the standard Xilinx toolchain from which

data can be extracted in different level of processing and analyzed on functional effects of SEU defects. The analyzed FPGA design enriched by general fault models of logic resources and empirical fault models of routing resources presented earlier is sequentially simulated for individual faults by the SAN simulation tool to obtain individual fault observability.

Another approach have been publish in [2] and [54]. Here SEU emulator has been used to evaluate influence of SEU to the digital circuit implemented in FPGA. Architecture of Atmel FPSLIC[9] device have been used where microcontroller control artificial SEU bit-stream injection and FPGA hosts tested design in one half and testing circuits in other half. With knowledge of bitstream structure authors have been able to classified possible defects and connect them with corresponding bits. In this way they have been able to predict sensitivity of individual bits in bitstream (according to connected defect) and then compare it with real results from emulation runs. Authors used this system for characterizing several different designs implemented onto FPSLIC architecture.

---

[9]FPSLIC device from Atmel is combined device compound from AVR 8-bit microprocessor and Atmel FPGA chip in one package.

Overview of Our Approach

Topic of this dissertation thesis is "Reliable FPGA Architecture". Main field of research is an FPGA architecture with emphasis to its sensitivity (and immunity) to radiation induced faults. Based on the background knowledge and previous works from this field (mentioned in Chapter 2 *and* Chapter 3*) we have decided to propose our own method describing how to create and use a simulation model of an FPGA which will provide a tool to quantitatively characterize designs in terms of radiation dependability.*

Proposed steps to create and used this model are presented in this chapter.

4.1   Proposed Simulation Method

In this section, we describe the proposed method of an arbitrary FPGA design fault analysis and fault simulation. The preliminary variant of this method was already presented in [A.3] and [A.4].

The goal of the proposed method is to quantitatively characterize a given digital design implemented on a given FPGA device with respect to its behavior in an environment where SEU can occur. This characterization is done by a computer simulation, thus it doesn't need a target environment (e.g. space or a high energy experiment) neither its analogy (e.g. source of ionizing radiation used for ALT). On the other hand, it needs a simulation model of the given FPGA architecture as precise as possible and also a set of tools (computer applications) able to work with this model.

The method itself is universal as it can be used to characterize any given digital FPGA design on any given FPGA architecture provided the appropriate model for this FPGA architecture exists. The particular model is thus specific for the FPGA architecture being used to analyze.

A model created with the proposed method can be further used for investigation of FPGA architecture and for evaluation of these architectures according to their susceptibility to soft-errors. It can help to characterize possibilities of architecture improvement and quantify its contribution.

We will now describe the model itself, what it should contain and how it can be created and calibrated. Later, we will describe also how the model is going to be used to characterize an arbitrary digital design targeting the selected FPGA device.

4.1.1 FPGA Simulation Model Creation

The model has to contain fault models for all faults which can be caused by a SEU in the given FPGA and enough information about the FPGA architecture itself to allow application of those individual fault models in the simulation stage. The model should also contain detailed characteristic of the device radiation sensitivity to allow quantitative characterization of the FPGA design in the simulation step. Different memory elements used to control different FPGA resources can have different sensitivity (cross-section) [72, 56, 45], so a detailed characterization of individual categories of architectural elements is needed for correct analysis of the given implemented design. We are aware of the lack of detailed description of any commercial FPGA actually available, thus we propose the following approach to obtain as precise model as possible.

As illustrated in Figure 4.1, a preliminary version of the model is created taking into account all information available from FPGA documentation and also from the FPGA vendor EDA toolchain. Various memory elements (like dedicated blocks of RAM and flip-flops present in islands of logic) are easy to model, as their intrinsic model usually corresponds to their implementation in FPGA devices. Islands of logic can be also relatively well modeled, because their structure is usually described in device documentation to such extent, which permits us to construct reasonably accurate models. To some extent, implementation details of some elements can be estimated based on patents being published by individual FPGA manufactures [73, 74]. The biggest challenge in the FPGA architecture processing and model preparation are routing resources. The actual implementation is really rarely available and only a general structure is known.

Once the model is created, its quality can be assessed by comparing results obtained for a selected design via simulation of this model and results from ALT of the same design on the given FPGA architecture. If any discrepancy is observed, the model should be refined (calibrated) using results obtained from the ALT. The sufficiently calibrated model then provides an environment for successful evaluation of different digital designs targeting given FPGA architecture.

The process of creating a well calibrated model can be lengthy and laborious and usage of a target environment or its analogy (when using ALT) can be needed, but this process needs to be done only once for a given FPGA architecture.

4.1.2 Model Usage — Design Simulation

The simulation flow shown in Figure 4.2 presents the usage of the model generated according to the previous section. The model creation process needs to be performed just once for a given FPGA architecture, but can consume a significant amount of time and
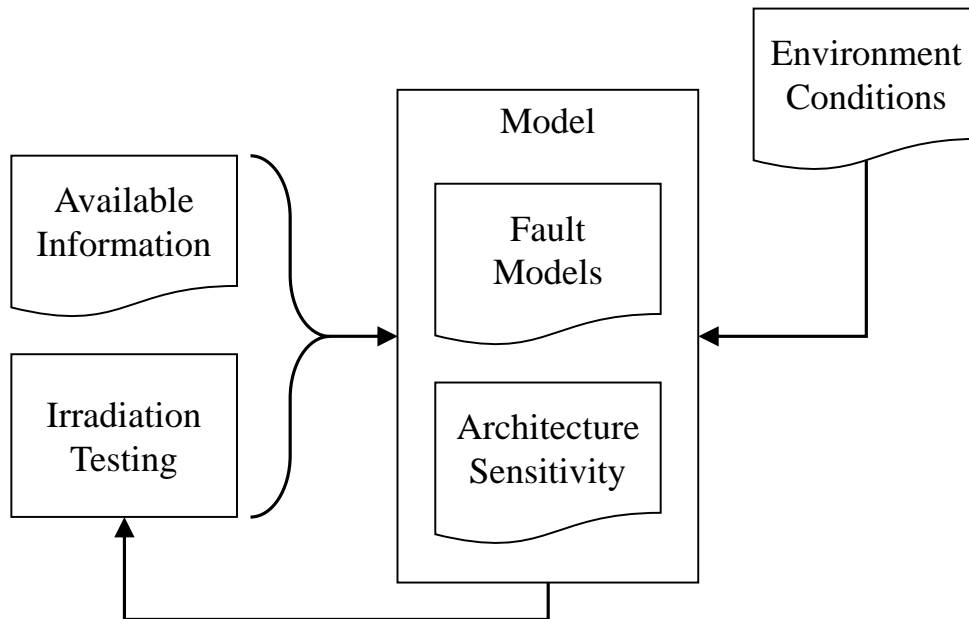
Figure 4.1: The schema of the model creation and calibration process. The model is initially built from available information from FPGA documentation and vendor tools and it is calibrated by an irradiation testing later, as needed. The resulting model consists of a set of fault models and an architecture SEU sensitivity under given environment conditions.

resources. The simulation, on the other hand, is meant to be a lightweight process which can be repeatedly used for characterization of various designs.

As input to this simulation flow, a calibrated model described in the previous section needs to be supplied. Also a design to be characterize is supplied with adequate test vectors. The design is enriched with individual fault models described in the supplied FPGA model first and then a fault simulation is executed using test vectors. Real test vectors should be used to simulate the design under realistic conditions. The use of the HOPE fault simulator described in section 3.4 is suggested to perform this step.

As a result of the simulation, a set of detectable and undetectable faults is generated. When combined with sensitivity characteristics of individual FPGA elements (contained in the FPGA model) a real overall failure rate for the design under test is calculated. An arbitrary FPGA digital design targeting the given FPGA architecture can be simulated by this flow.

4.2 Proposed Implementations

In this section, proposed implementations of our method are presented. The first attempt to implement the method was done with the VTR project. A short overview of the idea behind and the outcome of this approach are presented in following text. Later on, we
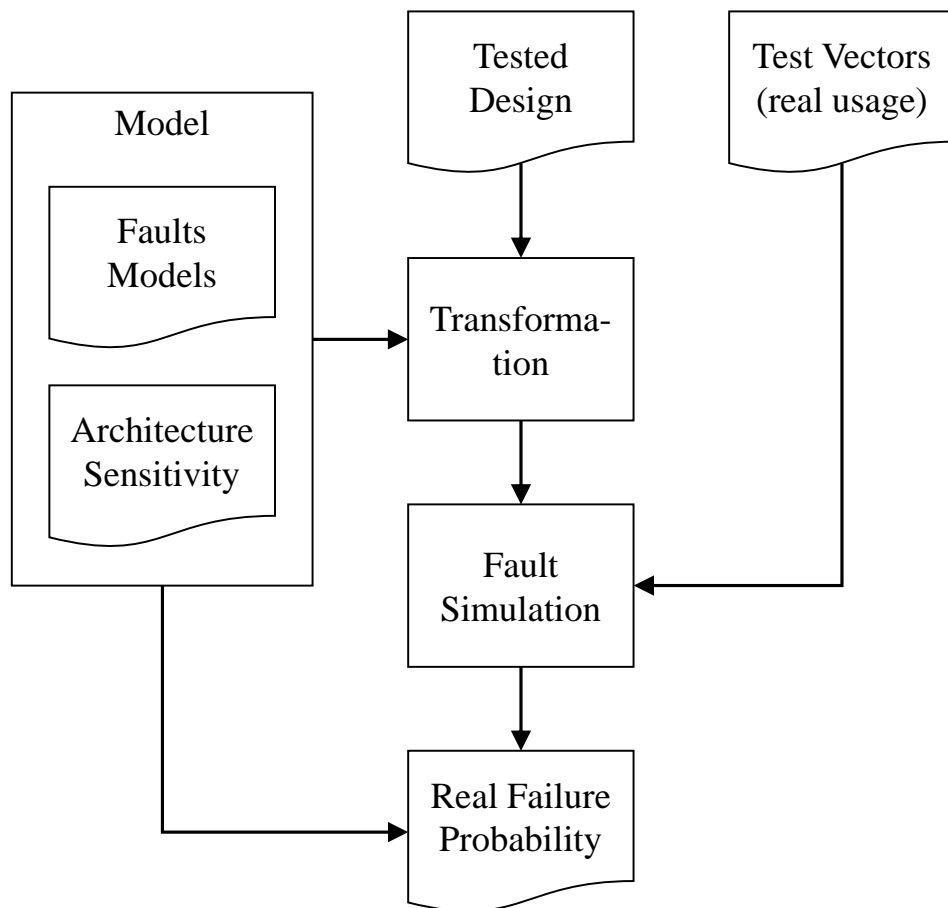
Figure 4.2: When created, the model can be used to simulate an arbitrary digital design targeting the selected FPGA architecture under given environment conditions. The tested design is enriched by fault models first and the transformed circuit description is then fault simulated. If real test vectors are supplied for the simulation, the real failure probability of the design in question can be calculated.

tried a different approach and the RapidSmith framework has been used for the implementation. Only a short overview is given in the following text with a detail description of this implementation in Chapter 5.

4.2.1 VTR Implementation

The original idea was as follows. The VTR toolchain, described in section 3.4, has been chosen to be used for implementation of the proposed method, especially its back-end part VPR. The VPR provides the ability to pack, place and route a design already mapped to FPGA architecture primitives. We have chosen this toolchain because none modern FPGA architecture is known to details and none commercial tools are open enough to allow us intended modifications. On the other hand, both the VTR toolchain and any format used in it are completely open, so the whole toolchain can be modified or only a part of it can be used.

The selected FPGA architecture (Xilinx Spartan-3 in this case) should be fully described as a VTR architecture and only VTR toolchain should be used for design processing and analysis. The given design is synthesized on the given architecture and from the modified back-end of the VPR tool a detailed resource graph and an annotated physical netlist of the design is obtained. All important information about implemented circuit should be available in this stage of the design flow and also the final placement and routing should be known. As a next step, artificial SEUs are introduced through a bit-flip injection and the modified physical netlist is simplified. The fault analysis can be performed subsequently.

A method to classify fault behavior by an arbitrary predicate under any combinational fault model have been published in [75]. This method, however, depends on a detailed model of the fault behavior, which turned out to be a difficult task without any detailed information about the selected FPGA architecture.

4.2.1.1 Implementation Details

Modifications of the toolchain were proposed to create and process the model of the Xilinx Spartan-3 architecture in cooperation with the Xilinx ISE design suit. The commercial ISE tool is used to perform front-end operations, i.e. design synthesis and real technology mapping. Output from this stage is then used as a primary input to the VPR to perform packing, placing and routing there. The ISE design flow is completed in parallel and its real architecture placement is used constrain this step in the VPR.

This keeps the last stage of the design flow "opened" and allows us to interact with the process. It also keeps the VPR processed design as close as possible to the real implementation of the test design to make the model more relevant. Using this proposed technique, a calibrated model for this method should be provided and more realistic estimates of dependability and safety of the design in question should be possible. These techniques have been earlier described, e.g. in [76]. The original VPR design flow and its modified part as a component of the proposed method can be seen in Figure 4.3.

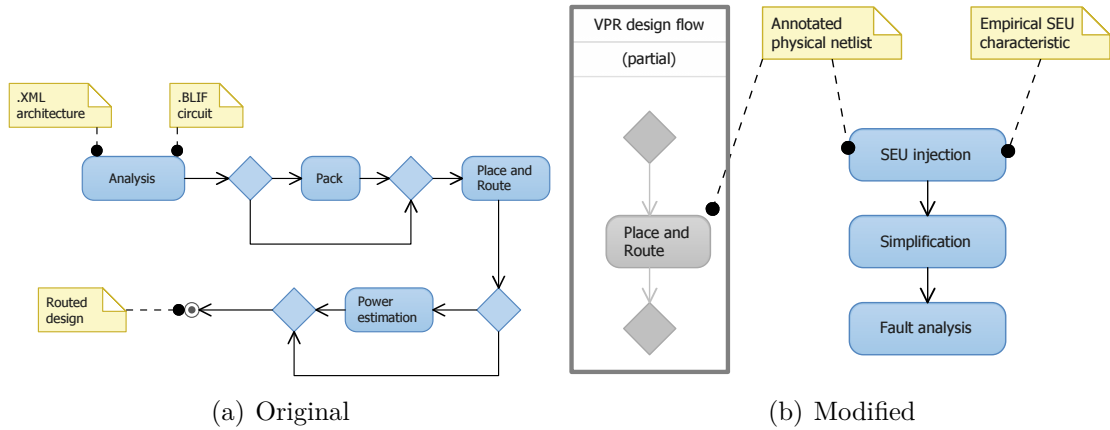(a) Original   (b) Modified

Figure 4.3: The proposed VPR design flow is shown here. The original VPR flow is shown on the left with all individual steps: an input file analysis, a primitives packing, a circuit placement and routing and an optional power estimation. On the right, a modified design flow is presented, where the place and route part of the original flow is constrained by results of a commercial EDA tool.

4.2.1.2   Results

We studied the VTR toolchain and started its proposed modification towards Xilinx ISE integration. The preliminary VPR architecture based on the Xilinx Spartan-3 architecture was created. Example results of the VPR tool can be seen in Appendix A. An initial placement of the circuit is shown in Figure A.1, final placement with unrouted nets is shown in Figure A.2 and finally fully routed design can be seen in Figure A.3.

   This implementation was not finished, as it was found too difficult to interface it properly with the Xilinx ISE toolchain and also to accurately described the selected Spartan-3 architecture for the VPR tool. Another approach was tested and it was found more appropriate for the given task (see the following text).

4.2.2   RapidSmith Implementation

The RapidSmith framework described in section 3.4 was selected as a second option for implementing our proposed method. The framework was extended and complemented with other tools, both commercial and open source. As this option was successful at the end, it is described in more detail in Chapter 5.

Method Implementation

A thorough exploration of the topic of our work have been done and described in Chapter 2 and Chapter 3. The theoretical background has been studied and a summary of the existing research have been collected. Our proposed approach has been described in Chapter 4.

In this chapter we present a proof-of-concept implementation of our proposed method based on the RapidSmith framework. The Xilinx Spartan-3 chip is selected as a target FPGA, because it is well supported in Xilinx Design Language features and we already have previous experience with this family architecture. It is true that this chip family is about 15 years old, but the organization of the internal architecture follows nowadays FPGAs in a sufficient way for our implementation.

