VLSI DESIGN TEST/POWER

PROJECT 1

GCD SYNTHESIS,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 write a testbench for the GCD module and verify the functionality after synthesizing it using it Design Compiler

Solution of the Problem:

The VHDL code is modified by either adding the default statement or initial values and the same is synthesized and the test bench is designed in VHDL to verify the operation functionally.

Help from the tool to solve(Script):

The following is the test bench for the VHDL GCD module.

```
library ieee;
use ieee.std_logic_1164.all;

entity tb_gcd_bsd is
end tb_gcd_bsd;

architecture tb_gcd_bsd_ar of tb_gcd_bsd is

component gcd_bsd is

   port( rst : in std_logic;
            clk : in std_logic;
            go_i : in std_logic;
            x_i : in std_logic_vector (3 downto 0);
            y_i : in std_logic_vector (3 downto 0);
        d_o : out std_logic_vector (3 downto 0));

end component;
constant bit_width: integer:=4;
signal t_x_i : std_logic_vector(bit_width-1 downto 0);
signal t_y_i : std_logic_vector(bit_width-1 downto 0);
signal t_d_o : std_logic_vector(bit_width-1 downto 0);
signal t_rst,t_clk,t_go_i : std_logic:='0';
signal counter : integer :=0;
begin
GCD: gcd_bsd
            port map(t_rst,t_clk,t_go_i,
                        t_x_i,t_y_i,
                        t_d_o);
tb_clock_process: process(t_clk)
begin
            t_clk <= not t_clk after 50 ns;
end process tb_clock_process;
tb_reset_process: process
begin
t_rst <= '1'; wait for 100 ns;
t_rst <= '0';
wait;
end process tb_reset_process;
```

```
tb_gcd_bsd_ar_process2: process(t_clk)
begin
        if(rising_edge(t_clk)) then
                if (counter<290) then
                counter<=counter+1;
        elsif (counter=291) then
                counter<=0;
end if;
end if;
end process tb_gcd_bsd_ar_process2;


tb_gcd_bsd_ar_process: process(t_clk)
begin
if(rising_edge(t_clk)) then
        if(t_rst<='0')then
        t_go_i<='1';
        end if;
        if(counter=0) then
        t_x_i<="1111";
        t_y_i<="0101";

        elsif(counter=20) then
        t_x_i<="1110";
        t_y_i<="0111";

        elsif(counter=40) then
        t_x_i<="0011";
        t_y_i<="0101";

        elsif(counter=60) then
        t_x_i<="1001";
        t_y_i<="0101";

        elsif(counter=80) then
        t_x_i<="1001";
        t_y_i<="0101";

        elsif(counter=100) then
        t_x_i<="1111";
        t_y_i<="0001";

        elsif(counter=120) then
        t_x_i<="1100";
        t_y_i<="0010";

        elsif(counter=140) then
        t_x_i<="1000";
        t_y_i<="1000";

        elsif(counter=160) then
        t_x_i<="1001";
        t_y_i<="0110";

        elsif(counter=180) then
        t_x_i<="1001";
```

```
            t_y_i<="0110";

            elsif(counter=200) then
            t_x_i<="0001";
            t_y_i<="0101";

            elsif(counter=220) then
            t_x_i<="0111";
            t_y_i<="0011";

            elsif(counter=240) then
            t_x_i<="1101";
            t_y_i<="0100";

            end if;
            end if;
end process tb_gcd_bsd_ar_process;
end tb_gcd_bsd_ar;
```

extensive test cases for the VHDL code verification after synthesis:

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;

entity tb_gcd_bsd is
end tb_gcd_bsd;

architecture tb_gcd_bsd_ar of tb_gcd_bsd is

component gcd_bsd is

   port( rst : in std_logic;
            clk : in std_logic;
            go_i : in std_logic;
            x_i : in std_logic_vector (3 downto 0);
            y_i : in std_logic_vector (3 downto 0);
          d_o : out std_logic_vector (3 downto 0));

end component;
constant bit_width: integer:=4;
signal t_x_i : std_logic_vector(bit_width-1 downto 0);
signal t_y_i : std_logic_vector(bit_width-1 downto 0);
signal t_d_o : std_logic_vector(bit_width-1 downto 0);
signal t_rst,t_clk,t_go_i : std_logic:='0';
signal counter : integer :=0;
begin
GCD: gcd_bsd
            port map(t_rst,t_clk,t_go_i,
                        t_x_i,t_y_i,
                        t_d_o);
tb_clock_process: process(t_clk)
```

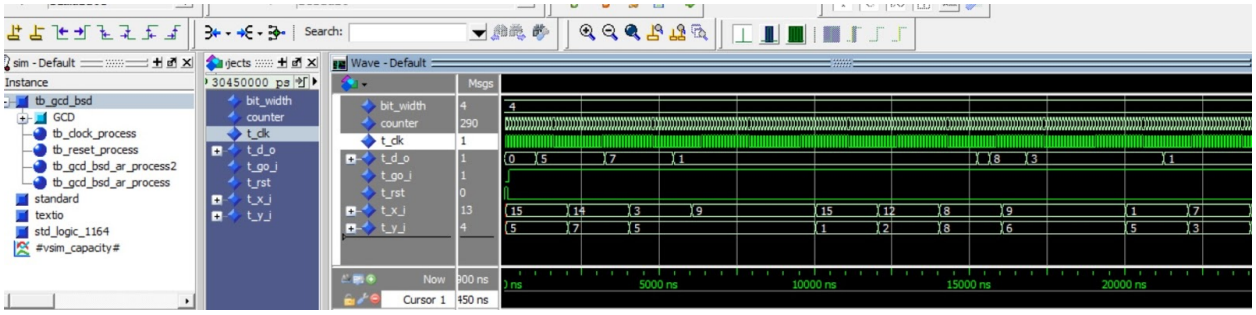
```
begin
	t_clk <= not t_clk after 50 ns;
end process tb_clock_process;
tb_reset_process: process
begin
t_rst <= '1'; wait for 100 ns;
t_rst <= '0';
wait;
end process tb_reset_process;

tb_gcd_bsd_ar_process2: process(t_clk)
begin
	if(rising_edge(t_clk)) then
		if (counter<1000000) then
		counter<=counter+1;
	elsif (counter=1000000000) then
		counter<=0;
end if;
end if;
end process tb_gcd_bsd_ar_process2;


tb_gcd_bsd_ar_process: process
begin
--if(rising_edge(t_clk)) then
	if(t_rst<='0')then
	t_go_i<='1';
	end if;
	for i in 2**(bit_width)-1 downto 1 loop
	for j in 2**(bit_width)-1 downto 1 loop
		t_x_i <= conv_std_logic_vector(i,bit_width);
		t_y_i <= conv_std_logic_vector(j,bit_width);
		wait for 5000 ns;
	end loop;
	end loop;
	--end if;
end process tb_gcd_bsd_ar_process;
end tb_gcd_bsd_ar;
```

Results reported by the tool:

GCD waveform capture for sample inputs:



GCD waveform capture for the all the possible inputs :

Important Observations:

The VHDL simulation is run on the modified and synthesized code and the results have been checked for the proper GCD output.The results are as expected.

Task2:
1.Objective of the task:

The objective is to synthesize the given GCD module after replacing the latches with the Flip-flops in the vhdl design using the library saed90nm_typ_ht_verilog.zip,`saed90nm.sdb and saed90nm_typ_ht.db`

2.Solution of the Problem:

The Synopsys Design Compiler can be used to synthesize the modified and functionally verified VHDL code.

The GCD VHDL code given has some latches inferred in the synthesis because of not defining an edefault case in the case statement.So to avoid the latches inference the following code is added before the case statement.

We can fix this by adding the default statement in the case or by defining an initial value outside as shown below.The latches inference is avoided and it confirmed using the simulator tool

