Report
Confirmation of Originality
Faculty of Engineering and Computer Science

Course Name & Number/Term: COEN 6521 **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: Thenmozhie Rajan   ID No: 40192527   Signature: _Thenzhie.R_   Date: 01-08-2022
(please print clearly)

Name: Ragul Nivash Rangasamy Sekar   ID No: 40169564   Signature: _R.S. Ragul Nivash_   Date: 01-08-2022
(please print clearly)

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

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

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

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

Do Not Write in this Space – Reserved for Instructor

COEN6521 Design for Testability

Project Report
Design Testing of 8-Bit Multiplier Using JTAG

Submitted to
Dr. Mahmoud Masadeh

Submitted by
Group 2

Ragul Nivash Rangasamy Sekar (**40169564**)
Thenmozhie Rajan (**40192527**)

DATE: 06-08-2022

Abstract

Design for testability (DFT), is a design strategy that, by including extra circuitry on the chip, makes testing a chip feasible and affordable. It also enhances the controllability and observability of internal nodes to allow the testing of embedded functionalities. This project aims to test an 8-bit array multiplier which is a combinational circuit by incorporating JTAG IEEE std 1149.1 (boundary-scan) within it.

The following tools are used in this project,
- Mentor Graphics VHDL Modelsim tool
- Mentor Graphics Precision RTL Logic Synthesis tool

This project report provides brief description about,
  i. Designed VLSI system (8-bit multiplier)
 ii. Testing the VLSI system (structural vs behavioral)
iii. JTAG inserted into the VLSI system.
 iv. RTL synthesis of various built combinational components.
  v. Overhead and benefits of the added JTAG.
 vi. Testing the VLSI using the JTAG.

Table of Contents

1. Introduction

Complex chips are being planned, developed, and manufactured as advances in integrated circuit (IC) manufacturing technology continue to lower defect density and minimum feature size. It can be difficult to ensure that a VLSI circuit is fault-free, and the time and effort required to test for issues can dramatically raise the cost of IC fabrication. The significant time and labor costs associated with creating test vector sequences for VLSI circuits are attempted to be reduced by design-for-testability (DFT) methodologies. Board-level test issues were first resolved by the Joint Test Action Group (JTAG). This project aims to create JTAG for testing 8-bit array multiplier. The JTAG design includes Test Access Ports (TAP), Data Registers (Boundary Scan Register, Bypass Register), Instruction Register, Instruction Decoder, TAP Controller.

2. Designed VLSI system (8-bit array multiplier)

The 8-bit combinational array multiplier has two 8-bit values, multiplies them, and gives 16-bit output. The multiplier has 64 AND gates and 56 full adders in its construction. These are arranged in an array fashion, with the input of the subsequent row of full adders being the nth output from the first row's full adders. An output joins the product when it appears at the start of a row.

3. Testing the VLSI system

Between 60 and 80 percent of the design process is spent testing. To achieve high yield and accurate detection of defective chips after production, a well-structured testing procedure must be used.

3.1. Structural Testing

Structural test is a testing procedure that finds all chip structure faults. Stuck-at faults, coupling faults, short-circuits, crosstalk, etc. are a few examples. The structural test offers a way to check for even unmodeled manufacturing flaws. All possible directions of datapath can be tested during structural level testing. The process variation-aware test values aid in the detection of flaws even when process variations are present.

Without considering the circuit's overall functionality, structural testing aims to find manufacturing flaws by validating the correct operation of a circuit's underlying

gates and connections. A structured DFT is being used to analyze the circuit, by evaluating the internal system and accepting control input. There are several ways to carry this out. Scan design is a popular and helpful approach that may be used to change the internal sequential circuitry architecture.

3.2. Functional Testing

Even if a circuit passes a structural test, it does not guarantee that it will operate as expected. A functional test is a testing procedure that confirms the tested circuit performs as intended. Only the functional behavior might be understood by the users in VLSI devices; the precise circuit implementation is always unknown. Based on the functional behavior of the circuit being tested, a systematic method may be utilized to identify and locate stuck-at and bridging faults on the primary input and output lines.

4. JTAG

JTAG is an industry standard for verifying designs and testing printed circuit boards after production. It was called after the Joint Test Action Group, which defined it. As an additional tool to digital simulation in electronic design automation (EDA), JTAG provides standards for on-chip instrumentation.

JTAG components,
- Test Access Port (TAP)
- TAP controller
- Registers
  - Instruction Register,
  - Boundary Scan Register,
  - Bypass Register
- Instruction Decoder

4.1. TAP

JTAG offers four mandatory pins in the configuration as follows,
- TDI (Test Data Input),
- TDO (Test Data Output),
- TCK (Test Clock),
- TMS (Test Mode Select)

These pins are collectively referred to as the Test Access Port (TAP).

4.2. TAP Controller

A 16-state machine called a TAP controller is programmed by the inputs Test Mode Select (TMS) and Test Clock (TCK), and it regulates the flow of data bits to the Instruction Register (IR) and the Data Registers (DR).

On the positive edge of TCK, bits are shifted in, and on the negative edge, they are moved out. The register into which the bits are moved is controlled by the TMS signal (instruction register, bypass register or boundary scan register). A register is captured, a new value is shifted in from TDI while the old value is simultaneously shifted out on TDO, and then the register is updated with the new value. The instruction register or one of the other registers can be used by the TAP controller to shift values.
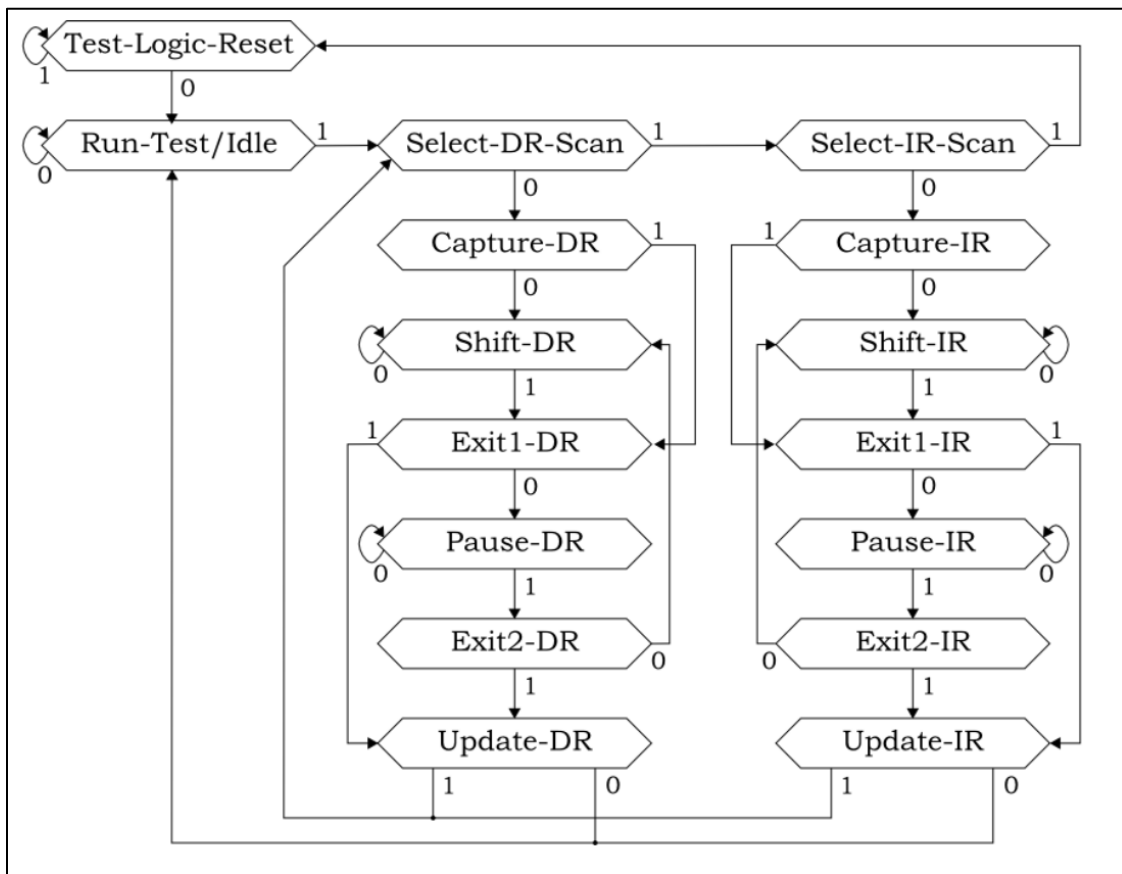
Figure 1 : TAP Controller Finite State Machine

