BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE 468 (July 2023)
VLSI Laboratory

**Final Project Report**

**Section: G2 Group: 02**

# 12 bit Asynchronous Counter

**Course Instructors:**
**Nafis Sadik, Lecturer**
**Rafid Hassan Palash, Part-Time Lecturer**

**Signature of Instructor:** ______________________________

**Academic Honesty Statement:**

**IMPORTANT! Please carefully read and sign the Academic Honesty Statement, below. Type the student ID and name, and put your signature.** ***You will not receive credit for this project experiment unless this statement is signed in the presence of your lab instructor.***

*"In signing this statement, We hereby certify that the work on this project is our own and that we have not copied the work of any other students (past or present), and cited all relevant sources while completing this project. We understand that if we fail to honor this agreement, We will each receive a score of ZERO for this project and be subject to failure of this course."*

| | | |
|---|---|---|
| **Signature:** ____________ **Full Name: Abid Abdullah** **Student ID: 1906108** | **Signature:** ____________ **Full Name: Shahriyer Hasan** **Student ID: 1906128** | **Signature:** ____________ **Full Name: Khondaker Nafis Ahmed** **Student ID: 1906147** |
| **Signature:** ____________ **Full Name: Manjur Muntasir** **Student ID: 1906150** | **Signature:** ____________ **Full Name: Fahim Shahriar Anim** **Student ID: 1906178** | |

**Table of Contents**

## 1 Abstract

A 12-bit asynchronous counter, also known as a ripple counter, is a type of digital counter used in electronics to count pulses or events. This allows the counter to count from 0 to $2^{12}-1$, or 0 to 4095 in decimal. Our goal is to design an optimized 12-bit asynchronous counter where all control signals (Load, Up/Down, Reset, Enable) are asynchronous.

## 2 Introduction

A **counter** is a sequential logic circuit that goes through a predetermined sequence of states in response to clock pulses. Counters are fundamental components in digital electronics and are widely used in various applications such as counting events, timing, and frequency division. In an asynchronous counter, flip-flops are not clocked simultaneously. Each flip-flop's output serves as the clock input for the next flip-flop in the sequence. The advantage of an asynchronous counter is it possesses simple design and requires fewer hardware, but the major disadvantage is propagation delay as the clock signal ripples through the flip-flops, which limits speed.

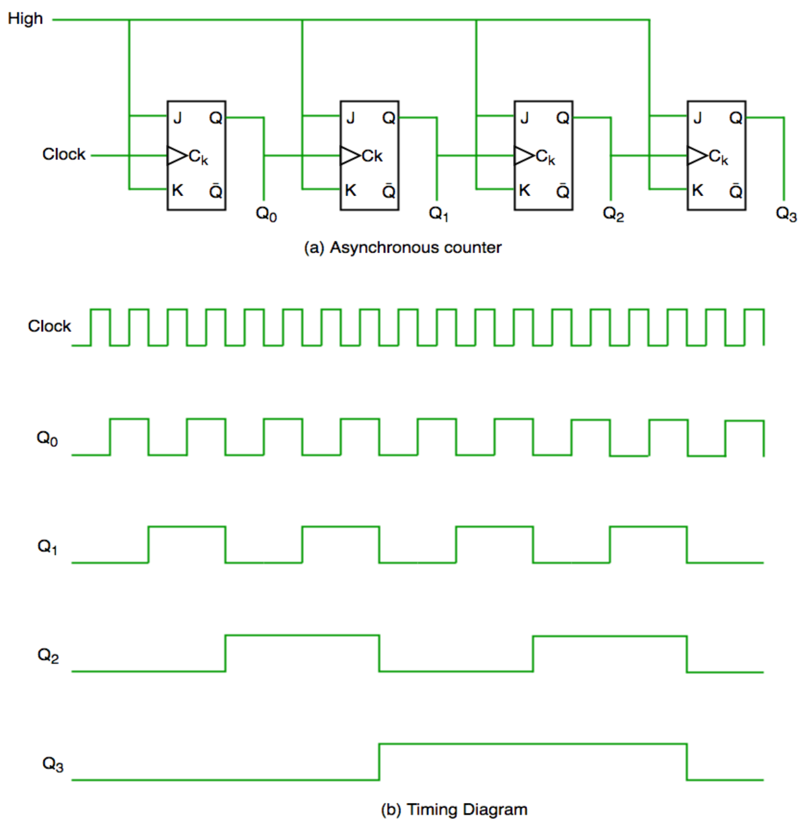


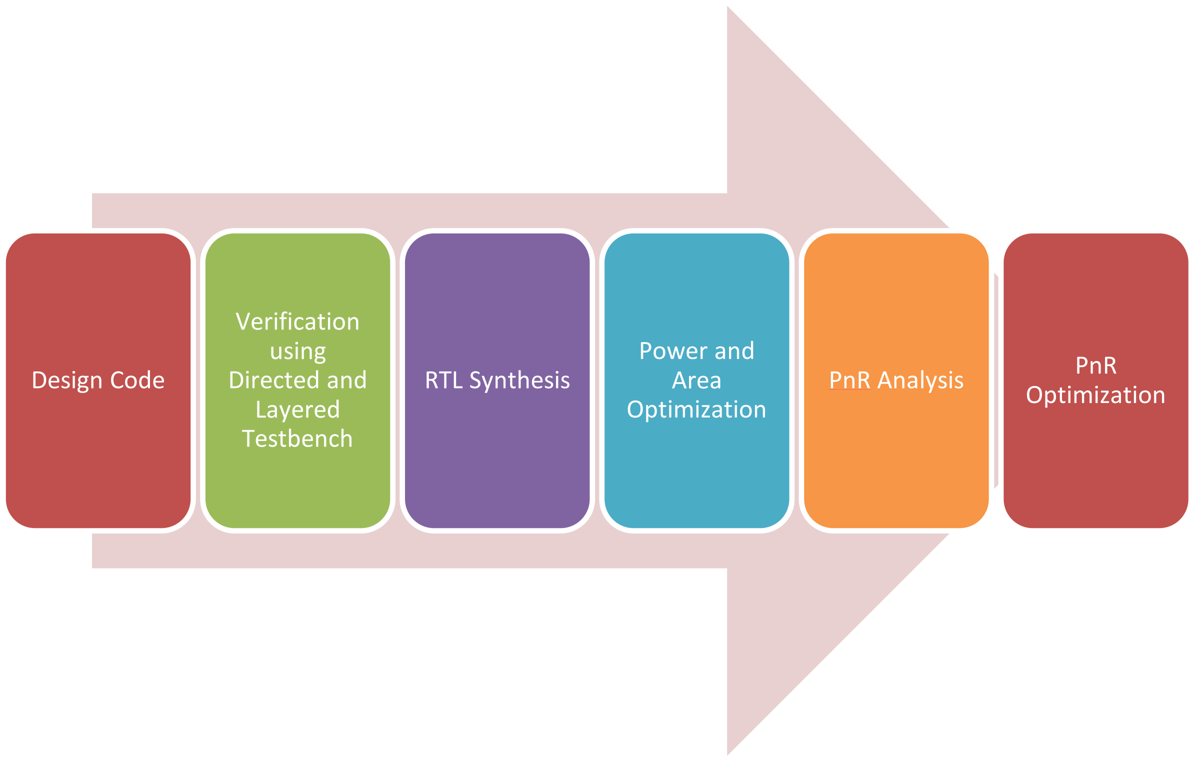
Figure 1.1: A 4 bit Asynchronous Counter

## 3 Design

### 3.1 Problem Formulation

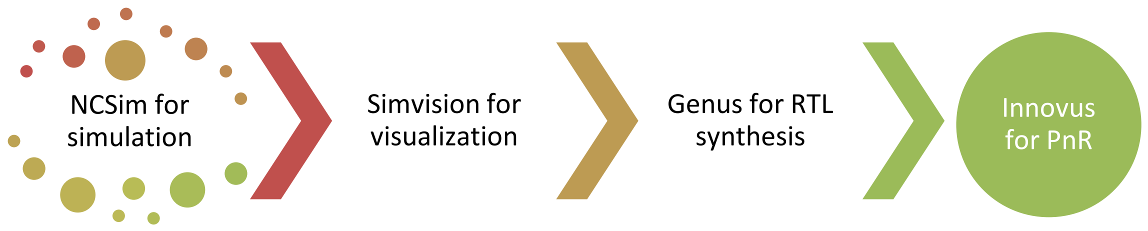
| Project Name | Inputs | Outputs | Additional Description |
|---|---|---|---|
| 12 bit asynchronous counter | **Load** (If load is high, parallel load will take place), Number to be loaded **A[11:0]**, **Up/$\overline{\textbf{Down}}$** (If high, up counts, if low, down counts), **Reset** (If high, counter will be reset to count zero), **Enable** (If low, counter output won't change), Clock signal **Clk** | **Count[11:0]** | All control signals (Load, Up/$\overline{\text{Down}}$, Reset, Enable) are asynchronous |

### 3.2 Problem Analysis


Design Code
Verification using Directed and Layered Testbench
RTL Synthesis
Power and Area Optimization
PnR Analysis
PnR Optimization

## 3.3Tools Used

Cadence CAD Tools



# 4 Design

## 4.1 Verilog Code

```
 module counter(clk, reset, up_down, load, data, count, enable);

  //define input and output ports
  input clk, reset, load, up_down, enable;
  input [11:0] data;
  output reg [11:0] count;

  //always block will be executed at each and every positive edge of the clock

  always@(posedge reset)
    begin
      if(!enable) count<=count;
      else count<=0;
    end

  always@(negedge reset)
    begin
      if(!enable) count<=count;
      else if(load) count<=data;
      else if(up_down) count<=count+1;
      else if(!up_down) count<=count-1;
    end

  always@(posedge load)
    begin
      if(!enable) count<=count;
      else if(!reset) count<=data;
    end

always@(negedge load)
    begin
      if(!enable) count<=count;
      else if(reset) count<=0;
      else if(up_down) count<=count+1;
      else if(!up_down) count<=count-1;
    end
always@(posedge clk)
    begin
      if(!enable) count<=count;
      else if(!reset & !load)
        begin
          if(up_down==1) count<=count+1;
          else if(up_down==0) count<=count-1;
        end
    end

  always@(posedge up_down)
    begin
      if(!enable) count<=count;
      else if(reset) count<=0;
      else if(load) count<=data;
      else count<=count+1;
    end

  always@(negedge up_down)
    begin
      if(!enable) count<=count;
      else if(reset) count<=0;
      else if(load) count<=data;
      else count<=count-1;
    end
  always@(posedge enable)
    begin
      if(reset) count<=0;
      else if(load) count<=data;
      else if(up_down) count<=count+1;
      else if(!up_down) count<=count-1;
    end

  always@(data)
    if(load & !reset) count<=data;

endmodule
```

## 4.2 Directed Testbench

```
module counter_tb;
  reg clk,reset,load,up_down, enable;
  reg [11:0] data;
  wire [11:0] count;

  // instance counter design
  counter
ct_1(.clk(clk), .reset(reset), .load(load), .up_down(up_down), .enable(enable), .d
ata(data), .count(count));

  //clock generator
  initial
    begin
      clk = 1'b0;
      repeat(30) #3
        clk= ~clk;
    end

  //insert all the input signal
  initial
    begin
      reset=1'b1; #7
      reset=1'b0; #33
      reset=1'b1; #2
      reset=1'b0;
    end

  initial
    begin
      enable=1'b1; #12
      enable=1'b0; #22
      enable=1'b1;
    end


  initial
    begin
      load=1'b1; #18
      load=1'b0; #22
      load=1'b1; #8
      load=1'b0; #12
      load=1'b1; #10
      load=1'b0;
    end

  initial
    begin
      up_down=1'b1; #40
      up_down=1'b0; #20
      up_down=1'b1; #8
      up_down=1'b0; #10
      up_down=1'b1; #8
      up_down=1'b0;
    end

  initial
    begin
      data=12'b100011001000; #14
      data=12'b110100111100; #2
      data=12'b111100001111; #20
      data=12'b101001011010; #8
      data=12'b110000111100;
    end

  //monitor all the input and output ports at times when any inputs changes its
state
```

```
  initial
    begin
      $monitor("time=%0d, reset=%b, load=%b, up_down=%b, data=%d, count=%d",
$time, reset, load, up_down, data, count);
    end

  initial
    begin
      $dumpfile("dump.vcd");
      $dumpvars();
      #100
      $finish;
    end

endmodule
```

## 4.3 Layered Testbench

Testbench.sv

```
`include "testcase01.sv"
//`include "test.sv"
`include "interface.sv"
module testbench;
  bit clk;

  initial begin
    forever #5 clk =~clk;
  end

  int tnum=30;
  counter_if countif(clk);

  test test01(tnum,countif);

  initial begin
    $dumpfile("dump.vcd");
    $dumpvars;
    #200;
    $finish;
  end

  counter DUT (
    .reset(countif.reset),
    .up_down(countif.up_down),
    .load(countif.load),
    .enable(countif.enable),
    .data(countif.data),
    .count(countif.count),
    .clk(clk)
  );
endmodule
```

Interface.sv

```
interface counter_if(input clk);
  logic reset,load, up_down, enable;
  logic [11:0] data;
  logic [11:0] count;

  clocking driver_cb @(negedge clk);
    default input #0 output #0;
    output reset,load, up_down, enable, data;
  endclocking

  clocking mon_cb @(clk, reset, enable, load, up_down, data);
    default input #0 output #0;
    input reset,load, up_down, enable, data;
    input count;
  endclocking

  modport DRIVER (clocking driver_cb, input clk);
  modport MONITOR (clocking mon_cb, input clk);

endinterface
```

Transaction.sv

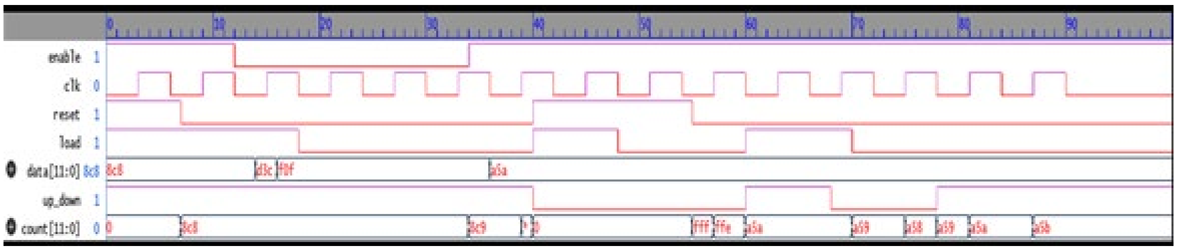
```
class transaction;
  bit [11:0] data = 5;
  rand bit reset;
  rand bit up_down;
  bit enable = 1;
  rand bit load;
  bit [11:0] count;

endclass:transaction
```