```
process( proceed, comparison, cState )
   begin
                        enable <= '0';
                        xsel <= '0';
                        ysel <= '0';
                        xld <= '0';
                        yld <= '0';
                        nState <= s0;
        case cState is

        when init =>    if( proceed = '0' ) then
                           nState <= init;
                        else
                           nState <= s0;
                        end if;
```

This can also be done by initializing all the variables with some value in the default statement.

3.Help from the tool to solve(Script):
Synthesis constraints (some fucntions are done using Design Vision):

```
###################################
#Library setup
###################################
set link_library [/home/alamalhh/Semester2/testnpower/db/saed90nm_typ_ht(1).db]
set target library [/home/alamalhh/Semester2/testnpower/db/saed90nm_typ_ht(1).db]
set symbol_library [/home/alamalhh/Semester2/testnpower/db/saed90nm(1).sdb]
```

```
read_file -format vhdl src/GCD_Calculator.vhd

current_design gcd_bsd


###################################
#link to standard cell library and uniquify
###################################
link
uniquify

###################################
# Create clock and other constraints
###################################
create_clock clk -period 100 -waveform {0 50} -name clk
set_clock_latency 0.3 clk


###################################
# Input and output delays wrt to clk
###################################
set_input_delay 2.0 -clock clk [all_inputs]
set_output_delay 1.65 -clock clk [all_outputs]
set_load 0.1 [all_outputs]
set_max_fanout 1 [all_inputs]
set_fanout_load 8 [all_outputs]
report_port

###################################
# Set to maximize the area
###################################
set_max_area 0

###################################
#compile design
###################################
compile_ultra
check_design
```

Note: The script is used after the tool is invoked and there are some commands executed by the tool GUI itself.

4.Results reported by the tool:

1. **Report Design :**

```
****************************************
Report : design
Design : gcd_bsd
Version: H-2013.03-SP2
```

Date : Sun Feb 14 00:54:29 2016
****

Design allows ideal nets on clock nets.

Library(s) Used:

saed90nm_typ_ht (File: /home/alamalhh/Semester2/testnpower/db/saed90nm_typ_ht(1).db)

Local Link Library:

{/home/alamalhh/Semester2/testnpower/db/saed90nm_typ_ht(1).db}

Flip-Flop Types:

No flip-flop types specified.

Latch Types:

No latch types specified.

Operating Conditions:

Operating Condition Name : TYPICAL
Library : saed90nm_typ_ht
Process : 1.00
Temperature : 125.00
Voltage : 1.20
Interconnect Model : balanced_tree

Wire Loading Model:

Selected automatically from the total cell area.

Name : 8000
Location : saed90nm_typ_ht
Resistance : 0.0015727
Capacitance : 0.000312
Area : 0.01
Slope : 90.6464

Fanout	Length	Points	Average Cap	Std Deviation
1	13.94			
2	31.80			
3	51.61			
4	73.61			
5	98.05			

6 125.17
7 155.23
8 188.46
9 225.12
10 265.45
11 309.71
12 358.13
13 410.96
14 468.46
15 530.86
16 598.42
17 671.38
18 749.98
19 834.49
20 925.13

Wire Loading Model Mode: enclosed.

Timing Ranges:

No timing ranges specified.

Pin Input Delays:

None specified.

Pin Output Delays:

None specified.

Disabled Timing Arcs:

No arcs disabled.

Required Licenses:

None Required

Design Parameters:

None specified.

2.Report_area:

****

Report : area

Design : gcd_bsd
Version: H-2013.03-SP2
Date : Sun Feb 14 00:57:04 2016
****

Library(s) Used:

saed90nm_typ_ht (File: /home/alamalhh/Semester2/testnpower/db/saed90nm_typ_ht(1).db)

Number of ports: 15
Number of nets: 15
Number of cells: 1
Number of combinational cells: 0
Number of sequential cells: 0
Number of macros/black boxes: 0
Number of buf/inv: 0
Number of references: 1

Combinational area: 999.936006
Buf/Inv area: 154.828804
Noncombinational area: 483.840008
Macro/Black Box area: 0.000000
Net Interconnect area: 56.845148

Total cell area: 1483.776013
Total area: 1540.621162

3.Report cell :

****
Report : cell
Design : gcd_bsd
Version: H-2013.03-SP2
Date : Sun Feb 14 00:57:57 2016
****

Attributes:
b - black box (unknown)
h - hierarchical
n - noncombinational
r - removable
u - contains unmapped logic

Cell	Reference	Library	Area	Attributes
M_GCD	gcd		1483.776013	h, n

Total 1 cells 1483.776013

4.Report Power:

Loading db file '/home/alamalhh/Semester2/testnpower/db/saed90nm_typ_ht(1).db'
Information: Propagating switching activity (low effort zero delay simulation). (PWR-6)
Warning: Design has unannotated primary inputs. (PWR-414)
Warning: Design has unannotated sequential cell outputs. (PWR-415)

```
****************************************
Report : power
        -analysis_effort low
Design : gcd_bsd
Version: H-2013.03-SP2
Date   : Sun Feb 14 00:59:22 2016
****************************************
```

Library(s) Used:

saed90nm_typ_ht (File: /home/alamalhh/Semester2/testnpower/db/saed90nm_typ_ht(1).db)

Operating Conditions: TYPICAL Library: saed90nm_typ_ht
Wire Load Model Mode: enclosed

Design	Wire Load Model	Library
gcd_bsd	8000	saed90nm_typ_ht
gcd	8000	saed90nm_typ_ht
fsm	8000	saed90nm_typ_ht
mux_0	ForQA	saed90nm_typ_ht
mux_1	ForQA	saed90nm_typ_ht
regis_0	ForQA	saed90nm_typ_ht
regis_2	ForQA	saed90nm_typ_ht
comparator	ForQA	saed90nm_typ_ht
subtractor	8000	saed90nm_typ_ht
regis_1	ForQA	saed90nm_typ_ht

Global Operating Voltage = 1.2
Power-specific unit information :
Voltage Units = 1V
Capacitance Units = 1.000000ff
Time Units = 1ns
Dynamic Power Units = 1uW (derived from V,C,T units)

Leakage Power Units = 1pW

Cell Internal Power = 990.0998 nW (66%)
Net Switching Power = 515.1822 nW (34%)

Total Dynamic Power = 1.5053 uW (100%)

Cell Leakage Power = 19.6555 uW

Power Group	Internal Power	Switching Power	Leakage Power	Total Power (%)	Attrs
io_pad	0.0000	0.0000	0.0000	0.0000 (0.00%)	
memory	0.0000	0.0000	0.0000	0.0000 (0.00%)	
black_box	0.0000	0.0000	0.0000	0.0000 (0.00%)	
clock_network	0.0000	0.0000	0.0000	0.0000 (0.00%)	
register	0.1304	0.1309	5.7311e+06	5.9925 (28.32%)	
sequential	0.0000	0.0000	0.0000	0.0000 (0.00%)	
combinational	0.8597	0.3842	1.3924e+07	15.1683 (71.68%)	
Total	0.9901 uW	0.5152 uW	1.9655e+07 pW	21.1608 uW	

5.Report Port:

```
****************************************
Report : port
Design : gcd_bsd
Version: H-2013.03-SP2
Date   : Sun Feb 14 01:00:31 2016
****************************************
```

Port	Dir	Pin Load	Wire Load	Max Trans	Max Cap	Connection Class	Attrs
clk	in	0.0000	0.0000	--	--	--	
go_i	in	0.0000	0.0000	--	--	--	
rst	in	0.0000	0.0000	--	--	--	
x_i[0]	in	0.0000	0.0000	--	--	--	
x_i[1]	in	0.0000	0.0000	--	--	--	
x_i[2]	in	0.0000	0.0000	--	--	--	
x_i[3]	in	0.0000	0.0000	--	--	--	
y_i[0]	in	0.0000	0.0000	--	--	--	
y_i[1]	in	0.0000	0.0000	--	--	--	
y_i[2]	in	0.0000	0.0000	--	--	--	
y_i[3]	in	0.0000	0.0000	--	--	--	

```
d_o[0]        out    0.1000   0.0000   --     --     --
d_o[1]        out    0.1000   0.0000   --     --     --
d_o[2]        out    0.1000   0.0000   --     --     --
d_o[3]        out    0.1000   0.0000   --     --     --
```

6.Report_timing:

```
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : gcd_bsd
Version: H-2013.03-SP2
Date   : Sun Feb 14 01:01:38 2016
****************************************

Operating Conditions: TYPICAL   Library: saed90nm_typ_ht
Wire Load Model Mode: enclosed

  Startpoint: rst (input port clocked by clk)
  Endpoint: M_GCD/Y_REG/output_reg[1]
            (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  gcd_bsd            8000                  saed90nm_typ_ht
  gcd                8000                  saed90nm_typ_ht
  subtractor         8000                  saed90nm_typ_ht
  regis_2            ForQA                 saed90nm_typ_ht

  Point                                    Incr       Path
  ---------------------------------------------------------------------------
  clock clk (rise edge)                    0.00       0.00
  clock network delay (ideal)              0.30       0.30
  input external delay                     2.00       2.30 f
  rst (in)                                 0.00       2.30 f
  M_GCD/rst (gcd)                          0.00       2.30 f
  M_GCD/U_COMP/rst (comparator)            0.00       2.30 f
  M_GCD/U_COMP/U3/QN (NOR2X0)              0.22       2.52 r
  M_GCD/U_COMP/output[1] (comparator)      0.00       2.52 r
  M_GCD/X_SUB/cmd[1] (subtractor)          0.00       2.52 r
  M_GCD/X_SUB/U4/ZN (INVX0)                0.16       2.68 f
  M_GCD/X_SUB/U16/QN (NAND2X1)             0.15       2.83 r
  M_GCD/X_SUB/U40/Q (XOR2X1)               0.21       3.04 f
  M_GCD/X_SUB/U19/QN (NOR2X0)              0.10       3.14 r
  M_GCD/X_SUB/yout[1] (subtractor)         0.00       3.14 r
```

```
M_GCD/Y_MUX/result[1] (mux_1)                          0.00       3.14 r
M_GCD/Y_MUX/U7/Q (AO22X1)                              0.17       3.31 r
M_GCD/Y_MUX/output[1] (mux_1)                          0.00       3.31 r
M_GCD/Y_REG/input[1] (regis_2)                         0.00       3.31 r
M_GCD/Y_REG/U7/Q (AO22X1)                              0.14       3.45 r
M_GCD/Y_REG/output_reg[1]/D (DFFARX1)                  0.03       3.48 r
data arrival time                                                 3.48

clock clk (rise edge)                                100.00     100.00
clock network delay (ideal)                            0.30     100.30
M_GCD/Y_REG/output_reg[1]/CLK (DFFARX1)                0.00     100.30 r
library setup time                                    -0.17     100.13
data required time                                              100.13
--------------------------------------------------------------------------
data required time                                              100.13
data arrival time                                               -3.48
--------------------------------------------------------------------------
slack (MET)                                                     96.65
```

Important Observations:

The combination ,sequential and the net wiring are numbers are observed in the synthesis results.Also the timing is verified at 10MHz.There is no negative slack observed and timing is met.

Task3:

Objective of the task:

To insert DFT logic into the synthesized VHDL code for DFT

Solution of the Problem:

The Synopsys Design Compiler can be used to insert the DFT logic into the synthesized RTL when the tool is given with additional scan configuration and scan insertion commands.

Help from the tool to solve(Script):

The following is the script used for the scan insertion and DFT generation for the GCD VHDL code.

```
#1 Setup variables for scan chain insertion
set dft_runname scan ; # name appended to output files
set scan_library [list /home/alamalhh/Semester2/testnpower/db/saed90nm_typ_ht(1).db] ; #
Library with scan chain cells
set scancell SDFFARX1 ; # Name of ScanFF Cell
# Setup timing variables for dft_drc command
set test_default_delay 0 ; # define time when values are applied to input ports
set test_default_bidir_delay 0 ; # Defines the default switching time of bidirectional
 # ports in a tester cycle.
set test_default_strobe 40 ; # default strobe time in a test cycle for output ports
 # and bidirectional ports in output mode
set test_default_period 100 ; # Defines the default length of a test vector cycle
# Setup scan chain for insert_dft
set test_default_scan_style multiplexed_flip_flop;
# Defines the default scan style for the insert_dft command.


