Experiment-6

Introduction to the synthesis of Digital circuits

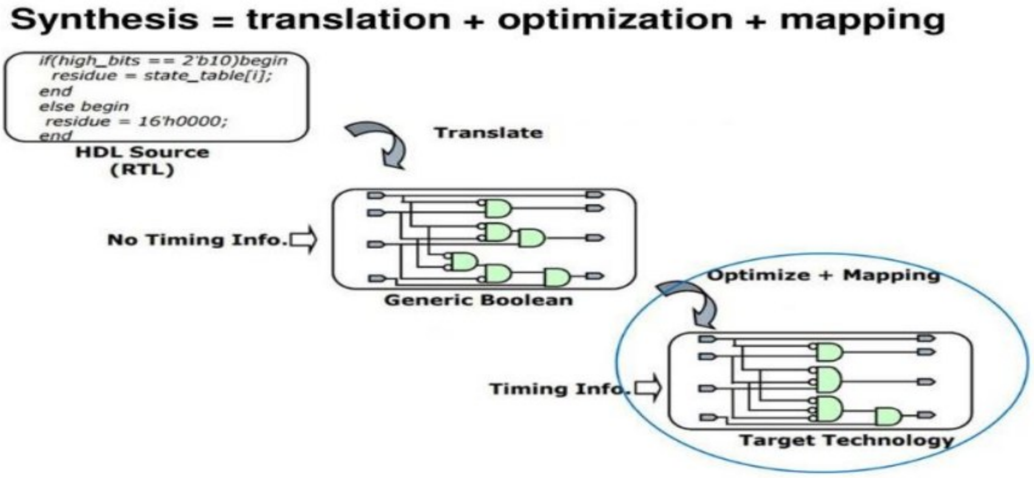
Objective:

The main objectives of this lab are:

- Familiarization with synthesis flow.
- Setting up synthesis constraints.
- Generating optimized gate-level netlist and Standard Design Constraints.

Introduction

Synthesis is a process of transforming RTL (a description of a circuit expressed in a language such as Verilog or VHDL written in behavioral modeling or data flow modeling) to technologydependent or independent gate-level netlist including nets, sequential and combinational cells, and their connectivity. The main goal of synthesis is obtaining a gate-level netlist, logic optimization, inserting a clock-gating cell for power reduction, inserting DFT (Design for Testability) cell, and maintaining the logical equivalence between RTL and gate-level netlist. The best output of place and route depend on the synthesis.



```
if(high_bits == 2'b10)begin
  residue = state_table[i];
end
else begin
 residue = 16'h0000;
end
```

Fig: Steps of Synthesis

Purpose of Logic Circuit Synthesis

- **Automates Design:** Reduces the manual effort required to design circuits by automating the generation of gate-level implementations.
- **Optimization:** Ensures the design meets specifications for performance, power consumption, and area (PPA metrics).
- **Bridges Abstraction Levels**: Converts a functional specification into hardware-level representations.

Steps in Logic Circuit Synthesis

- **High-Level Design Description:**
  - Starts with a functional specification in a **hardware description language (HDL)** such as **Verilog** or **VHDL**.
  - Example: Writing if-else conditions, loops, or operations at a behavioral level.
- **Translation:**
  - The HDL code is converted into an **intermediate representation** (IR) or **Boolean expressions**.
- **Logic Optimization:**
  - Simplifies Boolean expressions using optimization techniques:
    - Reducing the number of gates.
    - Eliminating redundant paths.
  - Ensures the circuit adheres to constraints like timing, area, and power.
- **Technology Mapping:**
  - Maps the optimized Boolean expressions to the **technology library** of standard cells (pre-designed logic gates like AND, OR, NOT, multiplexers, etc.).
  - Each standard cell is characterized by a specific process technology node (e.g., 7nm, 28nm).
- **Generation of Gate-Level Netlist:**
  - Produces a **netlist** that describes the connectivity between gates.

Types of Synthesis

- **Behavioral Synthesis:**
  - Converts high-level algorithms into a circuit at the RTL (Register Transfer Level).
- **Logic Synthesis:**
  - Converts RTL descriptions into a **gate-level design**.
- **Physical Synthesis:**
  - Prepares the gate-level netlist for layout by considering physical constraints such as placement and routing.

Key Components of Synthesis

- **Technology Library:**
  - Contains predefined logic cells, characterized for performance, power, and area.
- **Constraints File:**

        - Specifies design constraints such as:
        - Maximum delay (timing constraints).
        - Target area or power limits.
- **Optimization Objectives:**
    - Ensure the design is:
        - **Fast** (meets timing).
        - **Small** (uses minimal chip area).
        - **Power-efficient** (reduces energy consumption)

Example:

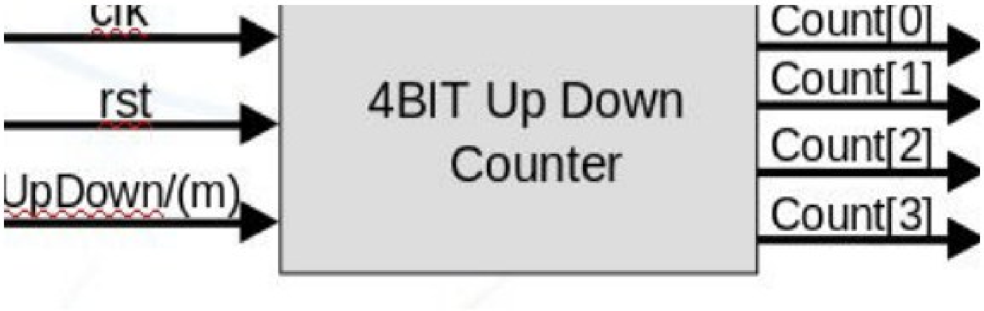
To write a Verilog code for 4bit up/down asynchronous rest counter and its test-bench for verification and do the synthesizing.

Tool Required:

- Functional Simulation: Incisive Simulator (ncvlog, ncelab, ncsim).
- Synthesis: Genus

Design Information and Bock Diagram:

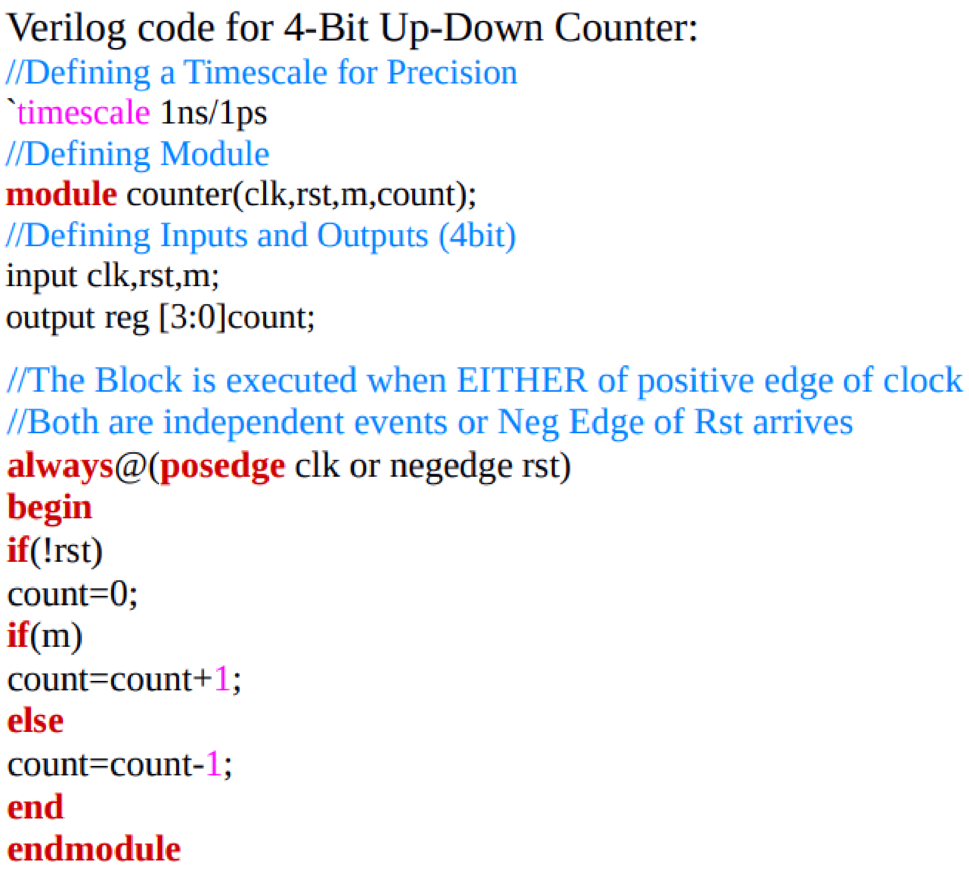
- An up/down counter is a digital counter which can be set to count either from 0 to MAX_VALUE or MAX_VALUE to 0.
- The direction of the count(mode) is selected using a single bit input. The module has 3 inputs - clk, reset which is active high and an Up Or Down mode input. The output is Counter which is 4 bits in size.
- When Up mode is selected, counter counts from 0 to 15 and then again from 0 to 15.
- When Down mode is selected, the counter counts from 15 to 0 and then again from 15 to 0.
- Changing mode doesn't reset the Count value to zero.
- You must apply high value to reset, to reset the Counter output.