Scoreboard.sv

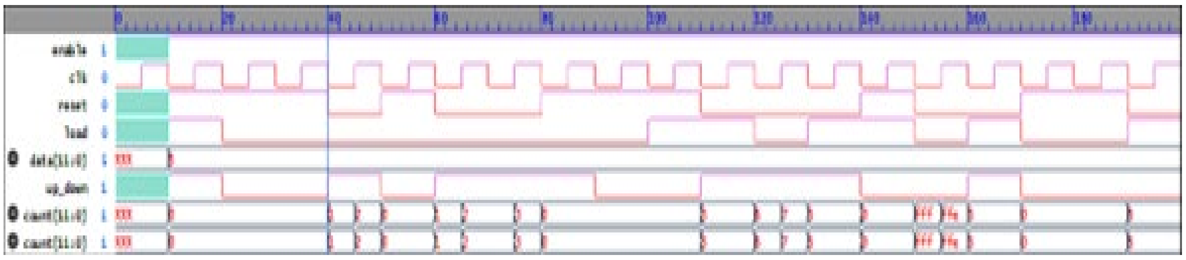
```
class scoreboard;
  mailbox driv2sb;
  mailbox mon2sb;

  transaction d_trans;
  transaction m_trans;

  event driven;

  function new(mailbox driv2sb, mon2sb);
    this.driv2sb=driv2sb;
    this.mon2sb=mon2sb;
  endfunction

  task main(input int tnum);
    $display("------------------Scoreboard Test Starts--------------------");
    repeat(tnum) begin
      m_trans=new();
      mon2sb.get(m_trans);
      report();

      if(m_trans.count != d_trans.count)
        $display("time:%d, Failed : reset=%d up_down=%d load=%d enable=%d data=%d
Expected out=%d  Resulted out=%d",$time, d_trans.reset,d_trans.up_down,
d_trans.load, d_trans.enable, d_trans.data, d_trans.count,
m_trans.count);
      else
        $display("time:%d, Passed : reset=%d up_down=%d load=%d enable=%d data=%d
Expected out=%d  Resulted out=%d",$time, d_trans.reset,d_trans.up_down,
d_trans.load, d_trans.enable, d_trans.data, d_trans.count, m_trans.count);
    end
    $display("------------------Scoreboard Test Ends--------------------");
  endtask:main

  task report();
    d_trans=new();
    driv2sb.get(d_trans);
    if (d_trans.enable==1)
     begin
       if(d_trans.reset==1) d_trans.count=0;
       else if(d_trans.load==1) d_trans.count=d_trans.data;
       else if(d_trans.up_down==1) d_trans.count=d_trans.count+1;
       else if(d_trans.up_down==0) d_trans.count=d_trans.count-1;
     end
    else d_trans.count=d_trans.count;

  endtask:report


endclass:scoreboard
```

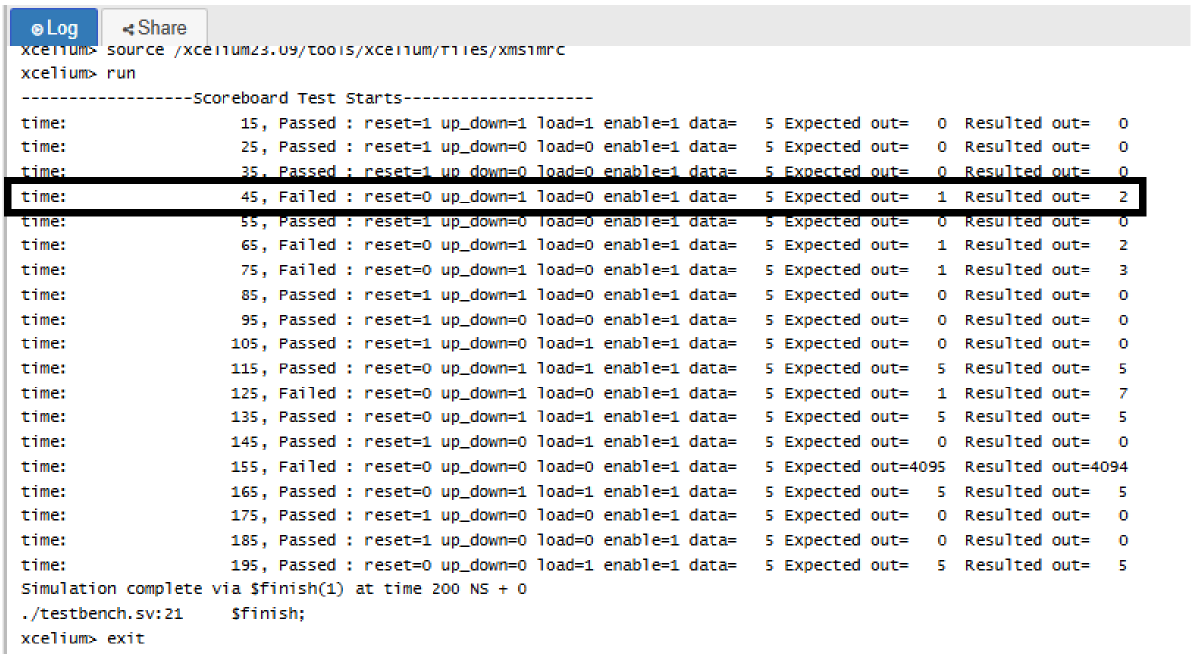
# 5 Implementation of Design and Testbench

## 5.1 Directed Testbench Waveform



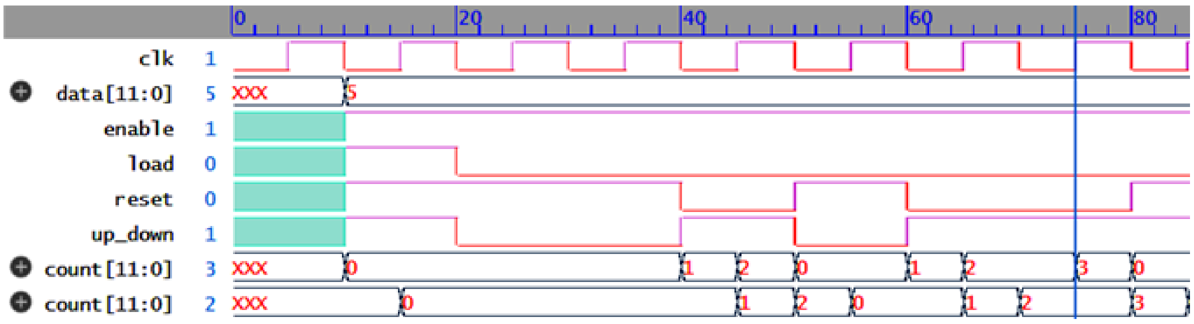
## 5.2 Layered Testbench Output



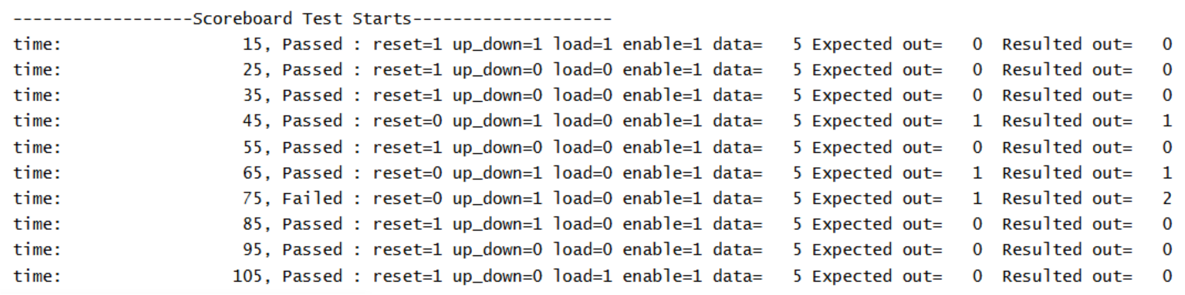


```
xcelium> source /xcelium23.09/tools/xcelium/files/xmsimrc
xcelium> run
------------------Scoreboard Test Starts--------------------
time:                    15, Passed : reset=1 up_down=1 load=1 enable=1 data=   5 Expected out=   0  Resulted out=   0
time:                    25, Passed : reset=1 up_down=0 load=0 enable=1 data=   5 Expected out=   0  Resulted out=   0
time:                    35, Passed : reset=1 up_down=0 load=0 enable=1 data=   5 Expected out=   0  Resulted out=   0
time:                    45, Failed : reset=0 up_down=1 load=0 enable=1 data=   5 Expected out=   1  Resulted out=   2
time:                    55, Passed : reset=1 up_down=0 load=0 enable=1 data=   5 Expected out=   0  Resulted out=   0
time:                    65, Failed : reset=0 up_down=1 load=0 enable=1 data=   5 Expected out=   1  Resulted out=   2
time:                    75, Failed : reset=0 up_down=1 load=0 enable=1 data=   5 Expected out=   1  Resulted out=   3
time:                    85, Passed : reset=1 up_down=1 load=0 enable=1 data=   5 Expected out=   0  Resulted out=   0
time:                    95, Passed : reset=1 up_down=0 load=0 enable=1 data=   5 Expected out=   0  Resulted out=   0
time:                   105, Passed : reset=1 up_down=0 load=1 enable=1 data=   5 Expected out=   0  Resulted out=   0
time:                   115, Passed : reset=0 up_down=1 load=1 enable=1 data=   5 Expected out=   5  Resulted out=   5
time:                   125, Failed : reset=0 up_down=1 load=0 enable=1 data=   5 Expected out=   1  Resulted out=   7
time:                   135, Passed : reset=0 up_down=1 load=1 enable=1 data=   5 Expected out=   5  Resulted out=   5
time:                   145, Passed : reset=1 up_down=0 load=1 enable=1 data=   5 Expected out=   0  Resulted out=   0
time:                   155, Failed : reset=0 up_down=0 load=0 enable=1 data=   5 Expected out=4095  Resulted out=4094
time:                   165, Passed : reset=0 up_down=1 load=1 enable=1 data=   5 Expected out=   5  Resulted out=   5
time:                   175, Passed : reset=1 up_down=0 load=0 enable=1 data=   5 Expected out=   0  Resulted out=   0
time:                   185, Passed : reset=1 up_down=0 load=0 enable=1 data=   5 Expected out=   0  Resulted out=   0
time:                   195, Passed : reset=0 up_down=0 load=1 enable=1 data=   5 Expected out=   5  Resulted out=   5
Simulation complete via $finish(1) at time 200 NS + 0
./testbench.sv:21     $finish;
xcelium> exit
```

## 5.3 Layered testbench Output v2



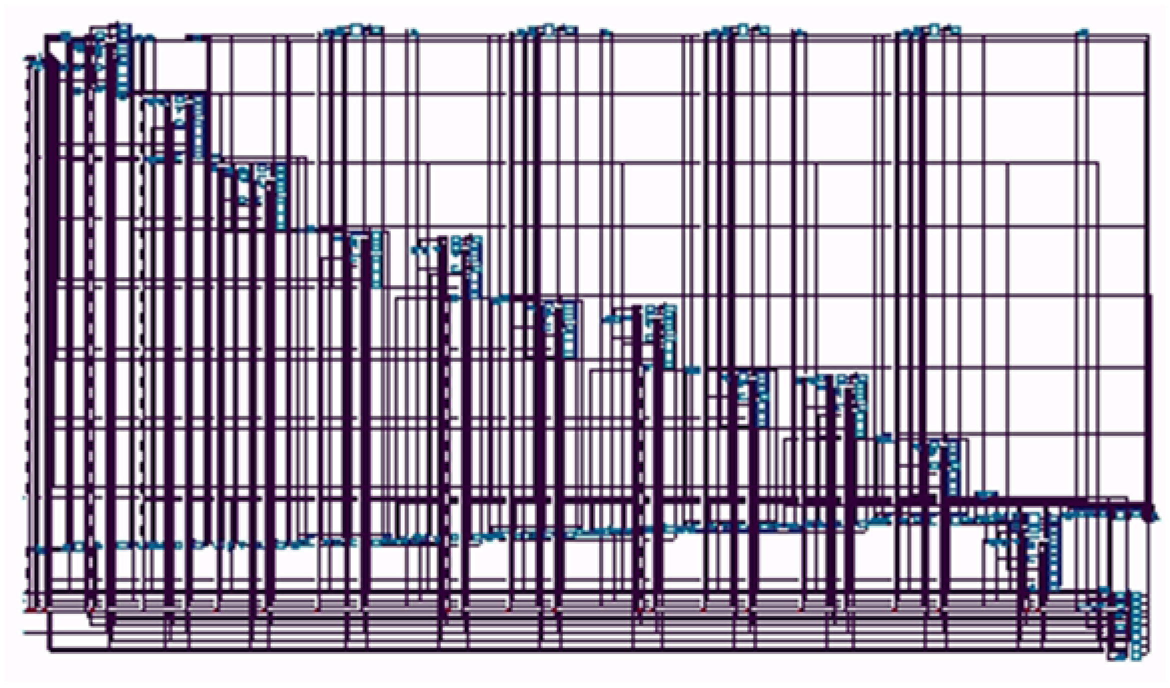
Scoreboard Monitor Result:


```
 ------------------Scoreboard Test Starts-------------------
 time:                    15, Passed : reset=1 up_down=1 load=1 enable=1 data=   5 Expected out=   0 Resulted out=   0
 time:                    25, Passed : reset=1 up_down=0 load=0 enable=1 data=   5 Expected out=   0 Resulted out=   0
 time:                    35, Passed : reset=1 up_down=0 load=0 enable=1 data=   5 Expected out=   0 Resulted out=   0
 time:                    45, Passed : reset=0 up_down=1 load=0 enable=1 data=   5 Expected out=   1 Resulted out=   1
 time:                    55, Passed : reset=1 up_down=0 load=0 enable=1 data=   5 Expected out=   0 Resulted out=   0
 time:                    65, Passed : reset=0 up_down=1 load=0 enable=1 data=   5 Expected out=   1 Resulted out=   1
 time:                    75, Failed : reset=0 up_down=1 load=0 enable=1 data=   5 Expected out=   1 Resulted out=   2
 time:                    85, Passed : reset=1 up_down=1 load=0 enable=1 data=   5 Expected out=   0 Resulted out=   0
 time:                    95, Passed : reset=1 up_down=0 load=0 enable=1 data=   5 Expected out=   0 Resulted out=   0
 time:                   105, Passed : reset=1 up_down=0 load=1 enable=1 data=   5 Expected out=   0 Resulted out=   0
```

