**VLSI DESIGN TEST/POWER**

**PROJECT 4**

**LOW ANALYSIS USING POWER COMPILER**

**Team members:**

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

**Task 1:**

**Objective of the task:**

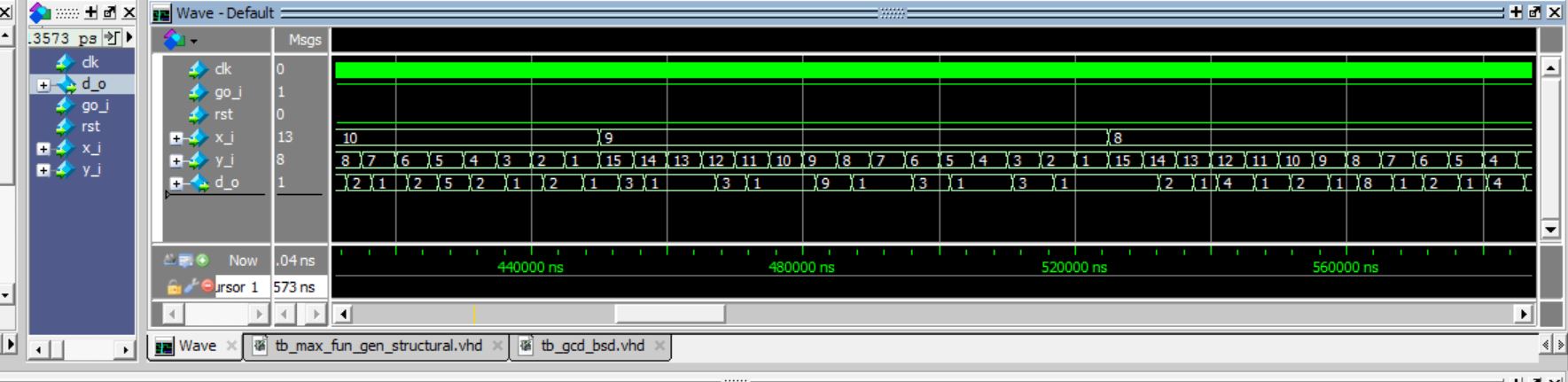
The objective is to write a test bench for the GCD module and verify the functionality and then to generate 15 different vectors to apply at least 15 test patterns (5 test patterns with small GCD for two numbers m and n, 5 with large GCD, 5 with 1 as the GCD) to the machine to compute GCDs.

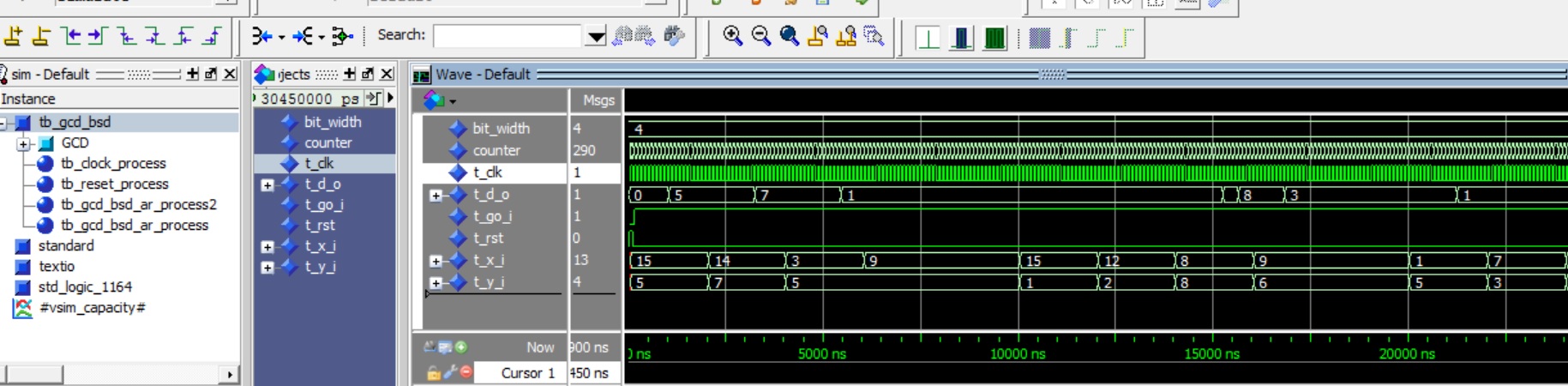
**Solution of the Problem:**

The test bench is written in VHDL with the specific patterns to check the correctness of the function and the different patterns are designed for the specifications.

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

**Results reported by the tool:**





**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.

**Tb code :**

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

use std.env.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 (9 downto 0);

y\_i : in std\_logic\_vector (9 downto 0);

d\_o : out std\_logic\_vector (9 downto 0));

end component;

constant bit\_width: integer:=10;

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 5 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<300000) then

counter<=counter+1;

elsif (counter=2999) 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<=conv\_std\_logic\_vector(1000,bit\_width);

t\_y\_i<=conv\_std\_logic\_vector(100,bit\_width);

elsif(counter=1100) then

t\_x\_i<=conv\_std\_logic\_vector(1000,bit\_width);

t\_y\_i<=conv\_std\_logic\_vector(998,bit\_width);

--Input values for Low value of GCD output

elsif(counter=2200) then

t\_x\_i<=conv\_std\_logic\_vector(587,bit\_width);

t\_y\_i<=conv\_std\_logic\_vector(947,bit\_width);

elsif(counter=3300) then

t\_x\_i<=conv\_std\_logic\_vector(751,bit\_width);

t\_y\_i<=conv\_std\_logic\_vector(887,bit\_width);

elsif(counter=4400) then

t\_x\_i<=conv\_std\_logic\_vector(859,bit\_width);

t\_y\_i<=conv\_std\_logic\_vector(997,bit\_width);

elsif(counter=5500) then

t\_x\_i<=conv\_std\_logic\_vector(929,bit\_width);

t\_y\_i<=conv\_std\_logic\_vector(937,bit\_width);

elsif(counter=6600) then

t\_x\_i<=conv\_std\_logic\_vector(991,bit\_width);

t\_y\_i<=conv\_std\_logic\_vector(997,bit\_width);

-- Inputs for high value GCD output

elsif(counter=7700) then

t\_x\_i<=conv\_std\_logic\_vector(1000,bit\_width);

t\_y\_i<=conv\_std\_logic\_vector(998,bit\_width);

elsif(counter=8800) then

t\_x\_i<=conv\_std\_logic\_vector(3,bit\_width);

t\_y\_i<=conv\_std\_logic\_vector(999,bit\_width);

elsif(counter=9900) then

t\_x\_i<=conv\_std\_logic\_vector(842,bit\_width);

t\_y\_i<=conv\_std\_logic\_vector(1000,bit\_width);

elsif(counter=11000) then

t\_x\_i<=conv\_std\_logic\_vector(567,bit\_width);

t\_y\_i<=conv\_std\_logic\_vector(666,bit\_width);

elsif(counter=12100) then

t\_x\_i<=conv\_std\_logic\_vector(998,bit\_width);

t\_y\_i<=conv\_std\_logic\_vector(332,bit\_width);

elsif(counter=13200) then

t\_x\_i<=conv\_std\_logic\_vector(840,bit\_width);

t\_y\_i<=conv\_std\_logic\_vector(4,bit\_width);