You must perform the function simulation using NCLaunch (Already discuss in previous experiments). This involves three stages :

- Step 1: Compilation- Process to check the correct Verilog language syntax and usage
- Step 2: Elaboration- To check the port connections in hierarchical design
- Step 3: Simulation- Simulate with the given test vectors over a period of time to observe the output behavior.

Verilog code for 4-Bit Up-Down Counter:



Verilog code for 4-Bit Up-Down Counter:

```
//Defining a Timescale for Precision
`timescale 1ns/1ps
//Defining Module
module counter(clk,rst,m,count);
//Defining Inputs and Outputs (4bit)
input clk,rst,m;
output reg [3:0]count;

//The Block is executed when EITHER of positive edge of clock
//Both are independent events or Neg Edge of Rst arrives
always@(posedge clk or negedge rst)
begin
if(!rst)
count=0;
if(m)
count=count+1;
else
count=count-1;
end
endmodule
```

Creating Test Bench:

- Similarly, create your test bench using gedit .v or .vhdl to open a new blank document (4bup_down_count_tb.v).

Test-bench code for 4-Bit Up-Down Counter:

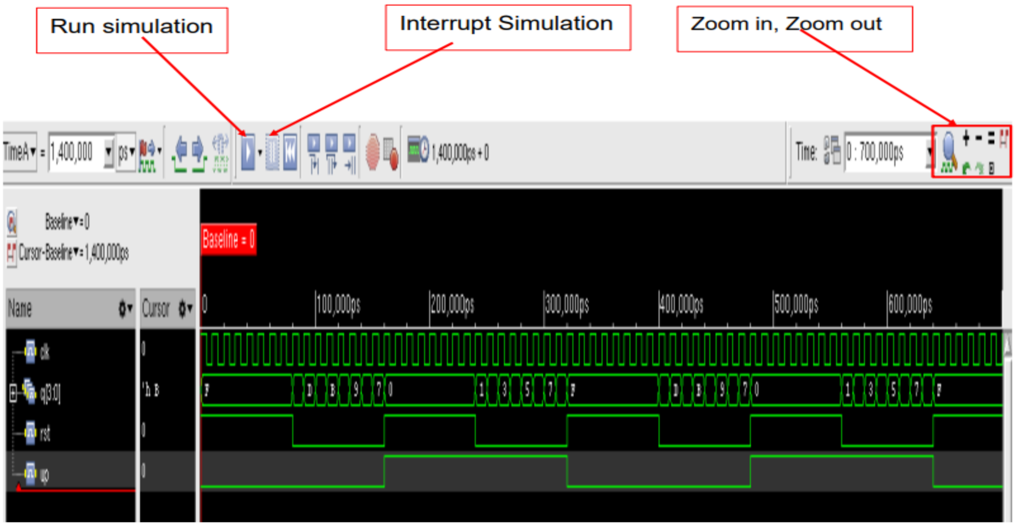
```
`timescale 1ns/1ps          // Creating Time Scale as in Source Code
module counter_test;        // Defining Module Name without Port List
reg clk, rst,m;             // Defining I/P as Registers [to Hold Values]
wire [3:0] count;            // Defining O/P as Wires [To Probe Waveforms]
Initial
begin
clk=0;                      // Initializing Clock and Reset
rst=0;#25;                  // All O/P is 4'b0000 from t=0 to t=25ns.
Rst=1;                      // Up-Down counting is allowed at posedge clk
end
initial
begin
m=1;                          // Condition for Up-Count
#600 m=0;                     // Condition for Down-Count
rst=0;#25;
rst=1;
#500 m=0;
end
counter counter1(clk,m,rst, count);    // Instantiation of Source Code
always #5 clk=~clk;                // Inverting Clk every 5ns
initial
#1400 $finish;                    // Finishing Simulation at t=1400ns
endmodule
```

You must perform the function simulation using NCLaunch (Already discuss in previous experiments).

This involves three stages :

- Step 1: Compilation- Process to check the correct Verilog language syntax and usage
- Step 2: Elaboration- To check the port connections in hierarchical design
- Step 3: Simulation- Simulate with the given test vectors over a period of time to observe the output behavior.

Final Wave form after simulation



Synthesize the design using Constraints and analyse reports, critical path and Max Operating Frequency.

Step 1 : Getting Started

- Make sure you close out all the Incisive tool windows first.
- Synthesis requires three files as follows,
  - Liberty Files (.lib)
  - Verilog/VHDL Files (.v or .vhdl or .vhd)
  - SDC (Synopsis Design Constraint) File (.sdc)

Step 2 : Creating an SDC File

- An SDC file (Synopsys Design Constraints file) is a widely used format in the IC design process to specify design constraints. These constraints guide EDA tools during synthesis, place, and route (P&R), and static timing analysis (STA). The file uses Tcl-based syntax and focuses on defining timing, clocking, and I/O constraints.

Key Features of SDC Files:

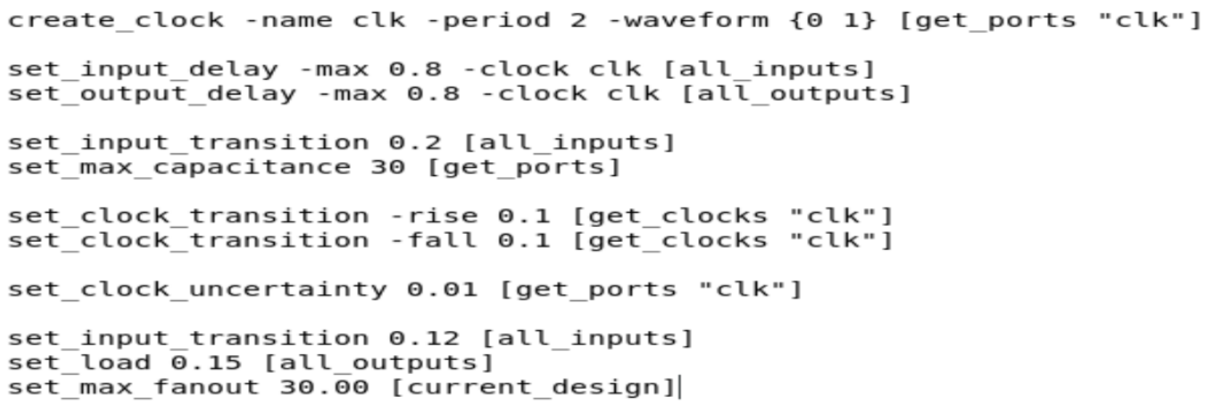
1. **Clock Definitions**:
   - Specify the clocks in the design, including their frequency, waveform, and latency.
2. **Input and Output Delays**:
   - Defines timing constraints for external inputs and outputs relative to a clock.
3. **Timing Exceptions**:
   - Includes specific paths that require exceptions, such as false paths or multi-cycle paths.
4. **Design Constraints**:
   - Specifies load, drive strength, and other characteristics to guide synthesis and STA.

