8-Bit Multiplier Integrated with JTAG Standard

Department of Electrical and Computer Engineering

COEN 6521 – Design for Testability

(Summer 2023)

PROJECT REPORT

Instructor: Dr. Mahmoud Masadeh

Done By:
Prithvik Adithya Ravindran (40195464)
Rushik Shingala (40221905)

Abstract:
The JTAG Standard, also known as IEEE 1149.1, serves as an industry-wide benchmark for electronic design testing. The Boundary Scan or JTAG method is merged into a given multiplier design with the aid of four fundamental JTAG instructions: EXTEST, BYPASS, SAMPLE, and PRELOAD. The multiplier takes two 8-bit inputs and generates a 16-bit output. Key elements of the JTAG standard encompass the TAP controller, Boundary Scan cell, Bypass Register, and Instruction Register. The primary goal is to seamlessly integrate the design and Boundary Scan to enhance device testability. This project involves employing VHDL language on Modelsim Simulator and conducting RTL synthesis using Precision.

Contents

Motivation:

In the 1970s, PCBs are tested by probing the backs of the boards with nails. It is possible because technology of that era was DIP packages (Dual in-line). Additionally, this technique was partially automated, and still relied extensively on ad-hoc testing insertion. The divide-and-conquer strategy of the bed-of-nails tester lets us test components in the circuit as if they were standing alone. However, this has the weakness that an open in an input line to an IC will cause the IC to appear to fail, when in fact it is good. With increasing complexity and reducing the distance between to components and pins, using this technique become difficult. To fulfill the needs, Joint Test Action Group formed they proposed the solution was to use internal serial shift register to access input, output and interconnect of circuits.

Objective:

The aim of this project is to successfully incorporate the JTAG (Joint Test Action Group) architecture into the provided multiplier design. Additionally, the project involves evaluating the area and power overhead associated with the implementation of JTAG, both in comparison to the original design without JTAG.

Overview:

The primary objective of this project is to successfully implement the JTAG (Joint Test Action Group) standard along with its essential components. These components encompass the Test Access Port (TAP), a 16-state Finite State Machine (FSM); Data registers, including the Boundary Scan Register (BSR) and Bypass Register; the Instruction Register; and the Instruction Decoder. Furthermore, the project entails the incorporation of the Boundary Scan Chain (BSC), involving four crucial operations: Normal, Scan, Capture, and Update. Additionally, the four mandatory JTAG instructions — EXTEST, BYPASS, SAMPLE, and PRELOAD — are to be integrated.

Following the design and integration of the JTAG architecture, a key aspect of this project involves evaluating the resulting area and power overhead. This assessment provides valuable insights into the impact of JTAG on the overall design, guiding future decisions and optimizations in the domain of integrated circuit development.

Description of the designed VLSI system (8-bit multiplier)

Our logic design is primarily 8-bit multiplier function. This operation involves taking two 8-bit integer inputs and generating a 16-bit output result. The logic design is represented through four VHDL files: " Exact_array_multi_8bit.vhd "(Top model), "creating_pp.vhd", "full_adder.vhd", and "myAND.vhd". First “Exact_array_multi_8bit.vhd” is Top model. It combines adder and ‘And’ operation to perform the multiplication. In second file, all ‘And’ operation performed. In third file, logic of And gate written and in last file full adder is design.

The structure of the design relies on the "creating partial product VHDL code" to construct individual P values as depicted in below Figure. Each P value is derived by employing the "myAND VHDL code" to perform bitwise AND operations. Subsequently, every p values are added according to figure by considering forwarded carry.

This design approach effectively utilizes these VHDL files to implement the multiplication process, which is centered on generating partial products, bitwise AND operations, and employing full adders to aggregate the results.

							y_7	y_6	y_5	y_4	y_3	y_2	y_1	y_0
							x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
							p_{70}	p_{60}	p_{50}	p_{40}	p_{30}	p_{20}	p_{10}	p_{00}
						p_{71}	p_{61}	p_{51}	p_{41}	p_{31}	p_{21}	p_{11}	p_{01}	
					p_{72}	p_{62}	p_{52}	p_{42}	p_{32}	p_{22}	p_{12}	p_{02}		
				p_{73}	p_{63}	p_{53}	p_{43}	p_{33}	p_{23}	p_{13}	p_{03}			
			p_{74}	p_{64}	p_{54}	p_{44}	p_{34}	p_{24}	p_{14}	p_{04}				
		p_{75}	p_{65}	p_{55}	p_{45}	p_{35}	p_{25}	p_{15}	p_{05}					
	p_{76}	p_{66}	p_{56}	p_{46}	p_{36}	p_{26}	p_{16}	p_{06}						
p_{77}	p_{67}	p_{57}	p_{47}	p_{37}	p_{27}	p_{17}	p_{07}							
s_{15}	s_{14}	s_{13}	s_{12}	s_{11}	s_{10}	s_9	s_8	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0

*Figure 1 Multiplication 8*8*

Description of testing the VLSI system (structural vs behavioral)

Structural and behavioral are two main testing approaches in VLSI system. Both are explain below.

Behavioral model: In this model complete test set is generated for given circuit for all combination. So, it will take more time to test and sometimes testing is impossible.

Structural model: This model use different fault models approach such as stuck at fault, bridging faults, assertion faults, cross-point faults, memory faults and etc.

For behavioral testing, 1-bit adder has 3 input and 2 outputs, so total possible combination is 2^5. In this testing is possible but when increase the complexity of circuit and for large is not possible.

For given 8-bit multiplier, total 2^32 combination is possible and this not possible test. Consider 1 GHz ATE then it takes years to test that is not affordable.

For same 8-bit given multiplier using structural models need very less time. By considering stuck at faults models,

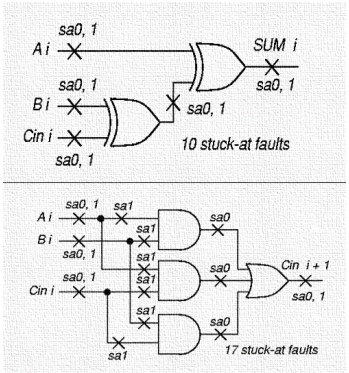


Figure 2 Full adder with stuck at faults

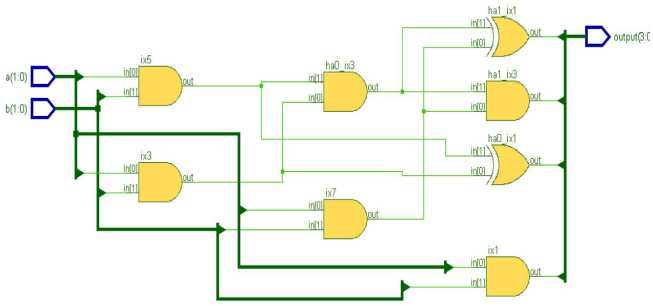


Figure 3 2-bit Multiplier stuck at faults.

Description of JTAG inserted into the VLSI System.

JTAG stands for Joint Test Action Group, which standardized interface used in digital circuits for testing, and debugging. It provides a way to access and control internal signals in a chip during manufacturing, testing, and operation. JTAG has four main components namely: TAP controller, Data register, Instruction Register, and Instruction decoder.

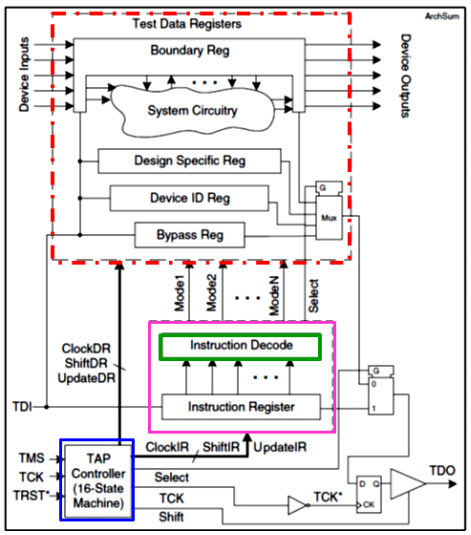


Figure 4 Architecture of JTAG standard

1. TAP (Test Access Port) Controller:
   TAP Controller is 16-state finite machine, which controlled by TMS (Test mode select) and TCK (Test Clock). TAP controller has four mandatory signals: TDI (Test data input), TDO (Test data output), TCK (Test Clock), and TMS (Test mode select). It also has one optional input signal TRST (Test Reset).

