FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Operation and Verification Framework for the FRad Experimental ASIC

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Technische Informatik

eingereicht von

Bernhard Fritz, BSc
Matrikelnummer 0828317

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr. Andreas Steininger
Mitwirkung: Dr. Varadan Savulimedu Veeravalli, MSc

Wien, 6. März 2018

Bernhard Fritz	Andreas Steininger

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Operation and Verification Framework for the FRad Experimental ASIC

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Computer Engineering

by

Bernhard Fritz, BSc
Registration Number 0828317

to the Faculty of Informatics

at the Vienna University of Technology

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr. Andreas Steininger
Assistance: Dr. Varadan Savulimedu Veeravalli, MSc

Vienna, 6th March, 2018

Bernhard Fritz

Andreas Steininger

Erklärung zur Verfassung der Arbeit

Bernhard Fritz, BSc
Landstrasser Hauptstrasse 104/4, 1030 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 6. März 2018

Bernhard Fritz

Kurzfassung

Das Institut für Technische Informatik der Technischen Universität Wien hat zur Untersuchung von strahlungsinduzierten Fehlern in hochintegrierten Schaltungen einen experimentellen Chip, den FRad-Chip, entwickelt. Im Laufe des Entwicklungsprozesses wurde das Design auf unterschiedliche Arten simuliert, um die Funktionalität sicherzustellen. Selbst die detailtreuesten Simulationen sind jedoch kein vollständiger Ersatz für Tests mit dem tatsächlichen Chip, da auch später im Entwicklungszyklus noch Fehler auftreten können. Die spezielle Schnittstelle des Chips und die hohen Ansprüche an die Datenintegrität erfordern einen auf die Aufgabe zugeschnittenen Controller.
Das Ziel der Arbeit ist es ein zuverlässiges Kontrollsystem zu entwickeln, um die Funktionalität des Chips nützen zu können und letztendlich durch eine Reihe unterschiedlicher Tests die Funktionalität des FRad-Chips zu verifizieren. Dieser Schritt ermöglicht es dem Institut mit dem Chip zuverlässige Strahlungsexperimente durchzuführen, um letztendlich ein neu entwickeltes Modell über strahlungsinduzierte Ladungen in Halbleitern zu kalibrieren.
Das entwickelte System besteht aus zwei Komponenten. Das "Field Programmable Gate Array" (FPGA) steuert den FRad-Chip an und liest die Messdaten aus. Das Computerprogramm bildet die Schnittstelle zu dem Benutzer, der einerseits die Messparameter anpassen kann und andererseits unmittelbar Rückmeldung über die Messergebnisse bekommt.
Ausgiebiges Testen hat eine Schwachstelle in dem Chip aufgedeckt, die eine Anpassung des Chipdesigns notwendig machte. Mit den funktionierenden Teilen des Chips konnte trotzdem ein erstes Experiment in einem Forschungsreaktor durchgeführt werden. Aufgrund der geringen Strahlendosis und Messdauer, konnten jedoch keine Effekte in dem Chip beobachtet werden.
Experimente in einer "micro-beam" Einrichtung könnten alle relevanten Bereiche des Chips individuell für kurze Zeit Strahlung aussetzen und somit effizienter und zielgerichteter Daten liefern, um den Chip zu analysieren.

Abstract

The Institute of Computer Engineering of the Vienna University of Technology developed the FRad-chip, an experimental application specific integrated circuit (ASIC) designed for the analysis of radiation effects in very large scale integrated circuits (VLSI). While simulations have been conducted successfully during the development process, the proper functionality of the fabricated chip has to be shown as well. However, as the chip has a non-standard interface and specific data integrity requirements, no means to access its functionality exist so far.
The aim of this thesis is the development of a reliable framework for accessing the capabilities of the FRad-chip and ultimately to verify the ASIC itself in a series of tests. This enables the project team to use the chip for calibrating their fault models.
The developed framework consists of two major parts. Firstly the field programmable gate array (FPGA) to control the ASIC and read out the measurement data and secondly a computer software to provide a front-end interface to configure the system and to analyse the resulting data on site.
Extensive testing of the ASIC samples have unveiled an issue within the ASIC and made it necessary to fabricate another batch of ASICs. Finally initial experiments in a research reactor have been conducted, however due to the nature of the reactor and the short experiment duration no radiation induced errors have been detected.
For a more efficient and comprehensive analysis of all ASIC areas a micro-beam experiment would be necessary to expose and test the individual target circuits in a controlled fashion.

Contents

CHAPTER 1

Introduction

1.1 Motivation

Decreasing structure sizes of VLSICs (Very Large Scale Integrated Circuits) lead to increasing vulnerability of the transistors to single event transients (SETs) caused by ionizing particles such as ions or protons. The particle rates electrical circuits are exposed to in aircraft and space-flight typically exceed the rates on the surface of the earth. However, further decreasing the transistor sizes in VLSICs also has lead to SETs being detected at ground level. An SET is only a transient change in the logic state of a transistor, however it can get latched and become a single event upset (SEU). The erroneous logic state of a SEU remains in the system, if it does not get corrected. SEUs can particularly be observed in big memories where structure sizes are small, the number of cells is high and error detection/correction techniques are applied [1]. The FATAL project group of the Technical University Vienna (Fault-tolerant Asynchronous Logic) in particular investigated the effects of radiation on VLSICs and developed models of circuit behavior when exposed to radiation [2]. To verify and calibrate the developed models, radiation experiments are necessary. For this purpose the FATAL project group has developed the FRad chip, an ASIC designed to reliably detect SETs and their consequences within simple circuits as well as measure their susceptibility to SETs. The project EASET (Accelerator-based Experimental Analysis and Simulation Modeling of Single-Event Transients) continues the work of FATAL by developing even more accurate analog SET models. The further development of the FRad chip, its fabrication and application in radiation experiments is realized in the Robust Nanoscale Logic Devices project (RoNaLD) of the Embedded Computing Systems Group. Its aim is to produce real measurement results using the FRad chip in order to calibrate the model developed in the EASET project. Before performing actual experiments with the chip, its functionality has to be verified. During development the VLSI design had to pass comprehensive simulations, however certain types of faults might not be visible in simulations. An

example are manufacturing faults which naturally can only be detected by abnormal behavior of the physical chip. So to ultimately verify the ASIC it has to be produced and tested under real conditions. This thesis addresses the verification of the chip including the development of the means necessary for accessing it. A previous work, "Reliable Gateway for Radiation Experiments on a VLSI Chip" [3], shows an approach to access an earlier version of the FRad chip. This thesis improves the proposed FPGA design to a more flexible framework capable of high speed measurements and data analysis. Furthermore, recent adoptions in the ASIC design have been incorporated. As a last step system tests of the whole measurement setup, including the ASIC, have been conducted to give a picture of the capabilities of the current setup. The knowledge gained in this step will be needed when planning experiments.

1.2 Problem Statement

The centerpiece of the FRad chip are target circuits, simple digital circuits that will propagate any SET to a system of counters that react on any signal activity, including SETs. The particular circuit design makes it possible to distinguish between normal activity and SETs during data analysis. In order to access the counter data, parallel-input/ serial-output shift registers (PISOs) provide an interface to the outside of the chip. As this rather simple interface is not supported by any computer, some kind of controller is needed to convert the data into a computationally processable form. In the previous controller version a field programmable gate array (FPGA) has been used to realize this controller. It controls the actions of the ASIC and forwards the measurement data to a custom software running on a PC. This is the general setup needed to run measurements on the FRad chip. Section 2 introduces the measurement setup in more detail.

As the design of the FRad chip and its experiment setup changed with the development of the RoNaLD project, the requirements of the FRad chip controller have to be adapted as well. The main changes can be divided into changes concerning the ASIC - FPGA interface and changes in the controller features to enable operation in various scenarios. The new requirements are listed in chapter 3.1.2. Implementing the extended requirements is a key element of the thesis.

Analyzing the measurement data is not straight forward, as the counters themselves can be affected by an SET. In this case the counter values are inconsistent. Therefore subtle interpretation of the measurement data by the analysis tool is necessary to detect those situations.

Another aspect is the verification of the setup including the controller and the front end software. The measurements should be able to run autonomously, without manual corrections over a long time period. To achieve this the setup has to be reliable. Errors in the setup are likely to invalidate a whole experiment session. As experiment sessions in a reactor or micro-beam facility are very time consuming in organization and preparation, verification of the system by simulating and testing is essential.

In addition, as the ASIC becomes available the reliability of its communication with the controller has to be verified.

The last problem to solve is built on the previous points and forms the actual research question. Are the provided FRad ASIC samples utilizable for radiation experiments? Before doing actual experiments with the chip, it has to be made sure all parts of the chip work as expected. This is challenging as the ASIC is optimized for operation rather than testing. As a result there is no dedicated test infrastructure implemented in the chip and all tests have to be based on the functionality used for normal operation. This makes testing of the chip more difficult. If certain parts do not behave as expected, a basic understanding has to be gained of what happened, whether all samples are affected and how to handle this problem. Either the problem just affects certain parts and is not relevant to others or there might be issues with an impact to the whole experiment. Possible solutions are to incorporate the gathered knowledge in the analysis of experiment results to come to reliable conclusions, to ignore the affected parts in the experiments, to make changes in the setup or to even change the ASIC design. The last option of course is the most serve one, since it requires new simulations, new fabrication of the chips and retesting. The empirical evidence used for reasoning about the research question is gathered by performing a number of verification tests on multiple ASIC samples, which will be introduced later.

1.3 Aim and Objectives

The aim of the thesis is to develop an FRad controller framework on the basis of the given requirements and thereby providing access to the measurement capabilities of the ASIC during the radiation experiments. On this basis the actual ASIC samples and PCBs should be analyzed whether they are ready for radiation experiments.

The objectives are:

- Develop a controller FPGA design and front-end PC software based on the previous version to control the ASIC, read out the experiment data and perform real-time analysis of the measurement results on site.
- Explore the possible cabling options between the FPGA board and the ASIC PCB with respect to performance, handling and price and find the maximum frequency of the setup.
- Elaborate a strategy for testing the ASIC after fabrication to ensure it can deliver useful results during radiation experiment.
- Elaborate a strategy for diagnosis to allow identifying the root cause of test failure (if seen) to either allow fixing the problem or identify working portions of the ASIC that can already be used in radiation experiments.

- Validate these strategies through actual test and diagnosis on physical ASIC samples.

1.4 Methodological Approach

The research methods used are a combination of the engineering design process with the goal of developing a reliable framework for the FRad chip and empirical data acquisition in order to collect data about the FRad chip samples under normal conditions as well as ambient radiation.

For developing a controller to match the needs of the project the controller requirements have been elaborated considering the constraints imposed by the FRad chip and the nature of the experiments. With the requirements kept in mind the characteristics of all cabling and controller hardware options (FPGAs, microcontrollers) have been compared and the best matching hardware and cables have been chosen. The relevant characteristics for the cables were the number of signals available, high frequency capability, susceptibility to electromagnetic interference, price of components and handling. For the controller hardware the processing speed, number and type of connectors, the input/output standards and the clocking capabilities, availability and price have been the decision criteria. A design concept suiting the chosen hardware has been developed, including state diagrams of the system modules, module interfaces and a clock domain model.

Once the concept was clear, the individual VHDL modules have been implemented. Since a Xilinx board has been chosen, the development environment was Xilinx ISE Design Suite 14.7, the most recent version of the suite.

After implementation the reliability of the framework has been assured by design verification. The VSim tool included in the Xilinx ISE framework has been used to perform behavioral simulations of the controller modules and subsequently integration tests by simulations of the whole FPGA design. Once the simulations confirmed the behavior of the design is correct, the design was uploaded to the FPGA and live tests of the FPGA components have been done using a logic analyzer to monitor certain control and state signals. This way the waveforms could be compared to the simulation results. Special attention has been paid to the chip communication to ensure exact timing of the control signals. The control signals all had to be monitored under different settings at different frequencies to ensure correct behavior. Any difference between the measured and expected results had to be used to improve the design. Also the impact of the cabling had to be investigated under different frequencies in order to find a reliable maximum frequency.

After successful FPGA verification the software was developed. Decisions about the operating system and programming language were made and the front-end tool as well as the data analysis tool were implemented. The data analysis software should satisfy the needs of careful data interpretation by performing feasibility checks to detect SETs

in the counters and readout logic. The software implementation had to be tested for correctness and as the FPGA design and front-end software were ready, their flawless cooperation has been verified using a logic analyzer as debug tool.

For verifying that the chosen cabling does not influence the ASIC - FPGA communication test measurements had to be performed. To get insight into the communication between the two boards, a logic analyzer and a scope were used, depending on the scenario. The logic analyzer is able to simultaneously show many more signals than the scope, however its results are digital only. The scope has only four channels, but gives analog results, which is necessary to detect and understand certain electronics problems, such as cabling issues or wrong logic levels.

As the framework development has been concluded, collecting the measurement data needed for the ASIC verification started. As mentioned before the ASIC only offers limited test access, so certain test patterns have been applied to the chip and the resulting data has been interpreted to conclude which functions work properly. This way insight into the behavior of the chip could be gained. There are dedicated test patterns to test the functionality of the LFSR and PISOs. The PISOs could be tested by resetting the counters and reading out the data. In this test case the data read out from the PISOs should equal the initialization values. The LFSRs could be tested by incrementing the counter values and again reading out the counters. The counter pattern was calculated and compared to the measurement data. Assuming the PISOs work properly, any data mismatch in this test indicates an issue with the LFSRs. For parts of the ASIC behaving unexpectedly the findings had to be confirmed by running the same tests on multiple samples in order to gain reliable data and exclude a subtle fabrication fault, PCB fault or bonding problem, which could not be detected in previous steps. In case some parts were faulty, the chip still might have been fit for a certain subset of experiments. The information about which exact target circuits or PISOs failed has been used to mask out erroneous parts in the data analysis.

To gather measurement data from a radiation environment, an experiment in a research reactor has been planned and carried out. The gained data gave information about the sensitivity of the target circuits to the radiation provided at the particular neutron source. Keeping the ASIC in the neutron beam and the controller outside was a challenge in the used radiation chamber.

1.5 Structure of the Work

Chapter 2 will give an overview of the final experiment setup and its purpose. It will explain the components of the experiment setup in more detail. Chapter 3 recapitulates the requirements of the previous FPGA controller version and points out the requirement changes in detail. It elaborates on the motivation to develop an extended version of the framework. Chapter 4 gives a more detailed insight into the structure and development of the framework. Chapter 5 is split up into the verification of the framework developed in the thesis, the verification of the ASIC board and the verification of the ASIC itself.

It will introduce the methods used for each task, followed by the verification results of the individual boards and chip samples. Furthermore it presents a radiation experiment performed in order to test the behavior of the FRad chip under neutron radiation. Chapter 6 points out optimization potential in all parts of the experiment setup. The last chapter 7 concludes the thesis with a summary of findings and future work.

CHAPTER 2

The experiment setup

This chapter starts by elaborating on the background of the experiment and afterwards introduces the FRad chip, the corresponding BCP and the FPGA board in more detail.

2.1 The experiment

In order to make predictions about the consequences of SETs in digital circuits, there are two common types of simulations. Physical level simulations with tools like Synopsis TCAD give very detailed information about the process. However, they are computationally expensive, only very small circuits can be simulated in such detail and they need information about the fabrication process, which typically is kept secret by the fabrication house. SPICE simulations on the other hand can simulate more complex circuits, however the SET itself has to be modeled in some way. Therefore precise models of the SETs are needed. The commonly used approach is using a double-exponential current model [4]. There are competing models of this type, however none of them perfectly describes the SETs.

In the EASET project a more accurate model of SETs is being developed. The purpose of the radiation experiments is twofold. In one type of experiment the aim is to gather data for calibrating the model for physical simulation (TCAD) of SETs. Here an ASIC with few simple target circuits is used, with on-chip analog amplifiers that allow observing the actual SET waveforms. Based on this physical model SPICE models can be calibrated [5, 6].

A second type of experiment is devoted to verifying predictions based on the new model and to collect data on SET distribution. The target of this experiment will be the FRad chip that this thesis is focusing on.

There are two kinds of experiment environments. Using a micro-beam facility the particle beam can be focused to small spots of the ASIC, typically in the range of micrometers.

This allows separate testing of the functionality of all target circuits and the measurement infrastructure elements. Using this very efficient technique detailed feedback of the functionality of the FRad chip can be gained, which is important for interpreting its results. Furthermore data of SETs within the circuits can be gained, which is required for further developing the SET models. One drawback is the extremely time limited access to micro-beam facilities.
The other kind of experiment will be conducted in an ambient radiation environment such as a nuclear reactor where in contrast to the micro-beam, the location of the hits and the particle rate cannot be predicted or controlled. Using an ambient radiation environment statistical SET data can be collected and in contrast to micro-beam facilities it is possible to perform long time experiments. A long experiment duration is necessary, since the location of the hits is randomly distributed and therefore sufficient test coverage can only be achieved by performing many measurements.

The data of both experiment types can be used to adjust the failure model for radiation induced transient faults in asynchronous digital circuits and to analyze and verify the redundancy technique and the ASIC itself proposed by [5].

The radiation experiments are conducted using the dedicated experiment ASIC, the FRad chip. It contains well chosen target circuits which can, if exposed to radiation, suffer SETs. These effects are recorded by the on-chip measurement infrastructure and can be read out by a controller, which at the same time coordinates the measurements. The data can later be analyzed on a PC. The FRad chip is, in contrast to existing SET test circuits, designed for long-term experiments. Figure 2.1 shows the overall hardware of the setup, although the final experiment setup details depend on the test facility used. For example the hardware could be extended by external trigger hardware or vice versa trigger an external radiation source.

2.2 The FRad chip

The FRad-chip is the core of the experiment as it is the only part exposed to radiation and performs the measurements. It is located on the ASIC board and controlled by the FPGA, which is spatially separated. The FRad chip consists of three main parts.

The **target circuits** are the actual circuits examined by the analysis. There are 7 types of target circuits such as inverter chains, inverter trees, NAND-NOR-trees, flip-flop chains, elastic pipelines, Sklansky Adders [7] and some custom circuits. To increase the SET value there exist multiple instances of most target circuits. In case a particle creates an SET within such a circuit, it will be recognized by the attached measurement logic.

The **measurement infrastructure** monitors the output of the target circuits and counts the SETs within the target circuits. The measurement infrastructure contains redundant counters for each target circuit, which makes it possible to accurately identify single and double hits in the counters or target circuits [5]. This is a realistic scenario in ambient radiation experiments, since the measurement duration can be controlled. In case the

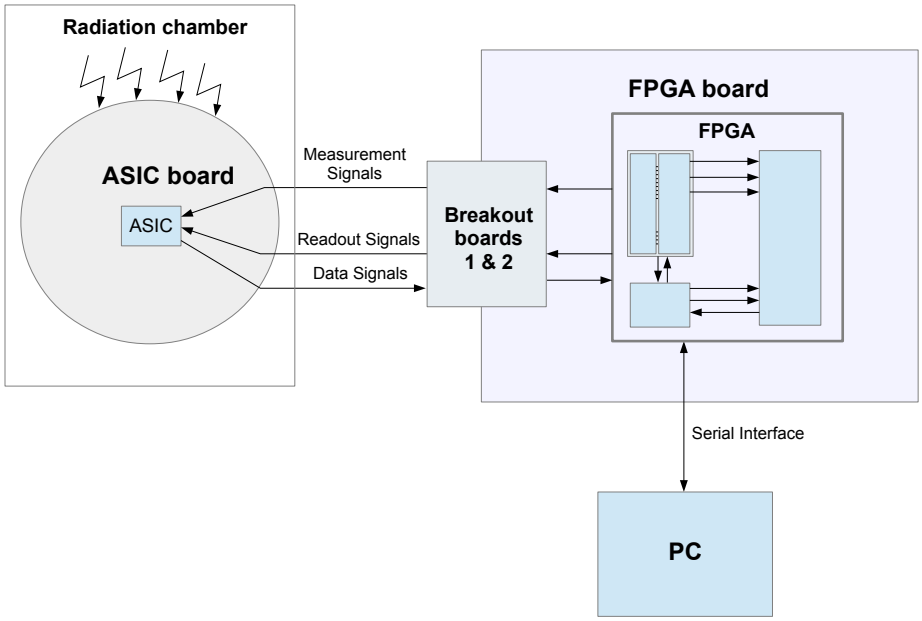

Figure 2.1: Overview of the chosen setup.

average number of hits becomes too high, the measurement time can be reduced through the front end software. This way the measurement time also can be increased, if the number of observed SETs is zero all the time.
Table 2.1 gives an overview of the counter combinations used for each target circuit. Most target circuits are implemented in combination with multiple counting schemes. This makes it possible to compare the counting approaches. For example the inverter chain exists with two different counter setups. In one version there are two 32 bit linear feedback shift registers (LFSRs) and two up-down counters connected to the inverter chain. The second version uses three up-down counters. The up-down counters are implemented as asynchronous elastic pipelines consisting of Muller C-elements [8]. This enables them to capture even very short SETs, in contrast to slow synchronous measurement techniques. The benefit of the LFSR counters compared to binary counters is a reduced area consumption and therefore smaller delays between the transistors and a higher maximum operating frequency. A drawback is the increased computation effort at the analysis step [9]. In contrast to the up-down counters SETs in the LFSRs can be reliably detected.

After each measurement the counter values are shifted into the parallel in serial out shift registers (PISOs).

Target Circuit	Measurement Circuits	PISO
Inverter Chain	2 32-bit LFSRs & 2 UDCs	84-bit
NAND-NOR Tree	2 32-bit LFSRs & 2 UDCs	84-bit
Flip-Flop Chain	2 32-bit LFSRs & 2 UDCs	84-bit
Elastic Pipeline	2 32-bit LFSRs & 2 UDCs	84-bit
Inverter Chain	3 UDCs	30-bit
NAND-NOR Tree	3 UDCs	30-bit
Flip-Flop Chain	3 UDCs	30-bit
Elastic Pipeline	3 UDCs	30-bit
Sklansky Adder	22 5-bit LFSRs	110-bit
Inverter Tree	15 5-bit LFSRs	75-bit
4:1 MUX	2 32-bit LFSRs & 2 UDCs	84-bit
4:1 MUX	3 UDCs	75-bit

Table 2.1: Target circuit overview and counter / PISO assignments.