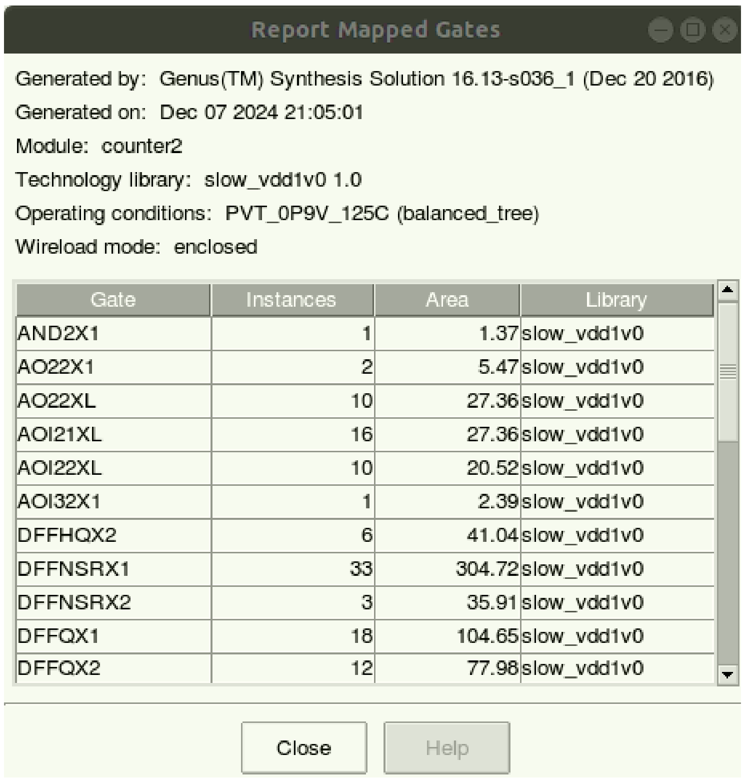
# 6 Synthesis

TCL code

```
#run Genus in Legacy UI if Genus is invoked with Common UI
::legacy::set_attribute common_ui false / ;
if {[file exists /proc/cpuinfo]} {
sh grep "model name" /proc/cpuinfo
sh grep "cpu MHz" /proc/cpuinfo
}
puts "Hostname : [info hostname]"
##############################################################################
### Preset global variables and attributes
##############################################################################
set DESIGN counter2
set SYN_EFF medium
set MAP_EFF medium
set OPT_EFF medium
# Directory of PDK
#set pdk_dir /home/wsadiq/buet_flow/GPDK045
set pdk_dir /home/cad/VLSI2Lab/Digital/library/
#set_attribute init_lib_search_path $pdk_dir/gsclib045/timing
#set_attribute init_hdl_search_path /home/wsadiq/buet_flow/rtl
set_attribute init_lib_search_path $pdk_dir
#set_attribute init_hdl_search_path ../../rtl
##Set synthesizing effort for each synthesis stage
set_attribute syn_generic_effort $SYN_EFF
set_attribute syn_map_effort $MAP_EFF
set_attribute syn_opt_effort $OPT_EFF
#set_attribute library "\
slow_vdd1v0_basicCells_hvt.lib \
slow_vdd1v0_basicCells.lib \
slow_vdd1v0_basicCells_lvt.lib"
set_attribute library "\
slow_vdd1v0_basicCells.lib"
set_dont_use [get_lib_cells CLK*]
set_dont_use [get_lib_cells SDFF*]
set_dont_use [get_lib_cells DLY*]
set_dont_use [get_lib_cells HOLD*]
# If you dont want to use LVT uncomment this line
#set_dont_use [get_lib_cells *LVT*]
####################################################################
### Load Design
#####################################################################
###source verilog_files.tcl
read_hdl "\
${DESIGN}.v"
elaborate $DESIGN
puts "Runtime & Memory after 'read_hdl'"
time_info Elaboration
check_design -unresolved
####################################################################
### Constraints Setup
#####################################################################
read_sdc counter2.sdc
report timing -encounter >> reports/${DESIGN}_pretim.rpt
##############################################################################################
########
### Synthesizing to generic
##############################################################################################
########
syn_generic
puts "Runtime & Memory after 'syn_generic'"
time_info GENERIC
report datapath > reports/${DESIGN}_datapath_generic.rpt
generate_reports -outdir reports -tag generic
write_db -to_file ${DESIGN}_generic.db
report timing -encounter >> reports/${DESIGN}_generic.rpt
##This synthesizes your code
syn_map
## This writes all your files
write -mapped > counter2_synth.v
## THESE FILES ARE NOT REQUIRED, THE SDC FILE IS A TIMING FILE
write_script > script
```

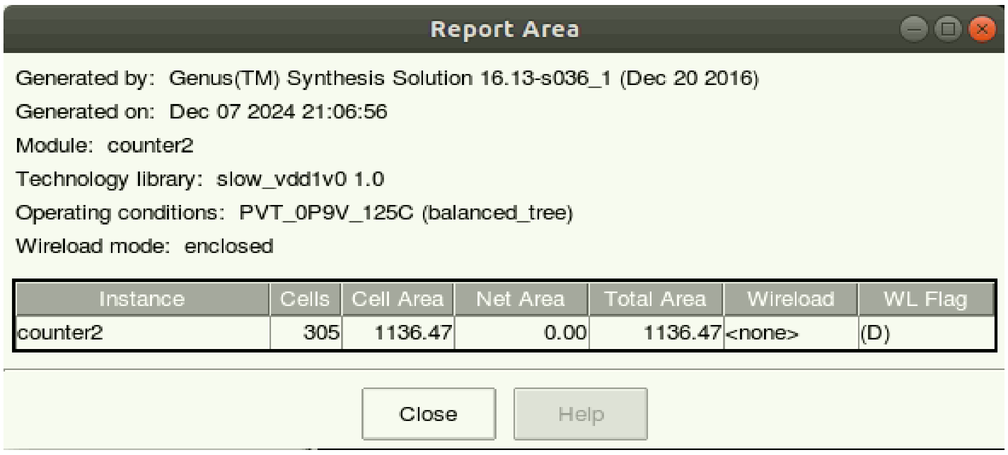


## 6.1 Reports Generated



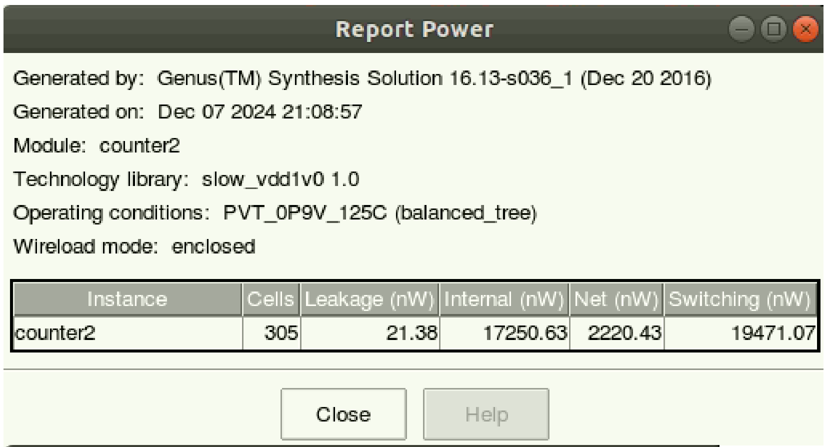
Generated by: Genus(TM) Synthesis Solution 16.13-s036_1 (Dec 20 2016)
Generated on: Dec 07 2024 21:05:01
Module: counter2
Technology library: slow_vdd1v0 1.0
Operating conditions: PVT_0P9V_125C (balanced_tree)
Wireload mode: enclosed

| Gate | Instances | Area | Library |
|---|---|---|---|
| AND2X1 | 1 | 1.37 | slow_vdd1v0 |
| AO22X1 | 2 | 5.47 | slow_vdd1v0 |
| AO22XL | 10 | 27.36 | slow_vdd1v0 |
| AOI21XL | 16 | 27.36 | slow_vdd1v0 |
| AOI22XL | 10 | 20.52 | slow_vdd1v0 |
| AOI32X1 | 1 | 2.39 | slow_vdd1v0 |
| DFFHQX2 | 6 | 41.04 | slow_vdd1v0 |
| DFFNSRX1 | 33 | 304.72 | slow_vdd1v0 |
| DFFNSRX2 | 3 | 35.91 | slow_vdd1v0 |
| DFFQX1 | 18 | 104.65 | slow_vdd1v0 |
| DFFQX2 | 12 | 77.98 | slow_vdd1v0 |



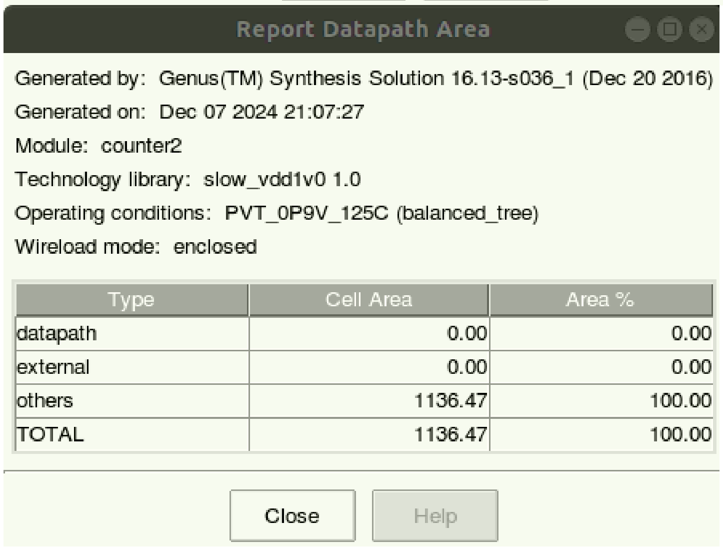
Generated by: Genus(TM) Synthesis Solution 16.13-s036_1 (Dec 20 2016)
Generated on: Dec 07 2024 21:06:56
Module: counter2
Technology library: slow_vdd1v0 1.0
Operating conditions: PVT_0P9V_125C (balanced_tree)
Wireload mode: enclosed

| Instance | Cells | Cell Area | Net Area | Total Area | Wireload | WL Flag |
|---|---|---|---|---|---|---|
| counter2 | 305 | 1136.47 | 0.00 | 1136.47 | <none> | (D) |



Generated by: Genus(TM) Synthesis Solution 16.13-s036_1 (Dec 20 2016)
Generated on: Dec 07 2024 21:08:57
Module: counter2
Technology library: slow_vdd1v0 1.0
Operating conditions: PVT_0P9V_125C (balanced_tree)
Wireload mode: enclosed

| Instance | Cells | Leakage (nW) | Internal (nW) | Net (nW) | Switching (nW) |
|---|---|---|---|---|---|
| counter2 | 305 | 21.38 | 17250.63 | 2220.43 | 19471.07 |

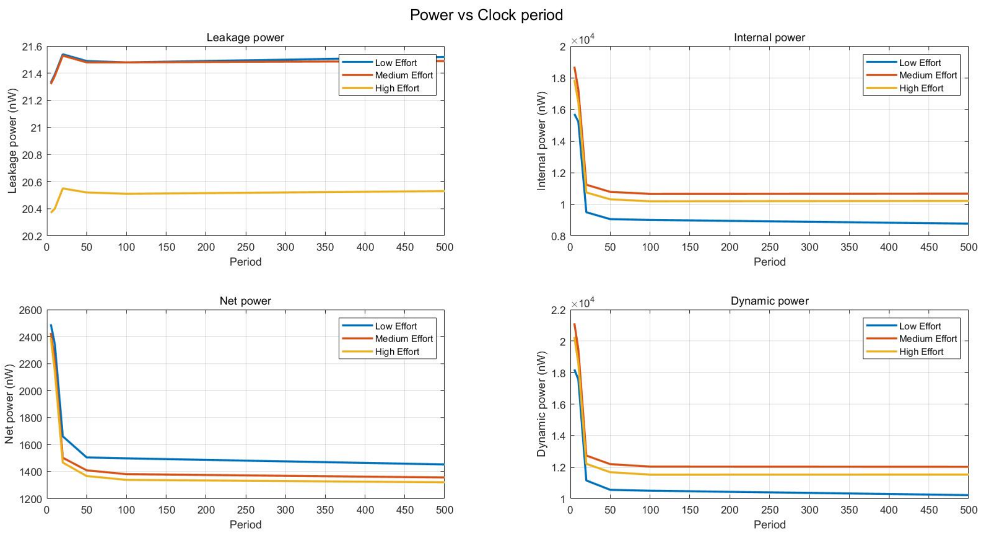


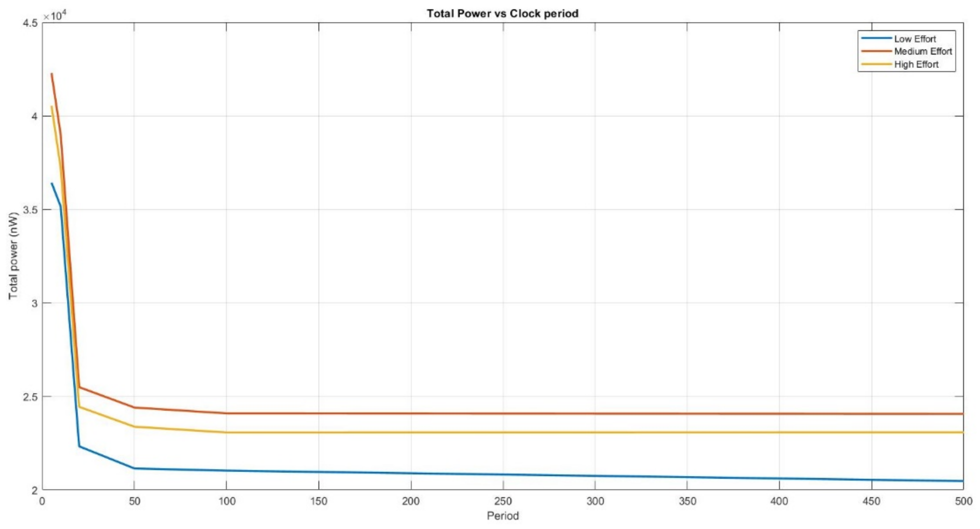
Generated by: Genus(TM) Synthesis Solution 16.13-s036_1 (Dec 20 2016)
Generated on: Dec 07 2024 21:07:27
Module: counter2
Technology library: slow_vdd1v0 1.0
Operating conditions: PVT_0P9V_125C (balanced_tree)
Wireload mode: enclosed

| Type | Cell Area | Area % |
|---|---|---|
| datapath | 0.00 | 0.00 |
| external | 0.00 | 0.00 |
| others | 1136.47 | 100.00 |
| TOTAL | 1136.47 | 100.00 |