On the rising edge of TCK, TMS is sampled and used to advance the state. The following steps were taken in each state:

Test-Logic-Reset
All test modes, including extest mode, are reset in this condition, disabling them from operating and enabling the chip to continue operating normally. The external

logic will force TMS high on startup for at least five TCK cycles. This ensures that the Test-Logic-Reset condition is reached and maintained.

Run-Test/Idle
When everything is running normally, this is the resting state.

Select-DR-Scan, Select-IR-Scan
These are the corresponding beginning states for accessing the instruction register or one of the data registers.

Capture-DR, Capture-IR
These capture the current value of the instruction register or one of the data registers into the scan cells, respectively. This is a little misnomer for the instruction register since, when using Capture-IR, status information is often captured rather than the actual instruction.

Shift-DR, Shift-IR
Shift a bit from the presently chosen data or instruction register into TDI (on the rising edge of TCK) and out onto TDO (on the falling edge of TCK), accordingly.

Exit1-DR, Exit1-IR
These are the shift states' respective exit states. From this point, the state machine has two options: update state or pause state.

Pause-DR, Pause-IR
Data shifting into the instruction or data register is paused. This enables, for instance, the reloading of buffers on test equipment that supplies TDO.

Exit2-DR, Exit2-IR
These are the equivalent pause state's departure states. The state machine can then either start shifting again or go into the update stage.

Update-DR, Update-IR
The chip's inputs or the connection are used to drive the value that was moved into the scan cells during the previous states (for outputs). With the ability to pause during the shifting, our basic state machine enables either data registers or the instruction register to complete its capture-shift-update cycle.

4.3.Register
A minimum of two data registers, the boundary scan register, and the bypass register, must be included in a boundary scan logic design.

4.3.1. Boundary Scan Data Register
The Boundary Register, which has a boundary-scan cell next to each input and output pin, is the most significant. The device's input and output pins can be controlled and monitored using this register. IEEE 1149.1 requires the Boundary register as a feature. By entering 0000 into the instruction register, the boundary scan data register is chosen. According to the requirements of the IEEE 1149.1 standard, the boundary scan register is operated by the TAP controller's Shift-DR, Update-DR, and Capture-DR states.

Each of the processor interface pins may be accessed serially using the boundary scan register. Therefore, it is possible to load and monitor logic values on the processor pins using the boundary scan register. Board-level connection testing is the primary use of the boundary scan register.

4.3.2. Bypass Register
Bypass register provides a direct link between TDI and TDO, bypassing the device. A single bit makes up the bypass register. Data is sent from TDI to TDO with a one TCK cycle delay when the BYPASS instruction is the current instruction in the IR: in the Shift-DR mode. The Bypass Register shortens the shift route between the test data input (TDI) and test data output (TDO) of a boundary scan device with a fixed binary "0" output when it is activated by the Bypass Instruction.

4.3.3. Instruction Register and Instruction Decoder

The instruction register must be at least 2 bits long. During the Update-IR state, the instruction is given to an instruction decoder. The TAP controller state machine defines the conditions under which the instruction decoder interprets and executes the instructions. IEEE 1149.1 mandates a minimum of 4 instructions:

BYPASS
Utilize the bypass entry to capture, move, and update data. This enables the chip to carry on with its routine operations. This instruction must include just 1s according to IEEE 1149.1.

SAMPLE

Data entering and leaving the chip through its inputs and outputs can be sampled by capturing and shifting data through the boundary scan register. The update phase, however, does not force data onto inputs or outputs.

PRELOAD

Shift information across the boundary scan register to create a value in the scan cells that may be used in the future. The previous value is not entered into the cell for this instruction during the capture phase, and neither is data loaded into the inputs or outputs during the update phase. This instruction was paired with SAMPLE in early versions of the standard.

EXTEST

Before data is recorded, shifted, and updated through the boundary scan register, the semiconductor is put into EXTEST mode. This is employed to check the connection of several semiconductors. The chip does not attempt to drive outputs or take inputs when in EXTEST mode. Prior to EXTEST, it is typical to use PRELOAD to configure the boundary scan register.