The **PISO**s are the output interface of the chip and forward the data to the FPGA on request. The reason for this simple data output by shift registers is the strongly limited die area. As much of the area as possible should be devoted to the actual radiation targets, with as little overhead by infrastructure as possible. As a result the I/O logic and therefore also the protocol had to be kept simple. Each target circuit has its dedicated PISO which gets filled with the respective counter values at the beginning of every read sequence. Table 2.1 shows the counter and PISO assignments for each target circuit. There are six different counter combinations used for the target circuits. As a result there are PISOs with six different lengths (1x25, 18x30, 1x55, 3x75, 18x84, 3x110). In total there are 44 PISOs, resulting in 44 data output signals. After the measurement has finished the FPGA signals the ASIC to load the counter values into the PISOs. The PISOs are filled with the measurement data and read out three times each round to detect and correct particle hits within the PISOs. The protocol will be explained in chapter 3.2 in more detail.

In order to store the experiment data, a gateway between the simple high speed protocol at the ASIC output and the more sophisticated protocols of high level data storage devices had to be developed. The task of protocol conversion and data buffering is done by the FPGA as well as applying the control signals to the ASIC.

It is important to recognize that the redundancy and diversity of the counters as well as the temporal redundancy of the PISO readout avoids data corruption caused by SETs. SETs in the target circuits do not have a negative consequences, since the counters specifically have been designed to detect and count them.

Figure 2.2 illustrates the structure of the ASIC layout. As can be seen all three sections are separated in the design, so a micro-beam even using a big focus can be aimed at one

section without affecting the others.

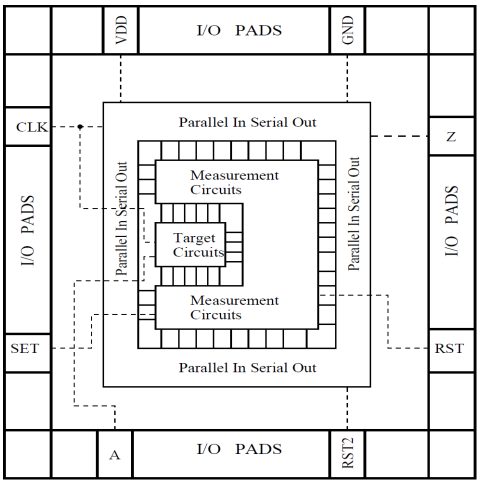

Figure 2.2: Structure of the FRad-chip.

Figure 2.3 is an image of the fabricated die. The pads are clearly visible on all four sides of the chip with a pad pitch of $80\mu m$. The square in the center of the ASIC confines the area with the target circuits. It provides an optical reference for micro-beam experiments. The ASIC is not packaged as the die should be exposed directly to radiation and a package would shield the chip to some extent.

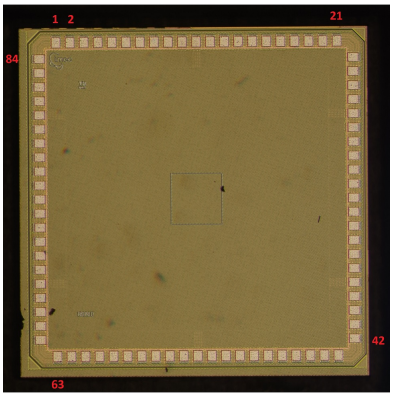

Figure 2.3: Microscope image of the die.

2.3 ASIC board

The PCB carrying the FRad-chip should be as small as possible in order to minimize delays and to match all size restrictions of the radiation chambers. It further should be

able to handle high frequencies with minimal interference between the individual signals and have a low delay in the signals. The PCB design and testing has been done by Václav Šimek, our partner in the Department of Microelectronics of the University Brno. He also organized and kindly managed the bonding procedure in Brno.

The layout of the ASIC board has been developed using Altium Designer and is shown in figure 2.4. The board embeds the FRad-chip (*IC1*), provides two LM1117 voltage

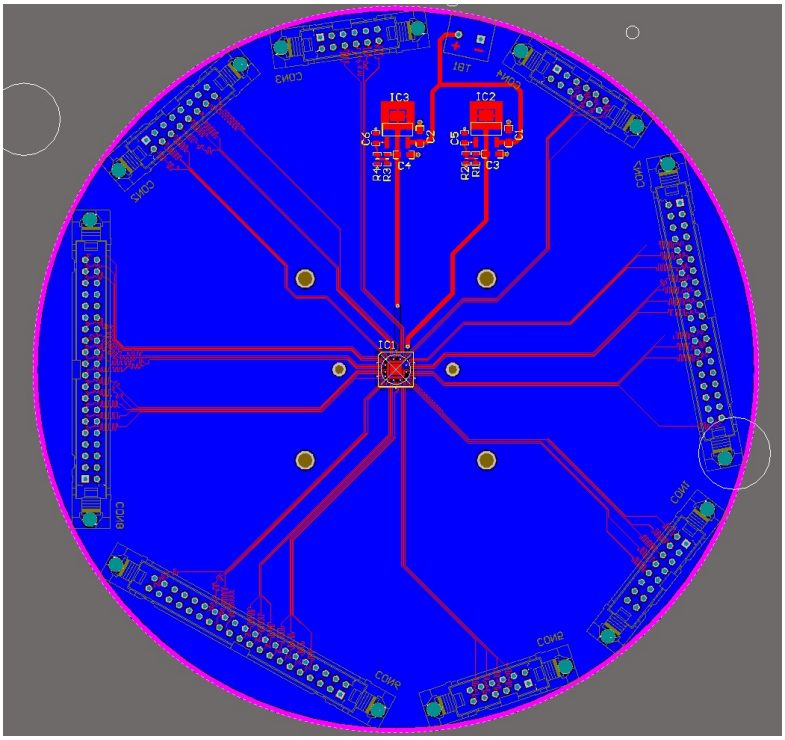

Figure 2.4: Layout of the PCB.

regulators for the core (1.25 V - *IC2*) and I/O (2.5 V - *IC3*) and connects the 60 ASIC control and data signals to 8 box headers (*CON1* - *CON8*). In the current board version ribbon cables are used for the connection to the FPGA board. In future versions of the setup they might be replaced by another solution. Section 4.1 elaborates on the cable options and the decision to use ribbon cables. The resistors and capacitors next to the voltage regulators are necessary for adjusting and smoothing the output voltage. The board is powered through a terminal block (*TB1*) by a conventional 5V power supply. Four holes in the PCB facilitate mounting a protective cover to avoid damage to the ASIC. The cover is removed only during the experiments.

As board substrate FR-4 has been chosen with a thickness of 1.5mm. The PCB layout has been realized as a double-sided design with the ASIC pads and most copper traces on the front side and all connectors on the back side. A special treatment has been

applied on the attachment locations for the bonding procedure using electroless nickel and immersion gold (ENIG) plating to ensure a reliable foundation for the bonding process. As the front side of the ASIC should be exposed to radiation, the back side of the FRad-chip has been attached to the PCB using a one-component epoxy based glue, Epotek H31D with Ag filler. In the next step the ASIC pads have been connected to the PCB by a gold bonding micro-wire using the wedge bonding procedure. Figure 2.5 shows the finished PCB with an FRad-chip.

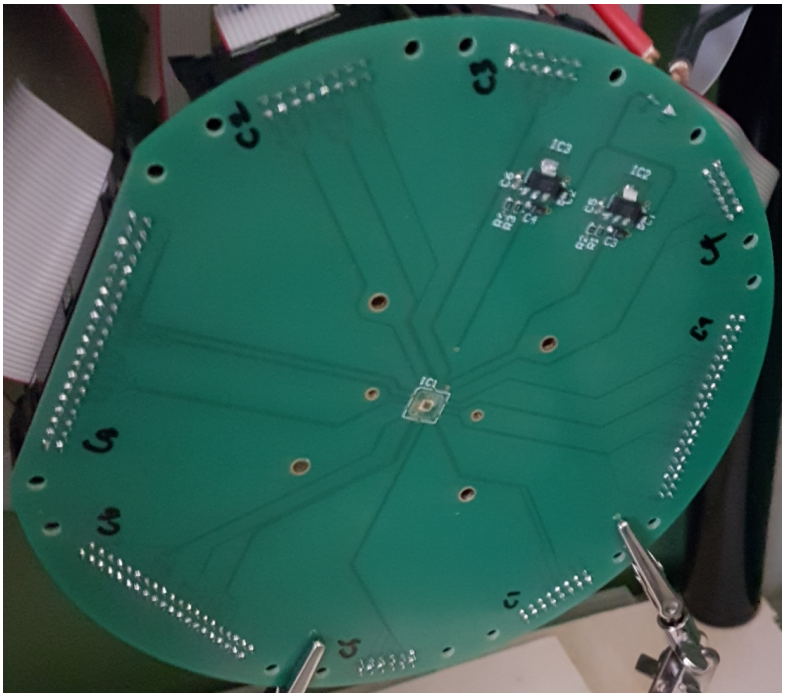

Figure 2.5: Image of the PCB with removed ASIC covering.

2.4 FPGA

The FPGA is used as the controller of the FRad-chip and is one focus of this thesis. It controls all input signals of the ASIC and processes its output signals. As stated above it acts as a controller, protocol converter and data buffer. These tasks have to be done reliably for a long period of time. However, even though the FPGA will not be exposed directly to the radiation, it still should be equipped with fault tolerance mechanisms to detect potential radiation induced errors.

The choice of the FPGA type depends on the available FPGA boards and breakout boards matching the project requirements. The large number of required high-speed connections strongly limits the choice of suitable FPGA boards. The final decision was to switch from an Altera board as used in the previous version to the Xilinx KC705

Evaluation Board [10] with a Kintex-7 XC7K325T FPGA [11], offering one high pin count (HPC) and one low pin count (LPC) FPGA Mezzanine Card (FMC) connector. Together the two connectors offer sufficient pins for transmitting all ASIC signals via the 2.5V LVCMOS standard and providing one GND pin between each two signals, minimizing crosstalk effects. Alternatively this setup enables us to switch to low voltage differential signaling (LVDS) for high frequency setups, depending on the experiment requirements and ASIC board specifications.

The requirements on the FPGA will be discussed in the following chapter.

CHAPTER 3

Existing approach and extended requirements

3.1 Existing approach

As stated in the introduction there has been a previous approach of developing a matching FRad controller using an FPGA. Back then the experiment setup was simpler. The FRad chip was supposed to be exposed to radiation for a fixed time of 42 seconds. After this fixed time the FPGA stopped the generation of measurement signals and applied the readout signals to the ASIC with subsequent data transmission to a PC. The experiment sequence was exactly the same every time and hence the program flow was static. A single static behavioral simulation would cover all possible states and the design therefore had the advantage of a good verifiability. The hardware components involved were the FPGA board HMX1-AS2 with the matching development board HPE_midi [12], both from Gleichmann Electronics Research. The FPGA in the setup was an Altera Stratix II [13]. This hardware was sufficient for the original controller requirements.

However the FRad interface as well as the assumed experiment conditions changed and the board did not suit the needs anymore, as will be discussed below. Another issue is the fact that Gleichmann Electronics Research does not exist anymore as the company it originally was. As a result getting additional products such as extension cards is not possible.

The ASIC-FPGA interface signals for data transmission in the original setup are shown in table 3.1.

These are all 16 signals used for the data transmission including the 100MHz clock, asic_clk. The asic_piso and asic_select signals are used to address the data which has to be shifted into the PISO. The asic_shift signal indicates that the FPGA is ready for

signal name	signal width [bit]	FPGA pin direction
asic_clk	1	out
asic_piso	4	out
asic_select	4	out
asic_shift	1	out
asic_ack	1	out
asic_data	5	in

Table 3.1: Pinout of the first FPGA-ASIC interface version.

reading the data and asic_ack acknowledges a previously sent data block. The actual data is transmitted by the asic_data signal.

As stated in the introduction chapter, the new FRad design does not have addressing signals, but outputs the data in a parallel fashion. Thus there is no need for asic_piso, asic_select and asic_ack anymore.

3.1.1 Original Requirements

This section summarizes the requirements of the previous design.

- The FPGA provides all control signals for the ASIC and processes the data output of the ASIC.
- The experiment data is irreproducible and if it gets lost anywhere along the track from the target circuit to the final storage, it cannot be recovered. So the framework has to ensure all data is transmitted reliably into the final storage. This requires accurate timing of the signals and a reliable program flow.
- The framework has to be reliable and no data may get corrupted by transmission errors or radiation effects inside the PISOs or FPGA. This is done by redundant readout of the PISO and redundant data storage within the FPGA.
  - The data has to be read from the ASIC three times reloading the PISOs with the counter data each time.
  - The data has to be stored within three independent memory blocks inside the FPGA.
  - It has to be ensured that the FPGA is not directly exposed to the radiation during its operation or at the preparation stage. This is mostly relevant for ambient radiation experiments, where the radiation cannot be confined to certain areas.

- The chip is designed for low particle rates, since only a few SETs can be distinguished within each interval. As a consequence the experiment has to run for a long time in order to collect all data required.
- Because in an ambient-radiation experiment the time and location of a hit cannot be predicted, but measurement data for all target circuits are needed, it must be possible to run those experiments in a stable fashion for many hours or even days.
- Experiment time is expensive. The measurement downtime has to be kept as small as possible in order to maximize the data yield and to minimize the number of missed events.
  - Communication between the ASIC and FPGA as well as between FPGA and PC has to be fast so the ASIC can resume the measurement quickly.
  - The measurement has to be automated as much as possible to ensure optimal usage of the measurement time and reduce the probability of errors.

3.1.2 Extended Requirements

As stated in the introduction chapter, the FRad interface and experiment conditions have changed and require an adaption at the controller side. This chapter deals with the changed requirements in more detail and pinpoints the impacts.

- **For every measurement run the following parameters should be adjustable at run-time on the front-end software.**
  - Frequency
  - Duration in clock cycles (Ignored if external trigger mode selected.)
  - Trigger Mode (external or internal)
  - Measurement parameters (pipeline-mode, SET0/1-setting, measurement-mode)

  The original solution was designed for a 42 second fixed length ambient radiation experiment. The current requirements, however, cover the use within ambient-radiation as well as within a micro-beam environment.

  In the ambient radiation experiment the ASIC can get hit anywhere at a particle rate which may be known but typically is not adjustable by the experiment operator.

  In the micro-beam experiment on the other hand the position of the hits can be controlled as well as the magnitude of the particle rate. This means no PISO or counter gets hit, if not desired. This is convenient for detailed verification of individual ASIC functions.

  A higher particle rate results in more measurement results within the given experiment duration. However as described in section 2.2 the number of SETs in a certain target which can be unambiguously detected and interpreted during each

measurement period is limited by the counter size. As a result the measurement duration has to be adjusted inversely proportional to the particle rate. As the particle rate is set during the experiment, consequently there has to be a possibility to adjust the measurement duration at run time as well.

This means the experiment duration has to be adaptable at run-time in dependence of the effective particle rate to make sure that the duration is long enough that some hits are recorded by the setup and that the duration is not too long and therefore the number of hits does not overstrain the measurement capabilities of the infrastructure. This enables us to optimally use the micro-beam facility for testing certain chip areas individually and to dynamically adapt the measurement to the current particle rate in real measurements.

The broader field of application comes at its cost:

All parameters adjustable at run-time are new requirements. The fact that those instructions have to be received at the FPGA, checked and responded to, forces us to use a more complex design structure, in contrast to the previous solution which had a static program flow. The change to a more flexible program flow has implications for the dependability strategy and verification of the FPGA as will be seen later.

- **The maximum target circuit operation frequency accepted by the framework should be as fast as possible.** To investigate the influence of the circuit activity on their susceptibility to SETs, a broad range of measurement frequencies is needed [14]. To get sufficient data within a reasonable time the used measurement frequencies should be as high as possible. In the original FPGA design the measurement frequency was fixed at 100MHz. Now the FPGA design itself should be able of handling even higher clock rates of up to 200MHz. The system outside of the FPGA such as the boards, cabling, connectors and pin layouts might impose restrictions on the clock frequencies, which cannot be influenced by the FPGA design. Therefore the 200MHz only have to be achieved by the FPGA itself.

  As all mentioned factors influence the maximum clock speed supported by the system, this frequency cannot be predicted and has to be determined through tests using the actual experiment hardware.

- **Each PISO type should have its own clock signal.** As there are PISOs with different sizes, one readout clock for all PISOs made data padding bits necessary in the previous version.

  Separate clock signals for each PISO type is an improvement that makes the ASIC signaling more difficult. From the FPGA design point of view the clock signals have to be generated appropriately and the data arrival has to be timed for each PISO type separately. Further these six fast clock signals have to be transmitted to the ASIC, which is a challenge to the cabling between the ASIC PCB and the FPGA board as well as for the boards themselves. Crosstalk effects can become a problem here.

The advantage is that only the actual data is sent to the FPGA and no dummy padding bits are used anymore.

- **The framework should support micro-beam as well as ambient radiation experiments.**
- **The experiment sequence still should be as autonomous as possible.** The dynamic experiment parameters should be specified in advance at the front-end which sends the instructions to the FPGA, waits for the data to be sent back and stores the data into the corresponding files. This way a high number of measurements can be performed sequentially without an operator interaction. Also a script can combine the front-end and analysis programs to create a automatic control loop.
- **The measurement data should be individually stored.** Each measurement run the data is read out and transmitted to the final storage device, independently of any occurring errors. The data should be stored in an own data file for each measurement, grouped by PISO for fast interpretation.
- **In case of an error, the error code should be transmitted as well.** In case of any error the associated information has to be sent to the PC together with the current measurement data. This way no data is lost and the state information supports the reconstruction of the cause of the error at the data analysis step. The original approach did not use any error logging to detect radiation induced errors in the program flow. Design specific errors were ruled out by exhaustive simulations and tests. As the change to a more complex design became necessary, error and information logging became part of the dependability strategy.
- **All transmitted information should be stored in a log file corresponding to its measurement.** Additionally to the data file each measurement run should have a file containing all settings and information related to the measurement run. In particular the dynamically adjusted parameters, actual measurement cycles and error codes have to be stored here.
- **An improved cabling solution should be used.** Since there are many signal lines, a cabling solution is needed which enables simple and reliable assembling of the setup. The FPGA board and constraints on the ASIC PCB further limit the options.

Table 3.2 summarizes the final FRad interface signals on the FPGA side.

It becomes apparent that the number of signals in the new interface is about four times higher than in the old one. This emphasizes the need for an FPGA board offering a high pin count connector.
The signals containing “PISO” in their names are related to the data output section of the FRad chip. They control the data readout. CNT_SET and the signals containing

signal name	signal width [bit]	FPGA pin direction
PISO_RST	1	out
PISO_SHIFT	1	out
PISO_CLK25	1	out
PISO_CLK30	1	out
PISO_CLK55	1	out
PISO_CLK75	1	out
PISO_CLK84	1	out
PISO_CLK110	1	out
CNT_SET	1	out
TAR_AD	1	out
RST_TAR_CNT	1	out
TAR_CLK	1	out
TAR_SET	2	out
TAR_READ	1	out
TAR_WRITE	1	out
25bit PISO I/O	1	in
30bit PISO I/O	18	in
55bit PISO I/O	1	in
75bit PISO I/O	3	in
84bit PISO I/O	18	in
110bit PISO I/O	3	in

Table 3.2: Pinout of the new FPGA-ASIC interface.

"TAR" (target) in their names stimulate the measurement itself. PISO_RST for example resets all PISOs while PISO_SHIFT enables the data shifting mechanism of the PISOs. The PISO_CLK signals provide the clocks for the readout logic. There is one clock for each PISO type. The PISO I/O signals transmit the output data.
The measurement interface is composed of the target circuit reset RST_TAR_CNT, the counter initialization CNT_SET, one common target clock TAR_CLK for all targets, two TAR_SET signals and one stimulus signal TAR_AD. TAR_READ and TAR_WRITE are the mode selection signals used for testing the elastic pipelines.

The exact protocol is covered in the following paragraphs. Figure 3.1 shows a waveform example of the relevant ASIC signals during a measurement.

The measurement sequence is initiated by the CNT_SET signal getting active (high) for the duration of one PISO_CLK cycle. Simultaneously RST_TAR_CNT gets active (low). Note PISO_CLK does not have to be active at this point, it is just important to keep the initialization signals active for the stated duration to give the ASIC enough time to reset its counters. Afterwards both signals go back to their original state.

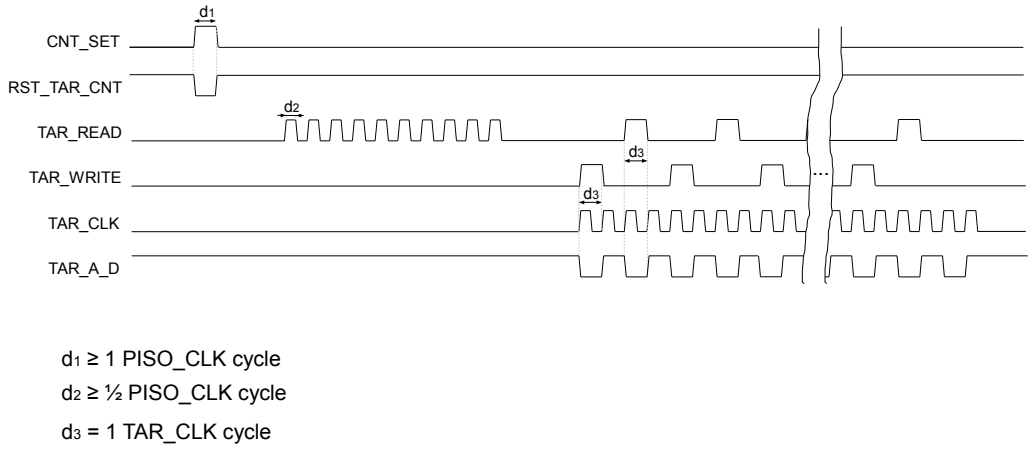

Figure 3.1: Control and data signals for a normal mode measurement

Now the *TAR_READ* signal gets toggled ten times by the same frequency as the PISO_CLK to clear the elastic pipeline. Again the clock is only shown for better illustration and does not have to be actually engaged.

