Report

Confirmation of Originality

Faculty of Engineering and Computer Science

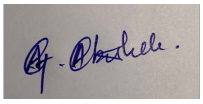
COEN 6521 - Design for Testability

Course Name & Number/Term: _____ **Section:** ____ **Instructor:** Dr. Mahmoud Masadeh

e.g., ENGR410/2 e.g. M

Having researched and prepared this report for submission to the Faculty of Engineering & Computer Science, the undersigned certify that the following statements are to the best of my/our knowledge true:

1. The undersigned have written this report myself/themselves.
2. This report consists entirely of ideas, observations, references, information and conclusions composed or paraphrased by the undersigned, as the case may be, except for statements contained within quotation marks and attributed to the best of my/our knowledge to their proper source in footnotes or otherwise referenced.
3. With the exception of material in appendices, the undersigned have endeavored to ensure that direct quotations make up a very small proportion of the attached report, not exceeding 5% of the word count.
4. Each paragraph of this report that contains material which the undersigned have paraphrased from a source (print sources, multimedia sources, web-based sources, course notes or personal interviews, etc), has been identified by numerical reference citation.
5. All of the sources that the undersigned consulted and/or included in the report have been listed in the Reference section of the report.
6. All drawings, diagrams, photos, maps or other visual items derived from sources have been identified by numerical reference citations in the caption.
7. Each of the undersigned has revised, edited and proofread this report individually.
8. In preparing this report the undersigned have read and followed the guidelines found in Form and Style, by Patrick MacDonagh and Jack Borden (Fourth Edition: May 2000), available at http://www.encs.concordia.ca/scs/Forms/Form&Style.pdf.

Name: Abishek Arumugam Thiruselvi (please print clearly) ID No: 40218896 Signature:  Date: 06 AUG 2022

Name: SRINIDHI HONAKERE SRINIVAS (please print clearly) ID No: 40221128 Signature:  Date: 06 AUG 2022

Name: _____ (please print clearly) ID No: _____ Signature: _____ Date: _____

Name: _____ (please print clearly) ID No: _____ Signature: _____ Date: _____

Name: _____ (please print clearly) ID No: _____ Signature: _____ Date: _____

Name: _____ (please print clearly) ID No: _____ Signature: _____ Date: _____

Do Not Write in this Space – Reserved for Instructor

Concordia University
Engineering and Computer Science

COEN 6521
Design for Testability
Summer 2022

Dr. Mahmoud Masadeh

PROJECT REPORT

Boundary Scan Testing on an 8-bit Array Multiplier Using JATG IEEE Standard 1149

Group Members	Student ID
Abishek Arumugam Thiruselvi	40218896
Srinidhi Honakere Srinivas	40221128

Abstract:

The project aims to use JTAG components to test an 8-bit array multiplier. The Modelsim and Synopsis tools from Mentor Graphics are used to test the design's correctness. We tried to understand the JTAG components' functionality at the project's beginning to test them. The outcomes are analysed and evaluated at the project's end.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1. MOTIVATION

Electronics play a predominant role in our day-to-day life. IC design's gate counts increase exponentially as per Moore's Law. As the technology increases and the problem of testing the design also increases. Testing is a crucial step in the design and manufacturing process. Testing not only helps remove the defective chip but also helps in testing the reliability of the design. Predictability, reliability, and quality are driving factors for staying in this market.

Testing impacts in high cost of IC manufacturing. The margin of error in the medical and aerospace industry should be minimal. The primary motivation behind the design for testability is to deliver an IC with quality and reliability to the end users.

1.2. OBJECTIVES

Design for testability methodologies helps us to render the design bug-free. But testing the design for 100% error free is practically not possible with billions of transistors. The main aim is to synthesize, generate, evaluate, apply, and observe the design to satisfy the specification and also have design techniques that make test generation, test application and test evaluation cost-effective.

In this project, we will implement the DFT in an 8-bit array multiplier using a full adder and perform boundary scan testing within it to test the interconnections. We will evaluate the area, power, delay and energy consumption of the 8-bit array multiplier before and after testing with JTAG designs. Also, the JTAG design should contain four main components and perform four operations.

1.3. OVERVIEW

Boundary scan testing is a structural method in DFT. In the current technology, densely packed ICs are standard, and it is necessary to test the gate functionality and interconnections between them. To make this testing cost-effective, a boundary scan testing concept was proposed.

Boundary scan architecture allows us to access the input and output of the IC externally. This can be achieved by connecting all the primary inputs and outputs to a shift register

with a boundary scan input and output. This idea is used to confirm whether the IC performs its intended functionality, and that ICs are interconnected correctly.

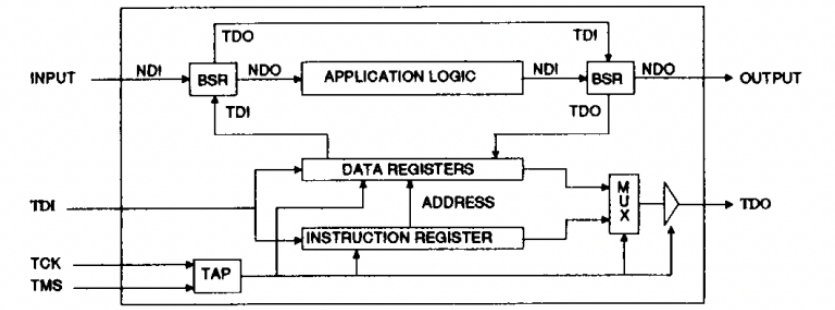


FIGURE 1.1 BOUNDARY SCAN ARCHITECTURE

As ICs are densely packed with thousands of I/O pins, the interconnections are very complex, and there is a probability that there will be stuck-at faults, stuck-open faults, and shorts.

The boundary scan architecture's test access port (TAP) idea makes it easier to test the communication between ICs. The test data in (TDI), test data out (TDO), test clock (TCK), and test mode select (TMS) ports make up the TAP. These ports can be used to access the boundary scan register, some user-defined data registers, or an instruction register in TAP.