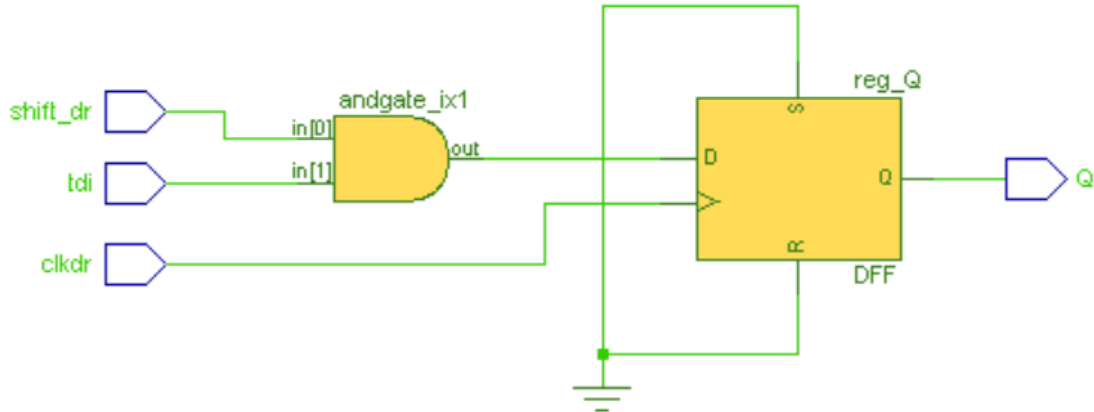## 5. RTL synthesis of various built combinational components