-- Inputs for the medium value GCD outputs

elsif(counter=14300) then

t\_x\_i<=conv\_std\_logic\_vector(400,bit\_width);

t\_y\_i<=conv\_std\_logic\_vector(800,bit\_width);

elsif(counter=15400) then

t\_x\_i<=conv\_std\_logic\_vector(1000,bit\_width);

t\_y\_i<=conv\_std\_logic\_vector(500,bit\_width);

elsif(counter=16500) then

t\_x\_i<=conv\_std\_logic\_vector(256,bit\_width);

t\_y\_i<=conv\_std\_logic\_vector(512,bit\_width);

elsif(counter=17600) then

t\_x\_i<=conv\_std\_logic\_vector(1023,bit\_width);

t\_y\_i<=conv\_std\_logic\_vector(1023,bit\_width);

elsif(counter=18700) then

t\_x\_i<=conv\_std\_logic\_vector(840,bit\_width);

t\_y\_i<=conv\_std\_logic\_vector(420,bit\_width);

elsif(counter=18720) then

stop(0);

end if;

end if;

end process tb\_gcd\_bsd\_ar\_process;

end tb\_gcd\_bsd\_ar;

**Task2:**

**Objective of the task:**

Use Model Sim to obtain the RTL switching information in VCD (value change dump) format by simulating the test bench developed in step 1 on the GCD calculator. Translate the VCD file to a SAIF (switching activity information format) file understood by Synopsys using the Synopsys-provided vcd2saif utility.

**Solution of the Problem:**

The waveform produced by the tool is saved in the wlf format which is the waveform log format by the modelsim and then the wlf file is converted to VCD( IEEE format for the value change dumps) by using the command wlf2vcd in the modelsim.

Then the VCD file then converted to the SAIF file which has the switching information for the particular patterns ,.This is done by using vcd2saif command

**Help from the tool to solve:**

Both the wlf2vcd and the vcd2saif are the inbuilt commands by the modelsim and the synopsys VCS tools respectively and they are used for the conversion

**Results reported by the tool:**

**The sample VCD file is pasted here :**

$date

Sat Apr 16 18:47:33 2016

$end

$version

ModelSim Version 10.3d

$end

$timescale

1ps

$end

$scope module tb\_gcd\_bsd $end

$var integer 32 ! bit\_width $end

$var integer 32 " counter $end

$var wire 1 # t\_clk $end

$var wire 1 $ t\_d\_o [9] $end

$var wire 1 % t\_d\_o [8] $end

$var wire 1 & t\_d\_o [7] $end

$var wire 1 ' t\_d\_o [6] $end

$var wire 1 ( t\_d\_o [5] $end

$var wire 1 ) t\_d\_o [4] $end

$var wire 1 \* t\_d\_o [3] $end

$var wire 1 + t\_d\_o [2] $end

$var wire 1 , t\_d\_o [1] $end

$var wire 1 - t\_d\_o [0] $end

$var wire 1 . t\_go\_i $end

$var wire 1 / t\_rst $end

**The sample SAIF file is pasted here:**

(SAIFILE

(SAIFVERSION "2.0")

(DIRECTION "backward")

(DESIGN )

(DATE "Sat Apr 16 21:14:29 2016")

(VENDOR "Synopsys, Inc")

(PROGRAM\_NAME "vcd2saif")

(VERSION "H-2013.03-SP2")

(DIVIDER / )

(TIMESCALE 1 ps)

(DURATION 187205000)

(INSTANCE tb\_gcd\_bsd

(NET

(bit\_width\[0\]

(T0 0) (T1 0) (TX 187205000)

(TC 0) (IG 0)

)

(bit\_width\[10\]

(T0 0) (T1 0) (TX 187205000)

(TC 0) (IG 0)

)

(bit\_width\[11\]

(T0 0) (T1 0) (TX 187205000)

(TC 0) (IG 0)

)

(bit\_width\[12\]

(T0 0) (T1 0) (TX 187205000)

(TC 0) (IG 0)

)

(bit\_width\[13\]

(T0 0) (T1 0) (TX 187205000)

(TC 0) (IG 0)

)

(bit\_width\[14\]

(T0 0) (T1 0) (TX 187205000)

(TC 0) (IG 0)

)

(bit\_width\[15\]

(T0 0) (T1 0) (TX 187205000)

(TC 0) (IG 0)

)

(bit\_width\[16\]

(T0 0) (T1 0) (TX 187205000)

(TC 0) (IG 0)

)

(bit\_width\[17\]

(T0 0) (T1 0) (TX 187205000)

(TC 0) (IG 0)

)

(bit\_width\[18\]

(T0 0) (T1 0) (TX 187205000)

(TC 0) (IG 0)

)

(bit\_width\[19\]

(T0 0) (T1 0) (TX 187205000)

(TC 0) (IG 0)

)

(bit\_width\[1\]

(T0 0) (T1 0) (TX 187205000)

(TC 0) (IG 0)

)

(bit\_width\[20\]

(T0 0) (T1 0) (TX 187205000)

(TC 0) (IG 0)

)

(bit\_width\[21\]

(T0 0) (T1 0) (TX 187205000)

(TC 0) (IG 0)

)

(bit\_width\[22\]

(T0 0) (T1 0) (TX 187205000)

(TC 0) (IG 0)

)

(bit\_width\[23\]

(T0 0) (T1 0) (TX 187205000)

(TC 0) (IG 0)

)

(bit\_width\[24\]

(T0 0) (T1 0) (TX 187205000)

(TC 0) (IG 0)

)

**Task3:**

**Objective of the task:**

Use the SAIF to find the power consumption of the RT-level design.

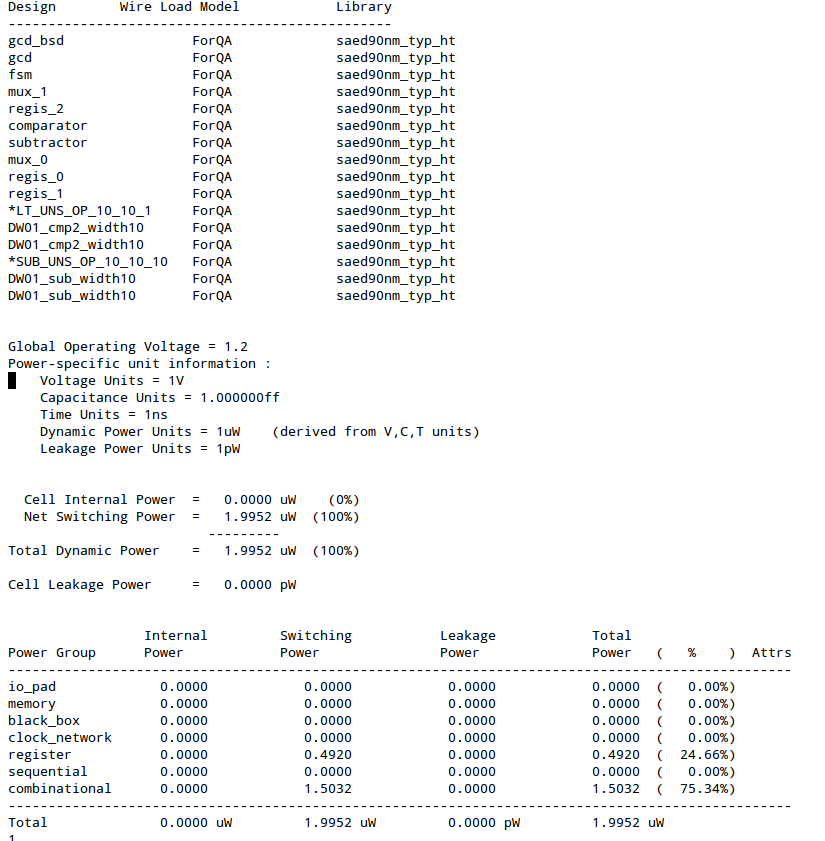
**Solution of the Problem:**

The design compiler and the embedded Power compiler engine needs to be used to measure the power

**Help from the tool to solve:**

The RTL is loaded into the Design Compiler and the Power compiler of the Design Compiler is used to get the conver taken for the switching activity for the particular SAIF file

**Results reported by the tool:**



**Important Observations:**

As we have have not synthesized and the power is captured only by just analyzing and elaborating the VHDL files into the Design compiler, we are not synthesizing wrt library ,so cell dynamic and leakage power is not found, but only the net switching power is obtained from the wire load models

**Task4:**

**Objective of the task:**

Use Design Compiler to synthesize your RTL GCD calculator into gate-level using a 90-nm cell library called saed90nm\_typ.db. Use the *RT-level switching activities* (SAIF file) obtained in Step 2 to estimate the power consumption of the gate level synthesized.