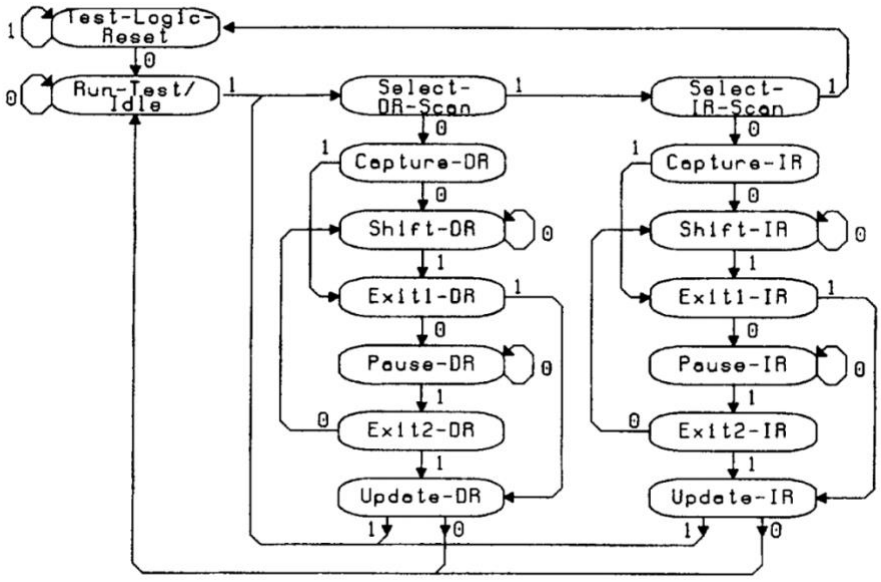


FIGURE 1.2 TAP STATE DIAGRAM

2. DESIGN OF 8X8 ARRAY MULTIPLIER

An array multiplier is used to multiply two binary numbers using an array of and gates, full adders, and half adders. There are three different methods to perform multiplication,

- Repeated Addition
- Shift and Add
- Shift and Add with multiple adders (more predominant)

Hardware speed and performance are always trade-offs; the more complicated the circuitry with more adders, the faster the multiplication output will be. Because if there are fewer adders, we must first store the sum and carry results before reusing the adders to obtain the result.

2.1. THE BASIC ALGORITHM OF THE 4X4 ARRAY MULTIPLIER

Constraints:

- A, B: inputs
- I, J, K, L: Partial Products
- P7: Carry from P6
- Result: Multiplier output

A (4bit Multiplicand): A3 A2 A1 A0

B (4bit Multiplier): B3 B2 B1 B0

A					A3	A2	A1	A0
B					B3	B2	B1	B0
I					A3B0	A2B0	A1B0	A0B0
J				A3B1	A2B1	A1B1	A0B1	
K			A3B2	A2B2	A1B2	A0B2		
L		A3B3	A2B3	A1B3	A0B3			
Result	P7	P6	P5	P4	P3	P2	P1	P0

The procedure to perform multiplication is as follows,

STEP 1. The inputs A and B are given.

STEP 2. The partial products are obtained and shifted left in consequent steps. The obtained partial products are the results from the AND gate.

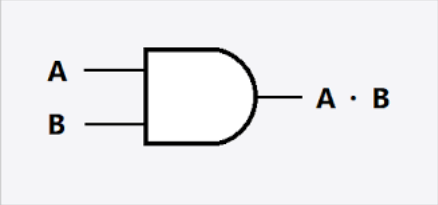


FIGURE 2.1 AND GATE

STEP 3. The Half Adders and Full Adders are used to generate the sum and carry of the partial products.

STEP 4. The sum is sent to P0-P6 (vertically), and the carry is set horizontally.

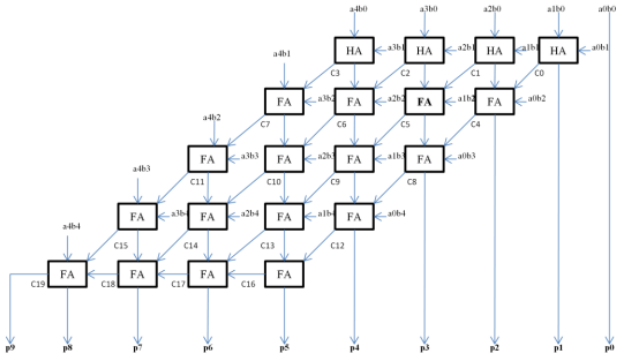


FIGURE 2.2 SUM AND CARRY PROPAGATION

STEP 5. The result is printed when all the sum and carries are propagated.

Summary of the procedure to design the required Array multiplier from the 4x4 array multiplier,

a) 16 AND Gates => A_n x B_n
b) 4 Half Adder => number of B_n
c) 8 Full Adders => (A_n -2) B_n
d) 12 Total Adders => (A_n -1) B_n

2.2. REQUIREMENTS FOR DESIGNING AN 8X8 ARRAY MULTIPLIER

From the summary of section 2.1, we can be able to design the 8x8 array multiplier. The requirements are stated below,

REQ 1. 64 AND Gates.

REQ 2. 56 Full Adders.

REQ 3. The total number of adders is 56.

To design a better array multiplier, we should consider the timing constraints as the new technologies in the market have clock frequency in GHz.

For example, the generic consideration of timing in 4x4 array multiplier circuitry is as follows,

T_c: Carry Propagation Time

T_s: Sum Propagation Time

T_A: AND result Propagation Time

if ($T_c > T_s$):

Total Multiplication time = $T_A + 6T_c$ (or) $T_A + [(A_n - 1) + (B_n - 1)] T_c$

if ($T_c < T_s$):

Total Multiplication time = $T_A + 3T_c + 3T_s$ (or) $T_A + (A_n - 1) T_c + (B_n - 1) T_s$

3. STRUCTURAL VS. BEHAVIOURAL IN TESTING SYSTEM IN VLSI

IC companies are searching for more efficient methods to accelerate the delivery of higher-quality products onto the market. Until now, manufacturers have depended on engineers to create test scripts to detect device function problems and have developed sophisticated automatic test resources (ATE) to prevent any bottlenecks in the ICs.