Creating SDC File

- In your terminal type “gedit counter_top.sdc” to create an SDC File if you do not have one.
- The SDC File must contain the following commands;

i. create_clock -name clk -period 2 -waveform {0 1} [get_ports "clk"]
ii. set_clock_transition -rise 0.1 [get_clocks "clk"]
iii. set_clock_transition -fall 0.1 [get_clocks "clk"]
iv. set_clock_uncertainty 0.01 [get_ports "clk"]
v. set_input_delay -max 0.8 [get_ports "rst"] -clock [get_clocks "clk"]
vi. set_output_delay -max 0.8 [get_ports "count"] -clock [get_clocks "clk"]
vii. set_input_transition 0.12 [all_inputs]
viii. set_load 0.15 [all_outputs]
ix. set_max_fanout 30.00 [current_design]

i → Creates a Clock named "clk" with Time Period 2ns and On Time from t=0 to t=1.
ii, iii → Sets Clock Rise and Fall time to 100ps.
iv → Sets Clock Uncertainty to 10ps.
v, vi → Sets the maximum limit for I/O port delay to 1ps.

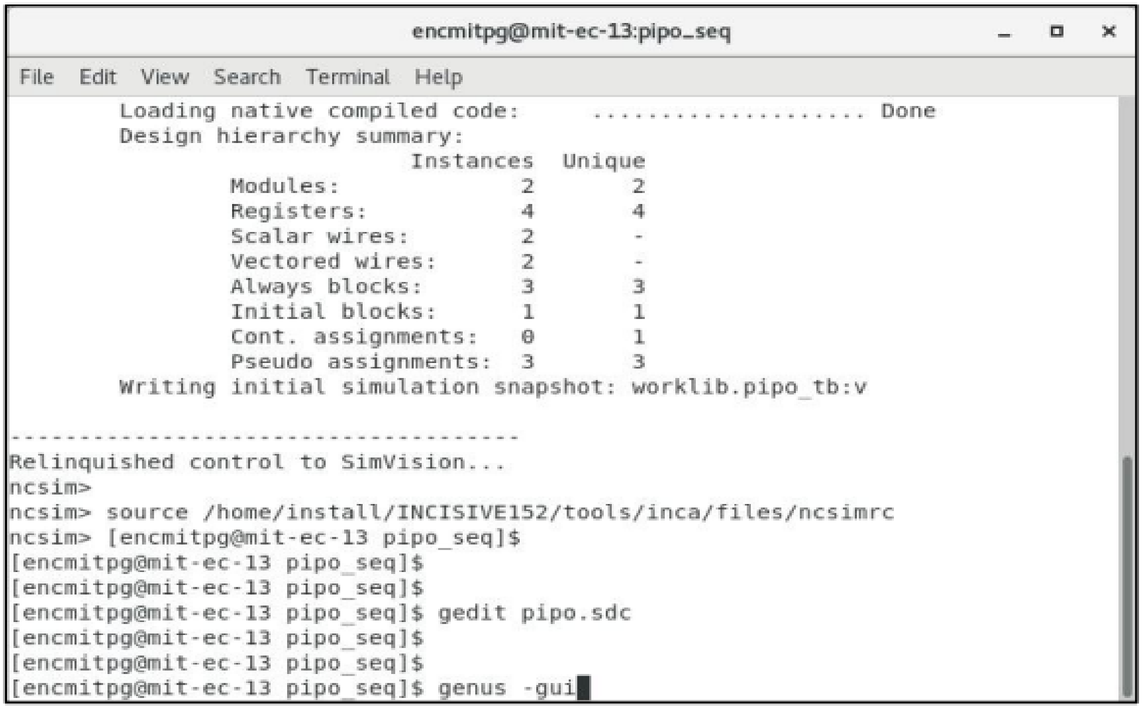


```
create_clock -name clk -period 2 -waveform {0 1} [get_ports "clk"]

set_input_delay -max 0.8 -clock clk [all_inputs]
set_output_delay -max 0.8 -clock clk [all_outputs]

set_input_transition 0.2 [all_inputs]
set_max_capacitance 30 [get_ports]

set_clock_transition -rise 0.1 [get_clocks "clk"]
set_clock_transition -fall 0.1 [get_clocks "clk"]

set_clock_uncertainty 0.01 [get_ports "clk"]

set_input_transition 0.12 [all_inputs]
set_load 0.15 [all_outputs]
set_max_fanout 30.00 [current_design]
```

Step 3: Performing Synthesis

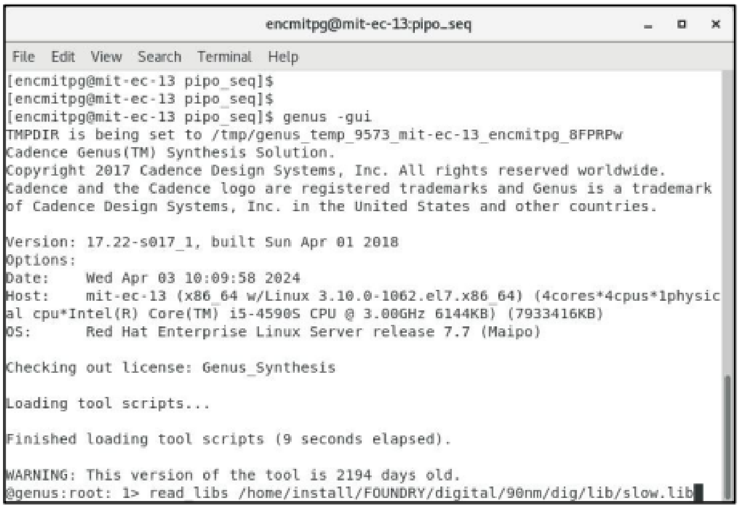
- The Liberty files are present in the below path,
  /home/install/FOUNDRY/digital/nm/dig/lib/ (This can be changed based on defined location)
- The Available technology nodes are 180nm ,90nm and 45nm.
- In the terminal, initialise the tools with the following commands if a new terminal is being used.
  - csh
  - source /home/install/cshrc
  - The tool used for Synthesis is "Genus". Hence, type "genus -gui" to open the tool.
- The Following are commands to proceed,
  1. read_libs /home/install/FOUNDRY/digital/90nm/dig/lib/slow.lib //corner analysis
  2. read_hdl counter.v
  3. elaborate
  4. read_sdc constraints_top.sdc //Reading Top Level SDC

5. set_db syn_generic_effort medium //Effort level to medium for generic, mapping and optimization
6. set_db syn_map_effort medium
7. set_db syn_opt_effort medium
8. syn_generic
9. syn_map
10. syn_opt //Performing Synthesis Mapping and Optimisation
11. report_timing > counter_timing.rep //Generates Timing report for worst datapath and dumps into file
12. report_area > counter_area.rep //Generates Synthesis Area report and dumps into a file
13. report_power > counter_power.rep //Generates Power Report [Pre-Layout]
14. write_hdl > counter_netlist.v //Creates readable Netlist File
15. write_sdc > counter_sdc.sdc //Creates Block Level SDC