5.1 Concept of Saboteurs

First of all, we will describe a concept of "saboteurs" used in the following text. The saboteur is a term used in connection with electrical circuit fault analysis [77, 78]. It can be used in both digital and analogue circuits and the main idea is the same. The saboteur is an additional block included into a circuit and connected to one or more existing blocks. It can, once activated, simulate a given fault, such as stuck-at, open, bridge, etc. By a direct control over these saboteurs, a fault analysis management part can decide which fault should be introduced into the circuit and when.

In this thesis, the term saboteur is used for an additional signal and an accompanying logic introduced into an RTL description of a digital circuit. Two examples of such saboteur can be seen in Figure 5.1.

5.2 Model Creation

In our case, the Spartan-3 family documentation [79, 80, 81] was analyzed first and preliminary fault models were proposed. Out of those, following fault models were implemented for islands of logic:
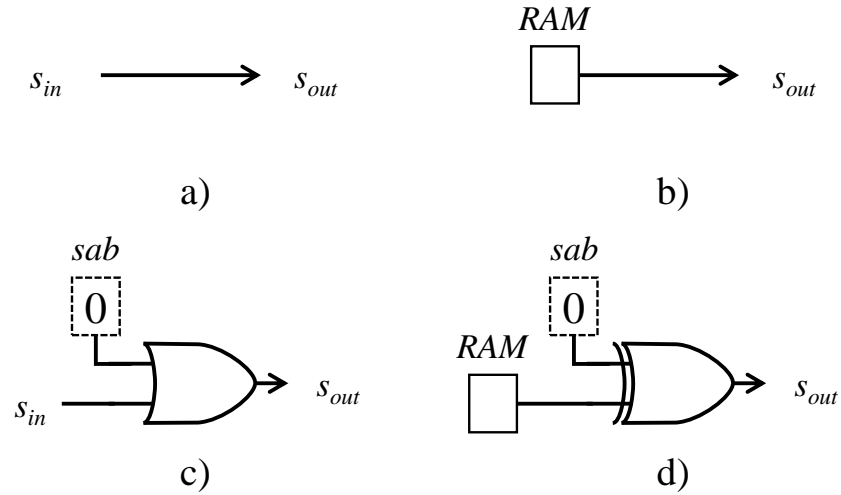
Figure 5.1: Two examples of a saboteur, a concept of circuit modification to gain the control over a fault injection. Original circuits are shown in the frame a) and b) representing a wire connecting a signal s_{in} into a signal s_{out} and a memory element with an output value s_{out} respectively. In the frame c), a saboteur for a stuck-at-1 fault is inserted to the original circuit from the frame a). The signal *sab* has a default value of 0 and only when activated by changing its value to 1, a fault on the s_{out} is introduced via an OR gate. Similarly in the frame d), a saboteur with a XOR gate is inserted to the output of the memory element. When activated, this saboteur can introduce a SEU fault, as the value s_{out} will be inverted.

○ **LUT Saboteur:** the 4-input look-up table used in the Spartan-3 family architecture is modeled as a multiplexer, where usual multiplexer inputs are connected to a constant 0 or 1 reflecting configuration memory bits containing the truth table of the implemented logic function. Usual multiplexer select inputs are connected to LUT inputs. For each configuration bit a "saboteur" is created (i.e. a signal with constant value of 0) and connected via an XOR gate with the original configuration bit output. A fault of the given LUT bit is simulated as a stuck-at-1 at the saboteur signal. See Figure 5.2 for an example implementation of the LUT saboteur.

○ **INV Saboteur:** inverting multiplexer is an element used in several types of islands of the given FPGA family (e.g. logic blocks, input/output block...). It is a two-in-one type multiplexer where both inputs are connected to the same signal: one directly and one via an inverter. The multiplexer select input is connected to the configuration memory cell which determines if the signal should be inverted (value 1 stored in the configuration memory) or not (value 0 stored). An example of such inverting multiplexer and its fault model with single saboteur is shown in Figure 5.3.
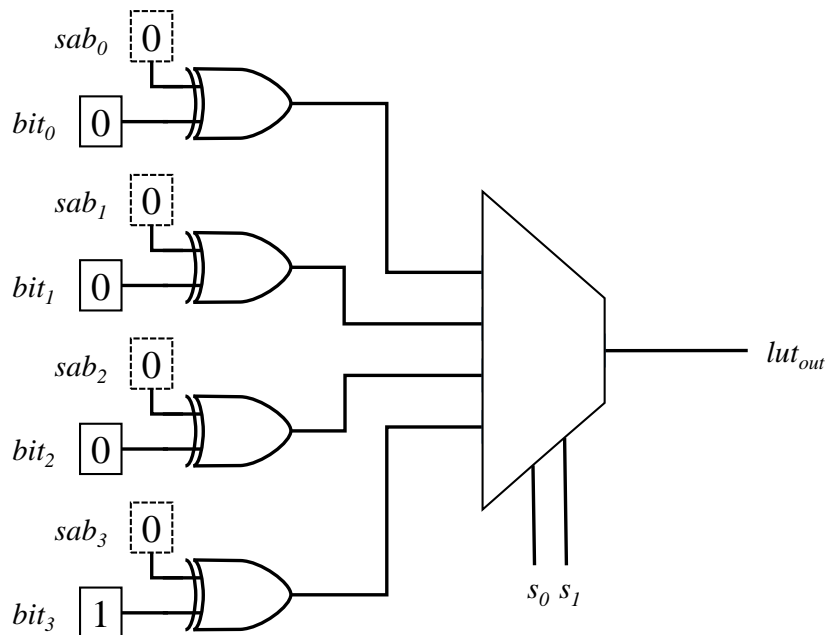
Figure 5.2: Here, a two variable function AND implemented in a 2-LUT with added saboteurs is shown. The truth table of the implemented function is encoded in bits bit_i and saboteur signals are represented by dashed boxes sab_i. LUT inputs are connected to multiplexer select inputs s_n and the resulting function is available on the signal lut_{out}. The actual SEU fault is simulated as a stuck-at-1 fault on a corresponding sab_i signal.

Memory elements were not modeled in this example implementations, as data memory soft error mitigation was not the primary scope of this work.

As a next step, the Xilinx FPGA Editor tool (part of the Xilinx ISE toolchain) was used to refine already implemented models and to help in creation of routing resources fault models. Also other publications were studied, e.g. [82] describing general routing models and [83] describing routing details of Xilinx Virtex-II family, which was found to be very similar to routing resources in Spartan-3.

The routing resources in Spartan-3 family were found to be organized as shown in Figure 2.1. Each logic block (LB, in Xilinx terminology CLB) is locally connected to the nearest switch box (SW) and SWs are connected between each other by different types of wires. Some wires run only between directly neighboring SWs, some span more rows or columns.

Inside the SW, an architecture of spare unidirectional multiplexers on SW outputs has been found. An average switch box has several hundred inputs and outputs, but only some connections between them are possible. When an output pin of the SW is considered, amount of other pins which can be connected vary between 1 and 25 having usual values 1, 8, 9, 14, 15, 24 and 25. We presumed that SW output connections are realized with
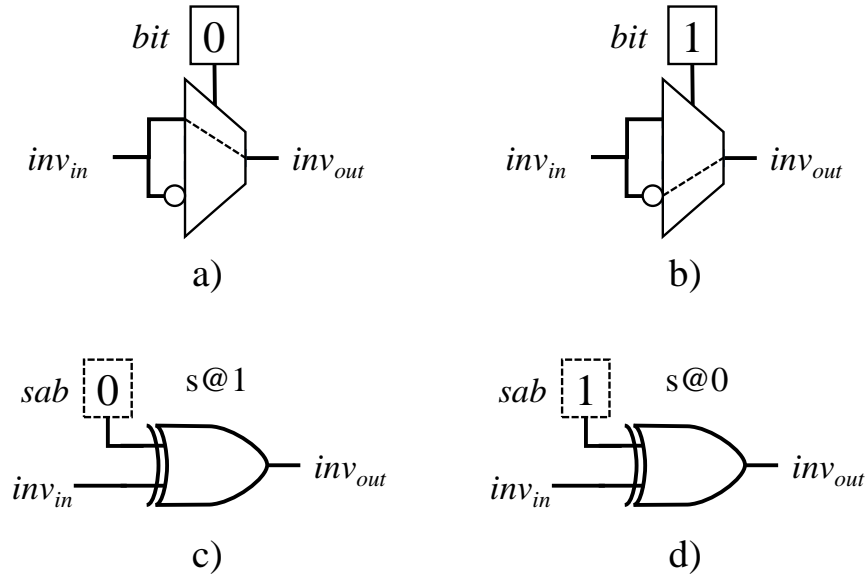
Figure 5.3: The example of an inverting multiplexer element present in the Xilinx Spartan-3 family architecture and its fault model implementation. inv_{in} is an input net, inv_{out} is an output net, *bit* is a configuration memory element and *sab* is the saboteur signal. The inverting multiplexer implementation in the FPGA is shown in frames a) and b) for the non-inverting and inverting configuration respectively. The fault model implementation is shown in frames c) and d) for the non-inverting and inverting configuration respectively. In non-inverting configuration, for example, the saboteur signal is set to 0 by default and a SEU fault in the configuration bit is simulated by a stuck-at-1 fault on the saboteur signal.

multiplexers of different sizes (from 2 to 5 select inputs) and not all multiplexer outputs are always used. By manual modification of the XDL definition of a placed and routed design, we discovered that when unused SW pin is selected in the multiplexer, its output has the value of logic high.

We have proposed two different fault models for the switch box routing interconnection described above. The first fault model is called "open" and realizes the situation when an unused SW pin is connected to the selected output. It can be either other regular SW pin which is not used in actual design, or it can be an unused multiplexer input which is not connected to any actual SW pin. To realize this fault model, a saboteur network is created with default value 0 and is ORed to the selected SW output pin. Defect causing the "open" fault is then simulated as stuck-at-1 on this saboteur signal.

The second fault model is called "swap" and realizes the situation where another SW pin already used in the design is selected in the multiplexer instead of the original input. This fault model is realized with a saboteur signal defaulting to 0 and, when activated by stuck-at-1 in the simulation, changing actual multiplexer output to the other signal value. An simplified example of the routing multiplexer architecture can be seen in Figure 5.4.
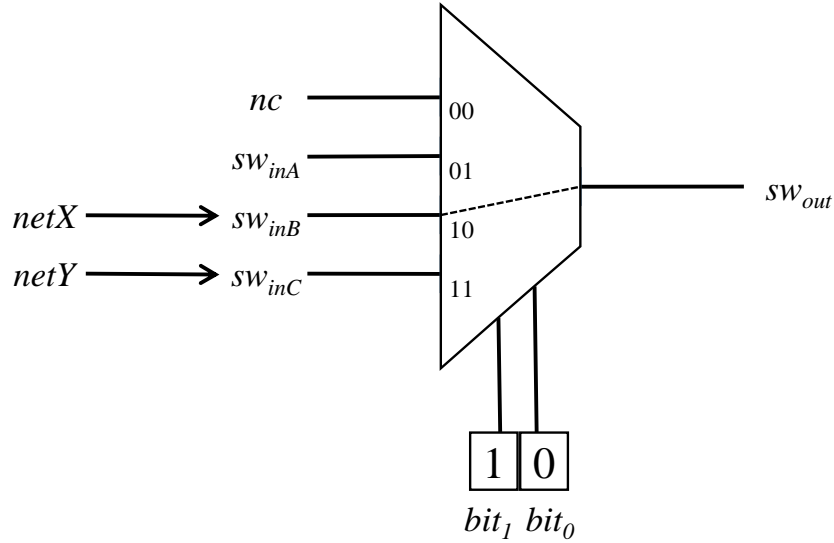
Figure 5.4: A two select inputs multiplexer at the SW output is shown here as an example of a routing switch box architecture. The SW output pin is labeled sw_{out} and configuration memory bits selecting the actual connection are labeled bit_i. It is possible to connect three other SW pins to this SW outputs pin, namely sw_{inA}, sw_{inB} and sw_{inC}. Two of them are actually members of some net in the design implemented by this example ($netX$ and $netY$). The fourth input to the multiplexer (with the address 00) is not used in this architecture. As indicated, the SW pin sw_{inB} is actually configured to be selected. A single soft error can cause either "open" fault (in case bit_1 is affected) or "swap" fault (when bit_0 is affected.)

Known all connection possibilities (available in the XDL resource file) and the actual configuration selected for a given implementation (available in the XDL design file) a probability of individual faults can be calculated taking into account multiplexer size and all possible connection it can implement in the actual design.

The last step of the model creation is to determine actual cross-section of the resources used in our fault models. For this step, data from dedicated irradiation campaign and other sources were obtained, as presented in section 3.2. Because we didn't obtain a specific cross-section for individual configuration memory sub-categories, we have combined several data in the following way.

We have chosen to calibrate our model for a proton beam with the selected energy of $E = 24.88\,\text{MeV}$, i.e. after the expected onset of the saturation. According to the measurement mentioned in Table 3.2, this corresponds to an average absolute cross-section of the configuration memory $\sigma_a = 1.64\,\text{cm}^2\,\text{bit}^{-1}$. Data in Table 3.3 were measured for an alpha irradiation, but we can assume, that relative sensitivity can be the same also for a proton irradiation. Our model do not distinguish between SEU flipping bit from 0 to 1 or from 1 to 0, thus we have used averaged values for both transition. Our model also do not distinguish between different types of PIPs, so a weighted average (with respect to number

FPGA resource	σ_r [–]	σ_a [$10^{-14}\,\mathrm{cm^2\,bit^{-1}}$]
LUT	1.145	2.67
MUX	0.535	1.25
Slice configuration	0.845	1.97
PIP	0.638	1.49

Table 5.1: Relative (σ_r) and absolute (σ_a) cross-sections for the configuration memory of individual FPGA resources calculated for the proton energy $\mathrm{E_p} = 24.88\,\mathrm{MeV}$.

of configuration bits) of values for FPGA resources "Decoded PIP" and "Non-decoded PIP" stated in Table 3.3 was used for all interconnect resources. In our test designs, no BRAM memories were used, so this value was omitted. An average relative sensitivity $\sigma_r = 0.702$ was calculated as a weighted average (with respect to a number of configuration bits) from all relative sensitivities to be used.

Relative sensitivities for different configuration memory sub-categories we obtained by processing described above are summarized in Table 5.1. Absolute values of cross-section (listed in the same table) were calculated from relative values and average cross-section σ_a according to the following equation:

$$\sigma_{a,i} = \sigma_{r,i} \cdot \frac{\sigma_a}{\sigma_r} \tag{5.1}$$

where $\sigma_{a,i}$ is absolute and $\sigma_{r,i}$ is relative cross-section of individual configuration memory sub-category.

5.3 Model Usage

Here we will describe a toolchain, which we used for the simulation of a design to be characterized, i.e. the example usage of our proposed model. Individual steps of the implementation and intermediate data being produced and used will be characterized. Overview of the whole model simulation flow can be seen in Figure 5.5. Command line scripts used in the described implementation can be found in Appendix B.