Even with improved device speed and complexity, DFT software and sophisticated test equipment can help manufacturers reduce cost-of-test and time-to-volume.

3.1. STRUCTURAL TESTING IN VLSI

The structural test aims to see whether the product was manufactured correctly concerning the specification. Structural testing aims to find flaws in a device's underlying gates and connections rather than testing for properly executing complicated functionalities. The central aspect of structural testing is given below,

- DFT method with additional test access points in the design
- Techniques like Scan, BIST and Path or Transition delay (increases CO and CC).
- Test assembly and preparation software
- Structural Test Equipment to aid engineers helping with ATE.

Built-in self-test (BIST) structures reduce the amount of data that needs to be transferred with ATE. These digital BIST elements, integrated into the designer's circuits, can run several tests required to evaluate the device's condition. Existing BIST techniques, frequently applied to

traditional designs like memory or data paths and increasingly to random logic, provide thorough fault coverage with little external assistance. This benefit applies to production testing.

A pure structural test technique would significantly simplify the restrictions placed on production test equipment. Structural test systems can handle much fewer pins than typical ATE, with the most they can support having the few pins required to work with a device's test access ports and use the TAP to test the circuitry.

3.2. BEHAVIOURAL TESTING IN VLSI

Functional tests are often found necessary for verification of the design. The functional test consists of the input vectors and the corresponding responses. They test the internal chip nodes to ensure that a verified design is operating correctly. Functional tests cover a relatively large portion of simulated defects in logic circuits. Functional vectors are frequently thought of as verification vectors, which check to see if the hardware truly adheres to the specification. In contrast, any applied vectors in the ATE are considered functional fault coverage vectors used in manufacturing testing. These two different functional tests may be the same.

4. JTAG IN VLSI SYSTEMS

The foundation of JTAG is a combination of boundary, internal, and scan-set technologies. This testing framework incorporates DFT and debugging logic, which saves on hardware costs for several scan chains and lowers design and verification costs. This structure can offer nearly 100% fault coverage and JTAG port debugging capabilities. A testing framework built on JTAG fulfils the need for fault testing and enables debugging via the JTAG port.

To make test code generation easier, testability design enhances circuits' controllability and observability. Increasing the observability and controllability of internal logic is the testability design task. Observability means the difficulty of the internal logic state observed from the primary output. Controllability means the difficulty of the internal logic states controlled by the primary input.

Figure 4 is a basic schematic of the testing structure based on JTAG. ICs are designed with a testing structure that integrates debugging logic and testing logic to reduce the area cost.

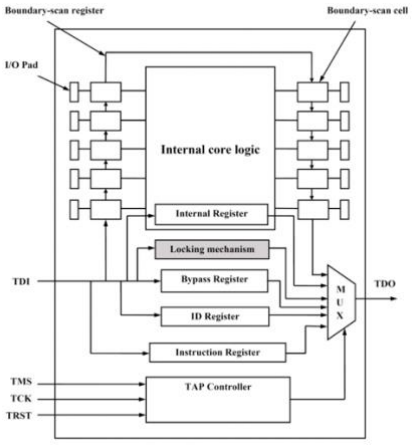


FIGURE 4. 4.1 JTAG STRUCTURE

The Test Access Port (TAP) in JTAG is used for testing and debugging. It has four required pins (TDI, TMS, TCK, and TDO) and one optional pin for asynchronous reset (TRST). A TAP controller is a module that manages and organises all of the test architecture's operations. A Hardware Description Language can be used to create a TAP controller like that of a logic design module (HDL). TAP/JTAG controllers are a common component of the VLSI ICs produced nowadays. Regardless no matter how difficult the testing task, the test process always employs the same states and signals (Figure 1. 2):

- TDO - test data out, test responses, or data that loops through
- TRST - test reset, an optional hardware reset signal for the test logic
- TDI - test data in, test vectors, and JTAG instructions are applied via TDI.
- TCK - test clock, All events occurring while in test mode occur on edges.
- TMS - test mode select determines the next state of the JTAG port

The Instruction Register is one of the state variables handled by the state machine (IR). The standard size of the instruction register is 4 to 16 bits. The JTAG standard requires some instructions, but developers are free to define as many as they want. The required instruction, BYPASS, is among the most important instructions. A JTAG-compliant chip inserts a single flip-flop between its TDI input and TDO output when the IR contains the BYPASS opcode. A series of chips acts similarly to a shift register in the BYPASS state.

5. JTAG INSERTED INTO THE VLSI SYSTEM

On today's VLSI design circuit, the JTAG implementation is frequently utilised, much like other contemporary gadgets. Joint Test Action Group, also known as JTAG, was first developed to

use the boundary scan method to test IC and VLSI circuit boards. The IEEE 1149.1 specification designator officially refers to it by engineers.

Integrated circuits with JTAG implementation have a JTAG Test Access Port for debugging. The Test Access System provides access to several Data Registers and one Instruction Register. One of these Data Registers must be used to implement a BYPASS Register. The information in the Register File decides which of the significant Data Registers should be utilised. Creating a JTAG interface only needs 4 or 5 pins. The Test Clock, The Logic TAP controller state machine, is governed by the Test Mode Select or TMS. Test Data In and Test Data Out. The optional TRST fifth signal can be asserted to reset the JTAG TAP controller state machine asynchronously.

5.1. AREA, POWER, DELAY AND ENERGY CONSUMPTION

Figure 5.1 corresponds to the JTAG and multiplier circuitry area report. The power in the circuit depends on the currents in the system, i.e. ($P \propto I$); the currents can be I_{static}, I_{delay} and $I_{switching\ if}$ the measured lengths of the JTAG IR and DR scan paths are invalid. This indicates that an error exists in the link delay or scan path. Switching operations mainly bring on power consumption in an SRAM-based device. Without JTAG activity, you can expect the standby supply current specified in the datasheet. By switching from parallel to serial buses, you can save cost, space, and power on your hardware. Although common, lower system performance may be acceptable in many situations.