5.1. RTL synthesis of Multiplexer
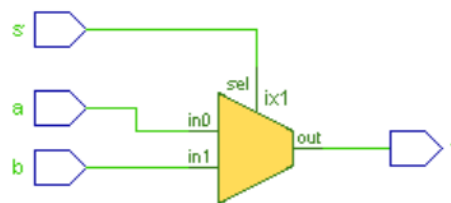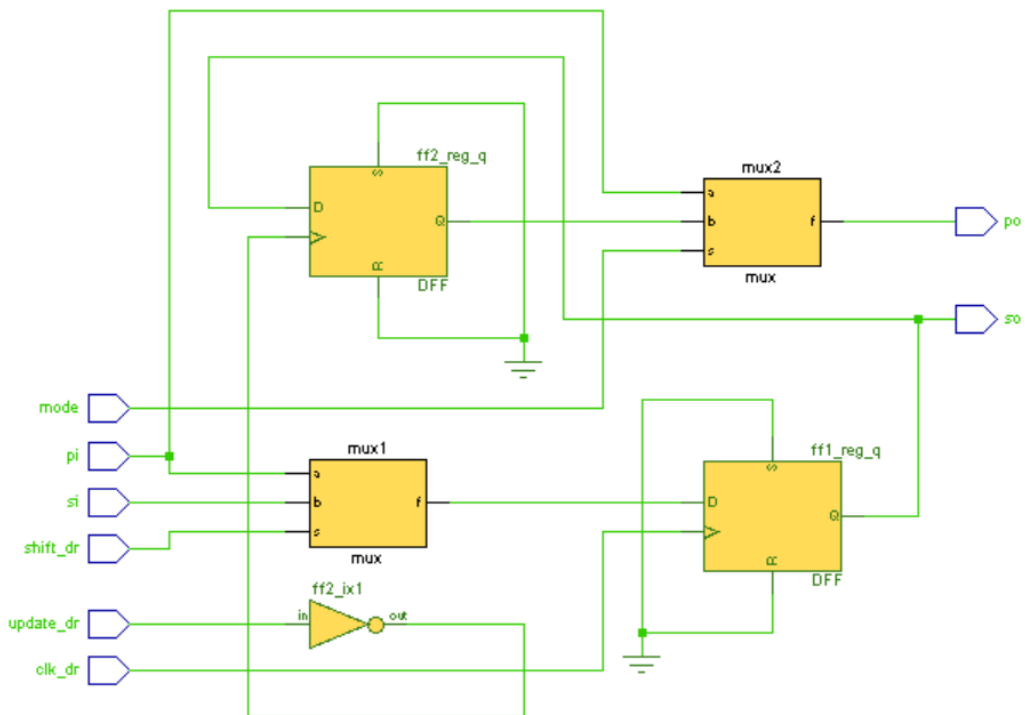

5.2. RTL synthesis of D-Flipflop for raising and falling edge
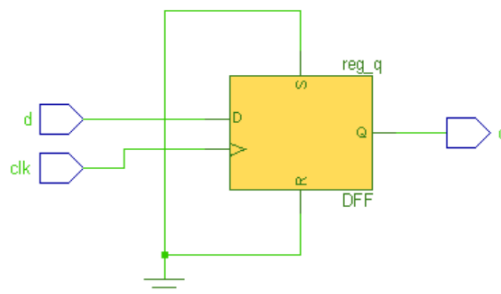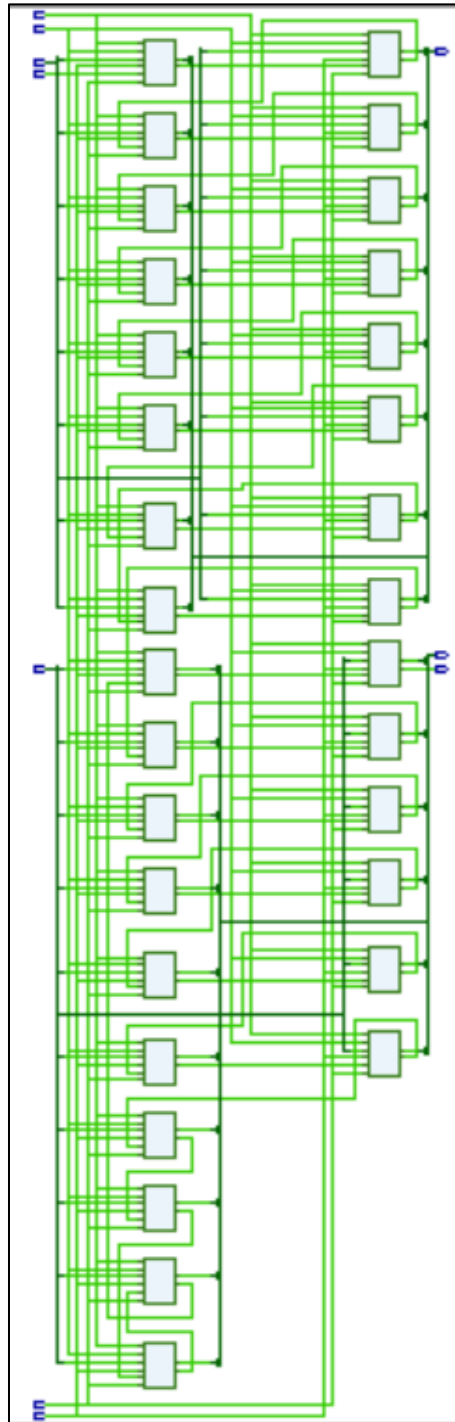
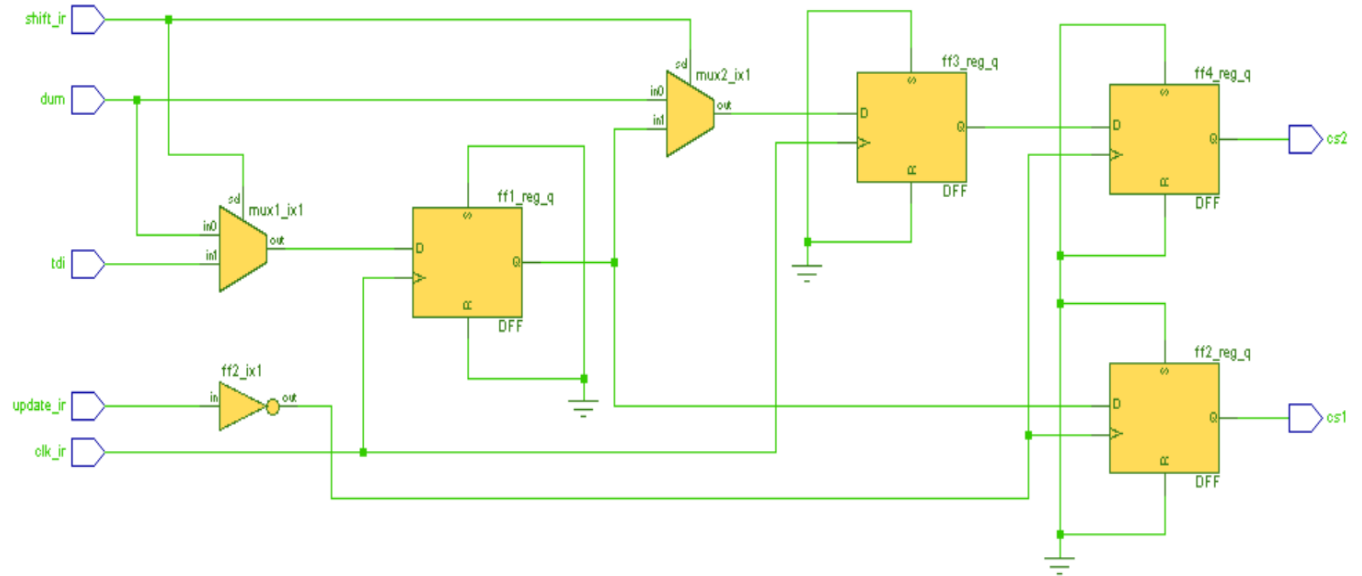## 5.3.RTL synthesis of Bypass Register
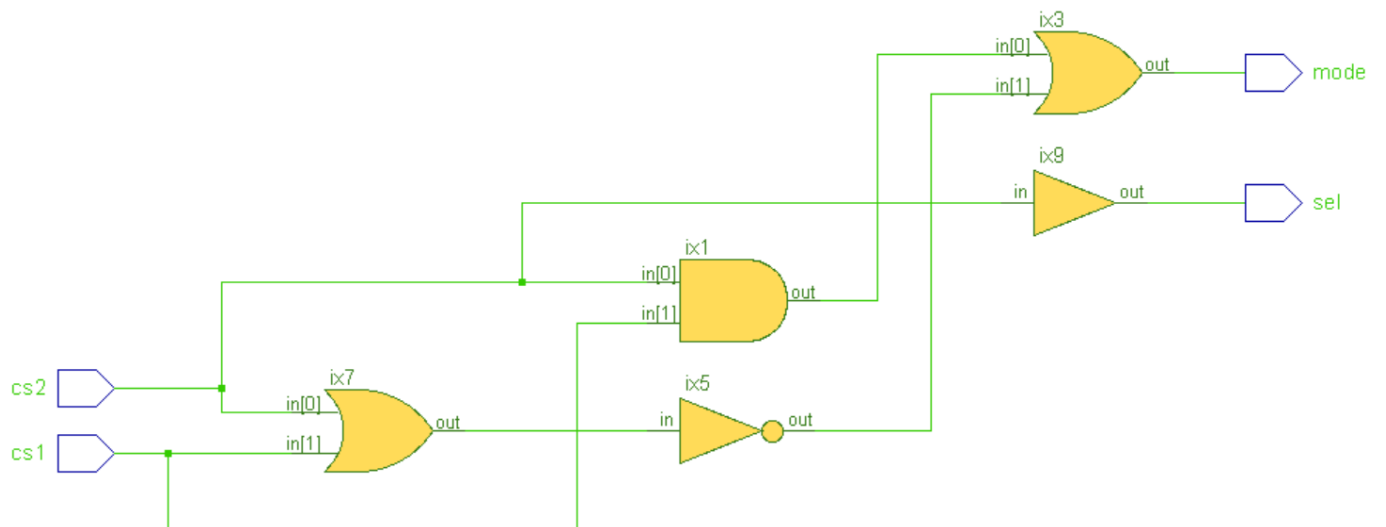

5.4.RTL synthesis of Boundary Scan Cell
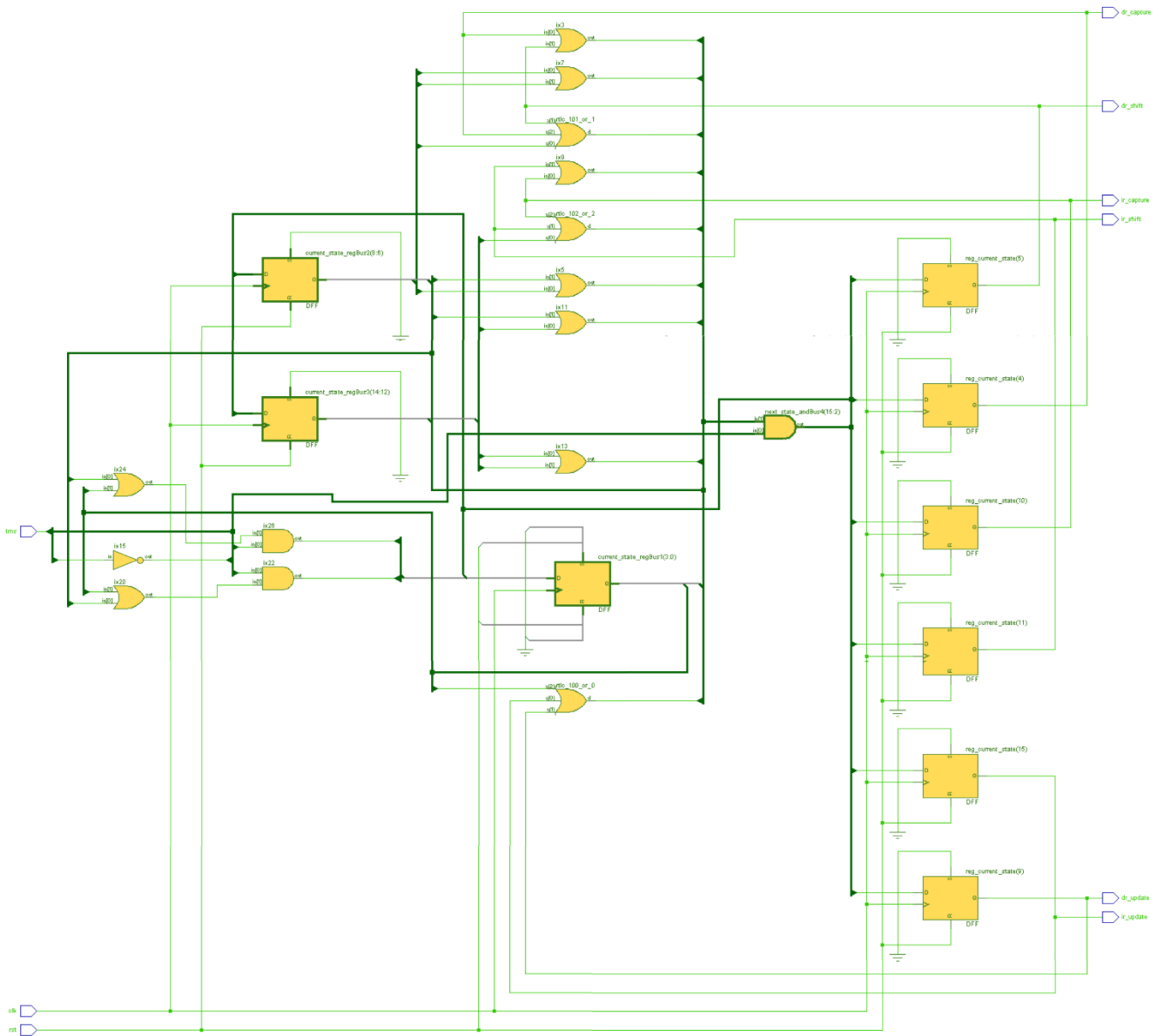
5.5.RTL synthesis of Boundary Scan Register

