**VLSI DESIGN TEST/POWER**

**PROJECT 2**

**GCD BOUNDARY SCAN INSERTION AND ATPG GENERATION**

**Team members:**

|  |  |
| --- | --- |
| **Alamalakala Hareesh** | **M08974088** |
| **Vinay Kumar Burugu** | **M08813508** |
| **Sai Chaitanya Nandipati** | **M08905912** |

**Task 1:**

**Objective of the task:**

The objective is to insert boundary scan to the synthesized and scan inserted GCD machine and also to insert a tap controller with the instruction EXTEST, SAMPLE, PRELOAD, and BYPASS.

**Solution of the Problem:**

The input and output pads (GTECH\_INBUF and GTECG\_OUTBUF) are added to the GCD core and TOP.v and passed to the design compiler.

Additional commands and BSD constraints are followed and the boundary scan insertion is performed and the TAP controller is designed from the Design compiler components and inserted into the circuit.

**Help from the tool to solve(Script):**

The design compiler has the intelligence to insert the boundary scan if the boundary scan commands are given properly and also it inserts the TAP controller with the specified instructions.

**Important Observations:**

Report from the Design compiler:

The following is the report for the boundary scan and TAP controller insertion from the Design compiler

Loading db file '/home/alamalhh/Semester2/testnpower/db/saed90nm\_typ\_ht(1).db'

Loading db file '/opt/CAD/Synopsys/Current/syn/libraries/syn/gtech.db'

Loading db file '/opt/CAD/Synopsys/Current/syn/libraries/syn/standard.sldb'

Loading link library 'saed90nm\_typ\_ht'

Loading link library 'gtech'

Loading verilog file '/home/alamalhh/Semester2/project2\_final\_2/src/gcd\_bsd\_scan.v'

Detecting input file type automatically (-rtl or -netlist).

Running DC verilog reader

Performing 'read' command.

Compiling source file /home/alamalhh/Semester2/project2\_final\_2/src/gcd\_bsd\_scan.v

Reading with netlist reader (equivalent to -netlist option).

Verilog netlist reader completed successfully.

Current design is now '/home/alamalhh/Semester2/project2\_final\_2/src/fsm\_test\_1.db:fsm\_test\_1'

Loaded 10 designs.

Current design is 'fsm\_test\_1'.

Loading verilog file '/home/alamalhh/Semester2/project2\_final\_2/src/TOP.v'

Detecting input file type automatically (-rtl or -netlist).

Running DC verilog reader

Performing 'read' command.

Compiling source file /home/alamalhh/Semester2/project2\_final\_2/src/TOP.v

Reading with netlist reader (equivalent to -netlist option).

Verilog netlist reader completed successfully.

Current design is now '/home/alamalhh/Semester2/project2\_final\_2/src/TOP.db:TOP'

Loaded 1 design.

Current design is 'TOP'.

Linking design 'TOP'

Using the following designs and libraries:

--------------------------------------------------------------------------

saed90nm\_typ\_ht (library) /home/alamalhh/Semester2/testnpower/db/saed90nm\_typ\_ht(1).db

Information: Writing 'my\_package' package description. (UIT-254)

Accepted pinmap file.

Accepted dft signal specification for modes: all\_dft

Accepted dft signal specification for modes: all\_dft

Accepted dft signal specification for modes: all\_dft

Accepted dft signal specification for modes: all\_dft

Accepted dft signal specification for modes: all\_dft

Accepted bsd configuration specification.

Accepted bsd instruction specification.

Accepted bsd instruction specification.

Accepted bsd instruction specification.

Accepted bsd instruction specification.

Accepted bsd instruction specification.

Accepted bsd compliance specification.

Accepted dft configuration specification.

Loading db file '/opt/CAD/Synopsys/Current/syn/libraries/syn/dw\_foundation.sldb'

Warning: The following synthetic libraries should be added to

the list of link libraries:

'dw\_foundation.sldb'. (UISN-26)

Information: Changed wire load model for 'regis\_test\_2' from '(none)' to '8000'. (OPT-170)

Information: Changed wire load model for 'subtractor' from '(none)' to '8000'. (OPT-170)

Information: Changed wire load model for 'comparator' from '(none)' to 'ForQA'. (OPT-170)

Information: Changed wire load model for 'regis\_test\_1' from '(none)' to '8000'. (OPT-170)

Information: Changed wire load model for 'regis\_test\_0' from '(none)' to '8000'. (OPT-170)

Information: Changed wire load model for 'mux\_1' from '(none)' to 'ForQA'. (OPT-170)

Information: Changed wire load model for 'mux\_0' from '(none)' to 'ForQA'. (OPT-170)

Information: Changed wire load model for 'fsm\_test\_1' from 'ForQA' to '8000'. (OPT-170)

Information: Changed wire load model for 'gcd\_test\_1' from '(none)' to '8000'. (OPT-170)

Information: Changed wire load model for 'gcd\_bsd' from '(none)' to '8000'. (OPT-170)

Info: Initializing compliance enable ports `TEST\_SE RST' with pattern 01.

Warning: Inferring default device package 'my\_package'. (TEST-601)

Warning: Enable pin(s) of pad driving port "D\_O[0]" always enable the driver; the port acts as a simple two-state output port and no BSR cell is added to the control pin(s). (TEST-433)

Warning: Enable pin(s) of pad driving port "D\_O[1]" always enable the driver; the port acts as a simple two-state output port and no BSR cell is added to the control pin(s). (TEST-433)

Warning: Enable pin(s) of pad driving port "D\_O[2]" always enable the driver; the port acts as a simple two-state output port and no BSR cell is added to the control pin(s). (TEST-433)

Warning: Enable pin(s) of pad driving port "D\_O[3]" always enable the driver; the port acts as a simple two-state output port and no BSR cell is added to the control pin(s). (TEST-433)

Warning: Enable pin(s) of pad driving port "TEST\_SO" always enable the driver; the port acts as a simple two-state output port and no BSR cell is added to the control pin(s). (TEST-433)

Generating the TAP

Structuring 'TOP\_DW\_tap\_uc\_width4\_id0\_idcode\_opcode1\_version0\_part0\_man\_num0\_sync\_mode1'

Mapping 'TOP\_DW\_tap\_uc\_width4\_id0\_idcode\_opcode1\_version0\_part0\_man\_num0\_sync\_mode1'

TOTAL

ELAPSED WORST NEG SETUP DESIGN

TIME AREA SLACK COST RULE COST ENDPOINT

--------- --------- --------- --------- --------- -------------------------

0:00:01 4296.9 0.00 0.0 0.0

Generating the BSR cells

Flattening 'TOP\_DW\_bc\_4' (Low effort) (Single Output Minimization)

Structuring 'TOP\_DW\_bc\_4'

Mapping 'TOP\_DW\_bc\_4'

0:00:01 1773.1 0.00 0.0 0.0

Flattening 'TOP\_DW\_bc\_2' (Low effort) (Single Output Minimization)

Structuring 'TOP\_DW\_bc\_2'

Mapping 'TOP\_DW\_bc\_2'

Flattening 'TOP\_DW\_bc\_1' (Low effort) (Single Output Minimization)

Structuring 'TOP\_DW\_bc\_1'

Mapping 'TOP\_DW\_bc\_1'

Mapping 12 unused opcode(s) to the BYPASS instruction

Generating the control logic

Writing implemented instructions information to the design

Creating Hierarchy for Boundary Scan Logic ...

Information: Changed wire load model for 'TOP\_Decoder\_inst\_design' from '(none)' to 'ForQA'. (OPT-170)

Information: Changed wire load model for 'TOP\_BSR\_mode\_inst\_design' from '(none)' to 'ForQA'. (OPT-170)

Information: Changed wire load model for 'TOP\_BSR\_top\_inst\_design' from '(none)' to 'ForQA'. (OPT-170)

Information: Changed wire load model for 'TOP\_DW\_tap\_uc\_width4\_id0\_idcode\_opcode1\_version0\_part0\_man\_num0\_sync\_mode1' from 'ForQA' to '8000'. (OPT-170)

Structuring 'TOP\_Decoder\_inst\_design'

Mapping 'TOP\_Decoder\_inst\_design'

0:00:01 6276.1 0.00 0.0 0.0

0:00:01 6225.4 0.00 0.0 0.0

Structuring 'TOP\_BSR\_mode\_inst\_design'

Mapping 'TOP\_BSR\_mode\_inst\_design'

Information: Changed wire load model for 'TOP\_BSR\_mode\_inst\_design' from 'ForQA' to '8000'. (OPT-170)

Information: Changed wire load model for 'TOP\_BSR\_top\_inst\_design' from 'ForQA' to '8000'. (OPT-170)

Beginning Mapping Optimizations

-------------------------------

Warning: The following synthetic libraries should be added to

the list of link libraries:

'dw\_foundation.sldb'. (UISN-26)

...Starting IEEE 1149.1 Compliance Checking.

...Initializing compliance enable ports `RST TEST\_SE ' with pattern 10.

...Finding set of sequential elements.

...Analyzing TAP and TAP Controller.

......Analyzing TAP.

......Finding the set of TAP controller sequential elements.

......Pruning set of TAP Controller sequential Elements

...Checking the TAP controller initialization.

...Analyzing the TAP controller reset condition.

...Traversing the TAP controller states.

...Inferring TAP controller clock outputs.

...Analyzing TRST port.

Warning: Undriven input port TRST is floating. When undriven, this port should behave as though it was driven by logic one. (TEST-819)

...Analyzing TMS Port.

Warning: Undriven input port TMS is floating. When undriven, this port should behave as though it was driven by logic one. (TEST-819)

...Analyzing TCK halt state.

Warning: Undriven input port TDI is floating. When undriven, this port should behave as though it was driven by logic one. (TEST-819)

...Analyzing the instruction register.

......Finding the update flops.

...Analyzing the BYPASS register.

...Analyzing the DIR register.

Device Identification Register doesn't exists

...Analyzing the boundary scan register.

......Finding the update flops.

......Finding the BSR cells controlling the design ports.

......Finding the BSR cells sensing the design ports.

......Finding the BSR cells driving the design ports.

...Using the implemented instructions information.

...Inferring the different test data register (TDR)s selected by instructions.

...Finding the BSR controlling cells PIs.

...Finding the BSR controlling cells POs.

...Finding the BSR non-controlling cells PIs.

...Finding the BSR cells POs.

...Modifying the BSR controlling/Output cells PIs.

...Analyzing the BYPASS instruction.

...Analyzing the EXTEST instruction.

...Analyzing the SAMPLE instruction.

...Analyzing the PRELOAD instruction.

...Analyzing the INTEST instruction.

...Analyzing the CLAMP instruction.

...Analyzing the HIGHZ instruction.

...Analyzing the RUNBIST instruction.

...Analyzing the IDCODE and USERCODE instructions.

...Analyzing Test-Logic-Reset Tap Controller State.

...Finished IEEE 1149.1 Compliance Checking.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IEEE 1149.1 Summary

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Test Logic Reset Method: Synchronous and Asynchronous(TRST)

16 state elements found in the TAP controller

TOP\_DW\_tap\_inst/U1/current\_state\_reg[0]

TOP\_DW\_tap\_inst/U1/current\_state\_reg[10]

TOP\_DW\_tap\_inst/U1/current\_state\_reg[11]

TOP\_DW\_tap\_inst/U1/current\_state\_reg[12]

TOP\_DW\_tap\_inst/U1/current\_state\_reg[13]

TOP\_DW\_tap\_inst/U1/current\_state\_reg[14]

TOP\_DW\_tap\_inst/U1/current\_state\_reg[15]

TOP\_DW\_tap\_inst/U1/current\_state\_reg[1]

TOP\_DW\_tap\_inst/U1/current\_state\_reg[2]

TOP\_DW\_tap\_inst/U1/current\_state\_reg[3]

TOP\_DW\_tap\_inst/U1/current\_state\_reg[4]

TOP\_DW\_tap\_inst/U1/current\_state\_reg[5]

TOP\_DW\_tap\_inst/U1/current\_state\_reg[6]

TOP\_DW\_tap\_inst/U1/current\_state\_reg[7]

TOP\_DW\_tap\_inst/U1/current\_state\_reg[8]

TOP\_DW\_tap\_inst/U1/current\_state\_reg[9]

--------------------------

TAP CONTROLLER DESCRIPTION

--------------------------

State Element Count: 16

State Encoding:

Test-Logic-Reset: 0000000000000001

Run-Test/Idle: 0000000010000000

Select-DR-Scan: 0000000100000000

Capture-DR: 0000001000000000

Shift-DR: 0000010000000000

Exit1-DR: 0000100000000000

Pause-DR: 0001000000000000

Exit2-DR: 0010000000000000

Update-DR: 0100000000000000

Select-IR-Scan: 1000000000000000

Capture-IR: 0000000000000010

Shift-IR: 0000000000000100

Exit1-IR: 0000000000001000

Pause-IR: 0000000000010000

Exit2-IR: 0000000000100000

Update-IR: 0000000001000000

4 cells found in the Instruction Register

TOP\_DW\_tap\_inst/U5/U1\_0/U3

TOP\_DW\_tap\_inst/U5/U1\_1/U3

TOP\_DW\_tap\_inst/U5/U1\_2/U3

TOP\_DW\_tap\_inst/U5/U1\_3/U3

4 update flops in Instruction Register

TOP\_DW\_tap\_inst/U5/U3\_0

TOP\_DW\_tap\_inst/U5/U3\_1

TOP\_DW\_tap\_inst/U5/U3\_2

TOP\_DW\_tap\_inst/U5/U3\_3

5 standard instructions found.