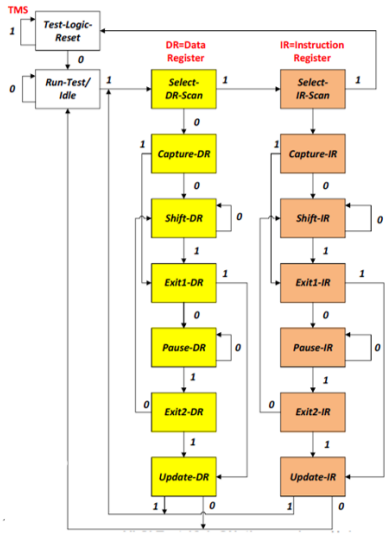


Figure 5 TAP Controller State Diagram.

2. Data Register
   In JTAG various data registers are used but Boundary scan register and Bypass register are mandatory.
   1. Boundary Scan Register:
      Boundary scan register is chain of Boundary scan cells (BSC). Architecture boundary scan cell are given below figure.
      Boundary scan cell are combined of 2 flipflops and 2 Mux. It has primary input, primary output, serial input and serial output. It has two control signals ShiftDR and Mode.

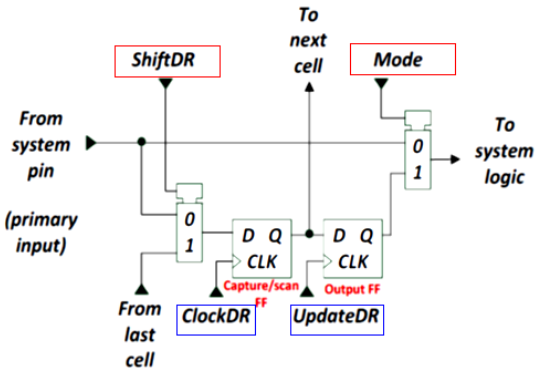
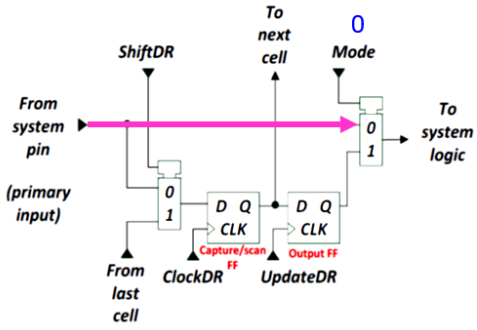


Figure 6 Boundary Scan cell

Boundary scan cell has four operations: Normal, Scan, Capture, and Update.

1. Normal mode:

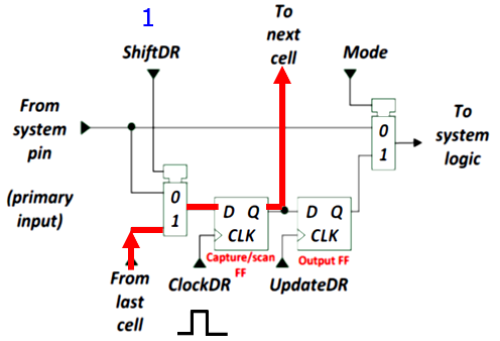
   In the "Normal" operational mode, the mode configuration is set to 1. This configuration ensures that the data input from "Data In" flows directly through and emerges as the output in "Data Out."
   In the "Shift" operational mode, the data transfer process occurs as follows: The "Scan In" (SI) input is shifted into the scan flip-flops (Scan F/F) through the activation of the "ClockDR" clock signal. During this process, the "ShiftDR" signal is maintained at 1, indicating the shift operation.



operation	Contr'l Signal		Clock
	Mode	ShiftDR	
Normal	0	X	system
Scan	X	1	ClockDR
Update	1	X	UpdateDR
Capture	X	0	ClockDR

Figure 7 Normal Mode BSC

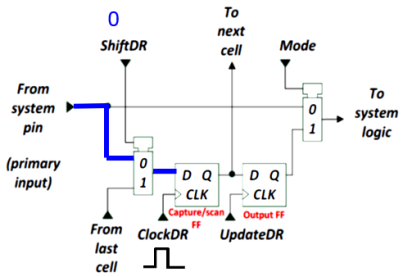
In the "Scan" operational mode, the mode setting is set to zero, and simultaneously, the "ShiftDR" signal is set to 1. This configuration establishes a scan path, enabling the data to move from the "Scan In" (SI) to the "Scan Out." This movement occurs when the clock signal for the scan flip-flops (Scan F/F) is triggered using the "ClockDR" signal.



operation	Contr'l Signal		Clock
	Mode	ShiftDR	
Normal	0	X	system
Scan	X	1	ClockDR
Update	1	X	UpdateDR
Capture	X	0	ClockDR

Figure 8 Scan Mode BSC

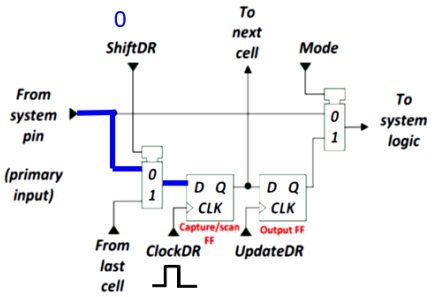
In the "Scan" operational mode, the mode setting is set to zero, and simultaneously, the "ShiftDR" signal is set to 1. This configuration establishes a scan path, enabling the data to move from the "Scan In" (SI) to the "Scan Out." This movement occurs when the clock signal for the scan flip-flops (Scan F/F) is triggered using the "ClockDR" signal.



operation	Contr'l Signal		Clock
	Mode	ShiftDR	
Normal	0	X	system
Scan	X	1	ClockDR
Update	1	X	UpdateDR
Capture	X	0	ClockDR

Figure 9 Capture BSC

In this mode, whatever data found in Capture/ Scan flipflop will be updated towards system logic.



operation	Contr'l Signal		Clock
	Mode	ShiftDR	
Normal	0	X	system
Scan	X	1	ClockDR
Update	1	X	UpdateDR
Capture	X	0	ClockDR

Figure 10 Update BSC

2. Bypass register
   Purpose of this circuit to provide shortcut from TDI to TDO. It is one bit Flipflop. When ShiftDR is 1 data shift from TDI to TDO. It will shorten boundary scan chain for chip under test.

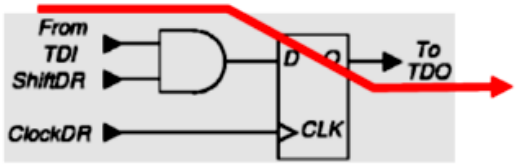


Figure 11 Bypass Register

3. Instruction Register
   The objective of this process is to enable the shifting of instructions from the Test Data Input (TDI) and storing these JTAG instructions within the Instruction Register (IR).

   The control signals associated with the Test Access Port (TAP), specifically "ShiftIR," "UpdateIR," and "ClockIR," play a crucial role. These signals serve to determine the operation being carried out. In cases where an instruction is being moved in and out through TDI and Test Data Output (TDO), the "ShiftIR" signal is activated. Conversely, if an instruction needs to be retained within the instruction register, the "UpdateIR" signal is employed. Finally, the "ClockIR" signal governs the clocking of data within the Instruction Register and its transmission to the instruction decoder.

   This sequence of actions effectively manages the flow of instructions, enabling them to be shifted, stored, and ultimately utilized by the instruction decoder for further processing.