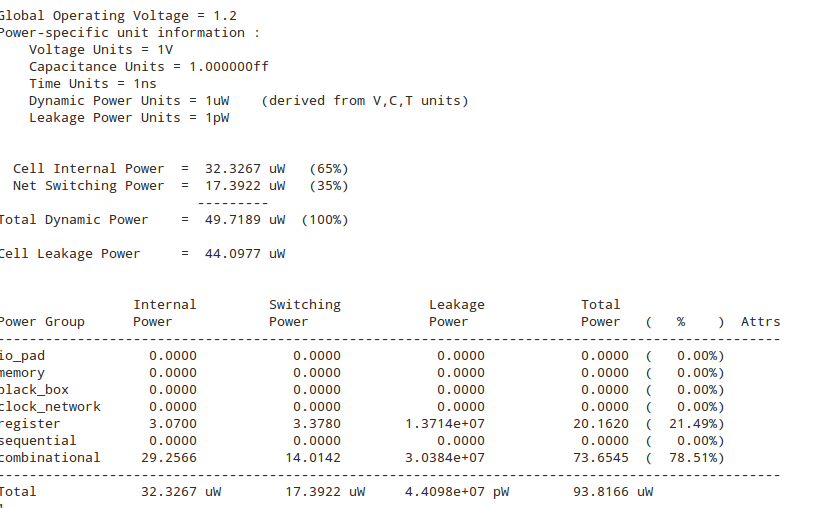
**Solution of the Problem:**

The tool is used and the power is calculated using the RTL level SAIF and the synthesized circuit

**Help from the tool to solve:**

The power compiler is used to measure the power by passing the SAIF to it and using the command report\_power

**Results reported by the tool:**



**Task5:**

**Objective of the task:**

Repeat Step 2 using the gate-level GCD calculator obtained in Step 4, the testbench developed in Step 1, and the cell library to get the gate-level switching activities represented by SAIF.

**Solution of the Problem:**

The synthesized GCD is passed again to the Modelsim tool and the gate level SAIF is obtained for the same simulation patterns given for the RTL simulation.

**Help from the tool to solve:**

The command wlf2vcd and the vcd2saif are used to get the required SAIF file.

The design compiler is used with the appropriate synthesis command to synthesize the circuit.

**Synthesis script used for the synthesis:**

#################################

#Library setup

#################################

set search\_path [list /home/alamalhh/Semester2/test\_n\_power/project4\_new/db /home/alamalhh/Semester2/test\_n\_power/project4\_new]

set link\_library [list /home/alamalhh/Semester2/test\_n\_power/project4\_new/db/saed90nm\_typ\_ht(1).db ]

set target\_library [list /home/alamalhh/Semester2/test\_n\_power/project4\_new/db/saed90nm\_typ\_ht(1).db \

]

set symbol\_library [list /home/alamalhh/Semester2/test\_n\_power/project4\_new/db/saed90nm(1).sdb]

analyze -format vhdl GCD\_Calculator.vhd

elaborate gcd\_bsd

current\_design gcd\_bsd

##################################

##link to standard cell library and uniquify

##################################

link

uniquify

#################################

# Create clock and other constraints

#################################

create\_clock clk -period 10 -waveform {0 5} -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

#################################

check\_design

compile\_ultra -no\_autoungroup

set basename gcd\_bsd

set runname power\_gating

set filebase [format "%s%s" [format "%s%s" $basename "\_"] $runname]

set filename [format "%s%s%s" ./reports\_new/ $filebase ".v"]

redirect change\_names { change\_names -rules verilog -hierarchy -verbose }

write -format verilog -pg -hierarchy -output $filename

read\_saif -input /home/alamalhh/Semester2/test\_n\_power/project4\_new/new\_gcd/gate\_level\_april\_16\_2.saif -instance tb\_gcd\_bsd/GCD -verbose

set filename [format "%s%s%s" ./reports\_new/ $filebase ".cell"]

redirect $filename { report\_cell }

set filename [format "%s%s%s" ./reports\_new/ $filebase ".port"]

redirect $filename { report\_port }

set filename [format "%s%s%s" ./reports\_new/ $filebase ".design"]

redirect $filename { report\_design }

set filename [format "%s%s%s" ./reports\_new/ $filebase ".timing"]

redirect $filename { report\_timing }

set filename [format "%s%s%s" ./reports\_new/ $filebase ".area"]

redirect $filename { report\_area }

set filename [format "%s%s%s" ./reports\_new/ $filebase ".power"]

redirect $filename { report\_power }

**Results reported by the tool:**

**Sample SAIF file obtained:**

(SAIFILE

(SAIFVERSION "2.0")

(DIRECTION "backward")

(DESIGN )

(DATE "Sat Apr 16 21:14:29 2016")

(VENDOR "Synopsys, Inc")

(PROGRAM\_NAME "vcd2saif")

(VERSION "H-2013.03-SP2")

(DIVIDER / )

(TIMESCALE 1 ps)

(DURATION 187205000)

(INSTANCE tb\_gcd\_bsd

(NET

(bit\_width\[0\]

(T0 0) (T1 0) (TX 187205000)

(TC 0) (IG 0)

)

(bit\_width\[10\]

(T0 0) (T1 0) (TX 187205000)

(TC 0) (IG 0)

)

(bit\_width\[11\]

(T0 0) (T1 0) (TX 187205000)

(TC 0) (IG 0)

)

(bit\_width\[12\]

(T0 0) (T1 0) (TX 187205000)

(TC 0) (IG 0)

)

(bit\_width\[13\]

(T0 0) (T1 0) (TX 187205000)

(TC 0) (IG 0)

)

(bit\_width\[14\]

(T0 0) (T1 0) (TX 187205000)

(TC 0) (IG 0)

)

(bit\_width\[15\]

(T0 0) (T1 0) (TX 187205000)

(TC 0) (IG 0)

)

**Important Observations:**

The SAIF file obtained from the gate level circuit synthesized by design compiler is more accurate as it captures the switching activity the gate level which captures all the register and the combination switching activity.

**Task6:**

**Objective of the task:**

Analyze the power consumption using the gate-level GCD calculator and the gate-level SAIF file.