encmitpg@mit-ec-13:pipo_seq
File Edit View Search Terminal Help

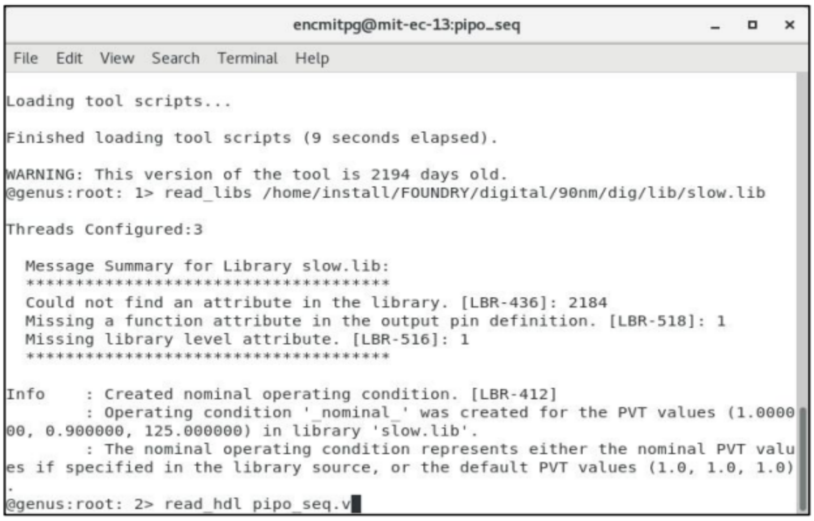
```
        Loading native compiled code:     .................... Done
        Design hierarchy summary:
                             Instances  Unique
                Modules:             2       2
                Registers:           4       4
                Scalar wires:        2       -
                Vectored wires:      2       -
                Always blocks:       3       3
                Initial blocks:      1       1
                Cont. assignments:   0       1
                Pseudo assignments:  3       3
        Writing initial simulation snapshot: worklib.pipo_tb:v

-------------------------------------
Relinquished control to SimVision...
ncsim>
ncsim> source /home/install/INCISIVE152/tools/inca/files/ncsimrc
ncsim> [encmitpg@mit-ec-13 pipo_seq]$
[encmitpg@mit-ec-13 pipo_seq]$
[encmitpg@mit-ec-13 pipo_seq]$
[encmitpg@mit-ec-13 pipo_seq]$ gedit pipo.sdc
[encmitpg@mit-ec-13 pipo_seq]$
[encmitpg@mit-ec-13 pipo_seq]$
[encmitpg@mit-ec-13 pipo_seq]$ genus -gui
```

Fig. 31: Launch genus


encmitpg@mit-ec-13:pipo_seq
File Edit View Search Terminal Help

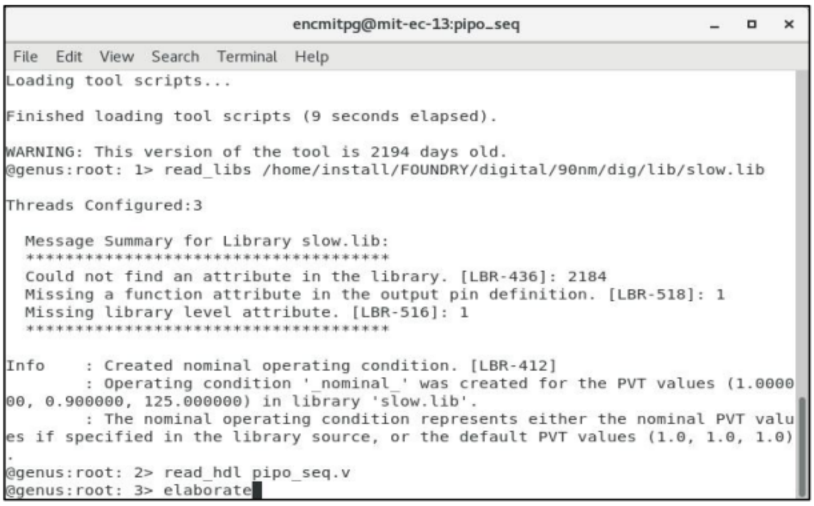
```
[encmitpg@mit-ec-13 pipo_seq]$
[encmitpg@mit-ec-13 pipo_seq]$
[encmitpg@mit-ec-13 pipo_seq]$ genus -gui
TMPDIR is being set to /tmp/genus_temp_9573_mit-ec-13_encmitpg_8FPRPw
Cadence Genus(TM) Synthesis Solution.
Copyright 2017 Cadence Design Systems, Inc. All rights reserved worldwide.
Cadence and the Cadence logo are registered trademarks and Genus is a trademark
of Cadence Design Systems, Inc. in the United States and other countries.

Version: 17.22-s017_1, built Sun Apr 01 2018
Options:
Date:    Wed Apr 03 10:09:58 2024
Host:    mit-ec-13 (x86_64 w/Linux 3.10.0-1062.el7.x86_64) (4cores*4cpus*1physic
al cpu*Intel(R) Core(TM) i5-4590S CPU @ 3.00GHz 6144KB) (7933416KB)
OS:      Red Hat Enterprise Linux Server release 7.7 (Maipo)

Checking out license: Genus_Synthesis

Loading tool scripts...

Finished loading tool scripts (9 seconds elapsed).

WARNING: This version of the tool is 2194 days old.
@genus:root: 1> read_libs /home/install/FOUNDRY/digital/90nm/dig/lib/slow.lib
```

Fig. 32: Read library to specify library path


encmitpg@mit-ec-13:pipo_seq
File Edit View Search Terminal Help

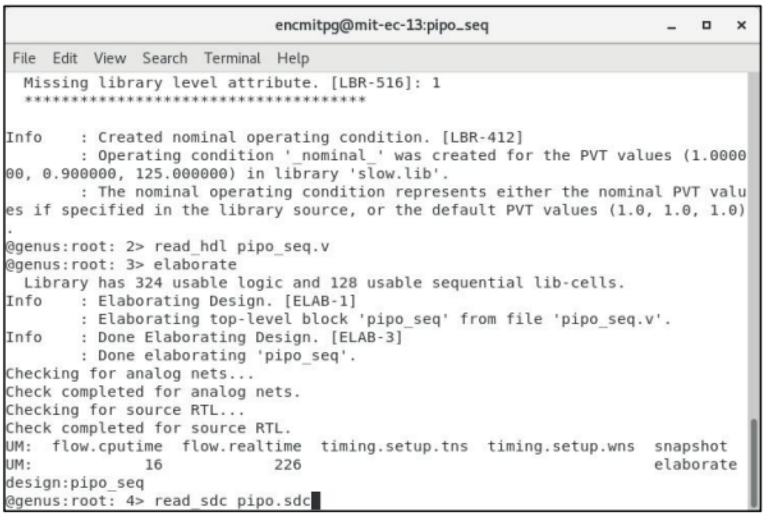
```

Loading tool scripts...

Finished loading tool scripts (9 seconds elapsed).

WARNING: This version of the tool is 2194 days old.
@genus:root: 1> read_libs /home/install/FOUNDRY/digital/90nm/dig/lib/slow.lib

Threads Configured:3

  Message Summary for Library slow.lib:
  ***************************************
  Could not find an attribute in the library. [LBR-436]: 2184
  Missing a function attribute in the output pin definition. [LBR-518]: 1
  Missing library level attribute. [LBR-516]: 1
  ***************************************

Info    : Created nominal operating condition. [LBR-412]
        : Operating condition '_nominal_' was created for the PVT values (1.0000
00, 0.900000, 125.000000) in library 'slow.lib'.
        : The nominal operating condition represents either the nominal PVT valu
es if specified in the library source, or the default PVT values (1.0, 1.0, 1.0)
.
@genus:root: 2> read_hdl pipo_seq.v
```