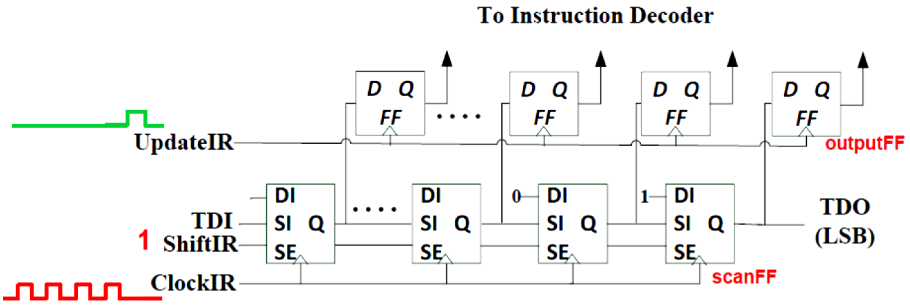


Figure 12 Instruction register.

JTAG has three mandatory instruction namely: Extest, Bypass, and Sample/ Preload.

1. Extest: Purpose is to test external off-chip wire interconnections among chips.
   Steps that are involved in Extest are:
   Step1: Scan in chip#1: shiftDR=1, clockDR…
   Step2: Update chip#1: updateDR
   Step3: Capture chip#2: shiftDR=0, clockDR
   Step4: Scan out chip#2: shiftDR=1, clockDR…
   Mode always = 1

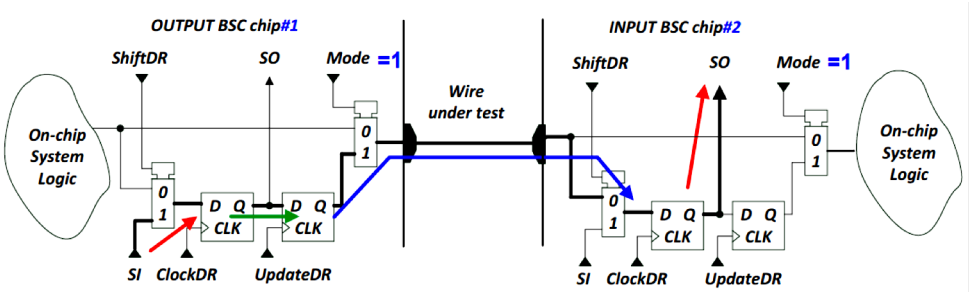


Figure 13 Extest instruction

2. Bypass
3. Sample/ Preload
   Step1: Capture: ShiftDR=0, ClockDR
   Step2: Scan out: ShiftDR=1, ClockDR …
   Mode = 0 to isolate system logic
   Does not interfere with normal operation of system logic.

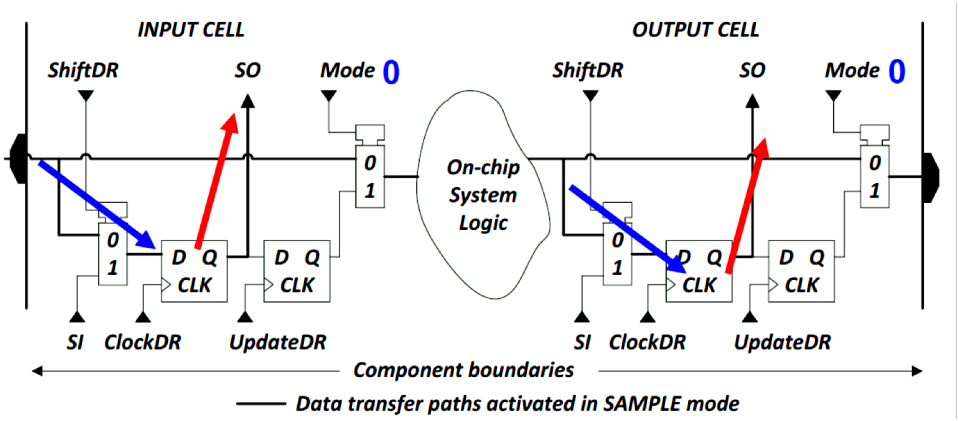


Figure 14 Sample instruction

Preload:
Step1:Scan in: ShiftDR=1, ClockDR…
Step2: Update: UpdateDR
Mode = 0 to isolate system logic

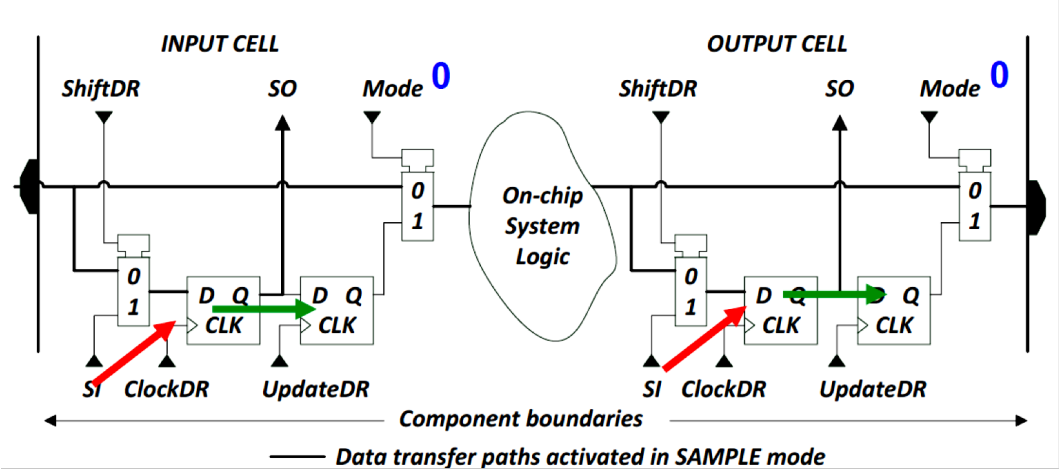


Figure 15 Preload Instruction

4. Instruction Decoder
   The primary function of the instruction decoder is to interpret the instructions received from the Instructions Register. The instruction decoder's task involves deciphering the content of these instructions and subsequently generating appropriate control signals based on their content.

RTL synthesis of various built combinational components

In the regular operational mode, utilizing the Precision synthesis Tool, we successfully conducted the synthesis process for both the design files and the additional files related to Boundary Scan. The synthesis of the 8-bit Multiplier was carried out prior to incorporating the Boundary Scan, as depicted in the following figure-18.