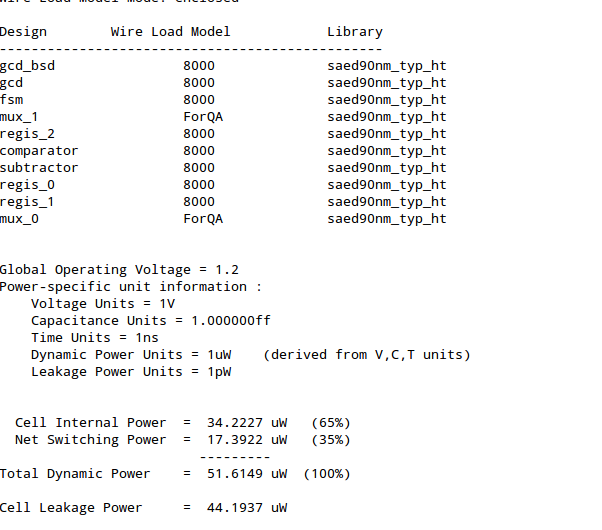
**Solution of the Problem:**

The gate level saif is used and the report\_power command gives the power consumed for the activity.

**Help from the tool to solve:**

Power compiler is used and the report\_command

**Results reported by the tool:**



**Total power consumed is 95.8086uw**

**Task7:**

**Objective of the task:**

Compare the power estimated by Steps 3 (RT SAIF on RT circuit), 4 (RT SAIF on gate-level circuit) and 5 (gate-level SAIF on gate-level circuit).

**Important Observations:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity** | **Dynamic Power** | **Leakage Power** | **Total power** |
| **(a) RTL SAIF RTL LOGIC** | **1.9952** | **0** | **1.9952** |
| **(b) RTL SAIF Gate Level logic** | **49.7189** | **44.0977** | **93.8166** |
| **(c) Gate Level SAIF and Gate Level Logic** | **51.6149** | **44.1937** | **95.8086** |

(a)We have observed that the power at the RTL SAIF with RTL circuit shows the leakage and switching power of cells as zero as we have not given the library to the power tool and does not have any power information .but from the wireload models it was estimating the switching power for the nets alone.

(b) Here we have observed that the power numbers are **nearly accurate** as the SAIF is passed for the power estimation, but as the SAIF is from the RTL switching these are less accurate than the SAIF taken on the gate level circuit.But at this level the power estimation and the process is faster as the SAIF is derived format the RTL logic

(c)The SAIF obtained from the gate level logic is **much more accurate** as it truly represents the exact switching of all the gates and the nets.So the power reported by the gate level SAIF are more accurate.

**Note: The a part can be done by also synthesizing the RTL m but it doesnt make any difference from b case if we synthesize so we have not attempted to synthesize.**

**Phase 2: Logic-level Power Optimization**

**The multi Vt cell are passed to the tool library as :**

**set search\_path [list /home/alamalhh/Semester2/test\_n\_power/project4\_new/db** /home/alamalhh/Semester2/test\_n\_power/project4\_new]

set link\_library [list /home/alamalhh/Semester2/test\_n\_power/project4\_new/db/saed90nm\_typ\_ht(1).db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_hth\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_htm\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_htm\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lth\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lth\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lt\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lt\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_ltm\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_ltm\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_nth\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_nth\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_nt\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_nt\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_hth.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_htm.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lth.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_ltm.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_nt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_nth.db ]

set target\_library [list /home/alamalhh/Semester2/test\_n\_power/project4\_new/db/saed90nm\_typ\_ht(1).db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_hth\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_htm\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_htm\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lth\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lth\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lt\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lt\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_ltm\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_ltm\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_nth\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_nth\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_nt\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_nt\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_hth.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_htm.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lth.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_ltm.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_nt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_nth.db ]

set symbol\_library [list /home/alamalhh/Semester2/test\_n\_power/project4\_new/db/saed90nm(1).sdb]

analyze -format vhdl GCD\_Calculator.vhd

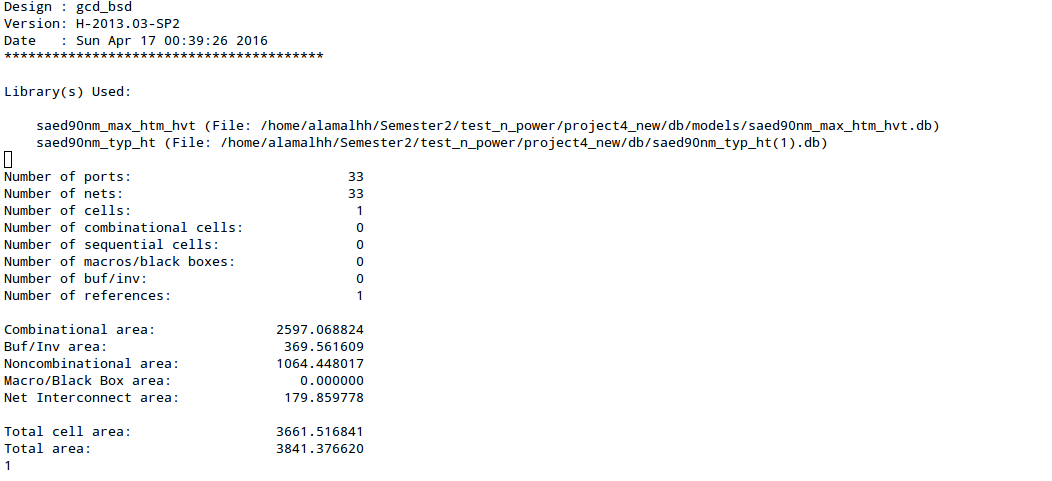
elaborate gcd\_bsd

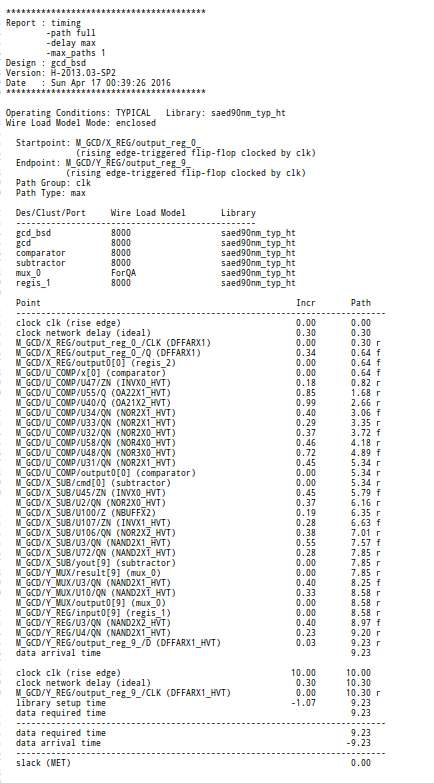
current\_design gcd\_bsd

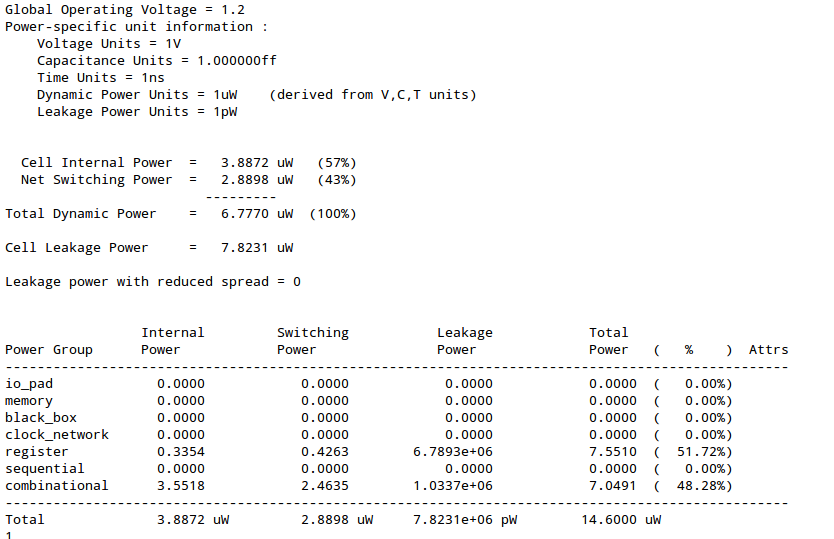
Then the synthesis is performed with additional constraint of :