Fig. 33 read the hdl design file


encmitpg@mit-ec-13:pipo_seq
File Edit View Search Terminal Help

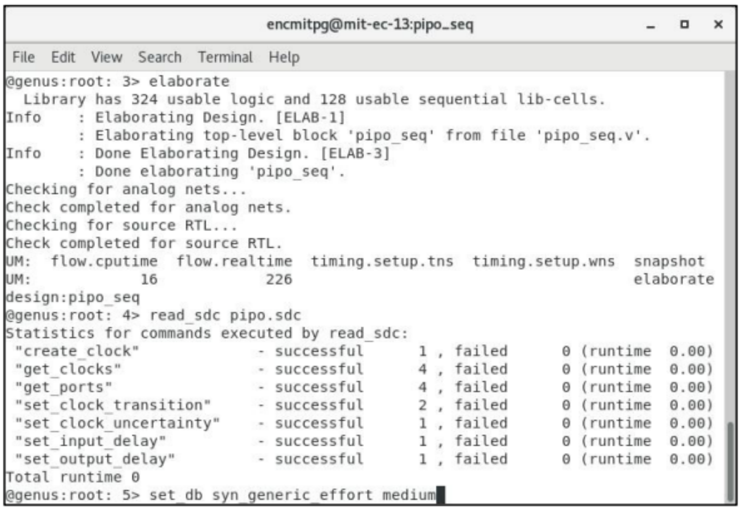
```
Loading tool scripts...

Finished loading tool scripts (9 seconds elapsed).

WARNING: This version of the tool is 2194 days old.
@genus:root: 1> read_libs /home/install/FOUNDRY/digital/90nm/dig/lib/slow.lib

Threads Configured:3

  Message Summary for Library slow.lib:
  ***************************************
  Could not find an attribute in the library. [LBR-436]: 2184
  Missing a function attribute in the output pin definition. [LBR-518]: 1
  Missing library level attribute. [LBR-516]: 1
  ***************************************

Info    : Created nominal operating condition. [LBR-412]
        : Operating condition '_nominal_' was created for the PVT values (1.0000
00, 0.900000, 125.000000) in library 'slow.lib'.
        : The nominal operating condition represents either the nominal PVT valu
es if specified in the library source, or the default PVT values (1.0, 1.0, 1.0)
.
@genus:root: 2> read_hdl pipo_seq.v
@genus:root: 3> elaborate
```

Fig. 34: elaborate


encmitpg@mit-ec-13:pipo_seq
File Edit View Search Terminal Help

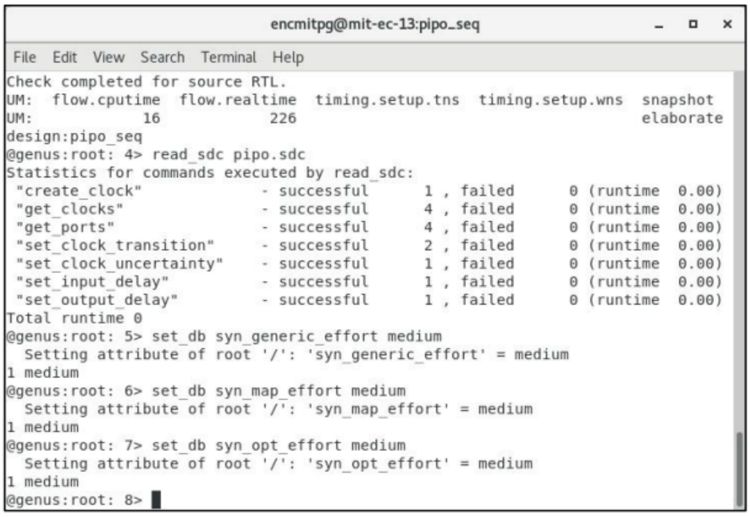
```
  Missing library level attribute. [LBR-516]: 1
  ************************************

Info    : Created nominal operating condition. [LBR-412]
        : Operating condition '_nominal_' was created for the PVT values (1.0000
00, 0.900000, 125.000000) in library 'slow.lib'.
        : The nominal operating condition represents either the nominal PVT valu
es if specified in the library source, or the default PVT values (1.0, 1.0, 1.0)
.
@genus:root: 2> read_hdl pipo_seq.v
@genus:root: 3> elaborate
  Library has 324 usable logic and 128 usable sequential lib-cells.
Info    : Elaborating Design. [ELAB-1]
        : Elaborating top-level block 'pipo_seq' from file 'pipo_seq.v'.
Info    : Done Elaborating Design. [ELAB-3]
        : Done elaborating 'pipo_seq'.
Checking for analog nets...
Check completed for analog nets.
Checking for source RTL...
Check completed for source RTL.
UM:  flow.cputime  flow.realtime  timing.setup.tns  timing.setup.wns  snapshot
UM:            16            226                                     elaborate
design:pipo_seq
@genus:root: 4> read_sdc pipo.sdc
```

Fig. 35: read constraint file


encmitpg@mit-ec-13:pipo_seq
File Edit View Search Terminal Help

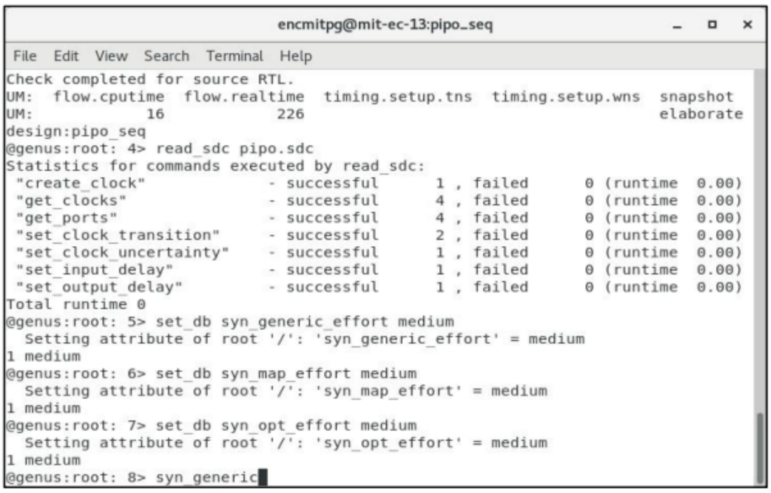
```
@genus:root: 3> elaborate
  Library has 324 usable logic and 128 usable sequential lib-cells.
Info    : Elaborating Design. [ELAB-1]
        : Elaborating top-level block 'pipo_seq' from file 'pipo_seq.v'.
Info    : Done Elaborating Design. [ELAB-3]
        : Done elaborating 'pipo_seq'.
Checking for analog nets...
Check completed for analog nets.
Checking for source RTL...
Check completed for source RTL.
UM:  flow.cputime  flow.realtime  timing.setup.tns  timing.setup.wns  snapshot
UM:            16            226                                     elaborate
design:pipo_seq
@genus:root: 4> read_sdc pipo.sdc
Statistics for commands executed by read_sdc:
 "create_clock"             - successful      1 , failed      0 (runtime  0.00)
 "get_clocks"               - successful      4 , failed      0 (runtime  0.00)
 "get_ports"                - successful      4 , failed      0 (runtime  0.00)
 "set_clock_transition"     - successful      2 , failed      0 (runtime  0.00)
 "set_clock_uncertainty"    - successful      1 , failed      0 (runtime  0.00)
 "set_input_delay"          - successful      1 , failed      0 (runtime  0.00)
 "set_output_delay"         - successful      1 , failed      0 (runtime  0.00)
Total runtime 0
@genus:root: 5> set_db syn_generic_effort medium
```

Fig. 36: set effort level for translation (generic)


encmitpg@mit-ec-13:pipo_seq
File Edit View Search Terminal Help

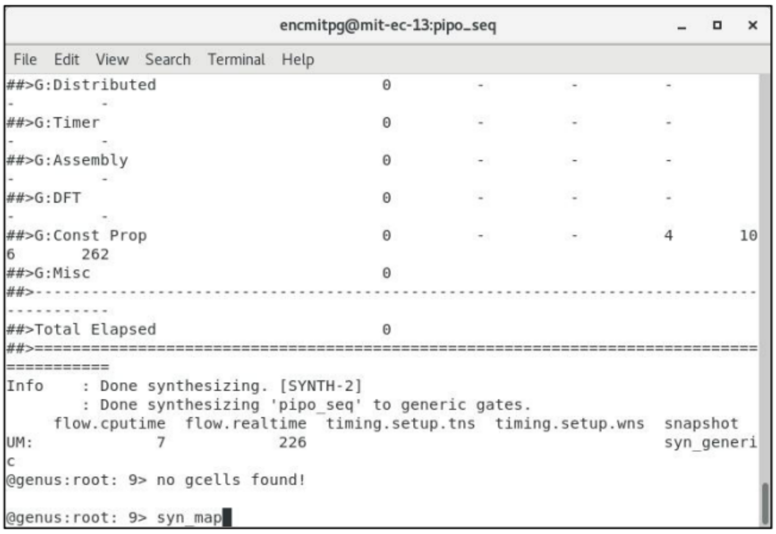
```
Check completed for source RTL.
UM:  flow.cputime  flow.realtime  timing.setup.tns  timing.setup.wns  snapshot
UM:            16            226                                     elaborate
design:pipo_seq
@genus:root: 4> read_sdc pipo.sdc
Statistics for commands executed by read_sdc:
 "create_clock"             - successful      1 , failed      0 (runtime  0.00)
 "get_clocks"               - successful      4 , failed      0 (runtime  0.00)
 "get_ports"                - successful      4 , failed      0 (runtime  0.00)
 "set_clock_transition"     - successful      2 , failed      0 (runtime  0.00)
 "set_clock_uncertainty"    - successful      1 , failed      0 (runtime  0.00)
 "set_input_delay"          - successful      1 , failed      0 (runtime  0.00)
 "set_output_delay"         - successful      1 , failed      0 (runtime  0.00)
Total runtime 0
@genus:root: 5> set_db syn_generic_effort medium
  Setting attribute of root '/': 'syn_generic_effort' = medium
1 medium
@genus:root: 6> set_db syn_map_effort medium
  Setting attribute of root '/': 'syn_map_effort' = medium
1 medium
@genus:root: 7> set_db syn_opt_effort medium
  Setting attribute of root '/': 'syn_opt_effort' = medium
1 medium
@genus:root: 8>
```