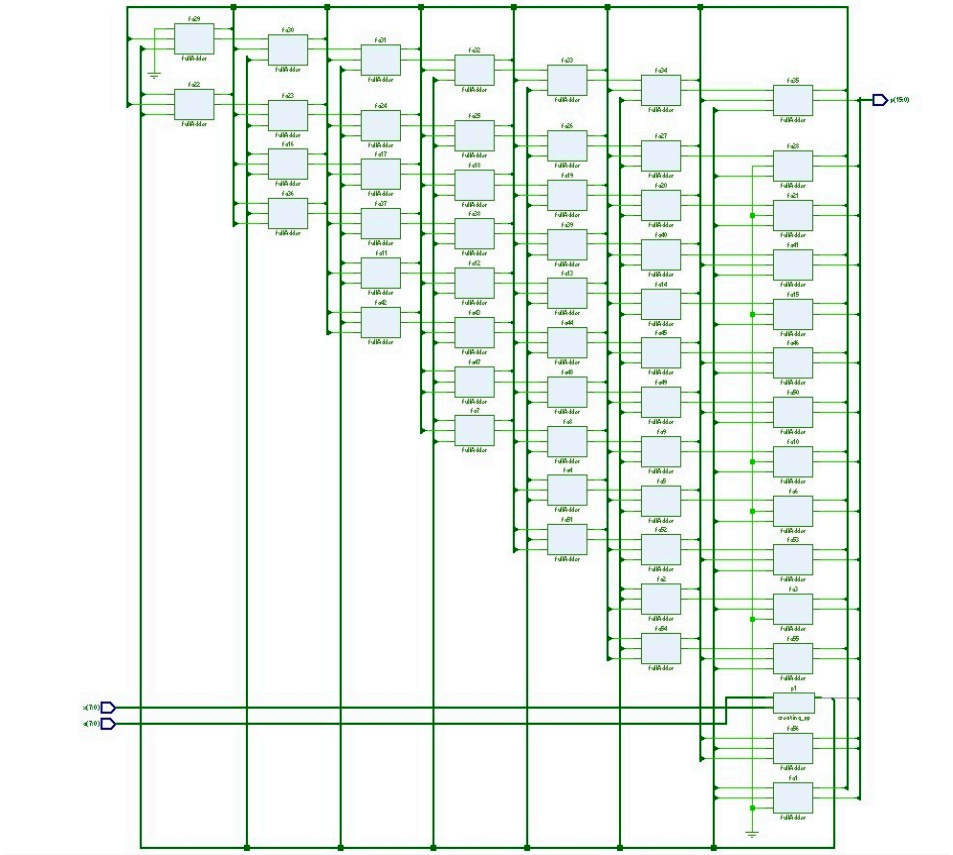


Figure 16 RTL Schematic for the 8-Bit Multiplier

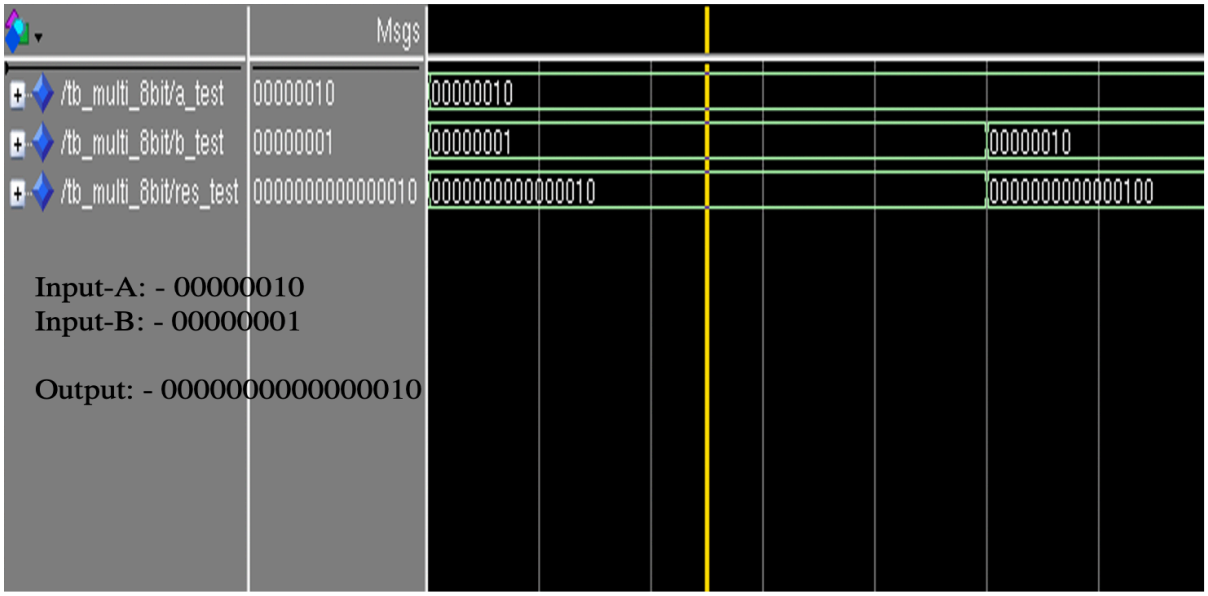


Figure 17 Results of 8-bit Multiplier

Boundary Scan Register (BSR): -

The Boundary Scan Register's RTL synthesis involves two multiplexers and two flip-flops, as illustrated in figure 19. The Boundary Scan cell operates in four distinct modes: Normal, Scan, Update, and Capture, determined by control signals received from the TAP controller and Instruction Decoder, as indicated in Table 1.

Operation	**Control Signal**		**Clock**
	Mode	**ShiftDR**	
Normal	0	X	System Clock
Scan	X	1	ClockDR
Update	1	X	UpdateDR
Capture	X	0	ClockDR

Table 1 Boundary Scan Cell Operations

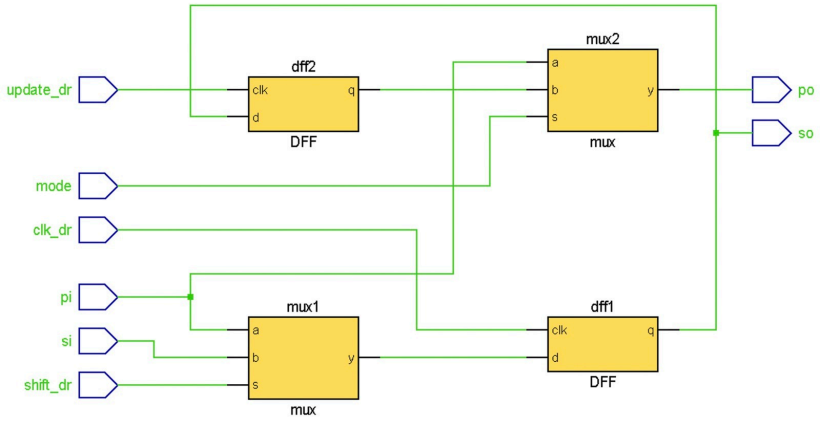


Figure 18 Boundary Scan Cell Schematic

Instruction Register (IR): -

The Instruction Register Cell's RTL synthesis involves one multiplexer and two flip-flops, as depicted in figure 20. The loading of instructions into the Instruction Register is determined by control signals originating from the TAP controller. When ShiftIR and ClockIR signals are active, the instruction is loaded into the register. Upon enabling UpdateIR, the loaded instruction is subsequently forwarded to the instruction Decoder.

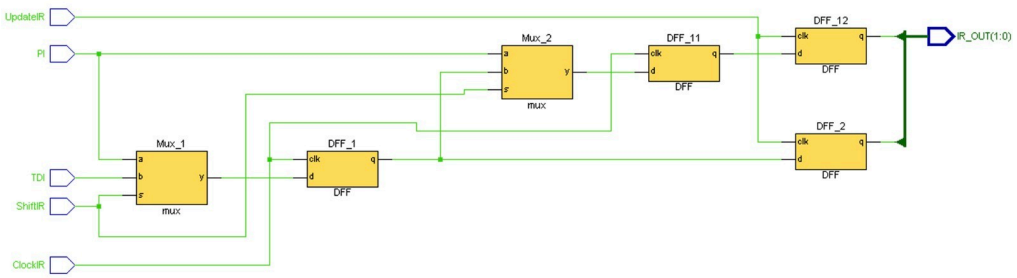


Figure 19 Schematic of Instruction Register

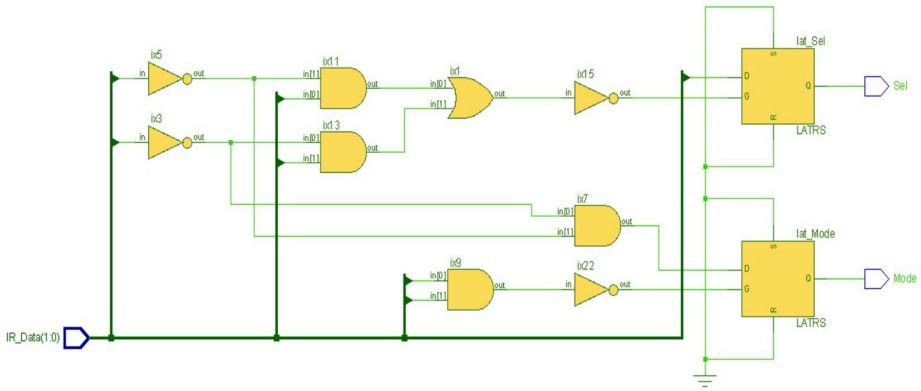


Figure 20 Schematic of Instruction Decoder.