Set\_max\_leakage\_power 0

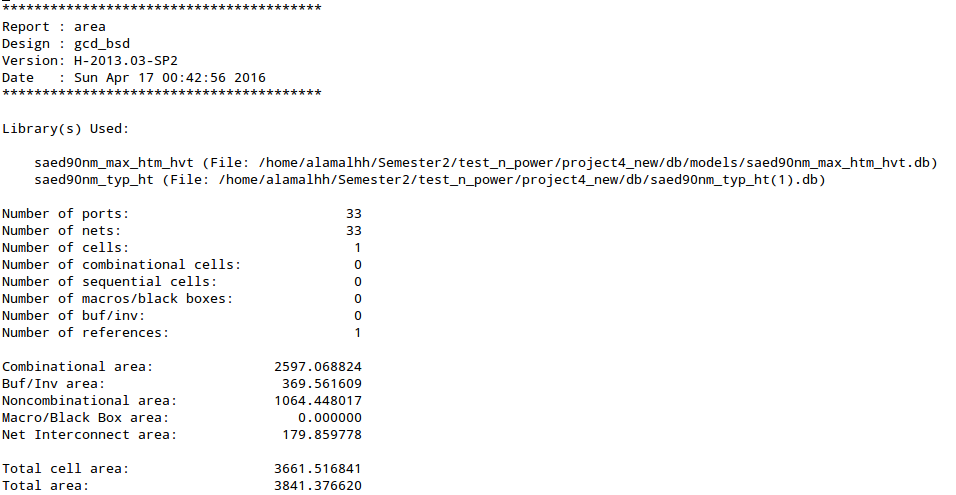
The area /timing/power numbers for Set\_max\_leakage\_power 0

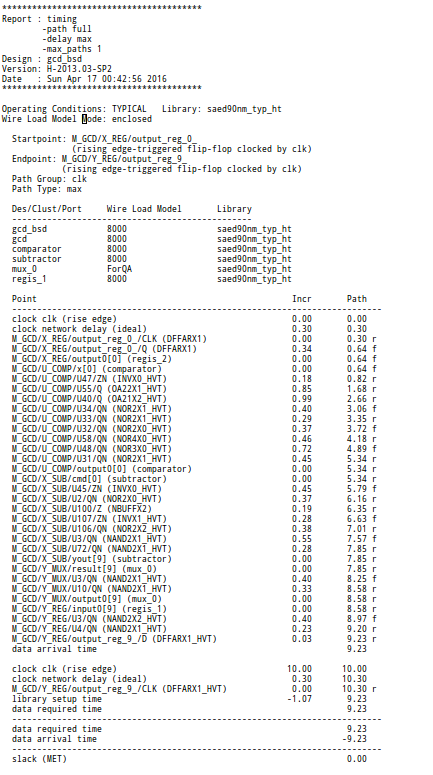


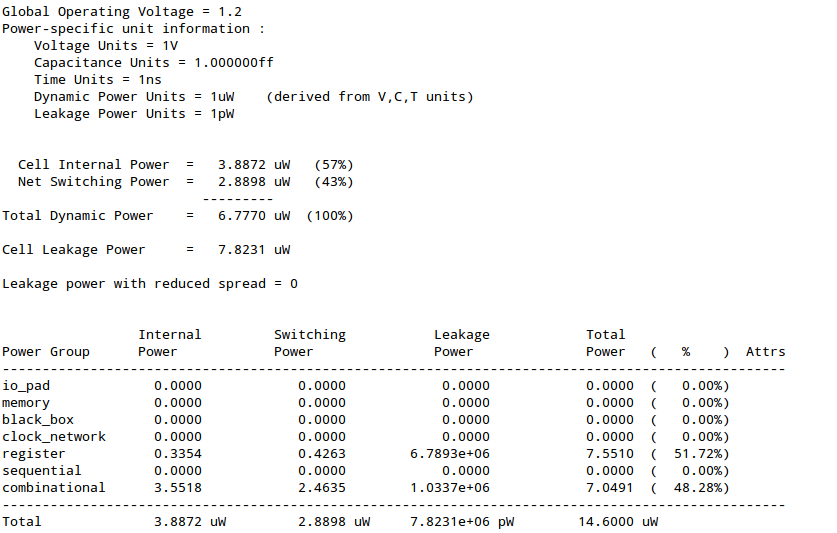




**The area /power numbers for Set\_max\_dynamic\_power 0**







|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Type** | **Area** | **Timing( slack)** | **Power** | | |
| **Dynamic(uW)** | **Leakage(uW)** | **Total** |
| Single Vt | Without any constraint | 3413.245651 | 6.14(posive) | 49.7189 | 44.0977 | 93.8166 |
| Multi Vt | Set\_max\_leakage\_power 0 | 3841.376620 | 0.00 | 6.7770 | 7.8231 | 14.600 |
| set\_max\_dynamic\_power 0 | 3841.376620 | 0.00 | 6.7770 | 7.8231 | 14.600 |
| With both (no SAIF)  set\_max\_dynamic\_power 0  Set\_max\_leakage\_power 0 | 3841.376620 | 0.00 | 8.3541 | 7.6768 | 16.0309 |
| **With both (with SAIF)**  **set\_max\_dynamic\_power 0**  **Set\_max\_leakage\_power 0** | **3841.376620** | **0.00** | **6.7770** | **7.8231** | **14.600** |

1. Set\_max\_leakage\_power 0 is used to reduce the leakage power consumption, the leakage power consumption is reduced by giving the Design compiler the multi Vt cells and the tool has optimised the paths based on the power and timing, this is visible from the positive slack obtained in the without any any constraint, but with the constraint the positive slack is not present meaning the excessive timing has been taken as advantage to reduce the power
2. The dynamic power consumption is reduced by set\_max\_dynamic\_power 0.

First the SAIF is not passed to the tool and then the timing,area and power metrics are captured then similarly the these metric are captured again after passing the SAIF.

The overall power consumption has been reduced with multi Vt cells and with the SAIF, which is clearly captured in the table above.

Because of the multi Vt cells the area is also increased a little.

**Conclusions:**

**The power numbers can differ from the SAIF file passed while calculating the power, so to get exact power it is always better to pass the SAIF file while capturing the power numbers.**

**Also by using some techniques likes clocking and the synthesis power optimization constraints the power consumed can be reduced further.Also we can use mui Vt cells also to reduce the power further.**

**From the reports we can also see the differences in the netlist in which the cells like AND and NOT are replaced with AND\_HVT and NOT \_HVT, meaning they are replaced with the high Vt cells. This difference can also be seen using the report cell command.**

All the above experiments capture these conclusions.