5.6.RTL synthesis of Instruction Register
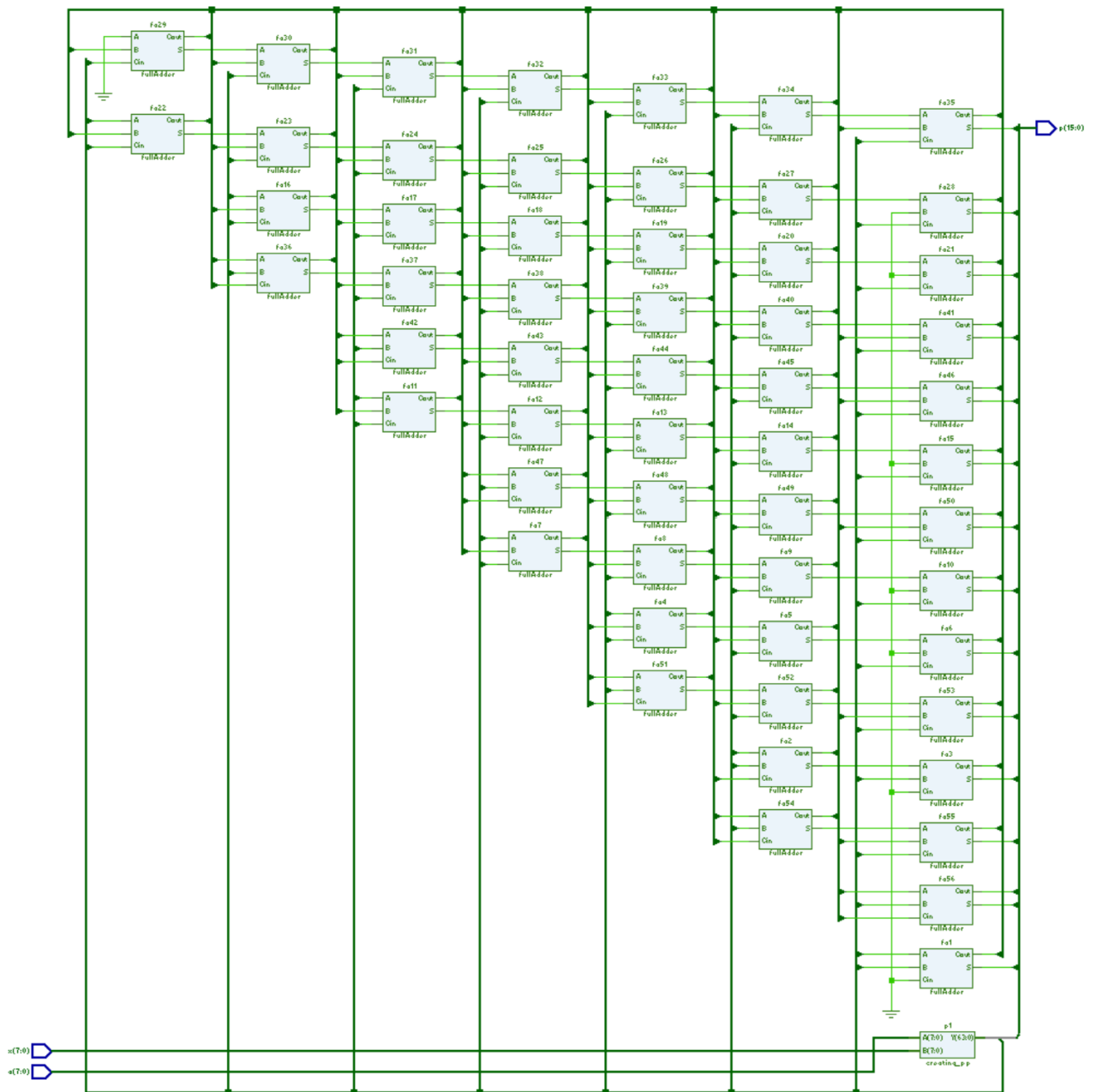