Fig. 37: set effort level for mapping and optimization


encmitpg@mit-ec-13:pipo_seq
File Edit View Search Terminal Help

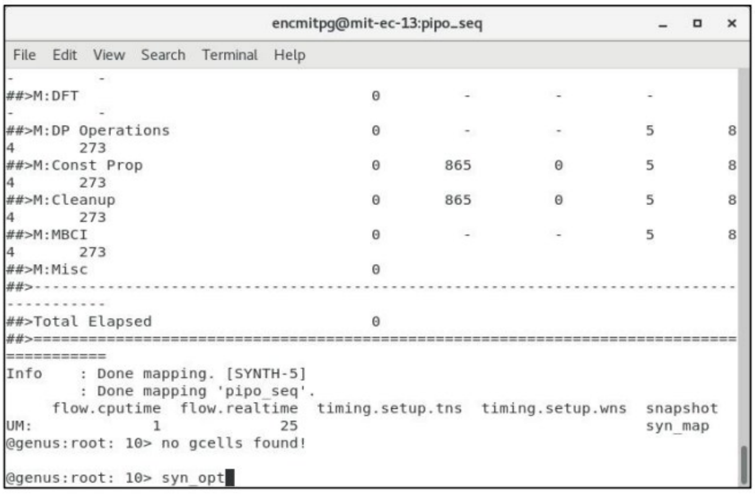
```
Check completed for source RTL.
UM:  flow.cputime  flow.realtime  timing.setup.tns  timing.setup.wns  snapshot
UM:            16            226                                       elaborate
design:pipo_seq
@genus:root: 4> read_sdc pipo.sdc
Statistics for commands executed by read_sdc:
 "create_clock"              - successful      1 , failed      0 (runtime  0.00)
 "get_clocks"                - successful      4 , failed      0 (runtime  0.00)
 "get_ports"                 - successful      4 , failed      0 (runtime  0.00)
 "set_clock_transition"      - successful      2 , failed      0 (runtime  0.00)
 "set_clock_uncertainty"     - successful      1 , failed      0 (runtime  0.00)
 "set_input_delay"           - successful      1 , failed      0 (runtime  0.00)
 "set_output_delay"          - successful      1 , failed      0 (runtime  0.00)
Total runtime 0
@genus:root: 5> set_db syn_generic_effort medium
  Setting attribute of root '/': 'syn_generic_effort' = medium
1 medium
@genus:root: 6> set_db syn_map_effort medium
  Setting attribute of root '/': 'syn_map_effort' = medium
1 medium
@genus:root: 7> set_db syn_opt_effort medium
  Setting attribute of root '/': 'syn_opt_effort' = medium
1 medium
@genus:root: 8> syn_generic
```

Fig. 38: perform translation


encmitpg@mit-ec-13:pipo_seq
File Edit View Search Terminal Help

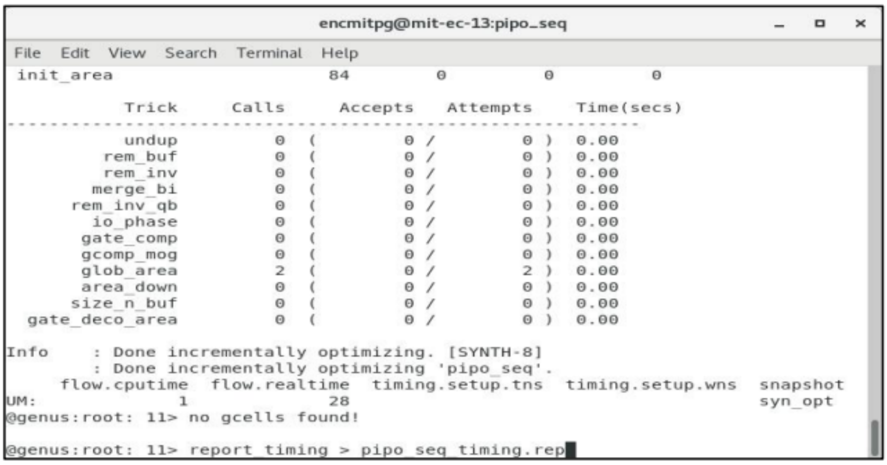
```
##>G:Distributed                         0         -         -         -
-         -
##>G:Timer                               0         -         -         -
-         -
##>G:Assembly                            0         -         -         -
-         -
##>G:DFT                                 0         -         -         -
-         -
##>G:Const Prop                          0         -         -         4        10
6       262
##>G:Misc                                0
##>-----------------------------------------------------------------------------
-----------
##>Total Elapsed                         0
##>=============================================================================
===========
Info    : Done synthesizing. [SYNTH-2]
        : Done synthesizing 'pipo_seq' to generic gates.
     flow.cputime  flow.realtime  timing.setup.tns  timing.setup.wns  snapshot
UM:             7            226                                      syn_generi
c
@genus:root: 9> no gcells found!

@genus:root: 9> syn_map
```

Fig. 39: Perform mapping


encmitpg@mit-ec-13:pipo_seq
File Edit View Search Terminal Help

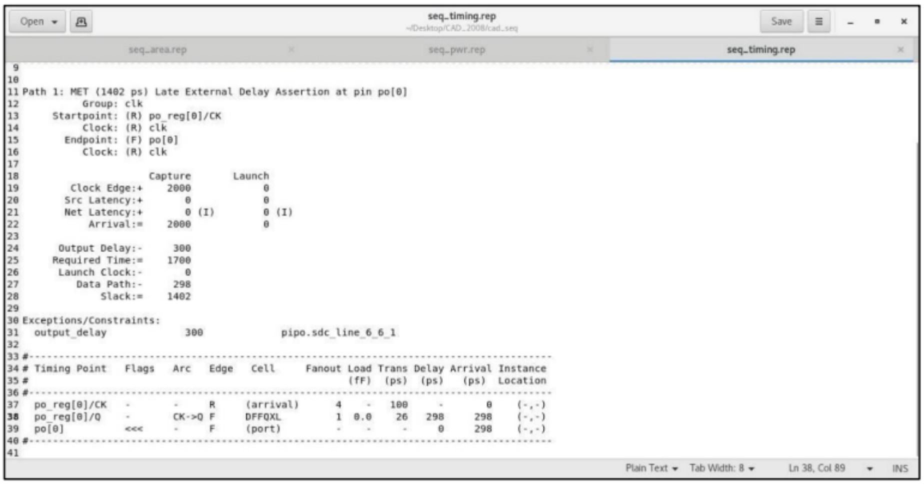
```
-         -
##>M:DFT                                 0         -         -         -
-         -
##>M:DP Operations                       0         -         -         5         8
4       273
##>M:Const Prop                          0       865         0         5         8
4       273
##>M:Cleanup                             0       865         0         5         8
4       273
##>M:MBCI                                0         -         -         5         8
4       273
##>M:Misc                                0
##>-----------------------------------------------------------------------------
-----------
##>Total Elapsed                         0
##>=============================================================================
===========
Info    : Done mapping. [SYNTH-5]
        : Done mapping 'pipo_seq'.
     flow.cputime  flow.realtime  timing.setup.tns  timing.setup.wns  snapshot
UM:             1             25                                      syn_map
@genus:root: 10> no gcells found!

@genus:root: 10> syn_opt
```

