8-2021

FPGA ARCHITECTURE AND VERIFICATION OF BUILT IN SELF-TEST (BIST) FOR 32-BIT ADDER/SUBTRACTER USING DE0-NANO FPGA AND ANALOG DISCOVERY 2 HARDWARE

Nedim Hamzic
nxh6162@rit.edu

FPGA Architecture and Verification of Built In Self-Test (BIST) for 32-bit adder/subtracter using DE0-NANO FPGA and Analog Discovery 2 Hardware

by

Nedim Hamzic

Graduate Paper

Submitted in partial fulfillment
of the requirements for the degree of
Master of Science
in Electrical Engineering

Approved by:

Mr. Mark A. Indovina, Senior Lecturer
Graduate Research Advisor, Department of Electrical and Microelectronic Engineering

---

Dr. Ferat Sahin, Professor
Department Head, Department of Electrical and Microelectronic Engineering

Department of Electrical and Microelectronic Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, New York

August, 2021

I dedicate this work to my wife Sabina who unquestionably and continuously has supported me in persuading and obtaining my master's degree program from Rochester Institute of Technology, to my mother Zekija and my father Hajder who sacrificed so much to ensure that my brother and I receive an excellent education, to my brother Nermin and his wife Edisa as well as their two kids Safia and Neyla, to my employer L3Harris and finally to rest of my family, friends and fellow engineers.

Declaration

Declaration I hereby declare and confirm with my signature that the work presented in this graduate paper on FPGA architecture and BIST is exclusively the result of my autonomous work based on my research of references published journals and articles. Research articles are properly cited, and credit to authors has been given following citation guidelines. I also declare that no part of the paper submitted has been made in an inappropriate way, whether by plagiarizing or infringing on any third person's copyright. Finally, I declare that no part of the paper submitted has been used for any other paper in another higher education institution, research institution, or educational institution.

Nedim Hamzic

August, 2021

Acknowledgements

I sincerely wish to express my gratitude and appreciation to Professor Mark A. Indovina for his continuous and unconditional support during my education at Rochester Institute of Technology. I had the privilege to participate in several of Professor Mark's classes, and I can truthfully say that structure of his classes is very much a representation of real work environments. Professor Mark's classes are based on real work problems and convey an approach on how to solve those problems. I would also like to express my gratitude to the whole RIT engineering department for having such strong engineering programs that prepare students to attack any modern-day technological challenges.

Abstract

The integrated circuit (IC) is an integral part of everyday modern technology, and its application is very attractive to hardware and software design engineers because of its versatility, integration, power consumption, cost, and board area reduction. IC is available in various types such as Field Programming Gate Array (FPGA), Application Specific Integrated Circuit (ASIC), System on Chip (SoC) architecture, Digital Signal Processing (DSP), microcontrollers (μC), and many more. With technology demand focused on faster, low power consumption, efficient IC application, design engineers are facing tremendous challenges in developing and testing integrated circuits that guaranty functionality, high fault coverage, and reliability as the transistor technology is shrinking to the point where manufacturing defects of ICs are affecting yield which associates with the increased cost of the part. The competitive IC market is pressuring manufactures of ICs to develop and market IC in a relatively quick turnaround which in return requires design and verification engineers to develop an integrated self-test structure that would ensure fault-free and the quality product is delivered on the market. 70-80% of IC design is spent on verification and testing to ensure high quality and reliability for the end-user. To test complex and sophisticated IC designs, the verification engineers must produce laborious and costly test fixtures which affect the cost of the part on the competitive market. To avoid increasing the part cost due to yield and test time to the end-user and to keep up with the competitive market many IC design engineers are deviating from complex external test fixture approach and are focusing on integrating Built-in Self-Test (BIST) or Design for Test

(DFT) techniques onto IC's which would reduce time to market but still guarantee high coverage for the product. Understanding the BIST, the architecture, as well as the application of IC, must be understood before developing IC. The architecture of FPGA is elaborated in this paper followed by several BIST techniques and applications of those BIST relative to FPGA, SoC, analog to digital (ADC), or digital to analog converters (DAC) that are integrated on IC. Paper is concluded with verification of BIST for the 32-bit adder/subtracter designed in Quartus II software using the Analog Discovery 2 module as stimulus and DE0-NANO FPGA board for verification.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The revolution and advanced progress of electronic technology are contributed by the development of integrated circuits (ICs). The ICs are at the core of almost everyday modern technological products and have been on an almost exponential evolvement path since their inception due to their configuration ability. IC allows integration of the semiconductor MOSFET transistors onto the very small silicon die that can accept billions of transistors which through the use of computer aid design tools can be configured for a specific application.

As the number of transistors in IC was increasing dictated by Moors law, the size of transistors is also decreasing almost double in size since 1971. The projected transistor size in 2023 is to be 2-3 nm as shown in Table 1.1. The push for transistor size reduction is to have integrated circuits that are more power-efficient and fast. With the size of transistors reducing and the number of transistors increasing designing, manufacturing, and verifying IC design poses gargantuan challenges. The time of testing complex IC design is associated with the cost of the part. The more complex the design that is it has a lot of transistors the manufacturing yield and fault coverage is becoming the issue that impacts the delivery of ICs onto the market.

Table 1.1: History of MOSFET scaling size

Year	1974	1974	1977	1981	1984	1987	1990	1993
MOSFET Scaling	10 μm	6 μm	3 μm	1.5 μm	1 μm	800 *nm*	600 *nm*	350 *nm*

Year	1996	1999	2001	2003	2005	2007	2009	2012
MOSFET Scaling	250 *nm*	180 *nm*	130 *nm*	90 *nm*	65 *nm*	45 *nm*	32 *nm*	22 *nm*

Year	2014	2016	2018	2020	2022	2023
MOSFET Scaling	14 *nm*	10 *nm*	7 *nm*	5 *nm*	3 *nm*	2 *nm*

The competitive market is demanding that parts be delivered in a relatively fast fashion with competitive pricing. Test fixtures for testing IC designs are also costly and have problems with loss measurements and degradation that require continuous maintenance. Not only functionality of IC needs to be tested the environmental and noise susceptibility, but power consumption also needs to be covered as well to ensure the top quality product is delivered on the market. A variety of ICs are available on the market, and they are reconfigurable for specific tasks but are also tailored designed based on customer specification. The focus of this paper is on FPGA IC, their architecture, and testing.

1.1 Field programming gate array or FPGA

One most the predominant ICs is FPGA since they have extraordinary reconfigurability and are relatively quick to design and produce. FPGAs have come a long way since the 80s where they were only designed to perform arithmetic operations. Modern FPGS are becoming more integrated with DSP cores, analog to digital converters which reduces inboard area size and for that reason, FPGAs are the primary choice of selection when designing circuit or system that has size constraints. The description “filled programming” allows for the FPGA to be configured for specific applications even when the parts are shipped from the manufacture. Feature use of FPGA is very much promising and lately, they are being used in artificial intelligence

designs due to the ability of FPGA to execute parallel tasks. Manufacturers of FPGAs are providing very easy-to-use programming software need for FPGA configuration and programming. JTAG testing allows FPGAs to be tested even when mounted on boards. With more hardware being added to FPGA to make them more attracted to the market the power consumption and efficiency are still a large obstacle for design engineers. To ensure high-quality products on the market, designers are adding the design to test or built-in self-test as part of the FPGA architecture to reduce the need for an external expensive test fixture.

1.2 Built in Self-Test or BIST

The concept of self-test is very basic, but the development of such tests depends on the complexity of ICs and their applications. The general idea of the self-test is to provide stimulus to the circuit under test (CUT) and monitor circuit outputs which are compared to the ideal working circuit response. A self-test system allows the designer as well as verification engineers to test billions of transistors on the die to ensure the functionality and quality of the integrated circuit being delivered to the end-user. BIST allows designers to integrate the testing circuit right on top of a die and with that allows the circuit to be self-tested. The addition of the self-test circuity, of course, poses the question of the power consumption, but various optimization techniques are available for engineers to optimize the power efficiency of the BIST circuit. Various BIST techniques are discussed in this paper ranging from development to application and optimization.

1.3 Research goals

In the process of researching integrated circuits their origins, application, development, and testing following research goals are to be answered and investigated:

- Purpose of BIST, types of BIST, and construction of BIST.
- Challenges and benefits with BIST.
- Application and implementation of BIST.
- Optimization of BIST for power reduction.
- Coverage of BIST.
- Architecture of FPGA.
- Programming and testing of FPGA
- Applications of FPGA.
- BIST for FPGA.

1.4 Organization

The structure of this graduate paper is organized as outlined below:

- Chapter 2: History of integrated circuits and progress of FPGA since the 80s up until now based on references provided. Present the concept of BIST for integrated circuits and challenges that need to be overcome to have a reliable BIST circuit.

- Chapter 3: Architecture of FPGA is presented in detail highlighting the main hardware aspect of FPGA. FPGA blocks are depicted and presented with detailed information on how FPGA IC is designed from the designer's perspective.
- Chapter 4: FPGA programming software and hardware tools are presented and testing of FPGAs using JTAG when FPGA is mounted to printed circuit board PCB.
- Chapter 5: Detail discussion of BIST circuit and components need to construct BIST pattern generator and output analyzer.
- Chapter 6: Applications of BIST on the various circuit, modeling of BIST using the mathematical approach based on probability.
- Chapter 7: Challenges and benefits of BIST design elaborated in this chapter.
- Chapter 8: Project overview for the 32-bit adder/subtracter presented with detailed block diagram.
- Chapter 9: Project results in form of waveform and screen captures presented regarding operation and verification of the 32-bit adder on the DE0-Nano FPGA board.
- Chapter 10: Conclusion and future work on BIST, FPGA direction.

Chapter 2

Bibliographical Research

2.1 History of IC

The roots of the integrated circuit concept are originating from the early 20th century but a significant breakthrough in integrated circuit design was made by Robert Noyce in 1959 with the first monolithic silicon IC. Transistor-transistor logic or TTL was first used in manufacturing IC throughout the 70s but then Metal Oxide Semiconductor Field Effective Transistors or MOSFET gained popularity in the 80s due to manufacturing reasons and isolation. The definition of the integrated circuit is shown in Table 2.1 which explains different nomenclatures of the integrated circuit based on the transistor and logic counts.

Table 2.1: Integrated classification based on number of transistors and logic gates

Acronym	Name	Year	Transistor Count	Logic Gates Number
SSI	Small-Scale Integration	1964	1 to 10	1 to 12
MSI	Medium-Scale Integration	1968	10 to 500	13 to 99
LSI	Large-Scale Integration	1971	50 to 20000	100 to 9999
VLSI	Very Large-Scale Integration	1980	20000 to 1000000	10000 to 99999
ULSI	Ultra Large-Scale Integration	1984	1000000 and more	100000 and more

Modern IC is considered ULSI because they easily have an excess of over a billion transistors and they can be classified as analog, digital, and mixed IC. Analog IC performs the function of converting the analog signal to digital and vice versa while digital IC has the task of performing logic and arithmetic operations where mixed IC is a combination of analog and digital design.

2.2 History of FPGA

FPGA are developed first by Xilinx in 1985. The roots of FPGAs are steaming from masked programmed gate arrays (MPGA) and traditional programming logic devices (PLDs) and the difference between MPGA and FPGA is an interconnect block. FPGA blocks are connected using programmable switches where MPGA uses integrated circuit fabrication to form metal interconnection [1]. The first FPGAs consisted of three main blocks:

- configuration logic block tasked to perform logic operations
- programmable interconnects tasked to connect logic blocks and IO pins
- input-output blocks allow interaction between FPGA IC and the rest of circuit design.

The simple FPGA block diagram is shown in Fig. 2.1 outlines the main blocks.

In Fig. 2.1 the logic blocks are shown in blue while interconnects in green and input/output block in red. Fig. 2.1 depicts the architecture of early FPGA but the latest modern FPGAs have more blocks integrated such as clock management block, DSP block, multiple analogs to digital conversions, and many more depending on the application that the end-user requires. Being able to do parallel operations is one of the main benefits of FPGA. Traditional microcontrollers depend on executing the code starting from top to bottom in sequential order, but

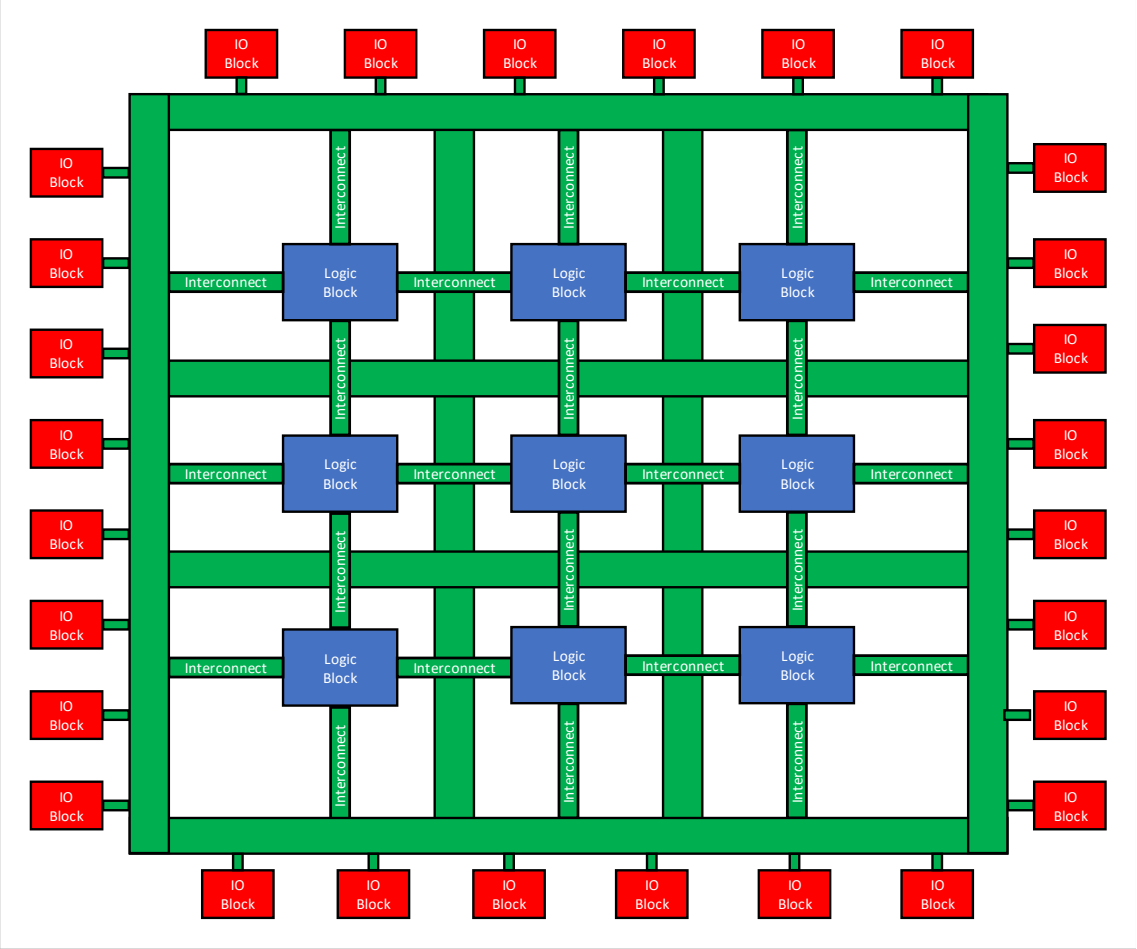

Figure 2.1: FPGA block diagram

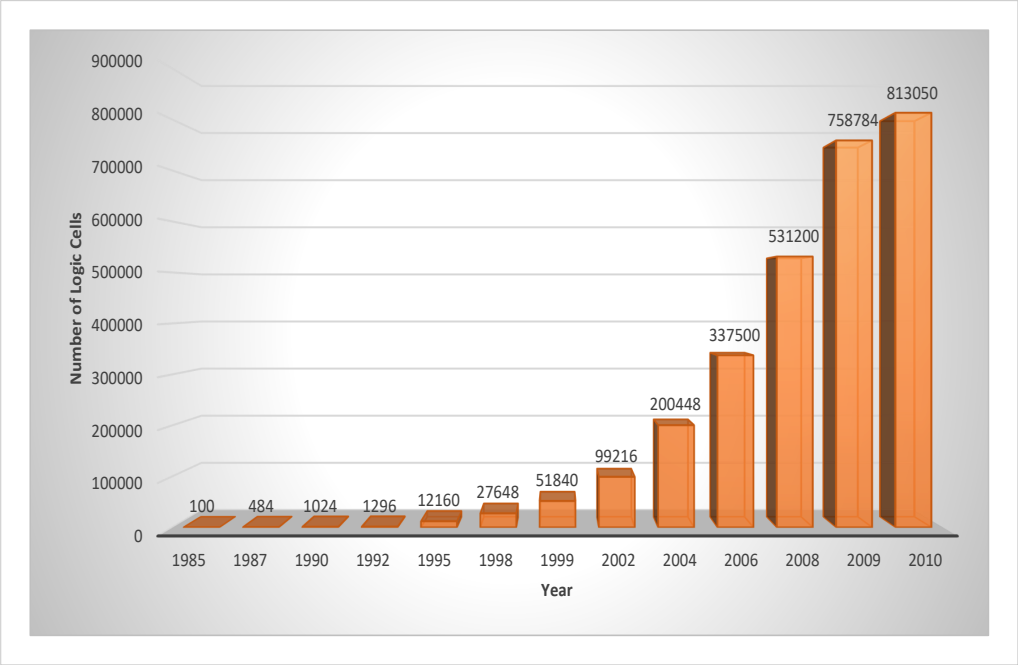

Figure 2.2: FPGA number of cell block history

FPGA can do multiple tasks in parallel which enhances speed and performance making them desired by design engineers [2].

Fig. 2.2 shows how the count of logic cell blocks is increasing since the 80s to mid-2000s which corresponds to MOSFET reduction shown in Fig. 2.3. As the size of MOSFET technology keeps decreasing the number of logic cells in FPGA kept increasing. Granted, this record is only valid through 2010 but the number of logic gates has been increasing till the present day as the technology of manufacturing integrated circuits keeps improving. Detailed FPGA architecture and analysis of FPGA blocks are more elaborated in Chapter 3.

2.3 History of BIST

Origins of BIST are also steaming from the 80s as the complexity and number of transistors have increased on IC which required expensive external test fixture [3]. The basic idea of

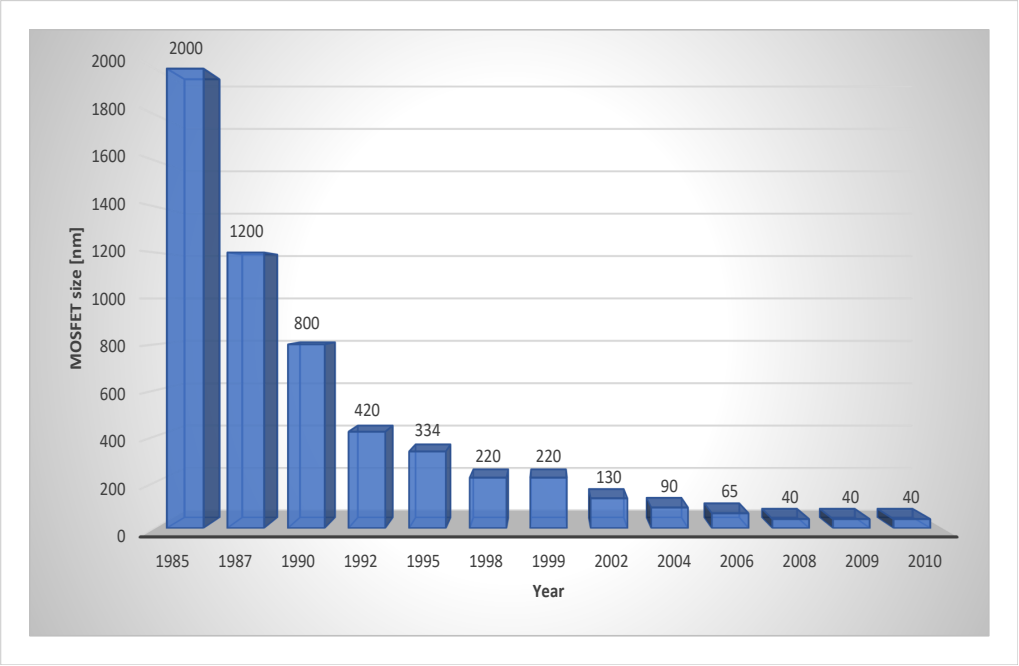

Figure 2.3: MOSFET size-reduction history

BIST is to add additional hardware that would allow to test the specific circuit and report if the circuit under test operates according to design specification [4, 5]. BIST offers a faster time of testing the circuit and provides a detailed fault description but mostly reduces the cost of testing [6]. The complexity of BIST test hardware is dependent of the circuit that is testing but general BIST consist of following blocks which are shown in Fig.2.4:

- source that is some type of test pattern generator that can have specific or random pattern generation which is a stimulus to CUT.
- circuit under test or CUT which can be logic block, analog to digital converter, or DSP block
- data compressor which collects data from various CUTs and sends it to the comparator.
- comparator block that compares response generated by the compressor and compares

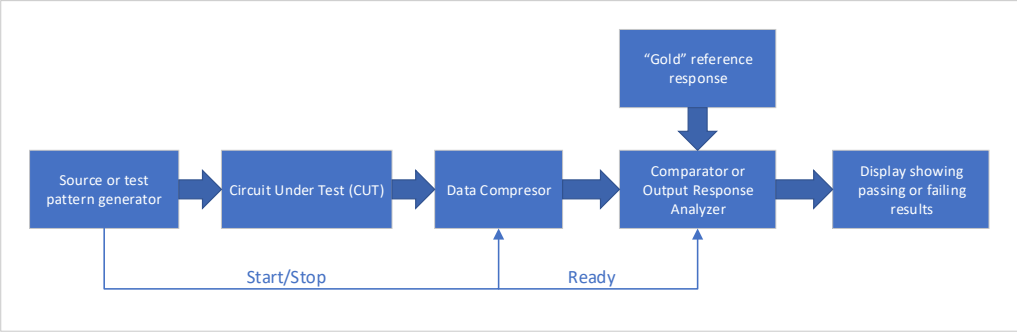

Figure 2.4: General BIST block diagram

that response to good "gold" reference model to determine if CUT is meeting design specification this is usually some type of output response analyzer.

- display block shows the coverage and fault detected.

With the addition of extra hardware to BIST, the BIST must not affect the normal circuit operation so therefore additional signals such as start, stop, and ready for comparison are added as shown in Fig. 2.4. Based on the amount of hardware being added to test IC the problem of power consumption becomes the problem that prompts designers to design and optimize more efficient hardware BIST.

The details of each block and implementation will be discussed more in detail in Chapters 5 and 6 as well as challenges of BIST with regards to optimization and area reduction.

Chapter 3

Architecture of FPGA

Fig 3.1 shows a generalized FPGA block diagram with fundamental logic blocks, interconnects, and IO blocks. However, modern FPGS have more blocks embedded and their IOs are equipped to handle various serial communication protocols such as USB, I2C, LAN, DSP blocks to compute various arithmetic operations, clock management blocks with phase-locked loops, embedded processor core, analog to digital blocks and memory blocks as shown in Fig. 3.1. This chapter presents fundamental FPGA architecture and covers the following FPGA components: logic, IO blocks, and interconnect structure between logic and IO blocks. The manufactures of FPGA offer detailed information on other blocks such as DSP, clock management blocks and others tailored specific blocks.