This initialization is the same for all pipeline-modes. The chosen waveform represents a "normal mode" measurement. Depending on the pipeline-mode selected in the instruction message, *TAR_READ* and *TAR_WRITE* behave further as follows:

- **normal mode**: Both signals continuously toggle during the measurement. In this mode the pipelines alternate between empty and containing one element. This is the standard mode used for the experiments, the other modes are used for testing the elastic pipelines.
- **empty mode**: Both signals stay low. As the elastic pipelines got cleared in the initialization step, they will stay empty for the rest of the measurement.
- **full mode**: The *TAR_WRITE* signal toggles ten times after the *TAR_READ* signal in the same manner. With every clock cycle one bit is shifted into the elastic pipelines, as a result they will become full. Both *TAR_READ* and *TAR_WRITE* stay low during the measurement, therefore the pipelines stay full.

At this point the actual measurement starts with the TAR_CLK and TAR_AD signals starting to toggle. In "normal mode" measurements the TAR_READ and TAR_WRITE signals also start to toggle in the shown fashion in order to generate activity in the elastic pipeline target circuits. Their working cycle is 25% with a frequency one fourth of

TAR_CLK. Also the two signals are shifted to each other by 180 degree. The number of TAR_CLK cycles is set through the front end software or determined by the trigger input signal, depending on the applied settings.

Figure 3.2 sketches the waveform for the PISO readout. After the measurement has ended, the data of the PISOs has to be read out. However, the FRad specification requires a pause of 100 periods of PISO_CLK before the readout. After the pause the PISO_RST signal becomes active (low) for the duration of one PISO_CLK cycle and thereby resets the PISOs in the ASIC. The clock itself stays low during this time. At this point the data is shifted into the PISOs.

After the the PISO_RST signal got high again the FRad specification requires a 100 μs pause. Then the actual data transfer starts with all PISO_CLK signals starting to toggle. When the clock is low for the first time, PISO_SHIFT becomes assigned to high. This causes the PISOs to shift out the next bit with every falling clock edge. The shift signal stays active for 111 PISO_CLK cycles. During this time the PISOs shift out their data on the falling clock edges depending on the lengths of the PISOs. For example the 25 bit PISO shifts only during the first 25 cycles, as the PISO_clk25 signal stops after 26 cycles. The 110 bit PISOs will shift on the first 110 cycles. In the 111^{th} cycle the shift signal goes low again, ending the transmission.

Next the PISO_CLK signals keep toggling and the whole readout procedure including the reset will be preformed two more times to provide time redundancy.

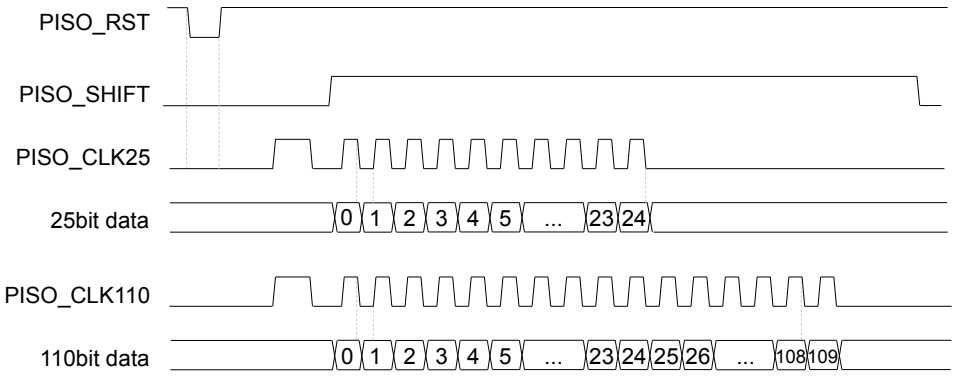

Figure 3.2: Control and data signals for PISO readout

CHAPTER 4

Implementation

As explained in the requirements chapter, the main tasks of the controller framework consist of sending the control signals to the ASIC, reading out its data, storing them on the PC and analyzing them. It therefore consists of three parts: The **FPGA design** acts as a link between the FRad chip and the permanent storage device, which is a PC in our case. The **software front-end** on the PC communicates with the FPGA, supplying it with measurement instructions and receiving the measurement data. Finally the **analyser software** interprets the bit stream read from the chip and summarizes the findings.

4.1 Controller FPGA design

This section deals with the structure of the FPGA and its design decisions. It will explain the modules in detail and how they interact.

Figure 4.1 gives an overview of the FPGA design and figure 4.2 is a schematic activity diagram of the FPGA. For connecting the FPGA and ASIC boards using FMC compatible cables, breakout boards in combination with ribbon cables and breakout boards in combination with coaxial cables have been considered. Coaxial cables can handle higher frequencies than ribbon cables, but would make the PCB design and handling of the cables very difficult, as there are 60 signals. VITA 57.1 high pin count FMC connectors offer a sufficiently high number of high speed signals and are directly supported by the FPGA board. So no breakout board would be required. However, the cables are expensive, the high pin count on a small area makes the connector difficult to solder onto the PCB and routing that many traces from only one connector to the ASIC is difficult. So it has been decided to use breakout boards with ribbon cables for the verification setup. They may not be optimal for high frequency signals, but offer a sufficient number of pins and are the most simple solution regarding handling. To reduce crosstalk between the

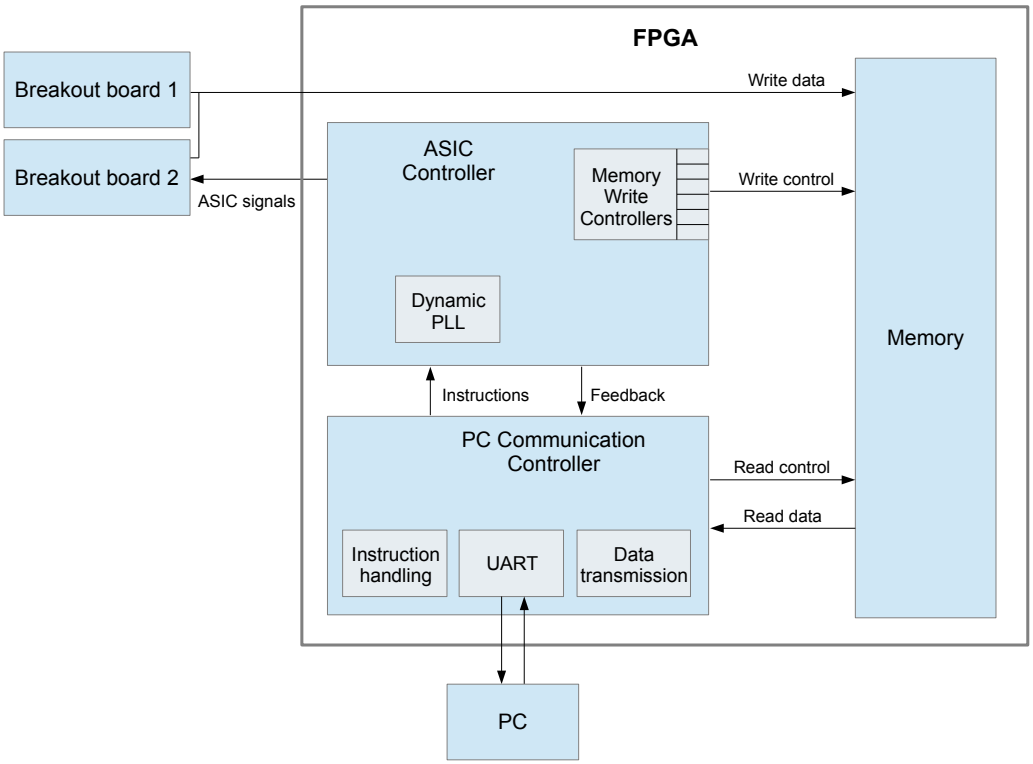

Figure 4.1: Overview of the FPGA

signals, every second wire has been set to ground on the FPGA side. For high frequency setups it is easily possible to exchange the cables using another ASIC PCB.

Another consideration was using LVDS instead of single-ended signals. Typically signals are single-ended, meaning their potential is measured relative to a common ground. This is simple to implement and needs only one wire per signal plus one ground common to all signals of the design. However, for high frequencies and/or long cables crosstalk poses a problem to this technique. Low-voltage differential signaling (LVDS) uses two wires per signal, where the potential is measured between those two signals, instead of a ground signal. The benefit is that both wires follow nearly the same physical path and therefore show nearly the same pattern of electromagnetic noise. As the logic level of the signal is determined by the potential difference of the two wires, the noise of both wires cancel each other and have much less influence on the logic level detection, compared to single-ended signals. On the backside LVDS is much more effort to implement, as it requires two wires for each signal, the right kind of cabling and more complex trace routing as well as termination electronics on the ASIC PCB. For experiment conditions that necessarily require high frequencies, this transmission technique seems to be an adequate solution. However, for the experiment environments currently available to us

the single-ended solution is suitable.

The price for using this simple cabling solution is a lower maximum frequency for the measurement as well as data readout and hence a slightly lower utilization of the experiment time for the actual data collection.

The utilization factor depends on the measurement frequency as well as measurement duration, which depends on the particle rate. The utilization can be calculated as follows:

$$utilization = \frac{Measurement\ duration}{Measurement\ duration + Readout\ duration} \tag{4.1}$$

In ambient radiation experiments the particle rate is low, therefore the measurement duration is higher. For an experiment lasting forty seconds and using long cables with a data transmission of 100kHz, the readout duration is about 10ms and the utilization nearly 100%. In a micro-beam experiment with a measurement duration of 100ms, the utilization decreases to 90%. Compared to the Serial communication, however, this transmission time is insignificant.

The communication with the host PC is done by the PC Communication Controller. Using the serial interface it repeatedly sends "beacons" to the PC in a certain interval showing its presence. As the PC becomes available, it will respond with an instruction message, containing the information about the measurement to perform. The message includes a hash which will be checked for correctness within the PC Communication Controller. A message will be sent back to the PC indicating whether or not the check was successful. In case it fails, the FPGA will again start to send beacons and wait for an instruction. Otherwise it will pass the instruction to the ASIC controller which in turn will start to process it.

Once the ASIC controller gets an instruction it will start to configure the frequency of the measurement PLL according to the divisor and multiplier factors given by the instruction. This takes a few microseconds, after which the FPGA is ready to be triggered.
It will signal its ready state to the environment on a 5V output pin, which can be used to trigger other external experiment equipment if needed. Depending on the *Internal_Trigger* bit of the instruction, it will immediately trigger once it is ready or wait for the external trigger to get active on a separate 5V input pin.

Once the measurement is triggered, it will initialize the ASIC by setting the counter values to their defaults and start the measurement. The exact procedure has been explained in 3.2.

As soon as the configured measurement duration expired, the measurement is complete and the ASIC controller will initialize the PISO readout and subsequently start the actual readout. It therefore activates the *Memory Write Controller* modules. There are six such modules, one for each kind of PISO. They generate the address and enable signals for the memory write accesses. Those memory control signals are synchronized with the data coming from the ASIC in such a way that the data transmission is possible with

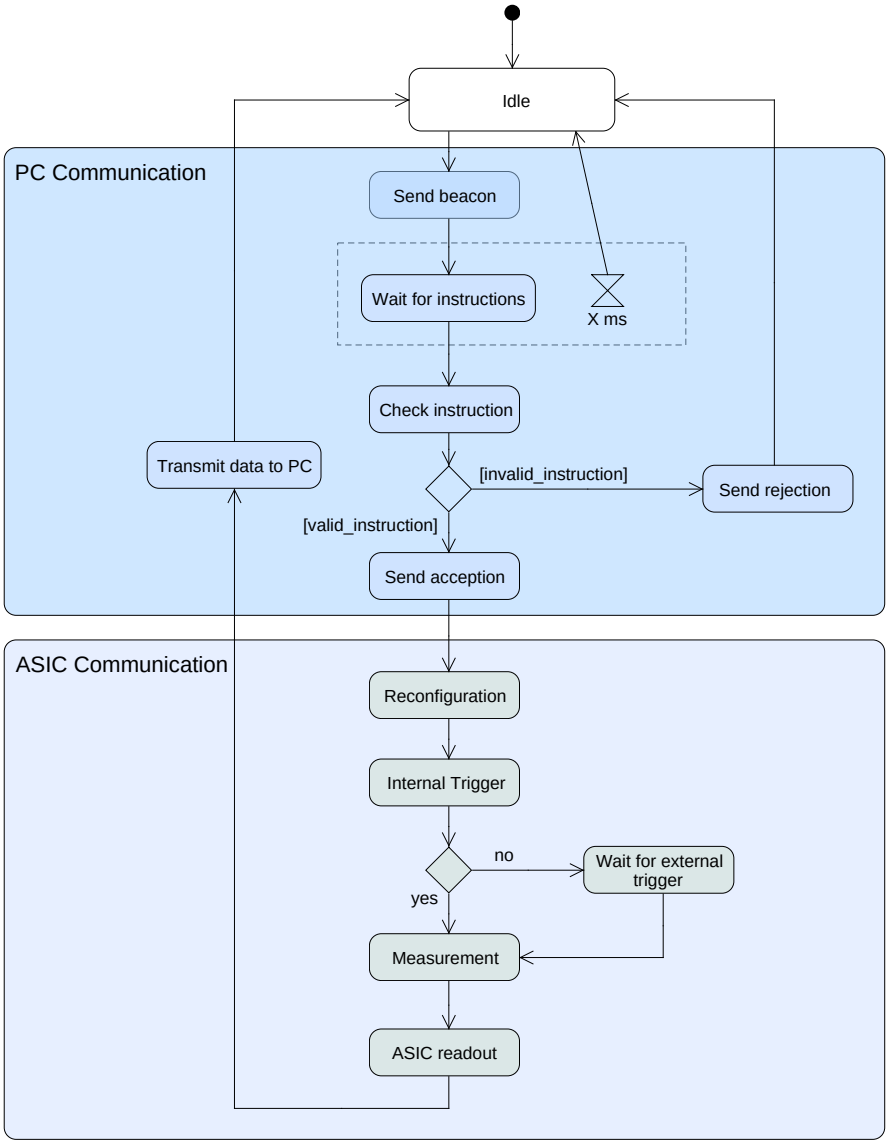

Figure 4.2: Activity diagram of the FPGA

high frequencies, if crosstalk effects are avoided in the cabling. Also the cable delay is compensated by the framework. A delay of 3.9ns per direction is expected. Since 60cm ribbon cables are used, 3ns are needed to pass only one cable. The remaining 0.9ns are due to FPGA board and ASIC board delays.

The memory elements used for storing the data have been implemented as a dual port ram, so the read clock is independent of the write clock. Through the separation of data

PISO type	Efficiency	Total bytes
25 bit	69%	39
30 bit	96%	630
55 bit	74%	83
75 bit	60%	450
84 bit	96%	1764
110 bit	60%	660

Table 4.1: Efficiency of the transmission protocol per PISO type.

collection from the ASIC and forwarding to PC into disjoint phases writing and reading are mutually exclusive in our design. This is beneficial for us, since it separates the two clock domains of the fast data signals coming from the ASIC and the slow read out accesses.

As a time redundancy strategy all data will be read from the ASIC three times before the PC communication controller module is activated again in order to forward the data to the PC. For forwarding the data the communication controller exclusively controls the memory module to avoid inconsistencies.

The data of every PISO is stored in an own memory block and all data is stored redundantly in three different memory blocks. So it can be ensured no data is lost if a memory cell breaks or if transmission between FPGA and PC temporarily fails.

The communication controller reads out the data, packs it into 8 bit packages and sends it via the serial interface to the front-end software running on the PC. The data conversion and transmission starts with the data of the smallest PISO (25 bit) and ends with the data of the biggest PISOs (110 bit).

The structure of the PISO data and its alignment in the memory make it necessary to introduce padding bits for the serial transmission. The efficiency of the packing (the ratio of data symbols to transmitted symbols) differs for every PISO type and is given in table 4.1. The overall throughput reduction caused by padding is 16%, which seems acceptable. A more complex transmission logic would increase the efficiency, but also the complexity of the design. So the implemented solution was a trade-off between efficiency and packing logic complexity.

Once all ASIC data is transmitted to the PC, the FPGA sends its own status information such as the number of actual *TAR_CLK* cycles sent and failure codes. Using a baud rate of 230 400, the serial transmission lasts takes about 130ms.

Then the procedure is complete and the FPGA again starts to send beacons to the PC, indicating it is ready for the next measurement.

The implementation has been kept as flexible as possible using VHDL constants. The PISO frequency, PISO scale factor, measurement scale factor, beacon interval and certain

hold times can be adjusted, to mention just the most important settings.

4.1.1 Clock distribution

As changing the measurement frequency during runtime is a major requirement and cable delay has to be compensated for, the design includes means of clock manipulation, such as PLLs and scaling techniques. This chapter describes all major clocks within the design, their purpose, how they are created and how clock domain crossings are handled. For clock generation the KC705 evaluation board comprises an SiT9102 fixed frequency oscillator. It provides a 2.5V LVDS signal with a fixed frequency of 200 MHz.

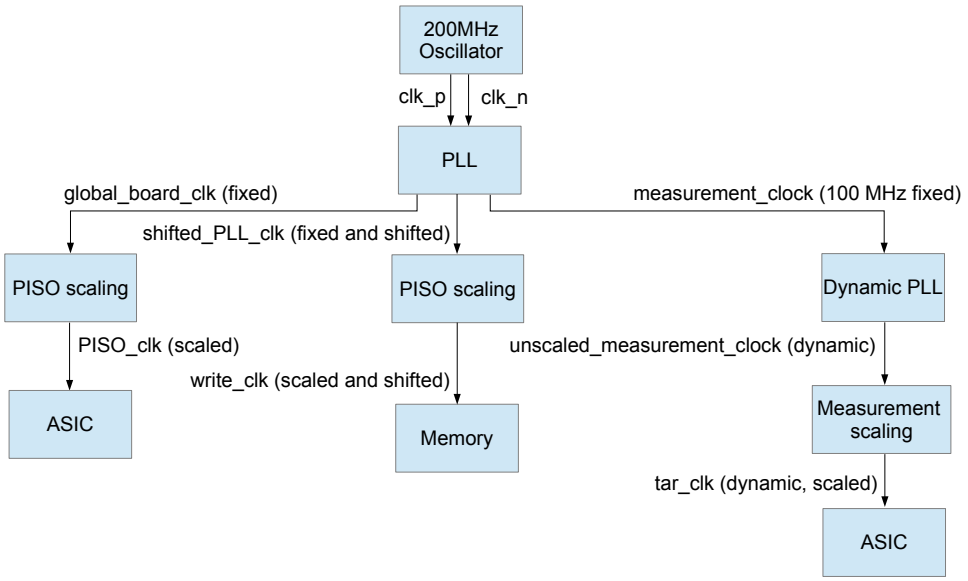

Figure 4.3: Clock hierarchy of the FPGA design.

Figure 4.3 illustrates the clock hierarchy within the design. As mentioned the clock signals *clk_p* and *clk_n* are generated by the 200MHz fixed frequency oscillator of the board. This differential pair drives the PLL which generates three clock signals:

- *global_board_clk*: The main clock running the main components of the design at a predefined frequency f_{PISO_base}. It is also used to generate *PISO_clk*, the PISO readout clock signal, by scaling down the *global_board_clk* signal by the predefined scale factor *PISO_scale*. This scale factor can be easily changed in the code, however in this case the FPGA design has to be resynthesized and uploaded again. We chose *PISO_scale* to be static, because the read out frequency is expected to be constant during an experiment, as it depends on the measurement setup. That means *PISO_scale* is a VHDL constant and cannot be changed at runtime. Also f_{PISO_base} can be easily changed in the design, if desired. The scaling of clock

signals has been implemented as a counter, counting to the preset scale factor, before toggling the output clock.

- *shifted_PLL_clk* is *global_board_clk* shifted by the small delay of the ASIC cabling. The shifted signal is then as well scaled by *PISO_scale* to match the frequency of *PISO_clk*. The resulting *write_clk* clock is used to provide a delay compensated write clock to the memory blocks. This way the write-clock of the memory is precisely in sync with the data coming from the ASIC. This is an essential aspect of the high speed data transfer.

- *measurement_clock*: A 100 MHz clock which is the input for the dynamic PLL used for generating the measurement clock. The dynamic PLL converts the 100 MHz clock to an intermediate frequency, which is then scaled by a preset factor *measurement_scale_factor* to the final measurement clock, *tar_clk*. As the dynamic PLL only has a limited range of divider and multiplier values, this second scaling stage is needed to provide lower frequencies if needed. It has been decided to use static scaling for the second stage, as the order of magnitude of the measurement frequency depends on the setup and typically does not change during an experiment. In case there is need to drastically slow down the measurement frequency, for example changing *tar_clk* from 100MHz to 100Hz, another bitstream file with an adapted scale factor can be uploaded to the FPGA.

Special attention has been paid to the clock domain crossings to avoid problems with lost data or metastability. Concerning the crossing between the read clock and write clock domains of the memory, the write and read accesses are mutual exclusive with sufficient pauses between the write and read section. For the other clock domain crossings handshaking is applied to ensure the integrity of the signals.

4.2 Front-end software

The front-end software is the part of the framework the user can directly interact with. Its main purpose is to send the measurement instructions to the FPGA, reliably communicate with the FPGA, monitor and log the FPGA controller status and store the measurement data. It can also be easily used by an external script which processes the measurement results and calculates the next measurement settings. This way radiation experiments can be fully automated and do not require human interaction. The analysis software checks the data of the PISOs and gives immediate feedback about the results. In combination with the front-end software it can be used as a control loop to keep a defined average of SETs per experiment. Constants have been defined in the code to allow for quick changes in the settings such as the baud rate.

4.3 Analysis software

The analysis tool is a PC program with the main purpose of performing checks on the measurement data. It should quickly interpret the data and present the results as output on the screen or as a return value, depending on whether a human experiment operator or a script handles the analysis. In case a script handles the measurement and the analysis, it is possible to create an automated control loop to adjust the measurement parameters of the next measurement based on the feedback of the analysis results. In case there are too many SETs, the experiment duration can be shortened. If there are no SETs, the experiment duration can be extended.

The results of the analysis software contain information about different parts of the ASIC.

Firstly the program parses the bit stream data file stored by the front-end software and groups the data by PISO. Then it runs a redundancy check to see whether there are inconsistencies indicating a hit in the readout logic of the ASIC or in the FPGA RAM. This is done by comparing the redundantly stored data for differences.