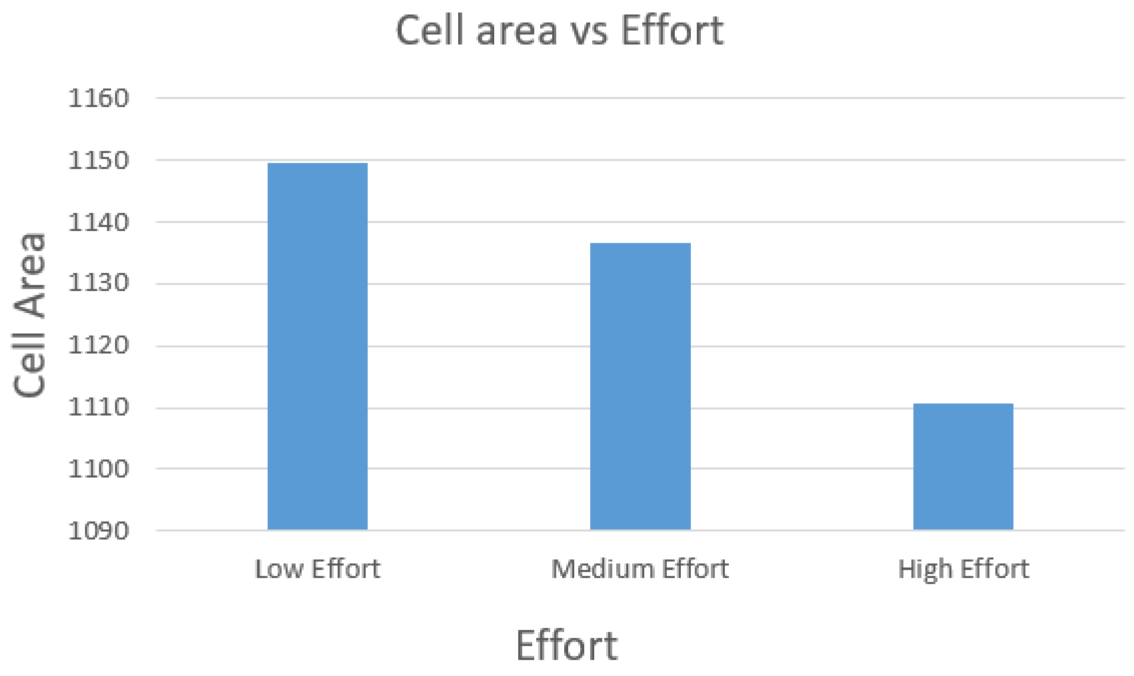
## 6.2 Power and Area Optimization

### Varying Clock Period:





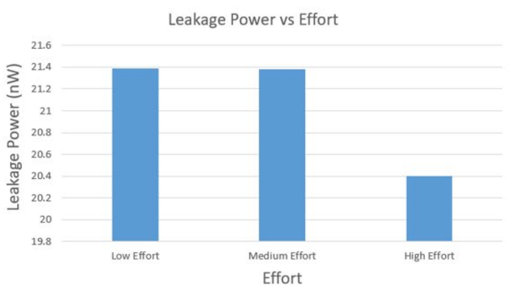
Leakage power increases with an increase in clock period and all other powers (internal, net and dynamic power) decreases exponentially with clock period. Total power (sum of all four powers) decrease exponentially as we increase clock period. Power consumption is highest for MEDIUM effort and lowest for LOW effort. However, the difference in power consumption between MEDIUM and LOW effort is around 0.04mW, and the difference in power consumption between HIGH and LOW effort is around 0.02mW.

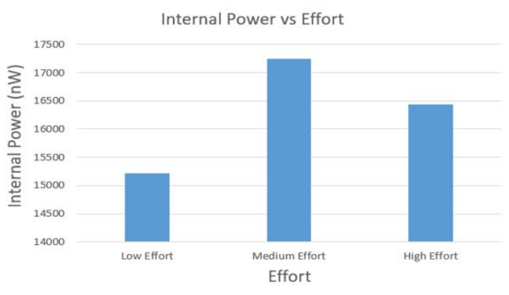


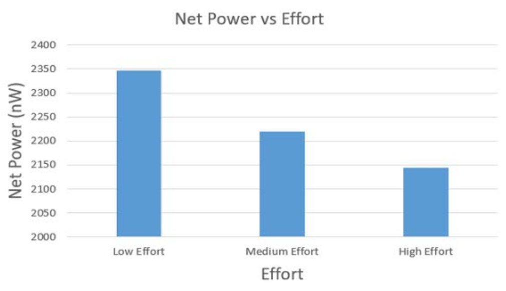
Cell area remains constant as we change clock period. Area only changes when effort level is changed. Variation of power in LOW and HIGH effort is 0.02 mW, but cell area is very high in Low effort. Here, we observed a power-area trade-off.

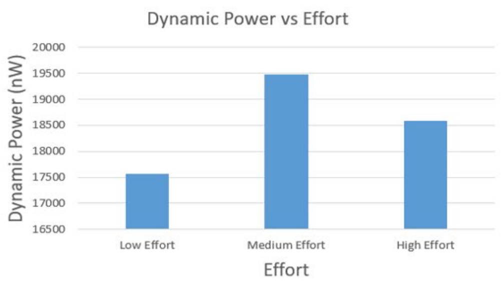
### Varying Setup Time/Hold Time/ Input and Output Delay (Clock period = 10ns):

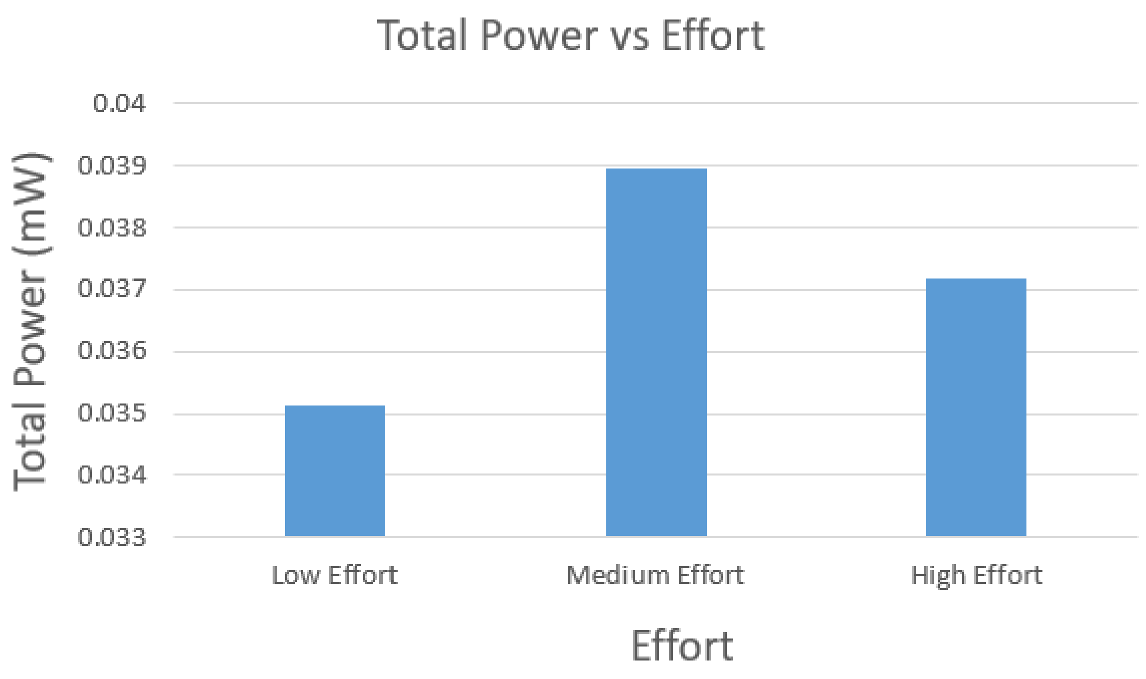
Power and Area do not change with respect to a change in the mentioned parameters. However, different effort gives different values of Power and area.



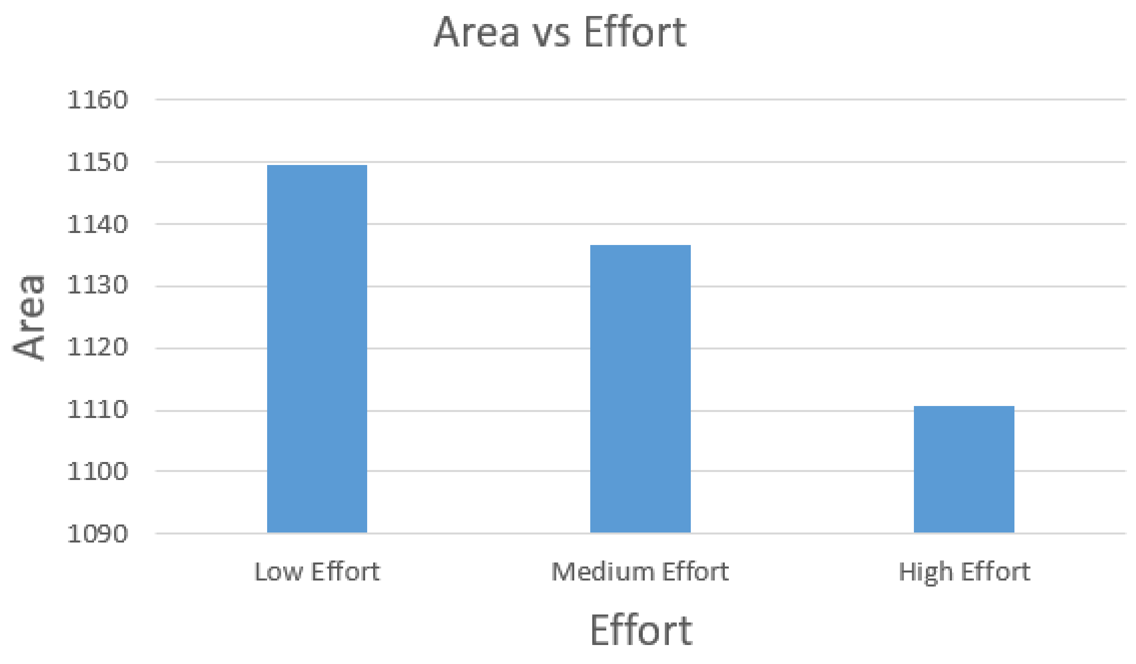






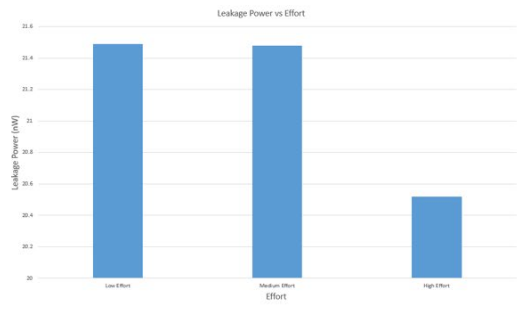


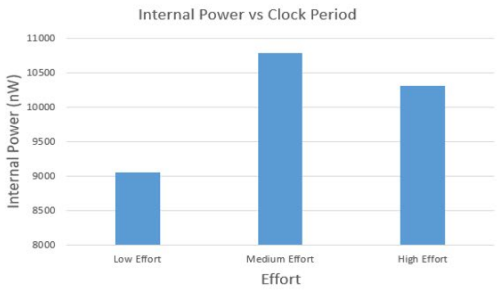
Total Power vs Effort
Total Power (mW)
0.04
0.039
0.038
0.037
0.036
0.035
0.034
0.033
Low Effort
Medium Effort
High Effort
Effort

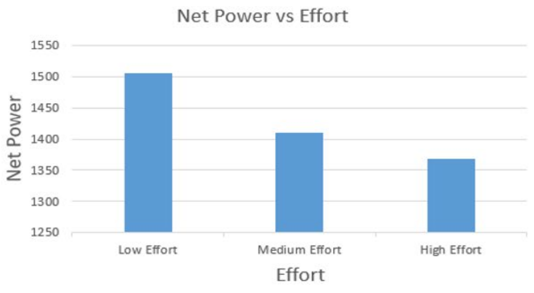


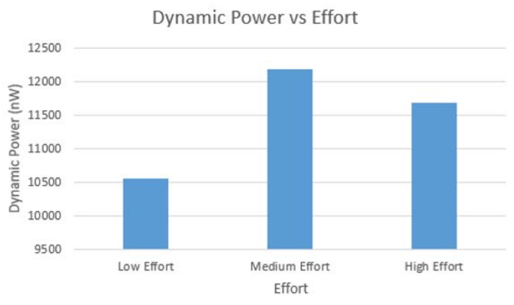
Area vs Effort
Area
1160
1150
1140
1130
1120
1110
1100
1090
Low Effort
Medium Effort
High Effort
Effort