3.1 Logic block

The logic blocks are the fundamental block of FPGA that can implement sequential and combinatorial circuits based on configuration specifications. The design of the logic block depends on several inputs and logic operation that is executing but it also depends on the manufacture

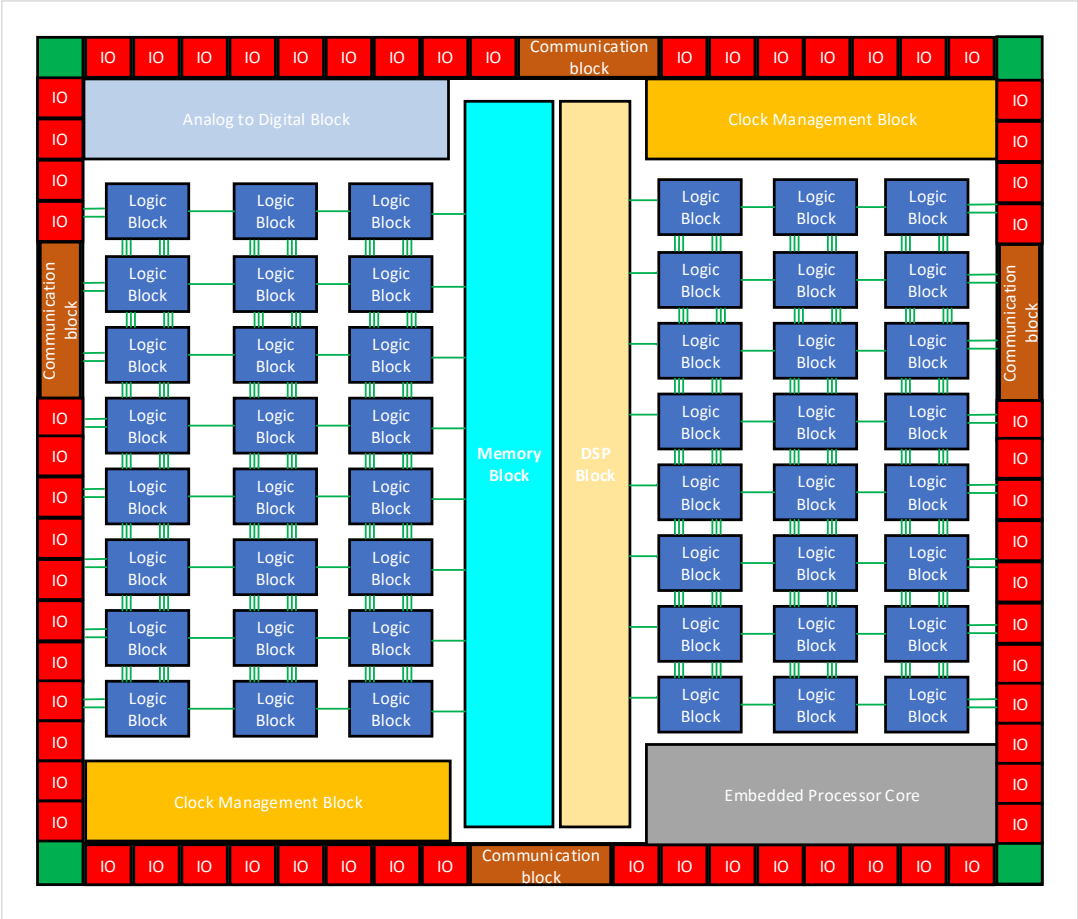

Figure 3.1: Modern FPGA block diagram

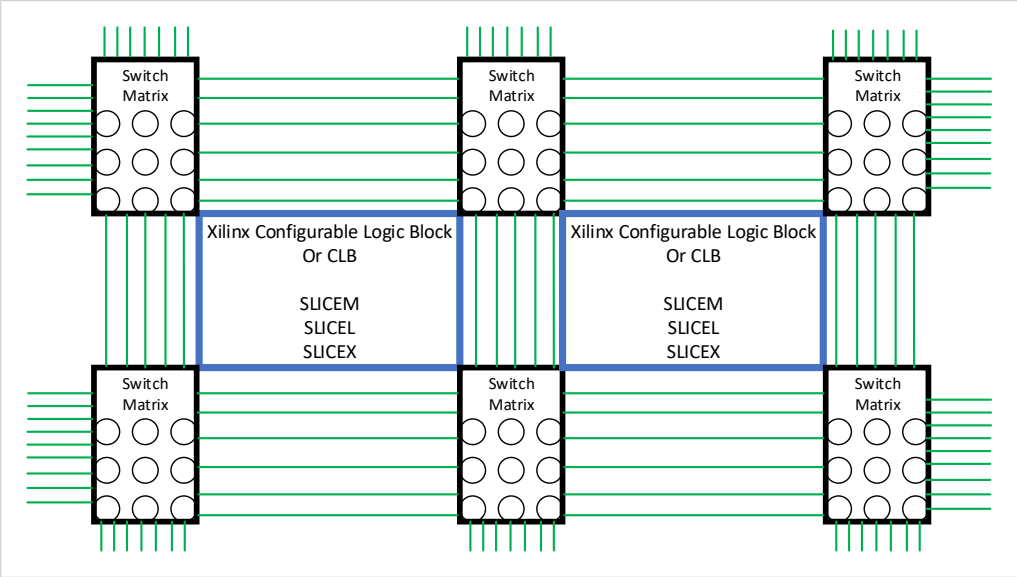

Figure 3.2: SRAM-based Xilinx logic blocks and interconnect

of FPGA and it can vary between the family of FPGAs even though they are from the same manufacturer. Several IO blocks, logic blocks, and DSP blocks vary among FPGA family products as a result of specific end-user design parameters[#nxh6162_ref14]. Fig. 3.2 and Fig. 3.3 show the difference in how logic blocks are named and connected for Xilinx and Altera FPGA.

From Fig. 3.2 it can be observed that the Xilinx block has pair of slices designated as SLICM or SLICEL and SLICEX which contain two slices labeled as slice 0 (located on the bottom CLB) and slice 1 (located on the top of CLB). Slice 1 and 0 are not connected. Each slice contains 6 input look-up tables (LUT) and 8 D-flip flops with MUXes as well as some additional circuits needed for sequential and logic operations. Specific SLICM or SLICEL and SLICEX block diagrams can be found on the Xilinx user manual for the logic block.

The Altera FPGA logic cell is depicted in Fig 3.3 and it is immediately obvious that it has a different structure than the Xilinx logic cell. Altera logic cell is constructed of Logic Array

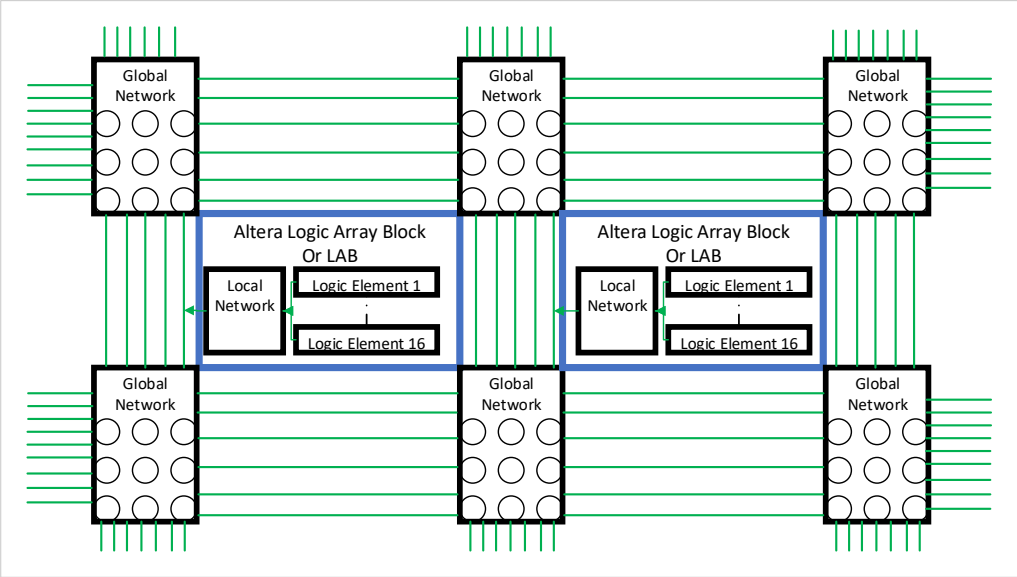

Figure 3.3: SRAM-based Altera logic blocks and interconnect

Blocks which contain up to 16 logic elements that are connected through the local network which then connects to a global network that interconnects all logic array cells. Just as Xilinx the logic elements also contain sequential components such as flip-flops and logic circuit components constructed of MUXes and lookup tables.

Regardless of the manufacture and configurable logic cell, the fundamental components are common among logic cell blocks. Logic block commonly contains the following components:

- NAND or exclusive OR gates
- One or multiple multiplexers
- Look up tables (LUTs)
- Transistor pairs
- Full adders
- D-flip flops

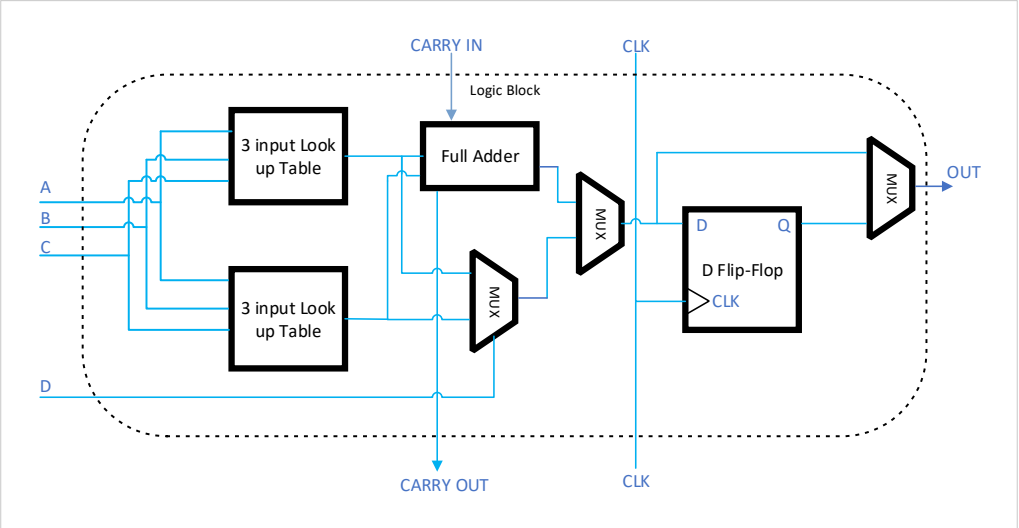

Figure 3.4: General logic block

The common logic block diagram is shown in Fig. 3.4 for 3 input logic cells, but caution must be exercised since the number of inputs to cell varies therefore resulting in different architecture that can affect FPGA power consumption, density, and area.

3.2 Interconnect of logic blocks

Interconnect structure consists of wire segments that interconnect logic blocks, IO and logic blocks, the clock signal to logic block as well as other blocks present in FPGA architecture. Just as logic blocks can vary amongst manufacture the routing interconnects structure can vary as well. For more information on routing architecture please see [2].The wires that interconnect blocks that consist of transistor switches allowing the interconnection to be configurable. Each of the circles in the switch matrix or global network shown in Fig. 3.2 and 3.3 consists of 6 transistor-based connection points illustrated in Fig. 3.5.

With 6 transistors shown in Fig. 3.5, connection between north-south, east-west, east-north, east-south, west-north, and finally, west-south is possible. The gates of transistors are

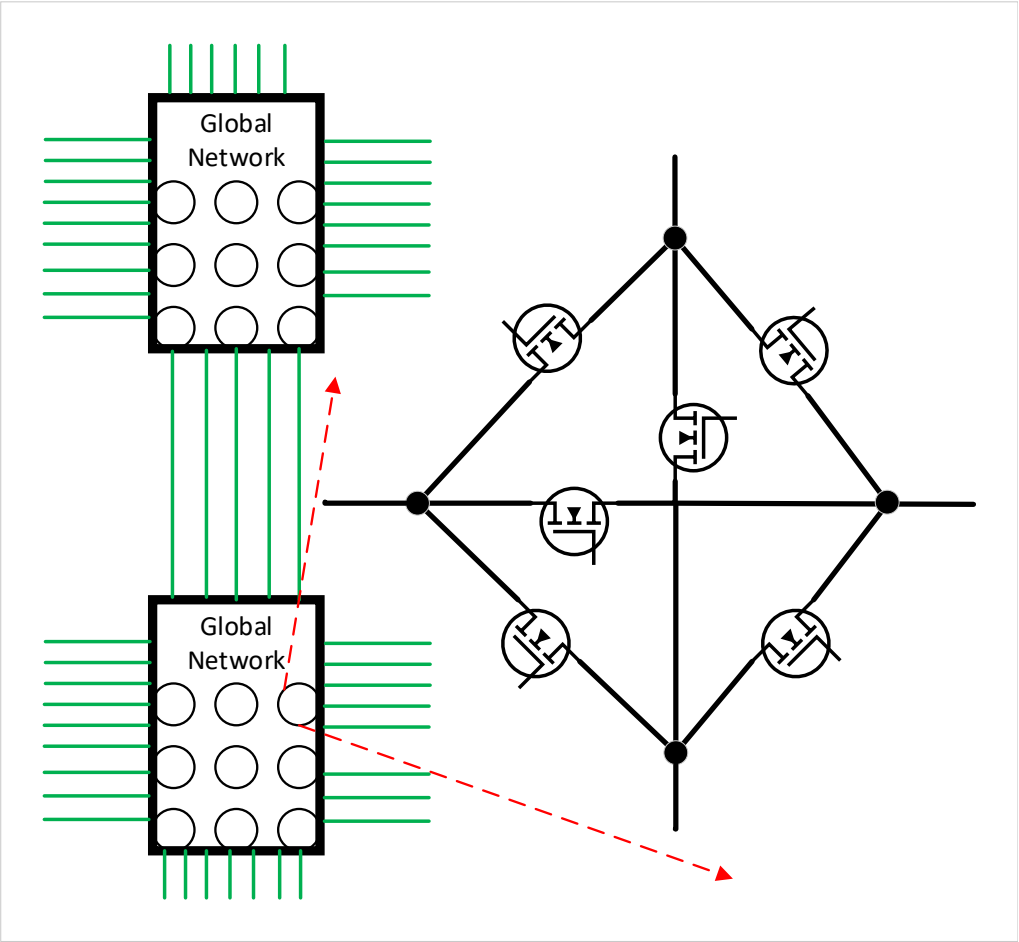

Figure 3.5: 6-transistor interconnect structure

shown not the be connected for simplicity, but internal to FPGA these gates are connected to memory cells that control the flow of switches. With a large number of logic blocks that can have up the 6 inputs, interconnection can come quite cumbersome with the addition of 6 transistors for each node therefore optimization techniques and the use of MUXes become quite beneficial when it comes to FPGA interconnect architecture. Transistors used in switch matrix should consume minimum chip area, have low on-resistance and very high off resistance to prevent leakage current, have low parasitic capacitance.

3.3 Input/Output Block

IO blocks allow interaction of internal FPGA components with other device peripherals. Fig. 3.6 shows the IO block by Xilinx. As it can be observed the block offers the following features:

- Passive pull-up and pull-down resistors
- Control of input or output with three-state control inverter
- D-flip flops for latching or direct connection to the output path
- Control of slew rate on the output with the use of buffer
- Additional MUX for selection of latch or non-latch capability
- Electrostatic discharge protection (ESD)

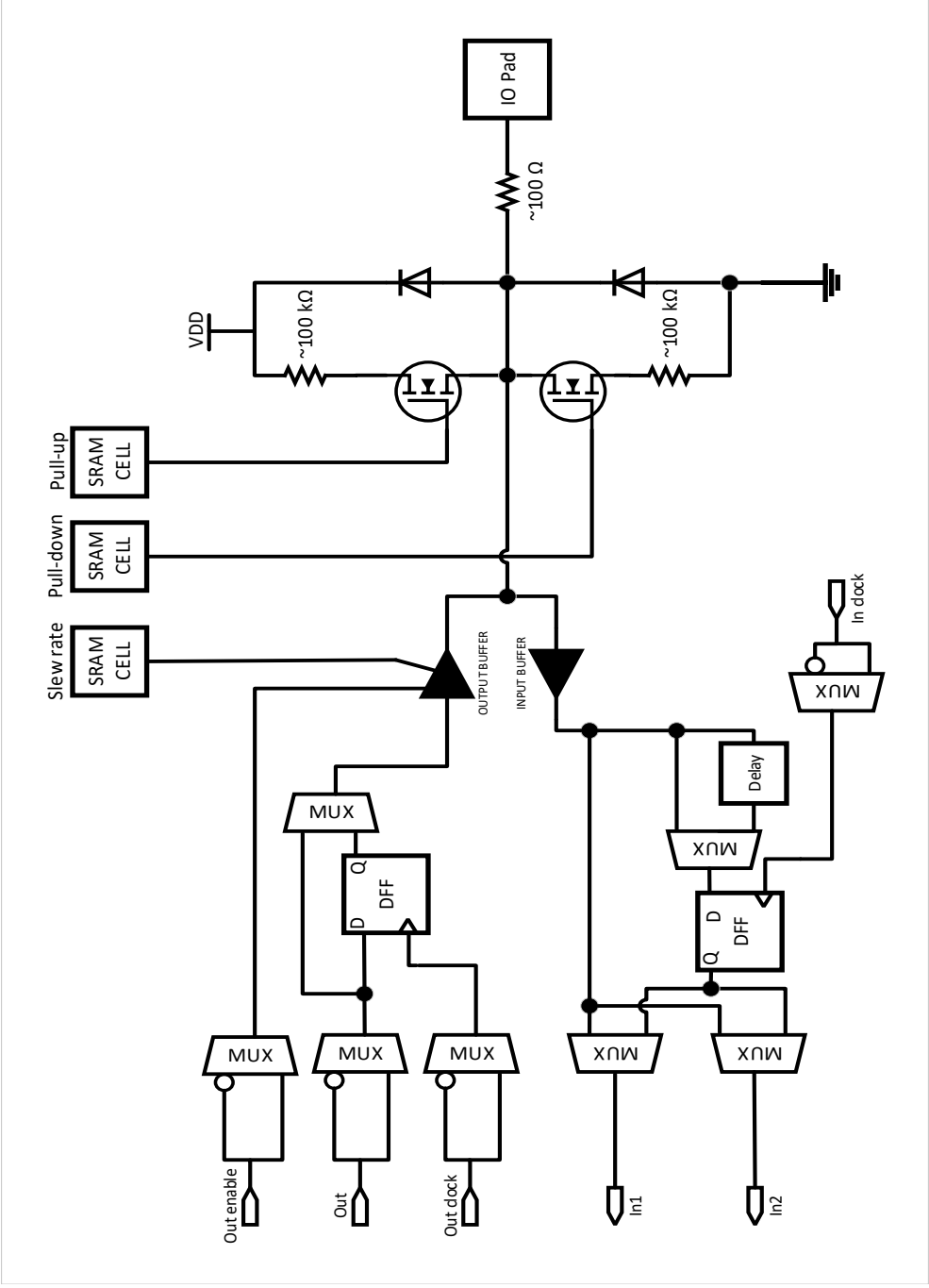

Figure 3.6: FPGA input/output block

Chapter 4

FPGA programming and JTAG boundary scan

To generate the connection between logic and IO blocks, the switch matrix needs to be programmed so that correct switches can be turned on and make the necessary connection between wires to achieve desired design parameters. Several programmable switch techniques are discussed in this chapter as well as advantages and disadvantages of those techniques and those are:

- SRAM technique
- Anti-fuse technique
- Floating gate programming technique

The process of designing the FPGA code and programming is described as well. Testing and verification of FPGA using the JTAG boundary scan and programming are elaborated in this chapter.

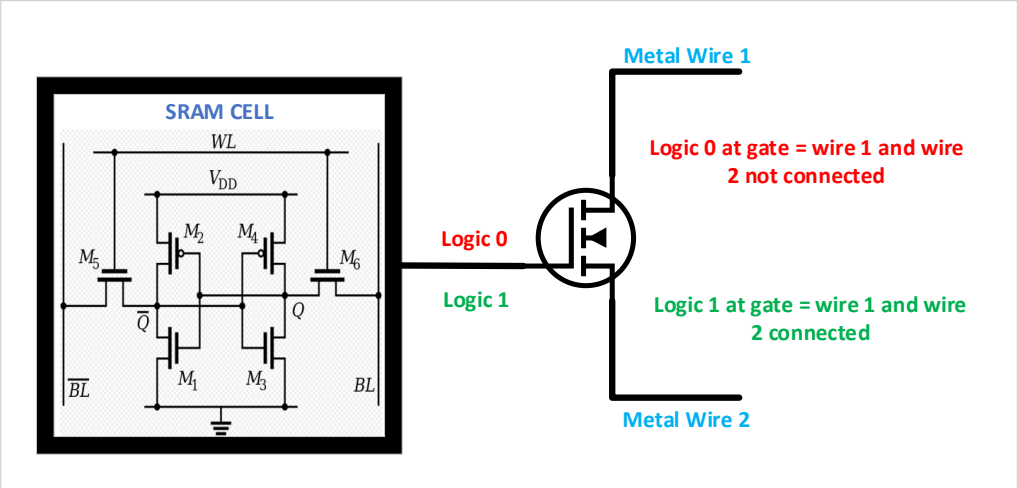

Figure 4.1: SRAM programming

4.1 SRAM programming technique

Fig. 4.1 shows the SRAM programming technique where it can be observed that connection or no connection between two wires is achieved through the use of the FET transistor. The gate of the FET is controlled by the logic value stored in the SRAM cell. If the SRAM cell has a logic value of 1, FET is in one position resulting in a low on-resistance between drain and source which in turn enables wire 1 and wire 2 are to be connected. Likewise, if SRAM stores the logic value of 0 the FET is off and two wires are not connected since there is large resistance between drain and source of FET.

The disadvantage of the SRAM programming technique is the SRAM cell. As it can be observed in Fig. 4.1 SRAM cell is consistent with 6 transistors but some of the SRAM cells are constructed with 4 transistors but even so, each configurable switch in the switch matrix of an FPGA requires an SRAM cell which significantly impacts density and area of the chip. Since SRAM are volatile the need for external storage elements (EEPROM) is required next to FPGA which can impact circuit board area and power consumption. The major advantage of the SRAM programming technique is fast reprogramming.

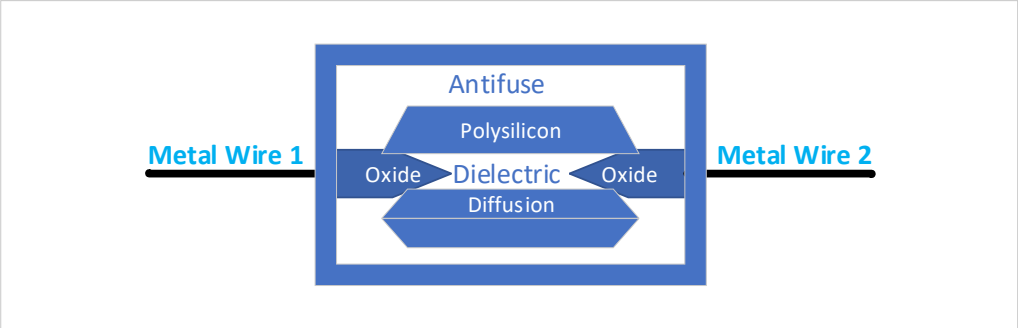

Figure 4.2: Anti-fuse programming

4.2 Anti-fuse programming technique

In the anti-fuse technique, the wires are connected via anti-fuse as shown in Fig. 4.2 by application of voltage more than 10-20V to create a dielectric breakdown that forms a conductive path that connects two wires. The downside of anti-fuse is that it can only be programmed once since the connection of two wires is permanent. The benefit of anti-fuse is that it requires no significant area on chip comparing to the SRAM technique, but the fact that it needs higher voltage to achieve connection requires the application of sturdy addressing transistors that can provide voltage and current to fuse. Lower resistance and parasitic capacitance are other benefits provided by anti-fuse programming techniques. Usually, this type of programming technique is performed in specific ASIC chips with stringent security requirements to prevent reverse engineering.

4.3 Floating gate programming technique

The floating gate programming is based on applications used in EPROM and EEPROM devices that utilize floating gate transistors to make the connection between bit and word lines. Fig. 4.3 shows the floating gate transistor used in EPROM. Transistor becomes permanently

Table 4.1: Programming FPGA technique

Programming Technology	SRAM	Antifuse	EPROM	EEPROM
Manufacturing	Desirable	Undesirable	Undesirable	Undesirable
Programmable	Yes in Circuit	No	Yes but not in Circuit	Yes in Circuit
On Resistance	600-800 Ω	100-8500 Ω	1-1.4 kΩ	1-1.4 kΩ
Off Capacitance	10-50 pF	3-5 pF	1 pF	10-50 pF
Power Consumption	Very Desirable	Desirable	Undesirable	Undesirable
Physical Size	Large	Small	Small	Small
Is it Volatile	Yes	No	No	No

"disabled" by injecting charge on floating gate 2 through gate 1 that is subjected to high voltage. Additional injected charge raises the threshold voltage of the transistor which results in the transistor is off. To turn on the transistor the UV light needs to be used on the floating gate to remove the charge. This same application can be used in making the connection between two wires. As seen in Fig. 4.3 extra pull-up resistor is required which adds to power consumption. This programming method does have the benefit of being non-volatile and does not require any extra storage devices like the SRAM. Despite its advantages, this method requires extra fabrication steps, and the resistance that bounds two wires are relatively high comparing to anti-fuse and SRAM techniques.

Table 4.1 shows a summary of the above programming techniques that highlight the advantages and disadvantages of each programming technique. Determine which technique to be used is dictated by design specifications and trade-offs. For quick programming and troubleshooting, one might choose SRAM techniques but for the final release, the anti-fuse technique might be selected for preventing design to be reversed engineered by a competitor.