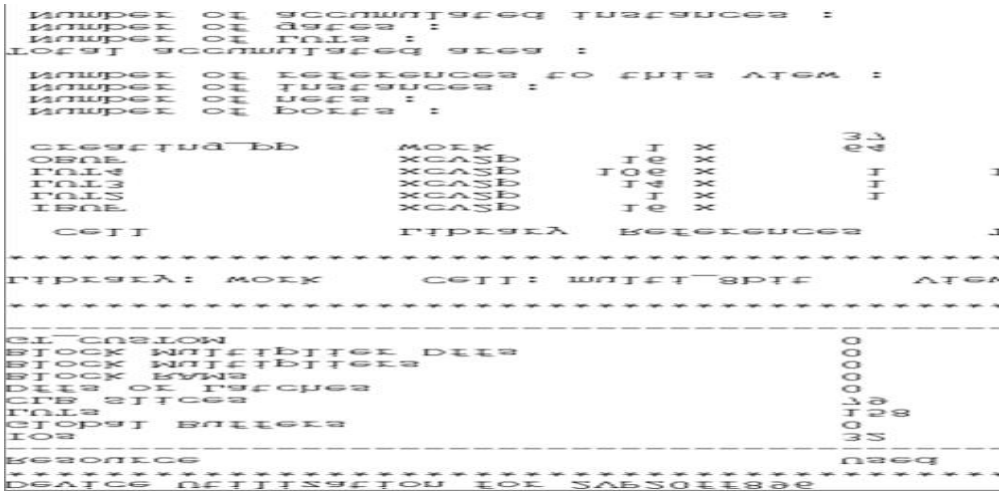


FIGURE 5.1 AREA REPORT

6. RTL SYNTHESIS FOR THE COMPONENTS.

6.1. 8BIT- MULTIPLIER RTL DIAGRAM

The description of the functionality of the 8-bit array multiplier is given in chapter 2.

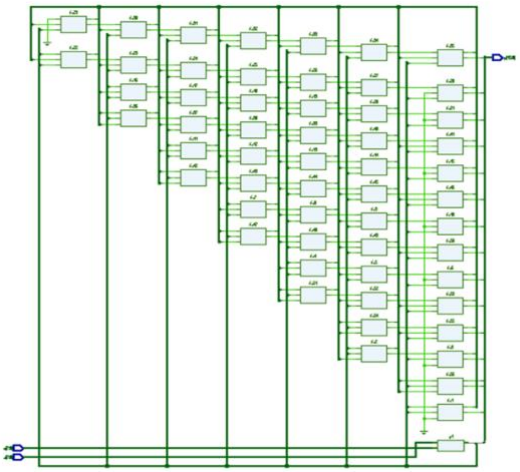


FIGURE 6.1 8BIT- MULTIPLIER RTL DIAGRAM

6.2. BOUNDARY SCAN CELL

A device's boundary-scan cells can drive data onto pins or capture data from pin or core logic signals. The collected data is serially shifted outside and compared with the predicted outcomes. The boundary-scan cells are serially moved with forced test data.

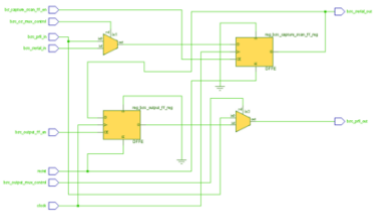


FIGURE 6.2 BOUNDARY SCAN CELL

6.3. BOUNDARY SCAN REGISTER

Boundary scan cells section 6.2. are used to define each shift register. With these boundary scan cells, you can regulate and monitor what happens at each input and output pin.

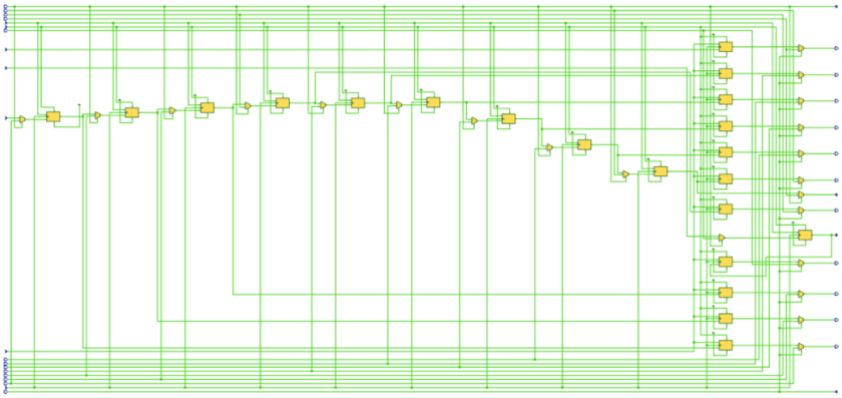


FIGURE 6.3 BOUNDARY SCAN REGISTER

6.4. TAP CONTROLLER

Figure 1.2 depicts the state machine described in the IEEE 1149.1-2013 standard.

The state machine is straightforward and has only two paths: The path used to load instructions is called the data register (DR). The instruction register (IR) path reads and writes data to and from data registers (BSR), including the boundary scan register. The behaviour of the state machine is controlled by the value of the test mode select (TMS) pin as it advances on the test clock (TCK) edge.

We start by clocking a TMS = 0 to enter the Run-Test/Idle state and then clock a TMS = 1 to start choosing a path, presuming the state machine starts at Test-Logic-Reset. The output MUX is switched based on the state chosen by the TAP controller in Figure 6.4, which also drives the state machine.

There are two methods:

- The capture-shift instruction route
- Data collection-shift path

One of the data registers is the boundary-scan register, which consists of the boundary-scan cells surrounding the I/O pins. Shift registers, or data registers, can have any length.