#2.Insert test structures in the design
# Update filebase
set basename gcd_bsd
set filebase [format "%s%s" [format "%s%s" $basename "_"] $dft_runname]
# Update target library
# Set the scan cells to use in the design
#set_scan_register_type -type {DFFPOSX1_SCAN} ;
set_scan_register_type -type ${scancell} ;
# Make sure to add a test_out port
set_scan_configuration -create_dedicated_scan_out_ports true
# Infer clock and reset lines
create_test_protocol -infer_async -infer_clock
dft_drc -verbose
# Replace flip flops with multiplexed flipflops
compile -scan
```

```
# Check for constraint violations
report_constraint -all_violators

#3.Build a scan chain
# connects all scan-enabled ff's together into scan-chain
# note, it creates two new ports: test_si & test_se
insert_dft
# set drive strength of the test ports to 2 (so it isn't assumed to be infinite)
set_drive 2 test_si
set_drive 2 test_se
# since you've already inserted scan-ff's, we don't want that to happen again,
# when we run insert_dft
set_scan_configuration -replace false
# run insert_scan again to set drive-strength constraints
insert_dft
# report any constraints that may have been violated by inserting the test
# structures
report_constraint -all_violators
dft_drc -verbose -coverage_estimate
report_scan_path -view existing -chain all
report_cell


#4.Save reports,test protocol amd DFT verilog design
 report dft_drc
set filename [format "%s%s%s" ./reports/ $filebase ".violators"]
redirect $filename { report_constraint -all_violators }
# report dft_drc
set filename [format "%s%s%s" ./reports/ $filebase ".dft_drc"]
redirect $filename { dft_drc -verbose -coverage_estimate }
# report scan path
set filename [format "%s%s%s" ./reports/ $filebase ".scan_path"]
redirect $filename { report_scan_path -view existing -chain all }
# report cells
set filename [format "%s%s%s" ./reports/ $filebase ".cell"]
redirect $filename { report_cell }
# Write out protocol
set filename [format "%s%s%s" ./src/ $filebase ".spf"]
write_test_protocol -output $filename
# Write out scan chain design
set filename [format "%s%s%s" ./src/ $filebase ".v"]
redirect change_names { change_names -rules verilog -hierarchy -verbose }
write -format verilog -hierarchy -output $filename
```

Results reported by the tool:

The

Important Observations:

Accepted scan configuration for modes: all_dft
In mode: all_dft...

Information: Starting test protocol creation. (TEST-219)
  ...inferring clock signals...
Information: Inferred system/test clock port clk (45.0,55.0). (TEST-260)
  ...inferring asynchronous signals...
Information: Inferred active high asynchronous control port rst. (TEST-261)
In mode: all_dft...
  Pre-DFT DRC enabled

Information: Starting test design rule checking. (TEST-222)
  Loading test protocol
  ...basic checks...
  ...basic sequential cell checks...
          ...checking for scan equivalents...
  ...checking vector rules...
  ...checking pre-dft rules...
  ...saving simulation value info...

Begin Pre-DFT violations...

 Warning: Clock rst connects to data input (D) of DFF M_GCD/TOFSM/cState_reg[0]. (D10-1)
 Warning: Clock rst connects to data input (D) of DFF M_GCD/TOFSM/cState_reg[2]. (D10-2)
 Warning: Clock rst connects to data input (D) of DFF M_GCD/TOFSM/cState_reg[1]. (D10-3)
 Warning: Clock rst connects to data input (D) of DFF M_GCD/X_REG/output_reg[3]. (D10-4)
 Warning: Clock rst connects to data input (D) of DFF M_GCD/X_REG/output_reg[2]. (D10-5)
 Warning: Clock rst connects to data input (D) of DFF M_GCD/X_REG/output_reg[1]. (D10-6)
 Warning: Clock rst connects to data input (D) of DFF M_GCD/X_REG/output_reg[0]. (D10-7)
 Warning: Clock rst connects to data input (D) of DFF M_GCD/Y_REG/output_reg[3]. (D10-8)
 Warning: Clock rst connects to data input (D) of DFF M_GCD/Y_REG/output_reg[2]. (D10-9)
 Warning: Clock rst connects to data input (D) of DFF M_GCD/Y_REG/output_reg[1]. (D10-10)
 Warning: Clock rst connects to data input (D) of DFF M_GCD/Y_REG/output_reg[0]. (D10-11)
 Warning: Clock rst connects to data input (D) of DFF M_GCD/OUT_REG/output_reg[3].
(D10-12)
 Warning: Clock rst connects to data input (D) of DFF M_GCD/OUT_REG/output_reg[2].
(D10-13)
 Warning: Clock rst connects to data input (D) of DFF M_GCD/OUT_REG/output_reg[1].
(D10-14)
 Warning: Clock rst connects to data input (D) of DFF M_GCD/OUT_REG/output_reg[0].
(D10-15)

Pre-DFT violations completed...

```
---------------------------------------------------------------
  DRC Report

  Total violations: 15

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

15 PRE-DFT VIOLATIONS
    15 Clock feeding data input violations (D10)


Warning: Violations occurred during test design rule checking. (TEST-124)

---------------------------------------------------------------
  Sequential Cell Report

  0 out of 15 sequential cells have violations

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

SEQUENTIAL CELLS WITHOUT VIOLATIONS
      *  15 cells are valid scan cells
         M_GCD/TOFSM/cState_reg[0]
         M_GCD/TOFSM/cState_reg[2]
         M_GCD/TOFSM/cState_reg[1]
         M_GCD/X_REG/output_reg[3]
         M_GCD/X_REG/output_reg[2]
         M_GCD/X_REG/output_reg[1]
         M_GCD/X_REG/output_reg[0]
         M_GCD/Y_REG/output_reg[3]
         M_GCD/Y_REG/output_reg[2]
         M_GCD/Y_REG/output_reg[1]
         M_GCD/Y_REG/output_reg[0]
         M_GCD/OUT_REG/output_reg[3]
         M_GCD/OUT_REG/output_reg[2]
         M_GCD/OUT_REG/output_reg[1]
         M_GCD/OUT_REG/output_reg[0]

Information: Test design rule checking completed. (TEST-123)
Current design is 'gcd_bsd'.
Current design is 'gcd_bsd'.
Information: Choosing a test methodology will restrict the optimization of sequential cells.
(UIO-12)


  Beginning Pass 1 Mapping
  ------------------------
```

```
  Processing 'regis_1'
  Processing 'subtractor'
  Processing 'comparator'
  Processing 'regis_2'
  Processing 'regis_0'
  Processing 'mux_1'
  Processing 'mux_0'
  Processing 'fsm'
  Processing 'gcd_0'
  Processing 'gcd_bsd'

  Updating timing information
Information: Updating design information... (UID-85)
Information: Design 'gcd_bsd' has no optimization constraints set. (OPT-108)

  Beginning Mapping Optimizations  (Medium effort)
  ------------------------------
  Structuring 'regis_1'
  Mapping 'regis_1'
  Structuring 'subtractor'
  Mapping 'subtractor'
  Structuring 'comparator'
  Mapping 'comparator'
  Structuring 'regis_2'
  Mapping 'regis_2'
  Structuring 'regis_0'
  Mapping 'regis_0'
  Structuring 'mux_1'
  Mapping 'mux_1'
  Structuring 'mux_0'
  Mapping 'mux_0'
  Structuring 'fsm'
  Mapping 'fsm'
Warning: The nextstate or clock_on object of the test_cell definition of the library cell
'SDFFSSRX2' has scan pins. (OPT-1209)
Warning: The nextstate or clock_on object of the test_cell definition of the library cell
'SDFFSSRX1' has scan pins. (OPT-1209)

                              TOTAL
   ELAPSED            WORST NEG   SETUP    DESIGN
    TIME      AREA      SLACK     COST    RULE COST         ENDPOINT
  --------- --------- --------- --------- --------- --------------------------
    0:00:00   1586.7      0.00       0.0       0.0
    0:00:01   1586.7      0.00       0.0       0.0
    0:00:01   1586.7      0.00       0.0       0.0
    0:00:01   1586.7      0.00       0.0       0.0
    0:00:01   1586.7      0.00       0.0       0.0
    0:00:01   1586.7      0.00       0.0       0.0
```