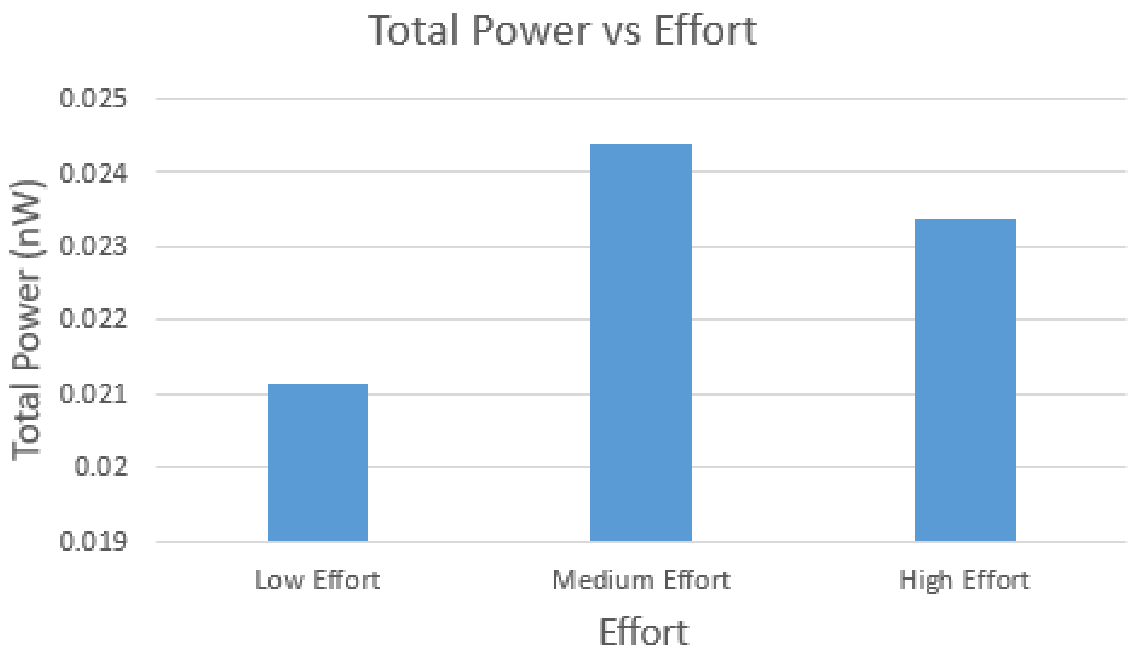
**Varying Setup Time/Hold Time/ Input and Output Delay (Clock period = 50ns):**











**Remarks**

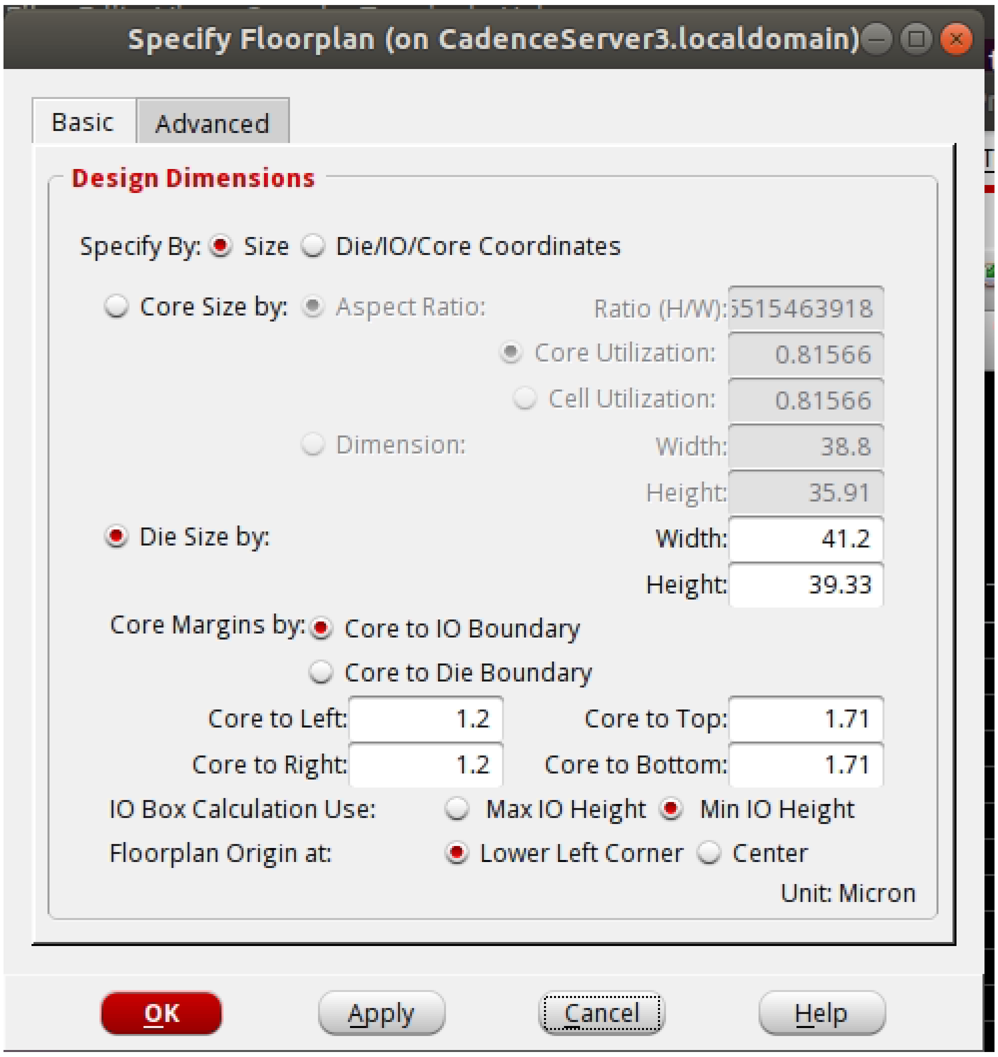
1. Low effort has lower power consumption.
2. However low effort requires highest cell area.
3. We can select medium or high effort since power consumption is not much higher (a few µW), which allows us to reduce cell area.
4. Power consumption shows exponentially decaying behavior with time period. We can choose clock period = 10ns (Power consumption curve almost saturates from this point).

**Final Optimized Design:**

| Parameters | Designed Value |
|---|---|
| Clock Period | 50ns |
| Setup Time | 3ns (Any) |
| Hold Time | 2ns (Any) |
| Input Delay | 0.4ns (Any) |
| Output Delay | 0.6ns (Any) |

# 7 Physical Design (PnR)

## 7.1 Default Layout (Area – 41.2 x 39.33, Spacing – 0.84, Set-to-set distance - 2)



**Design Dimensions**

Specify By: [✓ Size] Die/IO/Core Coordinates

Core Size by: [✓ Aspect Ratio:] Ratio (H/W): 5515463918

[✓ Core Utilization:] 0.81566

Cell Utilization: 0.81566

Dimension: Width: 38.8

Height: 35.91

[✓ Die Size by:] Width: 41.2

Height: 39.33

Core Margins by: [✓ Core to IO Boundary] Core to Die Boundary

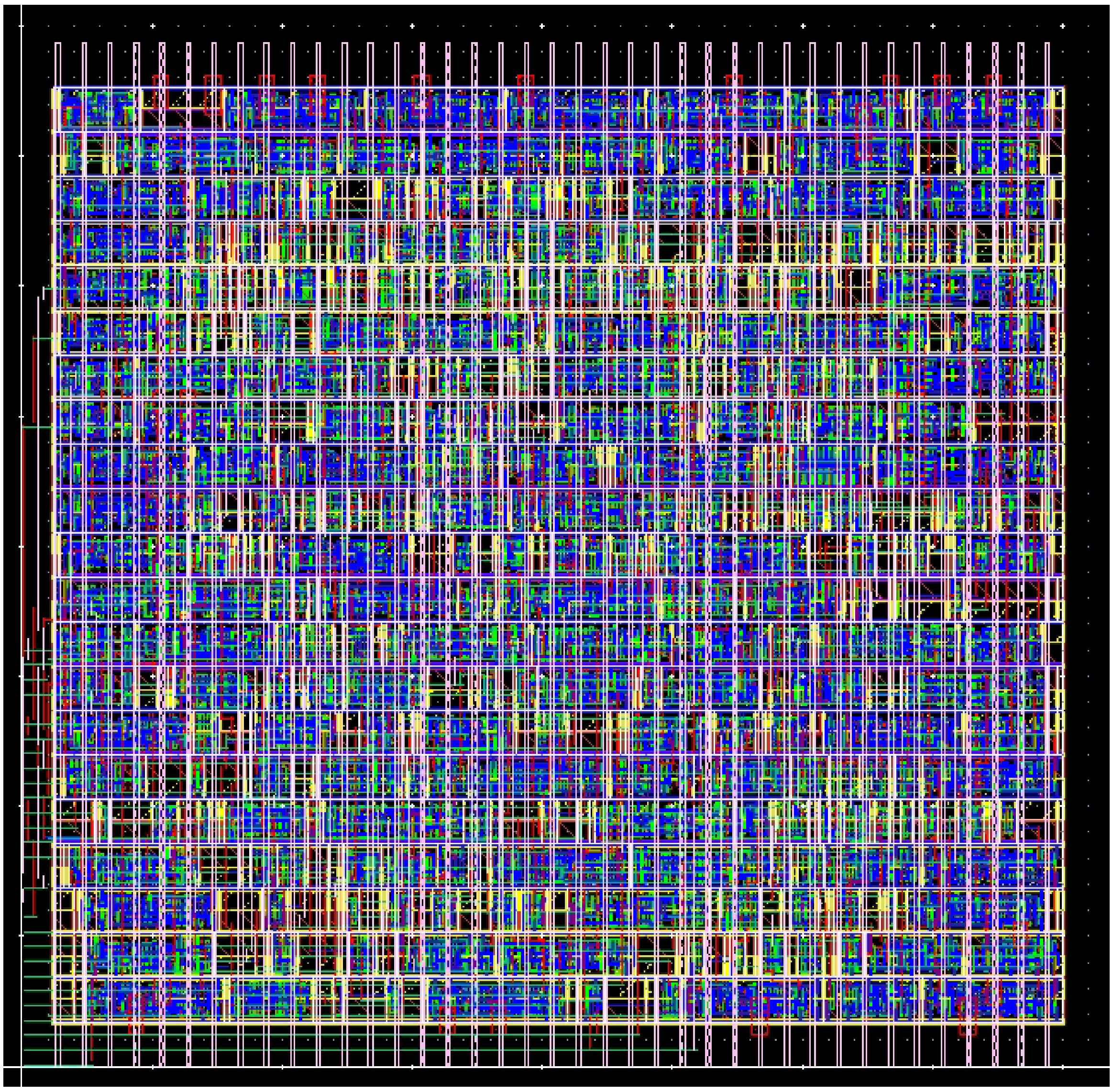
Core to Left: 1.2 Core to Top: 1.71

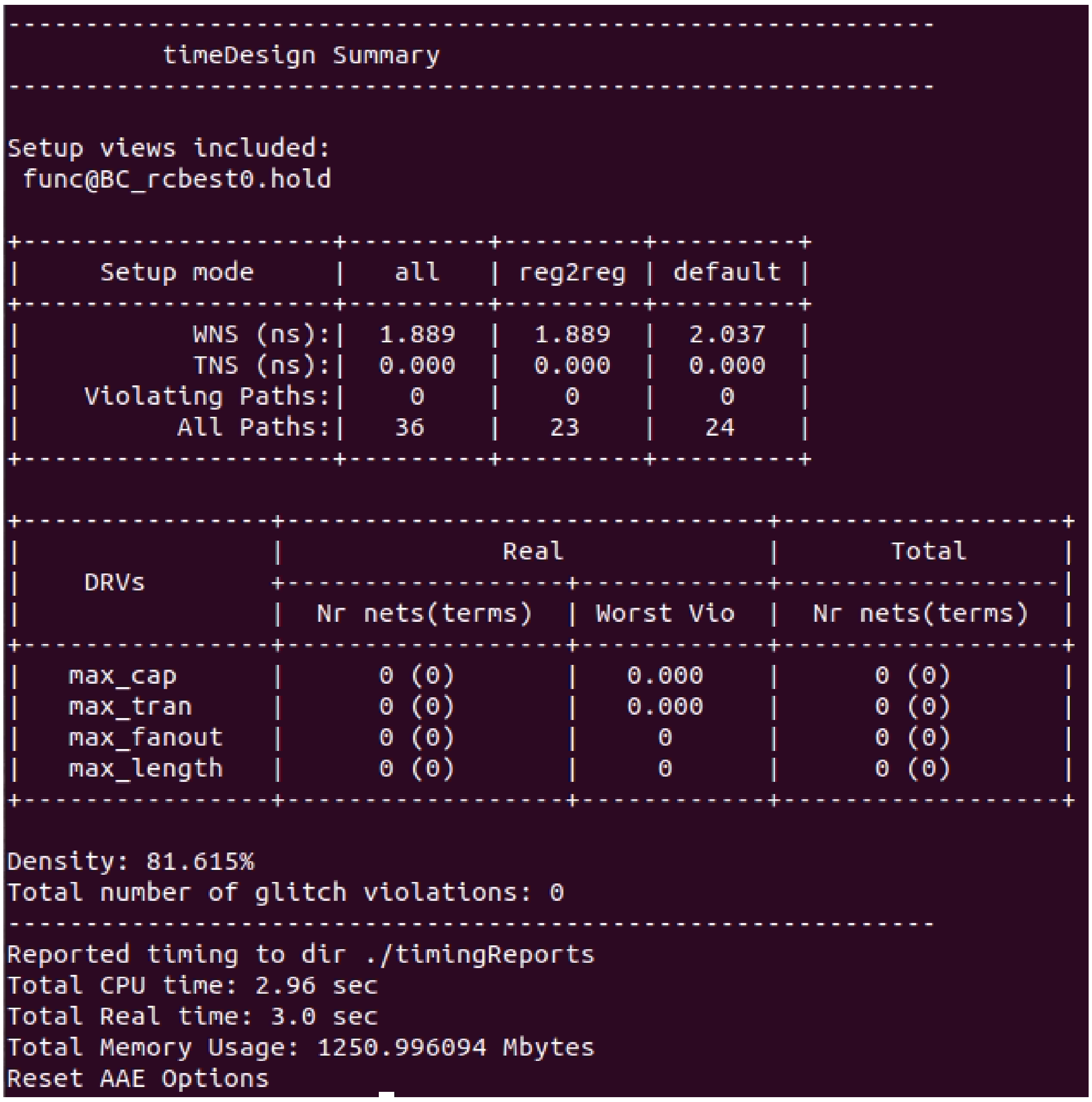
Core to Right: 1.2 Core to Bottom: 1.71

IO Box Calculation Use: Max IO Height [✓ Min IO Height]