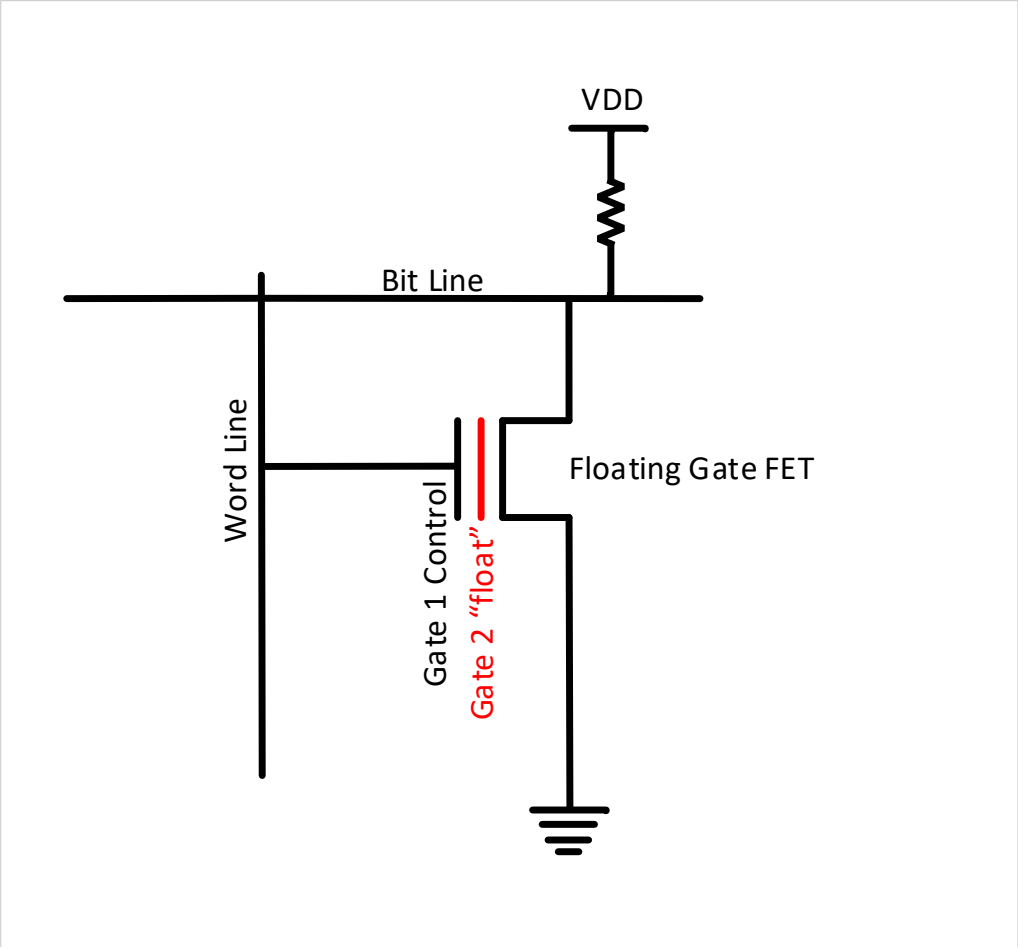

Figure 4.3: Floating gate application

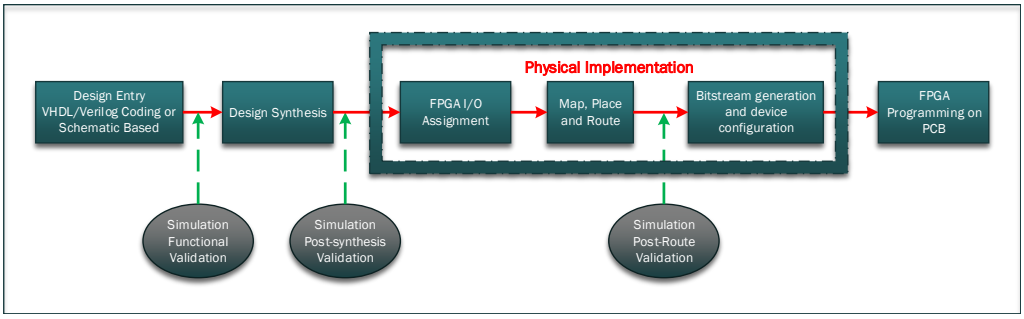

Figure 4.4: FPGA design flow

4.4 FPGA design flow

The selection of the FPGA by the design engineer is dependent on design specifications such as several input/output pins, communication protocols, analog to digital requirements, power consumption restrains, board area space, and others. Various manufactures as Altera or Xilinx offer FPGA in different packages that tailor specific design requirements. To speed up the design process and development of FPGA, development kits with Computer Aided Designing (CAD) tools are provided by manufacturers. Xilinx offers Integrated Synthesis Environment (ISE) tool, but it has been replaced with Vivado Design Suite while Altera offers Quartus II software tool. These are the most common software tools used in practice today and software packages and downloads can be obtained from the manufactures website. The basic flow of FPGA design is shown in Fig. 4.4 that starts from design entry and finishes with bitstream generation that is programmed into the targeted device.

Design entry is dependent on the designer and usually can have 4 approaches:

1. Schematic-based - which is preferred by designers who would understand design better from the hardware perspective.

2. Using Hardware Description Languages (HDL) such as VHDL or Verilog HDL - this approach is more favorable if the design is more of algorithmic nature and the designer

doesn't have solid hardware background and this is the most used approach.

3. Combination of schematic-based and HDL approaches.

4. State machine approach design entry for designers that view design as series of states, but this method is not common as above 3.

After the design entry HDL approach has been completed the next step is a synthesis of design but before synthesizing design verification of design is necessary. Functional Register Transfer Level (RTL) simulation is the first simulation performed to eliminate any code errors but most importantly verify that design is complying with design specified parameters. Synthesis translates VHDL or Verilog HDL code into device netlist format with logic elements such as NAND gate, MUXes, and flip-flops. Design optimization is executed by synthesis step and design is also check for errors by simulation post-synthesis validation.

The physical implementation consists of three steps as shown in Fig 4.4. FPGA I/O assignment is the process of assigning ports in the design to the physical elements of the targeted device such as control components like switches. The map step consists of dividing the whole design with logic elements into sub-blocks so that they can fit on the targeted device. Sub-blocks generated by mapping are connected via place and route steps. Post place and route simulation is executed to ensure that the design is obeying timing signal requirements by determining delays in design. For targeted FPGA to be programmed the design needs to be converted to bitstream through the use of the BITGEN program which through programmer can be loaded to target device located on printed circuit board.

4.5 Boundary cell and JTAG boundary scan

To reduce the area on the PCB board many FPGAs are packed and available in Ball Grid Array (BGA). Due to BGA verifying that FPGA is correctly soldered to PCB and that no ball pins are shorted or opened is very difficult to verify. Many circuits manufacture rely on X-raying the part to detect open or short pins which can be time-consuming and require a keen eye to observe and detect. To verify that FPGA or any BGA component is adequately soldered to PCB, the engineers have provided a standard called Joint Test Access Group (JTAG) IEEE 1149.1. JTAG allows for the component to be tested in the circuit to detect any open or short pins through the use of the JTAG scan cell and Test Access Port (TAP) controller. The FPGA or any target device needs to be equipped with JTAG scan cells and a TAP controller to be utilized as shown in Fig. 4.5 that depicts elements constructing JTAG boundary-scan.

As it can be observed in Fig. 4.5 there are several other registers besides the TAP controller. Those registers are instruction and data registers. The instruction register decides what to do with signals that are received, and the contest of these registers is used by the TAP controller. Data registers such as BSR which is the main testing data register and moves data from input/output pins, BYPASS register passes data from TDI to TDO ports without any specific data manipulation and IDCODES is the register that contains the identification number of the device being tested. The TAP controller is a state machine and for more detailed operation of the TAP controller please refer to the JTAG standard.

JTAG boundary scan is controlled externally via signals described in Table 4.2 visa software tools and Boundary Scan Description Language (BSDL) file that each device under test is required to have.

Boundary-scan cell or BS cell as shown in Fig. 4.5 is a 4-port cell with the following ports:

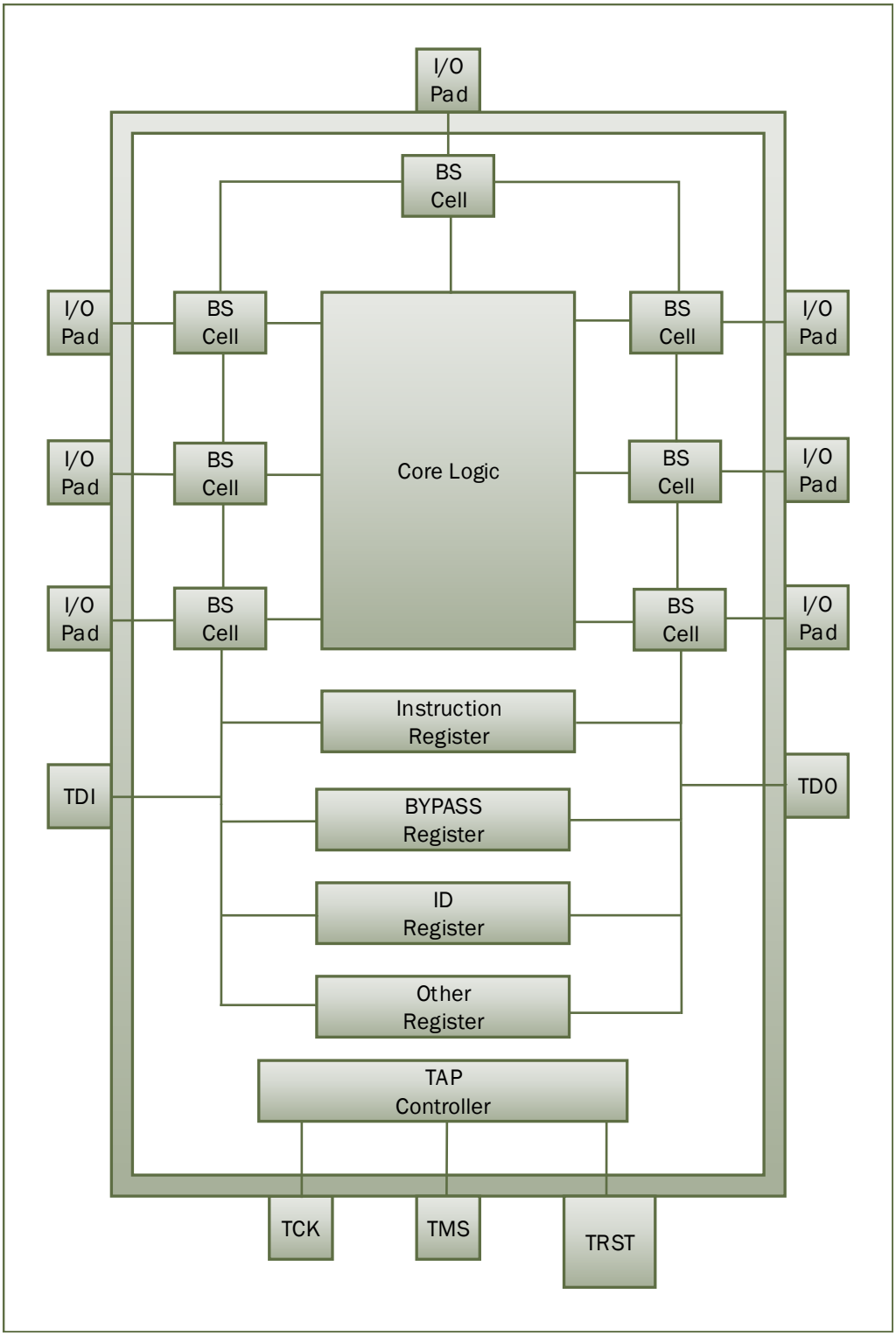

Figure 4.5: JTAG boundary scan block

Table 4.2: JTAG pins

Siganl I/O	Name	Description
TCK	Test Clock	Synchronizes the internal state machine operations.
TMS	Test Mode Select	Sampled at the rising edge of TCK to determine the next state.
TDI	Test Data In	Data shifted into the device's test or programming logic.
TDO	Test Data Out	Data shifted out of the device's test or programming logic
TRST	Test Reset	Optional and reset the TAP controller's state machine

parallel in, parallel out, a shift in, and shift out. The general schematic of the boundary scan cell is presented in Fig. 4.6. As observed from Fig. 4.6 boundary scan cell is constructed from a couple of MUXes and two D-flip flops. Data flow through the boundary scan cell as well as the mode is controlled through the TAP controller. JTAG pins can be also utilized to program FPGA and external EEPROM through use of the commercially available programmers.

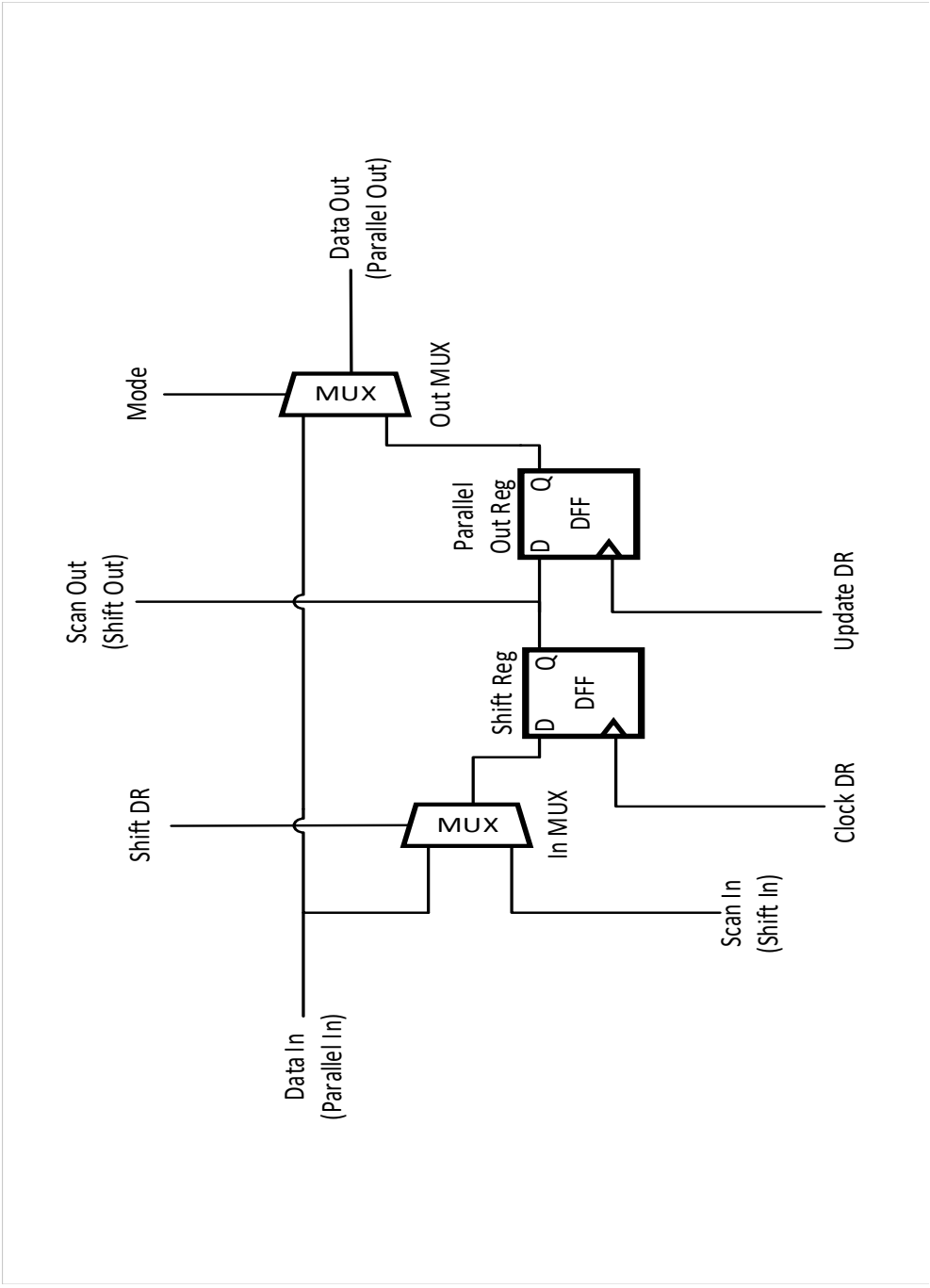

Figure 4.6: Boundary-scan cell

Chapter 5

Architecture of BIST

BIST is a relatively simple concept in which any combinational or specifically designed circuit can be tested by itself providing that test design is incorporated in the design and has a signal for controlling the test [7]. The main goal of BIST is to measure fault coverage which is defined as the percentage ratio of faults detected over the total possible circuit faults. To achieve this level of self-test and determine the functionality of the circuit being tested coordination of software and hardware tools is required. The number of combinational circuits in IC is keeping increasing and SoC architecture is becoming more complex which emphasizes the importance of using BIST to reduce test time and increase the guarantee that the product being shipped to market is 100% fault-free. BIST is consistent with the pattern generator which provides stimulus to the circuit under test and output analyzer that monitors the response of the circuit and compares that response to a fault-free model to determine test coverage [8]. This chapter elaborates on BIST components and how to utilize circuit logic to construct such components resulting in less circuitry need to be added. FPGA and SoC BIST have discussed that show how BIST is implemented on the gate level to validate circuit design.

When considering the BIST two aspects need to be taken into consideration to reduce test time with maximum fault coverage [9]:

- Controllability-forcing the value on a particular test node.
- Observability-drive the value of the test node to output so that is observed.

The two mentioned criteria of BIST are also affected by how FPGA is manufactured and packaged.

BIST can be applied to any IC that is designed for test (DFT) being analog or digital and nowadays is a standard of many ICs in order to eliminate factory fabrication faults such as stuck at 0, stuck at 1 or any other hidden faults [10].

5.1 BIST components

BIST can be categorized as follows:

- Intrusive or embedded BIST utilizes internal logic circuit components to construct test pattern generator and output analyzer
- Non-intrusive or non-embedded-utilizing external hardware and software for stimulus and analysis to achieve test coverage of CUT [3].
- Online BIST- testing while CUT is powered on and under normal operation.
- Offline BIST- testing while CUT is off and takes longer to execute.

Regardless of which category of BIST is deployed they both have common components and those components are depicted in Fig. 5.1.

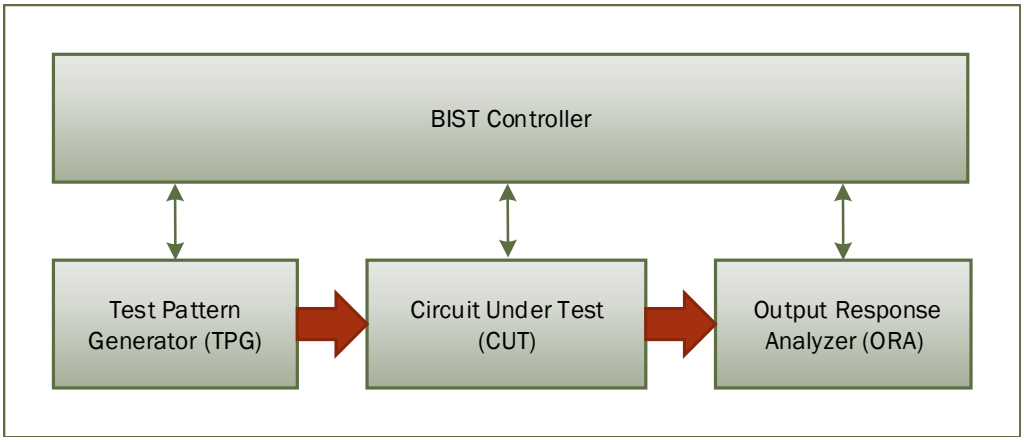

Figure 5.1: BIST components

The Test Pattern Generator (TPG) has the task of generating algorithmic test vectors that can be random or have a specific pattern. Some of the patterns are specifically designed to inject the fault to detect the fault on the output and verify that CUT is performing up to design specifications. The number of vectors is dependent on the CUT and most often they are consistent with linear feedback shift register LFSR which are used to generate pseudo-random numbers [11, 12]. The TPG is considered the most important component of the logic BIST and the number of possible input vectors depends on the number of inputs present in CUT [13]. Test pattern generation is not necessarily created by the use of LFSR but other types of random or specific generator techniques can be used.

ORA has the function of comparing the compressed response from CUT to good know working response to detect if CUT is exhibiting the fault. Compressing CUT response data is necessary for faster comparison and test time reduction. BIST controller determines when the specific node on the CUT will be placed in normal or test mode operation and it also controls the compression output data that is fed into the comparator which determines if CUT is fault free or fault is present in CUT [4, 5, 10]. The flow general flow diagram of BIST is shown in

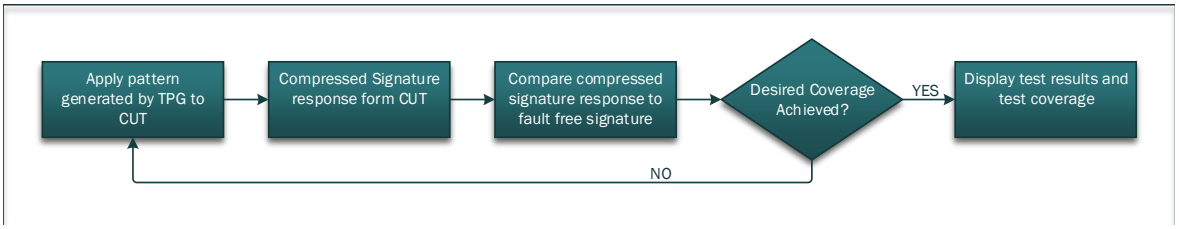

Figure 5.2: BIST flow diagram

Fig. 5.2.

5.2 FPGA BIST cases

As shown in the previous chapter that FPGA architecture is comprised of many logic, DSP, memory, and other blocks that are interconnected with the sea of network configurable switches. Each block and interconnection between blocks need to be tested and verified to deem the product shippable onto the market. Standard faults such as stuck on 1 or 0 are important in detecting but detecting delay faults is equality important due to large interconnection network which can cause undesired delays [14].

Testing of the FPGA DSP block is elaborated in [15] and optimization of DSP adders for better coverage is presented in [16]. BIST test of DSP slice of FPGA is based on the same outline elaborated in section 5.1. DSP block is tested by using two test pattern generators to escape any faults as compared to a circular test pattern generator. Output test response analyzers are constructed using the logic blocks that have been deemed fault-free. FPGA internal components are used to construct TPG and ORA and they need to be reconfigured to detect all faults. The reconfiguration of FPGA is time-consuming which can add to the longevity of a test. The main objective in this DSP BIST is the use of software to modify logic blocks on FPGA to self-test and detect faults in DSP cores.

FPGA multipliers BIST is described in [11] by using the reconfigurable VHDL files generated by CAD tools such as Quartus II to configure FPGA to produce TPG and ORA that are sufficient in testing the adder and multipliers of the FPGA arithmetic logic unit. Timing of the signal applied during LTU access is very important in FPGA that operates on the higher speeds and any degradation to the timing of that signal can affect FPGA performance so to detect a delay in the FPGA lookup table the method proposed in [14] elaborates on adding the D-flip-flop which would provide isolation in detecting the delay among CLB lookup tables logic blocks. The re-programmability of SRAM-based FPGA allows the verification engineer

to utilized FPGA blocks to create pattern generator and output analyzer by reconfiguring them over and over until desired fault coverage is achieved. Figure 3 in [17] shows how some of the FPGA blocks are configured as TPG and some as ORA to test certain rows and columns of logic blocks. Since SRAM FPGA can be reconfigured the need for external test equipment is not required. To reduce the test time of FPGA the configuration of TPG and ORA as well as blocks under test (BUT) can be constructed in the sequence where the start of one test sequence is dependent on the completion of the previous sequence.

5.3 SoC BIST cases

The software build in test (SBIT) is primarily used in SoCs to test processor cores and pipelines that connect them. Most of the SBIT tests are targeted on the functionality but the performance on the pipeline in terms of delay is not adequately covered by the functional test. Process outlined in [18, 19] specifically targets on testing pipeline functionality and performance. With the use of SBIT the configuration can be made so that generic SBIT with little to non-modification can be applied to specific SoC architecture without the need for RTL and gate modification.

The drawback of SBIT is that they are not targeting environmental variations such as temperature and noise. Just as in SRAM FPGA, SoC through the use of instructions generates a test pattern to CUT which response is stored in memory from which is read and analyzed for coverage.