It then splits up the PISO data at the counter level and checks whether any up-down counter values have changed during the measurement. This would indicate a hit in an up-down counter or the connected target circuit. This process is a simple comparison of the up-down counter values with the default values. The expected counter values are known, since U/D counters always are “0000011111” or “1111100000” in normal state. The next step is to check whether the LFSR counters match their expected states. A mismatch here would indicate a hit in a target circuit or LFSR counter. The expected states can be calculated using the number of transitions during the measurement. This is easy for the 5 bit LFSRs as there are only 32 possible states, but for the 32 bit LFSRs it is a computational challenge. Chapter 4.3.1 discusses this problem in more detail and presents a suitable solution.

All of these calculations are based on functions like calculating the n^{th} state of a certain LFSR or checking whether a certain LFSR state matches the expected state after n iterations. These functions have been implemented in an efficient way and might be needed by control scripts or other future applications. Therefore the analysis tool interprets command line parameters in order to determine whether it should run a regular data analysis or a specific LFSR computation based on other command line parameters. The complete list of commands can be found in the appendix.

The following command line call instructs the analysis tool to check whether the data file “data_dump.txt” contains the data expected after 6000 cycles.

```
./analyzer -f data_dump.txt -i 6000
```

In the next example the state of a 32 bit LFSR after 10 iterations is requested.

```
./analyzer -r -t 10
```

4.3.1 LFSR shortcut algorithm

When the state of an LFSR counter of the measurement data is checked, it is compared to a reference bit pattern, the expected state. If they are equal, it can be concluded that the connected target circuit worked as expected and there was no SET within that circuit or the LFSR itself. If they are not equal, one explanation are SETs in the target circuit. In this case the LFSR counter should show one additional transition for each SET. So the measured LFSR state must be a neighboring bit pattern of the expected state. Therefore the LFSR value is compared to the neighboring states of the expected state in order to find the relative LFSR position of the counter pattern. If even in the neighboring states no match has been found, an SET in the LFSR is assumed. In this scenario the LFSR can have any arbitrary value, so the algorithm terminates.
As the LFSR state and the expected state depend on the number of measurement cycles, the analysis software needs some means to find the LFSR bit pattern for any given number of LFSR iterations.
Efficiently finding the expected state for any number of iterations n is the aim of the method described here. We will also compare naive approaches with our solution.

One obvious approach to find the expected state of n iterations is to iterate through the LFSR n times. For the worst case it is obvious that calculating all 4.3 billion 32 bit LFSR states is time consuming. The worst case time for this approach was around 120 seconds on our hardware. Another naive approach, reading them from a pre-calculated file at run time is space (16GB) as well as time consuming.

A solution has been found by combining the two approaches and storing just certain states into a shortcut file. This tremendously increases the efficiency.
The states are partitioned into $N = \lceil \frac{2^{32}}{M} \rceil$ blocks with M states each. The last block might have less than M states, but for the explanation also the last block is assumed to be full. Every block B_i where $i = [1, N]$ contains the states of the LFSRs in the interval $I_i = [(i-1) * M, i * M - 1]$. Instead of storing all states, just the first state of each block is stored together with the corresponding sequence number as a shortcut into a file. Since the number of transitions all LFSRs should have gone through is known, it can be calculated in which block the expected state is. In case there have been some SETs in the connected target circuit, the sequence number will slightly differ from the expected one. The counter will show some transitions more than expected, but still be in a state close to the expected one. The LFSR bit pattern might be totally different though.

If the user is interested in the sequence number of a particular counter value cv_x, the algorithm reads in the precalculated shortcut file and jumps to the first state of the block containing the expected state $B_e = B_{\lfloor \frac{x}{M} \rfloor}$. From the corresponding shortcut state $s_{shortcut} = (\lceil \frac{x}{M} \rceil - 1) * M + 1$ it continues to iterate through the LFSR states until the iterating state matches the LFSR counter value cv_x the algorithm wants to find. The counter value should be either in the expected or the subsequent block. The subsequent block also has to be considered for the case that the counter value was expected at the very end of the block, but left the expected block due to SETs. If there is no match

within $2 * M$ transitions, there can be two reasons. Firstly the LFSR counter can be hit directly or there have been more target circuit hits than there are elements in M. To avoid the latter choosing a block size that is big enough is important. However if the block size is chosen as described below, it will perfectly suit this criterion. In case there was an SET in an LFSR a reasonable match most likely will not be found with this method. Still it immediately can be concluded that there was a SET in the LFSR. In any case the algorithm returns useful information. Either it returns the index of the counter value or if it cannot find the counter value it returns that there has been a hit in the LFSR.

The performance increase of this method depends on the choice of the block size M and the hardware used. The following chapter analyzes the performance of the algorithm.

4.3.1.1 Worst Case Run Time and Memory Analysis

The naive approach of iterating though all states has a worst case execution time of 2^{32} transitions. The presented algorithm will iterate through two blocks at most, so its worst case execution time is $2 * M = 2 * \frac{2^{32}}{N} = \frac{2^{33}}{N}$ transitions. So in a worst case comparison the number of transitions calculated is less by a constant factor of $\frac{N}{2}$. Reading the shortcut file from the hard drive is an additional constant overhead, although negligible if a feasible block size is used. Further it does not have to be reloaded for each calculation. The memory consumption for the shortcut values is $4 * N$ bytes. Using a reasonable block size storing the data in the RAM is no problem, in contrast to the approach of storing all 16GB of transitions data.

As the algorithm is a trade-off between storing and calculating all states, the run time increases for very small and very big block sizes. A block size of 2^{32} is equivalent to computationally iterating through all states at run time. A block size of one on the other hand is the same as the approach of storing all states on the hard drive. A good block size naturally is in between and slightly depends on the system used. To find a good block size for our system the worst case performance of the algorithm has been tested over different block sizes. The results are illustrated in figure 4.4. The optimum has been found at a block size of 100 000 to 300 000, where any LFSR state can be found within under 20 milliseconds. This is a remarkable performance improvement compared to the 150 seconds worst case calculation time of the brute force approach. Using a block size in the suggested range leads to a shortcut file of about 100 kB. This is five orders of magnitude smaller compared to the 16 GB needed for storing all states.

This algorithm helps the analysis tool to fulfill the need to get instant feedback of which parts of the ASIC got affected by the radiation.

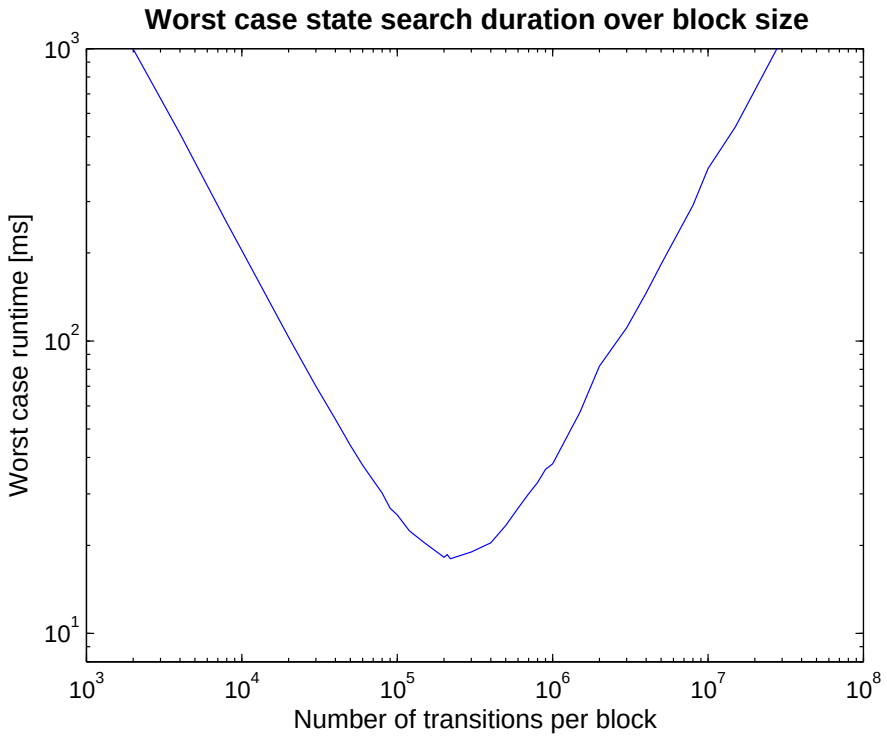

Figure 4.4: Worst case run times of the LFSR32 search using the shortcut algorithm with different block sizes.

CHAPTER 5

Verification

The availability of the ASIC and the PCB is required to verify the hardware setup and the ASIC. Since testing and debugging is time consuming and the exact date of the arrival of the bonded ASIC was unknown, firstly those parts of the framework that do not rely on the ASIC have been tested as far as possible. Any issues found at this stage could be dealt with immediately and as the bonded chip arrived, the whole setup including PCB and chip has been tested.

For the first step a test module has been created inside the FPGA to simulate the expected behavior of the ASIC. It was internally connected to the FPGA output/input signals instead of the real ASIC board. This way the framework could be tested and debugged before the actual chip was physically available for testing. Those tests aimed at finding problems with the data flow or control logic of the FPGA, with the PC - FPGA communication and the front-end software. This process is described in section 5.1.

As soon as the chip was available, the test module has been removed and the signals routed out of the FPGA to the real ASIC board. At this step it has been tested whether the cabling, board and ASIC behave as expected, whether there are any bonding or fabrication defects or any other unexpected behavior in the counters or PISOs. Chapter 5.2 and 5.3 explain these tasks in detail. The chapters 5.4 to 5.8 summarize the test results for each tested board and ASIC.

5.1 Framework Verification

As mentioned above, an own module has been developed to simulate the behaviour of the ASIC. It generates counter values in the same fashion as the real chip does and it shifts them out with the PISO clocks. This way it can be tested whether the data flow from the chip to the PC is fault free before the actual chip is available. This makes it possible to focus on verifying the chip once it is bonded to the PCB.

5.1.1 Simulations

The first step of the FPGA verification consisted of individual detailed ISim simulations for all modules during their development to test and ensure their correct behavior. Once the design was completed, an integration simulation has been performed to analyse their interaction and check their correctness.

As an example of the integration tests two simulated waveforms are depicting write accesses to the memory module for the 25 bit PISO. Figure 5.1 shows how the data is stored at the first out of three redundant transmissions. As can be seen the memory clock used for write accesses *wr_clk* is 10MHz in this example. The second clock *asic_piso_clk25* illustrates the clock signal which is routed to the ASIC. The slight shift between the two clocks has been introduced to compensate the signal delay from the FPGA to the ASIC and back and is about 7 ns.
Once the write process is enabled, the address vector *waddr2* is valid and *wr2* is set active to enable the writing at every rising clock edge. Observe *ram* changing as the memory is filled with data. With every cycle the address is increased by one until address 24 is reached. Since this is the simulation of a 25 bit PISO memory, this is the last cycle. The write enable signal will go low again and the address stops to increase.

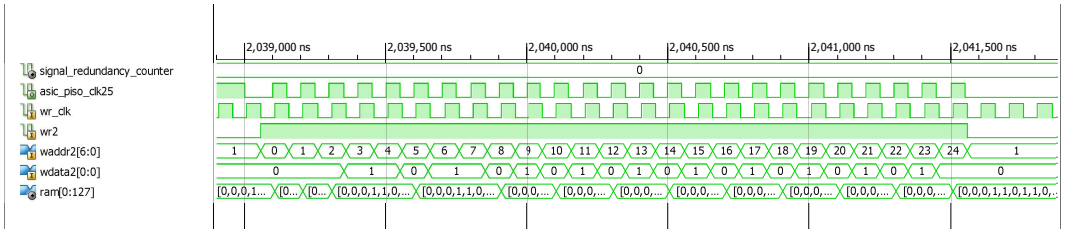

Figure 5.1: Simulation waveform of a memory module for a 25bit PISO. The trace shows the first storage process.

Similarly 5.2 illustrates the storage of the second readout for the same PISO. Thus *signal_redundancy_counter* is increased to 1 and the address continues at 25. After the third retransmission the address counter will halt at 74 which is the last address available in this particular memory. The biggest memory blocks used are for the 110 bit PISOs and comprise 330 addresses.

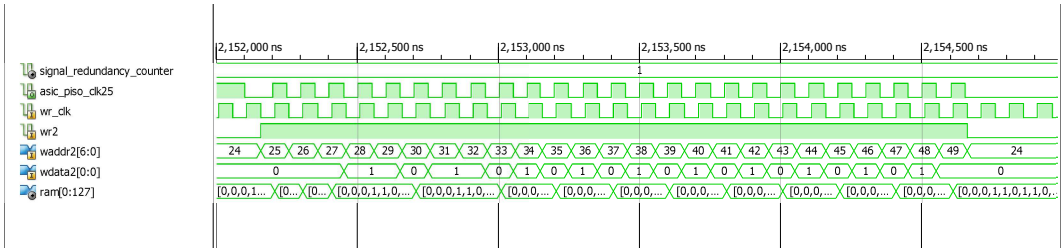

Figure 5.2: Simulation waveform of a memory module for a 25bit PISO. The trace shows the second storage process.

All components have been checked in a module test in this fashion for their correct behavior and corrected in case of problems.

Subsequently the verified modules have been integrated into the top level design and integration tests have been performed on the whole design. As a substitute for the front end software the test bench imitated its behavior and sent instructions to the controller design. The signals of the design then have been analyzed using ISim.

Once the simulation results have been satisfying, it has been switched to using the real FPGA together with the front end tool to test all, the FPGA program flow, the front end tool and their communication. This will be shown in the following.

5.1.2 Live verification

Once the simulation results fully matched the expectations, the live verification started. This means, in contrast to the preceding simulations, in this step the design had to be verified in real hardware. Incorrect simulation assumptions or implementation could mask errors that can cause problems during operation and have not been revealed during the simulations. Also having the design run in real-time rather than in a slow simulation allows more comprehensive testing. Therefore extensive live testing is an essential part of the framework verification.
For this step the ASIC test module within the FPGA was used again and the FPGA was connected via its serial interface to the PC. The front end software then has been used to send an instruction to the FPGA. The experiment controller executed the instruction, sent the corresponding signals to the FPGA child board pins as well as to the internal ASIC test module, read out the internally generated ASIC data and sent it back to the PC.

Since the onboard debugging capabilities of the Xilinx ISE framework are limited and not flawless, debug outputs on the ASIC signal pins in the top level module have been used in combination with a logic analyzer to get insight into the FPGA signals. Using the Agilent Technologies 16802A logic analyzer and the Xilinx FMC XM105 debug cards of the experiment setup, it is possible to trace up to 64 signals simultaneously.

The live verification consists of:

- **Verification of the individual modules**
  Tracing 64 signals is sufficient to monitor the most important signals for each module to verify their correct behavior in real hardware. In this test it is ensured that the simulation matches the live behavior of individual modules. Section 5.1.2.1 will show this step.

- **Verification of the ASIC signals**
  Additionally tests visualizing all ASIC signals have been performed to ensure they match the interface specification of the ASIC. The timing of these signals depends on the FPGA design synthesis and additional hardware used and thus has to be verified in a live test. The tests are explained in 5.1.2.2. They also cover a maximum frequency estimation based on signal measurements within the test setup. For investigating analog effects the Agilent Technologies MSO6104A scope has been used.

- **Verification of the front end interface and PC communication**
  Another aim of the live verification was the verification of the communication between FPGA and PC and the front-end software as explained in 5.1.2.3
  This part could not be simulated, as the UART, the PC with its operating system and the front-end software naturally are out of scope for VHDL simulation tools. Therefore live testing is essential.

5.1.2.1 Modular live verification

As a first step a live integration test has been performed where every module was analyzed individually by routing out its signals of interest. In contrast to a live unit test this test technique has the benefit of not requiring an own test bench for each module.

All relevant situations can be tested using this setup.

The main objective of this tests is to ascertain that the synthesis step and simulation assumptions did not change the behavior of the modules.

Modular tests for the PC communication module, the ASIC controller and the ASIC simulation module as well as for their subcomponents have been made. As an example of a live test figure 5.3 depicts the signals of the 30 bit memory write controller during a readout sequence.

The tests have been successful and showed that the real module signals indeed match the simulation signals. From this it can be concluded that the FPGA logic behaves as expected.

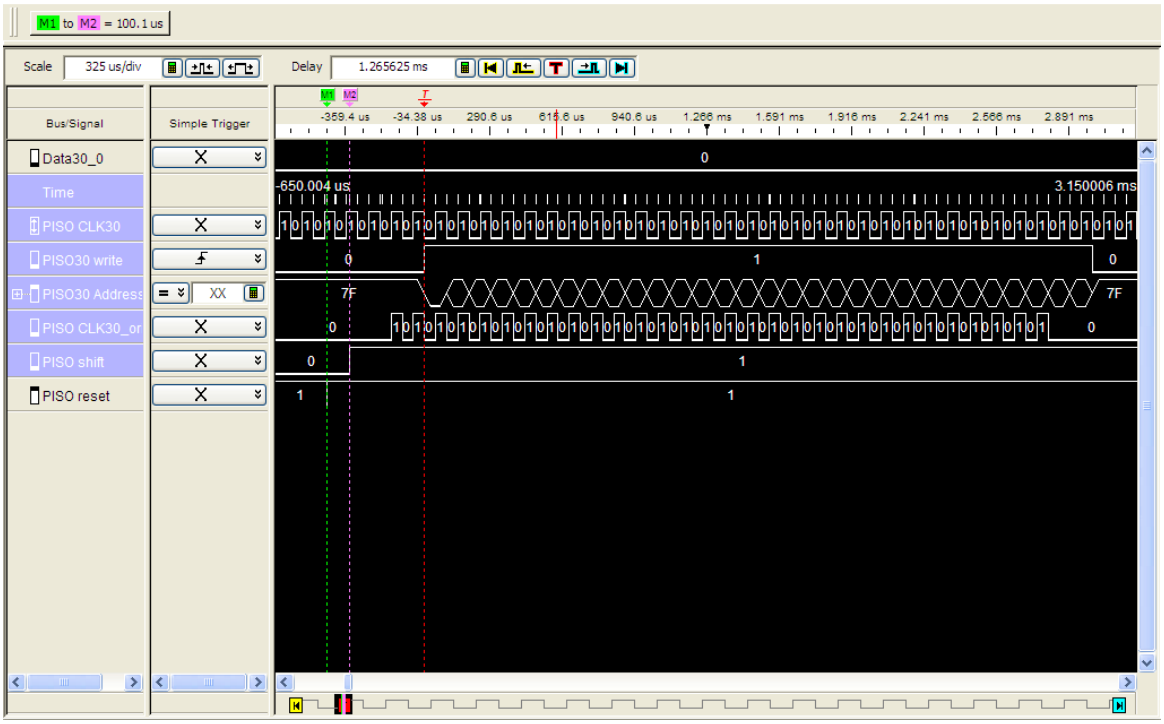

Figure 5.3: Module live test example.

5.1.2.2 Live verification of ASIC signals

One crucial test is the verification of the signals going to the ASIC board. They are a core feature of the FPGA design and therefore should be tested thoroughly.
The waveforms of the ASIC signals have been analysed using different measurement settings. The tests have been conducted with different target clocks (10kHz, 100kHz, 500kHz, 1MHz, 3.33MHz, 5MHz, 10MHz, 25MHz, 50MHz, 100MHz, 150MHz, 200MHz), different PISO clocks (10kHz, 100kHz, 500kHz, 1MHz, 3.33MHz, 5MHz, 10MHz, 25MHz, 50MHz, 100MHz, 150MHz, 200MHz), different measurement modes (normal, empty, full pipeline) and partly with different settings for SET0/SET1. Testing all combinations would have been too time consuming, so just samples including all important corner cases have been covered.

To achieve the selected frequencies, multiple versions of the design have been compiled with two different measurement-base-frequencies (100MHz / 200MHz), different scale factors for those frequencies, different PISO clocks and PISO scale factors. Using the front-end interface to adjust the measurement frequency these versions covered the desired range of test settings. The aim was to verify the correctness of the signals at low frequencies and then successively increase the frequency up to a point where the signals become too distorted to be recognized correctly. This way an upper bound for the

	Target clock			**PISO clock**		
ID	m_scale	$multiplier$ / $divisor$	f_{tar_clk} [MHz]	f_{PISO_base} [MHz]	$PISO\_scale$	f_{PISO_clk} [MHz]
1	1000	2/20	0.01	10	1000	0.01
2	1000	2/20	0.01	10	100	0.1
3	1000	2/20	0.01	10	1	10
4	1000	20/20	0.1	10	1000	0.01
5	100	2/20	0.1	10	20	0.5
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
40	1	4/2	100	200	1	200

Table 5.1: Overview of test cases for frequency boundary conditions.

operating frequency can be determined. The signal has been measured at the child-board pins to see the effects on the signal caused by the FPGA, FPGA board and child-boards. Also the influence of the cables on the signal quality has been investigated by measuring the same signals on the other end of the cable. By comparing the waveform before (figure 5.12) and after the cable (figure 5.13), the distortion caused by the ribbon cable becomes visible.

Table 5.1 is a compilation of boundary conditions which have been tested.

As explained in chapter 4.1.1, the base frequency of the measurement clock (*tar_clk*) always is 100MHz. All desired frequencies can be set by choosing the appropriate *measurement_scale_factor* and the dynamic PLL settings, as described in section 4.1.1. To calculate the actual measurement frequency, the base frequency is divided by the *m_scale* factor as well as the divisor of the dynamic PLL (*divisor*) and multiplied by the multiplier of the dynamic PLL (*multiplier*). So for the first test case the target clock would be calculated as follows.

$$f_{tar\_clk} = \frac{f_{base} * PLL\_multiplier}{measurement\_scale\_factor * PLL\_divisor} = \frac{100MHz * 2}{1000 * 20} = 10kHz \tag{5.1}$$

In contrast to the *tar_clk* clock, *PISO_clk* is not generated by a dynamic PLL, but only static components. Therefore the PISO base frequency (f_{PISO_base}) or its scale factor (*PISO_scale*) has to be adjusted for every change of f_{PISO_clk}. Changing any of the two parameters makes it necessary to resynthesize the design. The formula for the final PISO clock is the following:

$$f_{PISO\_clk} = \frac{f_{PISO\_base}}{PISO\_scale} \tag{5.2}$$

For the first test case this would be:

$$f_{PISO\_clk} = \frac{10MHz}{1000} = 10kHz \tag{5.3}$$

The signals have been measured directly at the child-board. Figures 5.4 and 5.5 show the waveforms of the measurement and readout signals during the first test case. According to the calculation both frequencies are 10kHz. This is the lowest frequency in these tests. The waveforms were as expected and match the simulation results.