Floorplan Origin at: [✓ Lower Left Corner] Center

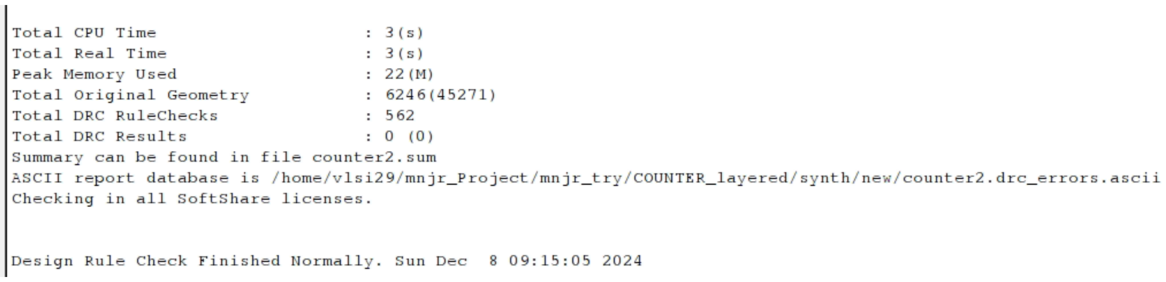
Unit: Micron





```
------------------------------------------------------------
          timeDesign Summary
------------------------------------------------------------

Setup views included:
 func@BC_rcbest0.hold

+--------------------+---------+---------+---------+
|     Setup mode     |   all   | reg2reg | default |
+--------------------+---------+---------+---------+
|           WNS (ns):|  1.889  |  1.889  |  2.037  |
|           TNS (ns):|  0.000  |  0.000  |  0.000  |
|    Violating Paths:|    0    |    0    |    0    |
|          All Paths:|   36    |   23    |   24    |
+--------------------+---------+---------+---------+

+----------------+-------------------------------+------------------+
|                |              Real             |       Total      |
|    DRVs        +------------------+------------+------------------|
|                |  Nr nets(terms)  | Worst Vio  |  Nr nets(terms)  |
+----------------+------------------+------------+------------------+
|   max_cap      |      0 (0)       |   0.000    |      0 (0)       |
|   max_tran     |      0 (0)       |   0.000    |      0 (0)       |
|   max_fanout   |      0 (0)       |     0      |      0 (0)       |
|   max_length   |      0 (0)       |     0      |      0 (0)       |
+----------------+------------------+------------+------------------+

Density: 81.615%
Total number of glitch violations: 0
------------------------------------------------------------
Reported timing to dir ./timingReports
Total CPU time: 2.96 sec
Total Real time: 3.0 sec
Total Memory Usage: 1250.996094 Mbytes
Reset AAE Options
```

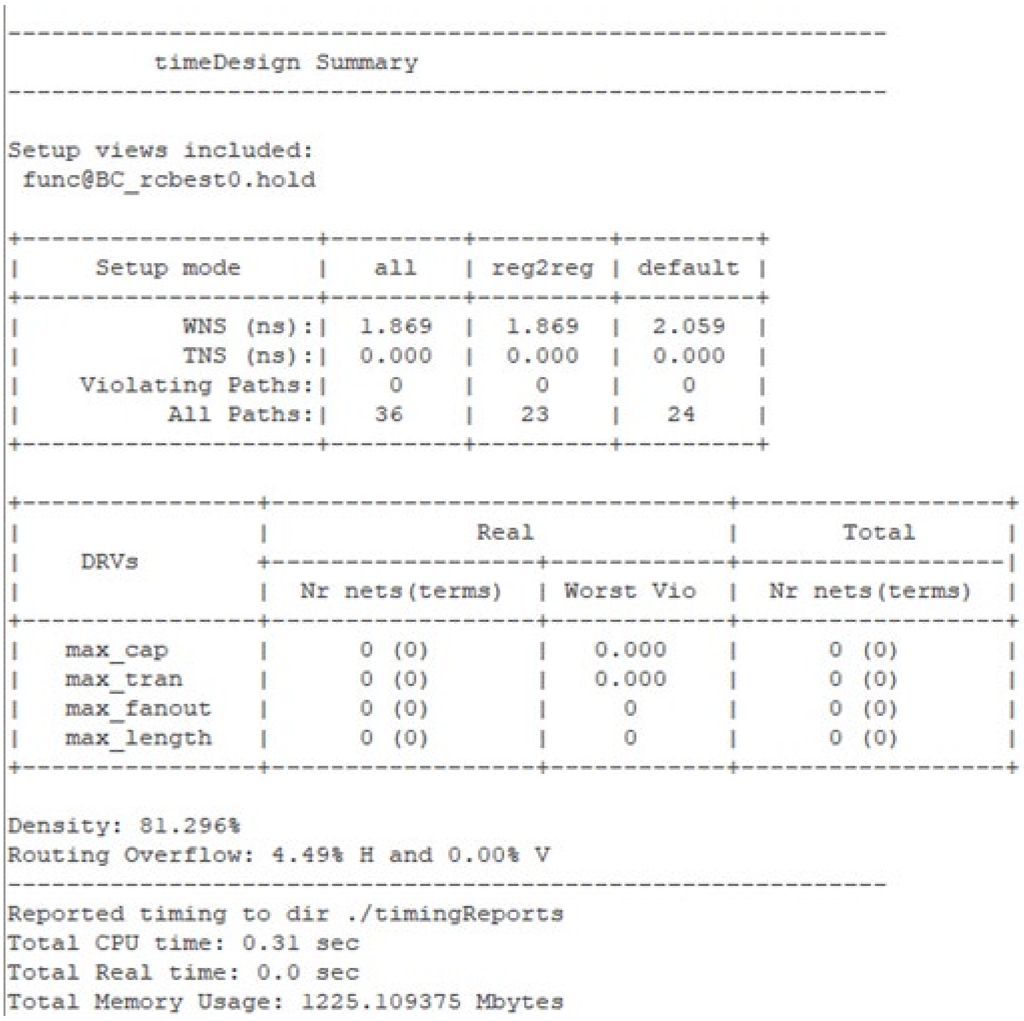
Density Acquired: 81.615 %



```
Total CPU Time                    : 3(s)
Total Real Time                   : 3(s)
Peak Memory Used                  : 22(M)
Total Original Geometry           : 6246(45271)
Total DRC RuleChecks              : 562
Total DRC Results                 : 0 (0)
Summary can be found in file counter2.sum
ASCII report database is /home/vlsi29/mnjr_Project/mnjr_try/COUNTER_layered/synth/new/counter2.drc_errors.ascii
Checking in all SoftShare licenses.


Design Rule Check Finished Normally. Sun Dec  8 09:15:05 2024
```

0 DRC errors found.

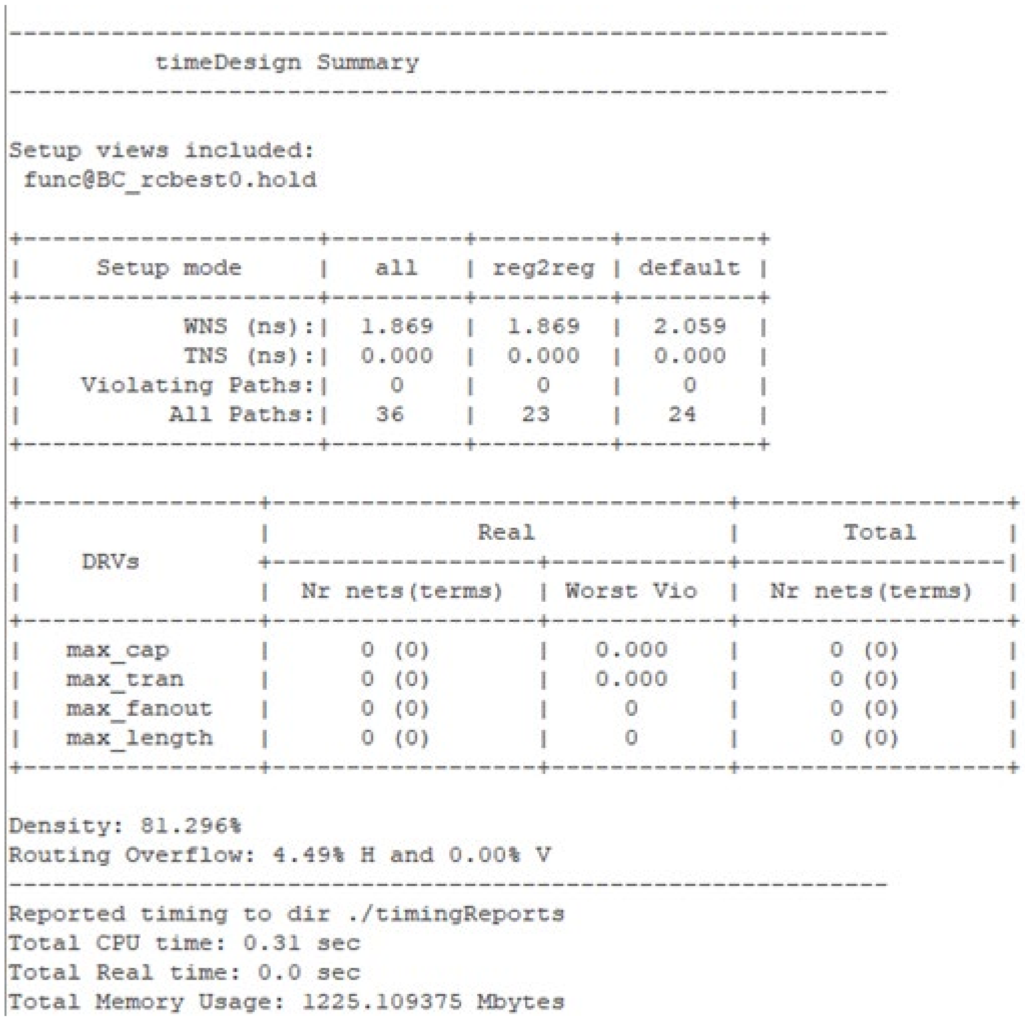
## 7.2 Layout 2 (Area – 41.37 x 39.33, Spacing – 0.15, Set-to-set distance – 2)



```
------------------------------------------------------------
          timeDesign Summary
------------------------------------------------------------

Setup views included:
 func@BC_rcbest0.hold

+--------------------+---------+---------+---------+
|     Setup mode     |   all   | reg2reg | default |
+--------------------+---------+---------+---------+
|           WNS (ns):|  1.869  |  1.869  |  2.059  |
|           TNS (ns):|  0.000  |  0.000  |  0.000  |
|    Violating Paths:|    0    |    0    |    0    |
|          All Paths:|   36    |   23    |   24    |
+--------------------+---------+---------+---------+

+----------------+-------------------------------+------------------+
|                |              Real             |       Total      |
|    DRVs        +------------------+------------+------------------|
|                |  Nr nets(terms)  | Worst Vio  |  Nr nets(terms)  |
+----------------+------------------+------------+------------------+
|   max_cap      |      0 (0)       |   0.000    |      0 (0)       |
|   max_tran     |      0 (0)       |   0.000    |      0 (0)       |
|   max_fanout   |      0 (0)       |     0      |      0 (0)       |
|   max_length   |      0 (0)       |     0      |      0 (0)       |
+----------------+------------------+------------+------------------+

Density: 81.296%
Routing Overflow: 4.49% H and 0.00% V
------------------------------------------------------------
Reported timing to dir ./timingReports
Total CPU time: 0.31 sec
Total Real time: 0.0 sec
Total Memory Usage: 1225.109375 Mbytes
```

Density Acquired: 81.296 %

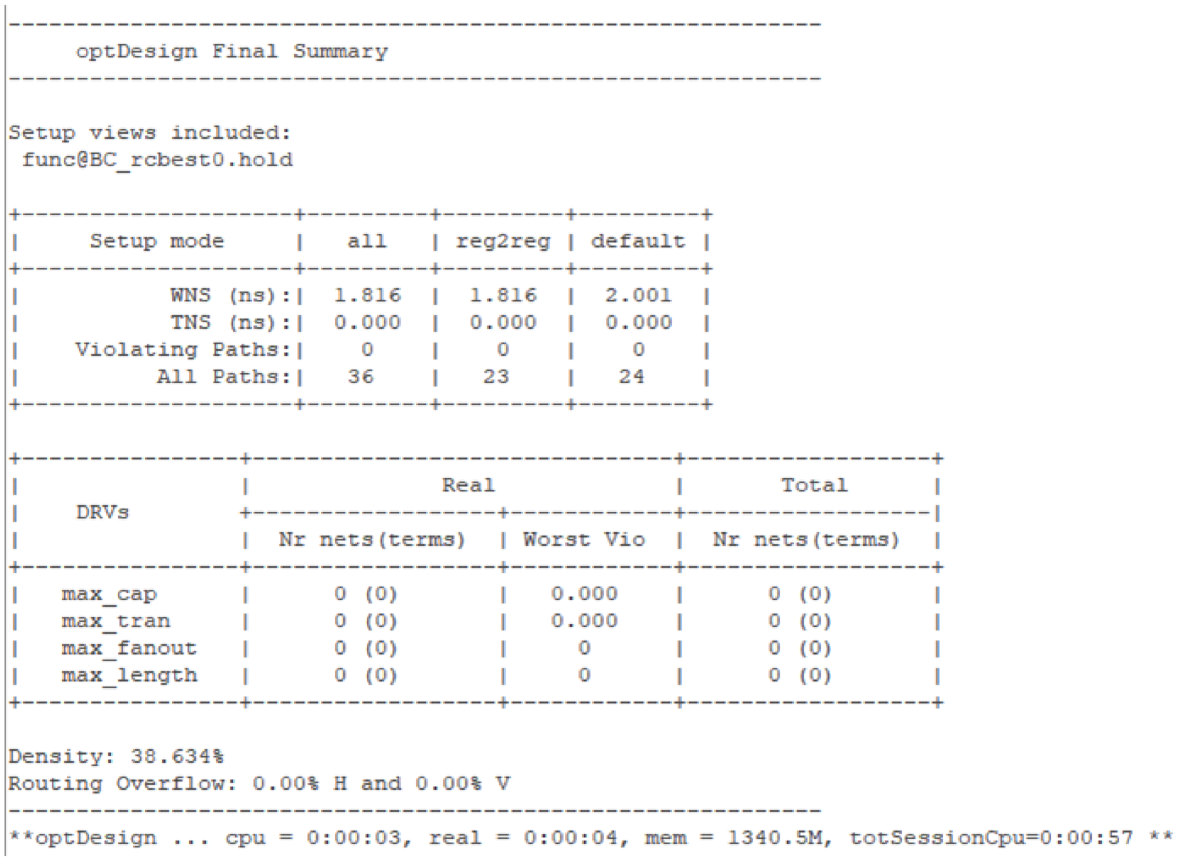
## 7.3 Layout 3 (Area – 41.37 x 39.33, Spacing – 0.6, Set-to-set distance – 2)



```
------------------------------------------------------------
          timeDesign Summary
------------------------------------------------------------

Setup views included:
 func@BC_rcbest0.hold

+--------------------+---------+---------+---------+
|     Setup mode     |   all   | reg2reg | default |
+--------------------+---------+---------+---------+
|           WNS (ns):|  1.869  |  1.869  |  2.059  |
|           TNS (ns):|  0.000  |  0.000  |  0.000  |
|    Violating Paths:|    0    |    0    |    0    |
|          All Paths:|   36    |   23    |   24    |
+--------------------+---------+---------+---------+

+----------------+-------------------------------+------------------+
|                |              Real             |       Total      |
|    DRVs        +------------------+------------+------------------|
|                |  Nr nets(terms)  | Worst Vio  |  Nr nets(terms)  |
+----------------+------------------+------------+------------------+
|   max_cap      |      0 (0)       |   0.000    |      0 (0)       |
|   max_tran     |      0 (0)       |   0.000    |      0 (0)       |
|   max_fanout   |      0 (0)       |     0      |      0 (0)       |
|   max_length   |      0 (0)       |     0      |      0 (0)       |
+----------------+------------------+------------+------------------+

Density: 81.296%
Routing Overflow: 4.49% H and 0.00% V
------------------------------------------------------------
Reported timing to dir ./timingReports
Total CPU time: 0.31 sec
Total Real time: 0.0 sec
Total Memory Usage: 1225.109375 Mbytes
```