Bypass Register (BR): -

The Bypass Register's RTL synthesis involves an AND gate and a flip-flop, illustrated in figure 22. The behavior of the Bypass Register is determined by control signals originating from the TAP controller. When ShiftDR and ClockDR signals are active, the Bypass Register can direct the TDI signal to TDO, while in the case of ShiftDR being disabled, it can isolate the TDI from TDO.

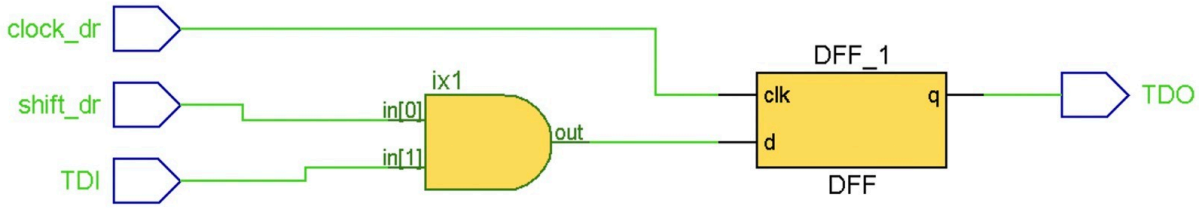


Figure 21 Schematic of Bypass Register.

TAP Controller: -

The TAP (Test Access Port) Controller serves as the core of the JTAG/IEEE 1149.1 standard. It plays a crucial role in producing control signals for various modules within the JTAG system. Operating with a 16-state transition sequence, the TAP Controller facilitates loading instructions into the instruction Register and performing actions such as reading the Boundary Scan cell, including Scan-in, Scan-out, and Capture processes as depicted in figure 23.

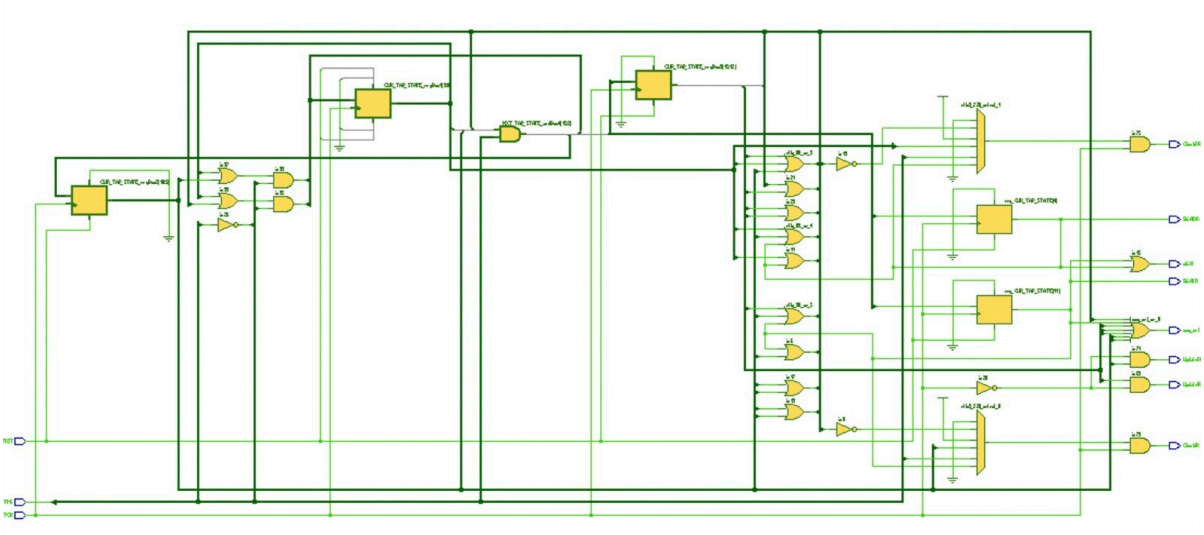


Figure 22 Schematic of TAP Controller.

Integration of JTAG with the Multiplier: -

The RTL synthesis of the integrated JTAG and Multiplier involves multiple components, with the primary ones being the Boundary Scan Block, Instruction Register, TAP Controller, Bypass Register, and the 8-bit Multiplier Design file, depicted in Figure 24. The functioning of JTAG depends on inputs like TDI, TMS, TCK, and System pins, enabling operations such as Normal, Scan-in, Scan-out, and Capture.

The synthesis process for the design files with the added Boundary Scan files is illustrated below.

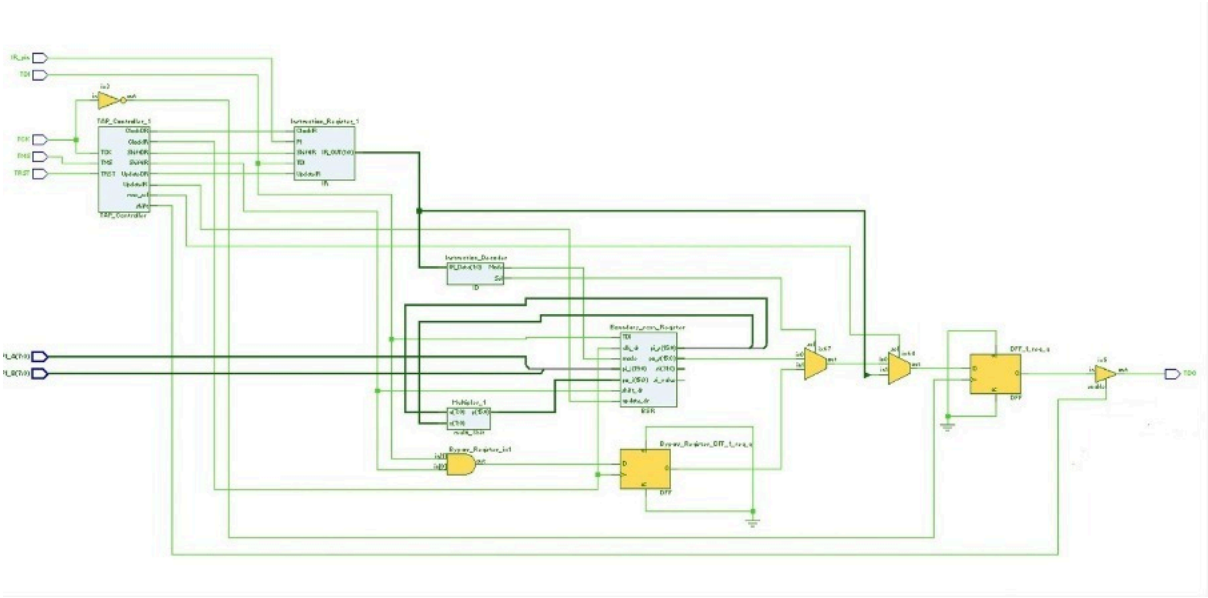


Figure 23 Schematic of the JTAG integrated with the Multiplier Design.

Description of overhead and benefits of the added JTAG

Overhead Impact: -
The RTL synthesis of the integrated JTAG and Multiplier involves several key modules, with significant components including the Boundary Scan Block, Instruction Register, TAP Controller, Bypass Register, and the 8-bit Multiplier Design file, as illustrated in Figure 24.

The JTAG's functionality, depending on inputs like TDI, TMS, TCK, and System pins, encompasses operations such as Normal, Scan-in, Scan-out, and Capture. While JTAG offers numerous advantages, it does come with certain trade-offs:

Area Impact: The area overhead, demonstrated in the figure, is calculated as (38-32)/32, equaling 18.75%, (Area Overhead (%) = ((Modified Design Area - Original Design Area) / Original Design Area) * 100).

Although this percentage seems substantial in this example due to the small system logic but according to the industry standards its way less than the one which we are getting.

Performance Impact: The addition of the Boundary Scan Chain (BSC) lengthens the critical path, leading to a delay in circuit performance.

Additional Effort and Costs: Integrating JTAG introduces extra effort and costs during design and implementation.