**Additional reports ( Netlist before and after power optimization and the synthesis constraints file):**

**Netlist before power optimization:**

module fsm ( rst, clk, proceed, comparison, enable, ysel, xld, yld, xsel\_BAR

);

input [1:0] comparison;

input rst, clk, proceed;

output enable, ysel, xld, yld, xsel\_BAR;

wire xsel, cState\_1\_, n1, n3, n4, n2, n5, n7, n8, n9, n10, n11, n12, n14,

n15, n16;

wire [2:0] nState;

DFFARX1 cState\_reg\_0\_ ( .D(nState[0]), .CLK(clk), .RSTB(n1), .Q(n7), .QN(n4)

);

DFFARX1 cState\_reg\_1\_ ( .D(nState[1]), .CLK(clk), .RSTB(n1), .Q(cState\_1\_)

);

DFFARX1 cState\_reg\_2\_ ( .D(nState[2]), .CLK(clk), .RSTB(n1), .Q(n8), .QN(n3)

);

NOR2X0 U3 ( .IN1(n3), .IN2(n7), .QN(xsel) );

INVX0 U4 ( .INP(xsel), .ZN(xsel\_BAR) );

NAND2X1 U5 ( .IN1(n16), .IN2(n15), .QN(yld) );

INVX0 U6 ( .INP(n14), .ZN(n15) );

AND2X1 U7 ( .IN1(cState\_1\_), .IN2(xsel), .Q(enable) );

NAND2X1 U8 ( .IN1(n16), .IN2(xsel\_BAR), .QN(xld) );

INVX0 U9 ( .INP(n9), .ZN(n12) );

NAND3X0 U10 ( .IN1(cState\_1\_), .IN2(n3), .IN3(n7), .QN(n9) );

INVX0 U11 ( .INP(rst), .ZN(n1) );

OR2X1 U12 ( .IN1(n9), .IN2(comparison[1]), .Q(n2) );

NAND2X0 U13 ( .IN1(proceed), .IN2(n4), .QN(n5) );

NAND4X0 U14 ( .IN1(n3), .IN2(n16), .IN3(n2), .IN4(n5), .QN(nState[0]) );

NAND2X0 U15 ( .IN1(n11), .IN2(n10), .QN(nState[1]) );

NAND2X1 U16 ( .IN1(cState\_1\_), .IN2(n4), .QN(n16) );

NOR3X0 U17 ( .IN1(n3), .IN2(cState\_1\_), .IN3(n4), .QN(n14) );

NAND3X0 U18 ( .IN1(n12), .IN2(comparison[0]), .IN3(comparison[1]), .QN(n11)

);

OA222X1 U19 ( .IN1(cState\_1\_), .IN2(n3), .IN3(cState\_1\_), .IN4(n4), .IN5(n8),

.IN6(n16), .Q(n10) );

OA21X1 U20 ( .IN1(comparison[1]), .IN2(comparison[0]), .IN3(n12), .Q(

nState[2]) );

OR2X1 U21 ( .IN1(n14), .IN2(enable), .Q(ysel) );

endmodule

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

input [9:0] load;

input [9:0] result;

output [9:0] output0;

input rst, sLine\_BAR;

wire n1, n3, n4;

INVX0 U2 ( .INP(sLine\_BAR), .ZN(n1) );

AO22X1 U3 ( .IN1(load[6]), .IN2(n3), .IN3(result[6]), .IN4(n4), .Q(

output0[6]) );

AO22X1 U4 ( .IN1(load[7]), .IN2(n3), .IN3(result[7]), .IN4(n4), .Q(

output0[7]) );

AO22X1 U5 ( .IN1(load[8]), .IN2(n3), .IN3(result[8]), .IN4(n4), .Q(

output0[8]) );

AO22X1 U6 ( .IN1(load[9]), .IN2(n3), .IN3(result[9]), .IN4(n4), .Q(

output0[9]) );

NOR2X0 U7 ( .IN1(rst), .IN2(sLine\_BAR), .QN(n4) );

NOR2X0 U8 ( .IN1(rst), .IN2(n1), .QN(n3) );

AO22X1 U9 ( .IN1(n4), .IN2(result[0]), .IN3(n3), .IN4(load[0]), .Q(

output0[0]) );

AO22X1 U10 ( .IN1(n4), .IN2(result[1]), .IN3(n3), .IN4(load[1]), .Q(

output0[1]) );

AO22X1 U11 ( .IN1(n4), .IN2(result[2]), .IN3(n3), .IN4(load[2]), .Q(

output0[2]) );

AO22X1 U12 ( .IN1(n4), .IN2(result[3]), .IN3(n3), .IN4(load[3]), .Q(

output0[3]) );

AO22X1 U13 ( .IN1(n4), .IN2(result[4]), .IN3(n3), .IN4(load[4]), .Q(

output0[4]) );

AO22X1 U14 ( .IN1(n4), .IN2(result[5]), .IN3(n3), .IN4(load[5]), .Q(

output0[5]) );

endmodule

module regis\_2 ( rst, clk, load, input0, output0 );

input [9:0] input0;

output [9:0] output0;

input rst, clk, load;

wire n11, n12, n13, n14, n15, n16, n17, n18, n19, n20, n21, n22, n23, n24,

n25, n26, n27, n28, n29, n30, n31, n1, n2, n3, n4, n6, n7, n8, n9,

n10, n32;

DFFARX1 output\_reg\_9\_ ( .D(n31), .CLK(clk), .RSTB(n30), .Q(output0[9]), .QN(

n11) );

DFFARX1 output\_reg\_8\_ ( .D(n29), .CLK(clk), .RSTB(n30), .Q(output0[8]), .QN(

n12) );

DFFARX1 output\_reg\_7\_ ( .D(n28), .CLK(clk), .RSTB(n30), .Q(output0[7]), .QN(

n13) );

DFFARX1 output\_reg\_6\_ ( .D(n27), .CLK(clk), .RSTB(n30), .Q(output0[6]), .QN(

n14) );

DFFARX1 output\_reg\_5\_ ( .D(n26), .CLK(clk), .RSTB(n30), .Q(output0[5]), .QN(

n15) );

DFFARX1 output\_reg\_4\_ ( .D(n25), .CLK(clk), .RSTB(n30), .Q(output0[4]), .QN(

n16) );

DFFARX1 output\_reg\_3\_ ( .D(n24), .CLK(clk), .RSTB(n30), .Q(output0[3]), .QN(

n17) );

**Netlist after power optimization:**

module fsm ( rst, clk, proceed, comparison, enable, xsel, ysel, xld, yld );

input [1:0] comparison;

input rst, clk, proceed;

output enable, xsel, ysel, xld, yld;

wire cState\_1\_, n3, n4, n1, n2, n5, n6, n7, n8, n9, n10, n11, n12, n13,

n14, n15, n16, n17, n19, n20, n21, n22, n23;

wire [2:1] nState;

DFFARX1\_HVT cState\_reg\_0\_ ( .D(n5), .CLK(clk), .RSTB(n19), .Q(n1), .QN(n4)

);

DFFARX1\_HVT cState\_reg\_1\_ ( .D(nState[1]), .CLK(clk), .RSTB(n19), .Q(

cState\_1\_), .QN(n2) );

DFFARX1\_HVT cState\_reg\_2\_ ( .D(nState[2]), .CLK(clk), .RSTB(n19), .Q(n6),

.QN(n3) );

NOR2X0\_HVT U3 ( .IN1(n1), .IN2(n3), .QN(xsel) );

INVX0\_HVT U4 ( .INP(xsel), .ZN(n20) );

NAND3X0\_HVT U5 ( .IN1(comparison[1]), .IN2(n12), .IN3(comparison[0]), .QN(

n11) );

INVX0\_HVT U6 ( .INP(rst), .ZN(n19) );

NOR2X2\_HVT U7 ( .IN1(n2), .IN2(n20), .QN(enable) );