Hardware version of BIST for SoC is shown with modified 2-D LFSR presented in [20] which shows significant gate reduction which results in less area and hardware verification by using transistors to generate test patterns that would optimize coverage. BIST of various SoC is comprised of the same fundamental components as shown in section 5.1. Construction of BIST on SoC is achieved by utilization of sources available on SoC to construct pattern generator and output analyzer which reduces the need for external test equipment [9]. External test equipment besides being costly and difficult to implement can introduce errors on critical delay measurements therefore using SoC elements to perform time delay measurements is more reassuring.

Chapter 6

BIST Applications and modeling

Mathematical modeling of the built-in test (BIT) is necessary analysis tools need to correctly select test parameters to ensure that test is providing reliability, testability, and performance of the product or circuit being subjected to test. Simulation of BIT with the assistance of computer-generated algorithms that can selectively inject a fault into the system is also a tool deployed by verification engineers to ensure high fault coverage and reliability. The probabilistic approach can also be used to determine the probability and confidence of fault occurring providing that there is an understanding of the circuit being tested. This chapter explores some mathematical and simulation approaches regarding BIST in section 6.1. The application of BIST on the mixed and analog circuit is explored in the 6.2 section of the chapter and section 6.3 provides the BIST technique used in other circuit test applications.

6.1 BIST modeling and simulation

Testability principles produce several simulation modeling methods as outlined in [21] and those models are presented below:

- Mathematical modeling method provides reliable analytical results, but system integration is difficult and good accuracy hard to achieve.
- Electronic Design Automation (EDA)-requires knowledge of circuit, requires more time to execute.
- Discrete event modeling- modeling based on system testing as a discrete sequence of the event which ignores internal logic faults.
- Multi-signal flow modeling- a popular method that addresses dependencies of faults.

The Stateflow which is part of MATLAB and Simulink's simulation model is far more superior to the models specified above. It is a graphical design tool that allows users to interactively make changes during testing to ensure coverage based on finite-state machine-based logic [21].The basic flow chart of the Stateflow modeling method is shown in Fig. 6.1.

Calculating fault probability allows accelerated testing to accommodate environmental variations such as temperature, noise, and aging. The understanding logic circuit that is subject to the testing probability of fault occurrence can be calculated given the random pattern being subjected to inputs of CUT [4].Fault coverage is directly related to the test vector patterns [22], but the test vector pattern needs to be optimized to meet high fault coverage without impact on test time. From probability calculation regarding the fault occurrence, the confidence in detecting the fault is also calculated. [22]

6.2 BIST mixed signal application

The concept of BIT testing the logic cores on FPGA can be applied to other circuits as well to ensure reliability and fault coverage. Modern SoC and FPGA have integrated mixed-signal circuits such as analog to digital converters and the logic BIT cannot be applied to these blocks

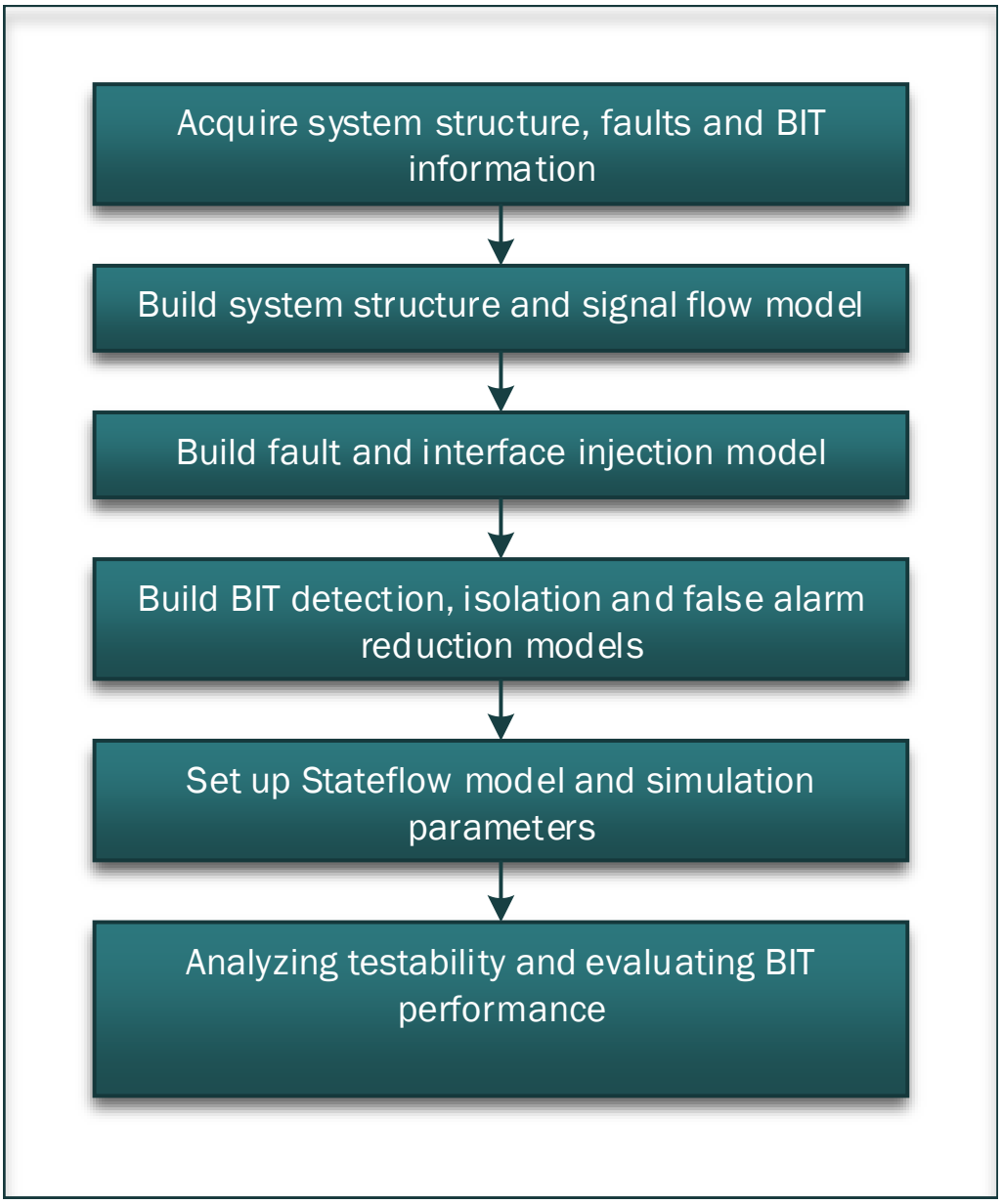

Figure 6.1: Stateflow diagram

due to their circuit architecture, therefore an additional circuit is required to test switches that are constructing the core of analog to digital converter (ADC). BIT test for analog to digital converter is discussed in [23] where ADC converter BIT is constructed by 4 parts: test generator, a converter circuit, output comparator, and timing circuit and the architecture of proposed ADC converter can be applied to any type of switching circuit. Three types of faults associated with testing ADC are discussed in the paper which are:

- Type 1 fault- occurs when some of the MOSFET switches are shorted or open.
- Type 2 fault produces a nonzero response without stimulus current/voltage provided.
- Type 3 fault-produces at leas logic value 1 when zero current is applied to the ADC.

High-precision ADC's need to have optimized dynamic and static parameters. Some of the dynamic parameters are offset, differential non-linearity while properties such as harmonics, intermodulation distortion, and signal to noise ratio are dynamic. To test all these properties requires the circuit that deploys Fast Fourier Transforms which can place strain on-chip area and power efficiency and to avoid such strains the ADC bit test proposed in [24] requires few blocks to achieve testability of static and dynamic parameters of the ADC.

Mixed-signal circuits such as op-amps and notch filters are also incorporated into SoC architecture and just as logic blocks, they need to be tested. Using external test equipment with long lead and probes can skew measuring parameters which prompt design engineers to implement BIST for testing mixed-signal circuits as well. To test filter or op-amp some type of oscillation stimulus is required which forms oscillation built-in self-test (OBIST) [8]. With the assistance of reactors and capacitors combined with logic blocks, the oscillation stimulus can be created in SoC architecture which limits the need for external oscillation testers.

Power blocks regulate power distribution on SoC and their performance can be affected by environmental factors and cause them to drift from stability. To ensure that stability of power supply converters is maintained the BIST is designed with a stimulus that is applied to the converter in order to check repose by determining poles and zeros from the transfer function. If the poles and zeros have been shifted from the targeted designed location converter loop is adjusted in other to compensate for that drift [25]. Transients occurrences can occur on SoC and [26] elaborates on the development of BIST to test node durability as exposed to transients or so-called single event testing. Proposed BIT for single event test is compromised of injecting the pulse to test node constructed of FETs, relative to clock to and measure the pulse duration and amplitude at which FET is damaged.

As it can be observed from the above-mentioned BIST application one general observation can be made that any circuit that is added to SoC architecture needs to be tested for functionality and durability. Some of BIST is more sophisticated and besides testing the circuit to ensure operation they provide a composition to the circuit to ensure any variation of the surrounding environment is taken into consideration. Some BITS can be modeled and simulated to allow test engineers to explore weak areas of design to reduce the amount of testing and provide a more durable design without cost addition. BIST techniques are dependent on CUT and CUT on the test specifications. BIST can be more involved and complex pending the circuit that they test and some of them can be simple as testing the switch on the chip. Regardless of complexity the use and application of BIT and DFT techniques are driven and evolving with the progress of modern-day IC.

6.3 Expanding BIST application

BIST application can be observed in microelectromechanical systems (MEMS) as presented in [27]. The MEMS BIST also is consistent with the same principle where the stimulus to plates is applied and their capacitance is measured which indicates with stimulus and analyzer the BIST is constructed. Dual plates in MEMS systems are introduced to achieve accurate measurements and detect defects in the etching manufacturing process. The dual proposed BIST in MEMS is applied to three devices and the simulated result of the proposed BIST showed that they can detect defects not just during the factory testing but infield as well. Besides the proposed dual test in MESM devices the concept of self-repair test (BISR) is introduced as well which in conjunction with BIST works to detect the fault and BISR corrects and adjusts that fault by switching to correctly fabricated plates that are introduced through redundancy in design.

Transceivers are very common and often found on the SoC architecture and their functionality needs to be verified and tested. The BIST for transceivers is consistent with providing the stimulus to the low noise amplifier (LNA) and measuring its response to acquire parameters such as signal to noise ratio, return loss, gain, and other critical parameters necessary for transceivers performance [28]. What is particular about this BIST is that combines external stimulus by use of RF generators, but the parameters of transceivers are acquired through mathematical equations which are executed by a processor located on SoC. The BIST controller also provides voltages produced by output impedance variation of LNA to be measured by the meter to ensure that calculated and measured voltages are consistent.

Switches that route the power signal to various SoC blocks are critical components and they as well need to be tested. When it comes to power distribution switches, the designer often adds redundant switches so in case one of the switches experiences the defect the other

switch assumes functionality [29]. The power distribution switch test module is consistent with adding few modules on SoC such as reset module constructed by D flip-flop, AND gate and inverters, ripple counter module consistent of just D flip-flop, and last indicator module made of D flip-flop and XOR gate. Even though the number of modules to test power or footer switch might seem high is still a very effective and relatively cost-efficient method comparing to expensive external test instrumentation.

Using the concept of BIST is not just utilized by SoC manufactures but the BIST concept can be applied to board circuit assembly or even larger system assembly. For example, many radio frequency (RF) designs depend on having the temperature compensated crystal oscillator (TCXO) tuned and the assembly on which TCXO is mounted has a BIST circuit that provides stimulus to TCXO until the desired frequency is reached and desired frequency is measured with an analyzer that can be internal or external to design. System-level BIT test is vastly used in industry as means of quickly detecting the defects in the system resulting in quick troubleshoot and cost associated with troubleshooting. Designing the BIT test in a large and complex system is strongly recommended and most manufacturers have some type of BIT in their product that assures customers that the product is operational. A more sophisticated system BIST test can provide detailed faults and actions that need to be taken to resolve those faults.

Chapter 7

Benefits and Challenges for BIST

Use of BIST technique regardless if it is being done on MOSFET gate level of SoC or board-level sub-system or larger more complex system does not come without of trade-off. Granted, there are benefits of adding BIST on gate SOC level but at the same time implementing BIST is not a straightforward process. The impact of an additional BIST circuit can affect the overall chip performance and it can affect some of the design specifications but mostly it can have an impact on factory yield and production. The validity of testing by BIST is questioned by manufacturing and many manufactures of SOC are considering not adding BIST and relay on external test validation. The benefits and challenges of implementing the BIST are discussed in this chapter.

7.1 Benefits using BIST

The benefits of deploying the BIST architecture are listed below with a more detailed explanation:

1. Faster test time-since the pattern generator and signature compressor/comparator are

added on-chip level they can use the chip clock to do scan and analysis faster than the frequency clocks provided by external test equipment. IC clocks can be used to drive BIST components to achieve higher scanning speeds.

2. No need for external equipment to test product, integration of pattern generator through use of linear feedback shift register eliminates the need for external expensive test pattern generators [30]. Due to the complexity of some IC designs the external test fixture does not offer an adequate number of channels to verify logic blocks.

3. BIST blocks such as pattern generators and output analyzers are constructed using basic logic blocks and they can be synthesized with regular logic blocks without additional modification to the manufacturing process.

4. Without external test equipment the cost of the part can be reduced since 20-30% of the part cost is associated with testing [31]. In other words, the less time and cost spent on the testing, manufacturers can market parts at lower prices.

5. BIST circuit can be activated through TAP controller that is part of JTAG architecture which limits the need for the extra controller to be added just to control pattern generator and scanning [31].

6. Reusability- logic blocks that are the core of IC can be reconfigured and reused to construct BIST components such as pattern generator and output analyzer through software reconfiguration. An example of such reconfiguration is shown with the FPGA where some blocks are used to construct BIST components.

7. Besides just testing the BIST can be used in the system as a self-correcting mechanism. BIST can be deployed to detect degradation of the circuit and correct performance of that circuit based on analysis from BIST.

8. BIST can be configured to measure the delay in logic circuits which is more preferred than using the external test equipment that with long wire leads can affect precise delay measurements.

9. Field testing- with BIST architecture present in SoC field testing can be initiated to analyze component's performance and this is particularly applied to critical applications such as military and space missions.

7.2 Challenges using BIST

Despite the advantages and benefits of using the BIST approach, there are some of the disadvantages of BIST that make the manufacture of IC question the validity and effectiveness of using the BIST. The disadvantages of using BIST are listed below with a detailed explanation:

1. Addition of extra logic blocks to construct the BIST components can affect the overall chip area. Each logic block in the IC core needs to be tested and adding the BIST for each of those blocks can exceed the allowable chip area requirement.

2. Power consumption- with the addition of extra circuitry to construct BIST components the power consumption of IC goes up. Extra current draws need to power the BIST scanning circuit can result in component stress and overheating making the IC more sustainable to latent field failures. Extra power demand requires more robust and redundant power transistors which can occupy large chip areas.

3. BIST components are synthesized in the same manner as the rest of logic blocks which also makes them prone to manufacturing defects [32]. To have working BIST the redundancy is added but again that can impact chip area and power consumption.

4. False failures can be reported by BIST even though the circuit that is being tested is fault free which can add extra time to test to confirm if failure is true or not.

5. Pattern generation- test patterns need to be optimized to test all logic components to achieve a short test time. The random pattern is used to detect all the faults with an exhaustive approach which can add time to test. The option of generating deterministic and pseudo-random patterns must be considered for optimization

6. BIST circuit components can be defective as well so subjecting the CUT to already defective test components can result in negative factory yield. Manufacturers need to scrap chips just because BIST components do not work even though the rest of the IC logic cores are fault-free.

7. Reliability-there are several faults that can affect a successful BIST scan:

   (a) IR-drop: change in power or ground rail voltage due to resistance between the node of interest and power/ground rail [30].

   (b) Cross talk: capacitive coupling between close nets can affect the scan output [30].

8. A large amount of data to process- output of the scan or so-called signature output needs to be stored and compared to a good working sample. Storing the scan data requires memory which can affect chip area and power consumption.

9. Variations between scans-some BIST scans can produce up to 99% coverage while the others anywhere from 60-80%

10. Security risk- BIST structure needs to be protected from external attacks since there are cases as elaborated in [#nxh6162_ref30] where manufactures of SOC BIST architecture

were attacked causing detrimental reputation to manufacture and significant recall cost to fix the issue.

As above shown benefits and challenges of using the BIST can be observed as to why some manufactures are reluctant of implementing the BIST in their product architecture since at the end of the day the goal is to have a high yield and less scrap. With transistor technology shrinking down the BIST is becoming more attractive since external test fixtures are becoming limited in bandwidth to test more complex SoC architectures that have millions of logic blocks. To make BIST more robust and reliable more attention needs to be devoted to optimizing the BIST architecture to address issues such as cross talk and IR drop. Understanding the circuit that is being tested is of the utmost importance in having efficient and reliable BIST. Test vectors produced by pattern generators need to be efficient as well since the longevity of the test, as well as fault coverage, is dependent on the deployment of test vectors. Despite its drawback, the BIST is the tool that has a promising future in the SoC manufacturing world to keep up with the fast pace IC competitive market.

Chapter 8

32-bit FPGA adder/subtracter design

Understanding developing, synthesize, mapping, and programming of the FPGA, as well as the application of BIST the 32-bit adder/subtracter design, is developed. The adder/subtracted is designed using the schematic approach in Quartus II where the design is synthesized, mapped, and stimulated before being programmed on the FPGA eval board to verify operation and eliminate any design errors. Once the design is verified it is programmed to eval the DE0-Nano board with the FPGA. To simulate BIST the Analog Discovery 2 is used as stimulus and analyzer to ensure that FPGA is programmed with a 32-bit adder/subtracter and that with provided bitstream correct output is generated. This chapter presents detailed steps taken in developing FPGA code and BIST for 32-bit adder/subtracter.

8.1 Necessary hardware/software tools

To construct the FPGA and BIST structure for 32-bit adder/subtracter following is necessary:

- PC with installed Intel Quartus 2 design environment necessary for generating adder/-subtracter schematic as well as a simulation for verification and pin assignment

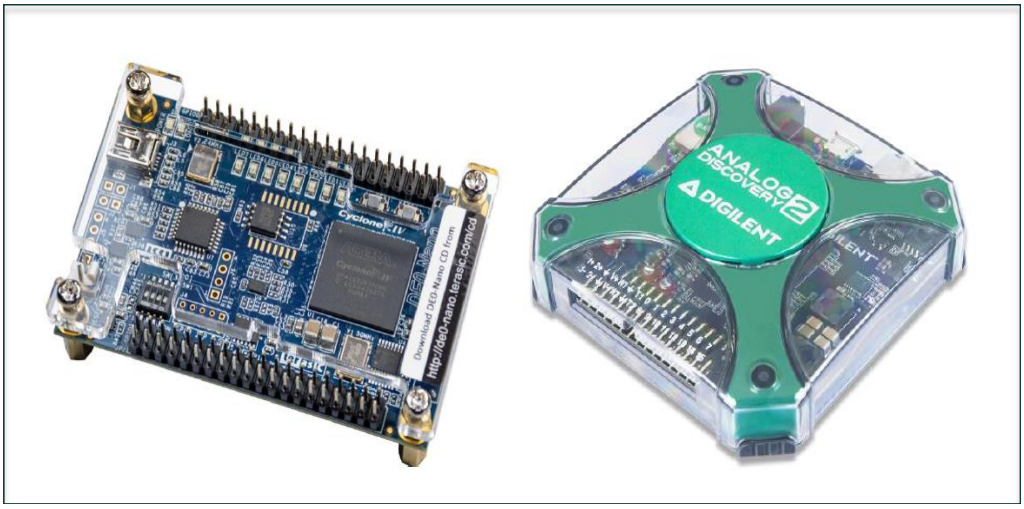

Figure 8.1: DE0-Nano board (left) Analog Discovery 2 (right)

- DE0-Nano eval board with USB cable necessary for programming. This evaluation board contains the FPGA that is programmed with the file generated from Quartus via USB blaster
- Analog Discovery instrument needs to provide stimulus to DE0-Nano board in terms of bits and it also collects the FPGA output response generated by adder/subtracter
- PC installed with Digilent WaveForm software necessary for configuring stimulus and analyze the output

Fig. 8.1 shows an image of Analog Discovery 2 and the DE0-Nano board to ensure correct hardware is acquired.

8.2 32-bit adder/subtracter BIST block

As indicated earlier the Analog Discovery tool has the function of providing the stimulus to the DN0-Nano board which in this case acts as a generator but it is also collecting the output

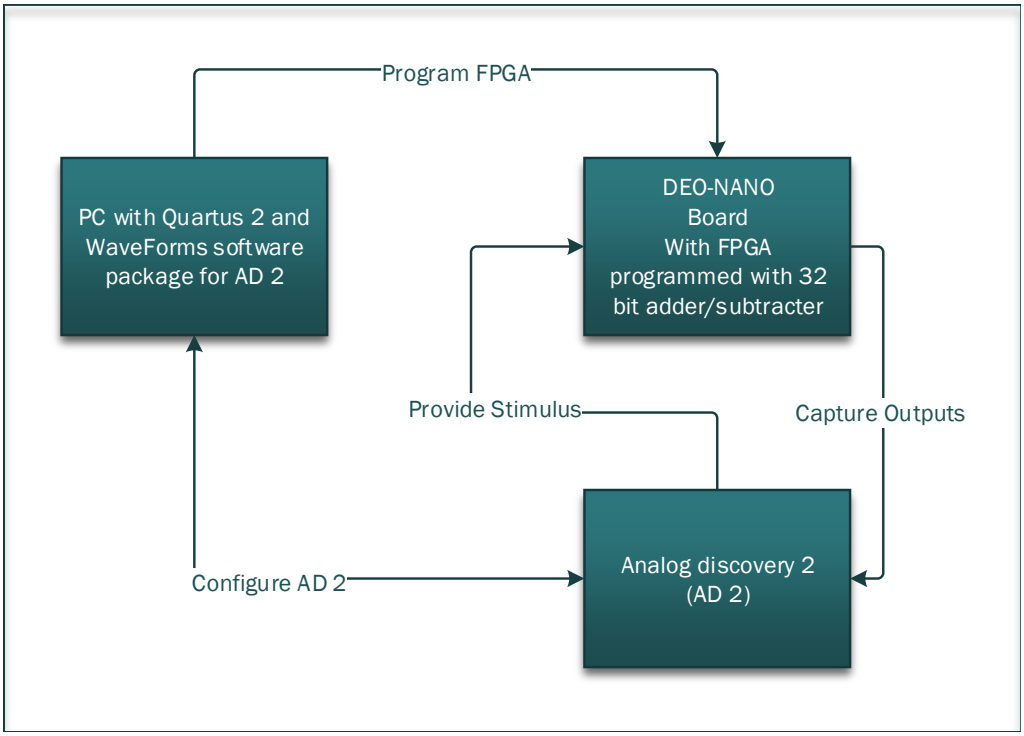

Figure 8.2: 32-bit adder/subtracter BIST block diagram

response from the DE0-Nano board which acts out as an output analyzer. DE0-Nano board is presenting a logic block that constructs the 32-bit adder/subtracter. A basic block diagram of the DE0-Nano board interacting with Analog Discovery 2 and PC software is shown in Fig 8.2. Quartus II 32-bit adder/subtracter is programmed onto FPGA located on DE0-Nano board using a USB blaster cable. Using the WaveForms software, the AD2 is configured to provide serial data input to FPGA added/subtracter and AD2 is capturing the response of the adder/subtracter to ensure that is performing according to the design specification.