Power Impact: There's an increase in power consumption due to the added JTAG components.

Pin Increase: Additional pins like Scan In, Scan Out, and ShiftDR are required, leading to an increase in pin count.
It's always important to weigh the overheads against the benefits of JTAG, which include improved testability and debugging capabilities.

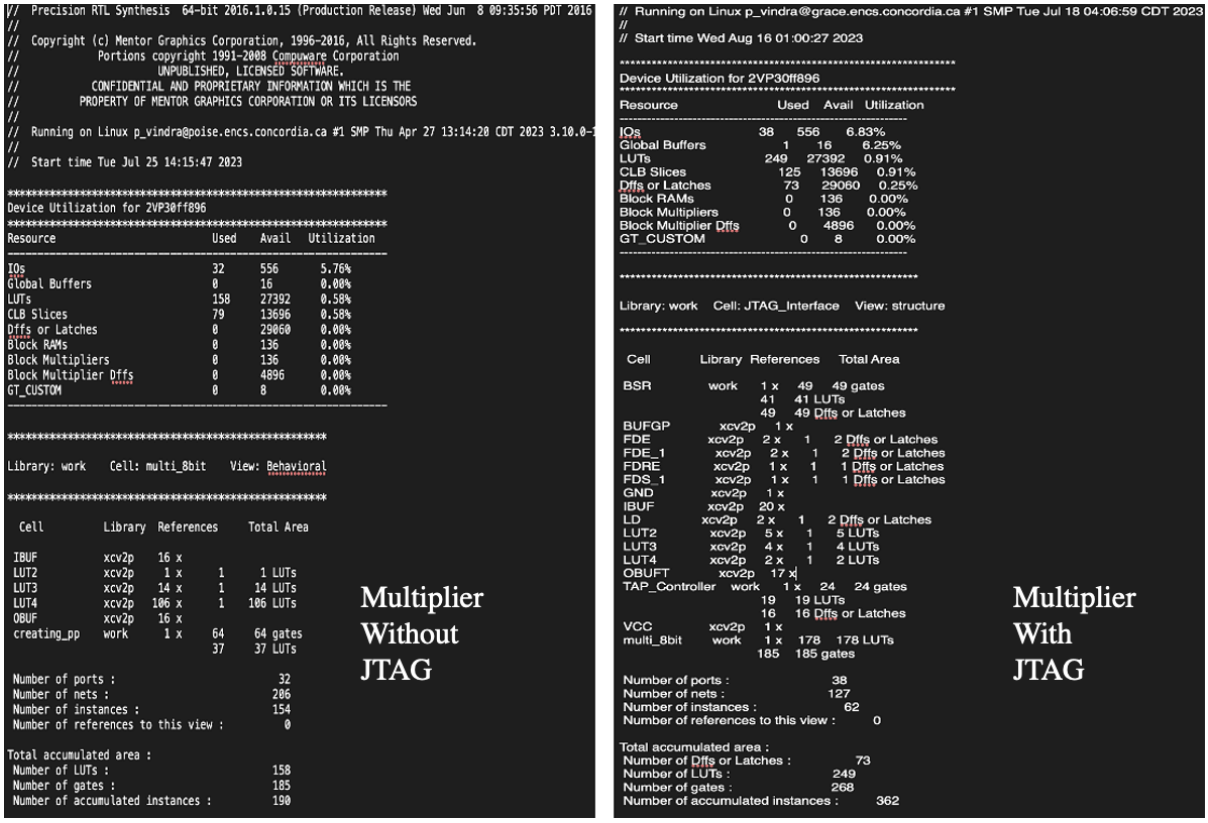


Figure 24 Area Report for Multiplier w/o JTAG and with JTAG

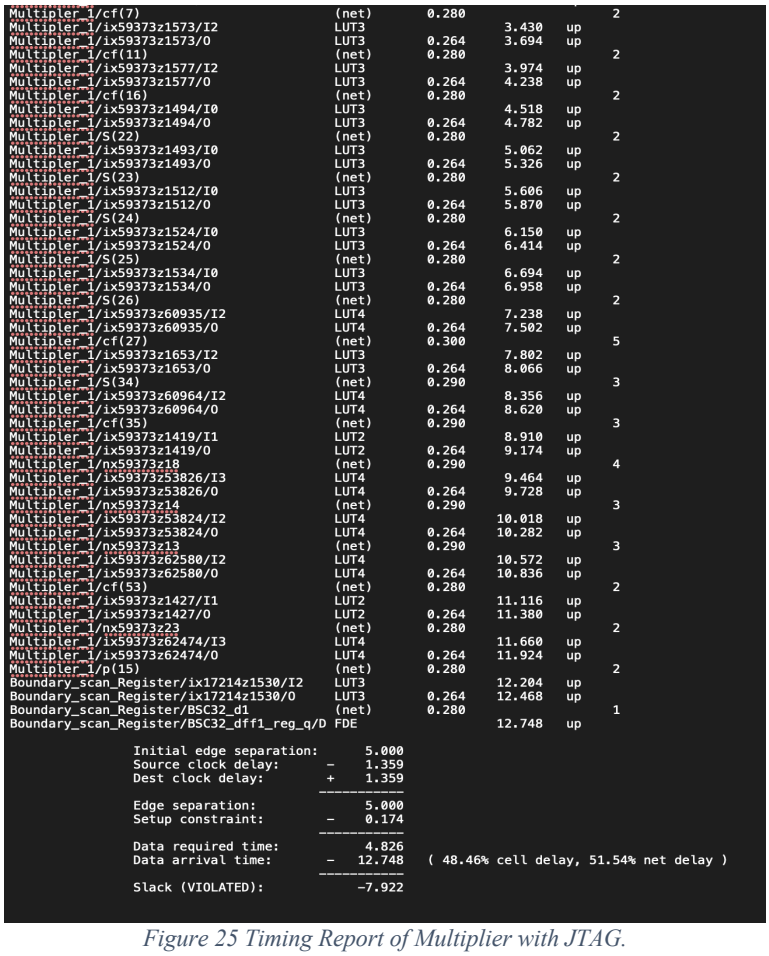


Figure 25 Timing Report of Multiplier with JTAG.

Benefits of JTAG: -
Despite the various overheads that JTAG introduces to design, area, and power aspects, it offers enhanced testability by not requiring knowledge of the internal design structure (External Scan). This characteristic has led to its application in various testing scenarios:

- Board-Level Testing and Diagnosis: JTAG is utilized in scenarios where multiple chips are present on the same board. These chips are interconnected in a JTAG SCAN CHAIN, enabling simultaneous testing of all chips.
- Interconnect Testing Among Chips: JTAG's ability to apply test patterns through TDI and receive responses at TDO allows for the testing of interconnections between different chips.
- Testing On-Chip System Logic: JTAG is effective for testing individual chips within a larger context, especially in cases where a board houses multiple chips.

In essence, while JTAG does introduce certain drawbacks, its ability to enhance testability and offer efficient testing methods has made it a valuable tool in various testing applications.

Description of testing the VLSI using the JTAG

For VLSI testing using JTAG, the process involves incorporating Boundary Scan Cells (BSC) to both inputs and outputs. These BSC units are interconnected, forming a shift register arrangement known as the JTAG SCAN CHAIN.

Instruction	Instruction Encoding
EXTEST	00
BYPASS	11
SAMPLE	01
PRELOAD	10

Table 2 Opcodes for the Instructions EXTEST, SAMPLE, BYPASS and PRELOAD

EXTEST Instruction:

The EXTEST instruction serves to evaluate the connectivity of the chip with external off-chip wires. Through the EXTEST Instruction, direct linkage is established between TDI (Test Data Input) and TDO (Test Data Output) by means of the Boundary Scan cell Register.