5.7.RTL synthesis of Instruction Decoder

5.8.RTL synthesis of TAP Controller

5.9.RTL synthesis of Multiplier

Area Report

```
Resource                            Used    Avail   Utilization
-------------------------------------------------------------
IOs                                 32      140     22.86%
Global Buffers                      0       16       0.00%
LUTs                                158     2816     5.61%
CLB Slices                          79      1408     5.61%
Dffs or Latches                     0       3236     0.00%
Block RAMs                          0       12       0.00%
Block Multipliers                   0       12       0.00%
Block Multiplier Dffs               0       432      0.00%
GT_CUSTOM                           0       4        0.00%
-------------------------------------------------------------


*********************************************************

Library: work    Cell: multi_8bit    View: Behavioral

*********************************************************

  Cell          Library  References     Total Area

  IBUF          xcv2p    16 x
  LUT2          xcv2p     1 x       1        1 LUTs
  LUT3          xcv2p    14 x       1       14 LUTs
  LUT4          xcv2p   106 x       1      106 LUTs
  OBUF          xcv2p    16 x
  creating_pp   work      1 x      64       64 gates
                                   37       37 LUTs

  Number of ports :                         32
  Number of nets :                         206
  Number of instances :                    154
  Number of references to this view :        0

Total accumulated area :
  Number of LUTs :                         158
  Number of gates :                        185
  Number of accumulated instances :        190
```
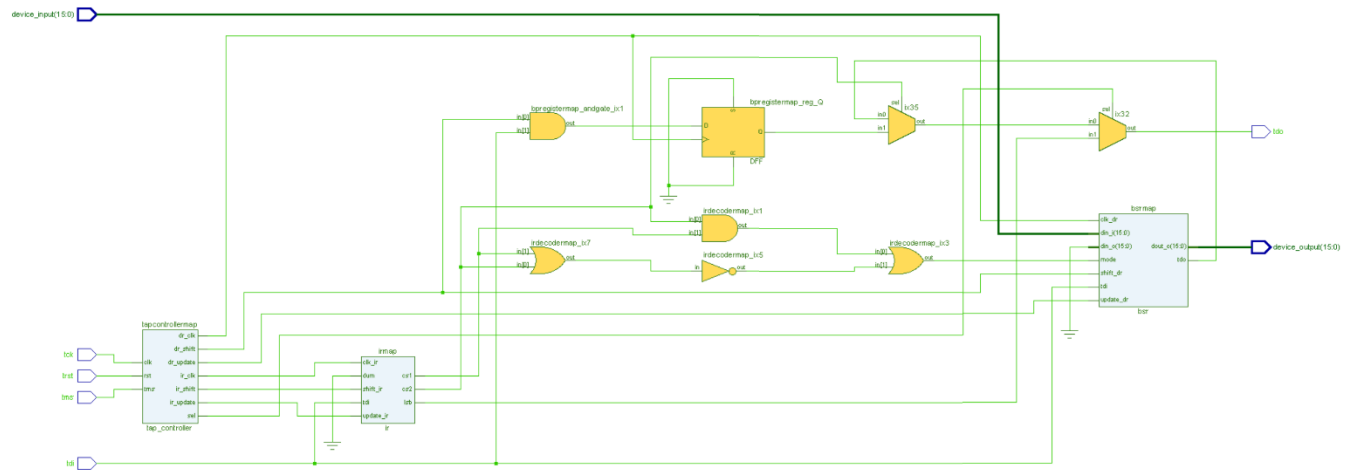
5.10. RTL synthesis of JTAG

Area Report

```
****************************************************************
Device Utilization for 2VP2fg256
****************************************************************
Resource                       Used    Avail    Utilization
----------------------------------------------------------------
IOs                            37      140      26.43%
Global Buffers                 1       16       6.25%
LUTs                           55      2816     1.95%
CLB Slices                     35      1408     2.49%
Dffs or Latches                69      3236     2.13%
Block RAMs                     0       12       0.00%
Block Multipliers              0       12       0.00%
Block Multiplier Dffs          0       432      0.00%
GT_CUSTOM                      0       4        0.00%
----------------------------------------------------------------


**************************************************

Library: work    Cell: top    View: behavioral

**************************************************

  Cell            Library  References    Total Area

  BUFGP           xcv2p     1 x
  FDR             xcv2p     3 x      1       3 Dffs or Latches
  FD_1            xcv2p     2 x      1       2 Dffs or Latches
  IBUF            xcv2p     19 x
  LUT1            xcv2p     2 x      1       2 LUTs
  LUT3            xcv2p     1 x      1       1 LUTs
  LUT4            xcv2p     1 x      1       1 LUTs
  OBUF            xcv2p     17 x
  bsr             work      1 x      49      49 gates
                                     48      48 LUTs
                                     48      48 Dffs or Latches
  tap_controller  work      1 x      21      21 gates
                                     21      21 LUTs
                                     16      16 Dffs or Latches

  Number of ports :                  37
  Number of nets :                   90
  Number of instances :              48
  Number of references to this view :  0

Total accumulated area :
  Number of Dffs or Latches :        69
  Number of LUTs :                   55
  Number of gates :                  76
  Number of accumulated instances :  179
```

Timing Report/Delay

```
NAME                                   GATE     DELAY    ARRIVAL DIR  FANOUT
tapcontrollermap/reg_current_state(5)/C FDC              0.000   up
tapcontrollermap/reg_current_state(5)/Q FDC     0.370    0.370   up
tapcontrollermap/dr_shift              (net)    0.640                  35
tapcontrollermap/ix22095z1329/I1       LUT2              1.010   up
tapcontrollermap/ix22095z1329/O        LUT2     0.264    1.274   up
tapcontrollermap/dr_clk                (net)    0.640                  34
tapcontrollermap/ix22095z1322/I1       LUT2              1.914   up
tapcontrollermap/ix22095z1322/O        LUT2     0.264    2.178   up
tapcontrollermap/next_state(6)         (net)    0.280                  1
tapcontrollermap/reg_current_state(6)/D FDC              2.458   up

              Initial edge separation:    10.000
              Source clock delay:      -   1.359
              Dest clock delay:        +   1.359
                                       -----------
              Edge separation:            10.000
              Setup constraint:        -   0.174
                                       -----------
              Data required time:         9.826
              Data arrival time:       -   2.458   ( 36.53% cell delay, 63.47% net delay )
                                       -----------
              Slack:                      7.368

End CTE Analysis ..... CPU Time Used: 0 sec.
```
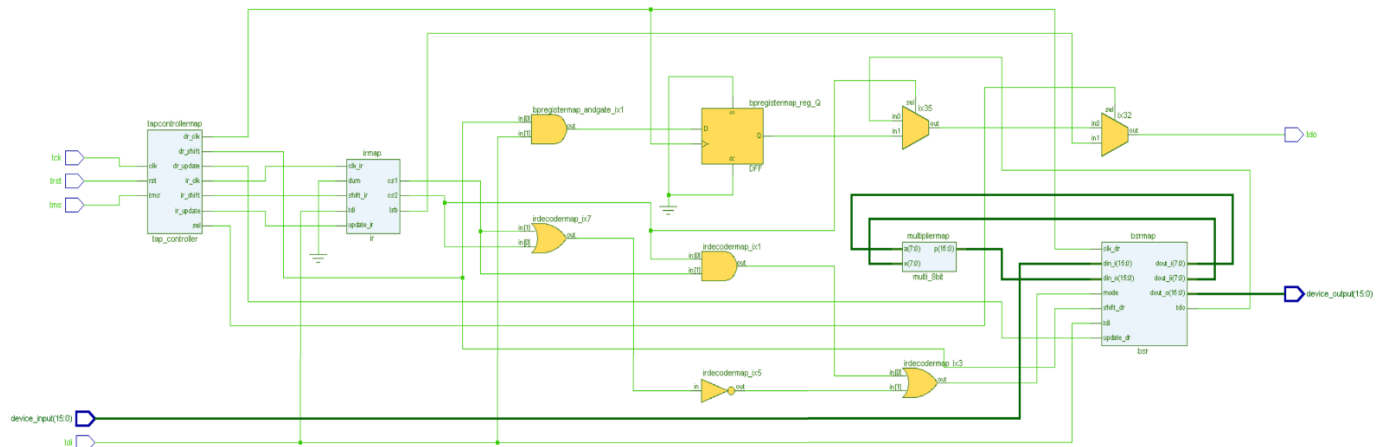
5.11. RTL synthesis of JTAG and Multiplier