0 user defined instructions found.

2 BYPASS register instructions found

1 opcodes in instruction

Opcodes are: 1111

Instruction Types: BYPASS

1 opcodes in instruction

Opcodes are: 0010

Instruction Types: CLAMP

3 boundary scan register instructions found

1 opcodes in instruction

Opcodes are: 0001

Instruction Types: EXTEST

1 opcodes in instruction

Opcodes are: 0100

Instruction Types: SAMPLE

1 opcodes in instruction

Opcodes are: 0100

Instruction Types: PRELOAD

1 cells in BYPASS register

TOP\_DW\_tap\_inst/U6/U4

16 cells in boundary scan register

TOP\_BSR\_top\_inst/TOP\_TEST\_SO\_bsr16/U1/U1/U3: Shift Flop

TOP\_BSR\_top\_inst/TOP\_TEST\_SO\_bsr16/U1/U3: Update Flop

0: Position from TDO

OUTPUT2: Primary Cell Type

TEST\_SO: Driven Port

core/M\_GCD/Y\_REG/output\_reg\_3\_: Primary Input

TOP\_BSR\_top\_inst/TOP\_TEST\_SO\_bsr16/\*cell\*660: Primary Output

BC\_1: BSDL Cell Type

TOP\_BSR\_top\_inst/TOP\_TEST\_SI\_bsr15/U1/U1/U3: Shift Flop

TOP\_BSR\_top\_inst/TOP\_TEST\_SI\_bsr15/U1/U3: Update Flop

1: Position from TDO

INPUT: Primary Cell Type

TEST\_SI: Sensed Port

TEST\_SI: Primary Input

TOP\_BSR\_top\_inst/TOP\_TEST\_SI\_bsr15/\*cell\*637: Primary Output

BC\_2: BSDL Cell Type

TOP\_BSR\_top\_inst/TOP\_D\_O[3]\_bsr14/U1/U1/U3: Shift Flop

TOP\_BSR\_top\_inst/TOP\_D\_O[3]\_bsr14/U1/U3: Update Flop

2: Position from TDO

OUTPUT2: Primary Cell Type

D\_O[3]: Driven Port

core/M\_GCD/OUT\_REG/output\_reg\_3\_: Primary Input

TOP\_BSR\_top\_inst/TOP\_D\_O[3]\_bsr14/\*cell\*660: Primary Output

BC\_1: BSDL Cell Type

TOP\_BSR\_top\_inst/TOP\_D\_O[2]\_bsr13/U1/U1/U3: Shift Flop

TOP\_BSR\_top\_inst/TOP\_D\_O[2]\_bsr13/U1/U3: Update Flop

3: Position from TDO

OUTPUT2: Primary Cell Type

D\_O[2]: Driven Port

core/M\_GCD/OUT\_REG/output\_reg\_2\_: Primary Input

TOP\_BSR\_top\_inst/TOP\_D\_O[2]\_bsr13/\*cell\*660: Primary Output

BC\_1: BSDL Cell Type

TOP\_BSR\_top\_inst/TOP\_D\_O[1]\_bsr12/U1/U1/U3: Shift Flop

TOP\_BSR\_top\_inst/TOP\_D\_O[1]\_bsr12/U1/U3: Update Flop

4: Position from TDO

OUTPUT2: Primary Cell Type

D\_O[1]: Driven Port

core/M\_GCD/OUT\_REG/output\_reg\_1\_: Primary Input

TOP\_BSR\_top\_inst/TOP\_D\_O[1]\_bsr12/\*cell\*660: Primary Output

BC\_1: BSDL Cell Type

TOP\_BSR\_top\_inst/TOP\_D\_O[0]\_bsr11/U1/U1/U3: Shift Flop

TOP\_BSR\_top\_inst/TOP\_D\_O[0]\_bsr11/U1/U3: Update Flop

5: Position from TDO

OUTPUT2: Primary Cell Type

D\_O[0]: Driven Port

core/M\_GCD/OUT\_REG/output\_reg\_0\_: Primary Input

TOP\_BSR\_top\_inst/TOP\_D\_O[0]\_bsr11/\*cell\*660: Primary Output

BC\_1: BSDL Cell Type

TOP\_BSR\_top\_inst/TOP\_Y\_I[3]\_bsr10/U1/U1/U3: Shift Flop

TOP\_BSR\_top\_inst/TOP\_Y\_I[3]\_bsr10/U1/U3: Update Flop

6: Position from TDO

INPUT: Primary Cell Type

Y\_I[3]: Sensed Port

Y\_I[3]: Primary Input

TOP\_BSR\_top\_inst/TOP\_Y\_I[3]\_bsr10/\*cell\*637: Primary Output

BC\_2: BSDL Cell Type

TOP\_BSR\_top\_inst/TOP\_Y\_I[2]\_bsr9/U1/U1/U3: Shift Flop

TOP\_BSR\_top\_inst/TOP\_Y\_I[2]\_bsr9/U1/U3: Update Flop

7: Position from TDO

INPUT: Primary Cell Type

Y\_I[2]: Sensed Port

Y\_I[2]: Primary Input

TOP\_BSR\_top\_inst/TOP\_Y\_I[2]\_bsr9/\*cell\*637: Primary Output

BC\_2: BSDL Cell Type

TOP\_BSR\_top\_inst/TOP\_Y\_I[1]\_bsr8/U1/U1/U3: Shift Flop

TOP\_BSR\_top\_inst/TOP\_Y\_I[1]\_bsr8/U1/U3: Update Flop

8: Position from TDO

INPUT: Primary Cell Type

Y\_I[1]: Sensed Port

Y\_I[1]: Primary Input

TOP\_BSR\_top\_inst/TOP\_Y\_I[1]\_bsr8/\*cell\*637: Primary Output

BC\_2: BSDL Cell Type

TOP\_BSR\_top\_inst/TOP\_Y\_I[0]\_bsr7/U1/U1/U3: Shift Flop

TOP\_BSR\_top\_inst/TOP\_Y\_I[0]\_bsr7/U1/U3: Update Flop

9: Position from TDO

INPUT: Primary Cell Type

Y\_I[0]: Sensed Port

Y\_I[0]: Primary Input

TOP\_BSR\_top\_inst/TOP\_Y\_I[0]\_bsr7/\*cell\*637: Primary Output

BC\_2: BSDL Cell Type

TOP\_BSR\_top\_inst/TOP\_X\_I[3]\_bsr6/U1/U1/U3: Shift Flop

TOP\_BSR\_top\_inst/TOP\_X\_I[3]\_bsr6/U1/U3: Update Flop

10: Position from TDO

INPUT: Primary Cell Type

X\_I[3]: Sensed Port

X\_I[3]: Primary Input

TOP\_BSR\_top\_inst/TOP\_X\_I[3]\_bsr6/\*cell\*637: Primary Output

BC\_2: BSDL Cell Type

TOP\_BSR\_top\_inst/TOP\_X\_I[2]\_bsr5/U1/U1/U3: Shift Flop

TOP\_BSR\_top\_inst/TOP\_X\_I[2]\_bsr5/U1/U3: Update Flop

11: Position from TDO

INPUT: Primary Cell Type

X\_I[2]: Sensed Port

X\_I[2]: Primary Input

TOP\_BSR\_top\_inst/TOP\_X\_I[2]\_bsr5/\*cell\*637: Primary Output

BC\_2: BSDL Cell Type

TOP\_BSR\_top\_inst/TOP\_X\_I[1]\_bsr4/U1/U1/U3: Shift Flop

TOP\_BSR\_top\_inst/TOP\_X\_I[1]\_bsr4/U1/U3: Update Flop

12: Position from TDO

INPUT: Primary Cell Type

X\_I[1]: Sensed Port

X\_I[1]: Primary Input

TOP\_BSR\_top\_inst/TOP\_X\_I[1]\_bsr4/\*cell\*637: Primary Output

BC\_2: BSDL Cell Type

TOP\_BSR\_top\_inst/TOP\_X\_I[0]\_bsr3/U1/U1/U3: Shift Flop

TOP\_BSR\_top\_inst/TOP\_X\_I[0]\_bsr3/U1/U3: Update Flop

13: Position from TDO

INPUT: Primary Cell Type

X\_I[0]: Sensed Port

X\_I[0]: Primary Input

TOP\_BSR\_top\_inst/TOP\_X\_I[0]\_bsr3/\*cell\*637: Primary Output

BC\_2: BSDL Cell Type

TOP\_BSR\_top\_inst/TOP\_GO\_I\_bsr2/U1/U1/U3: Shift Flop

TOP\_BSR\_top\_inst/TOP\_GO\_I\_bsr2/U1/U3: Update Flop

14: Position from TDO

INPUT: Primary Cell Type

GO\_I: Sensed Port

GO\_I: Primary Input

TOP\_BSR\_top\_inst/TOP\_GO\_I\_bsr2/\*cell\*637: Primary Output

BC\_2: BSDL Cell Type

TOP\_BSR\_top\_inst/TOP\_CLK\_bsr1/U1/U3: Shift Flop

15: Position from TDO

CLOCK: Primary Cell Type

CLK: Sensed Port

CLK: Primary Input

BC\_4: BSDL Cell Type

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IEEE 1149.1 Violation Summary

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2 Violations found in extraction of TAP Controller

Violates Rule: 3.3.1b Corresponds to Errors: TEST-819

Violates Rule: 3.6.1b Corresponds to Errors: TEST-819

check\_bsd successful.

Opened bsdl file /home/alamalhh/Semester2/project2\_final\_2/src/TOP\_bsd.bsdl for writing.

Warning: The following synthetic libraries should be added to

the list of link libraries:

'dw\_foundation.sldb'. (UISN-26)

...Generating the BSDL file for the design 'TOP'.

Warning: Inferring default device package 'my\_package'. (TEST-601)

Warning: No linkage ports were found for the design 'TOP'. (TEST-491)

write\_bsdl successful.

Warning: The following synthetic libraries should be added to

the list of link libraries:

'dw\_foundation.sldb'. (UISN-26)

.....Generating vectors to test the asynchronous test logic reset

.....Generating vectors to test the synchronizing sequence of 5 1's on tms

.....Generating vectors to test the TAP FSM

.....Generating vectors to test boundary scan instructions

.......Generating vectors to test the 'BYPASS' instruction.

.......Generating vectors to test the 'EXTEST' instruction.

.......Generating vectors to test the 'SAMPLE' instruction.

.......Generating vectors to test the 'PRELOAD' instruction.

.......Generating vectors to test the 'CLAMP' instruction.

.....Generating vectors to test the boundary scan register

.....Generating vectors to perform leakage test.

Created bsd patterns.

...Getting test program from design.

...Writing test patterns to file ./src/bsd\_patterns.stil.

...Writing STIL-DPV test bench to file ./src/bsd\_patterns.v.

...Getting test program from design.

...Writing native Verilog test bench to file ./src/BSD\_tb.v.v.

Writing ddc file './src/TOP\_bsd.ddc'.

Warning: The specified replacement character (\_) is conflicting with the specified allowed or restricted character. (UCN-4)

Writing verilog file '/home/alamalhh/Semester2/project2\_final\_2/src/TOP\_bsd.v'.

Writing the protocol file /home/alamalhh/Semester2/project2\_final\_2/system\_io\_scan\_bsd.spf...