The initialization of the Extest operation encompasses three steps:

1. **Loading Extest Instruction into Instruction Register:**

To load the Extest instruction into the JTAG system, the process involves transitioning the TAP controller to the ShiftIR state. This state must be maintained for two clock cycles, aligning with the **2-bit length** of the instruction register. During this state, the TDI signal is set to 00. The procedure for loading the EXTEST instruction is presented in the following table-3

	Initialize TAP		Load Instruction	
TMS	11111	01100	00	110
TDI	xxxxx	xxxxx	00	xxx
TDO				
Final State	Reset		Shift-IR	RUN-Test

Table 3 EXTEST - Loading Instructions.

2. Scan In and Update:

To load the output boundary cell of the first JTAG device, the process involves transitioning the TAP controller to the ShiftDR state. This state is maintained until all output boundary cells are loaded with corresponding TDI values. The complete operation of Scan In and Update is detailed in the following table-4.

		Scan In		Update	
TMS	110	00000000000000000000000000000000	1	1	1
TDI	xxx	11111111111111110000000000000000	x	x	x
TDO					
Final State		Shift-DR		Update-DR	Select-DR

Table 4 EXTEST Instructions for Scan in and Update.

3. Capture and Scan Out:

For capturing the values of the output boundary cells from the first JTAG device and transmitting them to the second JTAG device, the TAP controller is transitioned into the Capture and Scan Out state. This state is maintained until all the values have been received. The complete procedure for the Capture and Scan Out operation is presented in the following table 5.

	Capture		Scan-Out	
TMS	0	0	0000000000000000	110
TDI	x	x	xxxxxxxxxxxxxxxx	xxx
TDO			1111111111111111	
Final State	Capture-DR		Shift-DR	RUN-Test

Table 5 EXTEST Instruction for Capture and Scan out.

Upon applying the TMS, TDI, and TCK signals for the EXTEST Instruction, we proceed to capture the TDO at the first JTAG device. We have effectively managed to capture the output of the Boundary Scan Cell Register, as illustrated in Figure.

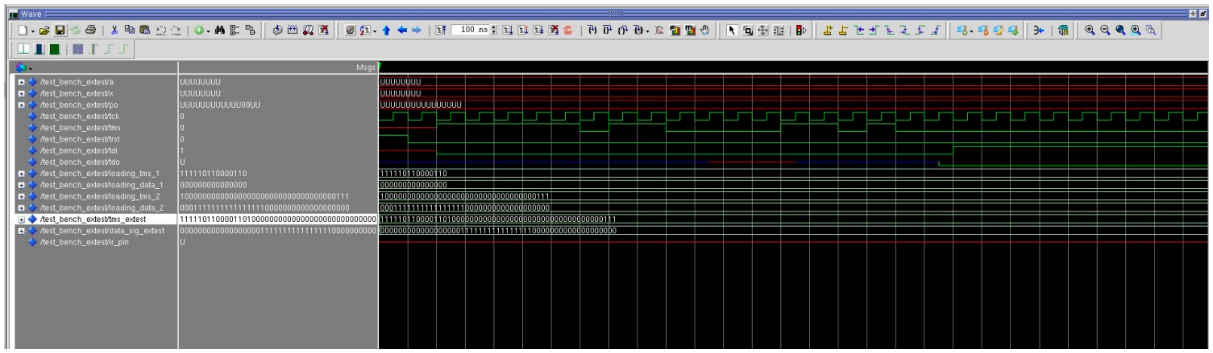


Figure 26 EXTEST Instruction Execution

Bypass Instruction:

The Bypass instruction involves passing a single bit through a Bypass register, which connects the TDI and TDO lines. This instruction facilitates the analysis of other interconnected JTAG devices without introducing additional overhead.

There are three steps in initializing the Bypass operation:

1. **Loading Bypass Instruction into the Instruction Register:**

To load the Bypass instruction into the JTAG system, the TAP controller must transition to the ShiftIR state. This state is maintained for two clock cycles, aligning with the 2-bit length of the instruction register. During this state, the TDI signal is set to 11. The procedure for loading the Bypass instruction is outlined in the following table-6.

	Initialize TAP		Load Instruction	
TMS	11111	01100	00	110
TDI	xxxxx	xxxxx	11	xxx
TDO				
Final State	Reset		Shift-IR	RUN-Test

Table 6 BYPASS Instruction Loading

2. **Scan In and Update:**

For bypassing the value from TDI to TDO, the TAP controller needs to transition into the ShiftDR state. This state is maintained until the values have been successfully bypassed. The complete operation of Scan In and Update in the context of the Bypass instruction is provided in the following table-7.

		Scan In		Update	
TMS	100	00000000000000000000000000000000	1	1	1
TDI	xxx	11111111111111110000000000000000	x	x	x
TDO					
Final State		Shift-DR		Update-DR	Select-DR

Table 7 BYPASS Instruction Scan In and Update

3. **Capture and Scan Out:**

For capturing the values in other interconnected JTAG devices, the TAP controller should be shifted into the Capture and Scan Out state. This state is maintained until all the values have been successfully received. The complete procedure for the Capture and Scan Out operation in the context of the Bypass instruction is provided in the following table 8.

	Capture		Scan-Out	
TMS	0	0	00000000000000000000000000000000	110
TDI	x	x	xxxxxxxxxxxxxxxx	xxx
TDO			11111111111111110000000000000000	
Final State	Capture-DR		Shift-DR	RUN-Test

Table 8 BYPASS Instruction for Capture and Scan Out

Following the application of TMS, TDI, and TCK signals for the Bypass Instruction, we proceed to capture the TDO at the first JTAG device. Our success in bypassing the TDI to TDO is clearly illustrated in Figure 26.

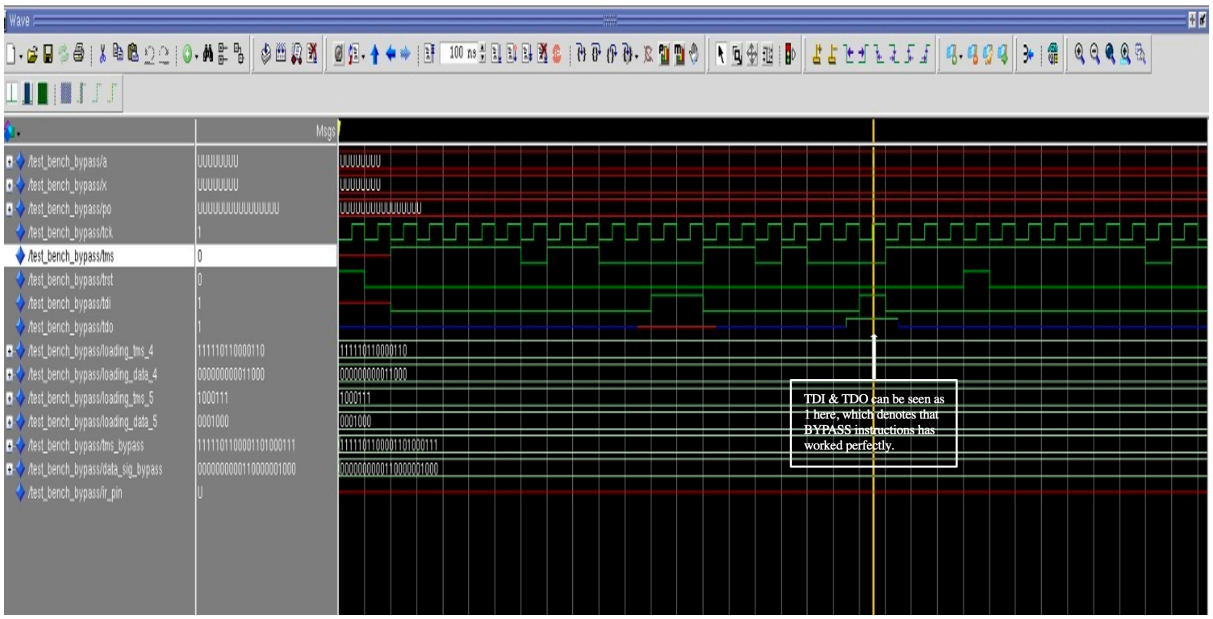


Figure 27 BYPASS Instruction Simulation

Sample Instruction:

The main purpose of the Sample instruction is to capture a snapshot of the system's input/output pins.