NAND2X1\_HVT U8 ( .IN1(cState\_1\_), .IN2(n4), .QN(n23) );

NAND2X0\_HVT U9 ( .IN1(n1), .IN2(n6), .QN(n17) );

NAND2X2\_HVT U10 ( .IN1(n23), .IN2(n20), .QN(xld) );

NOR2X0\_HVT U11 ( .IN1(cState\_1\_), .IN2(n17), .QN(n21) );

NAND2X2\_HVT U12 ( .IN1(n23), .IN2(n22), .QN(yld) );

INVX0\_HVT U13 ( .INP(n21), .ZN(n22) );

NAND3X0\_HVT U14 ( .IN1(n11), .IN2(n10), .IN3(n9), .QN(nState[1]) );

NAND2X0\_HVT U15 ( .IN1(n8), .IN2(n3), .QN(n10) );

NAND2X0\_HVT U16 ( .IN1(proceed), .IN2(n4), .QN(n14) );

INVX0\_HVT U17 ( .INP(n12), .ZN(n13) );

INVX0\_HVT U18 ( .INP(n23), .ZN(n8) );

NOR2X0\_HVT U19 ( .IN1(n2), .IN2(n7), .QN(n12) );

NAND2X0\_HVT U20 ( .IN1(n1), .IN2(n3), .QN(n7) );

OR2X1\_HVT U21 ( .IN1(n16), .IN2(n15), .Q(n5) );

NOR2X0\_HVT U22 ( .IN1(comparison[1]), .IN2(n13), .QN(n16) );

OA21X1\_HVT U23 ( .IN1(comparison[0]), .IN2(comparison[1]), .IN3(n12), .Q(

nState[2]) );

AO21X1\_HVT U24 ( .IN1(n3), .IN2(n4), .IN3(cState\_1\_), .Q(n9) );

NAND3X0\_HVT U25 ( .IN1(n23), .IN2(n3), .IN3(n14), .QN(n15) );

OR2X1\_HVT U26 ( .IN1(n21), .IN2(enable), .Q(ysel) );

endmodule

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

input [9:0] load;

input [9:0] result;

output [9:0] output0;

input rst, sLine;

wire n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15, n16,

n17, n18, n19, n20, n21, n22, n23;

NOR2X1\_HVT U2 ( .IN1(n1), .IN2(rst), .QN(n20) );

NAND2X1\_HVT U3 ( .IN1(result[8]), .IN2(n20), .QN(n19) );

NAND2X0\_HVT U4 ( .IN1(result[7]), .IN2(n20), .QN(n17) );

NAND2X0\_HVT U5 ( .IN1(result[6]), .IN2(n20), .QN(n15) );

NAND2X0\_HVT U6 ( .IN1(result[4]), .IN2(n20), .QN(n11) );

NAND2X0\_HVT U7 ( .IN1(result[5]), .IN2(n20), .QN(n13) );

NAND2X0\_HVT U8 ( .IN1(result[2]), .IN2(n20), .QN(n7) );

NAND2X0\_HVT U9 ( .IN1(result[3]), .IN2(n20), .QN(n9) );

NAND2X0\_HVT U10 ( .IN1(result[1]), .IN2(n20), .QN(n5) );

NAND2X0\_HVT U11 ( .IN1(result[0]), .IN2(n20), .QN(n3) );

NAND2X0\_HVT U12 ( .IN1(result[9]), .IN2(n20), .QN(n23) );

NOR2X1\_HVT U13 ( .IN1(sLine), .IN2(rst), .QN(n21) );

NAND2X0\_HVT U14 ( .IN1(n3), .IN2(n2), .QN(output0[0]) );

NAND2X0\_HVT U15 ( .IN1(n7), .IN2(n6), .QN(output0[2]) );

NAND2X0\_HVT U16 ( .IN1(n15), .IN2(n14), .QN(output0[6]) );

NAND2X0\_HVT U17 ( .IN1(n11), .IN2(n10), .QN(output0[4]) );

NAND2X1\_HVT U18 ( .IN1(n23), .IN2(n22), .QN(output0[9]) );

NAND2X0\_HVT U19 ( .IN1(n17), .IN2(n16), .QN(output0[7]) );

NAND2X0\_HVT U20 ( .IN1(n9), .IN2(n8), .QN(output0[3]) );

NAND2X0\_HVT U21 ( .IN1(n13), .IN2(n12), .QN(output0[5]) );

NAND2X0\_HVT U22 ( .IN1(n5), .IN2(n4), .QN(output0[1]) );

NAND2X0\_HVT U23 ( .IN1(n21), .IN2(load[3]), .QN(n8) );

NAND2X0\_HVT U24 ( .IN1(n21), .IN2(load[5]), .QN(n12) );

NAND2X0\_HVT U25 ( .IN1(n21), .IN2(load[0]), .QN(n2) );

NAND2X0\_HVT U26 ( .IN1(n21), .IN2(load[7]), .QN(n16) );

NAND2X0\_HVT U27 ( .IN1(n21), .IN2(load[2]), .QN(n6) );

NAND2X0\_HVT U28 ( .IN1(n21), .IN2(load[9]), .QN(n22) );

NAND2X0\_HVT U29 ( .IN1(n21), .IN2(load[4]), .QN(n10) );

NAND2X0\_HVT U30 ( .IN1(n21), .IN2(load[6]), .QN(n14) );

NAND2X0\_HVT U31 ( .IN1(n21), .IN2(load[8]), .QN(n18) );

NAND2X0\_HVT U32 ( .IN1(n21), .IN2(load[1]), .QN(n4) );

NAND2X1\_HVT U33 ( .IN1(n19), .IN2(n18), .QN(output0[8]) );

INVX0\_HVT U34 ( .INP(sLine), .ZN(n1) );

endmodule

module regis\_2 ( rst, clk, load, input0, output0 );

input [9:0] input0;

output [9:0] output0;

input rst, clk, load;

wire n11, n12, n13, n14, n15, n16, n17, n18, n19, n20, n21, n22, n23, n24,

n25, n26, n27, n28, n29, n31, n2, n3, n4, n5, n6, n7, n8, n9, n10,

n30, n32, n33, n34, n35, n36, n37, n38, n39, n40, n41, n42;

DFFARX1\_HVT output\_reg\_9\_ ( .D(n31), .CLK(clk), .RSTB(n42), .Q(output0[9]),

.QN(n11) );

DFFARX1 output\_reg\_0\_ ( .D(n21), .CLK(clk), .RSTB(n42), .Q(output0[0]), .QN(

n20) );

DFFARX1 output\_reg\_1\_ ( .D(n22), .CLK(clk), .RSTB(n42), .Q(output0[1]), .QN(

n19) );

DFFARX1 output\_reg\_3\_ ( .D(n24), .CLK(clk), .RSTB(n42), .Q(output0[3]), .QN(

n17) );

DFFARX1 output\_reg\_2\_ ( .D(n23), .CLK(clk), .RSTB(n42), .Q(output0[2]), .QN(

n18) );

DFFARX1 output\_reg\_5\_ ( .D(n26), .CLK(clk), .RSTB(n42), .Q(output0[5]), .QN(

n15) );

DFFARX1 output\_reg\_4\_ ( .D(n25), .CLK(clk), .RSTB(n42), .Q(output0[4]), .QN(

n16) );

DFFARX1 output\_reg\_6\_ ( .D(n27), .CLK(clk), .RSTB(n42), .Q(output0[6]), .QN(

n14) );