Fig. 40: perform optimization


encmitpg@mit-ec-13:pipo_seq
File Edit View Search Terminal Help

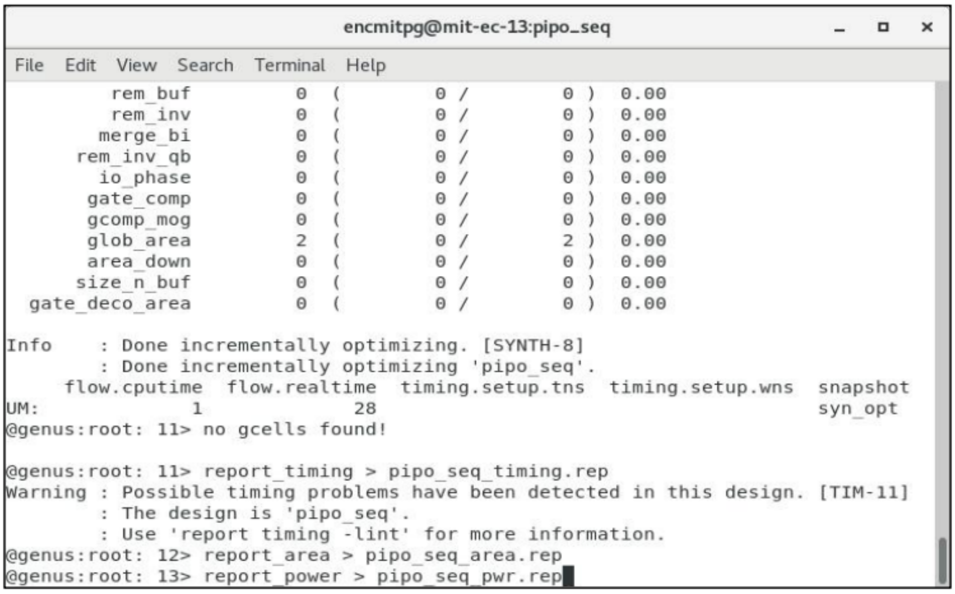
```
 init_area                    84         0          0          0

              Trick      Calls      Accepts   Attempts    Time(secs)
-----------------------------------------------------------------
              undup          0  (         0 /        0 )  0.00
            rem_buf          0  (         0 /        0 )  0.00
            rem_inv          0  (         0 /        0 )  0.00
           merge_bi          0  (         0 /        0 )  0.00
         rem_inv_qb          0  (         0 /        0 )  0.00
           io_phase          0  (         0 /        0 )  0.00
          gate_comp          0  (         0 /        0 )  0.00
          gcomp_mog          0  (         0 /        0 )  0.00
          glob_area          2  (         0 /        2 )  0.00
          area_down          0  (         0 /        0 )  0.00
         size_n_buf          0  (         0 /        0 )  0.00
     gate_deco_area          0  (         0 /        0 )  0.00

Info    : Done incrementally optimizing. [SYNTH-8]
        : Done incrementally optimizing 'pipo_seq'.
     flow.cputime  flow.realtime  timing.setup.tns  timing.setup.wns  snapshot
UM:            1             28                                       syn_opt
@genus:root: 11> no gcells found!

@genus:root: 11> report_timing > pipo_seq_timing.rep
```

Fig. 41: generate timing report



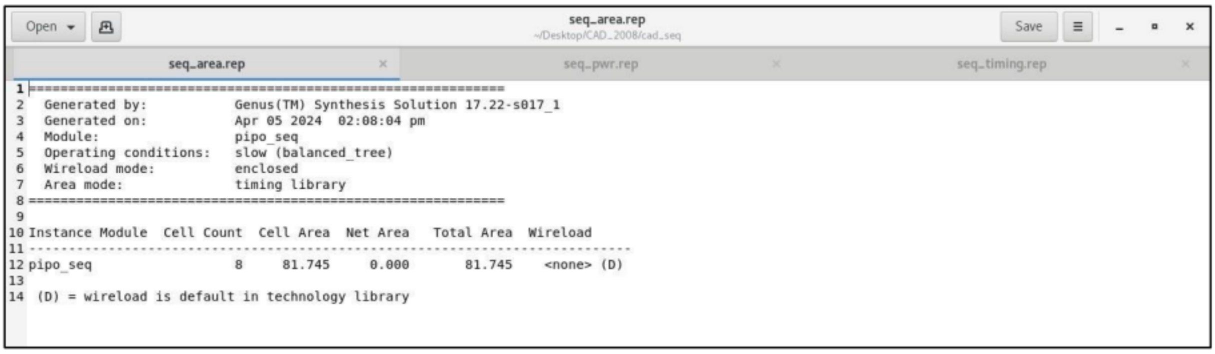
```


Path 1: MET (1402 ps) Late External Delay Assertion at pin po[0]
          Group: clk
     Startpoint: (R) po_reg[0]/CK
          Clock: (R) clk
       Endpoint: (F) po[0]
          Clock: (R) clk

                     Capture       Launch
        Clock Edge:+    2000            0
       Src Latency:+       0            0
       Net Latency:+       0 (I)        0 (I)
           Arrival:=    2000            0

      Output Delay:-     300
     Required Time:=    1700
      Launch Clock:-       0
         Data Path:-     298
             Slack:=    1402

Exceptions/Constraints:
  output_delay             300             pipo.sdc_line_6_6_1

#---------------------------------------------------------------------------
# Timing Point   Flags   Arc   Edge   Cell      Fanout Load Trans Delay Arrival Instance
#                                                      (fF)  (ps)  (ps)    (ps) Location
#---------------------------------------------------------------------------
  po_reg[0]/CK   -       -     R      (arrival)      4    -   100     -       0    (-,-)
  po_reg[0]/Q    -       CK->Q F      DFFQXL         1  0.0    26   298     298    (-,-)
  po[0]          <<<     -     F      (port)         -    -     -     0     298    (-,-)
#---------------------------------------------------------------------------

```

Fig. 42: Timing report


encmitpg@mit-ec-13:pipo_seq
File Edit View Search Terminal Help

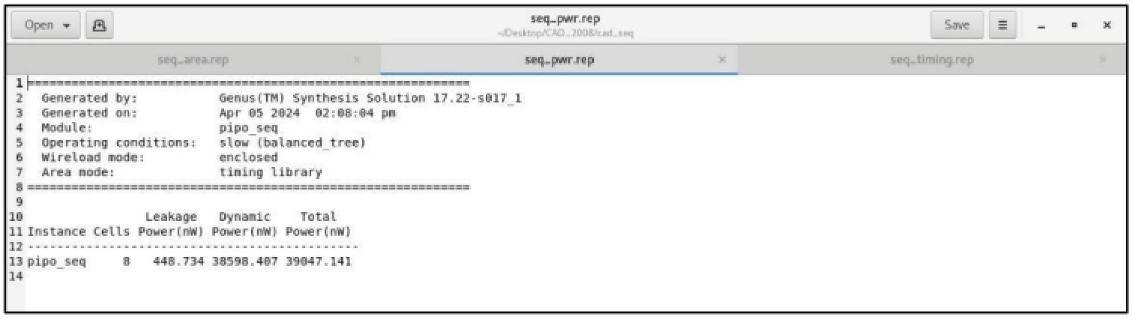
```
         rem_buf          0  (         0 /        0 )  0.00
         rem_inv          0  (         0 /        0 )  0.00
        merge_bi          0  (         0 /        0 )  0.00
      rem_inv_qb          0  (         0 /        0 )  0.00
        io_phase          0  (         0 /        0 )  0.00
       gate_comp          0  (         0 /        0 )  0.00
       gcomp_mog          0  (         0 /        0 )  0.00
       glob_area          2  (         0 /        2 )  0.00
       area_down          0  (         0 /        0 )  0.00
      size_n_buf          0  (         0 /        0 )  0.00
  gate_deco_area          0  (         0 /        0 )  0.00

Info    : Done incrementally optimizing. [SYNTH-8]
        : Done incrementally optimizing 'pipo_seq'.
     flow.cputime  flow.realtime  timing.setup.tns  timing.setup.wns  snapshot
UM:            1             28                                       syn_opt
@genus:root: 11> no gcells found!

@genus:root: 11> report_timing > pipo_seq_timing.rep
Warning : Possible timing problems have been detected in this design. [TIM-11]
        : The design is 'pipo_seq'.
        : Use 'report_timing -lint' for more information.
@genus:root: 12> report_area > pipo_seq_area.rep
@genus:root: 13> report_power > pipo_seq_pwr.rep
```