There are two steps involved in initializing the Sample operation.

1. **Loading Sample Instruction into Instruction Register:**

To load the Sample instruction into the JTAG system, the TAP controller must transition to the ShiftIR state. This state is maintained for two clock cycles, aligned with the 2-bit length of the instruction register. During this state, the TDI signal is set to 01. Additionally, the input values of the Multiplier, represented by 'a' and 'x', are set to ___and ___, respectively. The process for loading the Sample instruction is illustrated in the following table 9.

	Initialize TAP		Load Instruction	
TMS	11111	01100	00	110
TDI	xxxxx	xxxxx	01	xxx
TDO				
Final State	Reset		Shift-IR	RUN-Test

Table 9 SAMPLE Loading Instructions.

2. **Capture and Scan Out:**

To capture the output of the multiplier, the TAP controller needs to transition into the Capture and Scan Out state. This state is maintained until all input and output pin values have been successfully captured. The complete procedure for the Capture and Scan Out operation is presented in the following table 10.

		Capture	Scan In		Update	
TMS	10	0	00000000000000000000000000000000	1	1	0
TDI	xx	x	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	x	x	x
TDO						
Final State		Capture	Shift-DR		Update-DR	RUN-Test

Table 10 SAMPLE Capture and Scan-Out Instructions.

Following the application of TMS, TDI, and TCK signals to the TAP, along with setting 'a' and 'x' values for the multiplier in the context of the Sample Instruction, we successfully capture all input and output pin values. These values are depicted in Figure 27 , with output ports registering a value of ___ and input ports registering ___and __.

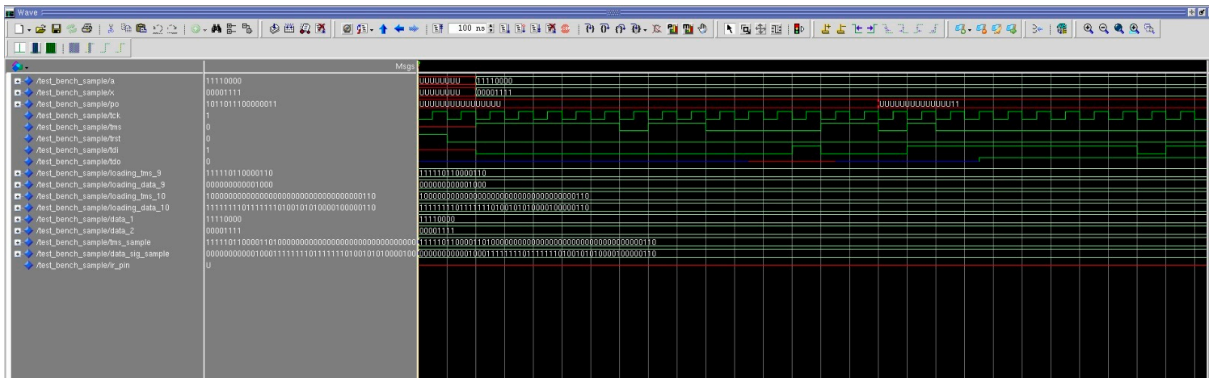


Figure 28 SAMPLE Instruction Execution.

Preload Instruction:

The core purpose of the Preload Instruction is to exercise control over all input and output pins within the system.

There are two steps involved in initializing the Preload operation:

1. **Loading Preload Instruction into Instruction Register:**

To load the Preload instruction into the JTAG system, the TAP controller must transition to the ShiftIR state. This state is sustained for two clock cycles, in line with the 2-bit length of the instruction register. During this state, the TDI signal is set to __. The procedure for loading the Preload instruction is outlined in the following table.

	Initialize TAP		Load Instruction	
TMS	11111	01100	00	110
TDI	xxxxx	xxxxx	10	xxx
TDO				
Final State	Reset		Shift-IR	RUN-Test

Table 11 PRELOAD loading Instructions.

2. **Scan In and Update:**

For loading all the inputs and outputs of the multiplier, the TAP controller should transition into the Scan In state. This state is maintained until all the I/O pins have been successfully loaded. Subsequently, the system should transition into the Update state to ensure the values are loaded into the I/O ports.

		Scan In		Update	
TMS	100	00000000000000000000000000000000	1	1	0
TDI	xxx	11111111111111111111111111111111	x	x	x
TDO					
Final State		Shift-DR		Update-DR	RUN-Test

Table 12 PRELOAD Scan in and Update Instructions

Upon applying TMS, TDI, and TCK signals to the TAP for the Preload Instruction, we have effectively loaded all input and output values, as depicted in Figure 28.

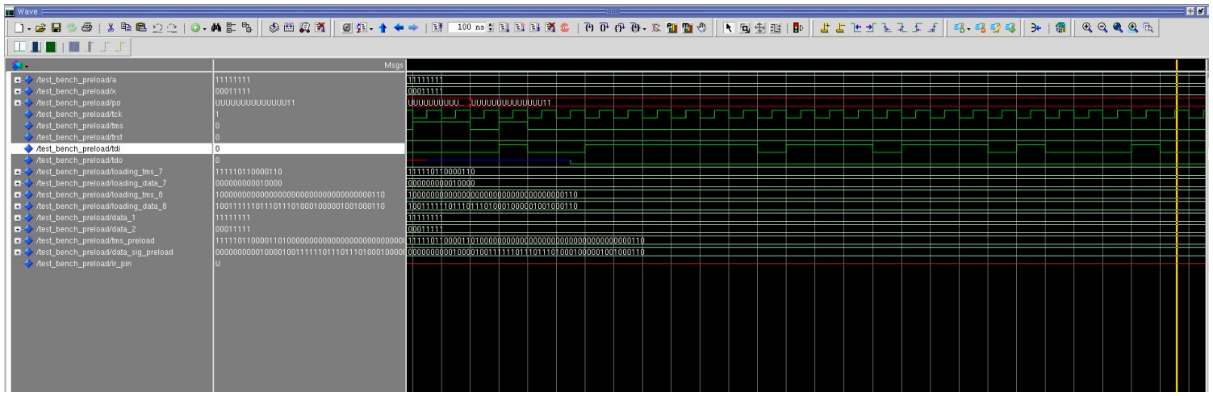


Figure 29 PRELOAD Instruction Execution

Conclusions (Summary, Discussion, Challenges)

Challenges & Discussion: -
Over the course of the project period, we were introduced slowly to the concepts required for the project such as boundary scan. So, during the initial period we found it really hard to implement the TAP controller as we weren't able to comprehend the concepts as efficient as possible since it involved 16-state machine and we were tasked to perform shift, update, capture and normal operation using it.

Apart from the TAP controller, what really hindered our progress was the integration of JTAG with the given design i.e. 8-Bit Multiplier design because it required the top module to be port mapped with respective components in the JTAG, which our utmost attention and patience.

Summary: -
In brief, the project aims to integrate and implement JTAG into a system logic while assessing the benefits it brings in comparison to the associated overheads in terms of area, power, and performance. The process commenced by synthesizing the existing "8-bit Multiplier" system logic, followed by integrating the main components of JTAG and its essential instructions. Individual testbenches were developed for each component and the entire circuit was comprehensively tested. Subsequently, JTAG was integrated into the system logic, leading to another synthesis round, enabling a direct comparison of area, power, and performance metrics.

References

[1]. "Technical Guide to JTAG - XJTAG Tutorial," XJTAG. https://www.xjtag.com/about-jtag/jtag-a- technicaloverview/
[2] IEEE 1149.1 JTAG AND BOUNDARY SCAN TUTORIAL
[3] http://cc.ee.ntu.edu.tw/~cmli/VLSItesting/
[4] Prof Dr. Mahmoud Masadeh, COEN 6521 Design for Testability, Lecture notes, Summer term 2023.