DFFARX1 output\_reg\_7\_ ( .D(n28), .CLK(clk), .RSTB(n42), .Q(output0[7]), .QN(

n13) );

DFFARX1\_HVT output\_reg\_8\_ ( .D(n29), .CLK(clk), .RSTB(n42), .Q(output0[8]),

.QN(n12) );

NAND2X0\_HVT U2 ( .IN1(n40), .IN2(n7), .QN(n29) );

INVX1\_HVT U3 ( .INP(rst), .ZN(n42) );

NAND2X0\_HVT U4 ( .IN1(input0[6]), .IN2(load), .QN(n38) );

NAND2X0\_HVT U5 ( .IN1(input0[2]), .IN2(load), .QN(n34) );

NAND2X0\_HVT U6 ( .IN1(input0[4]), .IN2(load), .QN(n36) );

NAND2X0\_HVT U7 ( .IN1(input0[0]), .IN2(load), .QN(n32) );

NAND2X1\_HVT U8 ( .IN1(input0[9]), .IN2(load), .QN(n41) );

NAND2X0\_HVT U9 ( .IN1(n33), .IN2(n30), .QN(n22) );

NAND2X0\_HVT U10 ( .IN1(n39), .IN2(n6), .QN(n28) );

NAND2X0\_HVT U11 ( .IN1(n32), .IN2(n9), .QN(n21) );

NAND2X0\_HVT U12 ( .IN1(n36), .IN2(n10), .QN(n25) );

NAND2X0\_HVT U13 ( .IN1(n34), .IN2(n2), .QN(n23) );

NAND2X0\_HVT U14 ( .IN1(n38), .IN2(n5), .QN(n27) );

NAND2X0\_HVT U15 ( .IN1(n37), .IN2(n4), .QN(n26) );

NAND2X0\_HVT U16 ( .IN1(n35), .IN2(n3), .QN(n24) );

NAND2X0\_HVT U17 ( .IN1(input0[1]), .IN2(load), .QN(n33) );

NAND2X0\_HVT U18 ( .IN1(input0[3]), .IN2(load), .QN(n35) );

NAND2X0\_HVT U19 ( .IN1(input0[7]), .IN2(load), .QN(n39) );

NAND2X0\_HVT U20 ( .IN1(input0[5]), .IN2(load), .QN(n37) );

NAND2X1\_HVT U21 ( .IN1(n41), .IN2(n8), .QN(n31) );

NAND2X1\_HVT U22 ( .IN1(input0[8]), .IN2(load), .QN(n40) );

OR2X1\_HVT U23 ( .IN1(load), .IN2(n19), .Q(n30) );

OR2X1\_HVT U24 ( .IN1(load), .IN2(n16), .Q(n10) );

OR2X1\_HVT U25 ( .IN1(load), .IN2(n18), .Q(n2) );

OR2X1\_HVT U26 ( .IN1(load), .IN2(n15), .Q(n4) );

OR2X1\_HVT U27 ( .IN1(load), .IN2(n13), .Q(n6) );

OR2X1\_HVT U28 ( .IN1(load), .IN2(n14), .Q(n5) );

OR2X1\_HVT U29 ( .IN1(load), .IN2(n12), .Q(n7) );

OR2X1\_HVT U30 ( .IN1(load), .IN2(n17), .Q(n3) );

OR2X1\_HVT U31 ( .IN1(load), .IN2(n20), .Q(n9) );

OR2X1\_HVT U32 ( .IN1(load), .IN2(n11), .Q(n8) );

endmodule

module comparator ( rst, x, y, output0 );

input [9:0] x;

input [9:0] y;

output [1:0] output0;

input rst;

wire n58, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15,

n16, n17, n18, n19, n20, n21, n22, n23, n24, n25, n26, n27, n28, n29,

n30, n31, n32, n33, n34, n35, n36, n37, n38, n39, n40, n41, n42, n43,

n44, n45, n46, n47, n48, n49, n50, n51, n52, n53, n54, n55, n56, n57;

NAND2X0\_HVT U3 ( .IN1(n19), .IN2(y[4]), .QN(n50) );

NOR2X0\_HVT U4 ( .IN1(n50), .IN2(n20), .QN(n25) );

NAND2X0\_HVT U5 ( .IN1(n11), .IN2(x[3]), .QN(n13) );

INVX0\_HVT U6 ( .INP(y[3]), .ZN(n11) );

INVX0\_HVT U7 ( .INP(x[2]), .ZN(n12) );

INVX0\_HVT U8 ( .INP(x[4]), .ZN(n19) );

NAND2X0\_HVT U9 ( .IN1(n12), .IN2(y[2]), .QN(n51) );

**Synthesis constraint File:**

#################################

#Library setup

#################################

set search\_path [list /home/alamalhh/Semester2/test\_n\_power/project4\_new/db /home/alamalhh/Semester2/test\_n\_power/project4\_new]

set link\_library [list /home/alamalhh/Semester2/test\_n\_power/project4\_new/db/saed90nm\_typ\_ht(1).db ]

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_hth\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_htm\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_htm\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lth\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lth\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lt\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lt\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_ltm\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_ltm\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_nth\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_nth\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_nt\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_nt\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_hth.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_htm.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lth.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_ltm.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_nt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_nth.db ]

set target\_library [list /home/alamalhh/Semester2/test\_n\_power/project4\_new/db/saed90nm\_typ\_ht(1).db ]

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_hth\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_htm\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_htm\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lth\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lth\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lt\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lt\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_ltm\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_ltm\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_nth\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_nth\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_nt\_hvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_nt\_lvt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_hth.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_htm.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_lth.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_ltm.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_nt.db \

/home/alamalhh/Semester2/test\_n\_power/project4\_new/db/models/saed90nm\_max\_nth.db ]

set symbol\_library [list /home/alamalhh/Semester2/test\_n\_power/project4\_new/db/saed90nm(1).sdb]

analyze -format vhdl GCD\_Calculator.vhd

elaborate gcd\_bsd

current\_design gcd\_bsd

##################################

##link to standard cell library and uniquify

##################################

link

uniquify

#################################

# Create clock and other constraints

#################################

create\_clock clk -period 10 -waveform {0 5} -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

#set\_max\_leakage\_power 0

#set\_max\_dynamic\_power 0

#################################

#compile design

#################################

check\_design

compile\_ultra -no\_autoungroup

set basename gcd\_bsd

set runname power\_gating

set filebase [format "%s%s" [format "%s%s" $basename "\_"] $runname]

set filename [format "%s%s%s" ./reports\_new/ $filebase ".vhdl"]

redirect change\_names { change\_names -rules verilog -hierarchy -verbose }

write -format vhdl -hierarchy -output $filename

read\_saif -input /home/alamalhh/Semester2/test\_n\_power/project4\_new/new\_gcd/gate\_level\_april\_16\_2.saif -instance tb\_gcd\_bsd/GCD -verbose

set filename [format "%s%s%s" ./reports\_new/ $filebase ".cell"]

redirect $filename { report\_cell }

set filename [format "%s%s%s" ./reports\_new/ $filebase ".port"]

redirect $filename { report\_port }

set filename [format "%s%s%s" ./reports\_new/ $filebase ".design"]

redirect $filename { report\_design }

set filename [format "%s%s%s" ./reports\_new/ $filebase ".timing"]

redirect $filename { report\_timing }

set filename [format "%s%s%s" ./reports\_new/ $filebase ".area"]

redirect $filename { report\_area }

set filename [format "%s%s%s" ./reports\_new/ $filebase ".power"]

redirect $filename { report\_power }