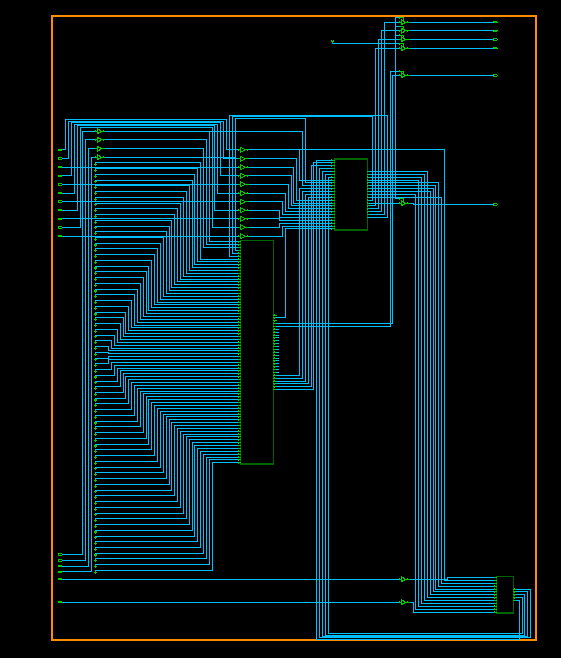
**Observations:**

The scan inserted design is loaded into the Design Compiler tool and the initial steps of setting library and liking design are followed.

The dft signals for the TAP insertion are accepted and the TAP controller is generated from the command insert\_dft.

The instructions required are generated in the TAP controller with the respective opcodes suggested in the boundary scan constraints.

The following is the whole chip schematic from the design compiler with the gcd\_core ,BSR and the TAP controller.



**Script used to generate the boundary scan and thee TAP controller generation:**

set link\_library [list dw\_foundation.sldb dft\_jtag.sldb /home/alamalhh/Semester2/testnpower/db/saed90nm\_typ\_ht(1).db]

set target\_library [list /home/alamalhh/Semester2/testnpower/db/saed90nm\_typ\_ht(1).db ]

set link\_library [concat /home/alamalhh/Semester2/testnpower/db/saed90nm\_typ\_ht(1).db \

$target\_library]

set synthetic\_library [list dw\_foundation.sldb ]

#Read in top-level design

read\_verilog ./src/gcd\_bsd\_scan.v

read\_verilog ./src/TOP.v

set current\_design TOP

link

#dont\_touch CORE and pads

set\_dont\_touch [list GTECH\_INBUF GTECH\_OUTBUF]

read\_pin\_map ./src/pin.txt

#read pin map for BSR cell order

#define Tap signals

set\_dft\_signal -view spec -type TCK -port TCK

set\_dft\_signal -view spec -type TDI -port TDI

set\_dft\_signal -view spec -type TDO -port TDO

set\_dft\_signal -view spec -type TMS -port TMS

set\_dft\_signal -view spec -type TRST -port TRSTN

#define functional clocks

create\_clock CLK -period 100 -waveform {0 50}

#Configure tap

set\_bsd\_configuration -ir\_width 4

#Define standard instructions

#opcodes: extest= 0010, sample & preload=0100, bypass=1111

set\_bsd\_instruction -view spec [list EXTEST] -code [list 0001] -reg BOUNDARY

set\_bsd\_instruction -view spec [list SAMPLE] -code [list 0100] -reg BOUNDARY

set\_bsd\_instruction -view spec [list PRELOAD] -code [list 0100] -reg BOUNDARY

set\_bsd\_instruction -view spec [list BYPASS] -code [list 1111] -reg BYPASS

#Define instructions for CLAMP [opcode=0010]

set\_bsd\_instruction -view spec [list CLAMP] -code [list 0010] -reg BYPASS

#Define compliance enable signals for TEST\_MODE=1; RESETN=1

set\_bsd\_compliance -name P1 -pattern {TEST\_SE 0 RST 1}

#Enable bsd-insertion

set\_dft\_configuration -bsd enable -scan disable

#preview\_dft –bsd all

#insert jtag..includes compile

insert\_dft

#Compliance checking

check\_bsd -verbose

#Generate bsdl file

write\_bsdl -out ./src/TOP\_bsd.bsdl

#Generate bsd patterns

create\_bsd\_patterns -type all

write\_test -format stil\_testbench -output ./src/bsd\_patterns

# generate verilog TAP testbench

write\_test -format verilog -output ./src/BSD\_tb.v

#write out jtag-inserted netlist

write -format ddc -hierarchy -output ./src/TOP\_bsd.ddc

change\_names -rules verilog -hier

write -format verilog -hierarchy -output ./src/TOP\_bsd.v

write\_test\_protocol -instruction STT -o system\_io\_scan\_bsd.spf

**TASK2:**

**Objective of the task( Method 1)**

The objective of the task is to integrate both the internal scan chain and the boundary scan chain into a single scan chain and generate ATPG ptterns using Tetramax

**Solution of the Problem:**

There can be an additional instruction developed by following additional BSD constraints and the tool inserts an appropriate logic for connecting both the scan chains.

**Help from the tool to solve(Script):**

Additional constraints and commands are given to the tool to generate the SCANCH instruction to integrate both the internal and boundary scan chains.

**Constraints and commands given for the insertion of SCANCH instruction:**

# STT register implementation

set\_dft\_signal -type tdi -hookup\_pin I0/test\_si

set\_dft\_signal -type tdo -hookup\_pin I0/test\_so

set\_dft\_signal -type bsd\_shift\_en -hookup\_pin I0/test\_se

set\_dft\_signal -type capture\_clk -hookup\_pin I0/clk

set\_scan\_path STT\_REG -class bsd -exact\_length 4 -hookup {BSR\_SI BSR\_SO core/test\_si core/test\_so core/test\_se core/clk}

set\_bsd\_instruction STT -reg STT\_REG

**Important Observations:**

**The area report for the design without the SCANCH instruction implementation is following:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report : area

Design : TOP

Version: H-2013.03-SP2

Date : Wed Mar 16 18:38:34 2016

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Library(s) Used:

saed90nm\_typ\_ht (File: /home/alamalhh/Semester2/testnpower/db/saed90nm\_typ\_ht(1).db)

gtech (File: /opt/CAD/Synopsys/Current/syn/libraries/syn/gtech.db)

Number of ports: 23

Number of nets: 144

Number of cells: 26

Number of combinational cells: 17

Number of sequential cells: 6

Number of macros/black boxes: 0

Number of buf/inv: 0

Number of references: 5

Combinational area: 3904.819179

Buf/Inv area: 939.110408

Noncombinational area: 2253.311981

Macro/Black Box area: 0.000000

Net Interconnect area: 188.737518

Total cell area: 6158.131160

Total area: 6346.868679

Information: This design contains unmapped logic. (RPT-7)

**Report area after the addition of the SCANCH instruction:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report : area

Design : TOP

Version: H-2013.03-SP2

Date : Wed Mar 16 18:47:17 2016

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Library(s) Used:

saed90nm\_typ\_ht (File: /home/alamalhh/Semester2/testnpower/db/saed90nm\_typ\_ht(1).db)

gtech (File: /opt/CAD/Synopsys/Current/syn/libraries/syn/gtech.db)

Number of ports: 23

Number of nets: 147

Number of cells: 26

Number of combinational cells: 17

Number of sequential cells: 6

Number of macros/black boxes: 0

Number of buf/inv: 0

Number of references: 5

Combinational area: 3998.822379

Buf/Inv area: 944.640008

Noncombinational area: 2278.195181

Macro/Black Box area: 0.000000

Net Interconnect area: 195.157166

Total cell area: 6277.017560

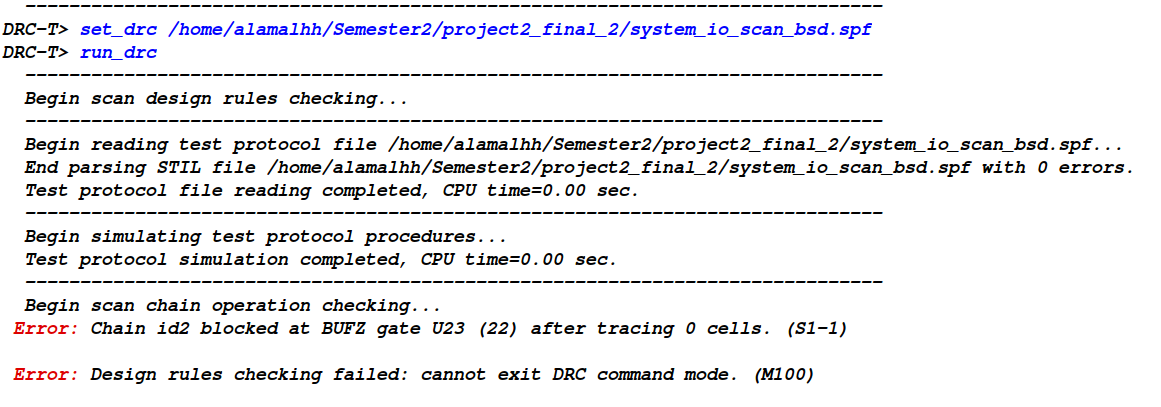
Total area: 6472.174726

Information: This design contains unmapped logic. (RPT-7)

The area before and after the SCANCH instruction shows that there is a modification inside the logic at the logic level.

**Generation of Tmax ATPG patterns :**

The ATPG patterns are tried to be generated from the Tetramax tool, and the following errors is seen at the pattern generation level.



This looks like there is an Error while generating the TAP controller insertion but from the report the compliance is passed and there were no severe Errors.

Note: We were able to generate patterns with 36% coverage but they were incorrect as they were from all the input and outputs pins but not just the scan In and Scan Out ports.

**Some additional reports generated: ( only first 4 pages are submitted):**

**The BSD patterns generated from the Design controller:\**

STIL 1.0 {

Design 2005;

}

Header {

Title "Minimal STIL for design `TOP'";

Date "Wed Mar 16 18:47:10 2016";

Source "DFT Compiler H-2013.03-SP2";

}

Signals {

"CLK" In;

"GO\_I" In;

"RST" In;

"TCK" In;

"TDI" In;

"TEST\_SE" In;

"TEST\_SI" In;

"TMS" In;

"TRSTN" In;

"X\_I[0]" In;

"X\_I[1]" In;

"X\_I[2]" In;

"X\_I[3]" In;

"Y\_I[0]" In;

"Y\_I[1]" In;

"Y\_I[2]" In;

"Y\_I[3]" In;

"D\_O[0]" Out;

"D\_O[1]" Out;

"D\_O[2]" Out;

"D\_O[3]" Out;

"TDO" Out;

"TEST\_SO" Out;

}

SignalGroups {

"all\_inputs" = '"CLK" + "GO\_I" + "RST" + "TCK" + "TDI" + "TEST\_SE" +

"TEST\_SI" + "TMS" + "TRSTN" + "X\_I[0]" + "X\_I[1]" + "X\_I[2]" + "X\_I[3]" +

"Y\_I[0]" + "Y\_I[1]" + "Y\_I[2]" + "Y\_I[3]"';

"all\_outputs" = '"D\_O[0]" + "D\_O[1]" + "D\_O[2]" + "D\_O[3]" + "TDO" +

"TEST\_SO"';

"all\_ports" = '"all\_inputs" + "all\_outputs"';

"\_pi" = '"all\_inputs"';

"\_po" = '"all\_outputs"';

}

Timing {

WaveformTable "\_default\_WFT\_" {

Period '100ns';

Waveforms {

"all\_inputs" {

0 {

'0ns' D;

}

}

"all\_inputs" {

1 {

'0ns' U;

PatternBurst "\_BSD\_burst\_" {

PatList {

"\_BSD\_block\_";

}

}

PatternExec {

PatternBurst "\_BSD\_burst\_";

}

Pattern "\_BSD\_block\_" {

W "BSD\_WFT";

C {

"all\_inputs" = 00100000000000000;

"all\_outputs" = \r6 X;

}

"pattern 0 Async Reset Vectors Test-Logic-Reset" : V {

"all\_inputs" = 0N1PN0NN0NNNNNNNN;

"all\_outputs" = XXXXTX;

}

"pattern 1 Async Reset Vectors Run-Test/Idle" : V {

"all\_inputs" = 0N1PN0N01NNNNNNNN;

"all\_outputs" = XXXXTX;

}

"pattern 2 Async Reset Vectors Select-DR-Scan" : V {

"all\_inputs" = 0N1PN0N11NNNNNNNN;

"all\_outputs" = XXXXTX;

}

"pattern 3 Async Reset Vectors Capture-DR" : V {

"all\_inputs" = 0N1PN0N01NNNNNNNN;

"all\_outputs" = XXXXTX;

}

"pattern 4 Async Reset Vectors Shift-DR" : V {

"all\_inputs" = 0N1PN0N01NNNNNNNN;

"all\_outputs" = XXXXLX;