```
  0:00:01    1586.7      0.00       0.0       0.0
  0:00:01    1586.7      0.00       0.0       0.0
  0:00:01    1586.7      0.00       0.0       0.0
  0:00:01    1586.7      0.00       0.0       0.0
  0:00:01    1586.7      0.00       0.0       0.0
  0:00:01    1586.7      0.00       0.0       0.0


  Beginning Delay Optimization Phase
  ----------------------------------

                          TOTAL
  ELAPSED            WORST NEG   SETUP    DESIGN
    TIME      AREA      SLACK     COST    RULE COST         ENDPOINT
---------- ---------- --------- --------- --------- --------------------------
  0:00:01    1586.7      0.00       0.0       0.0
  0:00:01    1586.7      0.00       0.0       0.0
  0:00:01    1586.7      0.00       0.0       0.0


  Beginning Area-Recovery Phase  (cleanup)
  -----------------------------

                          TOTAL
  ELAPSED            WORST NEG   SETUP    DESIGN
    TIME      AREA      SLACK     COST    RULE COST         ENDPOINT
---------- ---------- --------- --------- --------- --------------------------
  0:00:01    1586.7      0.00       0.0       0.0
  0:00:01    1586.7      0.00       0.0       0.0
  0:00:01    1575.2      0.00       0.0       0.0
  0:00:01    1569.5      0.00       0.0       0.0
  0:00:01    1569.5      0.00       0.0       0.0
  0:00:01    1569.5      0.00       0.0       0.0
  0:00:01    1569.5      0.00       0.0       0.0
  0:00:01    1569.5      0.00       0.0       0.0
  0:00:01    1569.5      0.00       0.0       0.0
  0:00:01    1569.5      0.00       0.0       0.0
  0:00:01    1569.5      0.00       0.0       0.0
  0:00:01    1569.5      0.00       0.0       0.0
  0:00:01    1569.5      0.00       0.0       0.0
Loading db file '/home/alamalhh/Semester2/testnpower/db/saed90nm_typ_ht(1).db'

  Optimization Complete
  ---------------------
Information: Updating design information... (UID-85)

****************************************
```

```
Report : constraint
        -all_violators
Design : gcd_bsd
Version: H-2013.03-SP2
Date   : Thu Feb 18 23:55:45 2016
****************************************

This design has no violated constraints.


Information: Starting test design rule checking. (TEST-222)
Warning: Violations occurred during test design rule checking. (TEST-124)
Information: Test design rule checking completed. (TEST-123)
  Architecting Scan Chains
  Routing Scan Chains
  Routing Global Signals
  Mapping New Logic
Resetting current test mode

Accepted scan configuration for modes: all_dft

Information: Starting test design rule checking. (TEST-222)
Warning: Violations occurred during test design rule checking. (TEST-124)
Information: Test design rule checking completed. (TEST-123)
  Architecting Scan Chains
  Routing Scan Chains
  Routing Global Signals
  Mapping New Logic
Resetting current test mode


  Beginning Phase 1 Design Rule Fixing  (max_transition)
  -----------------------------------

                          TOTAL
  ELAPSED           WORST NEG   SETUP    DESIGN
   TIME      AREA     SLACK     COST    RULE COST        ENDPOINT
 --------- --------- --------- --------- --------- -------------------------
   0:00:06    1605.9      0.00       0.0       3.8 test_se
   0:00:06    1605.9      0.00       0.0       3.8 test_se


  Beginning Phase 2 Design Rule Fixing  (max_transition)
  -----------------------------------

                          TOTAL
  ELAPSED           WORST NEG   SETUP    DESIGN
   TIME      AREA     SLACK     COST    RULE COST        ENDPOINT
```

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

In mode: Internal_scan...
  Design has scan chains in this mode
  Design is scan routed
  Post-DFT DRC enabled

Information: Starting test design rule checking. (TEST-222)
  Loading test protocol
  ...basic checks...
  ...basic sequential cell checks...
  ...checking vector rules...
  ...checking clock rules...
  ...checking scan chain rules...
  ...checking scan compression rules...
  ...checking X-state rules...
  ...checking tristate rules...
  ...extracting scan details...
  ...saving simulation value info...

Clock violations completed...

  DRC Report

  Total violations: 15

---

15 CLOCK VIOLATIONS
    15 Clock as data different from capture clock for stable cell violations (C26)

Warning: Violations occurred during test design rule checking. (TEST-124)

  Sequential Cell Report

  0 out of 15 sequential cells have violations

SEQUENTIAL CELLS WITHOUT VIOLATIONS

```
    *  15 cells are valid scan cells
       M_GCD/TOFSM/cState_reg[0]
       M_GCD/TOFSM/cState_reg[2]
       M_GCD/TOFSM/cState_reg[1]
       M_GCD/X_REG/output_reg[3]
       M_GCD/X_REG/output_reg[2]
       M_GCD/X_REG/output_reg[1]
       M_GCD/X_REG/output_reg[0]
       M_GCD/Y_REG/output_reg[3]
       M_GCD/Y_REG/output_reg[2]
       M_GCD/Y_REG/output_reg[1]
       M_GCD/Y_REG/output_reg[0]
       M_GCD/OUT_REG/output_reg[3]
       M_GCD/OUT_REG/output_reg[2]
       M_GCD/OUT_REG/output_reg[1]
       M_GCD/OUT_REG/output_reg[0]

....Inferring feed-through connections....
Information: Test design rule checking completed. (TEST-123)
 Running test coverage estimation...
1008 faults were added to fault list.
ATPG performed for stuck fault model using internal pattern source.
-----------------------------------------------------------
#patterns     #faults     #ATPG faults  test      process
stored     detect/active  red/au/abort  coverage  CPU time
---------  -------------  ------------  --------  --------
Begin deterministic ATPG: #uncollapsed_faults=866, abort_limit=10...
0            771     94         1/0/0    90.67%      0.00
0             68      7        16/0/0    99.29%      0.00
0              7      0        16/0/0   100.00%      0.00

          Pattern Summary Report
-----------------------------------------------
#internal patterns                          0
-----------------------------------------------


    Uncollapsed Stuck Fault Summary Report
-----------------------------------------------
fault class                    code   #faults
-----------------------------  ----  ---------
Detected                        DT        988
Possibly detected               PT          0
Undetectable                    UD         20
ATPG untestable                 AU          0
Not detected                    ND          0
-----------------------------------------------
total faults                              1008
```

```
 test coverage                        100.00%
 -------------------------------------------------
  Information: The test coverage above may be inferior
               than the real test coverage with customized
               protocol and test simulation library.