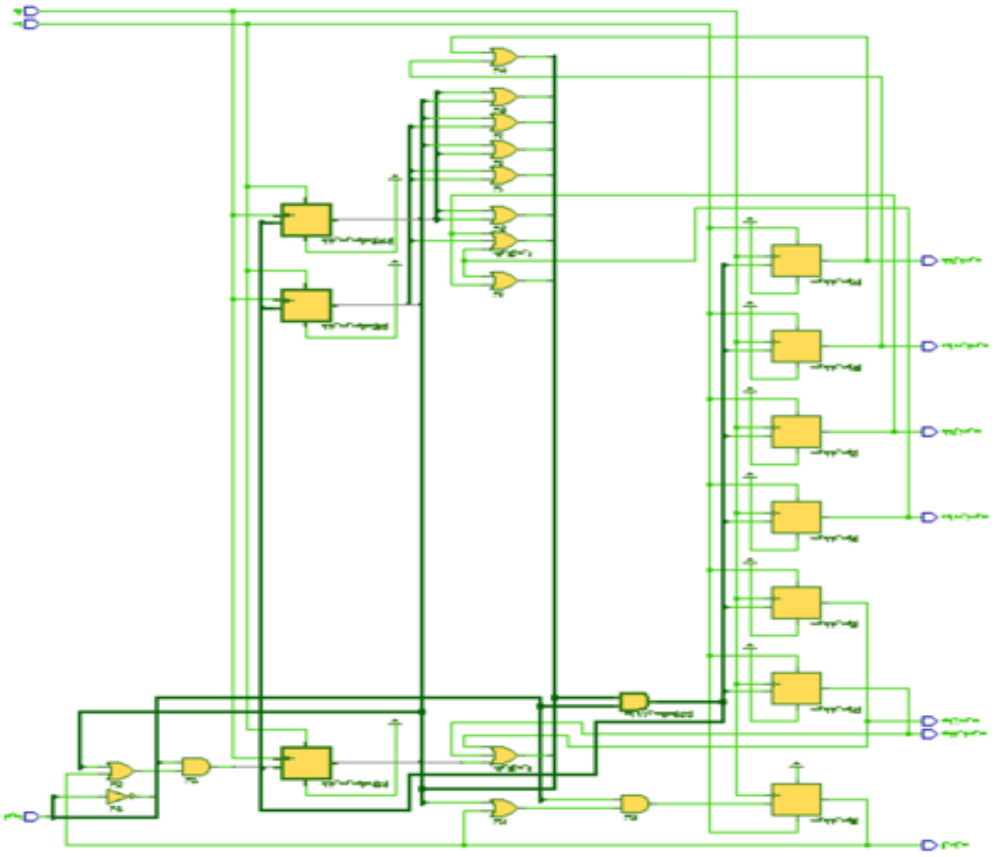


FIGURE 6.4 TAP CONTROLLER

6.5. BYPASS CELL

The functionality of the bypass cell is that by providing a direct link between TDI and TDO of 1-bit length, the device is bypassed.

When the current instruction in the IR is the BYPASS instruction:

Data is transmitted from TDI to TDO with a delay of one TCK cycle during the Shift-DR state. A logic 0 is loaded into this register during the Capture-DR state, and nothing happens during the Update-DR stage.

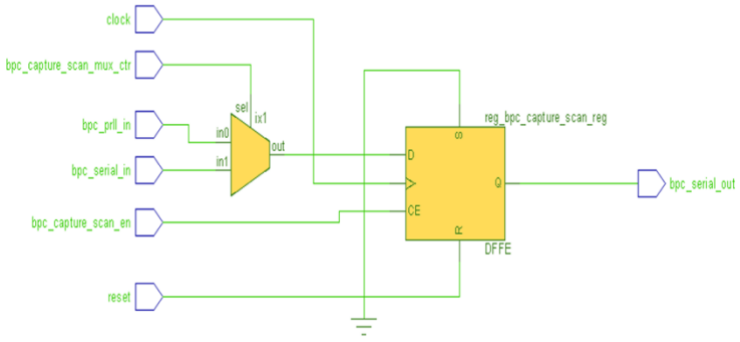


FIGURE 6.5 BYPASS CELL

6.6. IR REGISTER

A 4-bit instruction register is incorporated in the TAP controller (IR). During the Update-IR state, the instruction is given to an instruction decoder. For an overview of the TAP controller operational states, see Section 6.4. The TAP controller state machine defines the conditions under which the instruction decoder interprets and executes the three necessary instructions (BYPASS, SAMPLE/PRELOAD, and EXTEST).