}

"pattern 5 Async Reset Vectors Shift-DR" : V {

"all\_inputs" = 0N1P10N01NNNNNNNN;

"all\_outputs" = XXXXHX;

}

"pattern 6 Async Reset Vectors Shift-DR" : V {

"all\_inputs" = 0N1P10N01NNNNNNNN;

"all\_outputs" = XXXXHX;

}

"pattern 7 Async Reset Vectors Shift-DR" : V {

"all\_inputs" = 0N1P00N01NNNNNNNN;

"all\_outputs" = XXXXLX;

}

"pattern 8 Async Reset Vectors Shift-DR" : V {

"all\_inputs" = 0N1P00N01NNNNNNNN;

"all\_outputs" = XXXXLX;

}

"pattern 9 Async Reset Vectors Exit1-DR" : V {

"all\_inputs" = 0N1P10N11NNNNNNNN;

"all\_outputs" = XXXXTX;

}

"pattern 10 Async Reset Vectors Update-DR" : V {

"all\_inputs" = 0N1P10N11NNNNNNNN;

"all\_outputs" = XXXXTX;

}

"pattern 11 Async Reset Vectors Run-Test/Idle" : V {

"all\_inputs" = 0N1P10N01NNNNNNNN;

"all\_outputs" = XXXXTX;

}

"pattern 12 Sync Reset Vectors" : V {

"all\_inputs" = 0N1P10N11NNNNNNNN;

"all\_outputs" = XXXXTX;

}

"pattern 13 Sync Reset Vectors" : V {

"all\_inputs" = 0N1P10N11NNNNNNNN;

"all\_outputs" = XXXXTX;

}

"pattern 14 Sync Reset Vectors" : V {

"all\_inputs" = 0N1P10N11NNNNNNNN;

"all\_outputs" = XXXXTX;

}

"pattern 15 Sync Reset Vectors" : V {

"all\_inputs" = 0N1P10N11NNNNNNNN;

"all\_outputs" = XXXXTX;

}

"pattern 16 Sync Reset Vectors Test-Logic-Reset" : V {

"all\_inputs" = 0N1P10N11NNNNNNNN;

"all\_outputs" = XXXXTX;

}

"pattern 17 Sync Reset Vectors Run-Test/Idle" : V {

"all\_inputs" = 0N1P10N01NNNNNNNN;

"all\_outputs" = XXXXTX;

}

"pattern 18 Sync Reset Vectors Select-DR-Scan" : V {

"all\_inputs" = 0N1P10N11NNNNNNNN;

"all\_outputs" = XXXXTX;

}

"pattern 19 Sync Reset Vectors Capture-DR" : V {

"all\_inputs" = 0N1P10N01NNNNNNNN;

"all\_outputs" = XXXXTX;

}

"pattern 20 Sync Reset Vectors Shift-DR" : V {

"all\_inputs" = 0N1P10N01NNNNNNNN;

"all\_outputs" = XXXXLX;

}

"pattern 21 Sync Reset Vectors Shift-DR" : V {

"all\_inputs" = 0N1P10N01NNNNNNNN;

"all\_outputs" = XXXXHX;

}

"pattern 22 Sync Reset Vectors Shift-DR" : V {

"all\_inputs" = 0N1P10N01NNNNNNNN;

"all\_outputs" = XXXXHX;

}

"pattern 23 Sync Reset Vectors Shift-DR" : V {

"all\_inputs" = 0N1P00N01NNNNNNNN;

"all\_outputs" = XXXXLX;

}

"pattern 24 Sync Reset Vectors Shift-DR" : V {

"all\_inputs" = 0N1P00N01NNNNNNNN;

"all\_outputs" = XXXXLX;

}

"pattern 25 Sync Reset Vectors Exit1-DR" : V {

"all\_inputs" = 0N1P10N11NNNNNNNN;

"all\_outputs" = XXXXTX;

}

"pattern 26 Sync Reset Vectors Update-DR" : V {

"all\_inputs" = 0N1P10N11NNNNNNNN;

"all\_outputs" = XXXXTX;

}

"pattern 27 Sync Reset Vectors Select-DR-Scan" : V {

"all\_inputs" = 0N1P10N11NNNNNNNN;

"all\_outputs" = XXXXTX;

}

"pattern 28 Sync Reset Vectors Select-IR-Scan" : V {

"all\_inputs" = 0N1P10N11NNNNNNNN;

There are 695 patterns generated for the BSD logic, these are BSD patterns:

**Test bench generated from the Design compiler after boundary scan insertion and TAP insertion:**

// Native Verilog Test Bench File generated by write\_test command.

// Design : TOP

// Version : Synopsys Version H-2013.03-SP2

// Date : Wed Mar 16 18:47:10 2016

`timescale 1 ns / 10 ps

// Period : 100

// TCK Rise Time : 45

// TCK Fall Time : 55

// Input Delay : 0

// Bidir Delay : 0

// Strobe Time : 95

// Power up Reset Delay : 0

module TOP\_test;

/\* Flag to indicate if the simulation has passed. \*/

integer \_failed;

/\* Clock register. \*/

reg [0:0] \_ck;

/\* Primary Inputs \*/

reg [15:0] \_pi;

/\* Expected Primary Outputs & Primary Output Mask. \*/

reg [5:0] \_e\_po, \_m\_po;

/\* Observed Primary Outputs. \*/

wire [5:0] \_r\_po;

/\* Event variable that checks on primary outputs. \*/

event \_check\_po;

// Inputs

wire CLK, GO\_I, RST, TCK, TDI, TEST\_SE, TEST\_SI, TMS, TRSTN;

wire [3:0] X\_I;

wire [3:0] Y\_I;

// Outputs

wire TDO, TEST\_SO;

wire [3:0] D\_O;

assign {TCK} = \_ck;

/\* Input signal assignment \*/

assign {CLK,GO\_I,RST,TDI,TEST\_SE,TEST\_SI,TMS,TRSTN,X\_I[0],X\_I[1],X\_I[2],X\_I[3],

Y\_I[0],Y\_I[1],Y\_I[2],Y\_I[3]} = \_pi;

/\* Output Signal Assignment \*/

assign \_r\_po = {D\_O[0],D\_O[1],D\_O[2],D\_O[3],TDO,TEST\_SO};

/\* Register variable for pattern comments. \*/

reg [8\*50:0] pattern\_comment;

integer index;

/\* Event that checks on primary outputs. \*/

always @(\_check\_po) begin

for (index = 0; index < 7 ; index = index + 1) begin

if ((\_m\_po[index]) === 1'b1) begin

if ((\_e\_po[index]) !== (\_r\_po[index])) begin

$write("At time %t ",$time);

$write(": Incorrect output response (%b, expected %b)\n", \_r\_po[index], \_e\_po[index]);

$write("At label");

$write("\"%s\"\n\n",pattern\_comment);

\_failed = \_failed + 1;

end

end

end

end

TOP T1 ( .CLK(CLK), .GO\_I(GO\_I), .RST(RST), .TCK(TCK), .TDI(TDI), .TEST\_SE(

TEST\_SE), .TEST\_SI(TEST\_SI), .TMS(TMS), .TRSTN(TRSTN), .X\_I(X\_I), .Y\_I(Y\_I),

.TDO(TDO), .TEST\_SO(TEST\_SO), .D\_O(D\_O) );

initial begin

\_failed = 0;

/\* Initialization vectors. \*/

assign \_ck=1'b0;

assign \_pi=16'bXXXXXXXXXXXXXXXX;

#200 ; /\* 200 \*/

assign \_pi=16'b0X1X0XX0XXXXXXXX;

#45 ; /\* 245 \*/

assign \_ck=1'b1;

#10 ; /\* 255 \*/

assign \_ck=1'b0;

assign pattern\_comment ="pattern 0 Async Reset Vectors Test-Logic-Reset";

#40 ; /\* 295 \*/

assign \_e\_po=6'bXXXXZX;

assign \_m\_po=6'b000010;

-> \_check\_po;

#5 ; /\* 300 \*/

assign \_pi=16'b0X1X0X01XXXXXXXX;

#45 ; /\* 345 \*/

assign \_ck=1'b1;

#10 ; /\* 355 \*/

assign \_ck=1'b0;

assign pattern\_comment ="pattern 1 Async Reset Vectors Run-Test/Idle";

#40 ; /\* 395 \*/

assign \_e\_po=6'bXXXXZX;

-> \_check\_po;

#5 ; /\* 400 \*/

assign \_pi=16'b0X1X0X11XXXXXXXX;

#45 ; /\* 445 \*/

assign \_ck=1'b1;

#10 ; /\* 455 \*/

assign \_ck=1'b0;

assign pattern\_comment ="pattern 2 Async Reset Vectors Select-DR-Scan";

#40 ; /\* 495 \*/

assign \_e\_po=6'bXXXXZX;

-> \_check\_po;

#5 ; /\* 500 \*/

assign \_pi=16'b0X1X0X01XXXXXXXX;

#45 ; /\* 545 \*/

assign \_ck=1'b1;

#10 ; /\* 555 \*/

assign \_ck=1'b0;

assign pattern\_comment ="pattern 3 Async Reset Vectors Capture-DR";

#40 ; /\* 595 \*/

assign \_e\_po=6'bXXXXZX;

-> \_check\_po;

#0 ; /\* 595 \*/

#50 ; /\* 645 \*/

assign \_ck=1'b1;

#10 ; /\* 655 \*/

assign \_ck=1'b0;

assign pattern\_comment ="pattern 4 Async Reset Vectors Shift-DR";

#40 ; /\* 695 \*/

assign \_e\_po=6'bXXXX0X;

-> \_check\_po;

#5 ; /\* 700 \*/

assign \_pi=16'b0X110X01XXXXXXXX;

#45 ; /\* 745 \*/

assign \_ck=1'b1;

#10 ; /\* 755 \*/

assign \_ck=1'b0;

assign pattern\_comment ="pattern 5 Async Reset Vectors Shift-DR";

#40 ; /\* 795 \*/

assign \_e\_po=6'bXXXX1X;

-> \_check\_po;

#0 ; /\* 795 \*/

#50 ; /\* 845 \*/

assign \_ck=1'b1;

#10 ; /\* 855 \*/

assign \_ck=1'b0;

assign pattern\_comment ="pattern 6 Async Reset Vectors Shift-DR";

#40 ; /\* 895 \*/

assign \_e\_po=6'bXXXX1X;

-> \_check\_po;

#5 ; /\* 900 \*/

assign \_pi=16'b0X100X01XXXXXXXX;

#45 ; /\* 945 \*/

assign \_ck=1'b1;

#10 ; /\* 955 \*/

assign \_ck=1'b0;

assign pattern\_comment ="pattern 7 Async Reset Vectors Shift-DR";

#40 ; /\* 995 \*/

assign \_e\_po=6'bXXXX0X;

-> \_check\_po;

#0 ; /\* 995 \*/

#50 ; /\* 1045 \*/

assign \_ck=1'b1;

#10 ; /\* 1055 \*/

assign \_ck=1'b0;

assign pattern\_comment ="pattern 8 Async Reset Vectors Shift-DR";