5.3.1 Xilinx ISE Design Flow

The Xilinx ISE 14.7 is the last version of the FPGA integrated design flow tool provided by Xilinx for a development targeting their FPGAs. It also supports XDL features described in section 2.8. Its development was stopped in 2013 in favor of the Xilinx Vivado suit, its successor. The Vivado suit does not support XDL feature and also does not support our selected FPGA family.
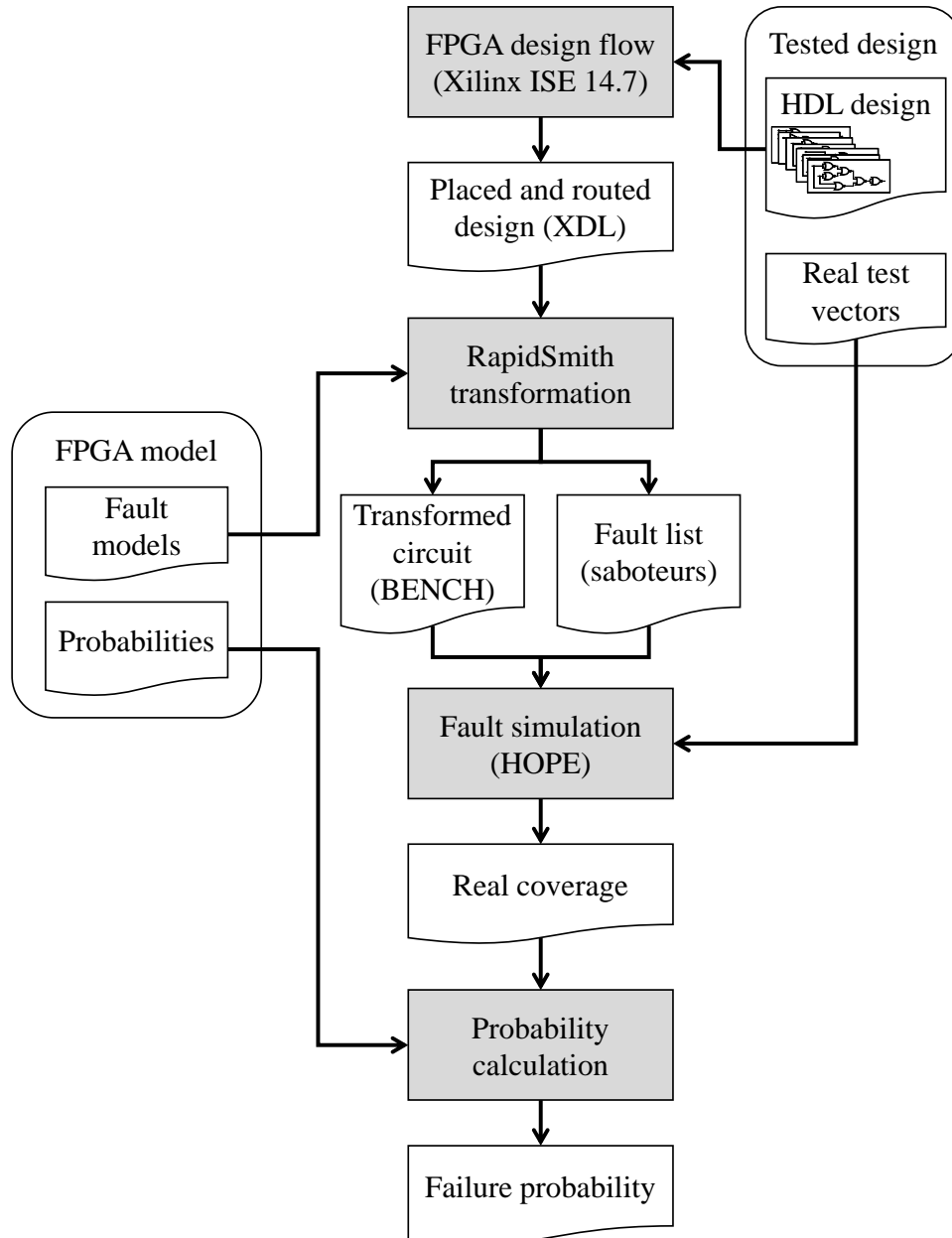
Figure 5.5: The example implementation of the simulation flow is shown here. Individual steps representing data processing are shown with a light gray background, intermediate data are represented by waved-bottom blocks.

Design synthesis needs to be constrained in such way, that no BRAM memories, DSP blocks or any other specialized primitives are used. It is hard to prepare accurate fault models for those primitives, as low amount of information about their inner structure is available. For this reason, parsing of those primitives is not implemented in later stages of our implement.

The tested design should be processed fully by the Xilinx ISE toolchain, i.e. commands `xst`, `ngdbuild`, `map`, and `par` should be run. The resulted placed and routed design then needs to be exported into XDL format containing detailed information about implemented design in form of a resource graph. Individual resources in the XDL file represents both logic elements and routing resources found in the selected FPGA family. A command line script used for this step of the implementation can be seen in Listing B.3.

5.3.2 RapidSmith Transformation

The RapidSmith framework was already described in section 3.4. In our implementation, only the XDL import function is used for the design analysis. In addition, a custom Java code `Xdl2Bench` was written to extend RapidSmith functionality for this step.

The XDL file is parsed by the RapidSmith framework into its internal data structure and this structure is then handed to our code. The `Xdl2Bench` code creates all necessary fault models with saboteurs and translates it into the BENCH format suitable for the next step. It also exports all generated saboteurs into a separate file, so only those stuck-at fault associated with generated saboteurs can be tested in the next step.

A pseudo-code algorithm of this step can be seen in Listing 5.1. On the line 5, the function `findTranslator()` selects proper translator depending on the primitive being processed. Five different translators were implemented in our code to support following Spartan-3 FPGA primitives: `BUFGMUX`, `IOB`, `SLICEL`, `SLICEM`, and `VCC`. In the `IOB` primitive only unidirectional combinational paths are implemented, i.e. IOB packed registers and tristate pins are not supported. In the `SLICEM` primitive only primary function of LUTs are implemented, i.e. translation is not available for shift registers or embedded memory mapped into LUTs. The translation of other primitives is otherwise fully implemented. Only saboteurs described in section 5.2 are implemented and inserted into generated BENCH description.

5.3.3 HOPE Simulation

The third step is a fault simulation done with the HOPE tool described in section 3.4. Main inputs for this simulation are a BENCH design file and a list of saboteur faults generated in the previous step. The fault simulation is then run and only specific faults on selected saboteur signals are simulated, i.e. only fault inserted to the circuit by our `Xdl2Bench` tool.

When supplied (which is a preferred way), a set of real test vectors specific to the given design can be also specified for the simulation. When no test vectors are supplied, a random simulation of selected length can be performed. We didn't used the HOPE test

```
1   RapidSmith.readXdl()
2   Xdl2Bench.convertToBench:
3     foreach xdlInstance:
4       findTranslator()
5       translate()
6         insertSaboteurs()
7     foreach xdlNet:
8       foreach PIP:
9         makeNewSegment()
10        connectNewSegment()
11          insertSaboteurs()
12    purgeClockNet()
13    checkNetsConsistency()
14  Xdl2Bench.saveBench()
15  Xdl2Bench.saveSaboteurFaults()
```

Listing 5.1: Pseudo-code of the `Xdl2Bench`

vector random generation for cased where no real test vectors were supplied. Virtually all sequential circuits contains at least one resetting input (the GSR signal) which would be activated frequently, if the internal HOPE random test vector generator would be used. This would limit the sequential depth of the circuit which can be tested, so our own random test generator script was written in the following way. A constant number of four resetting test vectors were first generated for all reset inputs (global set/reset signal (GSR) and possibly the `reset` input). Then a specific number of random test vectors were generated for all other (non-resetting) inputs.

The main output file from this step is a list of undetected faults.

5.3.4 Probability Calculation

For the last step, the probability calculation, an ad-hoc Python script was created. The result obtained from the simulation is a list of undetected faults (i.e. those, which cannot cause a design failure), the list of all fault is available from the RapidSmith transformation step. Another input needed for the probability calculation is a list of individual sensitivities, i.e. cross-section of corresponding configuration bits. From these inputs, an overall probability of failure for the tested design is calculated in assumption of a single SEU affecting a design (i.e. MBU are not taken into account). Probabilities of individual faults are combined to a total circuit probability as not mutually exclusive events according to the following formula:

$$P(x \cup y) = P(x) + P(y) - P(x \cap y) \tag{5.2}$$

A pseudo-code algorithm of this step can be seen in Listing 5.2.

```
1  read_sensitivities ()
2  all_faults = read_all_faults ()
3  faults = all_faults - read_undetected_faults ()
4  total_probability = 0
5  foreach fault :
6    probability = get_sensitivity ( fault )
7    total_probability = ( total_probability + probability - ↩
        total_probability * probability )
8  report total_probability
```

Listing 5.2: Pseudo-code of the probability calculation

CHAPTER 6

Benchmark Tests

Benchmarking tests performed with our implementation are described in this chapter and results are shown and discussed.

6.1 Runtime Parameters

The example implementation of the presented method was tested on several benchmarks from Politecnico di Torino subset of ITC99 benchmarks (labeled `b*`) [84] and also on the reduced design we used in irradiation tests (labeled `irrad`) [A.2] (only 5 `hexchain` units were generated for this test).

All tests were run on a personal computer with Intel Core2 T5600 processor running at 1.83 GHz with 3 GB of RAM.

The Register-Transfer Level (RTL) description in the VHDL language was used as an HDL input and synthesized and analyzed by Xilinx ISE development tool in version 14.7. The baseline FPGA part number used for this implementation was XC3S200 (i.e. smallest Spartan-3 chip), only designs `b20`, `b21` and `b22` were synthesized into XC3S400 chip.

For the JVM running RapidSmith framework and its extension `Xdl2Bench`, a maximum heap size was set to 1.2 GB.

For designs `b02` and `irrad` a deterministic test vectors were prepared activating all possible states of the circuit. This was possible because inner structure of those designs were known in detail. A random test patterns were generated for other designs. In case of design `b12`, the input signal `start` was marked as "spare" in test pattern generation process, so its probability to be set to 1 was only 1/256.

Actual cross-sections used for probability calculation are those mentioned in Table 5.1. For LUT faults, the cross-section "LUT" was used, for inverting multiplexer faults, the cross-section "MUX" was used and for both routing faults the cross-section "PIP" was used. Probability of routing faults (open and swap) were calculated according to the actual size of the multiplexer implementing corresponding PIP in the FPGA (as mentioned in section 5.2).

Design name	Resources used				
	DFF	LUT	IOB	Slices	Slices [%]
b02	4	4	4	2	< 1
b03	35	67	10	45	2
b04	67	143	21	75	4
b05	45	243	39	136	7
b12	141	352	13	216	11
b14	219	2500	88	1286	67
b20	435	4811	56	2505	70
b21	435	4990	56	2592	72
b22	619	6091	56	3173	89
irrad	320	320	6	160	8

Table 6.1: Resource usage characteristics of tested designs

Design name	Transform. time [s]	BENCH gates	Faults generated				
			LUT	INV	open	swap	Σ
b02	1.44	589	46	3	34	24	113
b03	2.45	18 288	860	49	637	2496	4042
b04	3.00	30 049	1588	64	1085	3699	6436
b05	3.81	57 180	3192	140	2180	7066	12 578
b12	5.19	90 913	4278	247	3202	12 208	19 935
b14	24.69	698 992	27 532	816	22 874	105 528	156 750
b20	46.04	1 287 677	53 310	1582	42 060	190 228	287 180
b21	46.38	1 293 050	52 584	1556	42 287	191 659	288 086
b22	65.16	1 721 586	65 206	2345	54 190	264 707	386 448
irrad	4.07	57 425	5120	164	2004	4051	11 370

Table 6.2: Runtime results of the RapidSmith transformation of tested designs

6.2 Results

Summary of designs' characteristics obtained from the Xilinx ISE design flow are shown in Table 6.1 in terms of FPGA resources used.

Runtime results from the RapidSmith transformation can be found in Table 6.2. The transformation time is shown, the number of gates generated into a BENCH output file and also the number of fault inserted into the processed design. Numbers of faults are shown for individual fault category and also the sum is displayed.

When the HOPE simulator was used to process generated BENCH files, a difficulty was encounter on several designs. See section 6.3 for more details on this issue. For

Design	Number of random	Undetected faults	
name	test vectors	Normal run	Excluding "swap" faults
b02	(deterministic)	6	6
b03	1000	—★	874
b04	500	—★	1223
b05	1000	—★	3736
b12	2000	—★	4715
b14	4000	—★	—★
b20	8000	—★	—★
b21	8000	—★	—★
b22	8000	—★	—★
irrad	(deterministic)	31	0

★ not calculated, as HOPE simulator was crashing in those cases

Table 6.3: Results of the HOPE simulation step

this reason, another RapidSmith transformation was done for all designs excluding "swap" faults suspected from causing this issue. Results of HOPE simulation are summarized in Table 6.3.

The probability calculation was run on all generated faults at the first, i.e. directly on results from the step "RapidSmith transformation". These results correspond to the individual design failure probability if all faults would be observable. Then individual probabilities were calculated based on the result from the "HOPE simulation" step, i.e. only observable faults were taken into account. Same was calculated also for a reduced set of generated faults, as described above (without "swap" faults). Results from this step are presented in Table 6.4 as a device cross-section of the tested design on the given FPGA.

6.3 HOPE Exception Issues

A problem was encounter with HOPE simulator when running test of several benchmarks. The simulator tool was regularly crashing on those faulty runs with a general exception mentioned in Listing 6.1.

```
1  [main] hope 3428 handle_exceptions: Exception: STATUS_ACCESS_VIOLATION
2  [main] hope 3428 open_stackdumpfile: Dumping stack trace to hope.exe.←
       stackdump
```

Listing 6.1: The example exception thrown by HOPE on some benchmarks

Combinational loops in the implemented designs enriched by fault models were identified as a root cause of some of those crashes, specifically "swap" fault models. These fault models can connect a different signal instead of original one, so a combinational loop can

Design name	σ_{device} [cm^2]		
	All faults	Observable faults	Reduced observable faults
b02	6.229×10^{-11}	6.213×10^{-11}	5.975×10^{-11}
b03	1.154×10^{-9}	—★	5.710×10^{-10}
b04	1.990×10^{-9}	—★	1.233×10^{-9}
b05	3.782×10^{-9}	—★	1.436×10^{-9}
b12	5.589×10^{-9}	—★	2.502×10^{-9}
b14	3.568×10^{-8}	—★	—★
b20	6.729×10^{-8}	—★	—★
b21	6.815×10^{-8}	—★	—★
b22	8.572×10^{-8}	—★	—★
irrad	4.540×10^{-9}	4.538×10^{-9}	4.231×10^{-9}

★ not calculated, as no HOPE simulator output was available for those cases

Table 6.4: Results of the probability calculation step expressed as a device cross-section (a design failure rate)

be created in originally fully synchronous design. As a result, partly asynchronous circuit description is supplied to the HOPE simulator, which cannot proceed.

A reduced RapidSmith transformation was run as described in previous section and HOPE simulator step was repeated. No combinational loops were present in those circuits, but still the simulator was crashing on few of them. The real reason for this was not discovered, but the size of tested BENCH files (over 10 MB) is suspected. Proper mitigation of these issue is subject of future research.

Conclusions

In the Chapter 1 the need of an accurate model of FPGA architectures and the problem of creation properly calibrated model for a device with a partially unknown structure have been discussed.

In the Chapter 2 a theoretical background of the discussed problem has been summarized. Information on digital circuits with focus on the CMOS technology, FPGA structures, radiation and radiation impacts on CMOS devices with emphasis on FPGA circuits have been composed. Also the classification of radiation induced defects has been presented.

In the Chapter 3 a brief market overview of radiation tolerant FPGA devices has been stated with a summary of what manufacturers nowadays use to increase the reliability of their devices. Also works related to our research and results of some already realized irradiation measurements have been mentioned.

In the Chapter 4 our proposed method has been presented. The FPGA model has been described and steps to create and use this module have been discussed. Two implementation based on two different frameworks have been proposed and one of them (based on the Java framework RapidSmith) has been selected.

In the Chapter 5 a proof-of-concept implementation of the proposed method has been presented. Individual steps using various commercial and open source tools have been described and input and output data have been pointed out.

In the Chapter 6 a set of benchmarks has been used to characterize our implementation. Results have been presented and discussed.

7.1 Summary

The method for a simulation-based evaluation of the radiation induced soft error impact to the SRAM-based FPGA configuration memory was presented in this thesis. The universal method concept was introduced consisting of a one-time step of model creation and calibration and a repetitive step of model usage for the digital design simulation. A precise characteristic in terms of overall failure rate can be calculated for an arbitrary digital

design implemented on a given FPGA providing well calibrated model is prepared for that FPGA.

As a proof-of-concept implementation of the presented method, a toolchain was constructed from Xilinx ISE design tools, the RapidSmith framework, the HOPE fault simulator and our own codes to model Xilinx Spartan-3 FPGA family. Individual steps of the model creation and usage have been described and results from the example implementation benchmarking were shown.

Those results and also method itself can be provided to other colleagues from out department for their research. We also assume this work to be released available to other research institutions.

As part of this research, the author was also involved in a collaboration with Tomáš Vaňát from CTU, FIT, Jozef Ferencei from NPI, CAS, and Filip Křížek from NPI, CAS. The cyclotron facility described in section 2.7 has been used for conducting electronics irradiation tests (some described in section 3.2) and beam control and diagnostic infrastructure of the cyclotron have been improved as described in [A.2, A.1].

7.2 Contributions of the Dissertation Thesis

In particular, the main contributions of the dissertation thesis are as follows:

1. The method for a simulation-based evaluation of radiation induced soft errors in the SRAM-based FPGA configuration memory based on parameters obtained from experiments on the real hardware is proposed.

2. The proof-of-concept toolchain for the chosen FPGA family model creation and usage through the simulation following the proposed method is implemented.

3. Individual implementation steps and intermediate data are described and test results obtained from a set of benchmarks are presented.