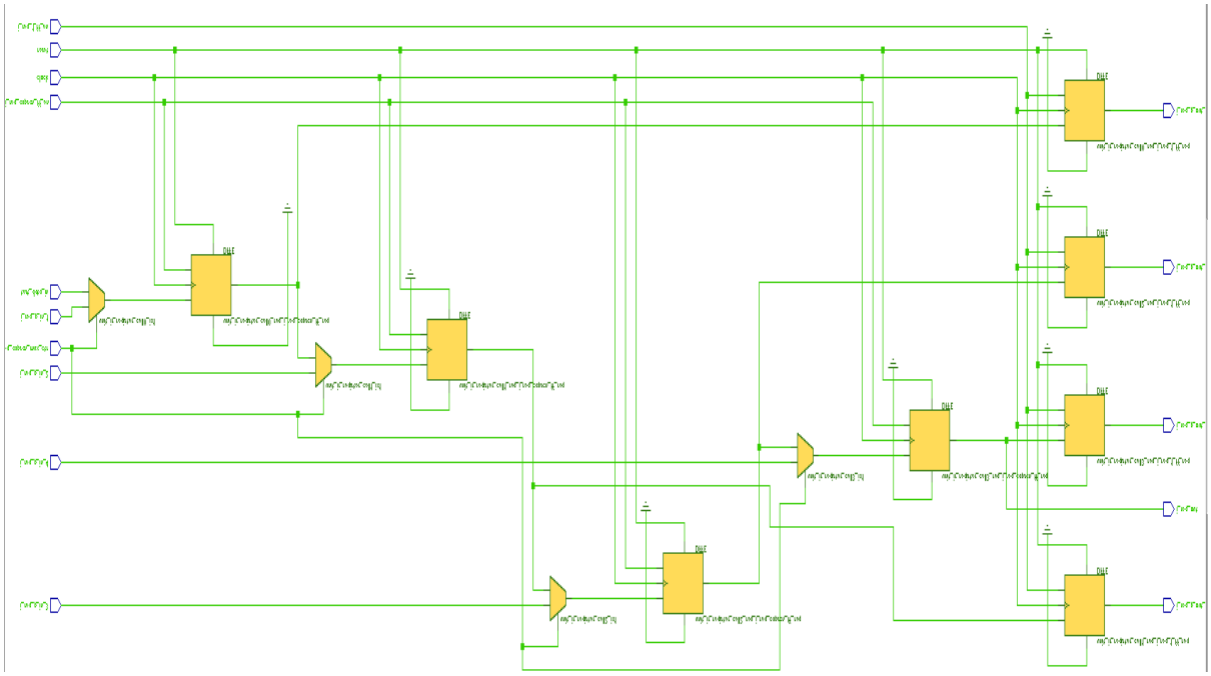


FIGURE 6.6 IR RGISTER

7. OVERHEAD AND BENEFITS OF THE ADDED JTAG

7.1. OVERHEAD

- Area overhead, I/O pin, performance degradation, and design effort are just a few of the expenses involved with converting an existing model into a JTAG design. The implementations Cost are area Overhead Cost, I/O Pin Cost, Performance Degradation Cost, Test Time Cost, and Design Efforts Cost.
- Performance decline brought on the JTAG implementation Failure or deterioration of a facility, process, system, or component that lowers the dependability of a facility's crucial chips, the loss or degradation of which precludes the system from carrying out its intended function.
- More space and pins in the hardware overhead.
- Yield loss will result from the JTAG implementation's larger area.
- More significant power usage while testing the JTAG Power overhead implemented.

7.2. BENEFITS

- JTAG allows users to test PC-board interconnects without requiring physical test probes or equipment.
- It does not need a bootable state for the board to make a problem diagnosis.
- Devices including Flash, CPLDs, FPGAs, and Serial EEPROMs can be programmed in-system via JTAG.
- Development of automated tests for memory, flash, and DSP initialisation
- Diagnostics at the device level
- Personalised error messages
- JTAG tests every external memory location exhaustively before running any boot code.
- Test vectors can be utilised again in manufacturing.

8. TESTING VLSI USING THE JTAG

8.1. SIMULATION OF JTAG COMPONENTS

Figure 8.1 shows the simulation result of 8-bit Multiplier output. The first input is given as binary one and the second as binary 3, and binary three was observed in output P.

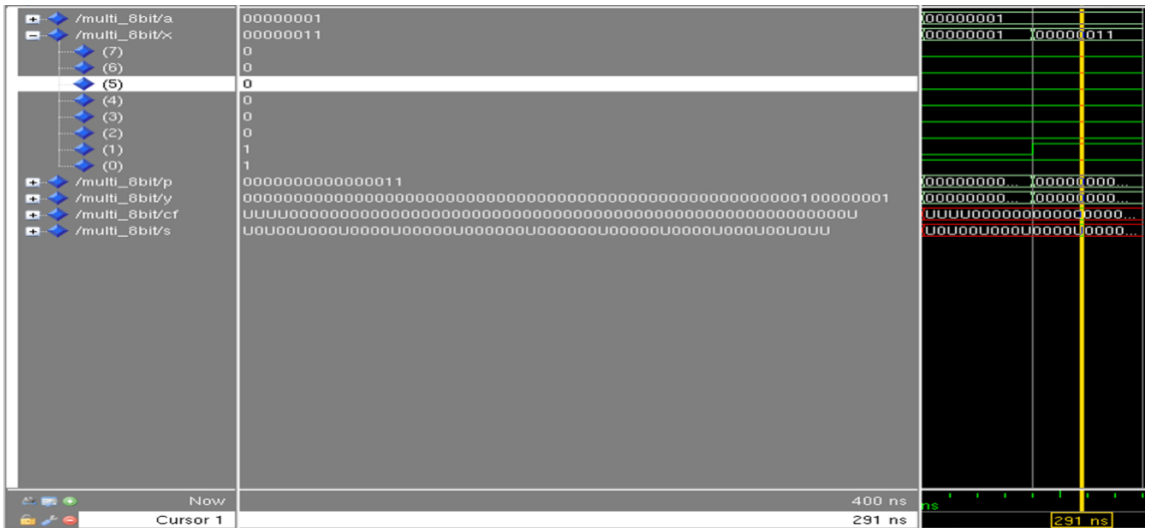


FIGURE 8.1 8BIT- MULTIPLIER TB_WAVEFORMS

The boundary-scan cell is simulated using ModelSim based on the test bench. The parallel data is directly transferred to parallel out when the output flipflop is on. When capture and scan flipflop is on, the serial in is selected and sent to the next boundary scan cell, as shown in figure 8.2.

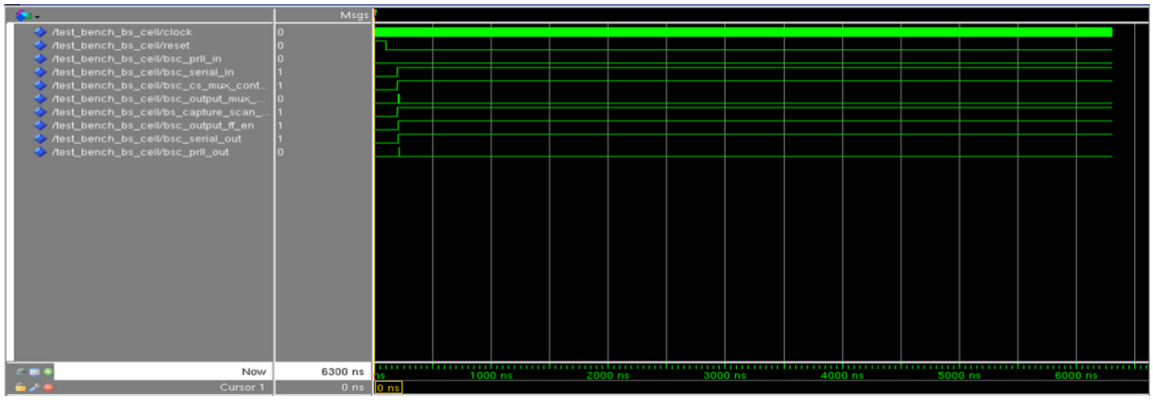


FIGURE 8.2 BOUNDARY-SCAN CELL TB_WAVEFORMS

Figure 8.3, Boundary-scan registers are simulated using models based on the test bench. The input is given to TDI, and data shifting is observed in all boundary scan cells until TDO.