#40 ; /\* 1095 \*/

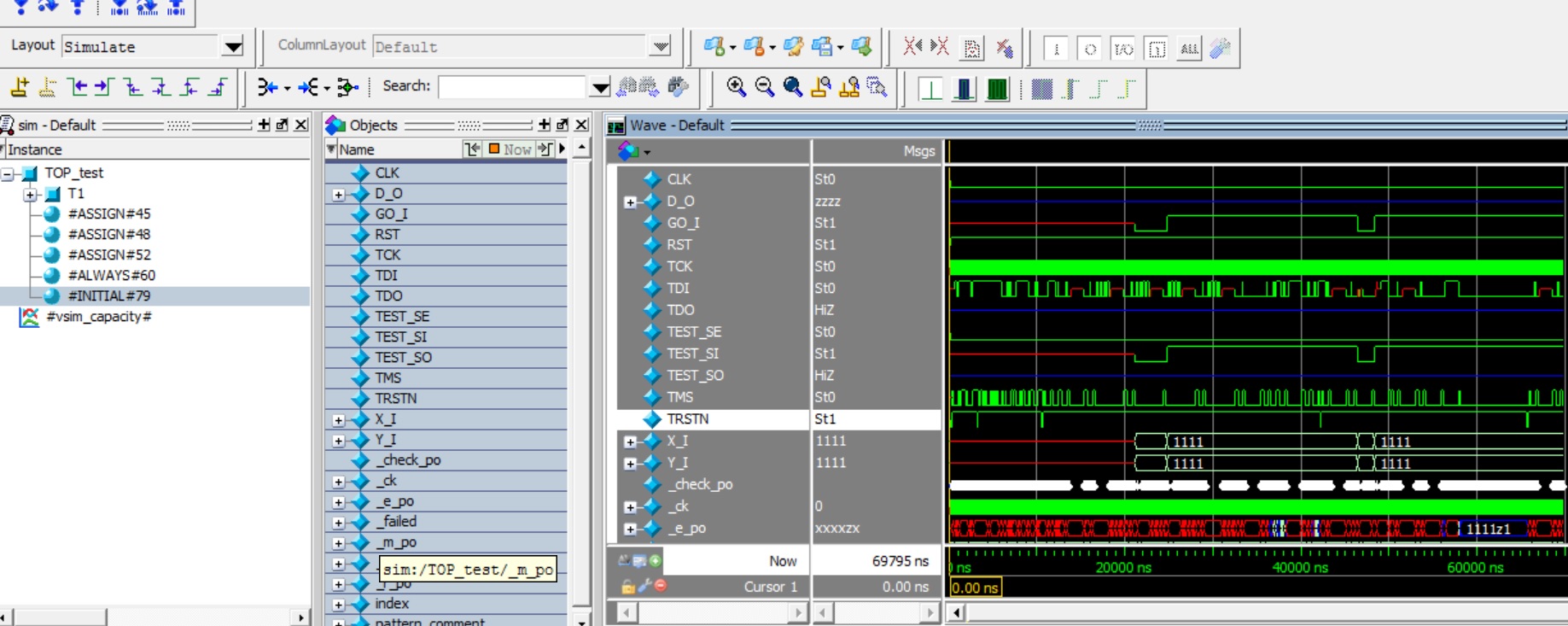
assign \_e\_po=6'bXXXX0X;

-> \_check\_po;

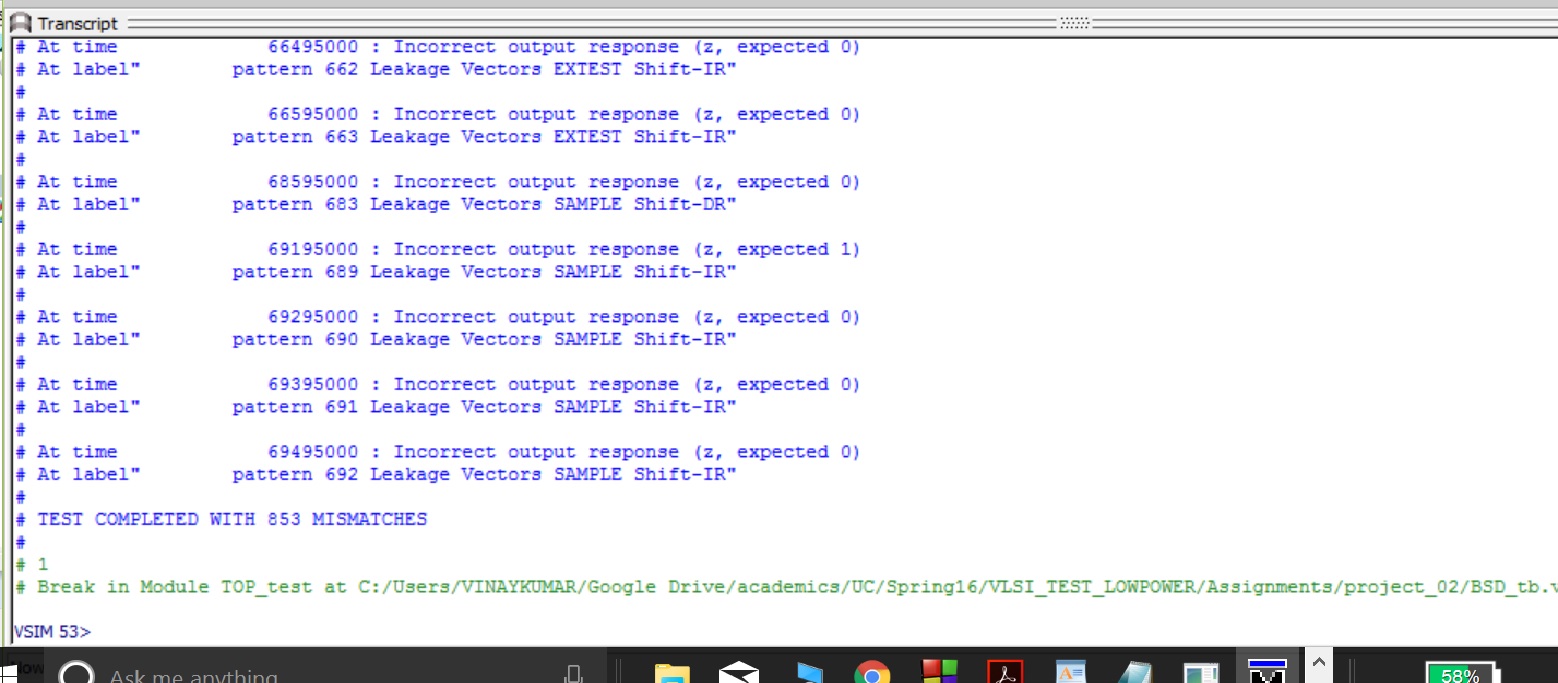
**Snapshots for the simulation with the TAP controller with the test bench generated by Design Compiler:**

The following is the snapshot for the simulation with test bench generated from the Design Compiler, without any SCANCH instruction.

But by default there are some Error been reported in the simulation because of x-propagation in the simulation.



The following are the Errors been reported by the tool after the simulation with the test bench generated by the Design Compiler itself.These errors because of the z propagation.



**BSDL file generated by Design Compiler:**

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- BSDL file for design TOP

-- Created by Synopsys Version H-2013.03-SP2 (Jun 03, 2013)

entity TOP is

-- Designer:

-- Company:

-- Date: Wed Mar 16 18:47:10 2016

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- This section identifies the default device package selected.

generic (PHYSICAL\_PIN\_MAP: string:= "my\_package");

-- This section declares all the ports in the design.

port (

CLK : in bit;

GO\_I : in bit;

RST : in bit;

TCK : in bit;

TDI : in bit;

TEST\_SE : in bit;

TEST\_SI : in bit;

TMS : in bit;

TRSTN : in bit;

X\_I : in bit\_vector (0 to 3);

Y\_I : in bit\_vector (0 to 3);

TDO : out bit;

TEST\_SO : buffer bit;

D\_O : buffer bit\_vector (0 to 3)

);

use STD\_1149\_1\_2001.all;

attribute COMPONENT\_CONFORMANCE of TOP: entity is "STD\_1149\_1\_2001";

attribute PIN\_MAP of TOP: entity is PHYSICAL\_PIN\_MAP;

-- This section specifies the pin map for each port. This information is

-- extracted from the port-to-pin map file that was read in using the

-- "read\_pin\_map" command.

constant my\_package: PIN\_MAP\_STRING :=

"CLK : P1," &

"GO\_I : P3," &

"RST : P2," &

"TCK : P23," &

"TDI : P21," &

"TEST\_SE : P18," &

"TEST\_SI : P16," &

"TMS : P22," &

"TRSTN : P19," &

"X\_I : (P4, P5, P6, P7)," &

"Y\_I : (P8, P9, P10, P11)," &

"TDO : P20," &

"TEST\_SO : P17," &

"D\_O : (P12, P13, P14, P15)";

-- This section specifies the TAP ports. For the TAP TCK port, the parameters in

-- the brackets are:

-- First Field : Maximum TCK frequency.

-- Second Field: Allowable states TCK may be stopped in.

attribute TAP\_SCAN\_CLOCK of TCK : signal is (10.0e6, BOTH);

attribute TAP\_SCAN\_IN of TDI : signal is true;

attribute TAP\_SCAN\_MODE of TMS : signal is true;

attribute TAP\_SCAN\_OUT of TDO : signal is true;

attribute TAP\_SCAN\_RESET of TRSTN: signal is true;

-- Specifies the compliance enable patterns for the design. It lists a set of

-- design ports and the values that they should be set to, in order to enable

-- compliance to IEEE Std 1149.1

attribute COMPLIANCE\_PATTERNS of TOP: entity is

"(RST, TEST\_SE) (10)";

-- Specifies the number of bits in the instruction register.

attribute INSTRUCTION\_LENGTH of TOP: entity is 4;

-- Specifies the boundary-scan instructions implemented in the design and their

-- opcodes.

attribute INSTRUCTION\_OPCODE of TOP: entity is

"BYPASS (1111)," &

"EXTEST (0001)," &

"SAMPLE (0100)," &

"PRELOAD (0100)," &

"CLAMP (0010)";

-- Specifies the bit pattern that is loaded into the instruction register when

-- the TAP controller passes through the Capture-IR state. The standard mandates

-- that the two LSBs must be "01". The remaining bits are design specific.

attribute INSTRUCTION\_CAPTURE of TOP: entity is "0001";

-- This section specifies the test data register placed between TDI and TDO for

-- each implemented instruction.

attribute REGISTER\_ACCESS of TOP: entity is

"BYPASS (BYPASS, CLAMP)," &

"BOUNDARY (EXTEST, SAMPLE, PRELOAD)";

-- Specifies the length of the boundary scan register.

attribute BOUNDARY\_LENGTH of TOP: entity is 16;

-- The following list specifies the characteristics of each cell in the boundary

-- scan register from TDI to TDO. The following is a description of the label

-- fields:

-- num : Is the cell number.

-- cell : Is the cell type as defined by the standard.

-- port : Is the design port name. Control cells do not have a port

-- name.

-- function: Is the function of the cell as defined by the standard. Is one

-- of input, output2, output3, bidir, control or controlr.

-- safe : Specifies the value that the BSR cell should be loaded with

-- for safe operation when the software might otherwise choose a

-- random value.

-- ccell : The control cell number. Specifies the control cell that

-- drives the output enable for this port.

-- disval : Specifies the value that is loaded into the control cell to

-- disable the output enable for the corresponding port.

-- rslt : Resulting state. Shows the state of the driver when it is

-- disabled.

attribute BOUNDARY\_REGISTER of TOP: entity is

--

-- num cell port function safe [ccell disval rslt]

--

"15 (BC\_4, CLK, clock, X), " &

"14 (BC\_2, GO\_I, input, X), " &

"13 (BC\_2, X\_I(0), input, X), " &

"12 (BC\_2, X\_I(1), input, X), " &

"11 (BC\_2, X\_I(2), input, X), " &

"10 (BC\_2, X\_I(3), input, X), " &

"9 (BC\_2, Y\_I(0), input, X), " &

"8 (BC\_2, Y\_I(1), input, X), " &

"7 (BC\_2, Y\_I(2), input, X), " &

"6 (BC\_2, Y\_I(3), input, X), " &

"5 (BC\_1, D\_O(0), output2, X), " &

"4 (BC\_1, D\_O(1), output2, X), " &

"3 (BC\_1, D\_O(2), output2, X), " &

"2 (BC\_1, D\_O(3), output2, X), " &

"1 (BC\_2, TEST\_SI, input, X), " &

"0 (BC\_1, TEST\_SO, output2, X) ";

end TOP;

**Additional files required for the Tetramax:**

The additional files GTECH\_INBUF and the GTECH\_OUTBUF are designed and passed to the Tetrmax tool:

module GTECH\_INBUF (PAD\_IN,DATA\_IN);

input PAD\_IN;

output DATA\_IN;

assign DATA\_IN= PAD\_IN;

endmodule

module GTECH\_OUTBUF (PAD\_OUT,DATA\_OUT,OE);

input OE;

input DATA\_OUT;

output PAD\_OUT;

assign DATA\_OUT= OE ? PAD\_OUT: 0;

endmodule

**The netlist generated from the Design Compiler is:**

odule fsm\_test\_1 ( rst, clk, proceed, comparison, enable, xsel, ysel, xld,

yld, test\_si, test\_so, test\_se );

input [1:0] comparison;

input rst, clk, proceed, test\_si, test\_se;

output enable, xsel, ysel, xld, yld, test\_so;

wire n7, n8, n3, n6, n9, n10, n11, n12, n13, n14, n15, n16, n17, n18, n19,

n20;

wire [2:0] nState;

SDFFARX1 cState\_reg\_2\_ ( .D(nState[2]), .SI(n6), .SE(test\_se), .CLK(clk),

.RSTB(n10), .Q(n9), .QN(test\_so) );

AND3X1 U19 ( .IN1(n3), .IN2(n9), .IN3(n7), .Q(n14) );