Density Acquired: 81.296 %

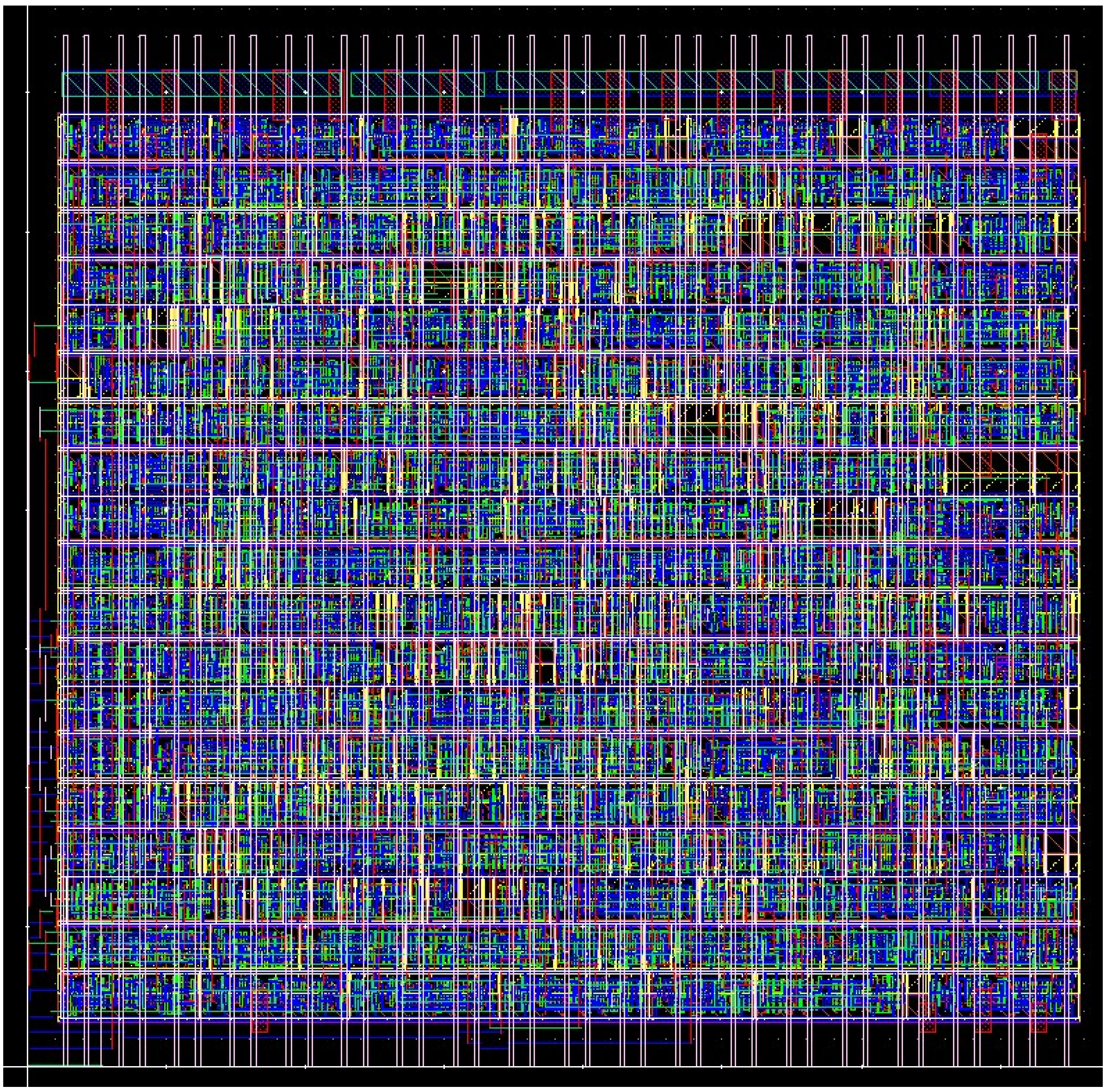
## 7.4 Layout 4 (Area – 60 x 50, Spacing – 0.84, Set-to-set distance – 2)

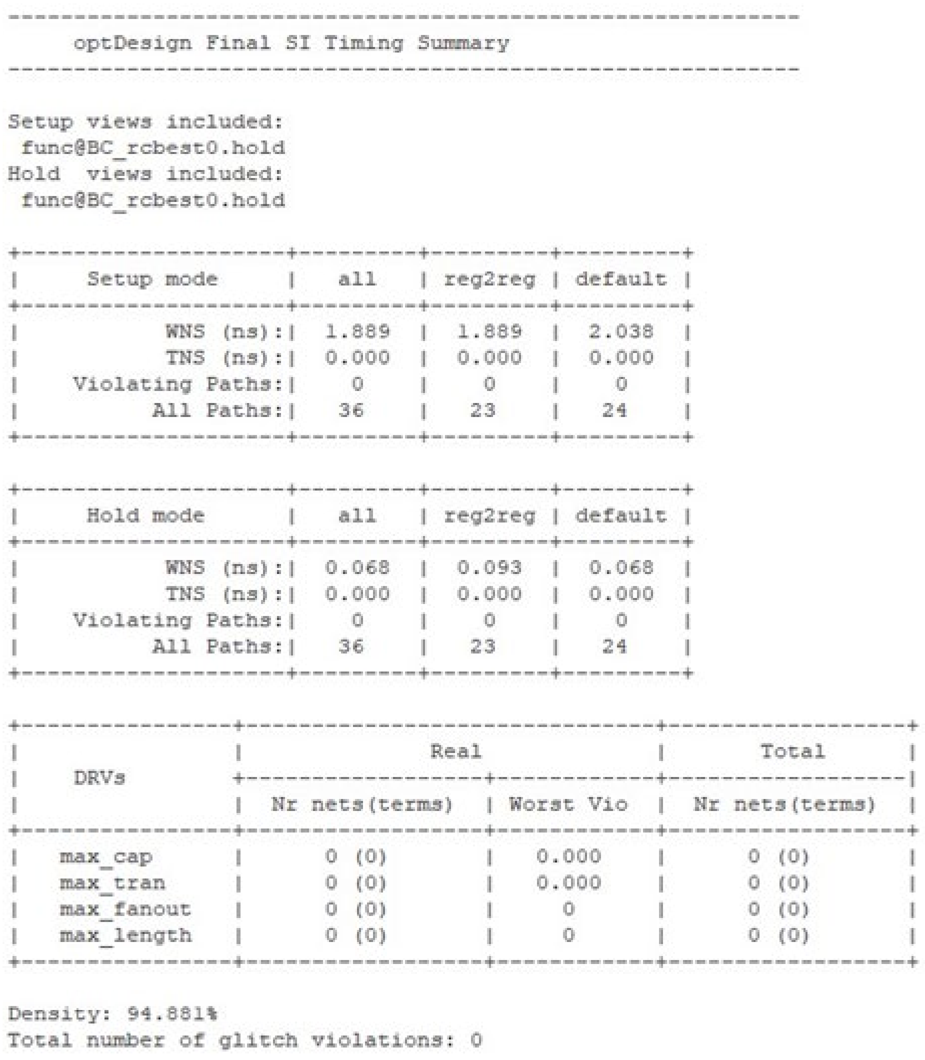


```
------------------------------------------------------------
     optDesign Final Summary
------------------------------------------------------------

Setup views included:
 func@BC_rcbest0.hold

+--------------------+---------+---------+---------+
|     Setup mode     |   all   | reg2reg | default |
+--------------------+---------+---------+---------+
|           WNS (ns):|  1.816  |  1.816  |  2.001  |
|           TNS (ns):|  0.000  |  0.000  |  0.000  |
|    Violating Paths:|    0    |    0    |    0    |
|          All Paths:|   36    |   23    |   24    |
+--------------------+---------+---------+---------+

+----------------+-------------------------------+------------------+
|                |              Real             |       Total      |
|    DRVs        +------------------+------------+------------------|
|                |  Nr nets(terms)  | Worst Vio  |  Nr nets(terms)  |
+----------------+------------------+------------+------------------+
|   max_cap      |      0 (0)       |   0.000    |      0 (0)       |
|   max_tran     |      0 (0)       |   0.000    |      0 (0)       |
|   max_fanout   |      0 (0)       |     0      |      0 (0)       |
|   max_length   |      0 (0)       |     0      |      0 (0)       |
+----------------+------------------+------------+------------------+

Density: 38.634%
Routing Overflow: 0.00% H and 0.00% V
------------------------------------------------------------
**optDesign ... cpu = 0:00:03, real = 0:00:04, mem = 1340.5M, totSessionCpu=0:00:57 **
```

Density Acquired: 38.634 %

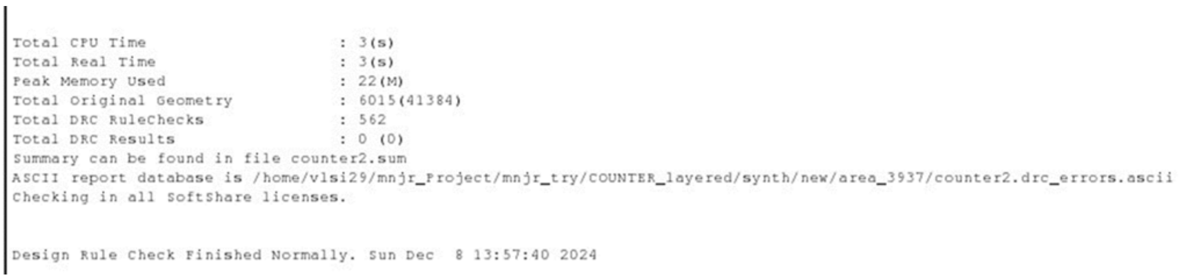
## 7.5 Final Layout (Area – 39 x 37, Spacing – 0.6, Set-to-set distance – 2)





```
------------------------------------------------------------
     optDesign Final SI Timing Summary
------------------------------------------------------------

Setup views included:
 func@BC_rcbest0.hold
Hold  views included:
 func@BC_rcbest0.hold

+--------------------+---------+---------+---------+
|     Setup mode     |   all   | reg2reg | default |
+--------------------+---------+---------+---------+
|           WNS (ns):|  1.889  |  1.889  |  2.038  |
|           TNS (ns):|  0.000  |  0.000  |  0.000  |
|    Violating Paths:|    0    |    0    |    0    |
|          All Paths:|   36    |   23    |   24    |
+--------------------+---------+---------+---------+

+--------------------+---------+---------+---------+
|     Hold mode      |   all   | reg2reg | default |
+--------------------+---------+---------+---------+
|           WNS (ns):|  0.068  |  0.093  |  0.068  |
|           TNS (ns):|  0.000  |  0.000  |  0.000  |
|    Violating Paths:|    0    |    0    |    0    |
|          All Paths:|   36    |   23    |   24    |
+--------------------+---------+---------+---------+

+----------------+-------------------------------+------------------+
|                |              Real             |       Total      |
|    DRVs        +------------------+------------+------------------|
|                | Nr nets(terms)   | Worst Vio  | Nr nets(terms)   |
+----------------+------------------+------------+------------------+
|   max_cap      |      0 (0)       |   0.000    |      0 (0)       |
|   max_tran     |      0 (0)       |   0.000    |      0 (0)       |
|   max_fanout   |      0 (0)       |     0      |      0 (0)       |
|   max_length   |      0 (0)       |     0      |      0 (0)       |
+----------------+------------------+------------+------------------+

Density: 94.881%
Total number of glitch violations: 0
```