7.3 Future Work

The author of the dissertation thesis suggests to address the following topics in the future work:

○ The simulation of a circuit containing combinational loops needs to be solved first. This phenomenon is inherently present in the fault models of FPGA routing resources. One possibility can be to consider an asynchronous fault simulation.

○ Fault models of user memories can be included into the model processing to cover also design data errors arising from the radiation induced soft errors.

○ The proof-of-concept implementation can be extended to cover all fault of the selected FPGA architecture to provide a complete evaluation tool.

○ Better approach for the random test vector generation can be exploited for cases where real test vectors are not available, e.g. an Automatic Test Pattern Generation (ATPG) can be employed.

○ Models created with the presented method can be used for analysis of various FPGA architectures and modifications leading to increasing of reliability of the FPGA as a platform can be proposed.

Bibliography

1. BERNARDI, P.; REORDA, M. S.; STERPONE, L.; VIOLANTE, M. On the evaluation of SEU sensitiveness in SRAM-based FPGAs. In: *Proceedings of the 10th IEEE International On-Line Testing Symposium.* 2004-07, pp. 115–120. Available from DOI: 10.1109/OLT.2004.1319668.

2. KUBALÍK, Pavel; KVASNIČKA, Jiří; KUBÁTOVÁ, Hana. Fault Injection and Simulation for Fault Tolerant Reconfigurable Duplex System. In: *2007 IEEE Design and Diagnostics of Electronic Circuits and Systems (DDECS).* 2007-04, pp. 1–4. Available from DOI: 10.1109/DDECS.2007.4295312.

3. KUON, Ian; TESSIER, Russell; ROSE, Jonathan. FPGA Architecture: Survey and Challenges. *Foundations and Trends in Electronic Design Automation.* 2007, vol. 2, no. 2, pp. 135–253. Available from DOI: 10.1561/1000000005. Modern survey of FPGA Architecture.

4. LEMIEUX, Guy G.; LEWIS, David M. Analytical Framework for Switch Block Design. In: GLESNER, Manfred; ZIPF, Peter; RENOVELL, Michel (eds.). *Field-Programmable Logic and Applications: Reconfigurable Computing Is Going Mainstream (FPL) 2002.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 122–131. ISBN 978-3-540-46117-3.

5. SELTZER, S. M.; BARTLETT, D. T.; BURNS, D. T.; DIETZE, G.; MENZEL, H.-G.; PARETZKE, H. G.; WAMBERSIE, A. ICRU report 85: Fundamental Quantities and Units for Ionizing Radiation. *Journal of the ICRU.* 2011-04, vol. 11.

6. BAKER, D. N. et al. A Long-Lived Relativistic Electron Storage Ring Embedded in Earth's Outer Van Allen Belt. *Science.* 2013. Available from DOI: 10.1126/science.1233518.

7. NATIONAL AERONAUTICS AND SPACE ADMINISTRATION. NASA Probes Studying Earth's Radiation Belts to Celebrate Two Year Anniversary. *Press release.* 2014-08-29. https://www.nasa.gov/press/2014/august/nasa-probes-studying-earth-s-radiation-belts-to-celebrate-two-year-anniversary/.

8. BAUMANN, R.C. Radiation-induced soft errors in advanced semiconductor technologies. *Device and Materials Reliability, IEEE Transactions on.* 2005, vol. 5, no. 3, pp. 305–316. ISSN 1530-4388. Available from DOI: 10.1109/TDMR.2005.853449.

9. WROBEL, F.; GASIOT, J.; SAIGNE, F.; TOUBOUL, A.D. Effects of atmospheric neutrons and natural contamination on advanced microelectronic memories. *Applied Physics Letters.* 2008, vol. 93, no. 6. ISSN 0003-6951. Available from DOI: 10.1063/1.2971203.

10. AUTRAN, Jean-Luc; SEMIKH, Sergey; MUNTEANU, Daniela; SERRE, Sébastien; GASIOT, Gilles; ROCHE, Philippe. Numerical Simulation — From Theory to Industry. In: InTech, 2012-09, chap. Soft-Error Rate of Advanced SRAM Memories: Modeling and Monte Carlo Simulation, pp. 309–336. ISBN 978-953-51-0749-1. Available from DOI: 10.5772/50111.

11. CLAEYS, C.L.; SIMOEN, E. *Radiation Effects in Advanced Semiconductor Materials and Devices.* Springer-Verlag Berlin Heidelberg, 2002. Springer Series in Materials Science. ISBN 978-3-540-43393-4. Available from DOI: 10.1007/978-3-662-04974-7.

12. FELIX, J.A; SCHWANK, J.R; FLEETWOOD, D.M; SHANEYFELT, M.R; GUSEV, E.P. Effects of radiation and charge trapping on the reliability of high-κ gate dielectrics. *Microelectronics Reliability.* 2004, vol. 44, no. 4, pp. 563–575. ISSN 0026-2714. Available from DOI: 10.1016/j.microrel.2003.12.005.

13. KANG, A.Y.; LENAHAN, P.M.; CONLEY, J.F. The Radiation Response of the High Dielectric-Constant Hafnium Oxide/Silicon System. *Nuclear Science, IEEE Transactions on.* 2002, vol. 49, no. 6, pp. 2636–2642. ISSN 0018-9499. Available from DOI: 10.1109/TNS.2002.805334.

14. ATMEL CORPORATION. *Aerospace Products Radiation Policy.* 2003. http://aerosupport.atmel.com/Atmel/doc4170.pdf.

15. BINDER, D.; SMITH, E.C.; HOLMAN, A. B. Satellite Anomalies from Galactic Cosmic Rays. *Nuclear Science, IEEE Transactions on.* 1975, vol. 22, no. 6, pp. 2675–2680. ISSN 0018-9499. Available from DOI: 10.1109/TNS.1975.4328188.

16. BENEDETTO, J.M.; EATON, P.H.; MAVIS, D.G.; GADLAGE, M.; TURFLINGER, T. Digital Single Event Transient Trends With Technology Node Scaling. *Nuclear Science, IEEE Transactions on.* 2006, vol. 53, no. 6, pp. 3462–3465. ISSN 0018-9499. Available from DOI: 10.1109/TNS.2006.886044.

17. WHEATLEY, C.F.; TITUS, J.L.; BURTON, D.I. Single-event gate rupture in vertical power MOSFETs; an original empirical expression. *Nuclear Science, IEEE Transactions on.* 1994, vol. 41, no. 6, pp. 2152–2159. ISSN 0018-9499. Available from DOI: 10.1109/23.340556.

18. TITUS, J.L.; JAMIOLKOWSKI, L.S.; WHEATLEY, C.F. Development of cosmic ray hardened power MOSFET's. *Nuclear Science, IEEE Transactions on.* 1989, vol. 36, no. 6, pp. 2375–2382. ISSN 0018-9499. Available from DOI: `10.1109/23.45451`.

19. BELLATO, M. et al. Evaluating the effects of SEUs affecting the configuration memory of an SRAM-based FPGA. In: *Proceedings Design, Automation and Test in Europe Conference and Exhibition.* 2004-02, vol. 1, pp. 584–589. ISSN 1530-1591. Available from DOI: `10.1109/DATE.2004.1268908`.

20. REORDA, M. Sonza; STERPONE, L.; VIOLANTE, M. Multiple errors produced by single upsets in FPGA configuration memory: a possible solution. In: *European Test Symposium (ETS'05).* 2005-05, pp. 136–141. ISSN 1530-1877. Available from DOI: `10.1109/ETS.2005.29`.

21. DIRK, J.D.; NELSON, M.E.; ZIEGLER, J.F.; THOMPSON, A.; ZABEL, T.H. Terrestrial thermal neutrons. *Nuclear Science, IEEE Transactions on.* 2003, vol. 50, no. 6, pp. 2060–2064. ISSN 0018-9499. Available from DOI: `10.1109/TNS.2003.821587`.

22. MILLER, J.; TAYLOR, L.; ZEITLIN, C.; HEILBRONN, L.; GUETERSLOH, S.; DIGIUSEPPE, M.; IWATA, Y.; MURAKAMI, T. Lunar soil as shielding against space radiation. *Radiation Measurements.* 2009, vol. 44, no. 2, pp. 163–167. ISSN 1350-4487. Available from DOI: `10.1016/j.radmeas.2009.01.010`.

23. DEPARTMENT OF ACCELERATORS, NUCLEAR PHYSICS INSTITUTE OF THE CAS, V. V. I. *Home page.* 2018. `http://accs.ujf.cas.cz/`.

24. STURSA, J; BEJSOVEC, V; BORKOVA, A; KRIVANEK, M; MARES, J; TREJBAL, Z. The Axial Injection System of the Isochronous Cyclotron. In: *Proceedings of the 3. European particle accelerator conference. V. 2.* 1992.

25. CIHAK, M; STURSA, J; LACMAN, J. New Control System for the Isochronous Cyclotron U-120M. In: *Proceedings of the 15th International Conference on Cyclotrons and their Applications, Caen, France.* 1998.

26. VAŇÁT, Tomáš. *Physical Fault Injection and Monitoring Methods for Programmable Devices.* 2017-02. Dissertation thesis. Faculty of Information Technology, Czech Technical University in Prague.

27. XILINX, INC. *Xilinx Design Language.* 2000-07-07. Version 1.6.

28. BECKHOFF, C.; KOCH, D.; TORRESEN, J. The Xilinx Design Language (XDL): Tutorial and use cases. In: *ReCoSoC 2011, 6th International Workshop on Reconfigurable Communication-Centric Systems-on-Chip.* 2011-06, pp. 1–8. Available from DOI: `10.1109/ReCoSoC.2011.5981545`.

29. XILINX, INC. *Home page.* 2018. `http://www.xilinx.com/`.

30. DSCC MIL-STD-883 GROUP. *Test Method Standard, Microcircuits.* 2010. Technical report. Department of Defense, USA.

31. XILINX, INC. *Virtex-4 and Virtex-5 QV FPGA CF Package Assembly Location Change.* 2018. http://www.xilinx.com/support/documentation/customer_notices/xcn13005.pdf.

32. XILINX, INC. *Xilinx UG496, Virtex-4 QV FPGA Ceramic Packaging and Pinout Specifications.* 2008.
http://www.xilinx.com/support/documentation/user_guides/ug496.pdf.

33. XILINX, INC. *TMRTool.* 2018.
https://www.xilinx.com/products/design-tools/tmrtool.html.

34. DAVID P. SCHULTZ, San Jose CA; SURESH M. MENON, Sunnyvale CA; EUNICE Y. D. HAO, Saratoga CA; JASON R. BERGENDAHL, Campbell CA; JIAN TAN, Milpitas CA. *Digitally controlled impedance for I/O of an integrated circuit device.* US6445245.

35. INTEL CORPORATION. *Intel FPGAs and Programmable Devices - Intel FPGA.* 2018. https://www.intel.com/content/www/us/en/products/programmable.html.

36. MICROSEMI CORPORATION. *Home page.* 2018. http://www.microsemi.com/.

37. BERG, M. et al. An Analysis of Single Event Upset Dependencies on High Frequency and Architectural Implementations within Actel RTAX-S Family Field Programmable Gate Arrays. *Nuclear Science, IEEE Transactions on.* 2006-12, vol. 53, no. 6, pp. 3569–3574. ISSN 0018-9499. Available from DOI: 10.1109/TNS.2006.886043.

38. WANG, J.J.; REZGUI, S. *RTAX-S TAMU Single Event Dielectric Rupture.* 2006. http://www.actel.com/documents/RTAX2000S_SEDR.pdf.

39. MICROSEMI CORPORATION. *RTSX-SU Radiation-Tolerant FPGAs (UMC).* 2012. http://www.actel.com/documents/RTSXSU_DS.pdf. Datasheet.

40. MICROCHIP TECHNOLOGY INC. *Home page.* 2018. http://www.microchip.com/.

41. ATMEL AEROSPACE. *Rad-Hard Integrated Circuits.* 2015-05-12. Product brochure. Atmel Corporation.

42. RAMASWAMY, S.; ROCKETT, L.; PATEL, D.; DANZIGER, S.; MANOHAR, R.; KELLY, C.W.; HOLT, J.L.; EKANAYAKE, V.; ELFTMANN, D. A radiation hardened reconfigurable FPGA. In: *Aerospace conference, 2009 IEEE.* 2009-03, pp. 1–10. Available from DOI: 10.1109/AERO.2009.4839506.

43. XILINX, INC. *Device Reliability Report, Second Half 2017.* 2018-05-03. User Guide.

44. VIOLANTE, M. et al. A New Hardware/Software Platform and a New 1/E Neutron Source for Soft Error Studies: Testing FPGAs at the ISIS Facility. *IEEE Transactions on Nuclear Science.* 2007-08, vol. 54, no. 4, pp. 1184–1189. ISSN 0018-9499. Available from DOI: 10.1109/TNS.2007.902349.

45. MANUZZATO, A.; GERARDIN, S.; PACCAGNELLA, A.; STERPONE, L.; VIOLANTE, M. Effectiveness of TMR-Based Techniques to Mitigate Alpha-Induced SEU Accumulation in Commercial SRAM-Based FPGAs. *IEEE Transactions on Nuclear Science*. 2008-08, vol. 55, no. 4, pp. 1968–1973. ISSN 0018-9499. Available from DOI: 10.1109/TNS.2008.2000850.

46. ROSE, Jonathan; LUU, Jason; YU, Chi Wai; DENSMORE, Opal; GOEDERS, Jeffrey; SOMERVILLE, Andrew; KENT, Kenneth B.; JAMIESON, Peter; ANDERSON, Jason. The VTR project: architecture and CAD for FPGAs from verilog to routing. In: *Proceedings of the ACM/SIGDA international symposium on Field Programmable Gate Arrays*. Monterey, California, USA: ACM, 2012, pp. 77–86. FPGA '12. ISBN 978-1-4503-1155-7. Available from DOI: 10.1145/2145694.2145708.

47. LUU, Jason et al. VTR 7.0: Next Generation Architecture and CAD System for FPGAs. *ACM Trans. Reconfigurable Technol. Syst.* 2014-06, vol. 7, no. 2.

48. VERILOG TO ROUTING (VTR) PROJECT. *Home page*. 2018. https://verilogtorouting.org/.

49. VERILOG TO ROUTING (VTR) PROJECT. *GitHub project page*. 2018. https://github.com/verilog-to-routing/vtr-verilog-to-routing.

50. LAVIN, Christopher; PADILLA, Marc; LAMPRECHT, Jaren; LUNDRIGAN, P.; NELSON, Brent; HUTCHINGS, Brad. RapidSmith: Do-It-Yourself CAD Tools for Xilinx FPGAs. In: *Proceedings of the 21th International Workshop on Field-Programmable Logic and Applications (FPL'11)*. 2011-09.

51. LEE, Hyung Ki; HA, Dong Sam. HOPE: an efficient parallel fault simulator for synchronous sequential circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 1996-09, vol. 15, no. 9, pp. 1048–1058. ISSN 0278-0070. Available from DOI: 10.1109/43.536711.

52. BRGLEZ, F.; BRYAN, D.; KOZMINSKI, K. Combinational profiles of sequential benchmark circuits. In: *IEEE International Symposium on Circuits and Systems,* 1989-05, vol. 3, pp. 1929–1934. Available from DOI: 10.1109/ISCAS.1989.100747.

53. REBAUDENGO, M.; REORDA, M. S.; VIOLANTE, M. A new functional fault model for FPGA application-oriented testing. In: *Proceedings of the 17th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT) 2002*. 2002-11, pp. 372–380. ISSN 1550-5774. Available from DOI: 10.1109/DFTVS.2002.1173534.

54. KVASNIČKA, Jiří; KUBALÍK, Pavel; KUBÁTOVÁ, Hana. Experimental SEU Impact on Digital Design Implemented in FPGAs. In: *2008 11th EUROMICRO Conference on Digital System Design Architectures, Methods and Tools*. 2008-09, pp. 100–103. Available from DOI: 10.1109/DSD.2008.119.

55. GRAHAM, Paul; CAFFREY, Michael; ZIMMERMAN, Jason; SUNDARARAJAN, Prasanna; JOHNSON, Eric. Consequences and categories of SRAM FPGA configuration SEUs. In: *Proceedings of the International Conference on Military and Aerospace Programmable Logic Devices*. 2003.

56. CESCHIA, M. et al. Identification and classification of single-event upsets in the configuration memory of SRAM-based FPGAs. *IEEE Transactions on Nuclear Science*. 2003-12, vol. 50, no. 6, pp. 2088–2094. ISSN 0018-9499. Available from DOI: 10.1109/TNS.2003.821411.