NAND3X0 U20 ( .IN1(n16), .IN2(n11), .IN3(n17), .QN(nState[1]) );

OR2X1 U21 ( .IN1(n6), .IN2(n18), .Q(n17) );

NAND4X0 U22 ( .IN1(comparison[1]), .IN2(comparison[0]), .IN3(test\_so), .IN4(

n3), .QN(n16) );

NAND4X0 U23 ( .IN1(test\_so), .IN2(n19), .IN3(n20), .IN4(n11), .QN(nState[0])

);

OR2X1 U24 ( .IN1(comparison[1]), .IN2(n7), .Q(n20) );

SDFFARX1 cState\_reg\_1\_ ( .D(nState[1]), .SI(n3), .SE(test\_se), .CLK(clk),

.RSTB(n10), .Q(n6), .QN(n7) );

SDFFARX1 cState\_reg\_0\_ ( .D(nState[0]), .SI(test\_si), .SE(test\_se), .CLK(clk), .RSTB(n10), .Q(n3), .QN(n8) );

INVX0 U3 ( .INP(n12), .ZN(ysel) );

NAND2X1 U4 ( .IN1(n18), .IN2(n6), .QN(n11) );

NOR2X0 U5 ( .IN1(n9), .IN2(n3), .QN(n18) );

NAND2X1 U6 ( .IN1(n13), .IN2(n11), .QN(xld) );

INVX0 U7 ( .INP(n13), .ZN(xsel) );

NAND2X1 U8 ( .IN1(n11), .IN2(n12), .QN(yld) );

NOR4X0 U9 ( .IN1(n8), .IN2(n7), .IN3(n15), .IN4(n9), .QN(nState[2]) );

NOR2X0 U10 ( .IN1(comparison[0]), .IN2(comparison[1]), .QN(n15) );

NOR2X0 U11 ( .IN1(n13), .IN2(n7), .QN(enable) );

NAND2X1 U12 ( .IN1(proceed), .IN2(n8), .QN(n19) );

NOR2X0 U13 ( .IN1(enable), .IN2(n14), .QN(n12) );

NAND2X1 U14 ( .IN1(n8), .IN2(n9), .QN(n13) );

INVX0 U15 ( .INP(rst), .ZN(n10) );

endmodule

module mux\_0 ( rst, sLine, load, result, output0 );

input [3:0] load;

input [3:0] result;

output [3:0] output0;

input rst, sLine;

wire n1, n2, n3;

AO22X1 U5 ( .IN1(result[3]), .IN2(n2), .IN3(load[3]), .IN4(n3), .Q(

output0[3]) );

AO22X1 U6 ( .IN1(result[2]), .IN2(n2), .IN3(load[2]), .IN4(n3), .Q(

output0[2]) );

AO22X1 U7 ( .IN1(result[1]), .IN2(n2), .IN3(load[1]), .IN4(n3), .Q(

output0[1]) );

AO22X1 U8 ( .IN1(result[0]), .IN2(n2), .IN3(load[0]), .IN4(n3), .Q(

output0[0]) );

NOR2X0 U2 ( .IN1(rst), .IN2(sLine), .QN(n3) );

NOR2X0 U3 ( .IN1(n1), .IN2(rst), .QN(n2) );

INVX0 U4 ( .INP(sLine), .ZN(n1) );

endmodule

module mux\_1 ( rst, sLine, load, result, output0 );

input [3:0] load;

input [3:0] result;

output [3:0] output0;

input rst, sLine;

wire n1, n2, n3;

AO22X1 U5 ( .IN1(result[3]), .IN2(n2), .IN3(load[3]), .IN4(n3), .Q(

output0[3]) );

AO22X1 U6 ( .IN1(result[2]), .IN2(n2), .IN3(load[2]), .IN4(n3), .Q(

output0[2]) );

AO22X1 U7 ( .IN1(result[1]), .IN2(n2), .IN3(load[1]), .IN4(n3), .Q(

output0[1]) );

AO22X1 U8 ( .IN1(result[0]), .IN2(n2), .IN3(load[0]), .IN4(n3), .Q(

output0[0]) );

NOR2X0 U2 ( .IN1(rst), .IN2(sLine), .QN(n3) );

NOR2X0 U3 ( .IN1(n1), .IN2(rst), .QN(n2) );

INVX0 U4 ( .INP(sLine), .ZN(n1) );

endmodule

module regis\_test\_0 ( rst, clk, load, input0, output0, test\_si, test\_so,

test\_se );

input [3:0] input0;

output [3:0] output0;

input rst, clk, load, test\_si, test\_se;

output test\_so;

wire n2, n3, n4, n6, n1, n5, n7, n8, n9;

AO22X1 U4 ( .IN1(n5), .IN2(output0[3]), .IN3(load), .IN4(input0[3]), .Q(n6)

);

AO22X1 U5 ( .IN1(output0[2]), .IN2(n5), .IN3(input0[2]), .IN4(load), .Q(n4)

);

AO22X1 U6 ( .IN1(output0[1]), .IN2(n5), .IN3(input0[1]), .IN4(load), .Q(n3)

);

AO22X1 U7 ( .IN1(output0[0]), .IN2(n5), .IN3(input0[0]), .IN4(load), .Q(n2)

);

SDFFARX1 output\_reg\_0\_ ( .D(n2), .SI(test\_si), .SE(test\_se), .CLK(clk),

.RSTB(n1), .Q(output0[0]), .QN(n9) );

SDFFARX1 output\_reg\_2\_ ( .D(n4), .SI(n8), .SE(test\_se), .CLK(clk), .RSTB(n1),

.Q(output0[2]), .QN(n7) );

SDFFARX1 output\_reg\_3\_ ( .D(n6), .SI(n7), .SE(test\_se), .CLK(clk), .RSTB(n1),

.Q(output0[3]), .QN(test\_so) );

SDFFARX1 output\_reg\_1\_ ( .D(n3), .SI(n9), .SE(test\_se), .CLK(clk), .RSTB(n1),

.Q(output0[1]), .QN(n8) );

INVX0 U2 ( .INP(load), .ZN(n5) );

INVX0 U3 ( .INP(rst), .ZN(n1) );

endmodule

module regis\_test\_1 ( rst, clk, load, input0, output0, test\_si, test\_so,

test\_se );

input [3:0] input0;

output [3:0] output0;

input rst, clk, load, test\_si, test\_se;

output test\_so;

wire n7, n9, n10, n11, n1, n2, n3, n4, n5;

SDFFARX1 output\_reg\_3\_ ( .D(n7), .SI(n3), .SE(test\_se), .CLK(clk), .RSTB(n1),

.Q(output0[3]), .QN(test\_so) );

SDFFARX1 output\_reg\_2\_ ( .D(n9), .SI(n4), .SE(test\_se), .CLK(clk), .RSTB(n1),

.Q(output0[2]), .QN(n3) );

SDFFARX1 output\_reg\_1\_ ( .D(n10), .SI(n5), .SE(test\_se), .CLK(clk), .RSTB(n1), .Q(output0[1]), .QN(n4) );

SDFFARX1 output\_reg\_0\_ ( .D(n11), .SI(test\_si), .SE(test\_se), .CLK(clk),

.RSTB(n1), .Q(output0[0]), .QN(n5) );

AO22X1 U4 ( .IN1(output0[2]), .IN2(n2), .IN3(load), .IN4(input0[2]), .Q(n9)

);

AO22X1 U5 ( .IN1(n2), .IN2(output0[3]), .IN3(input0[3]), .IN4(load), .Q(n7)

);

AO22X1 U6 ( .IN1(output0[0]), .IN2(n2), .IN3(input0[0]), .IN4(load), .Q(n11)

);

AO22X1 U7 ( .IN1(output0[1]), .IN2(n2), .IN3(input0[1]), .IN4(load), .Q(n10)

);

INVX0 U2 ( .INP(load), .ZN(n2) );

INVX0 U3 ( .INP(rst), .ZN(n1) );

endmodule

module TOP\_DW\_tap\_uc\_width4\_id0\_idcode\_opcode1\_version0\_part0\_man\_num0\_sync\_mode1 (

tck, trst\_n, tms, tdi, so, bypass\_sel, sentinel\_val, device\_id\_sel,

user\_code\_sel, user\_code\_val, ver, ver\_sel, part\_num, part\_num\_sel,

mnfr\_id, mnfr\_id\_sel, clock\_dr, shift\_dr, update\_dr, tdo, tdo\_en,

tap\_state, instructions, sync\_capture\_en, sync\_update\_dr, test );

input [2:0] sentinel\_val;

input [31:0] user\_code\_val;

input [3:0] ver;

input [15:0] part\_num;

input [10:0] mnfr\_id;

output [15:0] tap\_state;

output [3:0] instructions;

input tck, trst\_n, tms, tdi, so, bypass\_sel, device\_id\_sel, user\_code\_sel,

ver\_sel, part\_num\_sel, mnfr\_id\_sel, test;

output clock\_dr, shift\_dr, update\_dr, tdo, tdo\_en, sync\_capture\_en,

sync\_update\_dr;

wire tck\_n, dr\_sel\_int, shift\_ir\_int, net36, net39, tdo\_ffin, U6\_ffin,

U5\_U1\_3\_ffin, U5\_U1\_2\_ffin, U5\_U1\_1\_ffin, U5\_U1\_0\_ffin, U5\_rst\_n,

net44, net45, net46, U1\_N35, U1\_int\_tck\_n, net63, U1\_cs15\_n, U1\_cs8\_n,

U1\_sel\_cap\_ir, U1\_sel\_cap\_dr, U1\_dr\_sel\_int\_next, U1\_next\_state\_0\_,

U1\_next\_state\_1\_, U1\_next\_state\_2\_, U1\_next\_state\_3\_,

U1\_next\_state\_4\_, U1\_next\_state\_5\_, U1\_next\_state\_6\_,

U1\_next\_state\_7\_, U1\_next\_state\_8\_, U1\_next\_state\_9\_,

U1\_next\_state\_10\_, U1\_next\_state\_11\_, U1\_next\_state\_12\_,

U1\_next\_state\_13\_, U1\_next\_state\_14\_, U1\_next\_state\_15\_, net403,

net402, net401, net400, net399, net398, net397, net396, net395,

net394, net393, net392, net391, net390, net389, net388, net387,

net386, net385, net384, net383, net382, net381, net380, net379,

net378, net377, net376, net375, net374, net373, net372, net371,

net370, net369, net368, net367, net366, net365, net364, net363,

net362, net361, net360, net359, net358, net357, net356, net355,

net354, net353, net352, net351, net350, net349, net348, net347,

net346, net345, net344, net343, net342, net341, net340, net339,

net338, net337, net336, net335, net334, net333, net332, net331,

net330, net329, net328, net327, net326, net325, net324, net323,

net322, net321, net320, net319, net318, net317, net316, net315,

net314, net313, net312, net311, net310, net309, net308, net307,

net306, net305, net304, net303, net302, net301, net300, net299,

net298, net297, net296, net295, net294, net293, net292, net291,

net290, net289, net288, net287, net286, net285, net284, net283,

net282, net281, net280, net279, net278, net277, net276, net275,

net274, net273, net272, net271, net270, net269, net268, net267,

net266, net265, net264, net263, net262, net261, net260, net259,

net258, net257, net256, net255, net254, net253, net252, net251,

net250, net249, net248, net247, net246, net245, net244, net243,

net242;

wire [3:0] U5\_ff\_d;

NOR2X2 U1\_U1 ( .IN1(tap\_state[3]), .IN2(tap\_state[4]), .QN(U1\_sel\_cap\_dr) );

OR2X1 U1\_U3 ( .IN1(tck), .IN2(U1\_sel\_cap\_dr), .Q(clock\_dr) );

NOR2X2 U1\_U2 ( .IN1(tap\_state[10]), .IN2(tap\_state[11]), .QN(U1\_sel\_cap\_ir)

);

OR2X1 U1\_U4 ( .IN1(tck), .IN2(U1\_sel\_cap\_ir) );

NOR2X2 U1\_U5 ( .IN1(tck), .IN2(U1\_cs8\_n), .QN(update\_dr) );

NOR2X2 U1\_U6 ( .IN1(tck), .IN2(U1\_cs15\_n) );

INVX1 U1\_U11 ( .INP(tck), .ZN(U1\_int\_tck\_n) );

INVX1 U1\_U12 ( .INP(tap\_state[15]), .ZN(U1\_cs15\_n) );

INVX1 U1\_U13 ( .INP(tap\_state[8]), .ZN(U1\_cs8\_n) );

INVX1 u\_cell\_556 ( .INP(trst\_n), .ZN(U1\_N35) );