8.3 Basic 32-bit adder/subtracter block

A single-bit adder is used to construct the 32-bit adder/subtracter block by using the several full adder blocks. The schematic block of the full adder is shown in Fig. 8.3 and the full adder

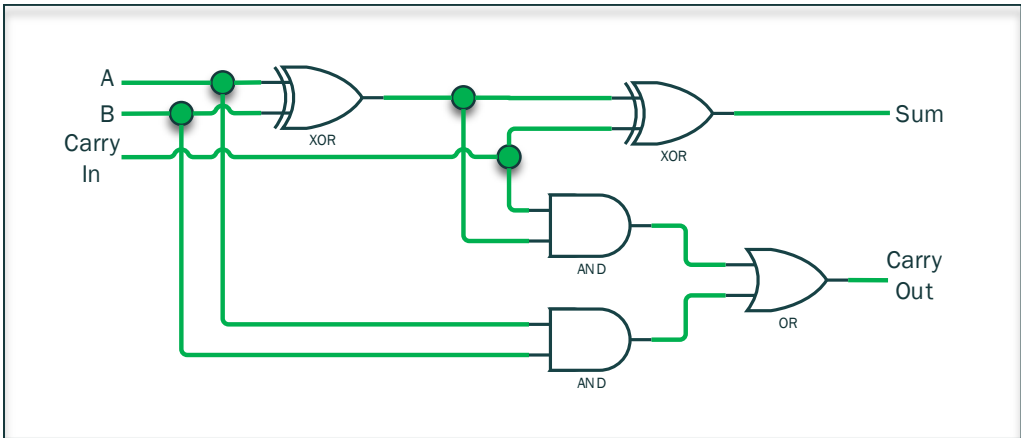

Figure 8.3: Single bit full adder block diagram

Table 8.1: Full adder truth table

A input	B input	Cary in	Sum	Carry out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

is constructed of several logic gates such as couple AND gates, couple XOR gates, and one OR gate. Full adder has three inputs which are bit A and B and Carry-In bit. The output of the full adder is Sum and Carry Out bit. The truth logic table for the full adder is shown in Table 8.1. Using the full adders and extra XOR gate the 32-bit adder/subtracter is constructed as shown in Fig 8.4. For simplicity, only the first couple and last couple full adders are shown, but 32 full adders are required to produce a 32-bit adder/subtracter which can process subtraction and addition of 32 bits. If the Operation $+/-$ bit is 0 then the addition of A and B bits is performed and likewise if the Operation$+/-$ is 1 the subtraction of A and B bits is executed. Carry-In of the first adder is connected to Operation $+/-$ bit and on the following cascaded stages Carry

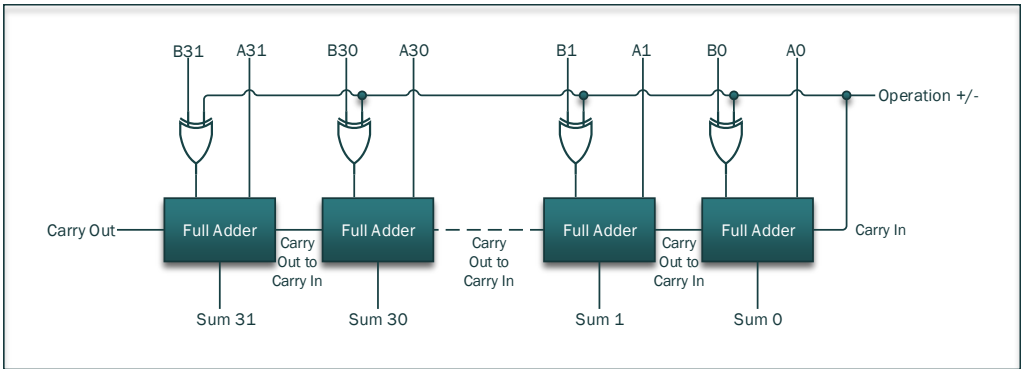

Figure 8.4: 32-bit full adder/subtracter block diagram

Out of the previous stage is connected to Carry In of the next full adder stage. The last full adder stage produces the final Carry Out bit.

8.4 Shift registers

Since Analog Discovery 2 tool has only 16 available digital IOs, which is not enough to provide two signals of 32 bits and monitor sum bits plus additional control signals such as reset, operation selection, carry in, and carry out. To overcome this issue of the number of IOs, two types of shift registers are used:

- Serial-in-parallel-out shift register used on the input of 32-bit adder shown in Fig. 8.5.
- Parallel-in-serial-out shift register used on the output of 32-bit added and its depicted in Fig. 8.6.

Shift registers are constructed mainly from the D flip-flops and the parallel data on the input shift register is controlled through the clock. The output shift register has a Shift line that dictates when the data is loaded and when shifted in the register and this line is tied to the MUX circuit that controls the shift and load path. If Shift is 0, input data is passed to the input of D flip-flops and when shift is 1 the data from flip-flops is right-shifted to the serial output.

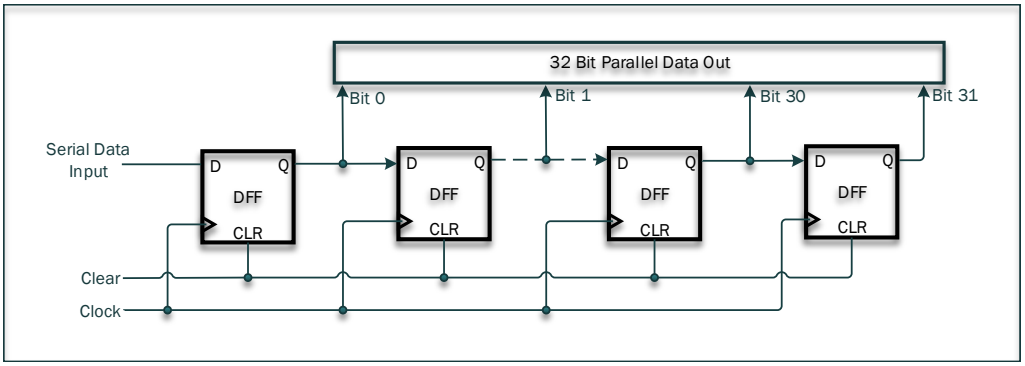

Figure 8.5: Serial-in-parallel-out shift register

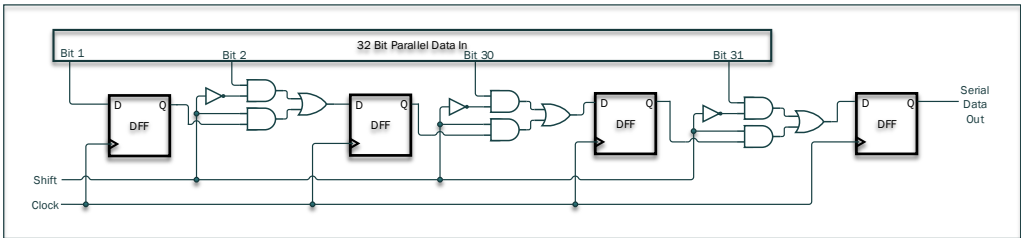

Figure 8.6: Parallel-in-serial-out shift register

8.5 Adder/subtracter with shift registers

Adder/subtracter adds two inputs designated as A and B that are 32 bits long. The subtraction or addition is controlled through Operation input. The clock input is added for the shift needed for the input and output register. Fig. 8.7 shows a full block diagram with input and output shift registers as well as their interconnect. All necessary inputs shown in Fig. 8.7 are located on the left while only two outputs are located on the right.

8.6 Wiring of Analog Discovery 2 and DE0-Nano

Using the wires from Analog Discovery 2 (AD2) make connections from AD2 to the DE0-Nano board as shown in Table 8.2. For pin location on AD2 and DE0-Nano board please refer to Appendix B where datasheets for both tools can be obtained. The last column is used during

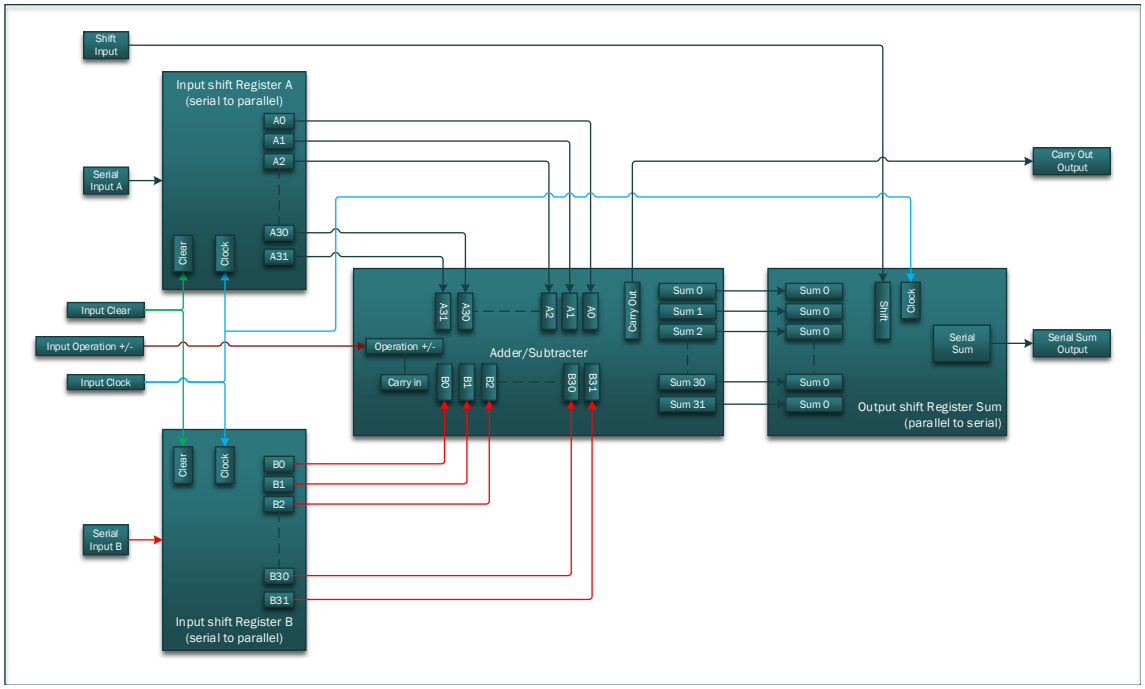

Figure 8.7: 32-bit adder/subtracter interconnect

the pin mapping.

8.7 32-bit adder/subtracter design flow

The process of building and verifying the 32-bit adder/subtracter design is outlined with the following steps:

1. In Quartus II build a schematic of simple adder shown in Fig. 8.3

2. Verify adder by performing the simulation and verify that simulation results are following Table 8.1

3. Create a block diagram of a single-bit adder.

4. Build schematic of full 32-bit adder/subtracter as shown in Fig. 8.4 using multiples of

Table 8.2: Interconnect pinout

Signal	Direction	AD2 Pins	DE0-Nano Pins	DE0-Nano Name	FPGA Pins
Serial A	Input	0	JP1 Pin 2	GPIO_00	PIN_D3
Clear	Input	1	JP1 Pin 4	GPIO_01	PIN_C3
Operation $+/-$	Input	2	JP1 Pin 6	GPIO_03	PIN_A3
Serial B	Input	3	JP1 Pin 8	GPIO_05	PIN_B4
Clock	Input	4	JP1 Pin 10	GPIO_07	PIN_B5
Carry Out	Output	6	JP1 Pin 14	GPIO_09	PIN_D5
Serial Sum	Output	7	JP1 Pin 16	GPIO_011	PIN_A6
Shift	Input	15	JP1 Pin 20	GPIO_015	PIN_C6
Gnd	NA	Ground	JP1 Pin 12	Ground	NA

single bit adder blocks created in the previous step.

5. Verify the 32-bit adder/subtracter by completing simulation in Quartus II and ensure that the design performs up to specifications.

6. Create a 32-bit adder/subtracter block.

7. Build serial-in-parallel-out shift register in Quartus II as shown in Fig. 8.5. Simulate design and verify functionality.

8. Create a block diagram for the serial-in-parallel-out shift register.

9. Build parallel-in-serial-out shift register in Quartus II as shown in Fig. 8.6. Simulate design and verify functionality.

10. Create a block diagram for the parallel-in-serial-out shift register.

11. Construct in Quartus II circuit as shown in Fig. 8.7 using blocks created in previous steps. Simulate 32-bit adder/subtracter with input and output registers and verify its functionality.

12. Synthesize design created in the previous step.

13. Mapped I/O pins on DE0-Nano FPGA using the Quartus II mapping feature.

14. Program DE0-Nano board using the USB blaster through Quartus II.

15. Connect the Analog Discovery 2 wires to corresponding mapped FPGA pins on the DE0-Nano board.

16. Using WaveForm software that controls AD2 to generate stimulus to DE0-Nano board and verify the response in WaveForm for 32-bit adder subtracter.

As seen from the above steps the process of building a 32-bit adder/subtracter is performed by building and verifying the smaller functional blocks first. Once the smaller design blocks have been verified, they are used to create the final design. This approach is typical in FPGA design where smaller blocks are built and verified ether through testbench if using VHDL design approach or simulation if using the schematic design approach. Once the final design has been verified the whole project is synthesized (it might take some time depending on design). Once the synthesis is a complete mapping of the FPGA pins is performed followed by programing of the targeted device. Hardware design is verified by providing the stimulus and observing response through the Analog Discovery 2 tool.

Chapter 9

Results of 32-bit FPGA adder/subtracter

Simulation results of the single full adder, 32-bit adder/subtracter, serial-in-parallel-out shift register, parallel-in-serial-out shift register, and 32-bit adder/subtracter with registers is presented in this chapter. The hardware simulation output is captured by WaveForm as proof that the DE0-Nano board has been successfully programmed with 32-bit adder/subtracter and register and that function of 32-bit adder/subtracter has been verified. Results are presented in form of the screen captures of Quartus II functional simulation window and WaveForm window for hardware verification.

9.1 Simulation captures

Fig 9.1 to 9.5 are screen captures of Quartus II simulation. Once all the schematics were created, they were synthesized and tested through Quartus II Waveform Editor. The captures show that each designed Quartus II schematic is performing as designed.

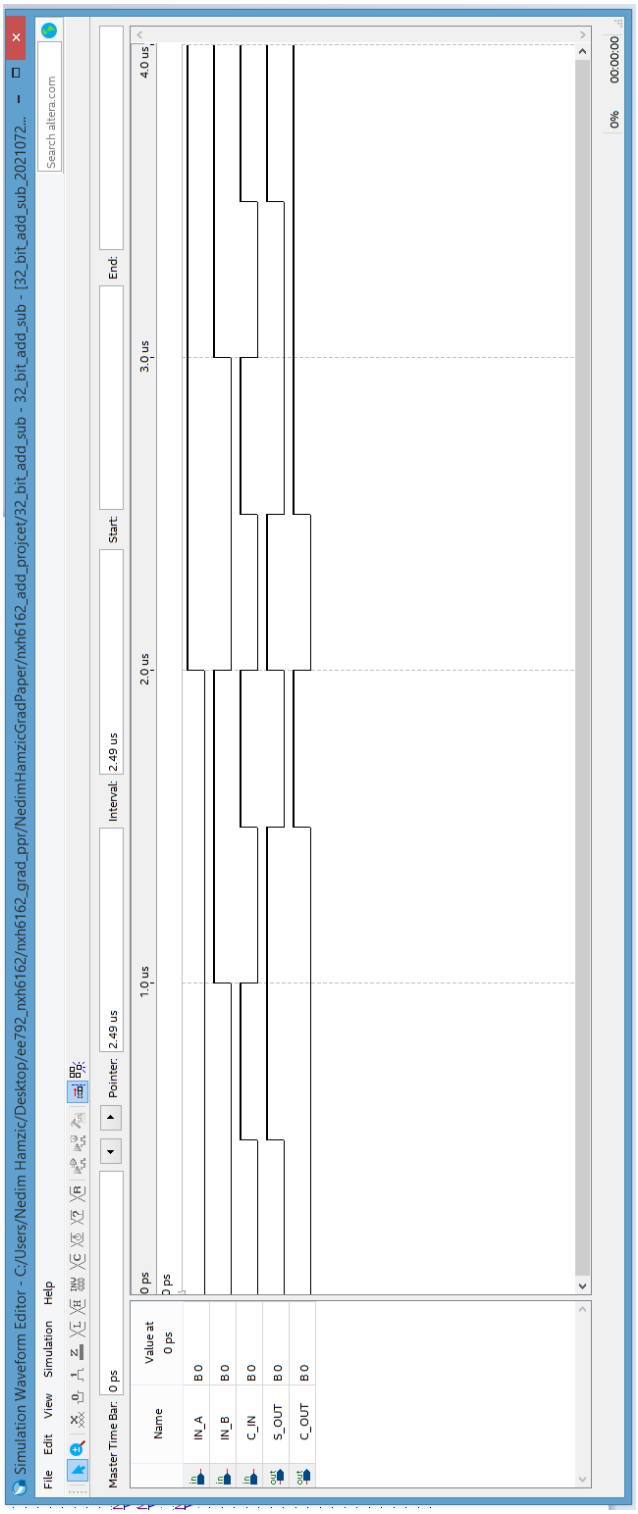

Figure 9.1: Simulation results of single bit adder

Figure 9.2: Simulation results of 32-bit adder/subtracter

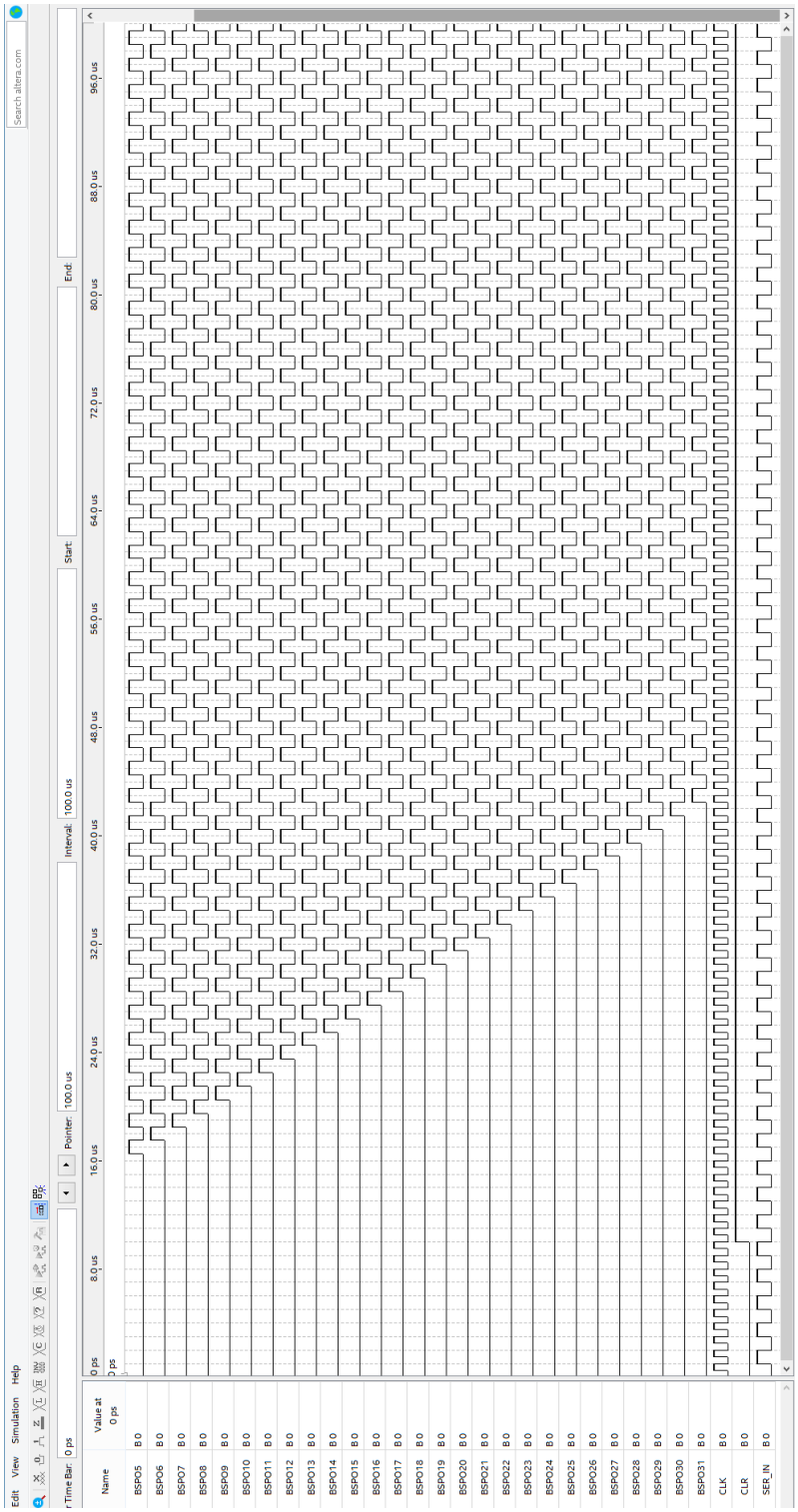

Figure 9.3: Simulation results of serial-in-parallel-out register

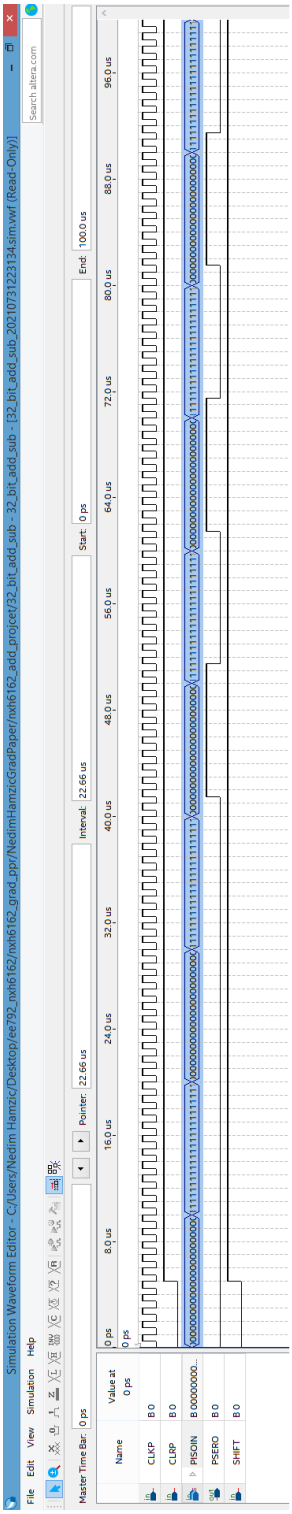

Figure 9.4: Simulation results of parallel-in-serial-out register

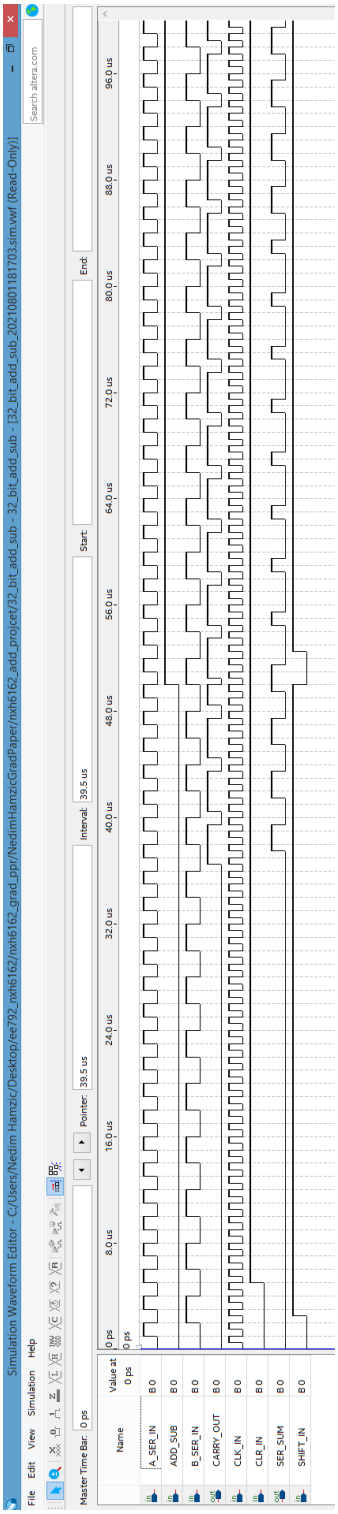

Figure 9.5: Simulation results of final adder subtracter

9.2 Simulation results discussion

Fig 9.2 shows the simulation of the single bit adder and from the figure, the output sum signal designated as S_OUT is producing the same logic values as specified in Table 8.1, and the carry-out signal labeled as C_OUT also produces correct response based on the input signals which are labeled as IN_A, IN_B, and C_IN that go from 0 to 7. Once the simple adder was verified the logic block for full adder was created and 32 of those full adders are used with a combination of XOR gate to create a 32-bit adder/subtracter which simulation waveform response is shown in Fig. 9.2. In Fig. 9.2 the input A and B are alternating at different periods to cover 0 and 1 cases. OPER signal determines if the addition is performed (logic 0 first half of simulation graph) or subtraction (logic 1 second half of simulation graph) and the sum result is designated as S. When OPER signal is low the addition of A and B signals is performed. When A is 32-bit long 0 and B is also the 32-bit long zero the addition is of course 32-bit long zero as indicated in simulation capture. When A is 32-bit long 1 and B is 32-bit long 0 the addition is 32-bit 1 as indicated on capture. When OPER signal is switched to high the subtraction of A from B is executed and the sum is shown to be 0 with leading 1 indicating that 32-bit long 0 (input A) has been subtracted from 32-bit long 1 (input B) and also 32-bit 1 (input A) is subtracted from 32-bit long 1 (input B) resulting in 32-bit long 0 sums.