57. BHUVA, B.L.; PAULOS, J.J.; GYURCSIK, R.S.; KERNS, S.E. Switch-level simulation of total dose effects on CMOS VLSI circuits. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*. 1989, vol. 8, no. 9, pp. 933–938. ISSN 0278-0070. Available from DOI: 10.1109/43.35545.

58. KAUL, N.; BHUVA, B.L.; KERNS, S.E. Simulation of SEU transients in CMOS ICs. *Nuclear Science, IEEE Transactions on*. 1991, vol. 38, no. 6, pp. 1514–1520. ISSN 0018-9499. Available from DOI: 10.1109/23.124140.

59. BAZE, M. P.; BUCHNER, S.; BARTHOLET, W.G.; DAO, T.A. An SEU analysis approach for error propagation in digital VLSI CMOS ASICs. *Nuclear Science, IEEE Transactions on*. 1995, vol. 42, no. 6, pp. 1863–1869. ISSN 0018-9499. Available from DOI: 10.1109/23.489228.

60. ASADI, Ghazanfar; TAHOORI, Mehdi B. An analytical Approach for Soft Error Rate Estimation of SRAM-based FPGAs. In: *Proceedings of the Military and Aerospace Applications of Programmable Logic Devices (MAPLD 2004), Washington D.C.* 2004, pp. 2991–2994.

61. JOHNSON, Eric; WIRTHLIN, Michael J; CAFFREY, Michael. Single-event upset simulation on an FPGA. In: *Proceedings of the International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA)*. 2002, pp. 68–73. http://www.ericjohnsonweb.net/papers/BYU_ERSA_2002_1017er.pdf.

62. SCHOTT, Brian; BELLOWS, Peter; FRENCH, Matthew; PARKER, Robert. Applications of Adaptive Computing Systems for Signal Processing Challenges. In: *Proceedings of the 2003 Asia and South Pacific Design Automation Conference*. Kitakyushu, Japan: ACM, 2003, pp. 465–470. ASP-DAC '03. ISBN 0-7803-7660-9. Available from DOI: 10.1145/1119772.1119868.

63. ROLLINS, Nathan; WIRTHLIN, Michael J; CAFFREY, Michael; GRAHAM, Paul. Evaluating TMR techniques in the presence of single event upsets. In: *International Conference, Washington DC*. 2003, vol. 11.

64. JOHNSON, E.; CAFFREY, M.; GRAHAM, P.; ROLLINS, N.; WIRTHLIN, M. Accelerator validation of an FPGA SEU simulator. *Nuclear Science, IEEE Transactions on*. 2003, vol. 50, no. 6, pp. 2147–2157. ISSN 0018-9499. Available from DOI: 10.1109/TNS.2003.821791.

65. VIOLANTE, M.; STERPONE, L.; CESCHIA, M.; BORTOLATO, D.; BERNARDI, P.; REORDA, M.S.; PACCAGNELLA, A. Simulation-based analysis of SEU effects in SRAM-based FPGAs. *IEEE Transactions on Nuclear Science*. 2004-12, vol. 51, no. 6, pp. 3354–3359. ISSN 0018-9499. Available from DOI: 10.1109/TNS.2004.839516.

66. STERPONE, L.; VIOLANTE, M. A new analytical approach to estimate the effects of SEUs in TMR architectures implemented through SRAM-based FPGAs. *IEEE Transactions on Nuclear Science*. 2005-12, vol. 52, no. 6, pp. 2217–2223. ISSN 0018-9499. Available from DOI: 10.1109/TNS.2005.860745.

67. BERNARDESCHI, C.; CASSANO, L.; DOMENICI, A.; MASCI, P. A Tool for Signal Probability Analysis of FPGA-Based Systems. In: *COMPUTATION TOOLS 2011: The Second International Conference on Computational Logics, Algebras, Programming, Tools, and Benchmarking*. 2011. ISBN 978-1-61208-159-5. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.666.2033.

68. BERNARDESCHI, Cinzia; CASSANO, Luca; DOMENICI, Andrea. Failure probability of SRAM-FPGA systems with Stochastic Activity Networks. In: *14th IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems*. 2011-04, pp. 293–296. Available from DOI: 10.1109/DDECS.2011.5783098.

69. BERNARDESCHI, C.; CASSANO, L.; DOMENICI, A. Failure Probability and Fault Observability of SRAM-FPGA Systems. In: *2011 21st International Conference on Field Programmable Logic and Applications*. 2011-09, pp. 385–388. ISSN 1946-147X. Available from DOI: 10.1109/FPL.2011.75.

70. BERNARDESCHI, C.; CASSANO, L.; DOMENICI, A.; STERPONE, L. Accurate simulation of SEUs in the configuration memory of SRAM-based FPGAs. In: *2012 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*. 2012-10, pp. 115–120. ISSN 1550-5774. Available from DOI: 10.1109/DFT.2012.6378210.

71. BERNARDESCHI, C.; CASSANO, L.; DOMENICI, A.; STERPONE, L. ASSESS: A Simulator of Soft Errors in the Configuration Memory of SRAM-Based FPGAs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 2014-09, vol. 33, no. 9, pp. 1342–1355. ISSN 0278-0070. Available from DOI: 10.1109/TCAD.2014.2329419.

72. FULLER, Earl; CAFFREY, Michael; BLAIN, Phil; CARMICHAEL, Carl. Radiation Test Results of the Virtex FPGA and ZBT SRAM for Space Based Reconfigurable Computing. In: *Proceedings of the Military and Aerospace Programmable Logic Device (MAPLD) 1999*. 1999.

73. PI, TAO; CROTTY, PATRICK J., XILINX, INC. *FPGA lookup table with transmission gate structure for reliable low-voltage operation*. US6667635B1.

74. CHIRANIA, MANOJ, XILINX, INC. *Integrated circuit including a multiplexer circuit*. US7423452B1.

75. SCHMIDT, J.; FIŠER, P.; BALCÁREK, J. The Influence of Implementation Technology on Dependability Parameters. In: *Digital System Design (DSD), 2012 15th Euromicro Conference on.* 2012, pp. 368–373. Available from DOI: 10.1109/DSD.2012.128.

76. YU, Chi Wai. A Tool for Exploring Hybrid FPGAs. In: *Field Programmable Logic and Applications, 2007. FPL 2007. International Conference on.* 2007, pp. 509–510. Available from DOI: 10.1109/FPL.2007.4380703.

77. JENN, E.; ARLAT, J.; RIMEN, M.; OHLSSON, J.; KARLSSON, J. Fault injection into VHDL models: the MEFISTO tool. In: *Proceedings of IEEE 24th International Symposium on Fault- Tolerant Computing.* 1994-06, pp. 66–75. Available from DOI: 10.1109/FTCS.1994.315656.

78. BARAZA, J. C.; GRACIA, J.; GIL, D.; GIL, P. J. Improvement of fault injection techniques based on VHDL code modification. In: *Tenth IEEE International High-Level Design Validation and Test Workshop, 2005.* 2005-11, pp. 19–26. ISSN 1552-6674. Available from DOI: 10.1109/HLDVT.2005.1568808.

79. XILINX, INC. *Spartan-3 FPGA Family Data Sheet.* 2013-06-27.
http://www.xilinx.com/support/documentation/data_sheets/ds099.pdf.

80. XILINX, INC. *Spartan-3 Generation FPGA User Guide.* 2011-06-13.
https://www.xilinx.com/support/documentation/user_guides/ug331.pdf.

81. XILINX, INC. *Spartan-3 Generation Configuration User Guide.* 2015-01-27.
https://www.xilinx.com/support/documentation/user_guides/ug332.pdf.

82. LEMIEUX, Guy; LEWIS, David. Circuit Design of Routing Switches. In: *Proceedings of the 2002 ACM/SIGDA Tenth International Symposium on Field Programmable Gate Arrays.* Monterey, California, USA: ACM, 2002, pp. 19–28. FPGA '02. ISBN 1-58113-452-5. Available from DOI: 10.1145/503048.503052.

83. CLAUS, Christopher; ZHANG, Bin; HÜBNER, Michael; SCHMUTZLER, Christoph; BECKER, Jürgen; STECHELE, Walter. An XDL-based busmacro generator for customizable communication interfaces for dynamically and partially reconfigurable systems. In: *Proceedings of The 2nd International Workshop on Reconfigurable Computing Education.* Porto Allegre, Brasil, 2007-05.
http://xputers.informatik.uni-kl.de/RCeducation07/busmacro_RC_education_claus.pdf.

84. CORNO, F.; REORDA, M. S.; SQUILLERO, G. RT-level ITC'99 benchmarks and first ATPG results. *IEEE Design Test of Computers.* 2000-07, vol. 17, no. 3, pp. 44–53. ISSN 0740-7475. Available from DOI: 10.1109/54.867894.

Publications of the Author

Reviewed Publications Relevant to the Thesis

A.1. KŘÍŽEK, Filip et al. Irradiation setup at the U-120M cyclotron facility. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*. 2018-06, vol. 894, pp. 87–95. ISSN 0168-9002. Available from DOI: 10.1016/j.nima.2018.03.066.

A.2. VAŇÁT, Tomáš; POSPÍŠIL, Jan; KŘÍŽEK, Filip; FERENCEI, Jozef; KUBÁTOVÁ, Hana. A System for Radiation Testing and Physical Fault Injection into the FPGAs and Other Electronics. In: *Proceedings of the 18th Euromicro Conference on Digital System Design, DSD 2015*. Los Alamitos, US: IEEE Computer Society, 2015-08, pp. 205–210. ISBN 978-1-4673-8035-5. Available from DOI: 10.1109/DSD.2015.98. [cited: 5x].

A.3. POSPÍŠIL, Jan; SCHMIDT, Jan; FIŠER, Petr. New SEU Modeling by Architecture Analysis. In: *Work in Progress Session, the 16th Euromicro Conference on Digital System Design (DSD)*. Austria: Institute for Systems Engineering and Automation, 2013-09.

A.4. POSPÍŠIL, Jan; SCHMIDT, Jan. Modelováním poruch ke spolehlivým architekturám FPGA. In: *Počítačové architektury a diagnostika (PAD) 2013*. Pilsen, Czech Republic: Západočeská univerzita v Plzni, 2013-09, pp. 117–122. ISBN 978-80-261-0270-0. http://www.kiv.zcu.cz/pad2013/template/sbornikpad2013.pdf.

Remaining Publications Relevant to the Thesis

A.5. POSPÍŠIL, Jan; VAŇÁT, Tomáš; SCHMIDT, Jan. *Towards Trusted Devices in FPGA by Modeling Radiation Induced Errors — First ALT Results* [published internally]. Amsterdam, The Netherlands, 2014-09. Joint MEDIAN-TRUDEVICE Open Forum.

A.6. VAŇÁT, Tomáš; POSPÍŠIL, Jan; FERENCEI, Jozef. *Single Event Upset Characterization of Commercial Grade SRAM and FLASH-based Field Programmable Gate Array Using Proton Irradiation* [published internally]. Noordwijk, The Netherlands: European Space Research and Technology Centre, 2014-09. SEFUW 2014, SpacE FPGA Users Workshop, 2nd Edition.

A.7. POSPÍŠIL, Jan; VAŇÁT, Tomáš; SCHMIDT, Jan. *Towards Trusted Devices in FPGA by Modeling* [published internally]. Roztoky u Prahy, 2014-06. PESW 2014, 2nd Prague Embedded Systems Workshop.

A.8. VAŇÁT, Tomáš; POSPÍŠIL, Jan; SCHMIDT, Jan. *Towards Trusted Devices in FPGA by Modeling Radiation Induced Errors* [published internally]. Freiburg, Germany, 2013-12. The Working Groups meeting of the TRUDEVICE Action.

A.9. POSPÍŠIL, Jan; VAŇÁT, Tomáš. *Towards Dependable Devices in FPGA* [published internally]. Tallinn, Estonia, 2013-06. BELAS 2013, Biannual European-Latin American Summer School on Design, Test and Reliability.

A.10. POSPÍŠIL, Jan. *Reliable FPGA architectures.* 2013-05. Doctoral Study Report. Faculty of Information Technology, Czech Technical University in Prague.

A.11. POSPÍŠIL, Jan; VAŇÁT, Tomáš; SCHMIDT, Jan. *A New Method of Modeling Radiation-Induced Soft Errors in FPGA by Architecture Analysis* [published internally]. Temešvár, Czech republic, 2013-05. ESW 2013, The 1st Embedded Systems Workshop.

Remaining Publications

A.12. POSPÍŠIL, Jan; BJORKQVIST, Oskar; BOCCARDI, Andrea; WENDT, Manfred. Digital Signal Processing Techniques to Monitor Bunch-by-Bunch Beam Positions in the LHC for Machine Protection Purposes. In: *Proc. of International Beam Instrumentation Conference (IBIC'17), 20–24 August 2017.* Geneva, Switzerland: JACoW, 2018-03, pp. 146–149. International Beam Instrumentation Conference, no. 6. ISBN 978-3-95450-192-2. Available from DOI: `10.18429/JACoW-IBIC2017-TU3AB3`.

A.13. KRIVDA, Marian; ALEXANDRE, Didier; BARNBY, Lee; EVANS, David; JONES, Peter G.; JUSKO, Anton; LIETAVA, Roman; POSPÍŠIL, Jan; VILLALOBOS BAILLIE, Orlando. The ALICE Central Trigger Processor (CTP) upgrade. *Journal of Instrumentation.* 2016-03, vol. 11, no. 03. Available from DOI: `10.1088/1748-0221/11/03/C03051`. [cited: 1x].

A.14. POSPÍŠIL, J.; NOVOTNÝ, M. Evaluating Cryptanalytical Strength of Lightweight Cipher PRESENT on Reconfigurable Hardware. In: *15th Euromicro Conference on Digital System Design (DSD).* Los Alamitos: IEEE Computer Society Press, 2012-09. ISBN 978-0-7695-4798-5. Available from DOI: `10.1109/DSD.2012.53`. [cited: 3x].

A.15.   POSPÍŠIL, J.; NOVOTNÝ, M. Lightweight Cipher Resistivity against Brute-Force Attack: Analysis of PRESENT. In: *15th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS)*. New York: IEEE Computer Society Press, 2012-04. ISBN 978-1-4673-1185-4. [cited: 2x].

A.16.   POSPÍŠIL, Jan. *Cryptanalysis of PRESENT Cipher*. 2011. Master thesis. Czech Technical University in Prague, Faculty of Electrical Engineering.

A.17.   POSPÍŠIL, Jan. *Adaptér pro kopírování dat mezi USB zařízeními*. 2008. Bachelor thesis. Czech Technical University in Prague, Faculty of Electrical Engineering.

Remaining Publications (ALICE Collaboration)

A.18.   ADAMOVÁ, D. et al. J/ψ production as a function of charged-particle pseudorapidity density in p-Pb collisions at $\sqrt{s_{NN}} = 5.02$ TeV. *Phys. Lett.* 2018, vol. B776, pp. 91–104. Available from DOI: `10.1016/j.physletb.2017.11.008`.

A.19.   ADAM, Jaroslav et al. Flow dominance and factorization of transverse momentum correlations in Pb-Pb collisions at the LHC. *Phys. Rev. Lett.* 2017, vol. 118, no. 16, pp. 162302. Available from DOI: `10.1103/PhysRevLett.118.162302`.

A.20.   ADAMOVA, Dagmar et al. Azimuthally differential pion femtoscopy in Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV. *Phys. Rev. Lett.* 2017, vol. 118, no. 22, pp. 222301. Available from DOI: `10.1103/PhysRevLett.118.222301`.

A.21.   ACHARYA, Shreyasi et al. Production of muons from heavy-flavour hadron decays in p-Pb collisions at $\sqrt{s_{NN}} = 5.02$ TeV. *Phys. Lett.* 2017, vol. B770, pp. 459–472. Available from DOI: `10.1016/j.physletb.2017.03.049`.

A.22.   ACHARYA, Shreyasi et al. Production of π^0 and η mesons up to high transverse momentum in pp collisions at 2.76 TeV. *Eur. Phys. J.* 2017, vol. C77, no. 5, pp. 339. Available from DOI: `10.1140/epjc/s10052-017-5144-7,10.1140/epjc/s10052-017-4890-x`. [Eur. Phys. J.C77,no.9,586(2017)].