****************************************
Report : Scan path
Design : gcd_bsd
Version: H-2013.03-SP2
Date   : Thu Feb 18 23:55:50 2016
****************************************

========================================
TEST MODE: Internal_scan
VIEW     : Existing DFT
========================================

========================================
AS SPECIFIED BY USER
========================================


========================================
AS BUILT BY insert_dft
========================================

Scan_path    Len   ScanDataIn  ScanDataOut ScanEnable  MasterClock SlaveClock
-----------  ----- ----------- ----------- ----------- ----------- -----------
I 1          15    test_si     test_so     test_se     clk         -


****************************************
Report : cell
Design : gcd_bsd
Version: H-2013.03-SP2
Date   : Thu Feb 18 23:55:50 2016
****************************************

Attributes:
    b - black box (unknown)
    h - hierarchical
    n - noncombinational
    r - removable
    u - contains unmapped logic

Cell                    Reference       Library            Area  Attributes
--------------------------------------------------------------------------------
```

```
M_GCD                  gcd_test_1                    1512.345585
                                                       h, n
U1                     INVX0          saed90nm_typ_ht 5.529600
U2                     INVX0          saed90nm_typ_ht 5.529600
U3                     DELLN2X2       saed90nm_typ_ht 15.667200
--------------------------------------------------------------------------------
Total 4 cells                                         1539.071985
Warning: Can't find object 'dft_drc' in design 'gcd_bsd'. (UID-95)
Writing test protocol file '/home/alamalhh/Semester2/testnpower/src/gcd_bsd_scan.spf' for
mode 'Internal_scan'...
Writing verilog file '/home/alamalhh/Semester2/testnpower/src/gcd_bsd_scan.v'.
```

Important Observations:

The scan chain has been generated and it is confirmed from the reported generated in the DFT insertion phase by the Design Compiler.
The scan chain size is 15 as 15 flip-flops are present. The scan input is test_si and output is test_so, test_se is scan enable and the clk is only used as the scan clock.
There are no DFT violations observed in the DFT insertion flow using the Design Compiler.

Task 4:

Objective of the task:

The objective of the task is to generated the test patterns for the synthesized and scan inserted netlist of the GCD module

Solution of the Problem:

The Synopsys Tetramax tool is used to generate the test patterns and check the coverage of the circuit

Help from the tool to solve(Script):

The required commands are executed in the GUI mode in the Tetramax tool

Results reported by the tool:

```
STIL 1.0 { Design 2005; }
Header {
  Title "  TetraMAX (TM)  H-2013.03-SP2-i130530_190956 STIL output";
  Date "Fri Feb 19 00:34:08 2016";
  Source "Minimal STIL for design `gcd_bsd'";
  History {
    Ann {*  Thu Feb 18 23:55:40 2016  *}
    Ann {*  DFT Compiler H-2013.03-SP2  *}
    Ann {*   Uncollapsed Transition Fault Summary Report *}
    Ann {* ----------------------------------------------- *}
    Ann {* fault class                  code   #faults *}
    Ann {* ------------------------------  ----  --------- *}
    Ann {* Detected                      DT       477 *}
    Ann {* Possibly detected             PT         0 *}
    Ann {* Undetectable                  UD         0 *}
    Ann {* ATPG untestable               AU         9 *}
    Ann {* Not detected                  ND         0 *}
    Ann {* ----------------------------------------------- *}
    Ann {* total faults                          486 *}
    Ann {* test coverage                       98.15% *}
    Ann {* ----------------------------------------------- *}
    Ann {*  *}
    Ann {*           Pattern Summary Report *}
    Ann {* ----------------------------------------------- *}
    Ann {* #internal patterns                     65 *}
    Ann {*     #full_sequential patterns           65 *}
    Ann {* ----------------------------------------------- *}
    Ann {*  *}
    Ann {* rule  severity  #fails  description *}
    Ann {* ----  --------  ------  ------------------------------ *}
    Ann {* N21   warning        1  unsupported UDP entry *}
    Ann {* C26   warning       15  clock as data different from capture clock for stable cell *}
```

```
    Ann {*  *}
    Ann {* clock_name        off  usage *}
    Ann {* ----------------  ---  -------------------------- *}
    Ann {* clk               0   master shift  *}
    Ann {* rst               0   master reset  *}
    Ann {*  *}
    Ann {* There are no constraint ports *}
    Ann {* There are no equivalent pins *}
    Ann {* There are no net connections *}
  }
}
Signals {
  "rst" In; "clk" In; "go_i" In; "x_i[3]" In; "x_i[2]" In; "x_i[1]" In; "x_i[0]" In;
  "y_i[3]" In; "y_i[2]" In; "y_i[1]" In; "y_i[0]" In; "test_si" In { ScanIn; } "test_se" In;
  "d_o[3]" Out; "d_o[2]" Out; "d_o[1]" Out; "d_o[0]" Out; "test_so" Out { ScanOut;
  }
}
SignalGroups {
  "_pi" = '"clk" + "go_i" + "rst" + "x_i[0]" + "x_i[1]" + "x_i[2]" + "x_i[3]" +
  "y_i[0]" + "y_i[1]" + "y_i[2]" + "y_i[3]" + "test_si" + "test_se"'; // #signals=13
  "_in" = '"rst" + "clk" + "go_i" + "x_i[3]" + "x_i[2]" + "x_i[1]" + "x_i[0]" +
  "y_i[3]" + "y_i[2]" + "y_i[1]" + "y_i[0]" + "test_si" + "test_se"'; // #signals=13
  "all_inputs" = '"clk" + "go_i" + "rst" + "x_i[0]" + "x_i[1]" + "x_i[2]" +
  "x_i[3]" + "y_i[0]" + "y_i[1]" + "y_i[2]" + "y_i[3]" + "test_si" + "test_se"';
  // #signals=13
  "_po" = '"d_o[0]" + "d_o[1]" + "d_o[2]" + "d_o[3]" + "test_so"'; // #signals=5
  "_si" = '"test_si"' { ScanIn; } // #signals=1
  "all_outputs" = '"d_o[0]" + "d_o[1]" + "d_o[2]" + "d_o[3]" + "test_so"'; // #signals=5
  "all_ports" = '"all_inputs" + "all_outputs"'; // #signals=18
  "_clk" = '"clk" + "rst"'; // #signals=2
  "_so" = '"test_so"' { ScanOut; } // #signals=1
  "_out" = '"d_o[3]" + "d_o[2]" + "d_o[1]" + "d_o[0]" + "test_so"'; // #signals=5
}
Timing {
  WaveformTable "_allclock_launch_capture_WFT_" {
    Period '100ns';
    Waveforms {
      "all_inputs" { 0 { '0ns' D; } }
      "all_inputs" { 1 { '0ns' U; } }
      "all_inputs" { Z { '0ns' Z; } }
      "all_inputs" { N { '0ns' N; } }
      "all_outputs" { X { '0ns' X; '40ns' X; } }
      "all_outputs" { H { '0ns' X; '40ns' H; } }
      "all_outputs" { L { '0ns' X; '40ns' L; } }
      "all_outputs" { T { '0ns' X; '40ns' T; } }
      "clk" { P { '0ns' D; '45ns' U; '55ns' D; } }
      "rst" { P { '0ns' D; '45ns' U; '55ns' D; } }
    }
```