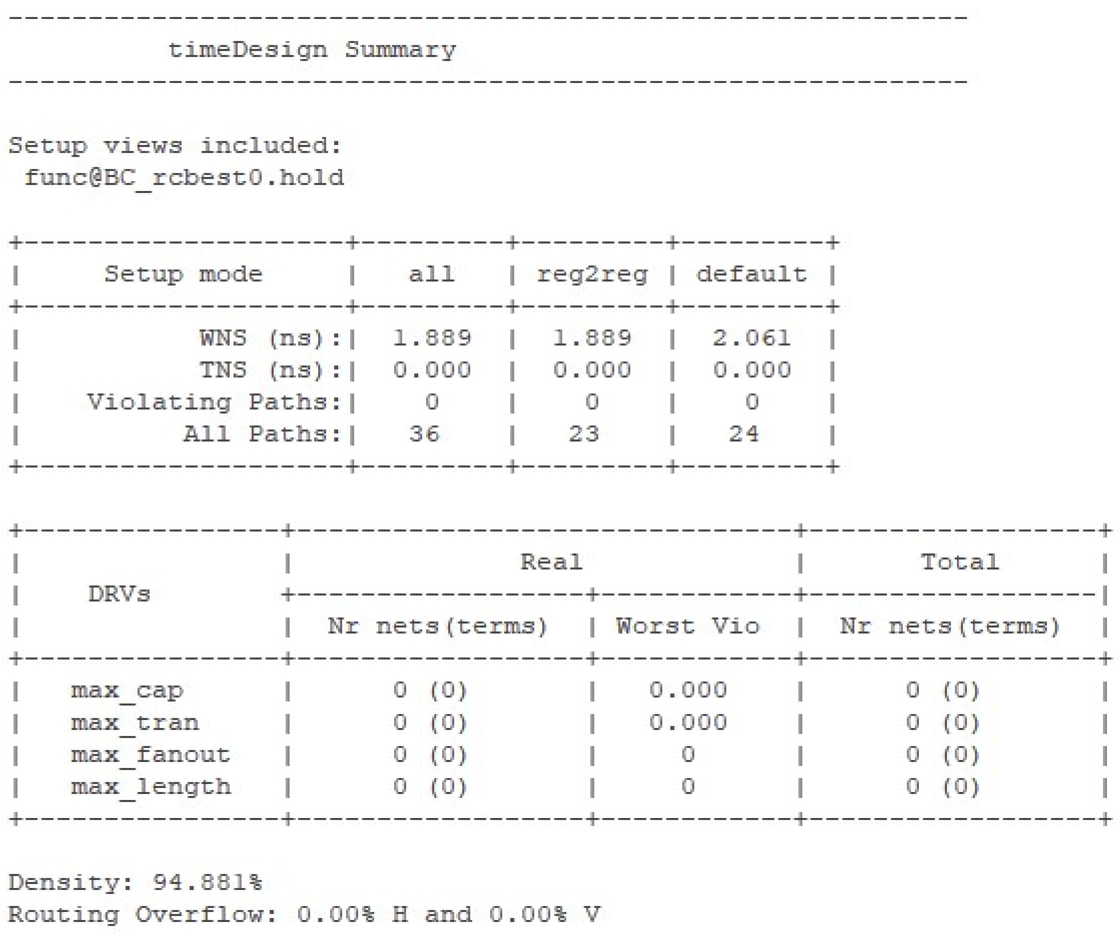
Density Acquired: 94.881 %



```
Total CPU Time                    : 3(s)
Total Real Time                   : 3(s)
Peak Memory Used                  : 22(M)
Total Original Geometry           : 6015(41384)
Total DRC RuleChecks              : 562
Total DRC Results                 : 0 (0)
Summary can be found in file counter2.sum
ASCII report database is /home/vlsi29/mnjr_Project/mnjr_try/COUNTER_layered/synth/new/area_3937/counter2.drc_errors.ascii
Checking in all SoftShare licenses.


Design Rule Check Finished Normally. Sun Dec  8 13:57:40 2024
```

0 DRC errors found.

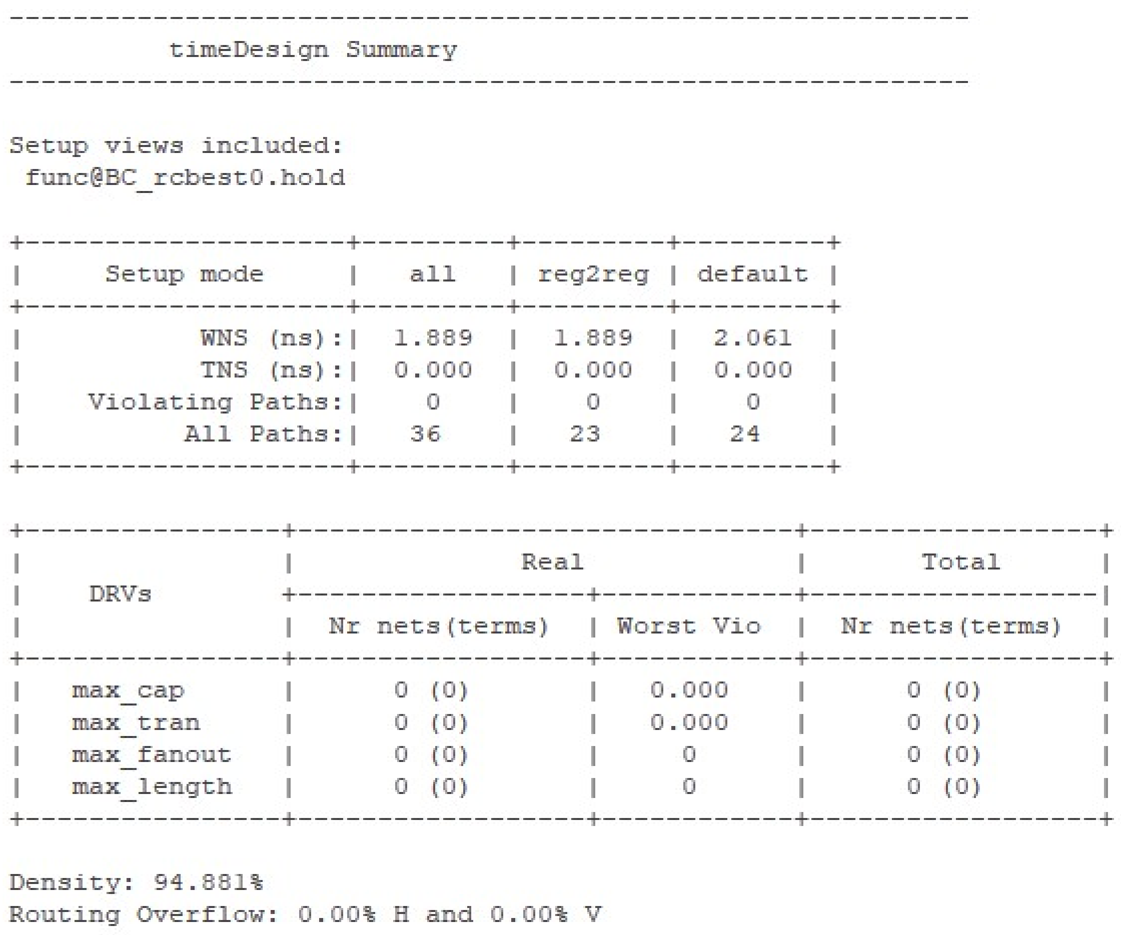
## 7.6 Final Layout (Area – 39 x 37, Spacing – 0.6, Set-to-set distance – 1)



```
------------------------------------------------------------
         timeDesign Summary
------------------------------------------------------------

Setup views included:
 func@BC_rcbest0.hold

+--------------------+---------+---------+---------+
|     Setup mode     |   all   | reg2reg | default |
+--------------------+---------+---------+---------+
|           WNS (ns):|  1.889  |  1.889  |  2.061  |
|           TNS (ns):|  0.000  |  0.000  |  0.000  |
|    Violating Paths:|    0    |    0    |    0    |
|          All Paths:|   36    |   23    |   24    |
+--------------------+---------+---------+---------+

+----------------+-------------------------------+------------------+
|                |              Real             |       Total      |
|    DRVs        +------------------+------------+------------------|
|                |  Nr nets(terms)  | Worst Vio  |  Nr nets(terms)  |
+----------------+------------------+------------+------------------+
|   max_cap      |    0 (0)         |   0.000    |      0 (0)       |
|   max_tran     |    0 (0)         |   0.000    |      0 (0)       |
|   max_fanout   |    0 (0)         |     0      |      0 (0)       |
|   max_length   |    0 (0)         |     0      |      0 (0)       |
+----------------+------------------+------------+------------------+

Density: 94.881%
Routing Overflow: 0.00% H and 0.00% V
```

Density Acquired: 94.881 %

## 7.7 Final Layout (Area – 39 x 37, Spacing – 0.6, Set-to-set distance – 3)



```
------------------------------------------------------------
          timeDesign Summary
------------------------------------------------------------

Setup views included:
 func@BC_rcbest0.hold

+--------------------+---------+---------+---------+
|     Setup mode     |   all   | reg2reg | default |
+--------------------+---------+---------+---------+
|           WNS (ns):|  1.889  |  1.889  |  2.061  |
|           TNS (ns):|  0.000  |  0.000  |  0.000  |
|    Violating Paths:|    0    |    0    |    0    |
|          All Paths:|   36    |   23    |   24    |
+--------------------+---------+---------+---------+

+----------------+-------------------------------+------------------+
|                |              Real             |       Total      |
|    DRVs        +------------------+------------+------------------|
|                |  Nr nets(terms)  | Worst Vio  |  Nr nets(terms)  |
+----------------+------------------+------------+------------------+
|   max_cap      |     0 (0)        |   0.000    |     0 (0)        |
|   max_tran     |     0 (0)        |   0.000    |     0 (0)        |
|   max_fanout   |     0 (0)        |     0      |     0 (0)        |
|   max_length   |     0 (0)        |     0      |     0 (0)        |
+----------------+------------------+------------+------------------+

Density: 94.881%
Routing Overflow: 0.00% H and 0.00% V
```

Density Acquired: 94.881 %

## 7.8 Final Optimized Table:

| Parameters | Designed Value |
|---|---|
| Width | 39 |
| Height | 37 |
| Set-to-set distance | Any |
| Spacing | 0.6 |
| Density | 94.881% |

# 8 Design Problems

## 8.1 Multiple inputs transition at the same time

```
module counter(clk, reset, up_down, load, data, count, enable);

  //define input and output ports
  input clk, reset, load, up_down, enable;
  input [3:0] data;
  output reg [3:0] count;

  always@(posedge reset)
    if(enable)
    begin
      count<=0;
    end

  always@(posedge load)
    if(enable)
    begin
      count<=data;
    end

  always@(posedge up_down or negedge up_down)
    if(enable)
    begin
      if(up_down) count<=count+1;
      else count<=count-1;
    end

//always block will be executed at each and every positive edge of the clock

always@(posedge clk)
  if(enable)
  begin
    //if(reset) count <= 0;            //Set Counter to Zero
    //else if(load) count <= data;     //load the counter with data value
    if(up_down) count <= count + 1;    //count up
    else count <= count - 1;           //count down
  end

endmodule
```

If multiple inputs are changed at the same time (for example, reset and load is set to high at same instance), an ambiguity arises. To fix this, we set reset as highest priority and load as second priority.

## 8.2 Reset, Load or Up-Down ON for a long time

```
module counter(clk, reset, up_down, load, data, count, enable);

  //define input and output ports
  input clk, reset, load, up_down, enable;
  input [11:0] data;
  output reg [11:0] count;
  logic r=0;
  logic l=0;
  logic ud=up_down;


  //always block will be executed at each and every positive edge of the clock
  always@(posedge clk or posedge reset or posedge load or posedge up_down or negedge up_down)
  begin
    if(enable)
    begin
      if(reset==1 & r==0)
        begin
          count<=0;
          r=1;
        end
      else if(load==1 & l==0)
        begin
          count<=data;
          l=1;
        end

else if(up_down==1 & ud==0)
        begin
          count<=count+1;
          ud=1;
        end
      else if(up_down==0 & ud==1)
        begin
          count<=count-1;
          ud=0;
        end
      else if(up_down==1)
        begin
          count<=count+1;
          ud=1;
        end

      else if(up_down==0)
        begin
          count<=count-1;
          ud=0;
        end

      if(reset==0) r=0;
      else r=1;
      if(load==0) l=0;
      else l=1;
    end

    else count<=count;

  end
endmodule
```

First, we considered reset and load as a pushbutton switch. If reset is pressed once and kept on for a long time, we will set count to zero at the positive edge of reset and if a clock pulse arrives next, counter will start counting. Later, we fixed this issue by setting count to zero as long as reset is on.

## 8.3 More than one posedge or negedge inputs not allowed in always

```
module counter(clk, reset, up_down, load, data, count, enable);

  //define input and output ports
  input clk, reset, load, up_down, enable;
  input [11:0] data;
  output reg [11:0] count;

  //always block will be executed at each and every positive edge of the clock

  always@(posedge reset)
    begin
      if(!enable) count<=count;
      else count<=0;
    end

  always@(negedge reset)
    begin
      if(!enable) count<=count;
      else if(load) count<=data;
      else if(up_down) count<=count+1;
      else if(!up_down) count<=count-1;
    end

  always@(posedge load)
    begin
      if(!enable) count<=count;
      else if(!reset) count<=data;
    end
  always@(negedge load)
    begin
      if(!enable) count<=count;
      else if(reset) count<=0;
      else if(up_down) count<=count+1;
      else if(!up_down) count<=count-1;
    end

 always@(posedge clk)
    begin
      if(!enable) count<=count;
      else if(!reset & !load)
        begin
          if(up_down==1) count<=count+1;
          else if(up_down==0) count<=count-1;
        end
    end

   always@(posedge up_down)
    begin
      if(!enable) count<=count;
      else if(reset) count<=0;
      else if(load) count<=data;
      else count<=count+1;
    end

   always@(negedge up_down)
    begin
      if(!enable) count<=count;
      else if(reset) count<=0;
      else if(load) count<=data;
      else count<=count-1;
    end

   always@(posedge enable)
    begin
      if(reset) count<=0;
      else if(load) count<=data;
      else if(up_down) count<=count+1;
      else if(!up_down) count<=count-1;
    end

   always@(data)
    if(load & !reset) count<=data;

 endmodule
```

If we keep more than one posedge or negedge of inputs, an error occurs when we synthesize our design.

## 9 Reflection on Individual and Team work

| ID | Task |
|---|---|
| **1906108** | Design Verilog and Directed Testbench |
| **1906128** | Synthesis and Physical Layer Design |
| **1906147** | Layered Testbench |
| **1906150** | PnR Optimization |
| **1906178** | RTL Synthesis Optimization |

## 10 References

1, https://www.allaboutcircuits.com/textbook/digital/chpt-11/asynchronous-counters/

2. Book: Weste N., Harris D. - CMOS VLSI Design: A Circuits and Systems Perspective-Pearson Education (2004)