Area Report

```
Device Utilization for 2VP2fg256
***********************************************************
Resource                     Used    Avail   Utilization
-----------------------------------------------------------
IOs                          37      140      26.43%
Global Buffers               1       16        6.25%
LUTs                         258     2816      9.16%
CLB Slices                   129     1408      9.16%
Dffs or Latches              85      3236      2.63%
Block RAMs                   0       12        0.00%
Block Multipliers            0       12        0.00%
Block Multiplier Dffs        0       432       0.00%
GT_CUSTOM                    0       4         0.00%
-----------------------------------------------------------


*************************************************

Library: work    Cell: top    View: behavioral

*************************************************

  Cell          Library  References      Total Area

 BUFGP          xcv2p     1 x
 FDR            xcv2p     3 x       1        3 Dffs or Latches
 FD_1           xcv2p     2 x       1        2 Dffs or Latches
 IBUF           xcv2p    19 x
 LUT1           xcv2p     2 x       1        2 LUTs
 LUT2           xcv2p     1 x       1        1 LUTs
 LUT3           xcv2p     1 x       1        1 LUTs
 LUT4           xcv2p     1 x       1        1 LUTs
 OBUF           xcv2p    17 x
 bsr            work      1 x      65       65 gates
                                   56       56 LUTs
                                   64       64 Dffs or Latches
 multi_8bit     work      1 x     178      178 LUTs
                                  185      185 gates
 tap_controller work      1 x      21       21 gates
                                   21       21 LUTs
                                   16       16 Dffs or Latches

 Number of ports :                     37
 Number of nets :                     130
 Number of instances :                 50
 Number of references to this view :    0

Total accumulated area :
 Number of Dffs or Latches :           85
 Number of LUTs :                     258
 Number of gates :                    277
 Number of accumulated instances :    382
```

Timing Report/Delay

```
NAME                                GATE    DELAY    ARRIVAL DIR  FANOUT
tapcontrollermap/reg_current_state(5)/C FDC              0.000    up
tapcontrollermap/reg_current_state(5)/Q FDC     0.370    0.370    up
tapcontrollermap/dr_shift           (net)   0.640                        35
tapcontrollermap/ix22095z1329/I1    LUT2             1.010    up
tapcontrollermap/ix22095z1329/O     LUT2    0.264    1.274    up
tapcontrollermap/dr_clk             (net)   0.640                        34
tapcontrollermap/ix22095z1322/I1    LUT2             1.914    up
tapcontrollermap/ix22095z1322/O     LUT2    0.264    2.178    up
tapcontrollermap/next_state(6)      (net)   0.280                         1
tapcontrollermap/reg_current_state(6)/D FDC              2.458    up

                Initial edge separation:    10.000
                Source clock delay:      -   1.359
                Dest clock delay:        +   1.359
                                           ----------
                Edge separation:            10.000
                Setup constraint:        -   0.174
                                           ----------
                Data required time:          9.826
                Data arrival time:       -   2.458   ( 36.53% cell delay, 63.47% net delay )
                                           ----------
                Slack:                       7.368

End CTE Analysis ..... CPU Time Used: 0 sec.
```

6. Overhead and benefits of the added JTAG

The memory takes up a large portion of most design. Circuitry that is added to test the memory creates area overhead. After each DFT pass, area grows along with the test logic and is extremely challenging to control. To facilitate design testing on ATE (Automatic Test Equipment), DFT logic circuitry is introduced throughout the design process (post-manufacturing). The addition of these logic circuits doesn't have an impact on the operation of any design. The DFT logic circuit causes some design space to be added after each iteration of the DFT. The region for a chip is over headed by this DFT circuitry. A DFT engineer must be highly adept at minimizing and controlling the area by maintaining its track. The overhead problem can be reduced by various techniques, such as test microprogram, scan protection scheme, etc., Figure 2 shows the comparison result of area report of multiplier with and without JTAG.

The following are the benefits of JTAG,

- JTAG offers the ability to evaluate PC-board interconnects without the need of physical test probes or test equipment.
- Doesn't need the board to be in a bootable condition to perform fault diagnostics.
- Device level diagnostics and automated test development for DSP initialization, memory, and flash are made possible by JTAG.