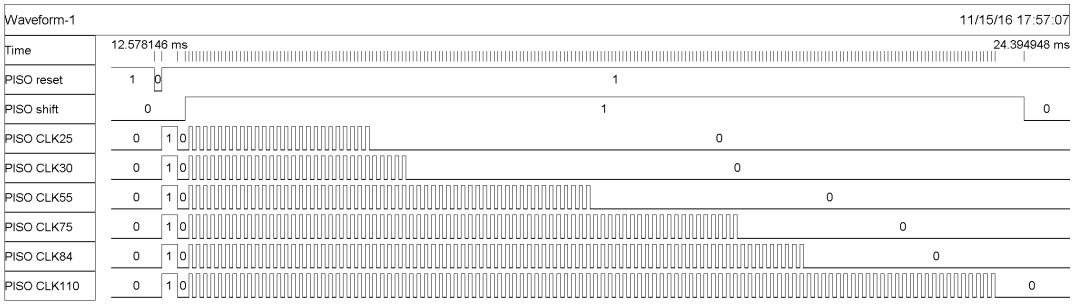

Figure 5.4: Waveform from the logic analyzer during the PISO readout at the first test case (10kHz).

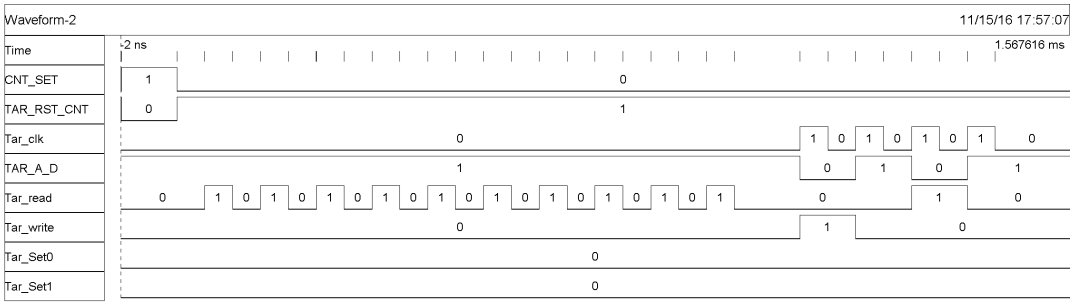

Figure 5.5: Logic analyzer waveform of the measurement signals at the first test case (10kHz). This is a normal mode measurement with four cycles and Tar_Set0 and Tar_Set1 set to low.

Also for test case 29 which has both frequencies set to 100MHz, the result was good as figures 5.7 and 5.8 show. One effect which can be observed in the images is that the duty cycle of the clocks seems to not be precisely 50% anymore. If the signal is observed with a scope, this effect cannot be seen anymore (figure 5.6). The signal looks very regular on the scope, therefore the previously seen effect seems to be caused by the logic analyzer

measurement setup. From the scope measurement it also is clear that the clock signal is not a square wave anymore.

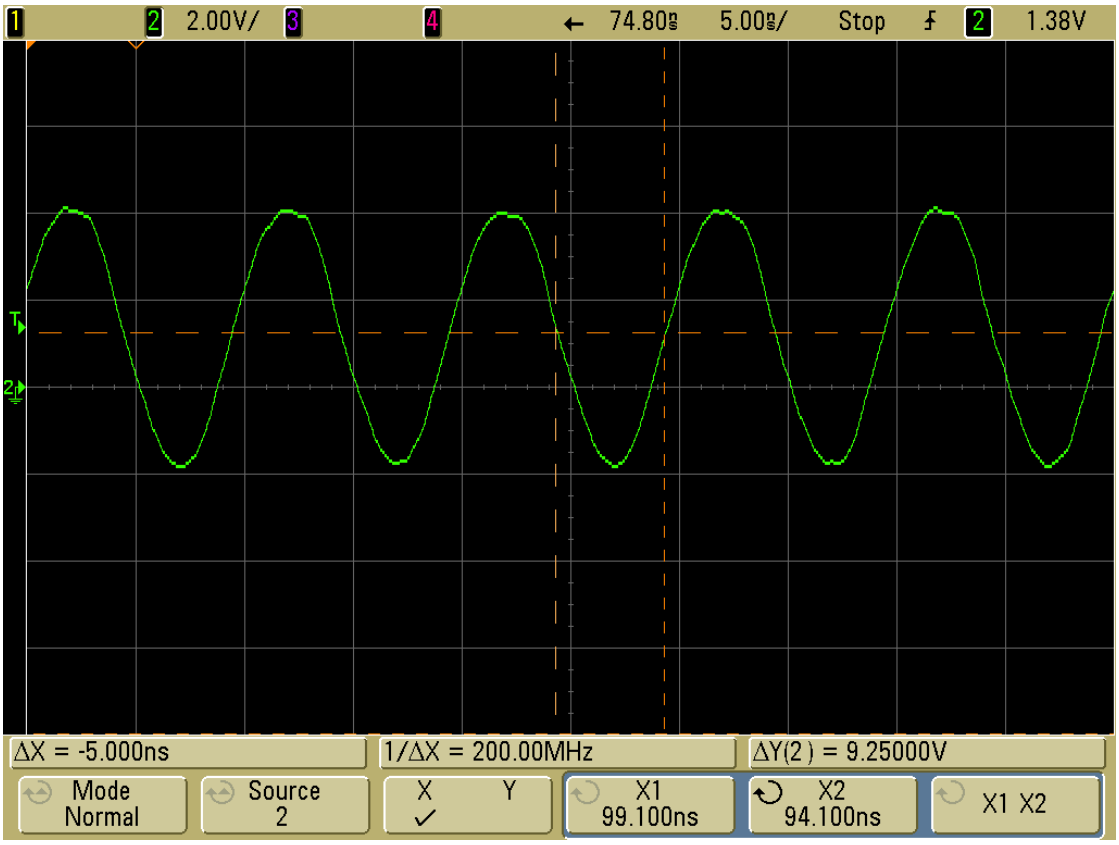

Figure 5.6: On the scope no asymmetric duty cycle can be seen. However, the clock is not a square wave anymore. The horizontal marker is set to 1.25V to indicate the logic level threshold of the logic analyzer.

Another finding at 100MHz was that problems occurred when using adjacent logic analyzer probes. This is shown in figure 5.9. In this case the neighboring clocks, which toggled simultaneously, influenced the nearby *PISO_shift* signal. This is however a measurement issue, not an FPGA or experiment setup flaw. Still it was unexpected as, in contrast to the child-board to ASIC board connection, the logic analyzer cables typically are less sensitive coaxial cables. As a consequence probes have been used which are further away from each other for tracing the PISO clocks at high frequency test cases. Further the clocks have been split up to multiple logic analyzer cables to be sure there are no crosstalk effects anymore.
It is expected that the connection between child-board and ASIC board is even more prone to crosstalk. This will be discussed in the following.

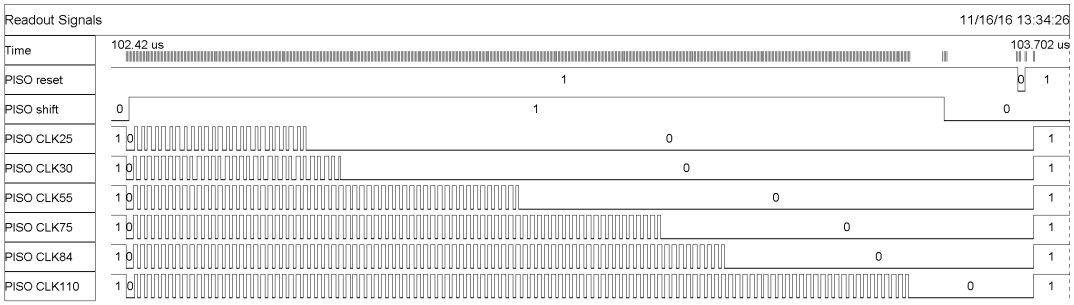

Figure 5.7: Waveform from the logic analyzer during the PISO readout at the test case 29 (TAR_CLK = 100MHz).

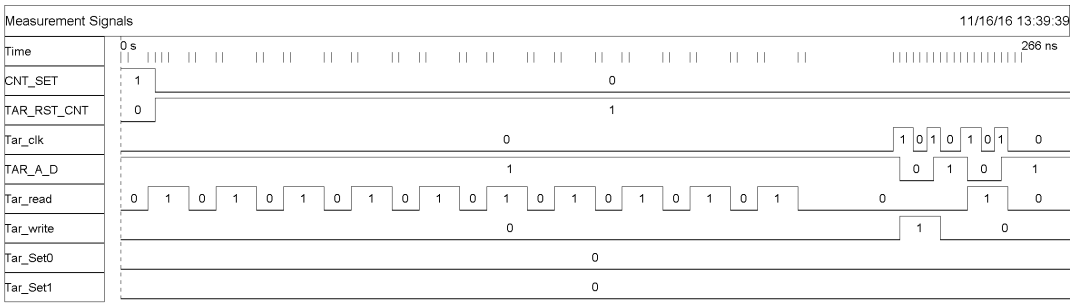

Figure 5.8: Logic analyzer waveform of a normal mode measurement with four cycles and Tar_Set0 and Tar_Set1 set to low using a TAR_CLK of 100MHz (test case 29).

Crosstalk Increased frequencies can lead to crosstalk problems between nearby signal lines. Especially high frequency signals are vulnerable to this phenomenon. The sensitivity of systems to such behavior can be investigated and detected with a scope long before it becomes an actual problem.
In the previous chapter our arrangement of the PISO clock signals already turned out to be indeed disadvantageous for high clock frequencies, since even the logic analyzer cables were problematic if adjacent probes have been used.
By design it has been tried to avoid crosstalk by having GND lines between each pair of signal lines. However, it is important to know up to which frequency this is sufficient. The type of connector and cabling are the most important factors when it comes to crosstalk and ribbon cables are not the best choice here. Unfortunately the child-boards limit us to use standard ribbon cables and standard connectors for the verification setup. This makes it more prone to crosstalk than coaxial cables or other high frequency solutions, but also less expensive and more comfortable to handle. Once the ASIC verification for lower frequencies is successful, also setups for higher frequencies can be tested.

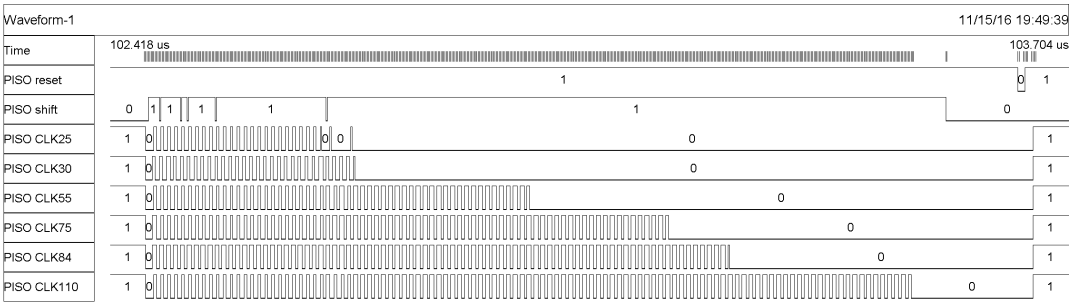

Figure 5.9: Waveform from the logic analyzer during the PISO readout at the test case 29 (TAR_CLK = 100MHz). As for this test neighboring logic analyzer probes have been used, they can influence other probe signals as the PISO_shift signal.

The signals of concern are especially the high frequency PISO clock signals at connector 2 of the PCB (figure 5.10) and the data signals at the child-board connectors. As can be seen in the schematics of the connector used for the PISO clocks, the odd pins (left) are connected to ground. The even pins (right) represent clock and control signals. While in the ribbon cable each pair of signals is separated by a grounded wire, this is not the case in the connector which has two pin rows. This means the clock signals are passing the connectors side by side and also are routed on the child-boards in this fashion. This is expected to be a source of crosstalk, additional to the long, parallel cables. As the clock can only be probed the at the cable connectors but not at the ASIC itself, the actual clock arriving at the ASIC can be assumed to be even more distorted. The clock distortion within the ASIC board can not be tested, however in chapter 5.2.4 crosstalk of neighboring data lines within the PCB will be discussed.
In the current chapter the investigation focuses the influence of crosstalk within the child-board, its connectors as well as within the ribbon cables.

A single high frequency clock is not a problem as the *tar_clk* signal proves. But since the six PISO clocks are operated synchronously at the same frequency, it seems the electrical fields of the clock signals add up to a point where they can influence each other or neighboring signals. For example in the situation where the PISO25 clock is disabled already but the other clocks are still active, this effect can produce spikes and thereby induce clock cycles in the PISO25 clock signal as will be seen in figure 5.12. This issue is not relevant in operation since additional clock cycles, after a PISO has been read out, do not affect the ASIC. However, as the clocks also can influence control signals such as *tar_shift*, crosstalk can become a problem. Therefore these issues will be addressed in detail in the following pages.

In the preceding pages it has been demonstrated that the signals come out of the child board very clean at frequencies much higher than 25MHz. As the logic analyzer gets connected to the other end of the matching ribbon cable, immense crosstalk can be

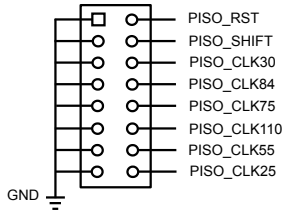

Figure 5.10: Schematic of the PISO clock connector. Pin one is marked by the rectangle. The clock signals pass the connector side by side.

seen. It can be concluded that the signal arrangement in combination with the ribbon cables is not a good choice. Figure 5.11 shows the logic analyzer result of this test. It is obvious that the *PISO_reset* and *PISO_shift* signals get distorted by the neighboring clocks in the first 30 cycles. As *PISO_CLK30* stops, they behave normally. Further *PISO_CLK25* and *PISO_CLK30* are influenced by the other clocks until *PISO_CLK55* stops. As stated above, additional clock transitions after the actual readout of a PISO do not disturb the functionality of the ASIC. The distortions of the control signals, however, are fatal.

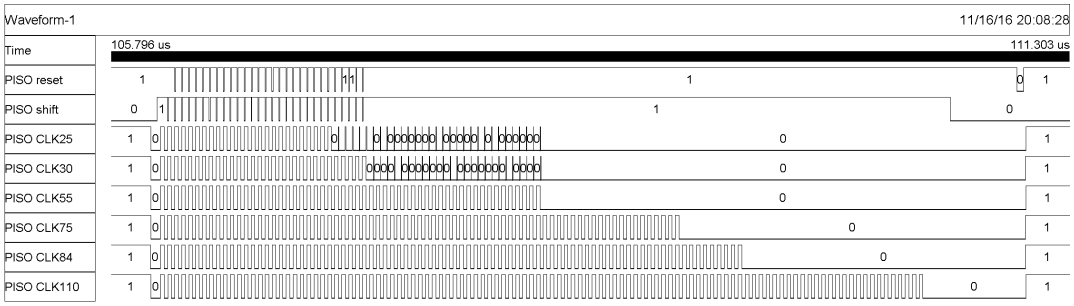

Figure 5.11: Waveform from the logic analyzer during the PISO readout at the test case 23 (*PISO_CLK*s = 25MHz).

As the logic analyzer readings are digital values and the corresponding threshold separating "high" from "low" is likely to differ from the threshold in the used CMOS technology, it is important to understand the signal in the analog domain. Therefore the following figures illustrate the same signal in the analog context.

Figure 5.12 is a screenshot from the scope illustrating the PISO25 clock as it is at the child board connector. The display has been shifted down by 1.25V to vertically center the image. The logic analyzer will detect signals above the x-axis line as "high" and below as "low". Note, however, that the logic analyzer cabling might have a different behavior at high frequencies and thus slightly change the results. It can be seen that

there is crosstalk in the PISO25 signal after it gets idle in the 26^{th} cycle which gets less every time another clock becomes inactive. However the maximum peaks recorded are very small with around 0.25V, while the threshold is at 1.25V. So there is a 1V margin to the threshold.

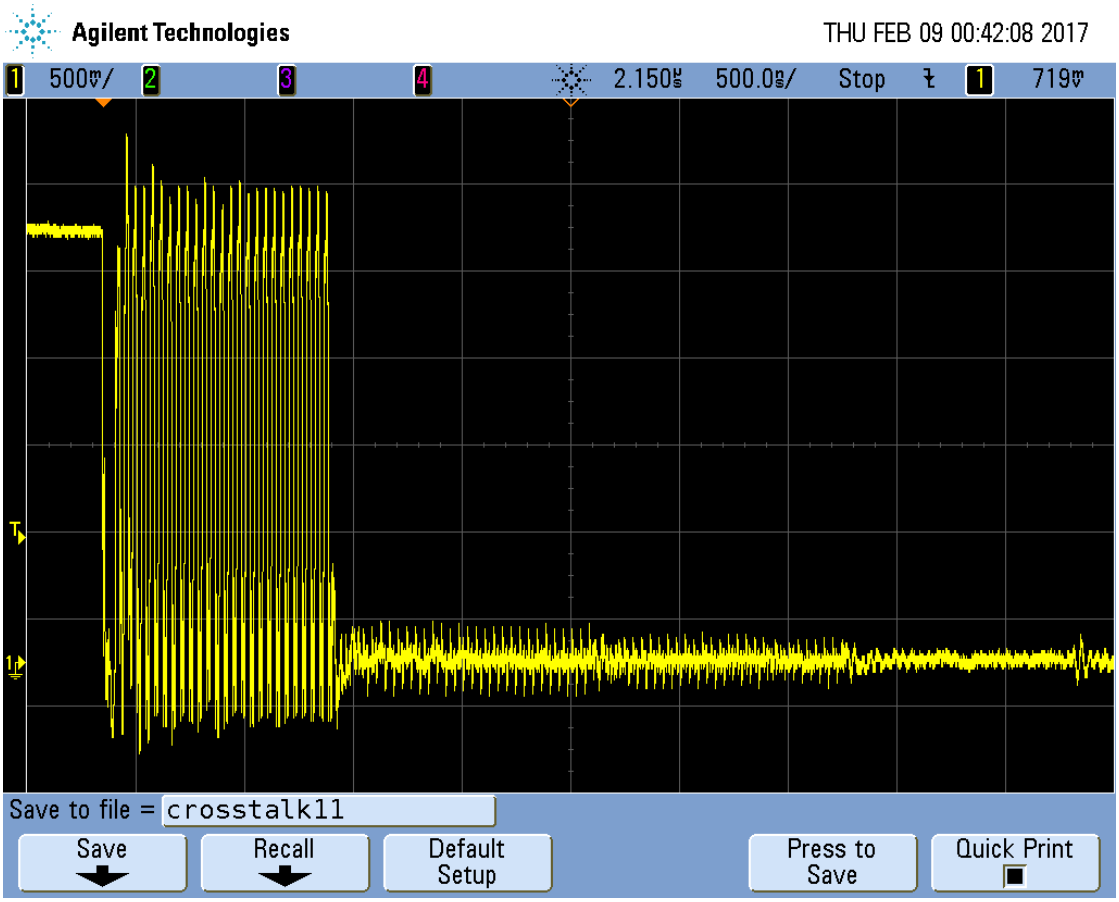

Figure 5.12: PISO_CLK25 signal at 25MHz at the child-board connector.

Figure 5.13 shows the same signal at the other end of the cable. It is obvious that the signal has been dramatically influenced by the neighboring clock lines. The maximum peak voltage recorded after the PISO25 clock got deactivated is about 0.75V and the margin to the logic analyzer threshold value is 0.5V now. This shows the influence of the crosstalk within the ribbon cables. Further crosstalk within the PCB can lead to even higher peaks and effectively change the logic level perceived by the ASIC. Therefore it is assumed that a PISO clock frequency of 25MHz is at the edge of the safe operating range for this setup. The real effective maximum frequency can only be determined by testing the whole setup.

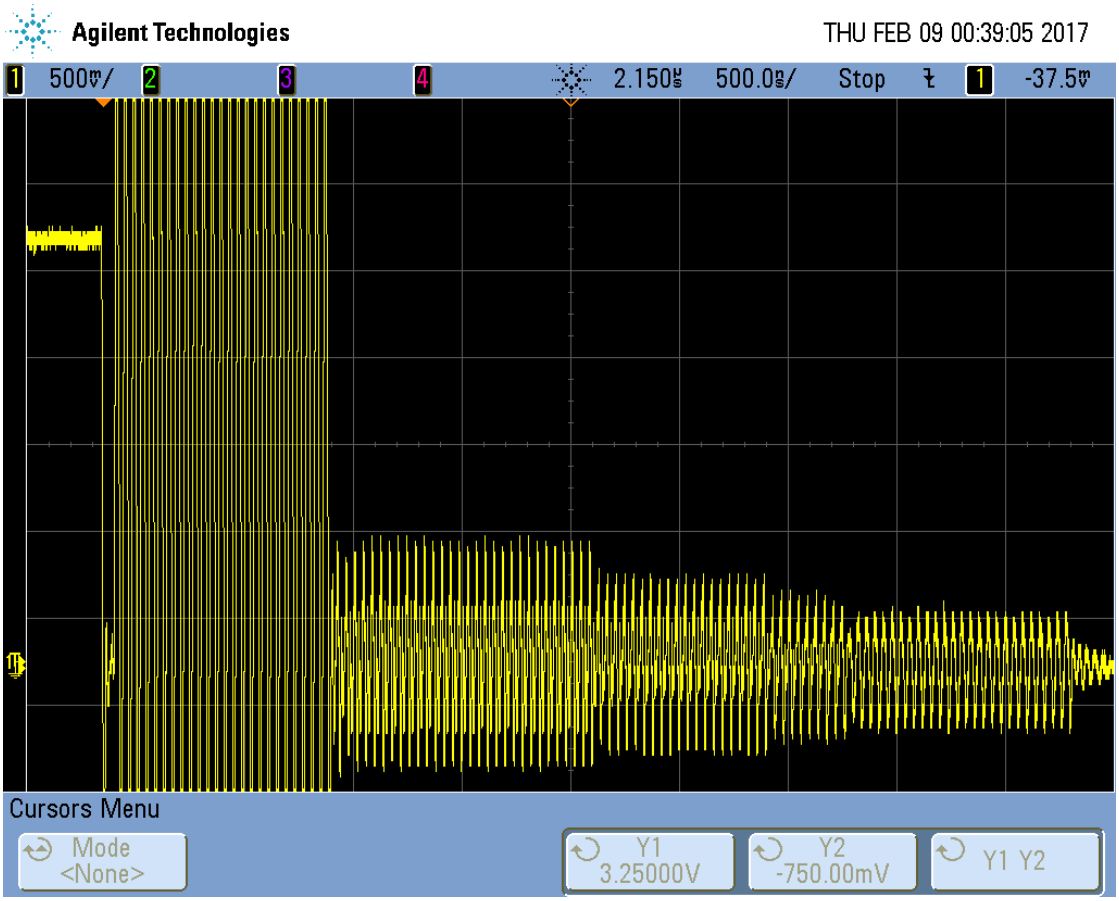

Figure 5.13: PISO_CLK25 signal at 25MHz at the end of the ribbon cable. It is obvious that the signal is under the influence of crosstalk and significantly distorted.