Fig 9.5 shows the serial-in-parallel out shift register performance and as it can be observed in the figure once CLR has been activated high the data is shifted at each CLK (clock) cycle resulting in classical shift register pattern shown by BSPO0 to BSPO31 . Fig 9.4 shows parallel-in-serial-out shift register and as depicted the data is also being shifted after 32us indicated by signal PSERO and the SHIFT signal being high indicting that operation of parallel-in-serial-out shift register has been achieved. With shift registers verified the final 32-bit adder/subtracter

schematic is created using shift registers on the input and output to accommodate lack of IO lines on AD 2. The simulation capture of final 32-bit adder subtracter is presented in Fig. 9.5. In this figure it can be observed that A_SER_IN and B_SER_IN inputs varies in period, operation addition is done first, and subtraction second as shown by ADD_SUB signal. SHIFT_IN and CLR_IN signals are activated first and clock is fixed frequency as indicated with CLK_IN signal. As the data is applied to input signals the carry out and sum signals are generated as indicated with CARRY_OUT and SER_SUM respectively indicating correct operation of 32-bit adder/subtracter.

9.3 Hardware results and discussion

Fig. 9.6 is captured from WaveForm user interface GUI and its shows the stimulus being provided by pattern file and analyzed by logic file. From capture in Fig. 9.6, it can be observed that carry out and sum signals are being updated as the stimulus to FPGA on the board is provided through the inputs. The same response between this hardware capture and simulation capture is apparent indicating that the 32-bit adder has been programmed successfully but also that the FPGA pin mapping and connection between AD 2 and DE0-Nano board are correct. The Fig 9.7. shows final synthesis record of 32-bit adder and Fig. 9.8 shows FPGA pinout assignment of all necessary input/output signals while Fig 9.9 indicates successful FPGA programming. 4 LED's indicators on the DE0-Nano board are to be turned on as a visual indication that the board has been programmed as Fig. xx shows. Hardware and software capture functional similarities have been confirmed and the process of developing the FPGA code with a schematic approach has been presented and demonstrated with a 32-bit adder/subtracter project using AD2 and FPGA located on the DE0-Nano board.

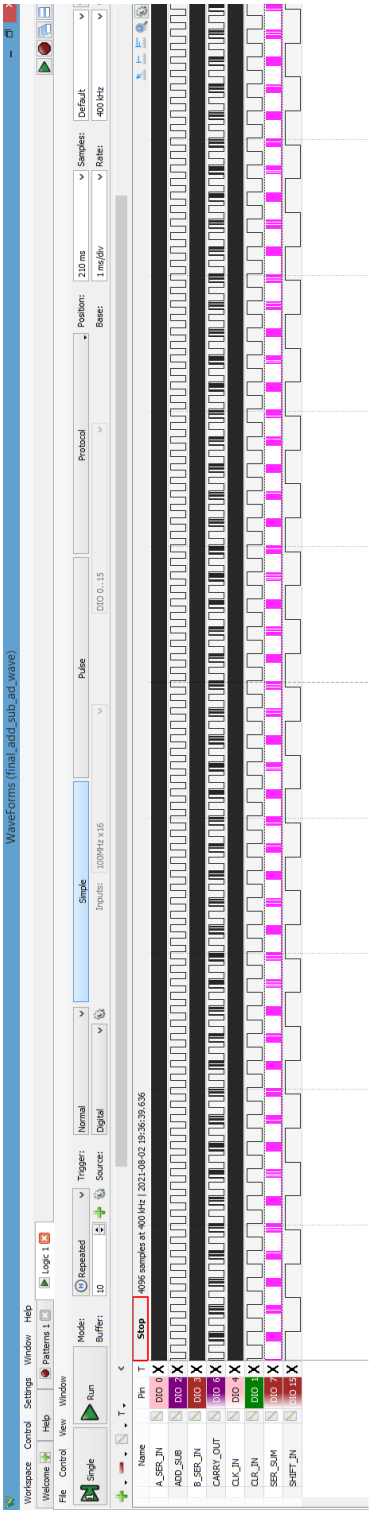

Figure 9.6: Hardware results of 32-bit adder subtracter with AD 2

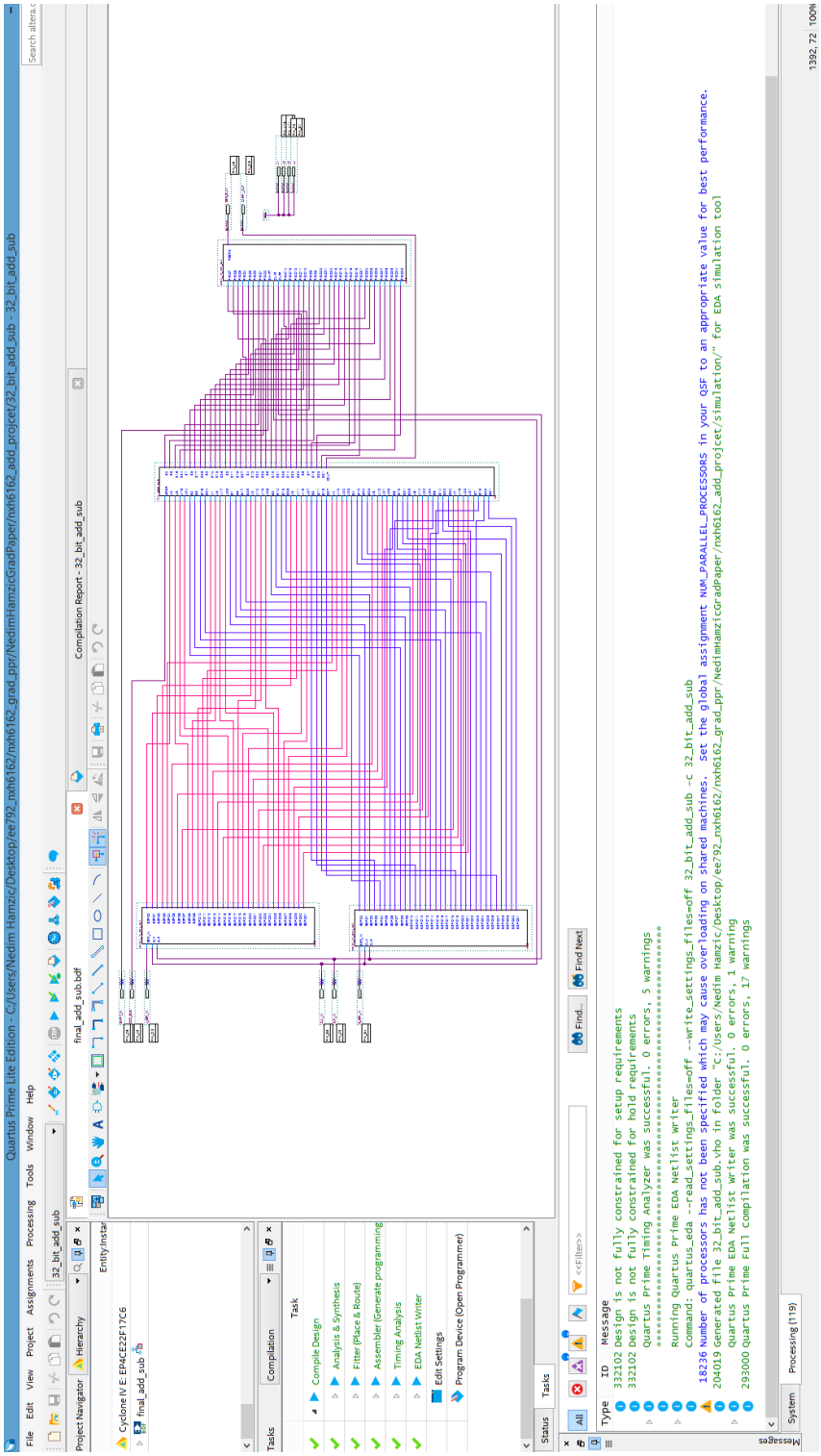

Figure 9.7: Synthesis of 32-bit adder/subtracter

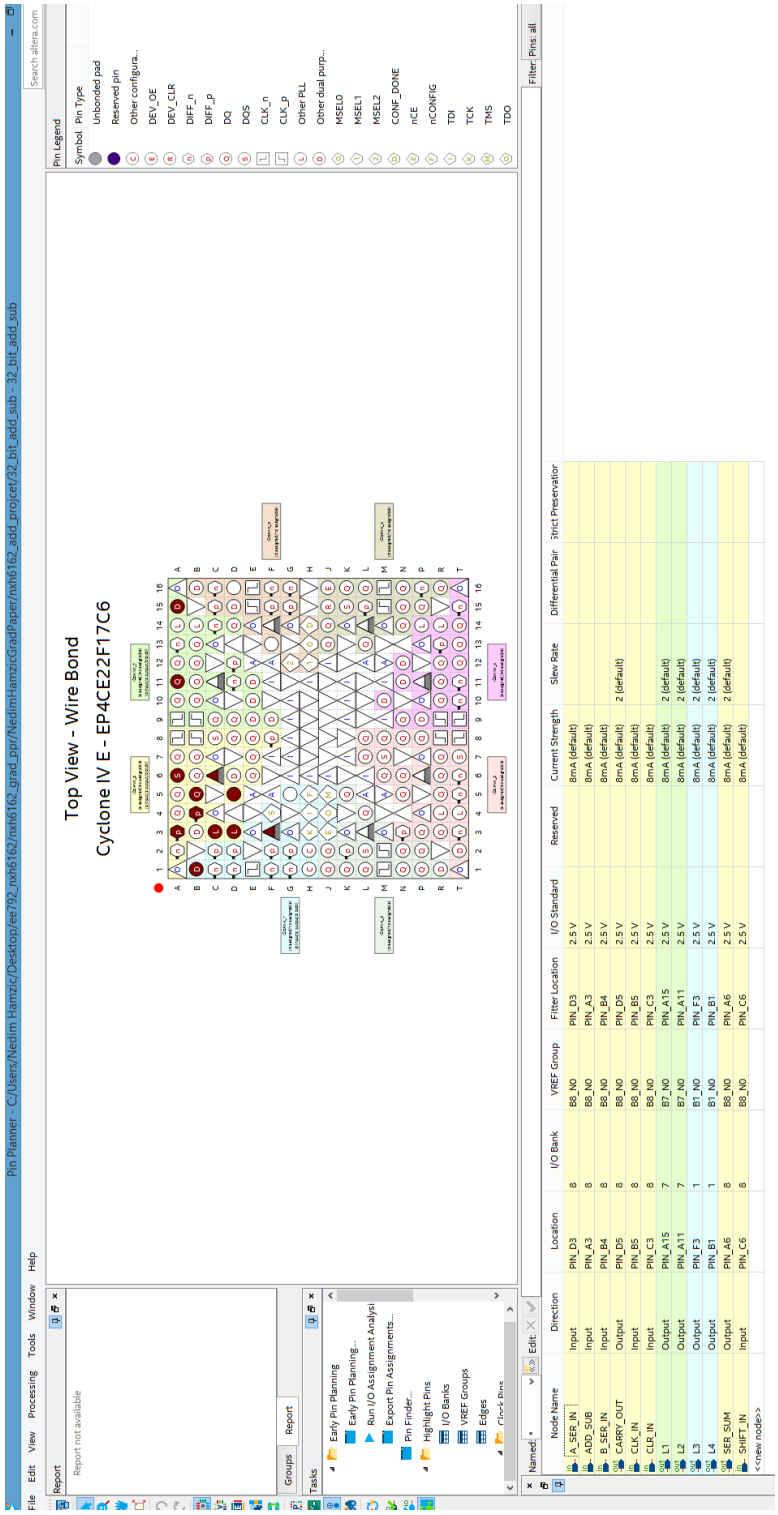

Figure 9.8: FPGA pin planning

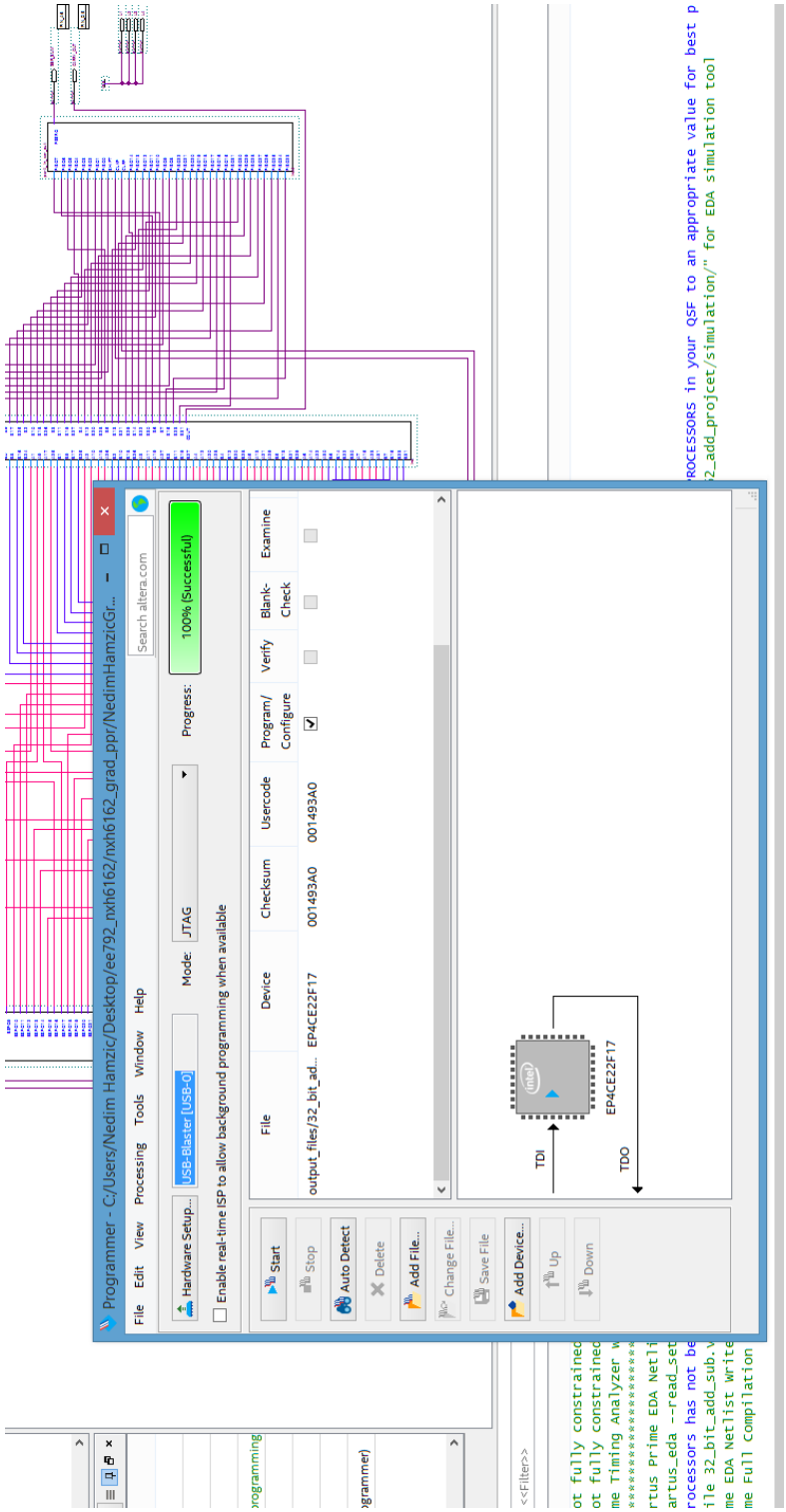

Figure 9.9: FPGA programming

Chapter 10

Conclusion

The evolution of IC and its application in modern-day technology is emphasized as well as the challenges in manufacturing and testing the IC as the transistor technology keeps reducing in size and the number of transistors in IC is reaching billions. Origins of several IC are discussed but close attention is devoted to FPGAs. Detail FPGA architecture is elaborated such as logic blocks, interconnect, and IO pads. Different manufactures of FPGA have a different structure of logic blocks and their interconnects can vary as demonstrated between Xilinx and Altera FPGAs. Several FPGA programming processes are presented such as SRAM, anti-fuse, and floating gate. The benefits of using one programming process as opposed to the other are discussed for example anti-fuse is one-time programming which is not beneficial during the development stage where SRAM can be reprogrammed multiple times making this type of FPGA more desired during the development stage. The software process of designing the FPGA project code is outlined in which design engineer can select to develop VHDL or schematic FPGA project and in some cases, the combination of schematic and VHDL approaches can be deployed during the design. Synthesis of FPGA design as well as mapping the FPGA pins are elaborated as critical steps need to be implemented to program FPGA tar-

geted device. Impotence and architecture of boundary-scan for troubleshooting is emphasized especially for the FPGAs that are soldered onboard and have BGA pin arrangement.

To eliminate the need for expensive test equipment and deliver the IC product on a competitive market with high reliability and fault coverage as well as in a short amount of time results in the development of BIST architecture. The cost of the IC is tied directly to the testing and for some manufacturing testing of the part is associated with its cost and for some manufactures, 30-40% of the part cost is related to the amount of testing. Components necessary to generate BIST architecture are presented such as pattern generator and output response analyzer. BIST consists of applying the test vector pattern through the pattern generator to the circuit under test and capturing the output response signature and comparing that signature to the good working circuit. Application of BIST architecture is discussed such as testing of analog to digital converts, RF oscillator, and MEMS designs. Cases of deploying the BIST to board and system levels are elaborated. Modeling of BIST using a mathematical and statistical approach such as Stateflow is presented. Benefits and drawbacks of deploying the BIST on the IC manufacturing level are discussed and some of the benefits are the elimination of external test equipment and faster test time while the drawbacks include false failure, power consumption, chip area impact, and manufacturing yield impact. Due to the shrinking transistor technology and the number of transistors and logic blocks increasing in design with the addition of more specific block applications onto IC, the BIST application is becoming more attractive for manufactures but still, resources need to be allocated to optimize BIST architecture due to drawbacks presented.

A 32-bit adder/subtracter project has been presented to solidify the FPGA project generation from a schematic design standpoint. To demonstrate BIST the DE0-Nano board to-

gether with Analog Discovery 2 is used to verify that 32-bit adder/subtracter performs in hardware. The adder/subtracter project has been created incrementally starting with a simple full adder for which block was created and that block is used to create a 32-bit adder /subtracter schematic. To verify in hardware that 32-bit adder/subtracter is operating as the designed external stimulus from AD 2 is used. Since AD 2 is only limited to 16 IOs serial shift registers have been created to address the lack of necessary pins. With the shift, registers designed the adder/subtracter project has been successfully synthesized in Quartus II and programmed into DE0-Nano FPGA using USB Blaster. The interconnect and FPGA pin mapping are also documented. Using the WaveForm software, the stimulus to FPGA via AD 2 is provided and the response from FPGA is collected and compared with simulation results. The research goals listed in chapter 1 have been achieved, but FPGA and BIST designs are very broad subjects however fundamentals of FPGA architecture and BIST implementation have been presented as groundwork for future research.

10.1 Future work

The fundamental architecture of FPGA has been presented but modern FPGA has several other applications worth studying and understanding such as DSP blocks, analog to digital converters, USB, CAN as well as other communication protocols. Interconnect of logic blocks in FPGA for optimization in terms of time delay is another aspect of future work. How does the interconnect impact the time delay and how time delay can be minimized in critical timing circuits? With parallelism capabilities of FPGA, the application of FPGA in artificial intelligence is another topic of investigation to determine the impact of FPGA in the AI industry. The investigation has shown that despite the benefits of BIST there are also several drawbacks such as chip area impact, power consumption, reliability, and yield so investigation in improv-

ing the BIST is also a topic for future work that would target specified drawbacks to make BIST more attractive for manufactures of ICs. How do BIST pattern generation and test vector impact the longevity of the BIST and how those patterns can be optimized to provide 100% fault coverage without significant test time? How the BIST signature output can be reduced so that it does not require significant memory addition? The project presented in the paper is developed by using the schematic design approach, but it would be beneficial in generating the project using the VHDL approach to crate blocks and test benches. Research indicated that during the BIST the output analyzer is comparing the response to a good working circuit but how is that good working response generated? Is it done by using external test equipment or is it based on simulation is something worth investigating as future work? Developing the JTAG boundary scan on the DE0-Nano board would be an interesting project to investigate so that understanding, and benefits of boundary scan can be obtained.

References

[1] Jonathan Rose, Abbas El Gamal, and Alberto Sangiovanni-Vincentelli. Architecture of Field-Programmable Gate Arrays. In *Proceedings of the IEEE*, 1993. doi:10.1109/5.231340.

[2] Eric Monmasson, Lahoucine Idkhajine, Marcian N. Cirstea, Imene Bahri, Alin Tisan, and Mohamed Wissem Naouar. FPGAs in Industrial Control Applications. *IEEE Transactions on Industrial Informatics*, 2011.

[3] Aiwu Ruan, Shi Kang, Yu Wang, Xiao Han, Zujian Zhu, Yongbo Liao, and Peng Li. A Built-In Self-Test (BIST) system with non-intrusive TPG and ORA for FPGA test and diagnosis. *Elsevier Ltd.*, 2012.

[4] Jacob Savir and Paul H. Bardell. Built-In Self-Test: Milestones and Challenges. *Gordon and Breach Science Publishers*, 1993.

[5] Ioannis Voyiatzis and Constantin Halatsis. A Low-Cost Concurrent BIST Scheme for Increased Dependability. *IEEE Transactions on Dependable and Secure Computing*, 2005.

[6] Johnny J. LeBlanc. LOCST A Built-In self-Test Technique. *IEEE Design & Test of Computers*, 1984. doi:10.1109/MDT.1984.5005689.

[7] G. Naveen Balaji and S. Chenthur Pandian. Design of test pattern generator (TPG) by an optimized low power design for testability (DFT) for scan BIST circuits using transmission gates. *Springer Science+Business Media*, 2018.

[8] Sunil R. Das, Jila Zakizadeh, Satyendra Biswas, Mansour H. Assaf, Amiya R. Nayak, Emil M. Petriu, Wen-Ben Jone, and Mehmet Sahinoglu. Testing Analog and Mixed-Signal Circuits With Built-In Hardware A New Approach. *IEEE Transactions on Instrumentation and Measurement*, 2007.

[9] Manibha Sharma and Jasdeep Dhanoa. Smart Logic Built In Self-Test In SOC. *IEEE International Conference on Recent Advances and Innovations in Engineering*, 2020.

[10] Voicu Groza, Rami Abielmona, Mansour H. Assaf, Mohammed Elbadri, Mohammad El-Kadri, and Arkan Khalaf. A Self-Reconfigurable Platform forBuilt-In Self-Test Applications. *IEEE Transactions on Instrumentation and Measurement*, 2007.

[11] Michael A. Lusco, Justin L. Dailey, and Charles E. Stroud. Built-In Self-Test for Multipliers in Altera Cyclone II Field Programmable Gate Arrays. *IEEE Southeast Symp. on System Theory*, 2011.

[12] W.-D. Tseng, L.-J. Lee, and R.-B. Lin. Deterministic built-in self-test using multiple linear feedback shift registers for test power and test volume reduction. *IET Computers & Digital Techniques*, 2009.

[13] M. Dodiya Chandni and V. Ravi. Built in Self Test Architecture using Concurrent Approach. *Indian Journal of Science and Technology*, 2016.

[14] Nachiketa Das, Pranab Roy, and Hafizur Rahaman. Built-in-self-test technique for diagnosis of delay faults in cluster-based field programmable gate arrays. *IET Computers & Digital Techniques*, 2013.

[15] Mary D. Pulukuri and Charles E. Stroud. Built-In Self-Test of Digital Signal Processors in Virtex-4 FPGAs. *IEEE Transactions on Instrumentation and Measurement*, 2009.

[16] Mary D. Pulukuri and Charles E. Stroud. On Built-In Self-Test for Adders. *Springer Science + Business Media*, 2009.

[17] Chun-Lung Hsu and Ting-Hsuan Chen. Built-in Self-Test Design for Fault Detection and Fault Diagnosis in SRAM-Based FPGA. *IEEE Transactions on Instrumentation and Measurement*, 2009.

[18] Dimitris Gizopoulos, Mihalis Psarakis, Miltiadis Hatzimihail, Michail Maniatakos, Antonis Paschalis, Anand Raghunathan, and Srivaths Ravi. Systematic Software-Based Self-Test for Pipelined Processors. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2008.

[19] Junshi Wang, Masoumeh Ebrahimi, Letian Huang, Xuan Xie, Qiang Li, Qiang Li, and Axel Jantsch. Efficient Design-for-Test Approach for Networks-on-Chip. *IEEE Transactions on Computers*, 2019.

[20] Kiran George and Chien-In Henry Chen. Logic Built-In Self-Test for Core-Based Designs on System-on-a-Chip. *IEEE Transactions on Instrumentation and Measurement*, 2009.

[21] Junyou Shi, Qingjie He, and Zili Wang. Integrated Stateflow-based simulation modelling and testability evaluation for electronic built-in-test (BIT) systems. *Elsevier Ltd.*, 2019.