```
}
WaveformTable "_multiclock_capture_WFT_" {
  Period '100ns';
  Waveforms {
    "all_inputs" { 0 { '0ns' D; } }
    "all_inputs" { 1 { '0ns' U; } }
    "all_inputs" { Z { '0ns' Z; } }
    "all_inputs" { N { '0ns' N; } }
    "all_outputs" { X { '0ns' X; '40ns' X; } }
    "all_outputs" { H { '0ns' X; '40ns' H; } }
    "all_outputs" { L { '0ns' X; '40ns' L; } }
    "all_outputs" { T { '0ns' X; '40ns' T; } }
    "clk" { P { '0ns' D; '45ns' U; '55ns' D; } }
    "rst" { P { '0ns' D; '45ns' U; '55ns' D; } }
  }
}
WaveformTable "_allclock_launch_WFT_" {
  Period '100ns';
  Waveforms {
    "all_inputs" { 0 { '0ns' D; } }
    "all_inputs" { 1 { '0ns' U; } }
    "all_inputs" { Z { '0ns' Z; } }
    "all_inputs" { N { '0ns' N; } }
    "all_outputs" { X { '0ns' X; '40ns' X; } }
    "all_outputs" { H { '0ns' X; '40ns' H; } }
    "all_outputs" { L { '0ns' X; '40ns' L; } }
    "all_outputs" { T { '0ns' X; '40ns' T; } }
    "clk" { P { '0ns' D; '45ns' U; '55ns' D; } }
    "rst" { P { '0ns' D; '45ns' U; '55ns' D; } }
  }
}
WaveformTable "_allclock_capture_WFT_" {
  Period '100ns';
  Waveforms {
    "all_inputs" { 0 { '0ns' D; } }
    "all_inputs" { 1 { '0ns' U; } }
    "all_inputs" { Z { '0ns' Z; } }
    "all_inputs" { N { '0ns' N; } }
    "all_outputs" { X { '0ns' X; '40ns' X; } }
    "all_outputs" { H { '0ns' X; '40ns' H; } }
    "all_outputs" { L { '0ns' X; '40ns' L; } }
    "all_outputs" { T { '0ns' X; '40ns' T; } }
    "clk" { P { '0ns' D; '45ns' U; '55ns' D; } }
    "rst" { P { '0ns' D; '45ns' U; '55ns' D; } }
  }
}
WaveformTable "_default_WFT_" {
  Period '100ns';
```

```
    Waveforms {
      "all_inputs" { 0 { '0ns' D; } }
      "all_inputs" { 1 { '0ns' U; } }
      "all_inputs" { Z { '0ns' Z; } }
      "all_inputs" { N { '0ns' N; } }
      "all_outputs" { X { '0ns' X; '40ns' X; } }
      "all_outputs" { H { '0ns' X; '40ns' H; } }
      "all_outputs" { L { '0ns' X; '40ns' L; } }
      "all_outputs" { T { '0ns' X; '40ns' T; } }
      "clk" { P { '0ns' D; '45ns' U; '55ns' D; } }
      "rst" { P { '0ns' D; '45ns' U; '55ns' D; } }
    }
  }
}
ScanStructures {
  ScanChain "1" {
    ScanLength 15;
    ScanIn "test_si";
    ScanOut "test_so";
    ScanInversion 0;
    ScanCells "gcd_bsd.M_GCD.OUT_REG.output_reg_0_.SI"
"gcd_bsd.M_GCD.OUT_REG.output_reg_1_.SI"
    "gcd_bsd.M_GCD.OUT_REG.output_reg_2_.SI"
"gcd_bsd.M_GCD.OUT_REG.output_reg_3_.SI"
    "gcd_bsd.M_GCD.TOFSM.cState_reg_0_.SI" "gcd_bsd.M_GCD.TOFSM.cState_reg_1_.SI"
    "gcd_bsd.M_GCD.TOFSM.cState_reg_2_.SI" "gcd_bsd.M_GCD.X_REG.output_reg_0_.SI"
    "gcd_bsd.M_GCD.X_REG.output_reg_1_.SI" "gcd_bsd.M_GCD.X_REG.output_reg_2_.SI"
    "gcd_bsd.M_GCD.X_REG.output_reg_3_.SI" "gcd_bsd.M_GCD.Y_REG.output_reg_0_.SI"
    "gcd_bsd.M_GCD.Y_REG.output_reg_1_.SI" "gcd_bsd.M_GCD.Y_REG.output_reg_2_.SI"
    "gcd_bsd.M_GCD.Y_REG.output_reg_3_.SI" ;
    ScanMasterClock "clk" ;
  }
}
PatternBurst "_burst_" {
  PatList { "_pattern_" {
  }
}}
PatternExec {
  PatternBurst "_burst_";
}
Procedures {
  "multiclock_capture" {
    W "_multiclock_capture_WFT_";
    C { "all_inputs"=0N0\r10 N ; "all_outputs"=XXXXX; }
    V { "_pi"=\r13 # ; "_po"=#####; }
  }
  "allclock_capture" {
    W "_allclock_capture_WFT_";
```

```
    C { "all_inputs"=0N0\r10 N ; "all_outputs"=XXXXX; }
    V { "_pi"=\r13 # ; "_po"=#####; }
  }
  "allclock_launch" {
    W "_allclock_launch_WFT_";
    C { "all_inputs"=0N0\r10 N ; "all_outputs"=XXXXX; }
    V { "_pi"=\r13 # ; "_po"=#####; }
  }
  "allclock_launch_capture" {
    W "_allclock_launch_capture_WFT_";
    C { "all_inputs"=0N0\r10 N ; "all_outputs"=XXXXX; }
    V { "_pi"=\r13 # ; "_po"=#####; }
  }
  "load_unload" {
    W "_default_WFT_";
    C { "all_inputs"=0N0\r10 N ; "all_outputs"=XXXXX; }
    "Internal_scan_pre_shift": V { "test_se"=1; }
    Shift {         W "_default_WFT_";
      V { "_clk"=P0; "_si"=#; "_so"=#; }
    }
  }
}
MacroDefs {
  "test_setup" {
    W "_default_WFT_";
    C { "all_inputs"=\r13 N ; "all_outputs"=XXXXX; }
    V { "clk"=0; "rst"=0; }
    V { }
  }
}
Pattern "_pattern_" {
  W "_multiclock_capture_WFT_";
  "precondition all Signals": C { "_pi"=\r13 0 ; "_po"=XXXXX; }
  Macro "test_setup";
  Ann {* full_sequential *}
  "pattern 0": Call "load_unload" {
    "test_si"=111001010101000; }
  Call "allclock_launch" {
    "_pi"=01P0100101101; }
  Call "allclock_capture" {
    "_pi"=P001100011110; }
  Ann {* full_sequential *}
  "pattern 1": Call "load_unload" {
    "test_so"=LLLLLLLLLLLLLL; "test_si"=010000101011011; }
  Call "allclock_launch" {
    "_pi"=P000010101010; }
  Call "allclock_capture" {
    "_pi"=P101010100000; }
```