A simple fix for this problem is to distribute the clocks over different connectors to avoid their amplification. Also the issue could be solved by using multiple diverse PISO clock frequencies or shifting selected clocks by 180° to mutually cancel out the current crosstalk effects. However the first option is the easiest to implement and use.

Another phenomenon observed are the overshooting signals in the first 25 cycles. The highest peaks seen at 25MHz reach 4V. This has to be considered when choosing the operating frequency as too high voltages can harm the chip.

Another crosstalk effect was seen on the neighboring *PISO_SHIFT* signal as shown in figure 5.14. The different stages of the data readout phase are clearly visible as the clock signals extremely distort the shift signal. This can lead to the shift signal being seen as sporadically toggling by the ASIC as shown before in figure 5.11.

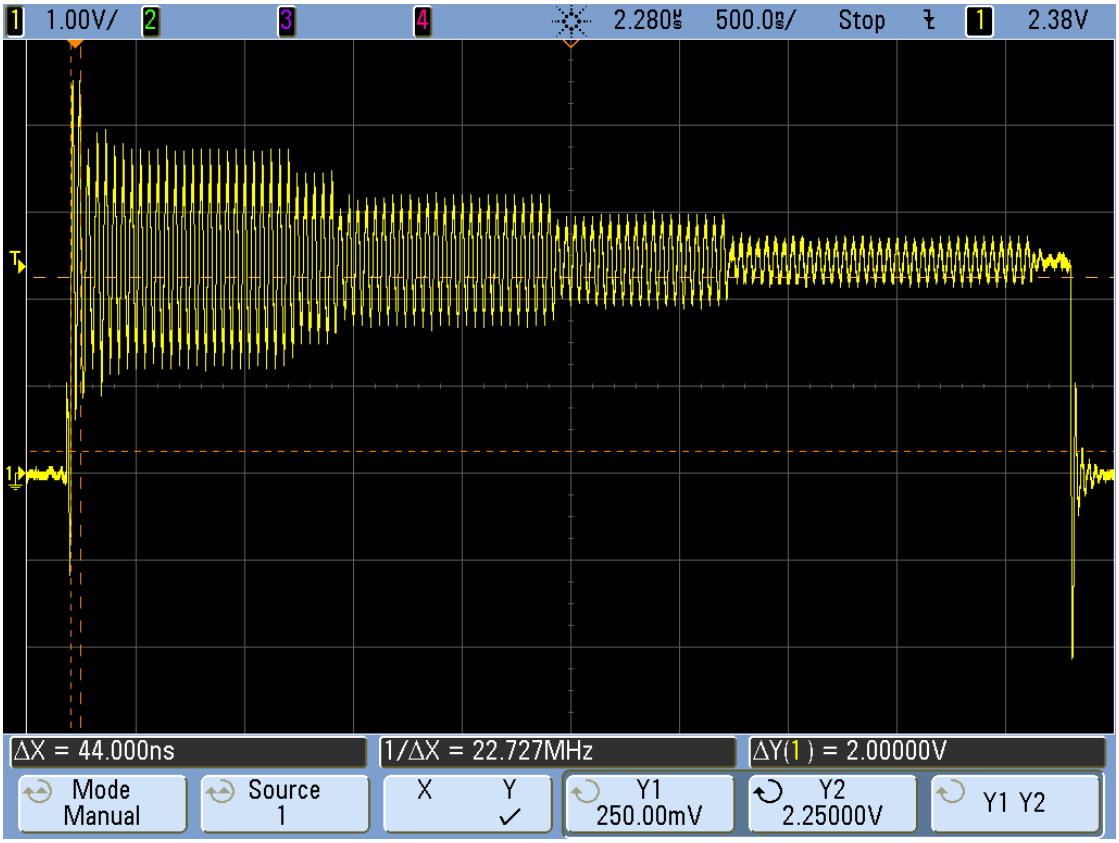

Figure 5.14: Shift signal under the influence of crosstalk at neighboring clocks at 25MHz. It is obvious that the signal is significantly distorted.

5.1.2.3 Live verification of front-end interface

As the FPGA - ASIC interface now has been tested, the communication with the PC had to be analyzed as well. As this is a high level test, no scope or logic analyzer is needed. Instead the received data has been analyzed on the PC using the evaluation tool introduced earlier. The ASIC test module responds with a certain data pattern, which is known and can be easily adapted in the VHDL code. With the help of the analyzer tool, it can be seen whether the received data matches the original pattern sent by the ASIC test module. If this is the case for all tested patterns, the data transmission from the ASIC to the PC also will work on a logic level. It is however only possible to simulate the expected logical behavior of the ASIC and of course not its microelectronic specifics. Therefore the real ASIC is needed.

For all tests the data patterns matched the expected pattern, which shows the data flow through the framework is correct.

5.2 Methods for ASIC board Verification

After it has been ensured that the framework works properly, also the PCB and ASIC, once available, have to be tested whether they are working as expected. However the PCB was optimized for high frequencies and not for debugging. Thus not many means for testing the board itself and its electrical properties existed. Still it allowed to perform some simple tests. The routing of the board has been visually checked as well as the ASIC wire bonding. Further an analysis of the conducting paths from the ASIC to the PCB connectors has been made. Therefore the framework has been prepared for an actual test measurement and the internal test module of the FPGA has been disconnected. Instead the control signals have been routed out to the child board connectors and the data signals coming from the child board connectors substituted the data signals from the internal test module. This way the PC receives the real ASIC data instead of the one generated by the FPGA.
In the following the steps taken to test the ASIC boards will be explained.

5.2.1 Visual Bonding Check

For a visual check of the wire bond interconnections a microscope was necessary, since the bond pad pitch of the ASIC is 80µm and the pads thus clearly are not visible with bare eye. This small pad pitch is the primary difficulty the microelectronics department faced when bonding the ASIC to the PCB. Another issue was the small bond pad size on the ASIC, which is roughly 40µm x 80µm. For ball bonding this requires a smaller wire than was available at the department in Brno for the first bond tests. For wedge bonding on the other hand the alignment of ASIC and PCB pads to each other turned out not to be optimal and the proper tool for handling 17.5µm wires was not available. Further the surface of the bond pads on the PCB was not optimal. Also there was the risk of creating a short circuit to the conducting paths passing the ASIC pads in very close distance.
These factors made long testing and adapting of the wire bonding parameters necessary until a satisfying result has been achieved.

So there were good reasons to visually check the bonding wires for typical bonding faults such as cracked heels, misplaced wires, open circuits or short circuits between the pads or the circuit area [15].

At the visual checks it has been differentiated between the important input signals, other input signals and output signals.
The signals from 1 to 22 in figure 5.15 are input signals. Thus most of those bondings are essential for the setup to work. (Note that VSS and VDD are not considered as signals as they are power supply pins.)
The output signals are at pins 23 to 84. Bonding problems with the output signals would result in unreliable or lost data for individual PISOs, but not effect the overall functionality of the ASIC. In contrast, bonding problems at important input signals, such as the *PISO_RST* would corrupt the entire chip.

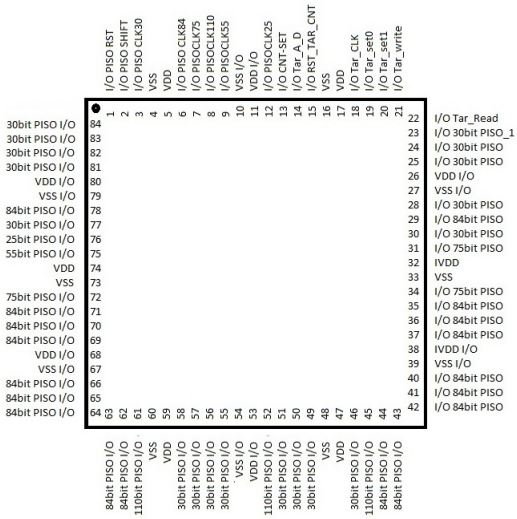

Figure 5.15: Schematic of the ASIC pins

5.2.2 Visual Routing Validation

The next step of testing the board was to check the routing of the signal lines on the PCB. Although the routing on the PCB design has been cross-checked with our colleagues from Brno multiple times during the development stage, it had to be ensured the final board routes the signals out on the same connector pins as the FPGA expects them to be. This test can be performed visually, with or without microscope as the conductive paths on the PCB are still visible without microscope.

5.2.3 Test of the voltage regulators

The output voltages of the 1.25 V and 2.5 V voltage regulators are important, since a too low voltage could influence the functionality of the chip and an overvoltage could damage it. The PCB has been connected to a 5 V power supply with a 300 mA current limit. The current limit is needed to protect the ASIC from high currents in case there is a problem with the AISC or the board. To ensure the voltage regulators are working properly, we changed the primary input voltage of the regulators and measured the voltage output of both regulators over different voltages. The results for each PCB are then listed in the corresponding results section.

5.2.4 Analysis of the conductive paths

In the previous steps it has been ensured that the ASIC has the correct voltage and is connected properly to the PCB. Another interesting aspect to test is the conductivity of the circuit paths. Unfortunately there are not any means of testing the circuit paths on

the PCB themselves once the ASIC is bonded. To save die area, boundary scan, as well as other test support features, have not been foreseen in the FRad ASIC design.

Therefore it has to be relied on detecting problems with faulty circuit traces at the ASIC testing step, because this kind of faults would have the same appearance as bonding problems. As a result the two faults cannot be easily distinguished.

One method to check certain electrical properties of the conducting paths between ASIC and PCB connectors is to analyze the data signals coming from the ASIC arriving at the PCB connectors using a scope. For this task the whole framework had to be assembled and a measurement run performed under normal conditions. This way not only the data read out could be seen from the PISOs, but also the real voltage level of the signals over time at the FPGA and their rise/fall times.
This information is important to estimate the overall loss of signal quality at different frequencies and to check for unexpected effects. The following issues have been analyzed:

- **Short-circuit faults:**
  Short-circuits of the data signal traces or bonding can be detected by analyzing the data signals with the scope. If the data signal of a PISO goes low for at least one readout cycle, it can be concluded there is no short-circuit between this signal and a VDD line. Similarly a short circuit to GND can be ruled out if the signal goes high once.
  Short circuits between data lines can be investigated as well, although only if their data patterns differ. Data signals from the same kind of PISO will have the same waveform and thus a short circuit between them will remain undetected. Using a multimeter the resistance between the pins can be measured as an alternative.

- **Voltage level:**
  The voltage level should be at 2.5 V for an active signal. This indicates that the power supply of the I/O area in the ASIC is working properly or otherwise indicates a conductivity problem with the connectors, PCB, bonding or the PISOs in the ASIC itself.

- **Switching characteristic:**
  Also with frequencies increased to a certain level the voltage level and rise/fall times of the data signals should still be in a range where the the FPGA can easily handle the data stream. The rise/fall times should be as short as possible and the edges should not contain glitches. If the clock period becomes shorter than the rise/fall times, the read out will not work anymore.

- **Crosstalk:**
  It has been shown in 5.1.2.2 that neighboring signals strongly influence each other in the ribbon cables. Similarly to crosstalk in ribbon cables, also neighboring PCB traces can influence each other.

This section completes the crosstalk analysis by investigating the vulnerability of the ASIC and its PCB to this effect.

This can be done by looking at the data signals coming out of the PCB. However not all signals are equally prone to this effect and not all signals can cause it. For instance a signal surrounded by data lines from different PISO types is less likely to be affected than a signal surrounded by equal PISO types. One reason for the detected crosstalk is that multiple signals change their state simultaneously. Signals from the same PISO type therefore are more likely to change their state simultaneously and cause crosstalk. Further transitions in a signal are required to produce crosstalk. Signals from a 110 bit PISO can not cause crosstalk in normal operation as its data bits are zero all the time.

5.3 Methods for ASIC Verification

This chapter finally discusses the methods used to analyze the ASICs themselves. The subsequent chapters will refer to those methods as a guideline for verification.

Before verifying the ASIC all tests for the board verification must have been conducted successfully. Otherwise the ASIC verification results can be influenced by PCB faults.

To test the ASIC the board has been connected to the FPGA and its power supply. The FPGA has been loaded with the controller design and the front end software is set up.

5.3.1 Test runs of measurement

One test is the normal mode operation test with different numbers of measurement cycles. The PISOs 25, 55, 75 and 110 contain all zeros if there was no SET in the corresponding target circuits and counters. So the correct behavior of those PISOs or the behavior of their connected target circuits or counters cannot be verified without stimulating the ASIC with an external source such as a micro-beam. However, it can be verified whether all PISOs show zeros when expected. Using this technique it can be checked for stuck at 1 faults in the PISO outputs.

As a consequence the 30 bit and 84 bit PISOs are in the focus of the analysis. Without a micro-beam facility the analysis of the 30 bit and 84 bit PISOs still can give us important feedback to the ASIC design.

As the 30 bit PISOs are connected to three up-down counters, they show a well defined bit pattern in case there was no SET. This makes it easy for us to verify its correct operation by just reading out its initialized value. If the expected pattern is read, the PISO is working. This pattern changes only when there is an SET, so as before the connected target circuits and counters themselves can not be verified without a micro-beam source. Also the PISOs can be tested for only one certain pattern what limits test coverage.

For 84 bit PISOs many values after initialization and after each cycle are different and can be checked what makes it possible to test the second most sophisticated PISO of the

design to a big extent, even without the use of a micro-beam facility. Additionally the corresponding 32 bit LFSR counters are tested at the same time. Since every kind of target circuit is connected to at least one LFSR counter, also the target circuits can be verified by toggling the *TAR_CLK* signal and analyzing the LFSR counter values.

- **0 cycle measurement** The tests begin with a normal measurement with 0 cycles. So the targets get initialized and immediately read out. There is no activity at the target clock. This test shows whether the targets get reset and read out properly. The data values being read out should correspond to the initialization values. If this test succeeds, the behavior at *TAR_CLK* activity can be tested.
- **1 cycle measurement** Like in the "0 cycle measurement" the target circuits get initialized. However, subsequently the *TAR_CLK* signal gets toggled once. This initiates a transition in the LFSRs. At the readout step the LFSR value gets stored in the 84bit PISOs and sent to the PC. The value then can be compared to the expected LFSR value. In case of a match it is shown that the 84bit PISOs work correctly and that the changed flip-flops within the LFSRs do not suffer from a stuck-at fault. This measurement does not give information about the unchanged flip-flops of the LFSRs yet.
- **n cycle measurement** This test is the same as the "1 cycle measurement", however the number of *TAR_CLK* cycles is increased with every measurement run and as a result other flip-flops in the LFSRs are toggled in every run. The tests will be repeated in this fashion with an increasing number of cycles until all flip-flops have changed their values at least once. If all LFSR positions have been high and low once, stuck at faults in the LFSR can be ruled out.

5.3.2 Tests without the FPGA

To eliminate the FPGA and the front end software as a source of error, another test setup exists only using a logic analyzer together with its internal pattern generator to generate the stimuli for the ASIC. In case of unexpected measurement data this setup can be used to repeat certain measurements to narrow down the source of the unexpected data to either the ASIC or the framework. The advantage of this solution is it can be adapted much quicker and without side effects than changing the FPGA design. Changing the FPGA design makes recompiling necessary and usually requires simulations to verify the behavior is as expected. Changing the logic analyzer tests is very easy and no simulations are required. The tests can run immediately.

All tests mentioned in the previous section also can be run directly on the logic analyzer.

Reset while readout test

One additional test that can be performed using this setup is to keep the target circuits reset while reading them out. This way it is guaranteed the initial counter values and

not some corrupted values are read into the PISOs. In case the measurement results of this test are not as expected, the counter initialisation or the PISOs are faulty.

5.4 Verification results of the first ASIC board

Bonding an ASIC with as small pads as ours was a premier for our partners in Brno. Thus it was not expected to gain perfect results for the first ASIC, but to gather experience and know-how to achieve satisfying results with the next ASIC samples. As the first bonding attempt was complete they showed us the result in the Microelectronics Department Brno. The visual bonding check revealed multiple problems with the bonding such as short circuits between ASIC pads, short circuits between the bond pads and the surrounding chip area, non-sticking bonds and random shorts caused by broken bond wires. Figure 5.16 gives an impression of the result.

The top left pin is pin 1 where bonding started. The pins have been bonded clockwise.

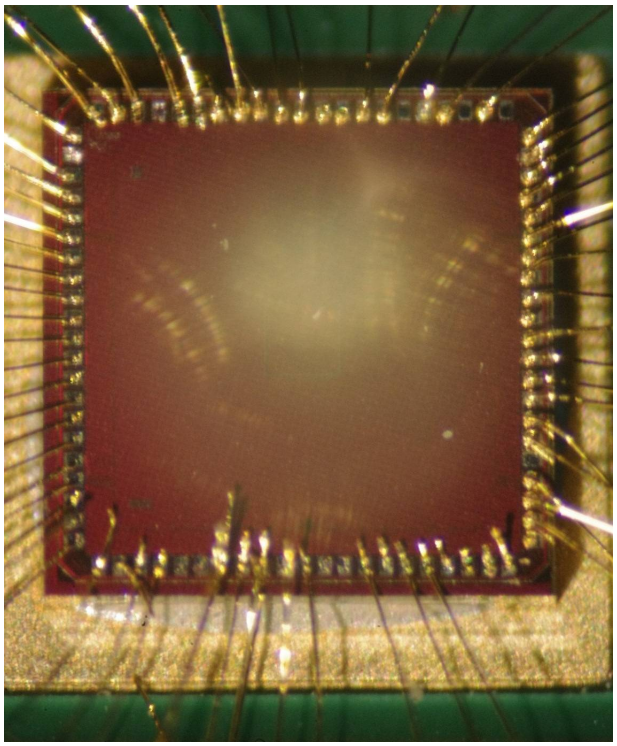

Figure 5.16: Overview of the first bonding results. The wrong equipment caused the bondings to be unreliable.

The first pins have been ball bonded with a 25µm wire. The wire turned out to be too thick for the small bond pad pitch, so it has been exchanged for a 17.5µm wire. The image shows that for both wire types the ball bond is exceeding the pad and for some

pairs of pads the ball bonds seem to cause a short circuit. The pads on the bottom of the ASIC have been pre heated before bonding. This showed some improvement, although the success rate still was too low. The left pads of the ASIC have been wedge bonded using the 17.5µm wire. As expected the wedge bonds are much smaller and stay within the pads. However the accuracy and reliability has not been sufficient yet. The main reason for the unsatisfying bonds was that the available equipment was suitable for 25µm, but not 17.5µm bonding wires. Also the small bond pad pitch and the difference of the bond pad pitches of the PCB and the ASIC made bonding difficult. For the following ASICs a suitable tool has been ordered. The microscope image reveals that multiple input signals could not be properly connected. Among them was the PISO_RST signal, which is crucial for initializing the chip. As a result it was clear the ASIC sample could not be used and bonding quality had to be improved by changing the equipment. However, as most wedge bonds of this first attempt looked fine, we have been confident that further attempts would succeed.

As the ASIC could not be used for experiments anymore, this ASIC has not been tested.

5.5 Verification results of the second ASIC board

As the bonding parameters had been optimized, the second ASIC was bonded to a PCB. The bonding results looked promising so the board was brought to the Technical University Vienna for testing.

5.5.1 Results of the visual bonding check

The visual tests from chapter 5.2.1 have been performed on the second board.

Our partners at the microelectronics department in Brno already performed a visual check before the transport of the PCB and sent us their images. Those pictures got analyzed again to ensure there are no significant visual bonding problems.
No further visual bonding check has been performed once the ASIC arrived in Vienna, as it was considered unlikely that the bonding breaks during transport. Also short or open circuits caused by the bonding procedure were expected to be detected during the live tests.

The figures show the bonding of the PCB pads (5.17) as well as of the ASIC pads (5.18). Note that some of the bonding wires provide the core and I/O voltages for the ASIC and do not transmit signals. Since the power lines are redundant, a weak bond is not critical, although there must not be a short circuit to a neighboring pad.
Regarding the PCB pads the wire bond interconnections look well here.

The second figure shows the interconnections between bond wire and the ASIC pads. It is obvious that changing the tool from $25\mu m$ (figure 5.16) to $17.5\mu m$ brought a huge improvement in the bonding quality. Only the few interconnections in table 5.2 seem

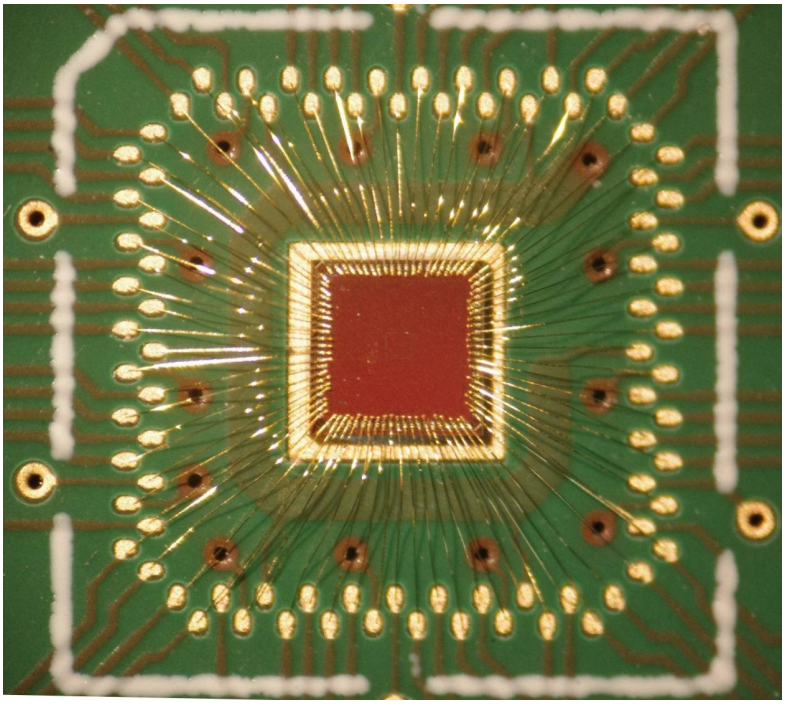

Figure 5.17: Overview of the bonding of the second sample, in particular the PCB pads are nicely visible here.

irregularly placed, although not alarming. The results have been kept in mind when the next tests were performed.