A.23.   ACHARYA, S. et al. First measurement of jet mass in Pb–Pb and p–Pb collisions at the LHC. *Phys. Lett.* 2018, vol. B776, pp. 249–264. Available from DOI: `10.1016/j.physletb.2017.11.044`.

A.24.   ACHARYA, Shreyasi et al. Measurement of D-meson production at mid-rapidity in pp collisions at $\sqrt{s} = 7$ TeV. *Eur. Phys. J.* 2017, vol. C77, no. 8, pp. 550. Available from DOI: `10.1140/epjc/s10052-017-5090-4`.

A.25.   ACHARYA, Shreyasi et al. Energy dependence of forward-rapidity J/ψ and ψ(2S) production in pp collisions at the LHC. *Eur. Phys. J.* 2017, vol. C77, no. 6, pp. 392. Available from DOI: `10.1140/epjc/s10052-017-4940-4`.

A.26.   ADAM, Jaroslav et al. K*(892)^0 and ϕ(1020) meson production at high transverse momentum in pp and Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV. *Phys. Rev.* 2017, vol. C95, no. 6, pp. 064606. Available from DOI: `10.1103/PhysRevC.95.064606`.

A.27.   ADAMOVA, Dagmar et al. Production of $\Sigma(1385)^{\pm}$ and $\Xi(1530)^0$ in p-Pb collisions at $\sqrt{s_{NN}} = 5.02$ TeV. *Eur. Phys. J.* 2017, vol. C77, no. 6, pp. 389. Available from DOI: `10.1140/epjc/s10052-017-4943-1`.

A.28.   ADAM, Jaroslav et al. Insight into particle production mechanisms via angular correlations of identified particles in pp collisions at $\sqrt{s} = 7$ TeV. *Eur. Phys. J.* 2017, vol. C77, no. 8, pp. 569. Available from DOI: `10.1140/epjc/s10052-017-5129-6`.

A.29.   ADAM, Jaroslav et al. Centrality dependence of the pseudorapidity density distribution for charged particles in Pb-Pb collisions at $\sqrt{s_{NN}} = 5.02$ TeV. *Phys. Lett.* 2017, vol. B772, pp. 567–577. Available from DOI: `10.1016/j.physletb.2017.07.017`.

A.30.   ADAM, Jaroslav et al. W and Z boson production in p-Pb collisions at $\sqrt{s_{NN}} = 5.02$ TeV. *JHEP.* 2017, vol. 02, pp. 077. Available from DOI: `10.1007/JHEP02(2017)077`.

A.31.   ADAM, Jaroslav et al. Determination of the event collision time with the ALICE detector at the LHC. *Eur. Phys. J. Plus.* 2017, vol. 132, no. 2, pp. 99. Available from DOI: `10.1140/epjp/i2017-11279-1`.

A.32.   ADAM, Jaroslav et al. Measurement of the production of high-p_{T} electrons from heavy-flavour hadron decays in Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV. *Phys. Lett.* 2017, vol. B771, pp. 467–481. Available from DOI: `10.1016/j.physletb.2017.05.060`.

A.33.   ADAM, Jaroslav et al. Evolution of the longitudinal and azimuthal structure of the near-side jet peak in Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV. *Phys. Rev.* 2017, vol. C96, no. 3, pp. 034904. Available from DOI: `10.1103/PhysRevC.96.034904`.

A.34.   ADAM, Jaroslav et al. Anomalous evolution of the near-side jet peak shape in Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV. *Phys. Rev. Lett.* 2017, vol. 119, no. 10, pp. 102301. Available from DOI: `10.1103/PhysRevLett.119.102301`.

A.35.   ADAM, Jaroslav et al. Measurement of electrons from beauty-hadron decays in p-Pb collisions at $\sqrt{s_{NN}} = 5.02$ TeV and Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV. *JHEP.* 2017, vol. 07, pp. 052. Available from DOI: `10.1007/JHEP07(2017)052`.

A.36.   ADAM, Jaroslav et al. Jet-like correlations with neutral pion triggers in pp and central Pb–Pb collisions at 2.76 TeV. *Phys. Lett.* 2016, vol. B763, pp. 238–250. Available from DOI: `10.1016/j.physletb.2016.10.048`.

A.37.   ADAM, Jaroslav et al. J/ψ suppression at forward rapidity in Pb-Pb collisions at $\sqrt{s_{NN}} = 5.02$ TeV. *Phys. Lett.* 2017, vol. B766, pp. 212–224. Available from DOI: `10.1016/j.physletb.2016.12.064`.

A.38.   ADAM, Jaroslav et al. Enhanced production of multi-strange hadrons in high-multiplicity proton-proton collisions. *Nature Phys.* 2017, vol. 13, pp. 535–539. Available from DOI: `10.1038/nphys4111`.

A.39.   ADAM, Jaroslav et al. Higher harmonic flow coefficients of identified hadrons in Pb-Pb collisions at $\sqrt{s_{NN}}$ = 2.76 TeV. *JHEP*. 2016, vol. 09, pp. 164. Available from DOI: `10.1007/JHEP09(2016)164`.

A.40.   ADAM, Jaroslav et al. Elliptic flow of electrons from heavy-flavour hadron decays at mid-rapidity in Pb-Pb collisions at $\sqrt{s_{\mathrm{NN}}}$ = 2.76 TeV. *JHEP*. 2016, vol. 09, pp. 028. Available from DOI: `10.1007/JHEP09(2016)028`.

A.41.   ADAM, Jaroslav et al. *D*-meson production in *p*-Pb collisions at $\sqrt{s_{NN}}$ =5.02 TeV and in pp collisions at $\sqrt{s}$ =7 TeV. *Phys. Rev.* 2016, vol. C94, no. 5, pp. 054908. Available from DOI: `10.1103/PhysRevC.94.054908`.

A.42.   ADAM, Jaroslav et al. Measurement of azimuthal correlations of D mesons and charged particles in pp collisions at $\sqrt{s}$ = 7 TeV and p-Pb collisions at $\sqrt{s_{\mathrm{NN}}}$ = 5.02 TeV. *Eur. Phys. J.* 2017, vol. C77, no. 4, pp. 245. Available from DOI: `10.1140/epjc/s10052-017-4779-8`.

A.43.   ADAM, Jaroslav et al. Pseudorapidity dependence of the anisotropic flow of charged particles in Pb-Pb collisions at $\sqrt{s_{NN}}$ = 2.76 TeV. *Phys. Lett.* 2016, vol. B762, pp. 376–388. Available from DOI: `10.1016/j.physletb.2016.07.017`.

A.44.   ADAM, Jaroslav et al. Correlated event-by-event fluctuations of flow harmonics in Pb-Pb collisions at $\sqrt{s_{NN}}$ = 2.76 TeV. *Phys. Rev. Lett.* 2016, vol. 117, pp. 182301. Available from DOI: `10.1103/PhysRevLett.117.182301`.

A.45.   ADAM, Jaroslav et al. Measurement of transverse energy at midrapidity in Pb-Pb collisions at $\sqrt{s_{NN}}$ = 2.76 TeV. *Phys. Rev.* 2016, vol. C94, no. 3, pp. 034903. Available from DOI: `10.1103/PhysRevC.94.034903`.

A.46.   ADAM, Jaroslav et al. Centrality dependence of charged jet production in p–Pb collisions at $\sqrt{s_{NN}}$ = 5.02 TeV. *Eur. Phys. J.* 2016, vol. C76, no. 5, pp. 271. Available from DOI: `10.1140/epjc/s10052-016-4107-8`.

A.47.   ADAM, Jaroslav et al. Centrality dependence of $\psi(2S)$ suppression in p-Pb collisions at $\sqrt{s_{NN}}$ = 5.02 TeV. *JHEP*. 2016, vol. 06, pp. 050. Available from DOI: `10.1007/JHEP06(2016)050`.

A.48.   ADAM, J. et al. Measurement of D-meson production versus multiplicity in p-Pb collisions at $\sqrt{s_{\mathrm{NN}}}$ = 5.02 TeV. *JHEP*. 2016, vol. 08, pp. 078. Available from DOI: `10.1007/JHEP08(2016)078`.

A.49.   ADAM, Jaroslav et al. Particle identification in ALICE: a Bayesian approach. *Eur. Phys. J. Plus.* 2016, vol. 131, no. 5, pp. 168. Available from DOI: `10.1140/epjp/i2016-16168-5`.

A.50.   ADAM, Jaroslav et al. Anisotropic flow of charged particles in Pb-Pb collisions at $\sqrt{s_{\mathrm{NN}}}$ = 5.02 TeV. *Phys. Rev. Lett.* 2016, vol. 116, no. 13, pp. 132302. Available from DOI: `10.1103/PhysRevLett.116.132302`.

A.51.   ADAM, Jaroslav et al. Production of K* (892)^0 and ϕ (1020) in p–Pb collisions at $\sqrt{s_{NN}}$ = 5.02 TeV. *Eur. Phys. J.* 2016, vol. C76, no. 5, pp. 245. Available from DOI: `10.1140/epjc/s10052-016-4088-7`.

A.52.   ADAM, Jaroslav et al. Multiplicity dependence of charged pion, kaon, and (anti)-proton production at large transverse momentum in p-Pb collisions at $\sqrt{s_{NN}}$ = 5.02 TeV. *Phys. Lett.* 2016, vol. B760, pp. 720–735. Available from DOI: `10.1016/j.physletb.2016.07.050`.

A.53.   ADAM, Jaroslav et al. Multipion Bose-Einstein correlations in pp,p -Pb, and Pb-Pb collisions at energies available at the CERN Large Hadron Collider. *Phys. Rev.* 2016, vol. C93, no. 5, pp. 054908. Available from DOI: `10.1103/PhysRevC.93.054908`.

A.54.   ADAM, Jaroslav et al. Multi-strange baryon production in p-Pb collisions at $\sqrt{s_{NN}}$ = 5.02 TeV. *Phys. Lett.* 2016, vol. B758, pp. 389–401. Available from DOI: `10.1016/j.physletb.2016.05.027`.

A.55.   ADAM, Jaroslav et al. Centrality dependence of the charged-particle multiplicity density at midrapidity in Pb-Pb collisions at $\sqrt{s_{NN}}$ = 5.02 TeV. *Phys. Rev. Lett.* 2016, vol. 116, no. 22, pp. 222302. Available from DOI: `10.1103/PhysRevLett.116.222302`.

A.56.   ADAM, Jaroslav et al. Charge-dependent flow and the search for the chiral magnetic wave in Pb-Pb collisions at $\sqrt{s_{NN}}$ = 2.76 TeV. *Phys. Rev.* 2016, vol. C93, no. 4, pp. 044903. Available from DOI: `10.1103/PhysRevC.93.044903`.

A.57.   ADAM, Jaroslav et al. Measurement of an excess in the yield of J/ψ at very low p_{T} in Pb-Pb collisions at $\sqrt{s_{NN}}$ = 2.76 TeV. *Phys. Rev. Lett.* 2016, vol. 116, no. 22, pp. 222301. Available from DOI: `10.1103/PhysRevLett.116.222301`.

A.58.   ADAM, Jaroslav et al. Pseudorapidity and transverse-momentum distributions of charged particles in proton–proton collisions at $\sqrt{s}$ = 13 TeV. *Phys. Lett.* 2016, vol. B753, pp. 319–329. Available from DOI: `10.1016/j.physletb.2015.12.030`.

A.59.   ADAM, Jaroslav et al. Inclusive quarkonium production at forward rapidity in pp collisions at $\sqrt{s}$ = 8 TeV. *Eur. Phys. J.* 2016, vol. C76, no. 4, pp. 184. Available from DOI: `10.1140/epjc/s10052-016-3987-y`.

A.60.   ADAM, Jaroslav et al. Charged-particle multiplicities in proton–proton collisions at $\sqrt{s}$ = 0.9 to 8 TeV. *Eur. Phys. J.* 2017, vol. C77, no. 1, pp. 33. Available from DOI: `10.1140/epjc/s10052-016-4571-1`.

A.61.   ADAM, Jaroslav et al. Measurement of electrons from heavy-flavour hadron decays in p-Pb collisions at $\sqrt{s_{NN}}$ = 5.02 TeV. *Phys. Lett.* 2016, vol. B754, pp. 81–93. Available from DOI: `10.1016/j.physletb.2015.12.067`.

A.62.   ADAM, Jaroslav et al. Azimuthal anisotropy of charged jet production in $\sqrt{s_{NN}}$ = 2.76 TeV Pb-Pb collisions. *Phys. Lett.* 2016, vol. B753, pp. 511–525. Available from DOI: `10.1016/j.physletb.2015.12.047`.

A.63. ADAM, Jaroslav et al. Direct photon production in Pb-Pb collisions at $\sqrt{s_{\mathrm{NN}}} =$ 2.76 TeV. *Phys. Lett.* 2016, vol. B754, pp. 235–248. Available from DOI: `10.1016/j.physletb.2016.01.020`.

A.64. ADAM, Jaroslav et al. Centrality evolution of the charged-particle pseudorapidity density over a broad pseudorapidity range in Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV. *Phys. Lett.* 2016, vol. B754, pp. 373–385. Available from DOI: `10.1016/j.physletb.2015.12.082`.

A.65. ADAM, Jaroslav et al. Measurement of D_s^+ production and nuclear modification factor in Pb-Pb collisions at $\sqrt{s_{\mathrm{NN}}} = 2.76$ TeV. *JHEP.* 2016, vol. 03, pp. 082. Available from DOI: `10.1007/JHEP03(2016)082`.

A.66. ADAM, Jaroslav et al. Multiplicity and transverse momentum evolution of charge-dependent correlations in pp, p–Pb, and Pb–Pb collisions at the LHC. *Eur. Phys. J.* 2016, vol. C76, no. 2, pp. 86. Available from DOI: `10.1140/epjc/s10052-016-3915-1`.

A.67. ADAM, Jaroslav et al. Transverse momentum dependence of D-meson production in Pb-Pb collisions at $\sqrt{s_{\mathrm{NN}}} = 2.76$ TeV. *JHEP.* 2016, vol. 03, pp. 081. Available from DOI: `10.1007/JHEP03(2016)081`.

A.68. ADAM, Jaroslav et al. Coherent $\psi(2\mathrm{S})$ photo-production in ultra-peripheral Pb Pb collisions at $\sqrt{s}_{\mathrm{NN}} = 2.76$ TeV. *Phys. Lett.* 2015, vol. B751, pp. 358–370. Available from DOI: `10.1016/j.physletb.2015.10.040`.

A.69. ADAM, Jaroslav et al. Precision measurement of the mass difference between light nuclei and anti-nuclei. *Nature Phys.* 2015, vol. 11, no. 10, pp. 811–814. Available from DOI: `10.1038/nphys3432`.

A.70. ADAM, Jaroslav et al. Study of cosmic ray events with high muon multiplicity using the ALICE detector at the CERN Large Hadron Collider. *JCAP.* 2016, vol. 1601, no. 01, pp. 032. Available from DOI: `10.1088/1475-7516/2016/01/032`.

A.71. ADAM, Jaroslav et al. Centrality dependence of pion freeze-out radii in Pb-Pb collisions at $\sqrt{s}_{NN} = 2.76$ TeV. *Phys. Rev.* 2016, vol. C93, no. 2, pp. 024905. Available from DOI: `10.1103/PhysRevC.93.024905`.

A.72. ADAM, Jaroslav et al. Event shape engineering for inclusive spectra and elliptic flow in Pb-Pb collisions at $\sqrt{s_{\mathrm{NN}}} = 2.76$ TeV. *Phys. Rev.* 2016, vol. C93, no. 3, pp. 034916. Available from DOI: `10.1103/PhysRevC.93.034916`.

A.73. ADAM, Jaroslav et al. Elliptic flow of muons from heavy-flavour hadron decays at forward rapidity in Pb–Pb collisions at $\sqrt{s_{\mathrm{NN}}} = 2.76$ TeV. *Phys. Lett.* 2016, vol. B753, pp. 41–56. Available from DOI: `10.1016/j.physletb.2015.11.059`.

A.74. ADAM, Jaroslav et al. Production of light nuclei and anti-nuclei in pp and Pb-Pb collisions at energies available at the CERN Large Hadron Collider. *Phys. Rev.* 2016, vol. C93, no. 2, pp. 024917. Available from DOI: `10.1103/PhysRevC.93.024917`.

A.75. ADAM, Jaroslav et al. ϕ-meson production at forward rapidity in p-Pb collisions at $\sqrt{s_{NN}} = 5.02$ TeV and in pp collisions at $\sqrt{s} = 2.76$ TeV. *Phys. Lett.* 2017, vol. B768, pp. 203–217. Available from DOI: `10.1016/j.physletb.2017.01.074`.