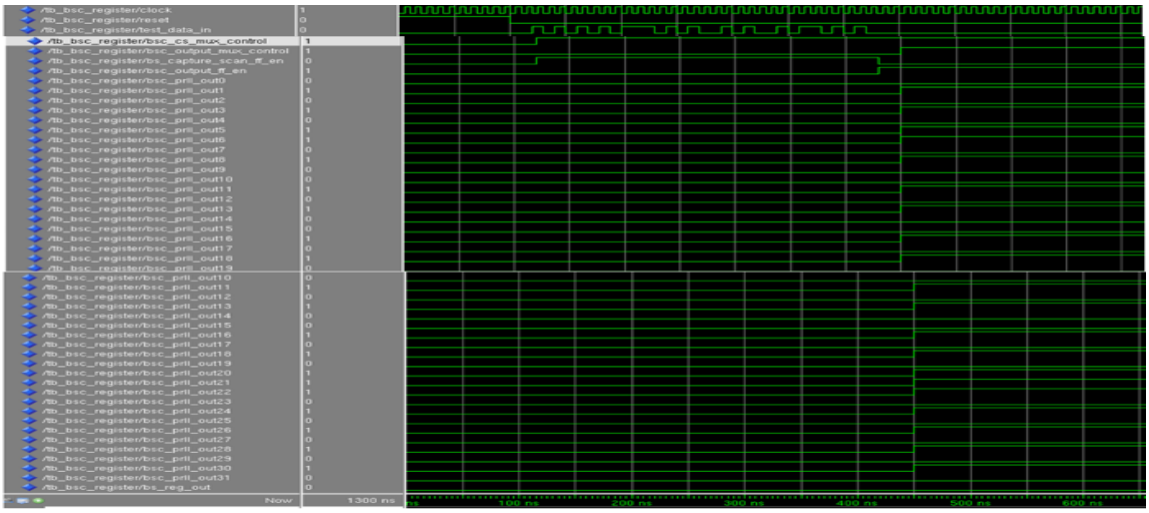


FIGURE 8.3 BOUNDARY SCAN REGISTER TB_WAVEFORMS

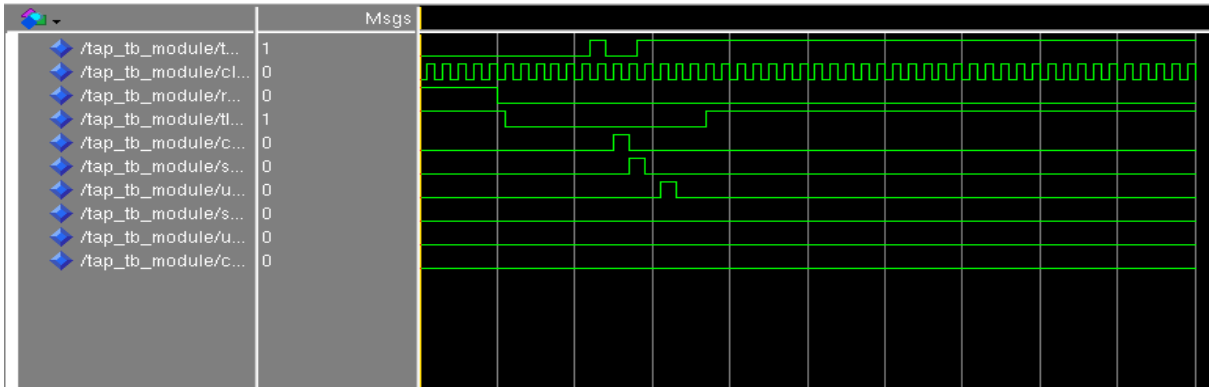


FIGURE 8.4 TAP CONTROLLER TB_WAVEFORM

The TMS value given in the testbench is 010011. We can observe in the Figure 8.4 that after the first the value of the TMS, i.e., 010, Capture-DR is triggered. Similarly, after the next TMS input of 0 the Shift-DR is triggered. Finally, after the next TMS input of 11 Update-DR is triggered.

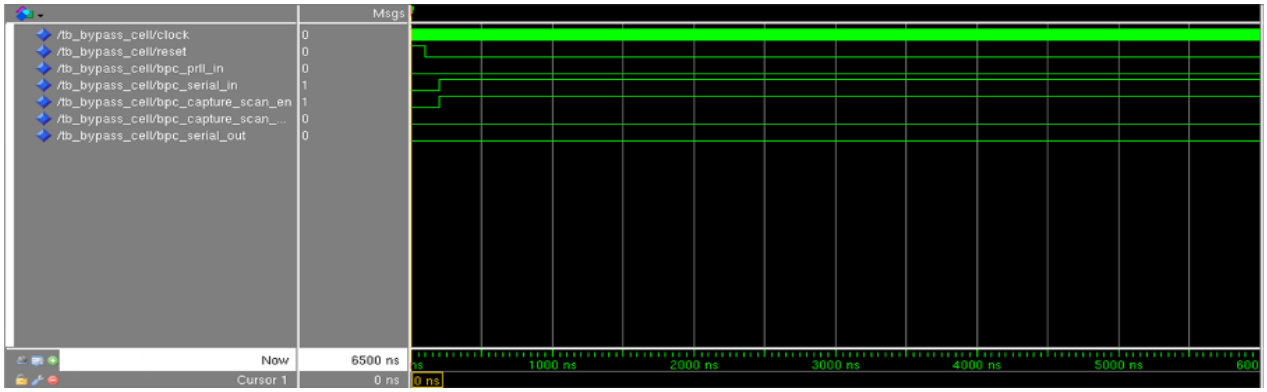


FIGURE 8.5 BYPASS CELL TB_WAVEFORM

Bypass cell is simulated using ModelSim based on the test bench. The parallel data is directly transferred to parallel out when the output flipflop is on. This is mainly used when data is transferred to another chip, as illustrated in figure 8.5.