```
  Ann {* full_sequential *}
  "pattern 2": Call "load_unload" {
    "test_so"=LLHLLLHLHHLHLHH; "test_si"=000110101011110; }
  Call "allclock_launch" {
    "_pi"=P100110111001; }
  Call "allclock_capture" {
    "_pi"=P000110111000; }
  Ann {* full_sequential *}
  "pattern 3": Call "load_unload" {
    "test_so"=LLHHLHLHHLLHHLL; "test_si"=111001101010101; }
  Call "allclock_launch" {
    "_pi"=P100011100101; }
  Call "allclock_capture" {
    "_pi"=P100100100100; }
  Ann {* full_sequential *}
  "pattern 4": Call "load_unload" {
    "test_so"=HLLHLLHLLHHHLHL; "test_si"=101001110101010; }
  Call "multiclock_capture" {
    "_pi"=0100100010100; }
  Call "allclock_launch_capture" {
    "_pi"=P000111000010; }
  Ann {* full_sequential *}
  "pattern 5": Call "load_unload" {
    "test_so"=LLLLHHHLLHHHLHL; "test_si"=110011000110111; }
  Call "allclock_launch" {
    "_pi"=P100110100011; }
  Call "allclock_capture" {
    "_pi"=P001111001000; }
  Ann {* full_sequential *}
  "pattern 6": Call "load_unload" {
```

The final ATPG patterns:

```
"test_si"=111001010101000; }
    "test_so"=LLLLLLLLLLLLLL; "test_si"=010000101011011; }
    "test_so"=LLHLLLHLHHLHLHH; "test_si"=000110101011110; }
    "test_so"=LLHHLHLHHLLHHLL; "test_si"=111001101010101; }
    "test_so"=HLLHLLHLLHHHLHL; "test_si"=101001110101010; }
    "test_so"=LLLLHHHLLHHHLHL; "test_si"=110011000110111; }
    "test_so"=LLLHHLLLLLHHLLL; "test_si"=001110101101000; }
    "test_so"=LLHLLHLHLHHLLLH; "test_si"=110001101110010; }
    "test_so"=HLLLLHLHLLHLHLH; "test_si"=110110011110010; }
    "test_so"=HLLLLLHHLLHLLHH; "test_si"=010011011011001; }
    "test_so"=HLLHHLHHHLLLLHH; "test_si"=001001011100101; }
    "test_so"=LHLLLHHHLHHHLHL; "test_si"=001111010001001; }
    "test_so"=LLHHHHLHLLHHLLH; "test_si"=111110001010101; }
    "test_so"=HHHHLLLHLHLHLHH; "test_si"=011101010100011; }
    "test_so"=HHHLHLHLLHHLHHH; "test_si"=000101011100101; }
```

"test_so"=LLLHLHLLLHLLHLL; "test_si"=100101010011011; }
"test_so"=LLLLLLLLLLHLLLL; "test_si"=110000001010111; }
"test_so"=HHLLLLLLHLHLHHH; "test_si"=111101111110011; }
"test_so"=HHHHLHHHLHLLLHH; "test_si"=110100110101001; }
"test_so"=LHLLLHHLLHHLLHL; "test_si"=100111111010110; }
"test_so"=LLLHHLHLLHHHHLL; "test_si"=110001110101001; }
"test_so"=HLLLHHHLLHHLLHH; "test_si"=111111001110100; }
"test_so"=HHHHHLLHHHLHLLH; "test_si"=101010101011011; }
"test_so"=LHLHLHLHLHHLHHH; "test_si"=001110110101101; }
"test_so"=LHLLHLLLLHHHHLH; "test_si"=100001010101000; }
"test_so"=LHLHHHHHLHHHLLL; "test_si"=110010100101110; }
"test_so"=LHLLHHLLLHHHHHL; "test_si"=001100110100001; }
"test_so"=HLHLLLHLLHHLLLH; "test_si"=101001110110111; }
"test_so"=LHLLHLHLLLHHLHL; "test_si"=010011101110011; }
"test_so"=HLLHLHLLLLHLHLL; "test_si"=101001100010101; }
"test_so"=LLLHLLLHLHHHLHH; "test_si"=110000010101001; }
"test_so"=LLHLLHLLLHHHLLH; "test_si"=100101001110001; }
"test_so"=LLHLLHHHLLHLHHH; "test_si"=101000001000101; }
"test_so"=HLHLLLLLHLHLHLH; "test_si"=011100100111111; }
"test_so"=HHHLLHLLLLHHHHH; "test_si"=100110011001010; }
"test_so"=HLLHHLLHHHLHLHL; "test_si"=000000000111100; }
"test_so"=LLLLLLLLLLHHLLL; "test_si"=110000111101110; }
"test_so"=HLLHLLHHLHLLLHH; "test_si"=011111010101100; }
"test_so"=LHHHHHHHHLLHHLL; "test_si"=010101111000111; }
"test_so"=LHLHLLHLHLHLHHH; "test_si"=010000110101100; }
"test_so"=HLLHHLLHHHLHHLL; "test_si"=010010111110010; }
"test_so"=LLHLLHHHLLHLHHH; "test_si"=100000110100001; }
"test_so"=HLLHHLLLHLHLLLH; "test_si"=010011010110001; }
"test_so"=HLLHLLLHLLHLLLH; "test_si"=010011001110000; }
"test_so"=HLLHHLLHLLHHLLH; "test_si"=010111011110001; }
"test_so"=HLHHHLHHLLHHLHH; "test_si"=011011101011011; }
"test_so"=HHLHHHLHHHLLHHH; "test_si"=010111001110110; }
"test_so"=LLHLHLLHLLHHLLH; "test_si"=010101010011011; }
"test_so"=HLHLHLHLHHLLHHH; "test_si"=101010111110001; }
"test_so"=LHLHLLHLLLLHLHL; "test_si"=101010011110010; }
"test_so"=LLHLLLHHLLHLLHH; "test_si"=101101010101011; }
"test_so"=HLHHHLLHHLHHLHH; "test_si"=101001000100000; }
"test_so"=HLLHHLLHHHLLLLL; "test_si"=010101011011011; }
"test_so"=HLHLHLHHHLLLHHL; "test_si"=111101101011011; }
"test_so"=HHHLHHLHHLHLHHL; "test_si"=111001101011110; }
"test_so"=HHLLHHLHHLLHHLH; "test_si"=110101001011100; }
"test_so"=HLHLHLLHHLHHLLH; "test_si"=111110000101100; }
"test_so"=HLLLHLLHHLLHHLL; "test_si"=111001011110011; }
"test_so"=LLLHHLHHLLHHLHH; "test_si"=111010100100001; }
"test_so"=HHLHLHLLLHHLLHL; "test_si"=011110000000100; }
"test_so"=HHHHLLLLLLLHLLH; "test_si"=100000100111000; }
"test_so"=LLLLLLLLLLLLLLL; "test_si"=010011001001011; }
"test_so"=HLLHHLLHLLHLHHH; "test_si"=100011110000001; }

"test_so"=LLLHHHHLLLLLLHH; "test_si"=100110011011010; }
"test_so"=LLLLLLLLLLLLLL; "test_si"=011101101001011; }

Important Observations:

The total faults possible reported by the ATPG tool are 486 of which 9 are ATPG untestable.So the ATPG test coverage is 98.15%.
There are 65 patterns generated by the ATPG tool