A.76. ADAM, Jaroslav et al. Centrality dependence of inclusive J/ψ production in p-Pb collisions at $\sqrt{s_{NN}} = 5.02$ TeV. *JHEP.* 2015, vol. 11, pp. 127. Available from DOI: `10.1007/JHEP11(2015)127`.

A.77. ADAM, Jaroslav et al. Differential studies of inclusive J/ψ and $\psi(2S)$ production at forward rapidity in Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV. *JHEP.* 2016, vol. 05, pp. 179. Available from DOI: `10.1007/JHEP05(2016)179`.

A.78. ADAM, Jaroslav et al. $^3_\Lambda\mathrm{H}$ and $^3_\Lambda\overline{\mathrm{H}}$ production in Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV. *Phys. Lett.* 2016, vol. B754, pp. 360–372. Available from DOI: `10.1016/j.physletb.2016.01.040`.

A.79. ADAM, Jaroslav et al. Forward-central two-particle correlations in p-Pb collisions at $\sqrt{s_{NN}} = 5.02$ TeV. *Phys. Lett.* 2016, vol. B753, pp. 126–139. Available from DOI: `10.1016/j.physletb.2015.12.010`.

A.80. ADAM, Jaroslav et al. One-dimensional pion, kaon, and proton femtoscopy in Pb-Pb collisions at $\sqrt{s_{NN}} =2.76$ TeV. *Phys. Rev.* 2015, vol. C92, no. 5, pp. 054908. Available from DOI: `10.1103/PhysRevC.92.054908`.

A.81. ADAM, Jaroslav et al. Search for weakly decaying $\overline{\Lambda \mathrm{n}}$ and $\Lambda\Lambda$ exotic bound states in central Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV. *Phys. Lett.* 2016, vol. B752, pp. 267–277. Available from DOI: `10.1016/j.physletb.2015.11.048`.

A.82. ADAM, Jaroslav et al. Centrality dependence of the nuclear modification factor of charged pions, kaons, and protons in Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV. *Phys. Rev.* 2016, vol. C93, no. 3, pp. 034913. Available from DOI: `10.1103/PhysRevC.93.034913`.

A.83. ADAM, Jaroslav et al. Centrality dependence of high-p_T D meson suppression in Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV. *JHEP.* 2015, vol. 11, pp. 205. Available from DOI: `10.1007/JHEP11(2015)205,10.1007/JHEP06(2017)032`. [Addendum: JHEP06,032(2017)].

A.84. ADAM, Jaroslav et al. Measurement of jet quenching with semi-inclusive hadron-jet distributions in central Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV. *JHEP.* 2015, vol. 09, pp. 170. Available from DOI: `10.1007/JHEP09(2015)170`.

A.85. ADAM, Jaroslav et al. Measurement of charm and beauty production at central rapidity versus charged-particle multiplicity in proton-proton collisions at $\sqrt{s} = 7$ TeV. *JHEP.* 2015, vol. 09, pp. 148. Available from DOI: `10.1007/JHEP09(2015)148`.

A.86. ADAM, Jaroslav et al. Inclusive, prompt and non-prompt J/ψ production at mid-rapidity in Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV. *JHEP.* 2015, vol. 07, pp. 051. Available from DOI: `10.1007/JHEP07(2015)051`.

A.87.   ADAM, Jaroslav et al. Measurement of pion, kaon and proton production in proton–proton collisions at $\sqrt{s} = 7$ TeV. *Eur. Phys. J.* 2015, vol. C75, no. 5, pp. 226. Available from DOI: `10.1140/epjc/s10052-015-3422-9`.

A.88.   ADAM, Jaroslav et al. Coherent ρ^0 photoproduction in ultra-peripheral Pb-Pb collisions at $\sqrt{s_{\mathrm{NN}}} = 2.76$ TeV. *JHEP.* 2015, vol. 09, pp. 095. Available from DOI: `10.1007/JHEP09(2015)095`.

A.89.   ADAM, Jaroslav et al. Rapidity and transverse-momentum dependence of the inclusive J/ψ nuclear modification factor in p-Pb collisions at $\sqrt{s_{N\,N}} = 5.02$ TeV. *JHEP.* 2015, vol. 06, pp. 055. Available from DOI: `10.1007/JHEP06(2015)055`.

A.90.   ADAM, Jaroslav et al. Measurement of dijet k_T in p–Pb collisions at $\sqrt{s_{NN}}$=5.02 TeV. *Phys. Lett.* 2015, vol. B746, pp. 385–395. Available from DOI: `10.1016/j.physletb.2015.05.033`.

A.91.   ADAM, Jaroslav et al. Measurement of charged jet production cross sections and nuclear modification in p-Pb collisions at $\sqrt{s_{NN}} = 5.02$ TeV. *Phys. Lett.* 2015, vol. B749, pp. 68–81. Available from DOI: `10.1016/j.physletb.2015.07.054`.

A.92.   ADAM, Jaroslav et al. Measurement of jet suppression in central Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV. *Phys. Lett.* 2015, vol. B746, pp. 1–14. Available from DOI: `10.1016/j.physletb.2015.04.039`.

A.93.   ADAM, J. et al. Two-pion femtoscopy in p-Pb collisions at $\sqrt{s_{NN}} = 5.02$ TeV. *Phys. Rev.* 2015, vol. C91, pp. 034906. Available from DOI: `10.1103/PhysRevC.91.034906`.

A.94.   ADAM, Jaroslav et al. Forward-backward multiplicity correlations in pp collisions at $\sqrt{s} = 0.9$, 2.76 and 7 TeV. *JHEP.* 2015, vol. 05, pp. 097. Available from DOI: `10.1007/JHEP05(2015)097`.

A.95.   ADAM, Jaroslav et al. Centrality dependence of particle production in p-Pb collisions at $\sqrt{s_{NN}}$= 5.02 TeV. *Phys. Rev.* 2015, vol. C91, no. 6, pp. 064905. Available from DOI: `10.1103/PhysRevC.91.064905`.

Author's Citations

B.1.   SIELEWICZ, K.M.; RINELLA, G. Aglieri; BONORA, M.; FERENCEI, J.; GIUBI-
        LATO, P.; ROSSEWIJ, M. J.; SCHAMBACH, J.; VANAT, T. Prototype readout
        electronics for the upgraded ALICE Inner Tracking System. *Journal of Instru-
        mentation.* 2017, vol. 12, no. 01, pp. C01008. http://stacks.iop.org/1748-
        0221/12/i=01/a=C01008.

B.2.   VAŇÁT, Tomáš; KŘÍŽEK, Filip; FERENCEI, Jozef; KUBÁTOVÁ, Hana. Compar-
        ing proton and neutron induced SEU cross section in FPGA. In: *2016 IEEE 19th
        International Symposium on Design and Diagnostics of Electronic Circuits Systems
        (DDECS).* 2016-04, pp. 1–4. Available from DOI: 10.1109/DDECS.2016.7482480.

B.3.   DUAN, X.; CUI, Q.; WANG, S.; FANG, H.; SHE, G. Differential power analysis
        attack and efficient countermeasures on PRESENT. In: *2016 8th IEEE Interna-
        tional Conference on Communication Software and Networks (ICCSN).* 2016-06,
        pp. 8–12. Available from DOI: 10.1109/ICCSN.2016.7586627.

B.4.   RAHMAN, Arifur; ALI, Maaruf. Analysis and Evaluation of Wireless Networks
        by Implementation of Test Security Keys. In: MIRAZ, Mahdi H.; EXCELL, Peter;
        WARE, Andrew; SOOMRO, Safeeullah; ALI, Maaruf (eds.). *Emerging Technologies
        in Computing.* Cham: Springer International Publishing, 2018, pp. 107–126. ISBN
        978-3-319-95450-9.

B.5.   JACINTO, Edwar; MARTÍNEZ, Fredy; MARTÍNEZ, Fernando. Learning Strate-
        gies for Cryptography Using Embedded Systems. In: *Smart Education and e-Lear-
        ning 2016.* Cham: Springer International Publishing, 2016, pp. 495–505. ISBN 978-
        3-319-39690-3.

B.6.   ATAT, R.; LIU, L.; CHEN, H.; WU, J.; LI, H.; YI, Y. Enabling cyber-physical
        communication in 5G cellular networks: challenges, spatial spectrum sensing, and
        cyber-security. *IET Cyber-Physical Systems: Theory Applications.* 2017, vol. 2, no.
        1, pp. 49–54. ISSN 2398-3396. Available from DOI: 10.1049/iet-cps.2017.0010.

B.7. LY, K.; SUN, W.; JIN, Y. Emerging challenges in cyber-physical systems: A balance of performance, correctness, and security. In: *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2016-04, pp. 498–502. Available from DOI: `10.1109/INFCOMW.2016.7562128`.

B.8. SIELEWICZ, K. M.; RINELLA, G. A.; BONORA, M.; GIUBILATO, P.; LUPI, M.; ROSSEWIJ, M. J.; SCHAMBACH, J.; VANAT, T. Experimental Methods and Results for the Evaluation of Triple Modular Redundancy SEU Mitigation Techniques with the Xilinx Kintex-7 FPGA. In: *2017 IEEE Radiation Effects Data Workshop (REDW)*. 2017-07, pp. 1–7. Available from DOI: `10.1109/NSREC.2017.8115451`.

B.9. DU, B.; STERPONE, L. Online monitoring soft errors in reconfigurable FPGA during radiation test. In: *2017 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. 2017-05, pp. 1–5. Available from DOI: `10.1109/I2MTC.2017.7969976`.

B.10. MATLOCHA, Tomas. Ion Source for a Single Particle Accelerator. In: *Šimáně 2016*. Acta Polytechnica CTU, 2016. Available from DOI: `10.14311/AP.2016.4.0050`.

B.11. COSTA, Filippo; KLUGE, Alex; VANDE VYVRE, Pierre. The detector read-out in ALICE during Run 3 and 4. *Journal of Physics: Conference Series*. 2017, vol. 898, no. 3, pp. 032011. Available from DOI: `10.1088/1742-6596/898/3/032011`.

Citation Links

Known publications citing my publications (auto-citations are not listed).

Publication [A.15] is cited in: [B.3] and [B.4].

Publication [A.14] is cited in: [B.5], [B.6] and [B.7].

Publication [A.2] is cited in: [B.2], [B.8], [B.1], [B.9] and [B.10].

Publication [A.13] is cited in: [B.11].

VPR Results

Some test results from the VTR based implementation of the proposed method are shown on next pages. The simplified test design described in section 3.2 was used to obtain results shown here. Only 20 stages of the original circuit were "synthesized" directly into the VPR input format (BLIF) by a custom script. The excerpt of this BLIF file can be seen in Listing A.1. More information can be found in section 4.2.1.

```
1   .model pipeline
2     .inputs clk
3     .clock clk
4     .outputs d3 d2 d1 d0
5
6     .subckt stage \
7       clk=clk \
8       din0=d0 din1=d1 din2=d2 din3=d3 \
9       dout0=wire_0_0 dout1=wire_0_1 dout2=wire_0_2 dout3=wire_0_3
10    ...
11    .subckt stage \
12      clk=clk \
13      din0=wire_18_0 din1=wire_18_1 din2=wire_18_2 din3=wire_18_3 \
14      dout0=d0 dout1=d1 dout2=d2 dout3=d3
15  .end
16
17  .model stage
18    .inputs clk din3 din2 din1 din0
19    .clock clk
20    .outputs dout3 dout2 dout1 dout0
21
22    .names din3 din2 din1 din0 c3
23      0-10 1
24      1101 1
25      -110 1
26      000- 1
27      1011 1
28      -000 1
29    .names din3 din2 din1 din0 c2
30      11-0 1
31      -101 1
32      0-1- 1
33    .names din3 din2 din1 din0 c1
34      0010 1
35      11-1 1
36      1-00 1
37      0-01 1
38      -10- 1
39    .names din3 din2 din1 din0 c0
40      0-00 1
41      01-1 1
42      -0-0 1
43      1-10 1
44    .latch c3 dout3 re clk 3
45    .latch c2 dout2 re clk 3
46    .latch c1 dout1 re clk 3
47    .latch c0 dout0 re clk 3
48  .end
```

Listing A.1: Excerpt of the BLIF test design used for the VPR testing

Initial Placement. Cost: 17.9458 BB Cost: 17.9458 TD Cost 0 Delay Cost: 0   Channel Factor: 100
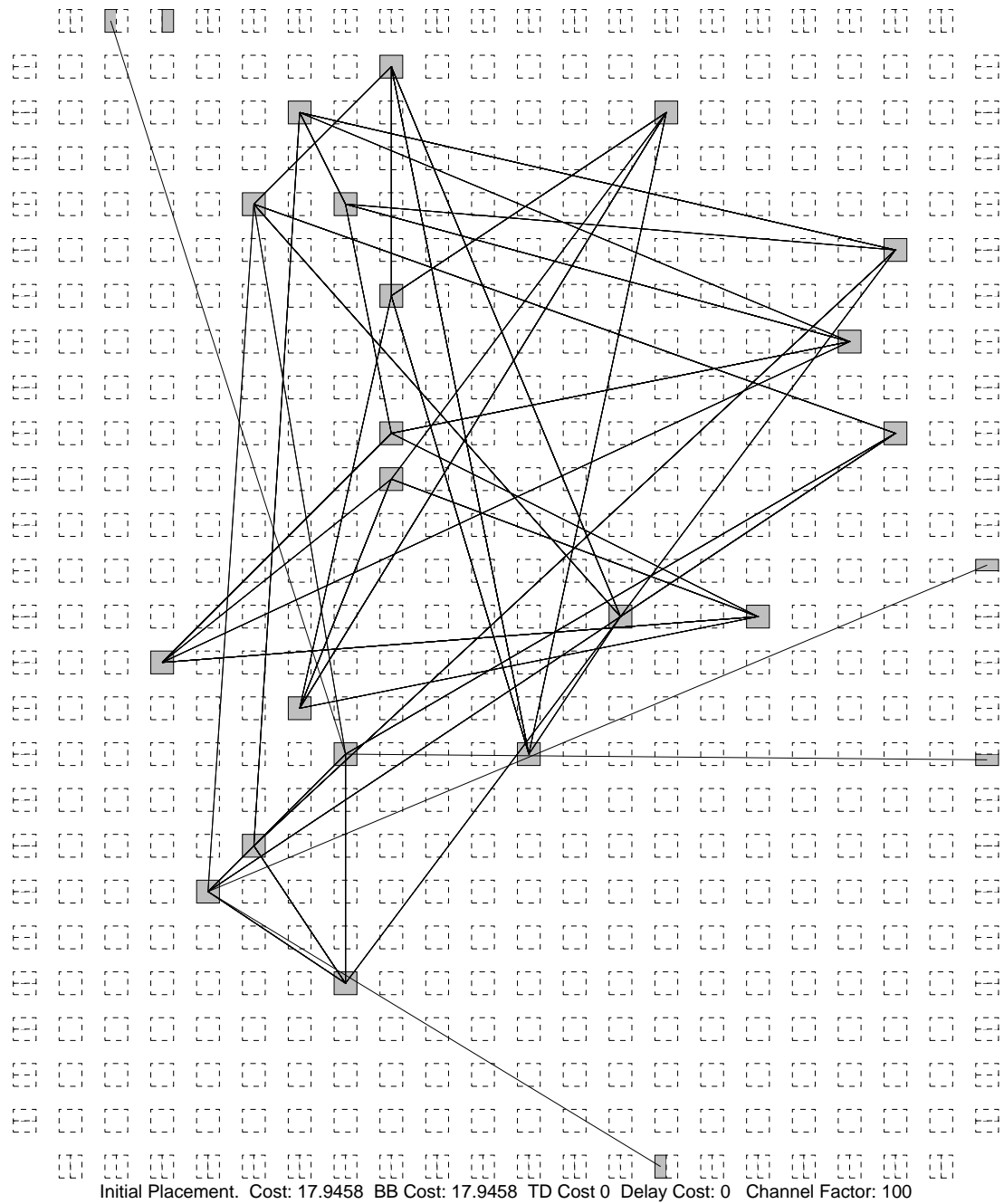
Figure A.1: VPR result: an initial placement