[22] Hamidreza Hashempour, Fred Jackie Meyer, and Fabrizio Lombardi. Analysis and Measurement of Fault Coverage in a Combined ATE and BIST Environment. *IEEE Transactions on Instrumentation and Measurement*, 2004.

[23] Chin-Long Wey. Built-In Self-Test Design of Current-Mode Algorithmic Analog-to-Digital Converters. *IEEE Transactions on Instrumentation and Measurement*, 1997.

[24] M. Senthil Sivakumar and S. P. Joy Vasantha Rani. Design of digital Built-In Self-Test for Analog to Digital Converter. *IEEE Transactions on Instrumentation and Measurement*, 2008.

[25] M. Shafiee, N. Beohar, P. Bakliwal, S. Roy, D. Mandal, B. Bakkaloglu, and S. Ozev. A Disturbance-Free Built-In Self-Test and Diagnosis Technique for DC-DC Converters. *ACM Transactions on Design Automation of Electronic Systems*, 2017.

[26] Anitha Balasubramanian, B. L. Bhuva, L. W. Massengill, B. Narasimham, R. L. Shuler, T. D. Loveless, and W. Timothy Holman. A Built-In Self-Test (BIST) Technique for Single-Event Testing in Digital Circuits. *IEEE Transactions on Nuclear Science*, 2008.

[27] Xingguo Xiong, Yu-Liang (David) Wu, and Wen-Ben Jone. A Dual-Mode Built-in Self-Test Technique for Capacitive MEMS Devices. *IEEE Transactions on Instrumentation and Measurement*, 2005.

[28] Jee-Youl Ryu, Bruce C. Kim, and Iboun Sylla. A New Low-Cost RF Built-In Self-Test Measurement for System-on-Chip Transceivers. *IEEE Transactions on Instrumentation and Measurement*, 2006.

[29] Chen Xin, Wu Ning, Bai Na, Huang Hui, and Hu Wei. Built-in self test design of power switch with clock-gated charge/discharge transistor. *IET Computers & Digital Techniques*, 2013.

[30] Nan Li, Gunnar Carlsson, Elena Dubrova, and Kim Petersen. Logic BIST: State-of-the-Art and Open Problems. *IEEE*, 2015.

[31] Graham Hetherington, Tony Fryars, Nagesh Tamarapalli, Mark Kassab, Abu Hassan, and Janusz Rajski. Logic BIST for Large Industrial Designs: Real Issues and Case Studies. *IEEE*, 1999.

[32] Y. Nakamura, J. Savir, and H. Fujiwara. BIST Pretest of ICs: Risks and Benefits. In *24th IEEE VLSI Test Symposium*, 2006. doi:10.1109/VTS.2006.23.

Appendix I

Research Notes

This appendix contains the notes collected from all research references used and they highlight critical points presented in each paper. The title of each reference paper is in bold

I.1 Reference paper 1

A Built-In Self-Test (BIST) system with non-intrusive TPG and ORA for FPGA test and diagnosis: Presented the test pattern generator and output response analyzer as the integral components of BITS which are in software and circuit under test which is hardware. Origins of BITS are discussed from the 1980s and a basic idea of BITS test is a circuit that is capable of testing itself but TPG and ORA are required as part of that test. Two types of BITS discussed are an intrusive and non-intrusive test also embedded and non-embedded BITS test is also mentioned. A major point of embedded BITS testing is disappearing when FPGA or CUT is being configured for its normal operation. BITS physically consist of software and hardware part with communication channels. Embedded is equal to intrusive as shown in the paper. Intrusive or embedded BITS are applied to FPGAs, but they do have limitations such as many

configurations which complicate the procedures and increases time cost. TPGs and ORAs are related to the complexity of FPGAs and often exchanged due to that complexity which is the downfall of this approach. A detailed description of proposed architecture and communication between hardware and software via implemented channels. Three steps approach data send from the software side through the channel to CUT and the data send from CUT which is hardware to software side to be analyzed for faults. Critical synchronization between hardware and software part elaborated. Detail picture of the test with hardware and software is shown and implementation of the BITS test with bulletin points outlining information exchange during CUT testing and test implementation. The type of FPGA faults such as stuck-at fault and functional faults are also discussed. Pseudocode for testing FPGA input/output IOB blocks as well as CLB block and finally IR block is shown in form if-else statement with some do and for loops. Lots of FPGA's have similar look-up tables and flip-flops which are part of control logic blocks, so the configuration of the test is similar. CLB of FPGA has two outputs as row and column as the coordinates of CLB that can be found if the fault is detected. Test of various FPGA's discussed and follow with the conclusion that states that non-intrusive test is more beneficial than intrusive test in the sense that reduces the number of configurations and can be easily modified for various FPGAs.

I.2 Reference paper 2

Integrated Stateflow-based simulation modelling and testability evaluation for electronic built-in-test (BIT) systems: Based on the Stateflow approach which is a powerful graphical tool model paper proposes the BIT test that is integrated and simulation-based method which highlights three methods which are the dynamic release of fault transmission and BIT monitoring process under test, diagnostics information through fault injection, and calculation BIT

performance metrics. Simulation method of BIT evaluation but it is still compatible with real hardware experiments. Discusses various mathematical models for BIT test and analysis and their drawbacks but emphasizes and superiority of the Stateflow mathematical model. BIT simulation modeling implemented for reliability and performance. Several modeling methods are mentioned ranging from mathematical to multi-signal flow model but the Stateflow method has the advantage of all other models proposed in the paper. The architecture of Stateflow is explained in one of the figures that show Stateflow objects, BIT modeling elements, and simulation model integration. Stateflow model simulation is integrating into MATLAB and Simulink and they can be used to create a Stateflow model. Various Stateflow models are elaborated such as fault model, BIT fault detection model which has simple threshold comparison and bus check bit, isolation model, BIT false alarm reduction model which consist of repetitive test BIT and voting BIT test and finally interface model which refers to repeated and random interface on BIT test modeling. How accumulated information from the model is processed and how various metrics from the Stateflow model are calculated through D-matrix generator and those are FDR, FIR, and FAP formulas for each are defined and elaborated in the paper. The Stateflow model is specifically outlined in figure 10 of the paper. The practical circuit in this power conversion board is modeled based on the Stateflow approach and the model and analytical BIT test are compared, and it was determined that the Stateflow model is very much consistent with the analytical approach.

I.3 Reference paper 3

Built-In Self-Test: Milestones and Challenges: Testing is nonlinearly related to the complexity of the circuit and therefore more sophisticated BIT tests are required to ensure the quality of the product since the quality of the product is related to the sophistication of the

testing approach. Add the additional circuits on the chip that can process and compress data to be compared with the expected response. HP 5004A analyzer introduced the idea of adding a compression circuit or signature analyzer to IC so that data coming out of the BIT test can be analyzed. Other signature analyzers were introduced such as one's counting but not popular at that time as signature analysis. Stimulus to CUT was allowed an issue and exhaustive and non-exhaustive methods are introduced but depending on the complexity of the circuit being tested there were not efficient. General early BIT structure is presented and elaborated with data compressor and comparator that compares reference model to the signature generated by compressor and obviously if compression signature is equivalent to reference model then BIT passes but reference model is generated form good known working operational circuit. Explanation on Built-in Logic Block observe is elaborated as compression or signature mechanism on IC. STUMPS for complex IC architecture but the negative effect is wiring needs to be added on IC to make all connections. Various input generators are discussed and their dependency in the BIT test. Probability and simulation mathematical approaches in computing the faults and confidence in testing are elaborated as well. Recommended computer aid tools for future use to do BIT test generation and analysis.

I.4 Reference paper 4

Built-In Self-Test Design of Current-Mode Algorithmic Analog-to-Digital Converters: Adding more devices on a single chip increases complexity, testability, and reliability which increases testing and cost as well as time but to reduce cost and time built-in test provides significant advantages. Proposes of adding BIT test to analog to digital converter to reduce offline testing with simple overhead and few additional pins. Explains the working circuitry of A/D converter with switching mechanism and the possibility of that switching mechanism

having one or more defective switches. 3 faults in the converter can occur which are types 1, 2, and 3. Type 1 produces strings of 0s or strings of 1s on the output if certain switches are defective. Type 2 produces a nonzero bit when the input current is zero. Type 3 has at least one in the output due to a set of switches not working in the converter. The proposed AD BIT design deploys two extra pins which are test enables and error indicators with 4 major parts test generator, a converter circuit, output comparator, and timing circuity. The Paper explains the proposed design operation with the BIT test implemented and states that this design can be added to any switch capacitor and switch current techniques.

I.5 Reference paper 5

Analysis and Measurement of Fault Coverage in a Combined ATE and BIST Environment: To have high confidence in fault coverage the BIST and ATE tests are used with random vectors to ensure high coverage. Test vectors are an issue for the BIST test from a quality perspective. Possibility of combing BIST test with automatic test equipment to ensure good coverage in a short amount of time which creates a hybrid model of testing integrated circuits. Fault coverage is modeled as a function of the number of both BIST and ATE vectors. Test vectors depend on the circuit being tested. Paper develops a model where BIST uses random test vectors to achieve certain fault coverage and then they switch to ATE that uses deterministic vectors to achieve the final circuit under test coverage. Probabilities calculation when combining and not combining BIST and ATE coverage and switchover time estimation is elaborated. Mathematical models and equations are presented in computing the fault coverage. The model has been evaluated and results are shown on how the combination of BIST and ATE can reduce time and cost when it comes to manufacturing chips.

I.6 Reference paper 6

A Low-Cost Concurrent BIST Scheme for Increased Dependability: BITS technique to reduce test latency for VLSI by introducing a new input vector monitoring routine to reduce specific concurrent and effective concurrent latency. Normal and test modes are elaborated based on figure 1 for input vector monitoring and new window-monitoring concurrent BIST (w-MCBIST) is proposed. The test set is divided into non-overlapping subsets called windows. Proposing w-MCBIST shows that latency of the test relative to overhead hardware needs to implement such architecture. Gate reduction which is hardware and test duration is also shown compared to the other BITS schemes that authors are presenting in the paper. The authors applied their proposed model to simulation and discovered that their proposed method requires less hardware overhead while reducing test time comparing to other BIST techniques presented in the paper. Various equations are shown that prove how effective the proposed model is in terms of efficiency compared to other BIST models presented in the paper.

I.7 Reference paper 7

A Dual-Mode Built-in Self-Test Technique for Capacitive MEMS Devices: BIST but for MEMS system is presented. Some MEMS information is shown with capacitance and plates moving to vary capacitance as they are exposed to voltage. An interesting symmetric BIST scheme of adding additional symmetric plates is added to show how BIST is added and that two plates, when exposed to the same stimulus, should have the same capacitance, and if not, then something is defective. Authors now are proposing a dual-mode BIST technique with more advantages over symmetric BIST, but they combine it symmetrically and sense BIST together as the proposed design. The control circuit is added in MEMS BIST to dictate when the device is in a normal or self-test mode that would detect defects if one of the plates is missing due

to the etching process. Proposed dual BIST is applied to three types of MEMS devices which are surface micromachined comb accelerometer, bulk-micromachined capacitive acetometer, and poly-Si surface micromachined microresonator. All there are explained with proposed BIST and how they detect defects for all 3 technologies by partitioning it into three blocks. Simulation results of all the mentioned technologies are subjected to BIST circuit and defects are detected by the proposed BIST according to simulated results. The benefit of having BIST is that can not only be deployed during factory testing and calibration but in the field as well. Since MEMS systems are widely used the authors not only emphasize BIST but also BISR which is a built-in self-repair test which in nutshell is more MEM system added for redundancy purposes.

I.8 Reference paper 8

A New Low-Cost RF Built-In Self-Test Measurement for System-on-Chip Transceivers: SoC RF transceiver that has BIST feature is presented in the paper that requires minimum external equipment but BIST covers fundamental functionality measurements of the transceiver by providing voltages to be interpreted by the onboard processor through the use of mathematical equations elaborated in the paper. They use impedance to relate to voltages so that they can detect the faults and plots of catastrophic and parametric variations are plotted. They build the circuit and showed that BIST that was added is very compatible with simulation as well as standard external RF measuring equipment.

I.9 Reference paper 9

A Disturbance-Free Built-In Self-Test and Diagnosis Technique for DC-DC Converters: BIST for DC-DC converter loop by injecting noise and having BIST correct converter and optimize its performance. The stability and dynamic performance of such a converter depend on the loop characteristic of the converter. When deploying the self-test, it needs to be transparent and not affect the normal operation of the circuit. Parametric method for system analysis is model developed bases on the observed behavior of the system. The non-parametric method is a method where the stimulus is provided to inputs of the system and its outputs are monitored for a response. Method to optimize compensation of DC converter based on its degradation due to temperature or aging. The proposed test consists of obtaining impulse response by injecting pseudo-random noise into the converter and characterize the closed-loop transfer function to determine pols and zeros for stability purposes and to adjust the converter to be more stable in the field. Possibility of combining converter with FPGA or DSP that will perform response of DC converter with injection noise to obtain transfer function. The experiment was conducted where the converter was injected with noise and capacitor and inductor Q parameters were obtained. This method can indicate and show degradation, but it can also be used to compensate the converter to be stable despite thermal or any other kinds of noises that can be generated from the system.

I.10 Reference paper 10

Testing Analog and Mixed-Signal Circuits With Built-In Hardwar A New Approach: The yield of SOC is related to cost so the emphasis is to increase yield by ensuring good coverage through testing to reduce the cost of the part for aggressive market competition. Oscillation built-in self-test discussed either added hardware on SOC but this test does not require stim-

ulus generator and analyzer. Testing the analog circuits is much farther behind than testing the digital circuit. Fault coverage is the percentage ratio of the number of faults detected to the total number of possible circuit faults. Design for test o DTF nodes that allows control and observability for verification. BIST is to test the circuity on the board instead of sending analog signals through long wires to be measured where those long wires can cause degradation of the analog signal. BIST reduces the need for additional expensive test equipment, therefore, reducing the cost of testing the IC. A sample of a simple bit test graph is shown on paper. OBIST is simple and requires no extensive modification to CUT. Three techniques of how oscillation (switch Op-Amp) is created and response compared to fault-free circuitry. Hard faults are usually discovered due to connectivity issues and soft faults affect performance degradation from their nominal operating limit. The analog circuit needs to be converted to an oscillator by adding some additional reactors and capacitors to the CUT. In the paper, only hard or catastrophic faults are considered and simulation on various analog circuits such as FET, notch filter op-amp, and others to demonstrate that analog circuits can be tested using oscillation circuit addition and that major hard faults are detectable.

I.11 Reference paper 11

A Self-Reconfigurable Platform for Built-In Self-Test Applications: Using the runtime reconfigurable techniques to minimize the circuit area as well as test generation and application time. CUT dynamic partial reconfiguration to detect stuck on faults by injecting circuits with faults to detect faults and to find any hard-to-reach faults. The device's durability correlates with the time spent on developing the test. Sequential hardware implementation partially parallels hardware implementations, and a hybrid of both BIST techniques is presented in the paper. The first test strategy discussed is software realization which requires no hardware just pro-

duces benchmark simulation results. The second strategy is the CTR realization strategy where CUT is synthesized and mapped on the target device with fault injection multiplexers which allows for sequential pattern injection and observing the circuit behavior for each pattern. The third is partially parallel realization and the last one is sequential RTR realization. A lot of detailed FPGA testing was discussed from a hardware and software perspective using parallel and sequential pattern injection and parallel testing were much faster than the simulation.

I.12 Reference paper 12

Systematic Software-Based Self-Test for Pipelined Processors: Doing the software BIT internally on the processor is advantageous in terms of reliability and elimination of external expensive testers. Process of modifying BIT test to test pipelines in processor besides other circuity and that proposed SBST provides much more improvement from fault coverage perspective comparing to conventional test. The shortcomings of having a built-in self-test is that does not entirely capture environmental variables such as voltage, noise, or temperature. Model-based on software improves stuck on faults coverage as well as propagation delay coverages. Testability problems 1 and 2 are discussed on how lack of testing on processor pipelines is affecting overall processor coverage. The example is shown using the assembly language. The proposed pipeline test is self-contained and configurable but does not require gate and RTL information just generic processor information to generate a test process that has two phases and phase 1 extracts several pipeline stages and phase 2 that performs self-test memory partitioning. The table shows that the proposed test does enhance the coverage of the pipeline as presented in table 3 of the paper and that this proposed test can be very generic and applicable to any pipelined processor.

I.13 Reference paper 13

A Built-In Self-Test (BIST) Technique for Single-Event Testing in Digital Circuits: Paper introduces a relatively inexpensive method to test circuit components due to transient occurrence which can damage the circuit resulting in latent defects. The proposed method eliminates the use of laser and bulky expensive external equipment. The proposal is to inject pulse into circuit nodes that are very in amplitude and arrival relative to the clock. Several circuits are presented on how to control and measure pulse width that is applied to circuit nodes. The nodes on which pulse is applied are selected since is not practical to apply pulse on nodes in more complex designs but with careful node selections from design, the large coverage can be obtained as the authors have shown in their adder circuit example.

I.14 Reference paper 14

Built-In Self-Test of Digital Signal Processors in Virtex-4 FPGAs: Developing the test by using already existing FPGA blocks to construct generators and analyzers that are critical in self-built test architecture. This paper proposes the BIT test to test DSPs on the FPGA. DSP architecture of FPGA is shown in detail with multiplier and adder/subtracter. Description of C program automatic tests procedures on FPGA programming blocks and measurements obtained are elaborated and tabulated. Five BIST are developed to test various FPGAs with very good coverage and test time reduction. This is another example of BIST using software to modified logic blocks on FPGA to self-test and detects faults in DSP cores.

I.15 Reference paper 15

Built-In Self-Test for Multipliers in Altera Cyclone II Field Programmable Gate Arrays: Also, elements of self-test specified as an integral part of every BIST are CUT, a test pattern generator, and an output response analyzer. Paper proposes 3 VHDL programs that will configure FPGA to self-test. The circular BIT test figure has shown that each CUT is tested circularly. ORA contains a pattern configuration that was generated from fault-free CUT and will be compared to the response generated from the DUT circuit. VHDL code is used to develop the test regardless of architecture and VHDL code does not require any customization to test different devices. Used Quartus 2 and eval board to verify some of Alter FPGS with VHDL code that would test multipliers and adders in FPGA.

I.16 Reference paper 16

Built-in-self-test technique for diagnosis of delay faults in cluster-based field programmable gate arrays: BIST for DPGA to detect delay faults in FPGAs configurable logic blocks such as DSP, multipliers, adders, and interconnects, and this BIST delay is much more superior than previous BIST that require no external testing. 80% of the FPGA area is dedicated to interconnects between configuration logic blocks. Various references elaborated on FPGA BIST and how those BIST omit delay tests that can be the result of interconnections. Proposed BIST for look-up table timing addresses the drawbacks of faulty flip-flops and clock delay as mentioned in previous references. Transition faults and timing faults models discussed and slow to rise and slow to fall faults elaborated. Testing look-up tables delay by adding the D flip-flop between cascaded LUT also following FPGA delay blocks delay analysis are presented in the paper: multiplexers, flip-flops, arithmetic logic multipliers, and DSP blocks. The ORA output response analyzer read through boundary-scan. Blocks in FPGA can be configured as a test

pattern generator and output analyzer. The proposed BIST test every block in 2x3 example and table documents all the cases proving that every block is tested and covered by the new proposed design. Simulation of results shows better coverage than previous proposed BIST regarding the references. Design for testability involves a scan chain.

I.17 Reference paper 17

Design of test pattern generator (TPG) by an optimized low power design for testability (DFT) for scan BIST circuits using transmission gates: Time and power reduction of BIST are the focus of this paper. Test scan is comprised of flip-flops that during the test are switching and capturing data and they are chained together which can consume power during the bit testing resulting in hot spots and circuit degradation. Formulas presented that calculate power consumption based on several switching components present in the scan chain. The tradeoff between several test vectors as well as the time needed for testing and power consumption during BIST. Multiplexers that determine test or normal mode of operation. Good figure that shows the design for test D flip-flop. Using the transmission gates to reduce power when constructing BIST circuit components.

I.18 Reference paper 18

On Built-In Self-Test for Adders: Finding the test algorithm independent of architecture to ensure high coverage of adder. Carry look-ahead adder elaborated. Adding flip-flops instead of inverters for better test coverage. The figure of adders shown but also the DSP of FPGA is elaborated and complexity of testing the DSP and its adder as well as multiplier.

I.19 Reference paper 19

Deterministic built-in self-test using multiple linear feedback shift registers for test power and test volume reduction: Another paper explains how the volume of data is important in testing but that requires more time and affects the power consumption so the idea is to add compress test vectors requiring less shifting and with that less time and power consumption. Various methods are proposed to improve the efficiency of BIST which is based on linear feedback shift register and decompression method to reduce the size of test vectors. Methods of compressing scan data vectors such as MTF transform, NB-XOR transform. Decompression of scanned data is also elaborated with test results that show power reduction by implementing multiple LFSR comparing to a single conventional BIST LFSR.

I.20 Reference paper 20

Built in Self Test Architecture using Concurrent Approach: Elaborates on BIST of the ALU in offline mode to reduce the time of the test as well as the latency of the test. Online mode testing is testing that is done while CUT is powered on and working condition while offline testing is done when CUT is powered off and this type of testing takes much longer to execute. The concurrent approach proposes that test vectors are applied to the inputs of CUT during the normal CUT operation. Has good figure of basic BIST operation. The approach presented in this paper uses a test pattern generator externally to reduce the time of test and overhead.

I.21 Reference paper 21

Built-in self test design of power switch with clock-gated charge/discharge transistor: Describes testing of header footer switches in SOC that are in charge of turning on and off or providing power to the blocks on SOC. Test response is irrelevant of test frequency which does not require external expensive test generators and discharge transistor is gated by the clock which decreases the number of test patterns, test vectors, and test time. A detailed description of BIST on how to test footer/header switches. The structure of one BIST to test header/footer switch is also demonstrated on several switch applications which can be found on complex SOC. Also, an application that needs to be added just to test the footer/header switch on SOC which shows how every aspect is tested.

I.22 Reference paper 22

Logic Built-In Self-Test for Core-Based Designs on System-on-a-Chip: Use in built-in test form of the linear shift register to speed up the process of testing IP cores and getting them on the market fast. Accessing BIST test is done by use of boundary‘ scan also there is the IEEE standard for testing IP cores by use of isolation and test vector application. BIST offers more advantages than scan-based tests on core-based chips because scan-based tests are not efficient in detecting delay faults while BIST is and also using BIST for delay faults is more cost-effective. Mentions several BIST techniques. The linear feedback shift register is also mentioned in this paper as an integral part of the BIST. The proposed BIST NEXT 2-D LFSR shows significant gate reduction which results in less area and hardware verification. Using transistors to generate test patterns that would optimize coverage.

I.23 Reference paper 23

Built-in Self-Test Design for Fault Detection and Fault Diagnosis in SRAM-Based FPGA: Testing the FPGA using the BIST that requires no extra hardware because logic cells are used to construct test pattern generator and output response analyzer and several coverage faults are detected such as open/short and delay faults in the wire channels, stuck on/off faults in PSs, and stuck-at-0/1 faults in LUTs. FPGAs are generally classified into two major types. One is one-time programmable FPGAs such as an anti-fuse type. The other is unlimited reprogrammable FPGAs such as a static RAM (SRAM)-based FPGA. Packaging development of VLSI affects the controllability and observability of internal IC nodes. FPGA is very much configurable, and testing must ensure that all configurations are covered. BIST to test both interconnects and cell logic blocks. One part of FPGA to develop test pattern and to be applied to CUT which is another side of FPGA and then roles are switched and this is to utilize the CLBs to do testing but only additional memory is need to store test patterns. The architecture of FPGA with very good figure representation. PIPs-PS programmed interconnect points are also part of FPGA architecture. Good and graphical representation of interconnections in FPGA. Interconnect test figures with orthogonal and diagonal features. Major faults are listed and elaborated as well as depicted. IOB is tested by the boundary-scan. Detailed description and flow chart of the test shown. Graph of fault coverage and test configuration is shown for CLBs and interconnect. This is a very detailed paper that has a lot of information regarding FPGA architecture.

I.24 Reference paper 24

LOCST: A Built-In self-Test Technique: The linear feedback shift register is used to control the timing of the test instead of retrieving it from different locations. Benefits of BIST: reduce test pattern storage requirements, reduce test time and detect defects on the IC. LSSD stands

for low-level sensitive scan design forms the LOCST which stands L for LSSD OCST on-chip self-test. The formula for random pattern numbers is shown. A picture of LFSR is shown which is an integral part of test pattern generation. Golden comparison is obtained through an ideal simulation operation. More of a historical document. BIST is driven to actually find defective components on IC.