Figure 2 :Accumulated Area of multiplier without JTAG and with JTAG
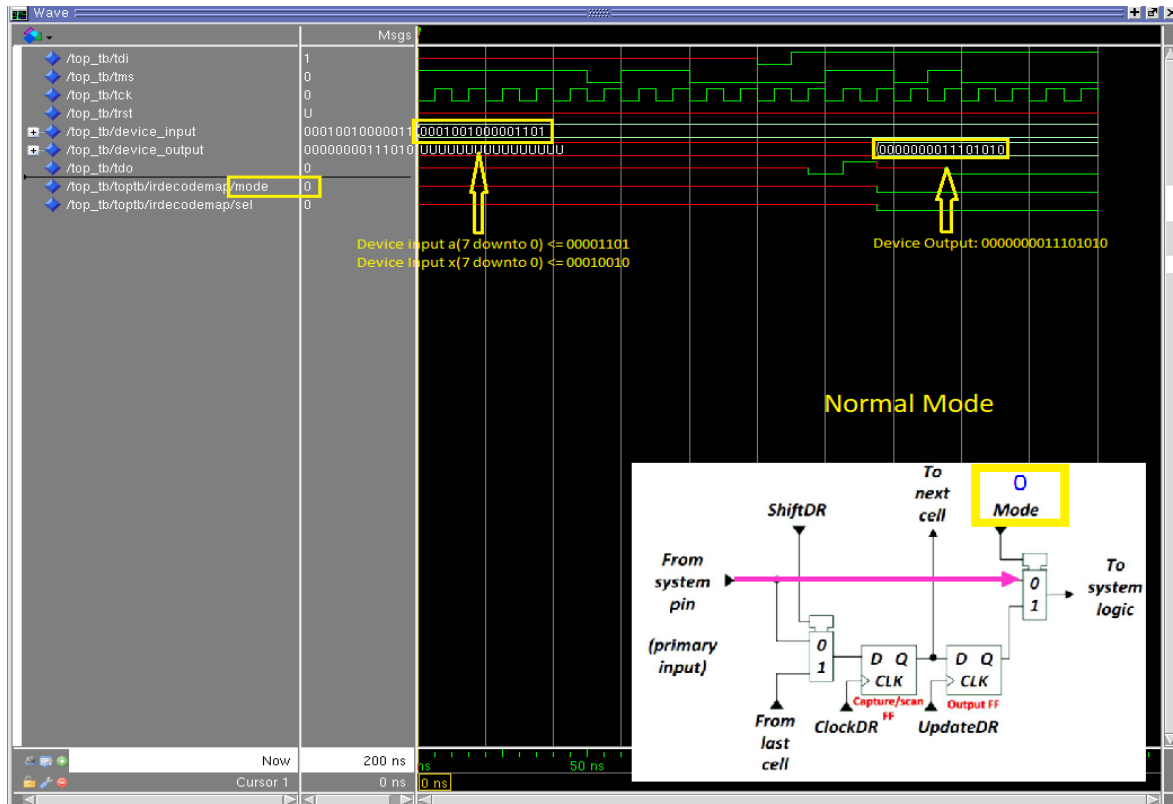
7. Testing 8-bit multiplier using the JTAG
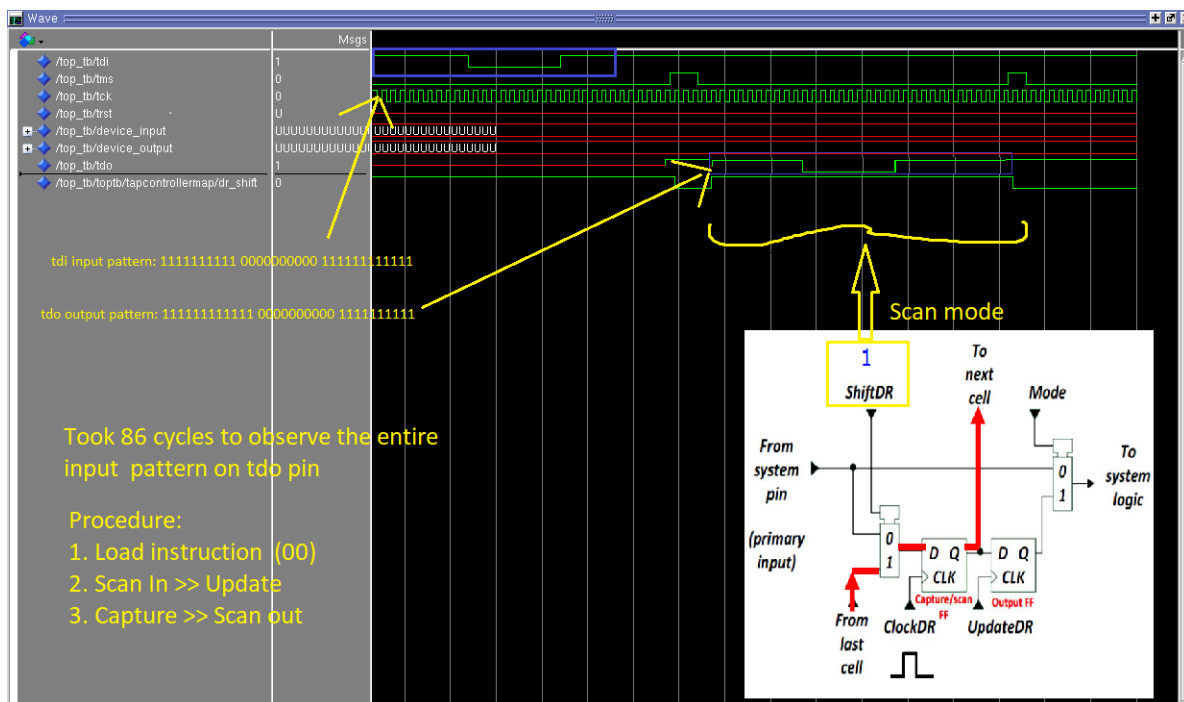# Simulated output of giver 8-bit multiplier design:
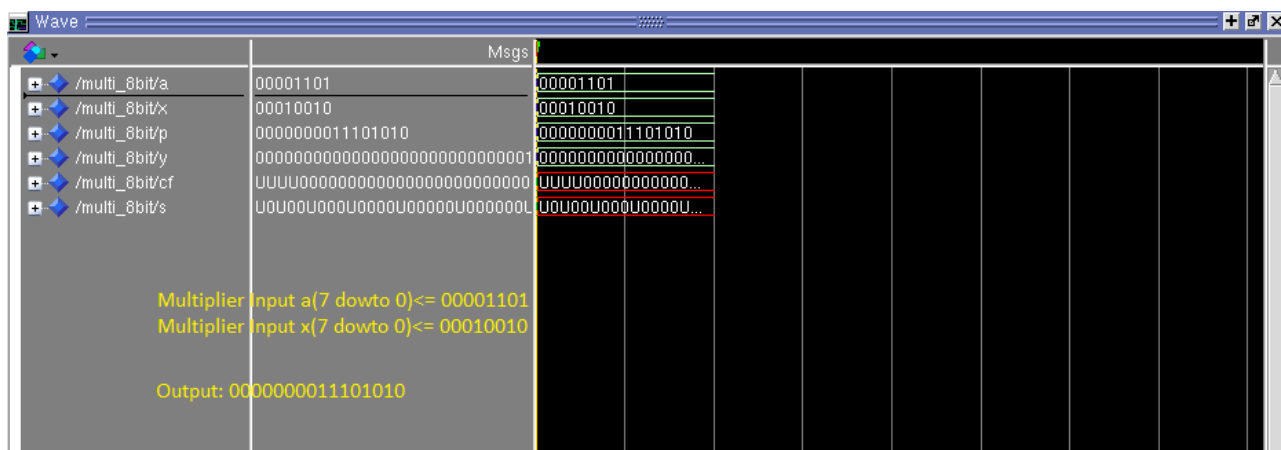
7.1.Output of multiplier with JTAG
JTAG operation - Normal:

JTAG operation – Scan-in, update, capture and scan-out through 32 cells:

8. Conclusion:

Scan-based DFT improves the system's testability by making its internal nodes easier to observe and control. A direct link to the debug logic is made possible by the JTAG path. The advantages of using JTAG methodology extend from chip design through all stages of system development and the device product lifecycle since the JTAG scan channel offers direct core access. Reusing test patterns at different levels of hierarchy, from chip to board to board to system level, is possible with the JTAG approach. As an illustration, the board-level test can employ all or a portion of the chip level test vectors as its base. JTAG technique is also cost-effective because it expedites the testing procedure, ensures greater manufacturing yields, and takes numerous other money-saving steps.

9. Reference:

1. http://cc.ee.ntu.edu.tw/~cmli/VLSItesting/
2. Kenneth P. Parker (auth.) - The Boundary-Scan Handbook-Springer International Publishing (2016)
3. Peter Y.K. Cheung - Boundary-Scan Interconnect Diagnosis (Frontiers in Electronic Testing) (2001)
4. Harry Bleeker, Peter van den Eijnden, Frans de Jong (auth.) - Boundary-Scan Test_ A Practical Approach-Springer US (1993)
5. Jtag (Ieee Std-1149.1)-Institute of Electrical & Electronics Enginee (2001)