ASIC Pin number	name	potential hazard
PIN1	*PISO_RST*	malfunction
PIN15	*RST_TAR_CNT*	malfunction
PIN26	*VDD I/O*	none
PIN36	*84bit PISO*	loss of two PISOs
PIN48	*VSS*	none

Table 5.2: Bonding problems with the first ASIC sample.

5.5.2 Results of visual routing verification

As a first test the conducting paths of the PCB have been checked without microscope and it has been verified that the pinout is correct. Also no problems with the traces have been detected.

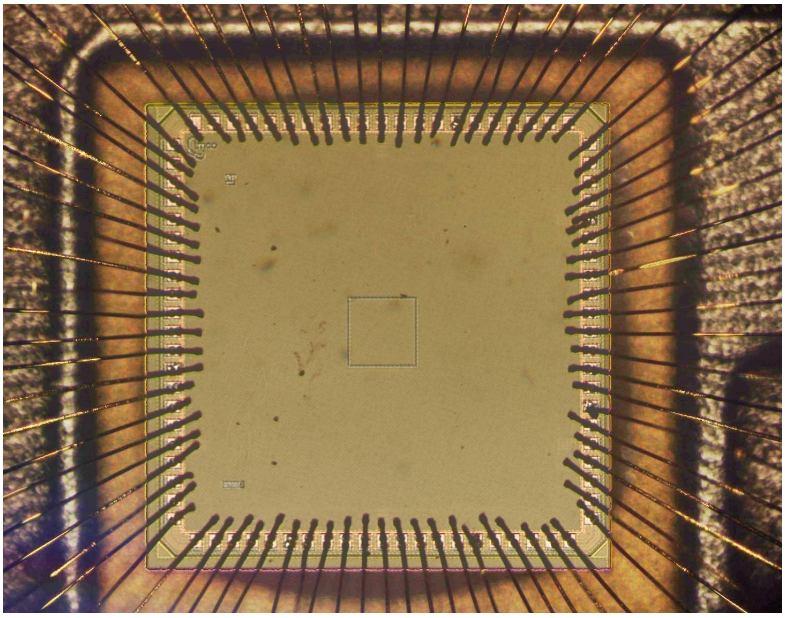

Figure 5.18: Overview of the ASIC pad bonding interconnections on the second sample.

5.5.3 Results of the conductive paths analysis of the second board

As explained earlier certain signal aspects had to be checked before the actual ASIC could be analyzed. For these conductivity tests a measurement has been started on the framework. This way the electrical signals coming from the ASIC can be analyzed.

To ensure the voltage regulators are working properly, the output voltages have been measured for 5V, 6V and 7V input voltage. They delivered the expected 2.5V and 1.25V levels for all of the tested input voltages. Also all active data signals coming from the ASIC reached the 2.5V. This indicates that the output pads are powered properly and there are no conduction problems between the ASIC and the PCB connectors. Figure 5.19 illustrates a rising edge of a 1 MHz data signal at the ASIC pin 66. The horizontal markers are set at 10% and 90% of VCC. The rise time between those potentials has been measured as approximately 7ns. As the signal had both values, high and low, it can be reasoned there is no stuck at fault for this data signal. This is the case for all 30 bit and 84 bit PISOs. As will be seen later on in figure 5.22 the 84 bit PISOs did not behave as expected. From their waveforms, however, it can be concluded there are no shorts between any 84 bit data signals. Especially the waveforms of pins 36 (Data84[15]) and 37 (Data84[14]) which showed a possible short in the microscope image, have been checked with special care. As there was no correlation between the two signals, the bonding and traces of the concerned signals can be considered fine.

Also with increased frequencies the electrical properties of the conducting paths proved to match the needs for flawless communication.

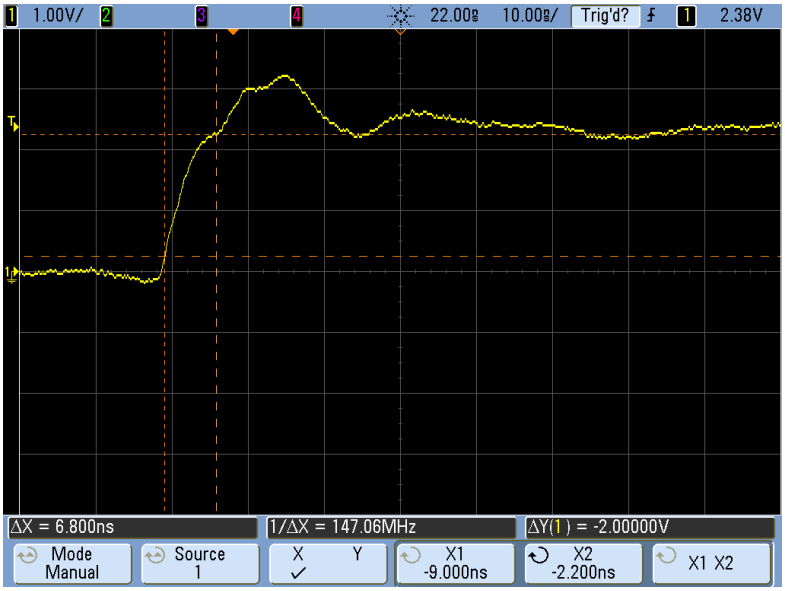

Figure 5.19: Rise time measurement of a 1 MHz data signal.

Crosstalk Pin 52 of the ASIC, a 110 bit PISO data signal, has been chosen as a test signal, since it is surrounded by five 30 bit data signals, that toggle all at the same time and are likely to cause the biggest crosstalk within the setup. Pin 51 of the ASIC has been chosen as a reference for illustrating the 30 bit signals. Figure 5.20 shows the stimulating signal (30 bit data - green) and the affected signal (110 bit data - yellow). It can be observed that the signal from the 30 bit signals gets partly transferred to the 110 bit signal and the maximum peak even was around 1V. The distortion, however, is not big enough to cause level changes.

As discussed earlier the cable has a big influence on the overall signal quality. Figure 5.21 depicts the corresponding signal on the other end of the ribbon cable. It can be seen that the maximum of the spikes of the 110 bit signal is around 250mV and therefore might be temporarily recognized as a level change by the FPGA. Nonetheless the data signals get sampled by the FPGA after half a clock cycle, at which time the signals should have stabilized already.

It can be concluded, however, that the data of this PISO might be distorted at higher frequencies as a result of crosstalk.

Overall no problems with the board could be detected, as long as the readout frequencies are chosen with care.

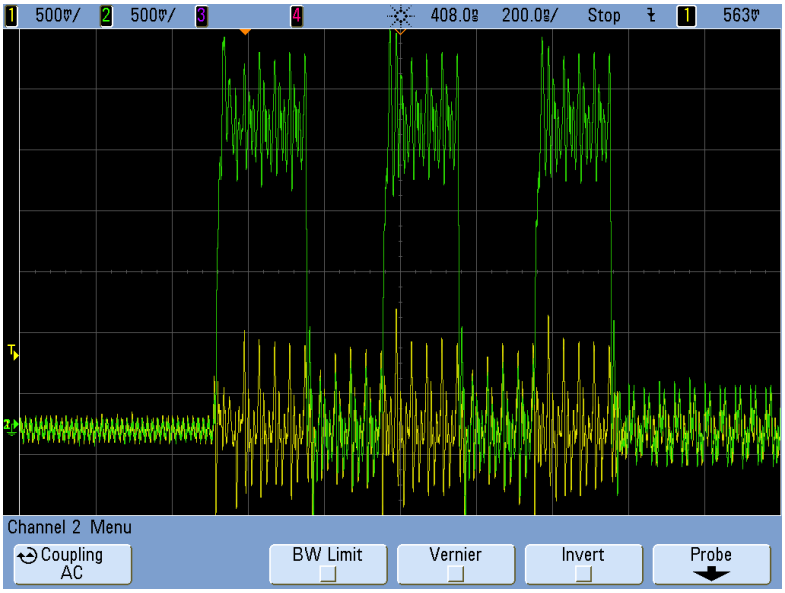

Figure 5.20: Crosstalk measurement of data signals at 25MHz directly at the PCB (green: PISO30[9], yellow: PISO110[1])

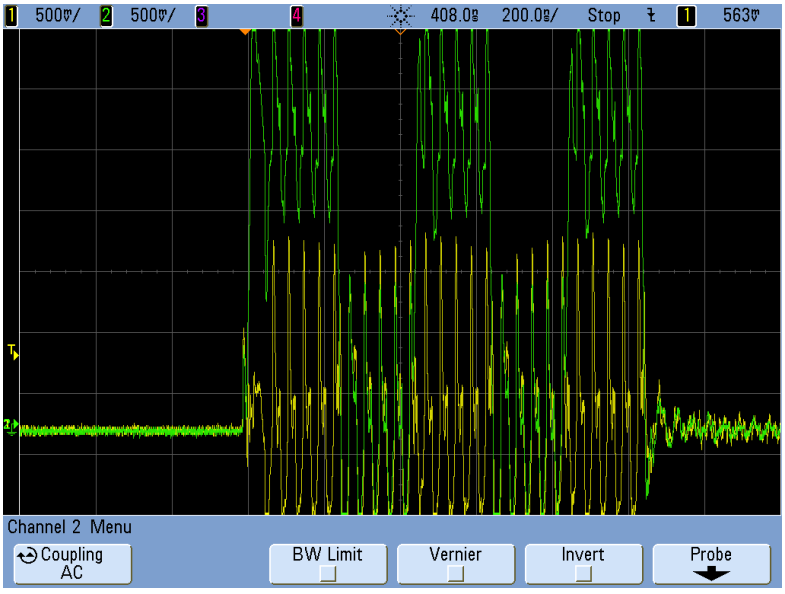

Figure 5.21: Crosstalk measurement of data signals at 25MHz after the ribbon cable (green: PISO30[9], yellow: PISO110[1])

5.6 Verification of the second ASIC

As no problems with the PCB could be discovered, the ASIC tests have been started. The first test was to simply initialize the ASIC and read out its data without any activity on *TAR_CLK*. Afterwards also tests with activity on *TAR_CLK* have been performed.

0 cycle measurement While for the 25, 55, 75 and 110 bit PISOs, which have been expected to be zero all the time, the values were correct, there have been some problems with the 30 bit and 84 bit PISOs.

Counter name	Counter size	Initialization data
Up-down counter 1	10 bits	1111100000
Up-down counter 2	10 bits	1111100000
Up-down counter 3	10 bits	1111100000

Table 5.3: The data structure of the 30 bit PISOs.

The data of a 30 bit PISO consists of the combined counter values of three up-down counters, as shown in table 5.3. This means the first ten bits represent the counter value of the first counter. The next ten bits correspond to the second counter and so on. Each individual up-down counter is initialized with the values "1111100000". So the expected pattern for the 30 bit PISOs was "111110000011111000001111100000". Most 30 bit PISOs showed the expected results, except for PISO30_7 which always showed the pattern "1111100001" for the second up down counter. The up-down counters use a unary code, meaning that the increment operation replaces the leftmost zero with a one and the decrement operation replaces the rightmost one with a zero. As a consequence the zeros and ones have to be grouped together. A pattern like "1111100001" is impossible in this type of counter. A possible cause could be a problem in the PISO, a problem in the counter itself causing this kind of behavior seems unlikely.

Counter name	Counter size	Initialization data
LFSR 1	32 bits	00000 ... 00000
Up-down counter 1	10 bits	1111100000
Up-down counter 2	10 bits	1111100000
LFSR 2	32 bits	00000 ... 00000

Table 5.4: The data structure of the 84 bit PISOs.

The data of the 84 bit PISOs starts with a 32 bit LFSR followed by two up-down counters with 10 bits each and again a 32 bit LFSR. The data after initialization is illustrated in table 5.4.

In the tests none of the 84 bit PISOs matched the expected pattern. Also the results of the 84 bit PISOs have been more diverse than those of the 30 bit PISOs.

To be sure the problem is within the ASIC and not the framework, the logic analyzer has been used to confirm the findings.

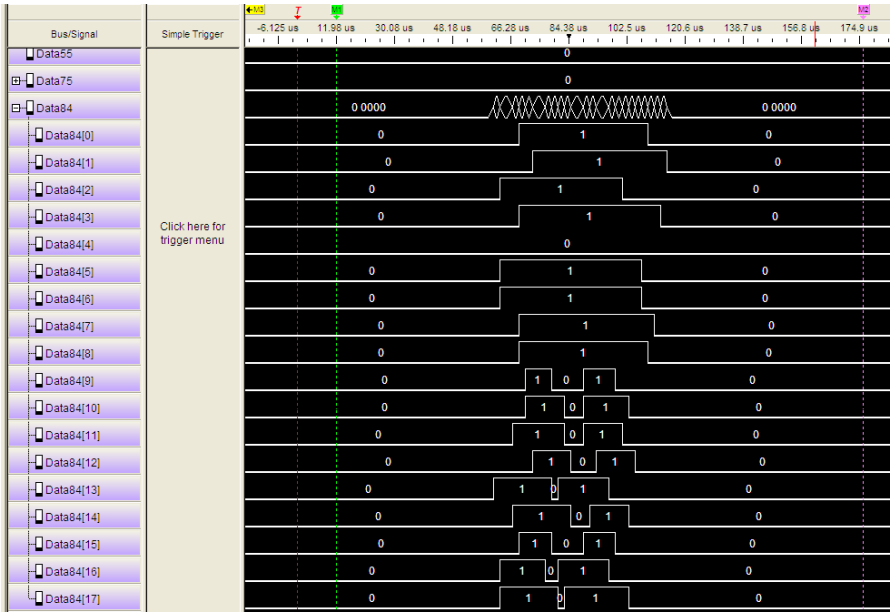

Figure 5.22: Data signals of 84 bit PISOs read out once with a PISO_CLK of 500kHz.

0 cycle measurement with logic analyzer Figure 5.22 shows the 84 bit PISO data as seen by the logic analyzer. It is obvious that all PISOs show different values. Further none of them shows the expected data pattern. Some output a pattern, like Data84[15], looking similar to the expected pattern, as they contain the expected sequence of the up-down counters: "11111000001111100000", however compared to the expected pattern they are shifted by a couple of zeros. The PISOs should contain 32 zeros before and after the values of the up-down counters. This constraint is violated by all PISOs.

A closer look (5.23) revealed that not even the three consecutive readouts of the PISOs have been identical. This indicates a problem with the PISOs themselves. Repeating this test showed that sometimes all three readouts are identical, sometimes not. If they are not, randomly the first, second or third transmission can contain the inconsistency. Since the PISO values can not be read reliably, it is difficult to analyze the underlying counters. However it could be shown that the FPGA framework is not the source of the error.

1 cycle measurement with logic analyzer Although the result of the previous test suggests all readouts of at least the 84 bit PISOs are corrupted, and hence unreliable, it has been tried to gather as much information as possible from the chip. For this

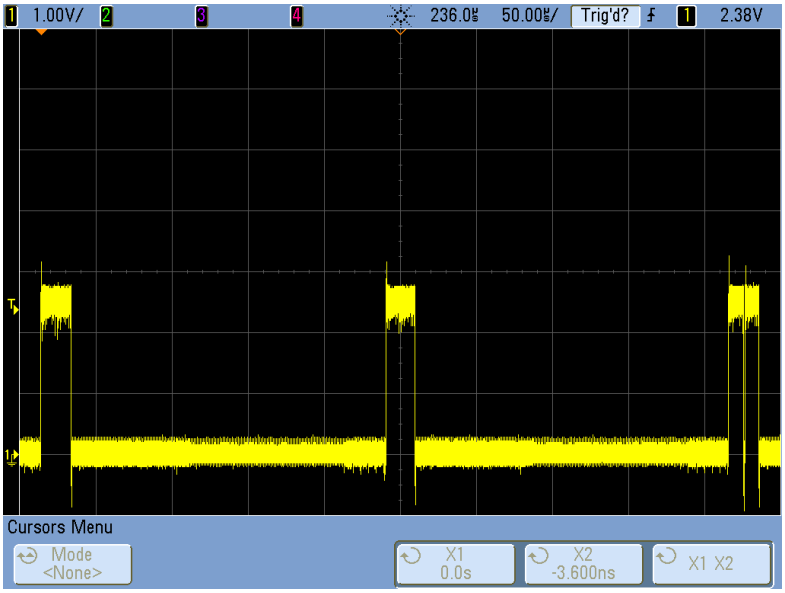

Figure 5.23: Three consecutive readouts of a 84 bit PISO. The down-spike in the third readout violates the expected matching between the redundant copies.

purpose another test with activity on the *tar_clk* signal has been performed using the logic analyzer to check whether the LFSR counters actually change their output with every clock cycle. Assuming the PISOs are filled correctly with the counter values, in this case the PISO readout pattern would be distinctly different to the previous one and similar to the expected, known pattern.

To keep it simple the clock has been toggled for one cycle to let the LFSR counters do exactly one transition before readout. The expected pattern was "000000000100000000000000000000110" for the LFSRs and "0000011111" for the up-down counters. The result again was different for each PISO. Many PISOs (like PISO84[7]) showed the exact same data as without clock activity, while others had, as expected, additional transitions in the data pattern. However none of the PISOs showed the expected pattern. The PISO matching the expected pattern best was PISO84[0], whose data is summarized in the waveform graphic 5.24. Three measurements have been performed with three redundant readouts each. "Run2_3" therefore is the third transmission of the data from the second measurement. It can easily be seen that none of the nine waveforms matches the expected data. Furthermore only one waveform pair (run2_3 and run3_2) is identical to each other, indicating the same fault as before. However a similarity between the expected and measured waveforms can be detected. This suggests the LFSRs are working properly and that the problem indeed is inside the PISOs.

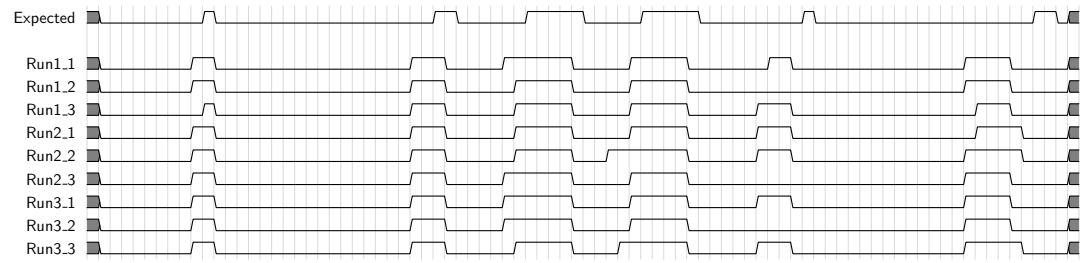

Figure 5.24: Data of PISO 84[0] at a 1 cycle measurement. It is clear that the expected data was not delivered by the chip.

5.6.1 Test Conclusion of the second ASIC

As the test results did not show the expected values, a problem in the ASIC had to be assumed. Since even reading the counter data in the reset state did not show the correct pattern for the 84 bit PISOs, the problem could be narrowed down to the reset logic of the counters or the PISO readout logic. The reset of the up-down counters for the 30 bit PISOs worked however. As the up-down counter data in the 84bit PISOs was distorted it can be concluded the problem most likely is in the readout logic of the 84 bit PISO. Further, the fact that sequential readouts without resetting the counters in between gives inconsistent results is an indication that the the problem lies within the PISOs. Whether other PISOs types are affected as well can not be tested. The glitches seen in the 30 bit PISO could be also caused by this problem. A fabrication fault could not be ruled out at this point, so a third ASIC has been bonded to a PCB and tested.

5.7 Verification of the third ASIC board

As the board is from the same batch as the previous one, there was no need to check the routing of the signals. However the board, bonding and ASIC has been checked for obvious damages with the bare eye. The bonding of this ASIC has been checked under the microscope by our colleagues in Brno. They confirmed the bonding was successful. For this ASIC the same bonding technique has been used as for the previous one.

5.7.1 Results of the electrical analysis of the third board

The board has also been checked for anomalies in the conducting properties. As for the previous board no problems have been found and the voltage regulator output was in the expected range for 5V, 6V and 7V. So it has been decided to proceed to the ASIC tests.

5.8 Verification of the third ASIC

The main goal of testing the third ASIC was to see whether it confirms the behavior of the second ASIC, whether it behaves as specified or if it shows another unexpected behavior. In particular the phenomenon with the second up-down counter in the 30 bit PISO and the inconsistent and wrong readouts of the 84 bit PISOs had to be analyzed in this ASIC sample.

0 cycle measurement Reading out the PISOs in normal ASIC operation before triggering the *tar_clk* signal, showed worse results than the second ASIC. Instead of one PISO (PISO[7]), three other PISOs (PISO30[8], PISO30[9], PISO30[14]) showed the problem known from the previous ASIC. Their waveforms also contain falling glitches as shown in following table. No rising glitches however could be observed in the 30 bit PISO results. The falling glitches have been marked as "L" in table 5.5.

PISO ID	pattern
expected	111110000011111000001111100000
[8]	111110000011111000011111100000
[9]	1111100001L111100001L111100000
[14]	1111100001L111000001L111000000

Table 5.5: 30 bit PISO anomalies in the third ASIC sample.

The 84bit PISOs as well had the same behavior as in the previous ASIC. Since the nature of the phenomenon has been shown in the previous chapter, the results have not been depicted here.

Reading out the PISOs with active target result signals showed the same results. Therefore it has been decided to not perform further tests on this ASIC.

5.8.1 Test Conclusion of the third ASIC and further actions

This ASIC sample showed the same symptoms as the previous one. In fact it showed even more problems with the 30 bit PISOs, while the 84 bit PISO data seemed to be equally distorted.
It can be concluded that the effects indeed are most likely a design problem within some PISO structures and not an unfortunate fabrication fault in the first ASIC sample. As a result the ASIC in the current version is unlikely to provide reliable measurement data for certain PISOs. A reinspection of the ASIC design revealed that a trace had been removed during a verification step. Therefore some clock signals have been floating, causing unreliable readouts for most of the PISOs. It is expected though that after fixing the problem in a new ASIC version the setup will work reliably.