I.25 Reference paper 25

Design of digital Built-In Self-Test for Analog to Digital Converter: BIST for ADC is presented in a paper that analyzes ADC output performance and well as coverage of that BIST concerning area, power, delay, and fault coverage. Criteria that high precision ADC needs to satisfy are low testing cost, less time on testing and testing algorithms have to have good fault coverage. Static and dynamic parameters of ADC need to be taken into consideration while developing the test. Some dynamic parameters are, harmonics, signal to noise ratio, intermodulation distortion while static parameters are gain, offset, differential non-linearity, accuracy, and others. FFT is used to test dynamic parameters but that requires a large chip area. This paper proposes a new digital-based BIST which is not required any signal generation, test pattern storage, or a test stimulus to test the static characteristics of an A/D converter. Formulas and definitions for ADC design are defined in the paper. Various parameters, as well as figures of those parameters, are shown in the paper. Proposed BIST for ADC requires fewer blocks and therefore less power and area.

I.26 Reference paper 26

Efficient Design-for-Test Approach for Networks-on-Chip: Testing the exchange of packets between cores. BIST activation based on fault detection. BIST needs to be isolated from normal circuit operation. EsyTest proposed that has test strategy of testing data path test and control path test. Wrappers used to isolate BIST on NOC. This paper is more of system BIST between cores data exchange. Testing the routers and finding the faults and when the BIST is triggered. This is more system-level but on SOC but it can be used on how BIST is utilized in different applications like DAC and MEMS systems.

I.27 Reference paper 27

Architecture of Field-Programmable Gate Arrays: Published in 1993 and is more historical documentation regarding the FPGA that shows logic blocks and commercial routing architectures. Logic blocks are based on one or more the following: transistor pair, basic small gates such as two input NANDs or exclusive ORs, multiplexers, look-up tables, and wide fan in AND-OR structures. Wire connections are achieved through switches and there is a balance between segments and wires and if too many wires are used results in poor FPGA design and if too many segments are used and not enough wire then the area gets wasted. Wire usage affects the FPGA performance. Switch programable technology used: SRAM where the switch is pass transistor controlled by the state of SRAM bit, anti-fuse when electrically programmed forms a low resistance path and EPROM where the switch is floating gate transistor that can be turned off by charge onto their floating gate. Impact of wiring that produces resistivity and parasitic capacitance but can be overcome with optimization of switching circuitry. SRAM programming requires external memory since this programing is volatile, but SRAM also requires additional circuitry which impacts the area and power consumption. Advantages and

disadvantages of anti-fuse programming are discussed such as advantage less area but disadvantage higher voltage required and its one-time use. Floating gate as programming is also discussed with pros and cons. Talk about granularity regarding the logic block. Logic block architecture is covered with pictures of several logic blocks from different manufactures. Sequential logic is also discussed with the use of flip-flops. Logic blocks are considered when optimizing the density and performance of FPGAs. Lost of good plots regarding performance and granularity. Definitions of routing such as wire segment, track, and routing channel are also good graphs of wiring architecture but wiring architectures depend on manufacture.

I.28 Reference paper 28

Smart Logic Built In Self-Test In SOC: Design for test checks manufacturing defects and verification check is chip is doing per specification. Compression of the pattern to eliminate the need for additional hardware. Controllability and observability of node and they are defined as controllability is the method of forcing a particular value on a node, and observability refers to the ability to drive the values of a node to the primary output such that they can be observed. Test patterns that are generated are random, pseudo-random, or deterministic to weed out faults. Optimization of BIST using the smart algorithm.

I.29 Reference paper 29

FPGAs in Industrial Control Applications: Published in 2011 and its new application of FPGA in industry. Talks about the diminishing factors of DSPs and microcontrollers which are based on pure software platform while it emphasizes the importance of FPGA's and that FPGA's are true system on chip architecture. FPGA's originated from the '80s that offer par-

allelism and throughput higher than their predecessors such as DSP and microcontrollers and FPGA can implement multiple RISC which are part of microcontrollers. FPGAs might have the problem of integrating the AD and DA converters but might be subject to change since new technology of producing the FPGA is emerging. Application of FPGA in automotive and aircraft systems as well as robotics. The next section discusses the use of FPGA in designing switch-mode power supplies. Mentioned other FPGA blocks such as memory blocks, DSP, clock manager, and communication blocks. Sequential operation implemented by use of flip-flops while combinational logic by use of lookup tables. Communication blocks implement various protocols such as CAN, PCI, SPI, and I2C. FPGA's can integrate 1 or several processors with analog peripherals after which are considered SoCs or System on Programmable Chips (SoPCs). The figure of the FPGA design process is shown which is beneficial to paper. SoC design is shown as well in one of the figures. technology allows the development of hardware architectures within a flexible programmable environment. Shows the experiment that outlines the steps taken in developing FPGA design. Application of FPGA in neural networks, fuzzy logic-based control systems, intelligent data acquisition devices, and evolvable hardware.

I.30 Reference paper 30

Logic BIST: State-of-the-Art and Open Problems: Specifies that many IC manufactures are reluctant to use logic BIST due to non-relevant faults that are generated by faults in the actual BIST circuit. The indication is that LBIST is only applicable in critical industries such as medical, military, aviation, and mission-critical such as space and NASA. Reliability and security are provided by LBIST and with that self-repair as technology is becoming an integral part of our lives. BIST is more practical than a burn-in test that can consume a lot of time

and still not detect latent failures in the field. Test execution time is dependent on time to generate stimulus, time to compute and compress response, and time to analyze the response. Information on how LBUST is constructed with linear feedback shift register as a pattern generator. Aliasing error is the error that produces the same response as a good working circuit but in fact, upon closer inspection circuit is faulty. A small and low-cost tester needs to initiate BIST. Specific advantages such as low cost and integration and disadvantages such as area of the chip, power consumption during the pattern generation, and high-speed switching during the scanning of transistors. Issues with fault positive meaning that BIST is reporting the fault when the circuit is fault-free. Heating of the circuit decrease the lifetime of the circuit itself. False positives can be caused by delays and cross-talk among interconnecting wires. Explanation of IR drop, and cross-talk elaborated. Impact on increasing the test frequency can produce false failures. BIST has varying percentage coverage some of BIST has 99% fault coverage while others have 65-80%. The improvements of the LBIST are discussed as well next course of action that needs to be taken to make LBIST more attractive. LBIS test can be subjected to hacker attacks and cause damage therefore must be protected.

I.31 Reference paper 31

Logic BIST for Large Industrial Designs: BIST performance in terms of fault coverage compared with automatic test pattern generator. Limitations of external in terms of capacity and test application time since external tests have a limited number of channels available. Testing can incur 25-30% of manufacturing cost so authors are presenting novel BIST as means of reducing the test time cost and increase reliability. External test issues are presented as well which results in the need for BIST as cost reduction and improvement over external testers. Detail explanation of BIST structure with pattern generators, shift registers, and analyzers is

shown in Figure 1. BIST can be issued through a TAP controller or applying stimulus to primary inputs that will drive the BIST structure. The compactor is a multiple-input shift register that compresses data before being analyzed to speed up testing. Data of BIST is presented in terms of coverage and chip impact area and that BIST is a viable solution for large chip designs. X-masking defined and elaborated in the paper

I.32 Reference paper 32

BIST Pretest of ICs: Risks and Benefits: BIST components are constructed in the same manner as functional logics that they test so they are prone to be faulty as well so the BIST pretest is proposed, The point of the pretest is to verify that actual BIST circuitry is fault free before it is subjected to test logic circuit. Two side effects of pretest: chips are disregarded as a failure because BIST circuit fails but functional logic might be good or chips pass functional testing due to faulty BIST performance. Some probability formulas regarding fault coverages are presented in the paper. Pretest of BIST circuit is done by itself by placing the seed values in LFSR and those seed values should produce known signature output that deemed BIST circuit operational. Provides the formulas on computing the chip yield with pretest BIT being deployed versus not being deployed. Risk of applying the pretest to avoid yield degradation.

Appendix II

Manuals

II.1 Analog Discovery 2 Manual

The reference manual for the Analog Discovery 2 tool can be located on the website shown. Manual has a detailed explanation of wiring and available applications for Analog Discovery 2.

Web: https://reference.digilentinc.com/test-and-measurement/analog-discovery-2/start

II.2 DE0-Nano Manual

Reference PDF manual for DEO-Nano board can be obtained from the website shown. This manual contains all the information regarding the connection and capability of the evaluation board.

Web: https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=593&PartNo=4

II.3 FPGA Datasheet

DEO-Nano board uses Altera Cyclone IV FPGA with part number EP4CE22F17C6N and the datasheet for this FPGA can be obtained on the website shown.

Web: `https://www.arrow.com/en/products/ep4ce22f17c6n/intel`

II.4 Quartus Manual

Manual for Quartus II can be obtained on the web:

`https://www.intel.com/content/www/us/en/programmable/documentation/spj1513986956763`
`html`

Appendix III

Quartus II schematics

Below are the schematic figures crated in Quartus II as source code and their relative block diagrams that are used on the higher-level design. Once each schematic was crated it was synthesized, verified through simulation, and the block was crated which is also shown indicated here. The first part of the appendix is the schematic captures created in Quartus II and the second part of the appendix is the block diagrams shown in Fig. III.5 that are created by using the command that allows creating symbols from the current working design.

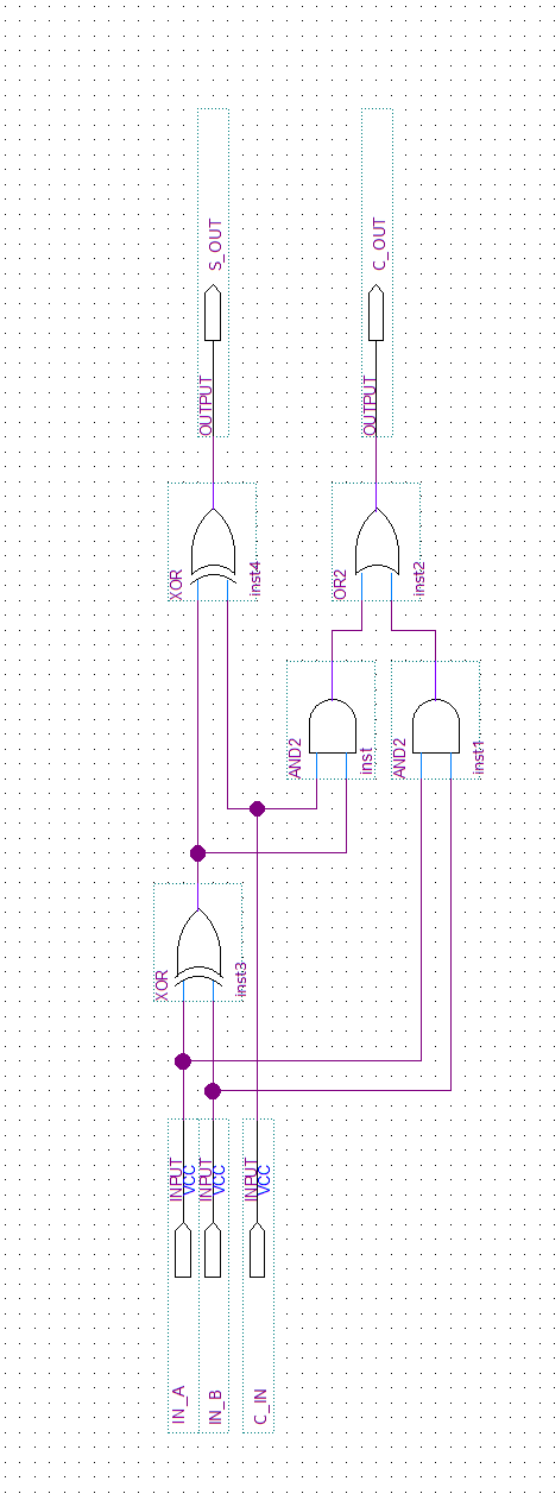

Figure III.1: Single full adder schematic

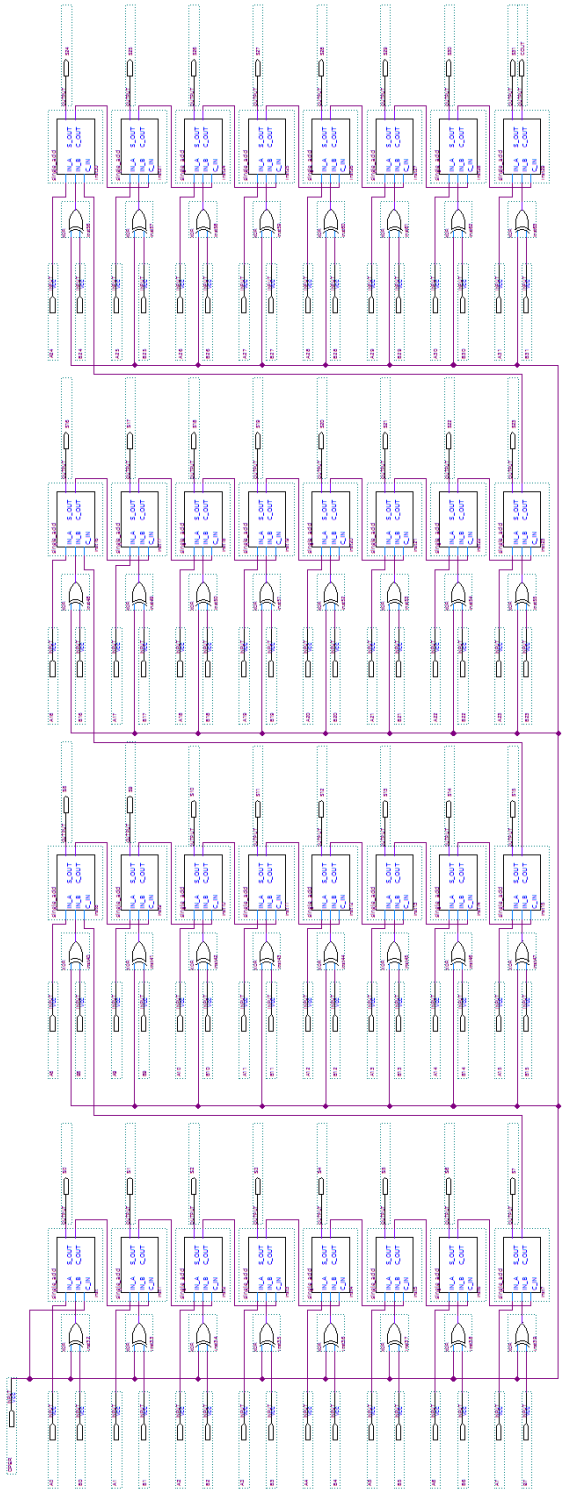

Figure III.2: 32-bit adder/subtracter schematic

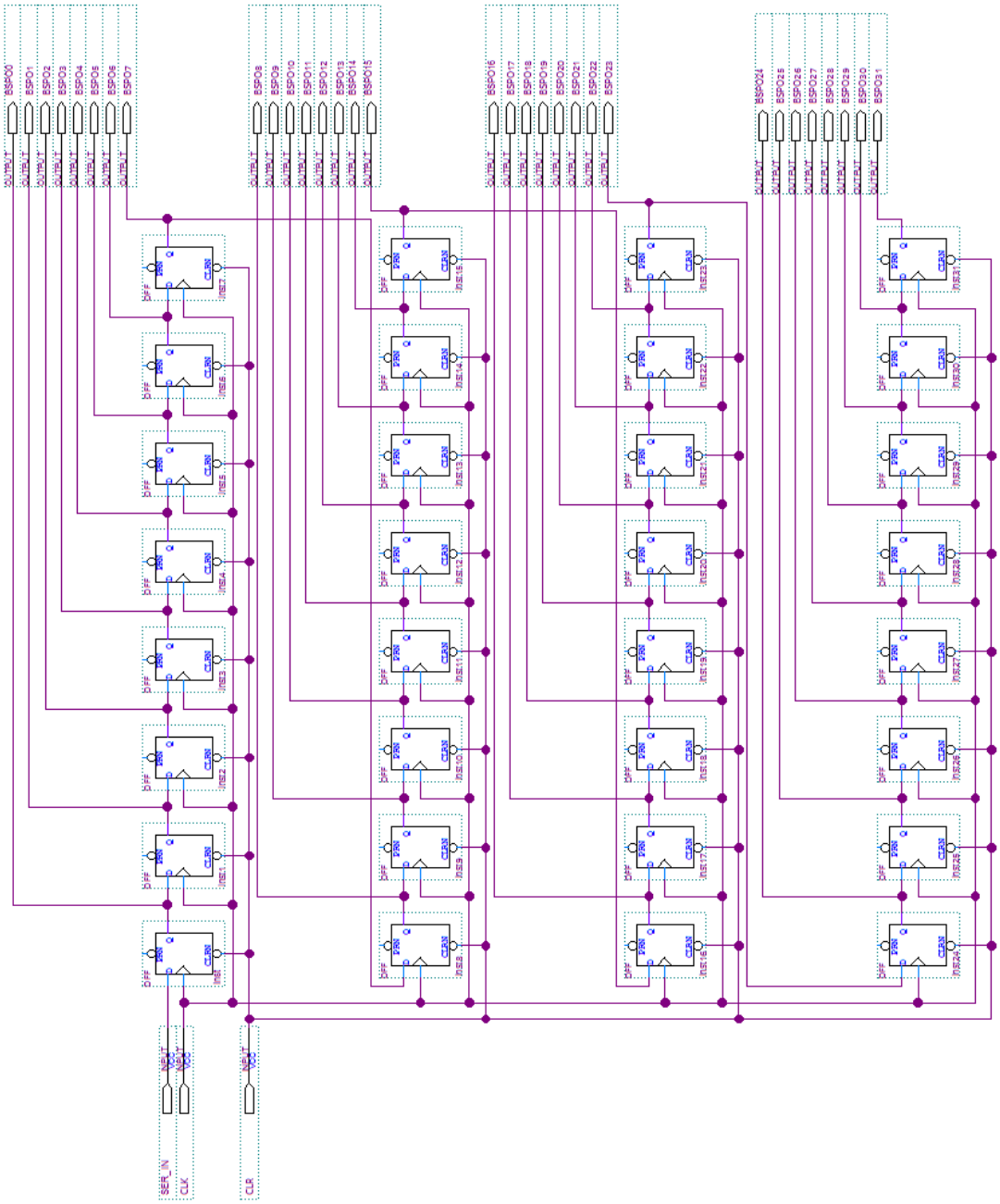

Figure III.3: Serial-in-parallel-out register schematic

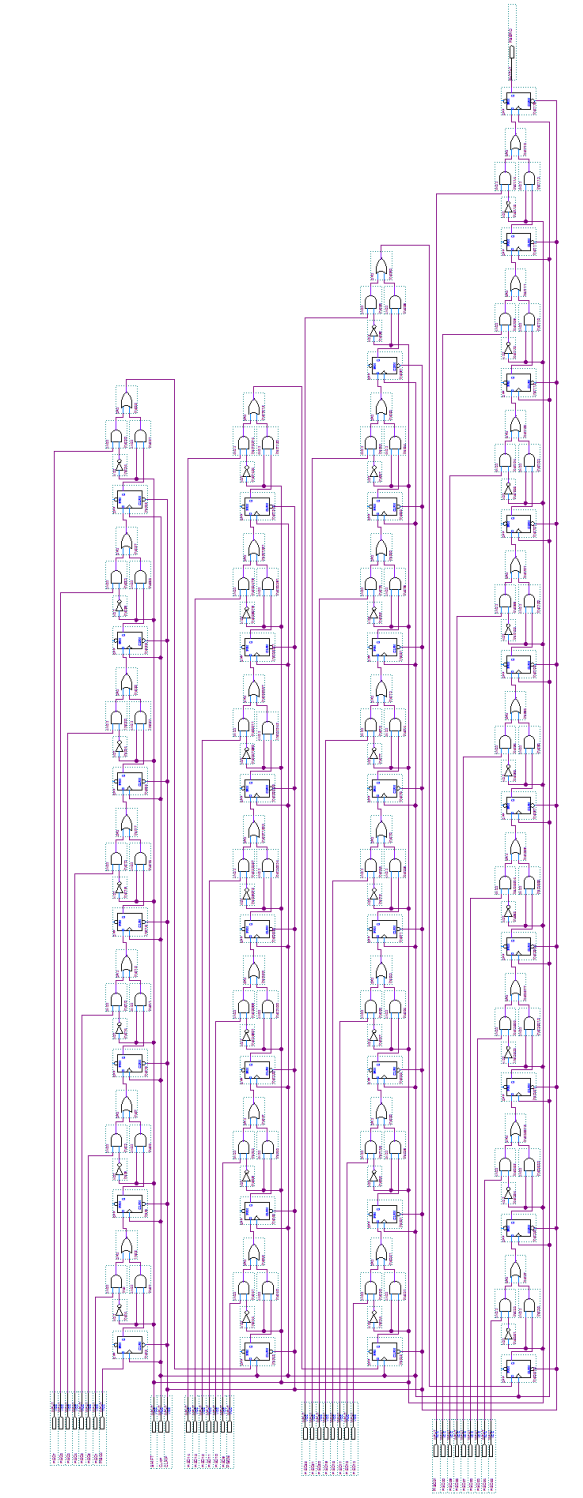

Figure III.4: Parallel-in-serial-out register schematic

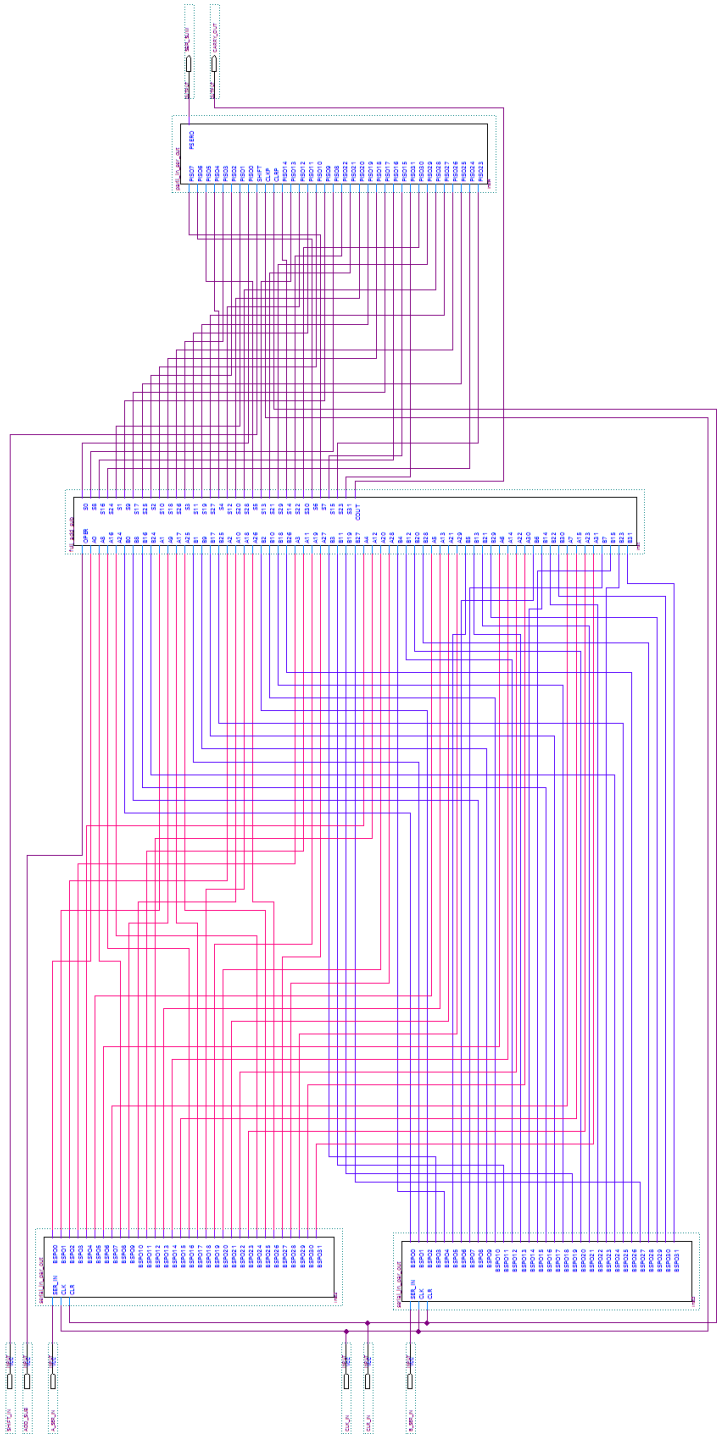

Figure III.5: Final 32-bit adder/subtracter schematic with shift input-output registers