Placement. Cost: 3.74876 bb_cost: 3.74876 td_cost: 0 Channel Factor: 100
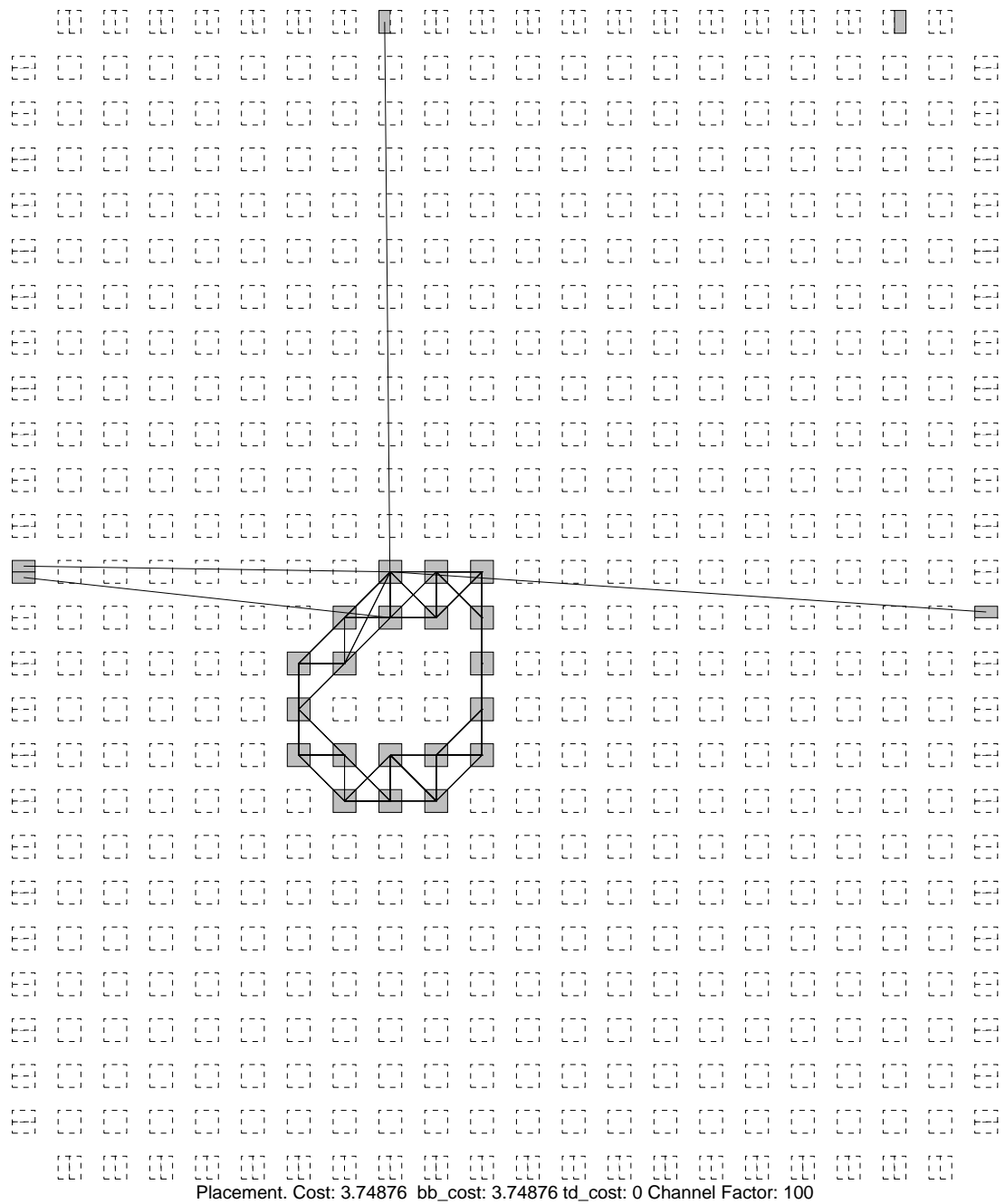
Figure A.2: VPR result: a final placement

Block 0 (n90) at (8, 12) selected.
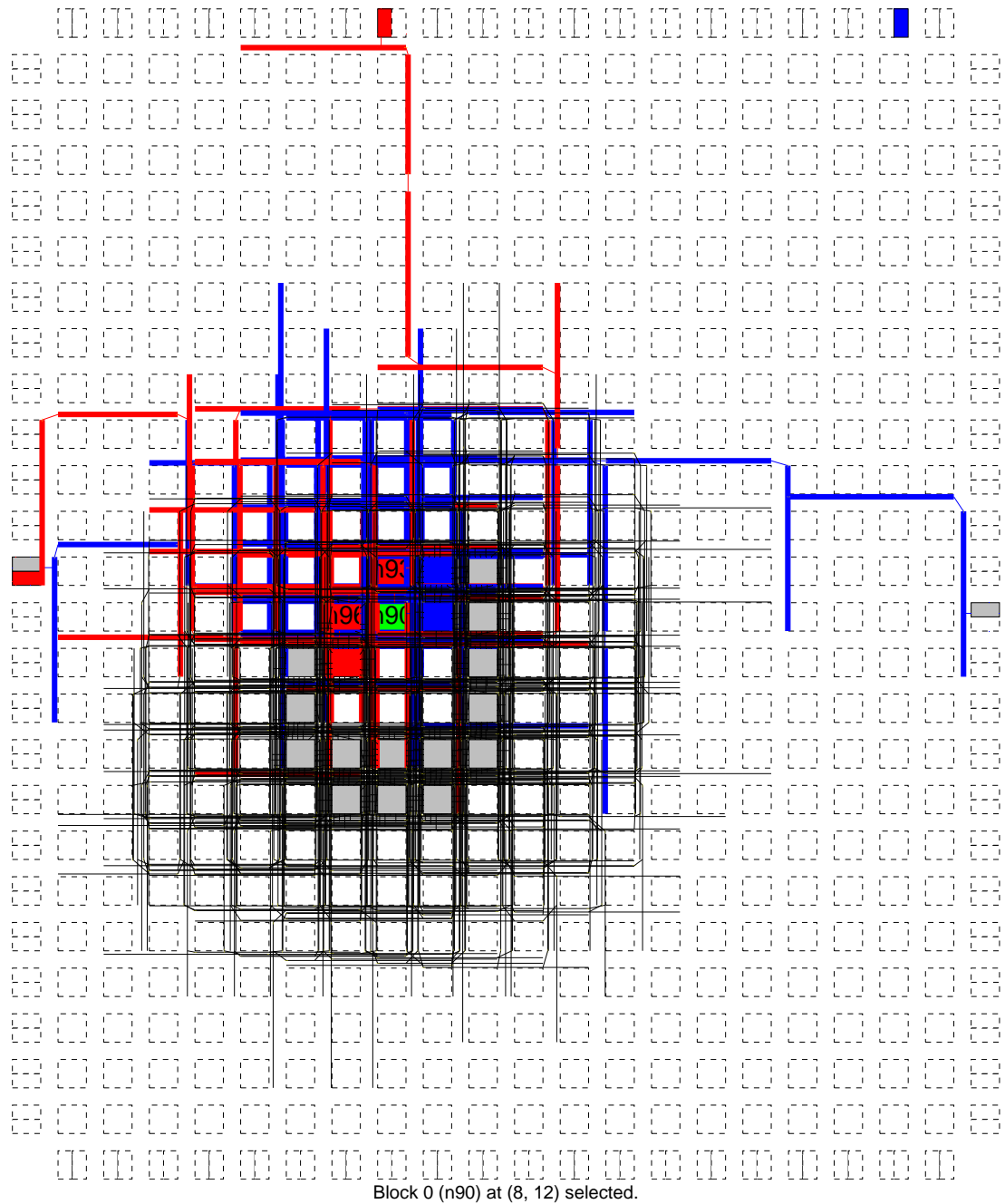
Figure A.3: VPR result: a fully routed design

Implementation Scripts

Windows command line scripts used in the model implementation mentioned in section 5.3 are shown here, namely:

- ○ Read-me text file describing basic features and commands of the implementation, see Listing B.1.

- ○ Script `0_prompt.cmd` for starting the flow and setting all environment variables, see Listing B.2.

- ○ Script `1_run_ISE.cmd` for running Xilinx ISE design flow, see Listing B.3.

- ○ Script `2_display.cmd` displaying placed and routed design in the Xilinx FPGA Editor floor-planning tool, see Listing B.4.

- ○ Script `3_netgens.cmd` generating simulation models of the placed and routed design, see Listing B.5.

- ○ Script `4_Xdl2Bench.cmd` running RapidSmith framework with our extension and transforming the placed and routed design into a BENCH format, including implemented fault models, see Listing B.6.

- ○ Script `5_hope.cmd` running the HOPE fault simulation, see Listing B.7.

- ○ Script `6_probability.cmd` calculating the failure probability, see Listing B.8.

- ○ Script `setProject.cmd` setting the actual project to be processed, see Listing B.9.

- ○ Script `runBasicFlow.cmd`, a shortcut for running scripts 1, 4 and 5, see Listing B.10.

```
1   XDL flow
2   Jan Pospisil (CVUT, FIT, DDD), 2018, jan.pospisil@fit.cvut.cz
3
4   The flow to parse HDL design into a placed and routed Xilinx XDL
5   representation and later into a BENCH netlist with a possibility to add
6   SEU-aware saboteurs. The BENCH netlist can be analyzed with fault
7   simulator HOPE. All this for Spartan-3 FPGA (partname XC3S200FT256).
8
9   1) prepare the following folder structure:
10      .\<projectName>\src\              - folder with HDL design sources
11                                          (name top entity "project")
12      .\<projectName>\src\project.ucf  - Xilinx UCF (User Constrain File)
13      .\<projectName>\src\project.prj  - HDL files listing (ISE format,
14                                          relative to ise_files folder)
15      .\<projectName>\ise_files\        - empty folder for ISE by-products
16      .\<projectName>\project.tst        - optional test pattern file in
17                                          HOPE format
18  2) set environment variable XDL_FLOW_PROJECT to <projectName>
19      SET XDL_FLOW_PROJECT=flow_test
20  3) control the flow via these commands:
21      1_run_ISE     - run ISE Synthesize and Implementation, generates
22                      placed and routed NCD and exports it as a XDL
23      2_display     - displays placed and routed NCD in the Xilinx FPGA
24                      Editor
25      3_netgens     - generates different simulation models of the design
26      4_Xdl2Bench   - translates XDL into a BENCH netlist with SEU
27                      saboteurs
28      5_hope        - runs HOPE fault simulation either with supplied
29                      test patterns or with random pattern generation
30      6_probability - calculates the failure probability of actual
31                      project
32  More commands you can use:
33      atalanta      - ATPG Atalanta
34      hope          - fault simulator HOPE
35      Xdl2Bench     - translator from XDL into BENCH format
36      setProject    - helper script for displaying and setting
37                      XDL_FLOW_PROJECT variable
38      runBasicFlow  - takes one argument (project name), call setProject
39                      and run scripts:
40                          1_run_ISE
41                          4_Xdl2Bench
42                          5_hope
43      and all Xilinx ISE commands
44  Advanced ISE project management:
45      If you wish to use your own settings for XST command, place your
46      own project.xst file in the folder
47      .\<projectName>\ise_files\project.xst
48      If this file is not found, default version is used.
```

Listing B.1: Implementation read-me file

```
1  @echo off
2
3  set ATALANTA_MAN=e:\apps\Atalanta\2.0_1997
4  set HOPE_MAN=e:\apps\Hope
5
6  set path=%ATALANTA_MAN%;%HOPE_MAN%;%PATH%
7
8  more +1 README.txt
9
10 cmd /k C:\Programy\Xilinx\14.7\ISE_DS\settings32.bat
```

Listing B.2: Implementation script `0_prompt.cmd`

```
1  @echo off
2
3  set ISE_PRJ_PATH=%XDL_FLOW_PROJECT%\ise_files
4
5  mkdir %ISE_PRJ_PATH% 2> nul
6
7  IF NOT EXIST "%ISE_PRJ_PATH%\project.xst" (
8      echo Copying default project.xst...
9      copy project.xst %ISE_PRJ_PATH%\.
10 )
11
12 pushd %ISE_PRJ_PATH%
13
14 mkdir xst 2> nul
15 mkdir xst\tmp 2> nul
16
17 echo.
18 echo Running XST synthesis
19 xst -ifn project.xst -intstyle xflow
20 if errorlevel 1 goto error
21
22 echo.
23 echo Running NGDBuild
24 ngdbuild -dd _ngo -uc ..\src\project.ucf -intstyle xflow project.ngc
25 if errorlevel 1 goto error
26
27 echo.
28 echo Running MAP
29 map -intstyle xflow project.ngd -o project_map.ncd
30 if errorlevel 1 goto error
31
32 echo.
33 echo Running Place and Route
34 par -w -intstyle xflow project_map.ncd project_par.ncd project_map.pcf
35 if errorlevel 1 goto error
36
37
```

```
38  echo.
39  echo Exporting XDL
40  xdl -ncd2xdl project_par.ncd ..\project.xdl
41  if errorlevel 1 goto error
42
43  echo.
44  echo Done :-)
45  goto end
46
47  :error
48  echo Error occured!
49
50  :end
51  popd
52
53  REM beep
54  echo \bel
```

Listing B.3: Implementation script `1_run_ISE.cmd`

```
1  @start fpga_editor.exe %XDL_FLOW_PROJECT%\ise_files\project_par.ncd ↩
       %XDL_FLOW_PROJECT%\ise_files\project_map.pcf
```

Listing B.4: Implementation script `2_display.cmd`

```
1  @echo off
2
3  set ISE_PRJ_PATH=%XDL_FLOW_PROJECT%\ise_files
4
5  pushd %ISE_PRJ_PATH%
6
7  mkdir netgen 2> nul
8  mkdir netgen\ecn_f 2> nul
9  mkdir netgen\ecn_c 2> nul
10  mkdir netgen\sim 2> nul
11  mkdir netgen\sta 2> nul
12
13  netgen -ecn formality -dir netgen\ecn_f -w -ngm project_map.ngm ↩
       project_par.ncd
14  netgen -ecn conformal -dir netgen\ecn_c -w -ngm project_map.ngm ↩
       project_par.ncd
15  netgen -sim -ofmt vhdl -dir netgen\sim -w project_par.ncd
16  netgen -sta -dir netgen\sta -w -pcf project_map.pcf project_par.ncd
17
18  popd
```

Listing B.5: Implementation script `3_netgens.cmd`

```
1  @echo off
2
3  set options=
4
5  REM add saboteurs
6  set options=%options% -b
7
8  REM skip simplification
9  REM set options=%options% -s
10
11 REM collapse clock nets
12 set options=%options% -c
13
14 REM treat resets as synchronous
15 set options=%options% -a
16
17 REM skip comb. loops check
18 REM set options=%options% -l
19
20 call Xdl2Bench %options% -xf %XDL_FLOW_PROJECT%\project.xdl 2>&1 | tee ←
       %XDL_FLOW_PROJECT%\Xdl2Bench.log
21
22 REM beep
23 echo \bel
```

Listing B.6: Implementation script `4_Xdl2Bench.cmd`

```
1  @echo off
2
3  pushd %XDL_FLOW_PROJECT%
4
5  del project.und 2> nul
6
7  IF EXIST project.tst (
8      echo Running HOPE simulation with supplied test patterns
9      hope -l project.hope.log -f project.flt -t project.tst -U project.←
           und -0 project.bench
10 ) ELSE (
11     echo Running HOPE simulation in random patterns mode
12     hope -l project.hope.log -f project.flt -U project.und -0 project.←
           bench
13 )
14 cat hope.warning
15
16 popd
17
18 REM beep
19 echo \bel
```

Listing B.7: Implementation script `5_hope.cmd`

```
1  @python probability.py %* %XDL_FLOW_PROJECT%\project.flt ↵
       %XDL_FLOW_PROJECT%\project.und probabilities.csv 2>&1 | tee ↵
       %XDL_FLOW_PROJECT%\probability.log
```

Listing B.8: Implementation script `6_probability.cmd`

```
1   @echo off
2
3   IF "%1"=="" (
4       echo Usage: %0 [^<projectName^>]
5       echo Actual project: %XDL_FLOW_PROJECT%
6       goto end
7   )
8
9   SET XDL_FLOW_PROJECT=%1
10
11  :end
```

Listing B.9: Implementation script `setProject.cmd`

```
1   @echo off
2
3   IF "%1"=="" (
4       echo Usage: %0 ^<projectName^>
5       goto end
6   )
7
8   call setProject %1
9   call 1_run_ISE
10  call 4_Xdl2Bench
11  call 5_hope
12
13  :end
```

Listing B.10: Implementation script `runBasicFlow.cmd`