NAND2X2 u\_cell\_554 ( .IN1(tap\_state[2]), .IN2(net337), .QN(net400) );

NOR2X2 u\_cell\_553 ( .IN1(tap\_state[5]), .IN2(tap\_state[3]), .QN(net401) );

NOR2X2 u\_cell\_552 ( .IN1(tap\_state[9]), .IN2(tap\_state[6]), .QN(net403) );

NAND2X2 u\_cell\_549 ( .IN1(net350), .IN2(net381), .QN(net330) );

INVX1 u\_cell\_548 ( .INP(net330), .ZN(net322) );

NAND2X2 u\_cell\_547 ( .IN1(net403), .IN2(net322), .QN(net336) );

INVX1 u\_cell\_546 ( .INP(net336), .ZN(net402) );

NAND2X2 u\_cell\_545 ( .IN1(net401), .IN2(net402), .QN(net372) );

NOR2X2 u\_cell\_544 ( .IN1(net400), .IN2(net372), .QN(net393) );

NOR2X2 u\_cell\_543 ( .IN1(tap\_state[1]), .IN2(tap\_state[15]), .QN(net398) );

NAND2X2 u\_cell\_540 ( .IN1(net374), .IN2(net360), .QN(net399) );

INVX1 u\_cell\_539 ( .INP(net399), .ZN(net352) );

NAND2X2 u\_cell\_538 ( .IN1(net398), .IN2(net352), .QN(net316) );

NOR2X2 u\_cell\_537 ( .IN1(tap\_state[12]), .IN2(tap\_state[0]), .QN(net397) );

INVX1 u\_cell\_536 ( .INP(net397), .ZN(net396) );

NOR2X2 u\_cell\_535 ( .IN1(net316), .IN2(net396), .QN(net344) );

NAND2X2 u\_cell\_532 ( .IN1(net395), .IN2(net345), .QN(net373) );

INVX1 u\_cell\_531 ( .INP(net373), .ZN(net279) );

NAND2X2 u\_cell\_530 ( .IN1(net344), .IN2(net279), .QN(net394) );

INVX1 u\_cell\_529 ( .INP(net394), .ZN(net391) );

NAND2X2 u\_cell\_528 ( .IN1(net393), .IN2(net391), .QN(net280) );

INVX1 u\_cell\_527 ( .INP(net280), .ZN(net338) );

INVX1 u\_cell\_526 ( .INP(tms), .ZN(net281) );

NAND2X2 u\_cell\_525 ( .IN1(net338), .IN2(net281), .QN(net291) );

NOR2X2 u\_cell\_524 ( .IN1(tap\_state[5]), .IN2(tap\_state[4]), .QN(net389) );

NAND2X2 u\_cell\_522 ( .IN1(net391), .IN2(net392), .QN(net390) );

INVX1 u\_cell\_521 ( .INP(net390), .ZN(net334) );

NAND2X2 u\_cell\_520 ( .IN1(net389), .IN2(net334), .QN(net318) );

NOR2X2 u\_cell\_518 ( .IN1(net336), .IN2(net321), .QN(net388) );

INVX1 u\_cell\_517 ( .INP(net388), .ZN(net387) );

NOR2X2 u\_cell\_516 ( .IN1(net318), .IN2(net387), .QN(net386) );

INVX1 u\_cell\_515 ( .INP(net386), .ZN(net383) );

NOR2X2 u\_cell\_514 ( .IN1(net372), .IN2(net337), .QN(net385) );

NAND2X2 u\_cell\_513 ( .IN1(net385), .IN2(net334), .QN(net384) );

NAND2X2 u\_cell\_512 ( .IN1(net383), .IN2(net384), .QN(net382) );

INVX1 u\_cell\_511 ( .INP(net382), .ZN(net289) );

NOR3X0 u\_cell\_510 ( .IN1(net381), .IN2(tap\_state[8]), .IN3(tap\_state[6]),

.QN(net377) );

NOR2X2 u\_cell\_509 ( .IN1(tap\_state[9]), .IN2(tap\_state[3]), .QN(net379) );

INVX1 u\_cell\_508 ( .INP(net318), .ZN(net380) );

NAND2X2 u\_cell\_507 ( .IN1(net379), .IN2(net380), .QN(net378) );

INVX1 u\_cell\_506 ( .INP(net378), .ZN(net329) );

NAND2X2 u\_cell\_505 ( .IN1(net377), .IN2(net329), .QN(net285) );

NAND2X2 u\_cell\_504 ( .IN1(net289), .IN2(net285), .QN(net376) );

NAND2X2 u\_cell\_503 ( .IN1(net376), .IN2(net281), .QN(net290) );

NAND2X2 u\_cell\_502 ( .IN1(net291), .IN2(net290), .QN(net375) );

INVX1 u\_cell\_501 ( .INP(net375), .ZN(U1\_dr\_sel\_int\_next) );

NOR3X0 u\_cell\_500 ( .IN1(net374), .IN2(tap\_state[1]), .IN3(tap\_state[13]),

.QN(net365) );

NOR2X2 u\_cell\_499 ( .IN1(tap\_state[15]), .IN2(tap\_state[0]), .QN(net367) );

NOR2X2 u\_cell\_498 ( .IN1(tap\_state[12]), .IN2(net373), .QN(net368) );

NOR2X2 u\_cell\_497 ( .IN1(tap\_state[4]), .IN2(tap\_state[2]), .QN(net370) );

INVX1 u\_cell\_496 ( .INP(net372), .ZN(net371) );

NAND2X2 u\_cell\_495 ( .IN1(net370), .IN2(net371), .QN(net369) );

INVX1 u\_cell\_494 ( .INP(net369), .ZN(net341) );

NAND2X2 u\_cell\_493 ( .IN1(net368), .IN2(net341), .QN(net355) );

INVX1 u\_cell\_492 ( .INP(net355), .ZN(net315) );

NAND2X2 u\_cell\_491 ( .IN1(net367), .IN2(net315), .QN(net366) );

INVX1 u\_cell\_490 ( .INP(net366), .ZN(net354) );

NAND2X2 u\_cell\_489 ( .IN1(net365), .IN2(net354), .QN(net307) );

NOR2X2 u\_cell\_487 ( .IN1(tap\_state[0]), .IN2(net364), .QN(net363) );

NAND2X2 u\_cell\_486 ( .IN1(net363), .IN2(net279), .QN(net362) );

NOR2X2 u\_cell\_485 ( .IN1(net362), .IN2(net316), .QN(net361) );

NAND2X2 u\_cell\_484 ( .IN1(net361), .IN2(net341), .QN(net302) );

NAND2X2 u\_cell\_483 ( .IN1(net307), .IN2(net302), .QN(net357) );

NOR3X0 u\_cell\_482 ( .IN1(net360), .IN2(tap\_state[1]), .IN3(tap\_state[14]),

.QN(net359) );

NAND2X2 u\_cell\_481 ( .IN1(net359), .IN2(net354), .QN(net303) );

NAND2X2 u\_cell\_480 ( .IN1(net289), .IN2(net303), .QN(net358) );

NOR2X2 u\_cell\_479 ( .IN1(net357), .IN2(net358), .QN(net324) );

NAND2X2 u\_cell\_477 ( .IN1(tap\_state[15]), .IN2(net317), .QN(net356) );

NOR2X2 u\_cell\_476 ( .IN1(net355), .IN2(net356), .QN(net353) );

MUX21X1 u\_cell\_475 ( .IN1(net353), .IN2(net354), .S(tap\_state[1]), .Q(net351) );

NAND2X2 u\_cell\_474 ( .IN1(net351), .IN2(net352), .QN(net347) );

NOR3X0 u\_cell\_473 ( .IN1(net350), .IN2(tap\_state[7]), .IN3(tap\_state[6]),

.QN(net349) );

NAND2X2 u\_cell\_472 ( .IN1(net349), .IN2(net329), .QN(net348) );

NAND2X2 u\_cell\_471 ( .IN1(net347), .IN2(net348), .QN(net346) );

INVX1 u\_cell\_470 ( .INP(net346), .ZN(net292) );

XOR2X1 u\_cell\_469 ( .IN1(net345), .IN2(tap\_state[11]), .Q(net342) );

INVX1 u\_cell\_468 ( .INP(net344), .ZN(net343) );

NOR2X2 u\_cell\_467 ( .IN1(net342), .IN2(net343), .QN(net340) );

NAND2X2 u\_cell\_466 ( .IN1(net340), .IN2(net341), .QN(net339) );

INVX1 u\_cell\_465 ( .INP(net339), .ZN(net305) );

NOR2X2 u\_cell\_464 ( .IN1(net305), .IN2(net338), .QN(net332) );

NAND3X0 u\_cell\_463 ( .IN1(tap\_state[5]), .IN2(net337), .IN3(net321), .QN(

net335) );

NOR2X2 u\_cell\_462 ( .IN1(net335), .IN2(net336), .QN(net333) );

NAND2X2 u\_cell\_461 ( .IN1(net333), .IN2(net334), .QN(net284) );

NAND2X2 u\_cell\_460 ( .IN1(net332), .IN2(net284), .QN(net326) );

NOR2X2 u\_cell\_458 ( .IN1(net330), .IN2(net331), .QN(net328) );

Note:Remaining part of the netlist is omitted.

Important Observations:

The schematic after the addition of the SCANCH command is observed.The area report has been taken as the reference to check whether any change is present after addition of SCANCH instruction.There is a considerable difference in the area after the SCANCH instruction is inserted.

**Procedure to GCD machine using the boundary Scan and internal scan chain:**

The GCD circuit is implemented with the internal scan and the full boundary scan capability,

GCD circuit can be tested using the boundary scan logic .As shown in the previous section,

this boundary scan design is IEEE1149.1 compliant, so testing will proceed as normal for this

Architecture.

The different modes of the JTAG can be controlled by the internal state machine with the help of the TMS pin.

The additional instruction SCANCH is used to access the internal scan chain using the TDI and TDO pins. When this mode is activated the scan chain for B will be accessed using the

TDI and TDO pins of the boundary scan architecture.

According to normal boundary scan procedure:

1. Scan the instruction register to test functionality. Once this is verified, using TMS to set

TDI and TDO to connect to instruction register will be the way to change the currently

execution instruction (assume using this step for following instruction sets).

2. Execute BYPASS instruction to be sure TDI and TDO are connected together properly

through the bypass register.

3. Execute each test, as designed using TetraMax in the previous section, using the

following process:

a. Set all necessary inputs using the SAMPLE/PRELOAD instruction, and scanning

in necessary inputs to boundary scan register.

b. Use the EXTEST instruction to apply the test and sample the output.

c. Use SAMPLE/PRELOAD instruction to access the data and compare to expected

output.

4. Any time a test requires the use of the GCD machine scan chain as designed in previous

sections, the SCANCH instruction is used and the scan chain is connected between TDI

and TDO pins. In this manner the scan chain is used as normal, with the exceptions of

accessing from these boundary scan pins, and the few extra registers in the chain that

must be accounted for in the test pattern.

Using this procedure the boundary scan architecture will first be tested for functionality, and then

the chip, GCD, will receive full testing as intended using the generated TetraMax test

patterns

**Objective of the task:( 2nd Method)**

The objective is to integrate both the scan chains into a single scan chain by making the changes in the netlist directly

**Solution of the Problem:**

We have to two scan chains namely the TDI to TDO which is boundary scan chain and the other one is the internal scan chain which is the TEST\_SI to TEST\_SO. Now there needs to be some netlist changes to be made integrate both the scan chains to a single one.

**Help from the tool to solve(Script):**

There is no help from the tool as we have done this manually to connect the scan chains

The following are the changes done to the netlist generated from the Design Compiler to integrate the scan chains to form a single scan chain.

The the aim is to connect the TDO output wire to the TEST\_SI .so the netlist changesa re done to achieve the same.