5.9 Experiment in an ambient radiation environment

Only some PISOs are expected to be malfunctioning, in particular the 30 bit PISOs seem to work as intended and behaved mostly well in the tests so far. To confirm the expected behavior of the ASIC, a test is required whether the target and measurement circuits work under radiation conditions as they did in the simulations. Ideally a micro-beam is used to test all structures individually, as the exact particle energy and track is known in such an environment. However experiment time in a micro-beam facility is very difficult to get, so also research reactors with ambient radiation have been considered for testing.

An ambient radiation experiment has been prepared and conducted with the support of the Institute of Atomic and Subatomic Physics of the Vienna University of Technology. The institute operates a research reactor and offered some experiment time for testing the whole system in an ambient radiation scenario [16].

5.9.1 The reactor experiment

The reactor available at the institute is a TRIGA MARK II, a pool-type research reactor for neutron experiments with a maximum continuous power output of 250kW. The reactor offers multiple radial beam tubes as well as a neutron radiography collimator. Latter has the advantage of fewer space restrictions at the opening of the collimator, but provides particles at a lower rate. In contrast to micro-beam experiments where charged particles, typically ions, are used to create SETs, a nuclear reactor creates mainly neutrons. Neutrons themselves do not have an electrical charge, are not ionizing and therefore do not directly deposit charges into the semiconductor. Instead, charged secondary particles formed by collisions of neutrons with silicon atoms, cause SETs in the semiconductors [17] [18]. The source of the ionized particle is the main difference to a micro-beam experiment, the SET detection and data processing is exactly the same for both experiment types.
As the experiment setup needs a lot of space, the neutron radiography collimator was the better choice. However, the opening is two to three meters below the platform and there is no direct access to the collimator. Figure 5.25 is a cross section of the reactor. In order to get the ASIC into the neutron beam, the circuit board with the chip mounted on it has to be placed on an elevator on the platform above the radiation chamber. The elevator is then lowered to the height of the radiation beam.
To connect the FPGA to the PCB, either long ribbon cables are necessary or the FPGA is lowered into the irradiation chamber as well. It has been decided that the FPGA is placed on the elevator as well for multiple reasons. Firstly it can be avoided that the FPGA is inside of the neutron beam, even if it is on the elevator. So the expected particle rate on the FPGA can be kept sufficiently low. Further the data is stored within the FPGA multiple times to avoid data loss. Finally long cables would drastically decrease the signal quality and therefore enforce a reduction of the frequency. My colleague Fabian Fuhrmann prepared and conducted the experiment during a project for the Institute for Embedded Systems [19]. During the preparations a frame has been developed to mount

the equipment on the elevator. The construction ensures the FPGA and its components are outside of the radiation beam while the ASIC is right inside the beam.
With this setup all conditions for the experiment framework are satisfied.

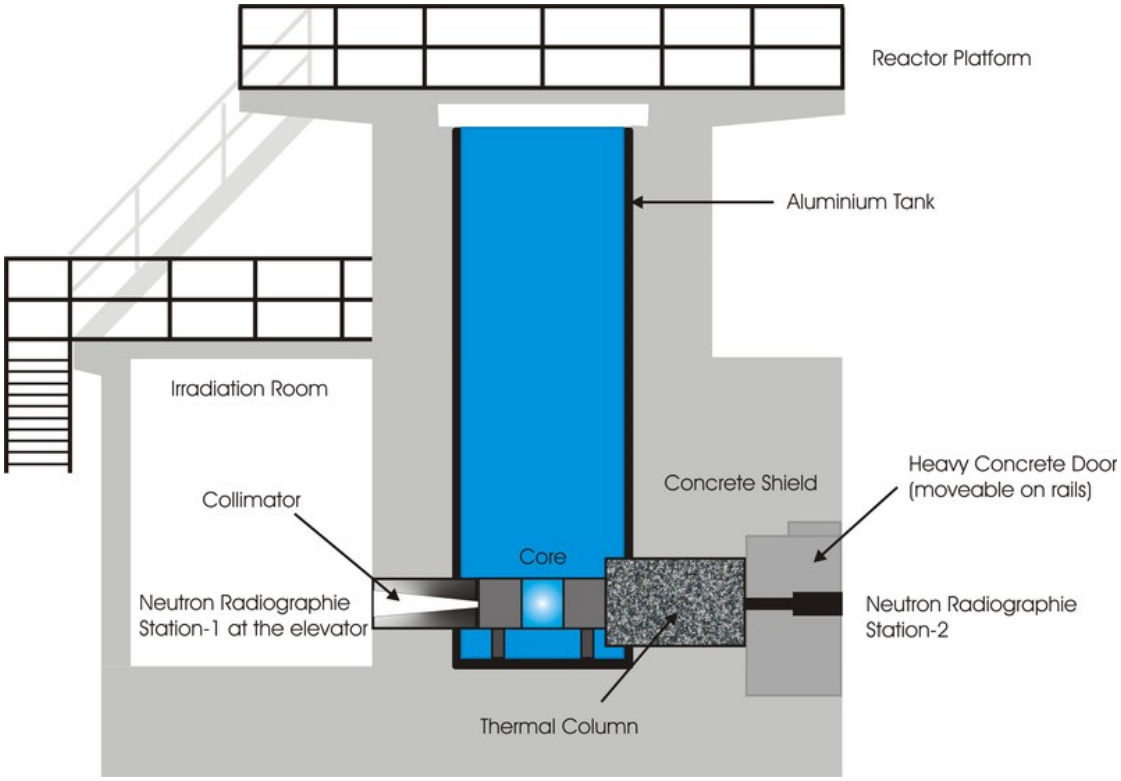

Figure 5.25: Cross section of the TRIGA MARK II[20].

5.9.2 Conducting the experiment

The reactor was out of order for large parts of the summer 2017 because of maintenance works. Thanks to Dipl.-Ing. Dr.techn. Mario Villa from the Institute of Atomic and Subatomic Physics and Dipl.-Ing. Dr.techn. Michael Hofbauer from the Institute of Electrodynamics, Microwave and Circuit Engineering we have been offered the possibility to perform some experiments between the end of the maintenance works and the start of the regular operation. This meant nearly two weeks of experiment time at partly full power and partly reduced power settings of the reactor.
A static mode setting has been chosen, meaning no activity on the *TAR_A_D*, *Tar_read* or *Tar_write* signals. As the experiment was not time critical and data integrity was a priority, a low read out frequency of 100 kHz has been chosen. Also a low target frequency of 100 Hz has been set, meaning 6000 cycles for a one minute measurement. The duration has been set to one minute in order to get a good data to overhead ratio and still having a reasonable time resolution.

Date	minutes under radiation	power [kW]
21.08.	387	250
23.08.	415	v
24.08.	414	v
25.08.	408	v
28.08.	408	0.1
29.08.	235	v
30.08.	410	0.1

Table 5.6: Timetable of experiments under radiation. The power setting "v" means variable, as the power setting varied on some days.

The irradiation chamber is shielded from the reactor core by two independent mechanisms: A shutter and a water tank. To initiate the experiment firstly the water tank is drained and then the shutter opened. This allows the neutrons created by the operating reactor to reach the ASIC and cause SETs, while the measurement infrastructure and the operators are kept safe from radiation during set up. The measurement is then started from a laptop running the front-end software. The exact operation duration and power settings varied every day, table 5.6 gives an overview. Days with varying power settings are marked with "v". Once the measurements are done for that day, the shutter is closed and the water pumped back into the shielding tank. The measurement is not stopped, as measurements without radiation serve as a good long term test of the framework.
The analysis tool filters the data collected during the day and compares it to the expected values. Only the 30 bit PISOS 1-7 as well as 9-18 have been considered in this analysis, as these PISOs have shown reliable values in the ASIC verification step. This automatic analysis makes it easy for the operator to spot abnormalities.

5.9.3 Experiment results

As can be seen from table 5.6 about 44 hours of experiment data have been collected, some of which have been captured at full and some at a reduced power setting. During the whole time no unusual activity in the observed ASIC parts could be detected. One explanation is that the particle rate is too low in the neutron radiography chamber, so the number of particles hitting the small susceptible area of the transistors is too low to create upsets during the experiment duration. Another possibility is that the particle rate is sufficient, however, the energy of the neutrons is not. Since only digital effects are detected by the measurement setup, the particles need a certain energy to create SETs, otherwise the charge induced by their impact will be unnoticed by the measurement infrastructure. It can be concluded that with this radiation source feasible measurements would take long time. However moving to a radial beam tube offering a higher particle rate could be an option for future experiments. To meet the space requirements of those at least a redesign of the PCB would be necessary. A micro-beam would be an even better

option to thoroughly test the ASIC modules and collect measurement data, however, it is difficult to get access.

CHAPTER 6

Discussion

6.1 Discussion of the ASIC verification

Verification of the ASIC samples showed that a problem with the first version existed due to a design flaw.

Reading out the counter values while counter reset was active delivered incorrect results, however the counter reset mechanism has been found to work. So the defect is unlikely to be a counter problem. The problem occurred in LFSR and UDC data, what is another evidence the fault is not specific to any of the counter types. Excluding the counters, there is just the PISO left as a source of error.

Further sequential readouts of the PISOs showed inconsistent results. This indicates a PISO problem.

Therefore the ASIC tests have been suspended and further intensive investigation in the ASIC design and simulations have been conducted by the ASIC development team. Simulations unveiled a problem with a clock trace, which resulted in a floating signal within the PISOs and led to unreliable readouts.

This finding lead to the decision to postpone the experiments and to correct the chip design in a second batch. Performing experiments is expensive in labour time as well as resources and using a corrupted ASIC sample would not yield usable data. Therefore it is beneficial the chip has been verified in advance.

The problem has been resolved in a new ASIC design version and the whole design has been re-simulated in a pad-to-pad simulation to analyze the solution and rule out any other design flaw.

Once produced the second ASIC batch will have to undergo the same test procedures as well, however as multiple test indicators pointed to the same PISO problem, it seems likely the next version will perform as expected.

There is no possibility to directly test the target circuits or the up-down counters on the ASIC. There could be untestable flaws hidden. However the simulations did not show any problems in the target circuits or up-down counters.

The LFSRs are testable as they change the pattern of the 84 bit PISOs in a certain fashion. However since its data can not be read out reliably, no definite conclusion can be drawn whether they work as intended. However, it can be said that read out patterns similar to the expected ones were observed, which signifies the LFSRs are working correctly.

6.1.1 Lessons learned for the ASIC redesign

Improvement of the PISO clock design: One modification in the new ASIC design version was to use only one common clock for all PISOs. This will reduce crosstalk in the cables and significantly speed up the pad-to-pad simulation, enabling a faster development cycle. It also frees up five ASIC pins which can be used for other on-chip tests.

Test access for elastic pipeline Since at the testing stage verifying the up-down counters individually was not possible, an additional, isolated testing circuit was suggested to be incorporated into the design which would enable more comprehensive testing of the the elastic pipeline. The elastic pipeline is the basis of the up-down counter and thus a key component of the design, so enhanced testing capabilities would be beneficial for the ASIC verification. The pipeline test would require a few pins and only little area overhead. To control the test logic some of the pins previously used as PISO clocks could be adopted. The test logic would be isolated from the actual measurement logic to avoid interference between them.

6.2 Framework discussion

Compared to the old version of the ASIC controller the new framework offers more functionality and flexibility. Because of the dynamic measurement duration and real time data analysis it is also capable of performing optimized radiation beam experiments. This maximises the data that can be collected within a typically very limited time slot in a radiation beam facility.

The front end software only offers a command line interface, which makes it less user friendly than a GUI, on the other hand it is more easily portable to other operating systems than a GUI. This is very important as it doesn't require a certain operating system on the experiment infrastructure computer. Also the command line interface is much more powerful than a GUI, since it can easily be controlled by a script.

The analysis tool still could be improved for getting even more information, depending on the experiment. For example a probability analysis in case of more than two SETs is not implemented yet. However the experiment duration can be controlled to largely avoid the case of more than two faults per target circuit.

6.2.1 Lessons learned for the framework optimisation

On-board high frequency oscillator An additional improvement was to add a resonator directly onto the PCB to generate the high frequency target clock for the ASIC and to incorporate a switch to select between the resonator and the target clock coming from the FPGA. However, using this approach the clock would be fixed to a certain frequency and would make the data analysis potentially challenging.

ASIC - FPGA signalling optimisation As pointed out in the verification chapter the setup has frequency limits because of the interconnection between the ASIC and the FPGA. The framework itself should be able of handling frequencies up to 200MHz. Using an appropriate cabling and signalling method this value seems feasible in practice. The following paragraphs discuss the signalling and methods to improve it.

The cable pin layout was not an optimal choice for high frequencies since it facilitates crosstalk. Alternatively to merging the PISO clocks, the pin layout can be improved to avoid neighbouring clock lines and to increase the maximum operating frequency.

The maximum frequency of a setup can only be tested once the readout behaves as expected, otherwise bit errors caused by the PISO readout cannot be distinguished from crosstalk errors caused by a frequency exceeding the operating range. As a result no maximum frequency can be stated for the current design. However the measurements indicated the maximum frequency of the current setup to be in the range of 20MHz.

Ribbon cables in combination with single-ended signalling are sufficient for low frequencies, but the effect of crosstalk on single-ended high speed signals has been underestimated. Therefore changing to LVDS and appropriate cables, like the FMC (Vita57.1) type, is crucial if high frequencies are needed. However, using LVDS implies the development of a more complex PCB design.

CHAPTER 7

Results and future work

7.1 Results

To validate the FRad chip of the FATAL project an experiment framework has been developed. The framework is capable of controlling the FRad chip in order to perform measurements with adjustable duration and it can provide a dynamic high speed clock to the targets on the chip. Further high speed data readouts are supported by the framework itself, presuming an adequate cabling solution is used. To ensure correct data readout under all circumstances data storage and transmission are redundant. After each measurement the framework can analyse the data in real time and give feedback about the single event upsets to automate experiment sessions. This is important in facilities with time-limited access, such as micro-beams.
Due to the adjustable measurement time the setup is also suitable for long term exposure experiments, such as in nuclear reactors. In this case, however, the FPGA should be spatially separated from the radiation area or shielded.

To ensure proper operation all parts of the framework have been exhaustively tested. The modules of the FPGA design have been tested in individual and integration simulations as well as integration tests in the real FPGA. Also the front-end software in combination with the FPGA design has been tested.
An essential point of the thesis was the verification of the PCBs and the ASICs. The PCBs and bonding have been checked visually and using the test procedures described to rule out short circuits and open circuits. Based on this framework finally the functionality of the FRad chip itself has been investigated by stimulating the ASIC as during a regular measurement and comparing its output to the expected data. Discrepancies in the data could be found and their origin narrowed down by further test routines. The ASIC turned out to suffer from a problem within the output logic. The findings were complemented by tests using alternative control equipment, to rule out the possibility of the infrastructure negatively affecting the ASIC behaviour. This way information about deficiencies of the

FRad chip could be provided to support the ASIC development process and to improve the design for future radiation experiments.

During an initial experiment in the research reactor of the Institute of Atomic and Subatomic Physics lasting 44 hours, we could not observe upsets anywhere in the FRad chip. Still, however, the experiment setup could be finalized and verified.

7.2 Future work

In order to meet a tapeout deadline for the second ASIC design version, incorporating further up-down counter tests into the ASIC circuit has been skipped. However, up-down counter tests could be done by performing specified micro-beam experiments.

Once the second generation of the ASIC is available all PCB and ASIC related tests have to be run again to ensure their correct behaviour.

The performance bottleneck, the cabling, can be exchanged in future projects with robust high frequency solutions such as coaxial, FMC (Vita 57) or Mictor cables. A suitable breakout board will be required for some connector types.

To further improve the performance LVDS signalling can be used to reach frequencies of multiple hundred MHz. This makes LVDS termination in the PCB and more expensive cabling (Vita 57) necessary. For this reason the idea has been abandoned for the validation step. For high frequency applications it might be considered.

Depending on the availability of suitable test environments further radiation experiments can be performed, either at a micro-beam facility or using an experiment chamber of a research reactor suitable for higher particle rates and energies. In this step actual measurement data could be acquired. However, for some experiment chambers the connectors of the PCB might need to be rearranged.

Depending on the requirements of the particular radiation environment, changes in the setup have to be made. The framework, however, was designed to offer flexibility to do so.

List of Figures

List of Tables

Bibliography

[1] E. Normand, "Single event upset at ground level", *IEEE transactions on Nuclear Science*, vol. 43, no. 6, pp. 2742–2750, 1996.

[2] V. S. Veeravalli, T. Polzer, U. Schmid, A. Steininger, M. Hofbauer, K. Schweiger, H. Dietrich, K. Schneider-Hornstein, H. Zimmermann, K.-O. Voss, B. Merk, and M. Hajek, "An infrastructure for accurate characterization of single-event transients in digital circuits", *Microprocessors and Microsystems*, vol. 37, no. 8, Part A, pp. 772 – 791, 2013. Special Issue DSD 2012 on Reliability and Dependability in MPSoC Technologies.

[3] B. Fritz, V. S. Veeravalli, and A. Steininger, "Reliable gateway for radiation experiments on a vlsi chip", *Austrochip 2012*, pp. 65–70, 2012.

[4] E. H. Neto, I. Ribeiro, M. Vieira, G. Wirth, and F. L. Kastensmidt, "Using bulk built-in current sensors to detect soft errors", *IEEE Micro*, vol. 26, pp. 10–18, Sept 2006.

[5] A. S. Varadan Savulimedu Veeravalli, Thomas Polzer and U. Schmid, "Architecture and design analysis of a digital single-event transient/upset measurement chip", *2012 15th Euromicro Conference on Digital System Design*, pp. 8–17, 2012.

[6] B. Fritz, A. Steininger, V. Šimek, and V. S. Veeravalli, "Setup for an experimental study of radiation effects in 65nm cmos", in *2017 Euromicro Conference on Digital System Design (DSD)*, pp. 329–336, Aug 2017.

[7] V. S. Veeravalli and A. Steininger, "Architecture for monitoring set propagation in 16-bit sklansky adder", in *Fifteenth International Symposium on Quality Electronic Design*, pp. 412–419, March 2014.

[8] I. E. Sutherland, "Micropipelines", *Commun. ACM*, vol. 32, pp. 720–738, June 1989.

[9] A. Ajane, P. M. Furth, E. E. Johnson, and R. L. Subramanyam, "Comparison of binary and lfsr counters and efficient lfsr decoding algorithm", in *2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1–4, Aug 2011.

[10] Xilinx, *UG810 - KC705 Evaluation Board for the Kintex-7 FPGA*. Xilinx, 07-08-2016 ed.

[11] Xilinx, *DS180 - 7 Series FPGAs Data Sheet: Overview*. Xilinx, 08-01-2017 ed.

[12] C. Saminger, *User Manual for Hpe midi*. Gleichman Electronics Research (Austria) GmbH & Co KG, 2007-11-15 ed.

[13] Altera Corporation, 101 Innovation Drive, San Jose, CA 95134, *Stratix II Device Handbook*, 2011.

[14] V. S. Veeravalli and A. Steininger, "Monitoring single event transient effects in dynamic mode", pp. 51–54, Jun. 2012.

[15] C. Wang and R. Sun, "The quality test of wire bonding", *Modern Applied Science*, vol. 3, no. 12, p. 50, 2009.

[16] H. Boeck and C. Kirchsteiger, "Probabilistic safety analysis for the triga reactor vienna", *IAEA*, p. 62, 1988. Available at `http://www.iaea.org/inis/collection/NCLCollectionStore/_Public/20/057/20057403.pdf`.

[17] T. Karnik and P. Hazucha, "Characterization of soft errors caused by single event upsets in cmos processes", *IEEE Transactions on Dependable and Secure Computing*, vol. 1, pp. 128–143, April 2004.

[18] S. V. Baranov, B. V. Vaselegin, P. N. Osipenko, A. I. Chumakov, and A. V. Yanenko, "Simulating single-event effects associated with high-energy neutrons for different vlsi technologies", *Russian Microelectronics*, vol. 37, pp. 47–54, Jan 2008.

[19] F. Fuhrmann, "Ti praktikum report", Sept 2017.

[20] Vienna University of Technology, Institute of Atomic and Subatomic Physics, Stadionallee 2, 1020 Vienna, Austria, *The TRIGA Mark-II Reactor*. Available at `https://ati.tuwien.ac.at/reactor/cross_section/EN/`.

Appendices

List of software commands

Frontend software

Run twenty measurements with 100 000 cycles each using a frequency divider and multiplier of two (100MHz):

```
./XT -n 20 -i 100000 -d 2 -m 2
```

Run five measurements with an external trigger using a frequency divider of twenty and a frequency multiplier of two (10MHz):

```
./XT -n 5 -d 20 -m 2
```

Run one measurement with 60 000 cycles in static mode (no clock activity on the ASIC):

```
./XT -n 1 -i 60000 -S
```

Run one measurement with 60 000 cycles and SET0 and SET1 set to one (per default frequency divider and multiplier are set to two):

```
./XT -n 1 -i 60000 -s 3
```

Run one measurement with 60 000 cycles and SET0 set to one and SET1 set to zero:

```
./XT -n 1 -i 60000 -s 2
```

Run one measurement with 60 000 cycles and SET0 set to zero and SET1 set to one:

```
./XT -n 1 -i 60000 -s 1
```

Run one measurement with 60 000 cycles and pipelines in “empty mode”:

```
./XT -n 1 -i 60000 -e
```

Run one measurement with 60 000 cycles and pipelines in “full mode”:

```
./XT -n 1 -i 60000 -f
```

Analysis tool

Check whether the data file “data_dump.txt” contains the data expected after a measurement with 6000 cycles:

```
./analyzer -f data_dump.txt -i 6000
```

Check whether the LFSR pattern after *expected position* iterations matches *LFSR pattern*, using the LFSR shortcut algorithm with a block size of 200 000:

```
./analyzer -C LFSR_pattern -t expected_position -b 200000
```

Generate a shortcut file with a block size of 200 000:

```
./analyzer -g -b 200000
```

Show the content of the shortcut file with block size 100 000:

```
./analyzer -s -b 100000
```

Show the state of a 32 bit LFSR after 100 000 iterations:

```
./analyzer -r -t 100000
```

Show the state of a 5 bit LFSR after 20 iterations:

```
./analyzer -r -S -t 20
```