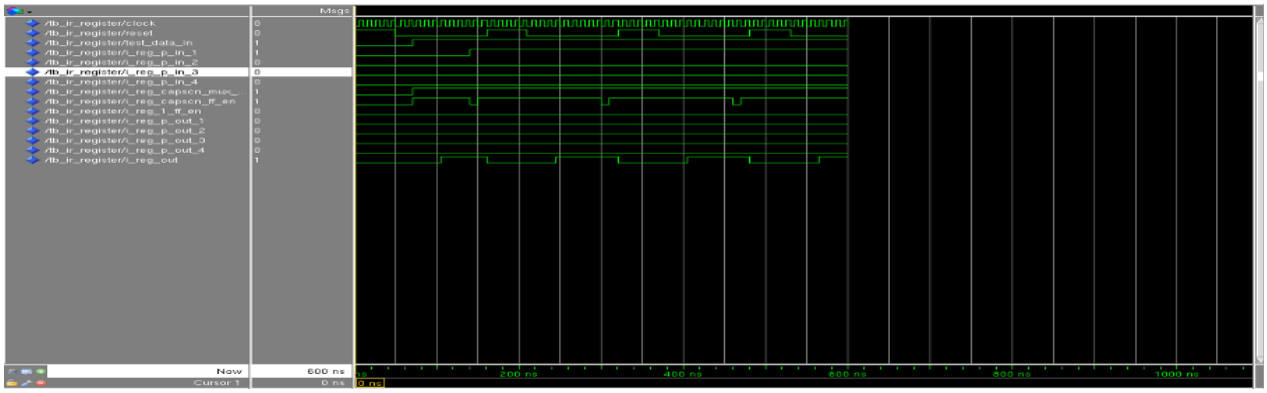


FIGURE 8.6 IR REGISTER TB_WAVEFORM

9. CONCLUSION

9.1. SUMMARY

JTAG offers an access mechanism well suited to interacting with many cores on SOC or multiple packages. It is utilised to reduce pin count and for better functionality. Maintain compatibility when creating, testing, and designing boards, ICs, etc., because of the lower pin capacity and increased functionality. They are straightforward to use and improve. Compared to the IEEE 1149.1 standard, the effort required to test and develop the TAP controller minimises using JTAG.

9.2. DISCUSSIONS AND RELATED WORKS

Using 1149.1 as a standard interface for wafer testing reduces test complexity and costs, according to Landis [10] and other boundary scan-related works. The architecture of a logic Device tester for components with BST and array BIST is covered by Bassett et al. in their article [11]. The economics of DFT measurements like scan, BIST, and BST are covered by Hassan et al., Levitt and Abraham [60], and Dislis et al. [12]. Levitt and Abraham consider the basic issue of whether the cost of including scans in an integrated circuit design justifies it. The number of dies per wafer, the cost per die, the influence on yield, and the rise in gate count are all factors. The impact on profitability is then taken into account.

9.3. CHALLENGES

- One of the most complex parts was the time management issues.
- The personal laptop was hard to use file debugging. Later, we worked on the whole project in-lab computers.
- Trying to match pattern flow was hard, but we learned how circuitry works.
- Understanding the functionality of VHDL was a difficult task.
- Initially, we were working from a different point of view. We could not port map all the individual components in developing the top file.

10. REFERENCES

[1] Bushnell, Michael, and Vishwani Agrawal. *Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits*. Vol. 17. Springer Science & Business Media, 2004.

[2] https://moodle.concordia.ca/moodle/course/view.php?id=145715https://www.synopsys.com/glossary/what-is-equivalence-checking.html.

[3] Hassan, Abu, Janusz Rajski, and Vinod K. Agarwal. "Testing and diagnosis of interconnects using boundary scan architecture." *International Test Conference 1988 Proceeding@ m_New Frontiers in Testing*. IEEE, 1988.

[4] Whetsel, Lee. "A proposed standard test bus and boundary scan architecture." *Proceedings 1988 IEEE International Conference on Computer Design: VLSI*. IEEE Computer Society, 1988.

[5] "https://www.design-reuse.com/articles/5100/what-designers-need-to-know-about-structural-test.html"

[6] Tehranipoor, Mohammad, and Cliff Wang, eds. *Introduction to hardware security and trust*. Springer Science & Business Media, 2011.

[7] Bhakthavatchalu, Ramesh, Saranya K. Kannan, and M. Nirmala Devi. "Verilog design of programmable JTAG controller for digital VLSI IC's." *Indian Journal of Science and Technology* 8.17 (2015): 1-7.

[8] P. Zhang, Y. Song, J. Zhang and Z. Xing, "Design of Testing Struture in Microprocessor Based on JTAG," 2009 Second International Symposium on Computational Intelligence and Design, 2009, pp. 223-226, doi: 10.1109/ISCID.2009.62.

[9] *How to future proof your JTAG debugging, Richard Copeman, Senior Support Engineer, Lauterbach UK*

[10] Landis, David L. "A self-test system architecture for reconfigurable WSI." *Proceedings.'Meeting the Tests of Time'., International Test Conference*. IEEE, 1989.

[11] Bassett, Robert W., et al. "Low-cost testing of high-density logic components." *IEEE Design & Test of Computers* 7.2 (1990): 15-28.

[12] Dislis, Chryssa, et al. "Cost analysis of test method environments." *Proceedings.'Meeting the Tests of Time'., International Test Conference*. IEEE, 1989.