1127c1127

< module TOP ( RST, CLK, GO\_I, X\_I, Y\_I, D\_O, TEST\_SO, TEST\_SE, TCK,

---

> module TOP ( RST, CLK, GO\_I, X\_I, Y\_I, D\_O, TEST\_SI, TEST\_SO, TEST\_SE, TCK,

1132c1132

< input RST, CLK, GO\_I, TEST\_SE, TCK, TRSTN, TDI, TMS;

---

> input RST, CLK, GO\_I, TEST\_SI, TEST\_SE, TCK, TRSTN, TDI, TMS;

1156,1157c1156

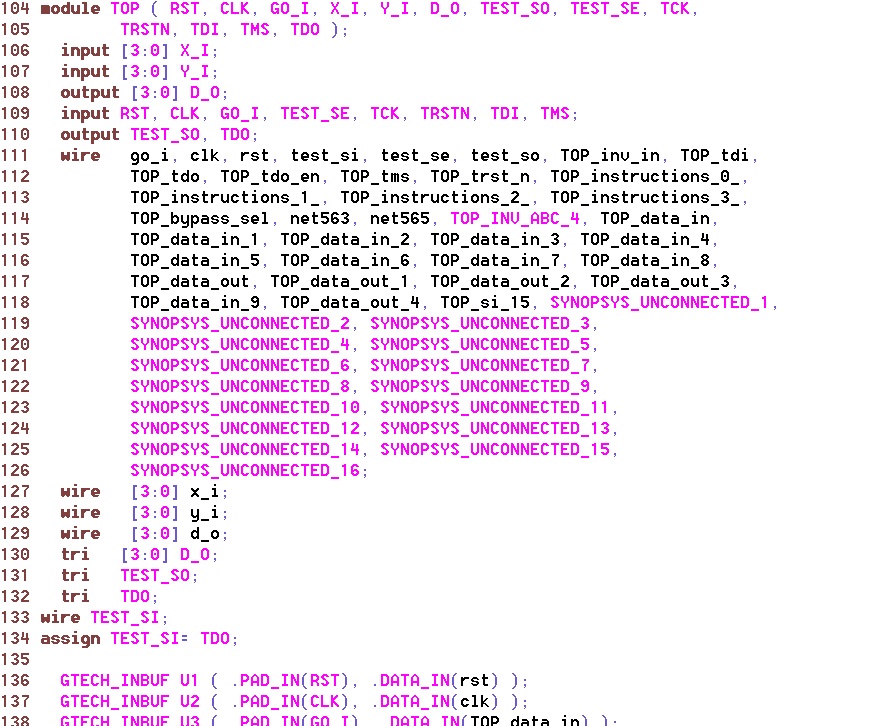
< wire TEST\_SI;

< assign TEST\_SI = TDO;

---

>

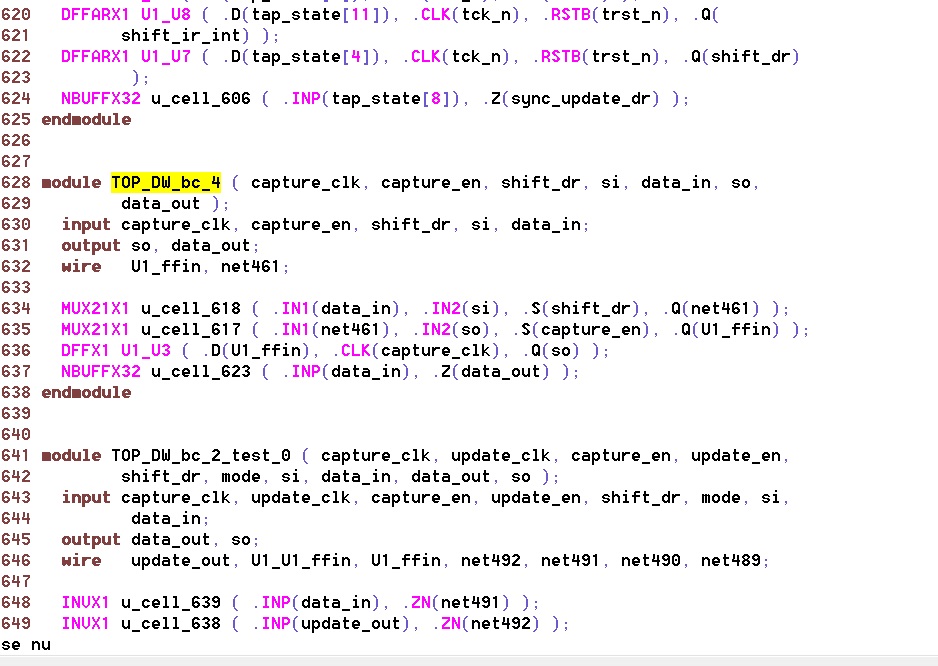
The following are the netlist changes for the integration of the both scan chains



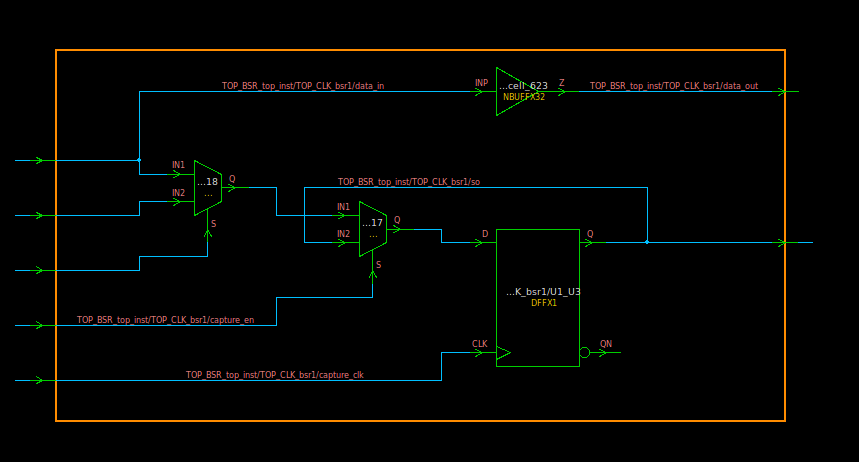
Once the netlist changes are made to connect the TDO to the TEST\_SI input pin and simulation is done using the flush pattern test bench, there are some Errors seen in the feedback path of the boundary scan cell as shown below.

The output of the flop DFFX1 is fed back to the mux IN2 , the value of the output of the flop is X and the same x is fed as the input of the mux and the same repeats making an X propagation in the boundary cell.

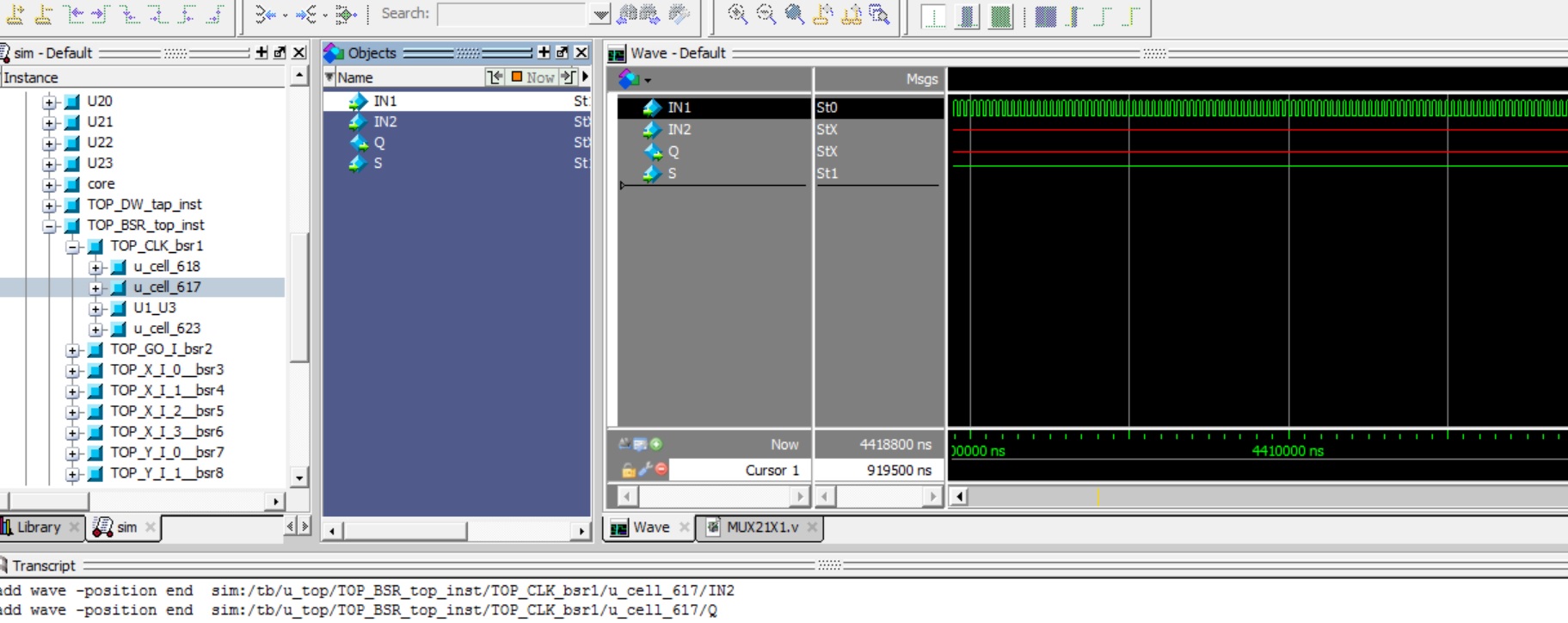
The netlist part of the cell which has X propagation is pointed below here:



The schematic of the cell which has X propagation after the boundary scan insertion is.



The same waveform is shown in the model below.



**The test bench for the Flush pattern is as follows:**

`timescale 1ns/1ps

module tb();

reg [4:0] pattern;

reg tb\_rst;

reg tb\_clk;

reg tb\_go;

reg tb\_x;

reg tb\_y;

wire tb\_d\_o;

wire tb\_test\_so;

reg tb\_test\_se;

wire tb\_tck;

reg tb\_trstn;

reg tb\_tdi;

reg tb\_tms;

wire tb\_tdo;

TOP u\_top (.RST(tb\_rst),

.CLK(tb\_clk),

.GO\_I(tb\_go),

.X\_I(tb\_x),

.Y\_I(tb\_y),

.D\_O(tb\_d\_o),

.TEST\_SO(tb\_test\_so),

.TEST\_SE(tb\_test\_se),

.TCK(tb\_tck),

.TRSTN(tb\_trstn),

.TDI(tb\_tdi),

.TMS(tb\_tms),

.TDO (tb\_tdo));

initial

begin

tb\_rst=0;

tb\_clk=0;

tb\_go=0;

tb\_x=0;

tb\_y=0;

tb\_test\_se=1;

tb\_trstn=0;

tb\_tdi=0;

tb\_tms=0;

**pattern =5'b01100;**

end

always #100 tb\_clk=~tb\_clk;

assign tb\_tck=tb\_clk;

initial begin

tb\_rst = 1'b0;

#400 tb\_rst = 1'b1;

end

always @(posedge tb\_clk)

pattern[4:0] <={pattern[0],pattern[4:1]};

always @(posedge tb\_clk)

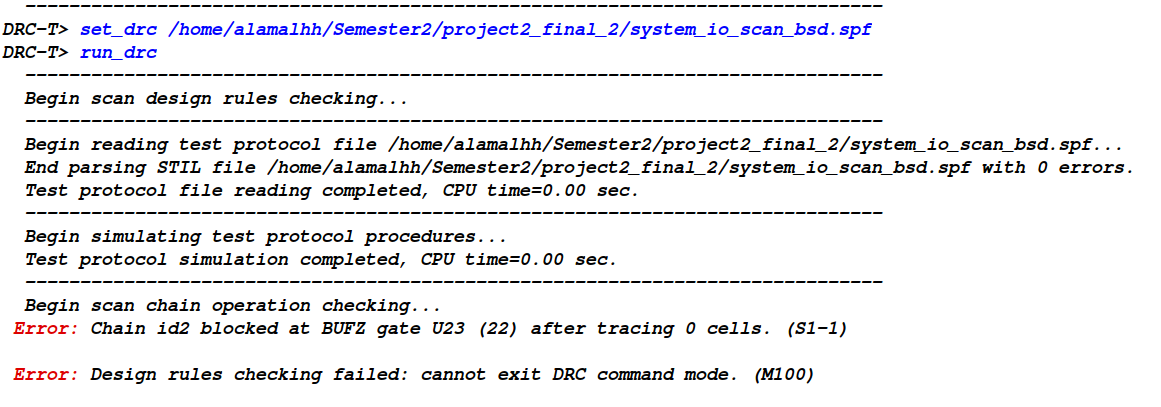
tb\_tdi <= pattern[0];

endmodule

**Important Observations:**

The integration of the internal scan chain is done with the boundary scan chain.And the flush pattern test bench is developed and the edited netlist for the single scan chain is simulated using the flush pattern test bench and the X propagation is seen, this may be an Error in the insertion of the TAP controller or the boundary scan.

**Generation of ATPG Patterns:**



The same Error obtained in the method to generate the ATPG patterns is obtained even here in the ATPG generation using Tetramax tool.