Fig. 43: Generate area and power report



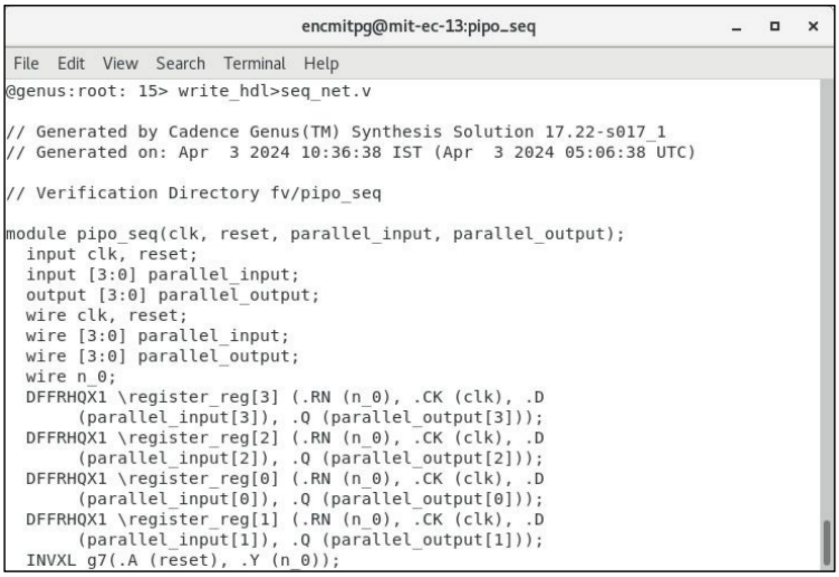
```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 17.22-s017_1
  Generated on:           Apr 05 2024  02:08:04 pm
  Module:                 pipo_seq
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

Instance Module  Cell Count  Cell Area  Net Area   Total Area  Wireload
-------------------------------------------------------------------------
pipo_seq                  8     81.745     0.000       81.745    <none> (D)

 (D) = wireload is default in technology library
```

Fig. 44: Area report



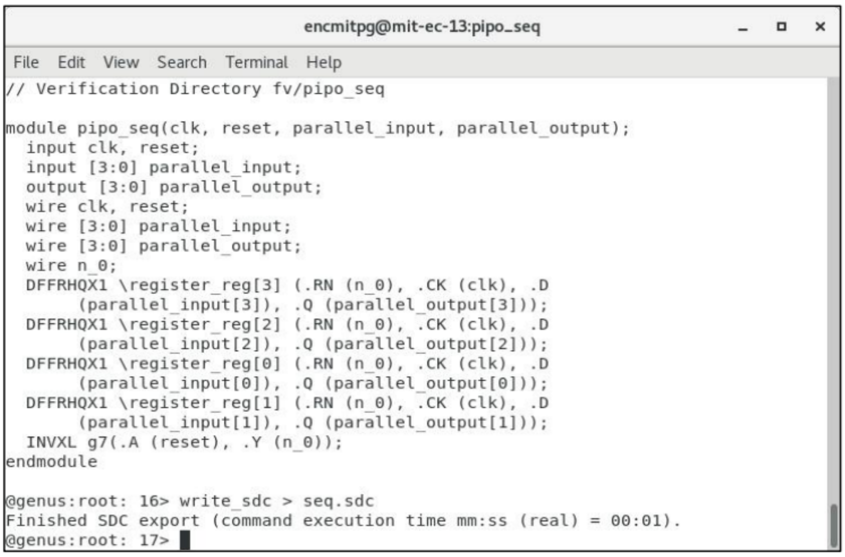
```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 17.22-s017_1
  Generated on:           Apr 05 2024  02:08:04 pm
  Module:                 pipo_seq
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

                Leakage   Dynamic    Total
Instance Cells Power(nW) Power(nW) Power(nW)
--------------------------------------------
pipo_seq     8   448.734 38598.407 39047.141
```

Fig. 45: Power report


encmitpg@mit-ec-13:pipo_seq
File Edit View Search Terminal Help

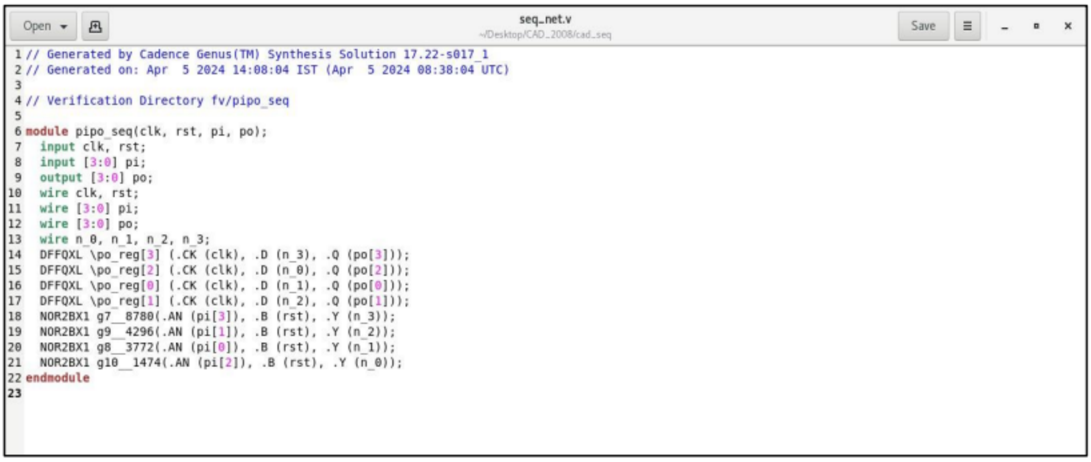
```
@genus:root: 15> write_hdl>seq_net.v

// Generated by Cadence Genus(TM) Synthesis Solution 17.22-s017_1
// Generated on: Apr  3 2024 10:36:38 IST (Apr  3 2024 05:06:38 UTC)

// Verification Directory fv/pipo_seq

module pipo_seq(clk, reset, parallel_input, parallel_output);
  input clk, reset;
  input [3:0] parallel_input;
  output [3:0] parallel_output;
  wire clk, reset;
  wire [3:0] parallel_input;
  wire [3:0] parallel_output;
  wire n_0;
  DFFRHQX1 \register_reg[3] (.RN (n_0), .CK (clk), .D
       (parallel_input[3]), .Q (parallel_output[3]));
  DFFRHQX1 \register_reg[2] (.RN (n_0), .CK (clk), .D
       (parallel_input[2]), .Q (parallel_output[2]));
  DFFRHQX1 \register_reg[0] (.RN (n_0), .CK (clk), .D
       (parallel_input[0]), .Q (parallel_output[0]));
  DFFRHQX1 \register_reg[1] (.RN (n_0), .CK (clk), .D
       (parallel_input[1]), .Q (parallel_output[1]));
  INVXL g7(.A (reset), .Y (n_0));
```

Fig. 46: Generate netlist


encmitpg@mit-ec-13:pipo_seq
File Edit View Search Terminal Help

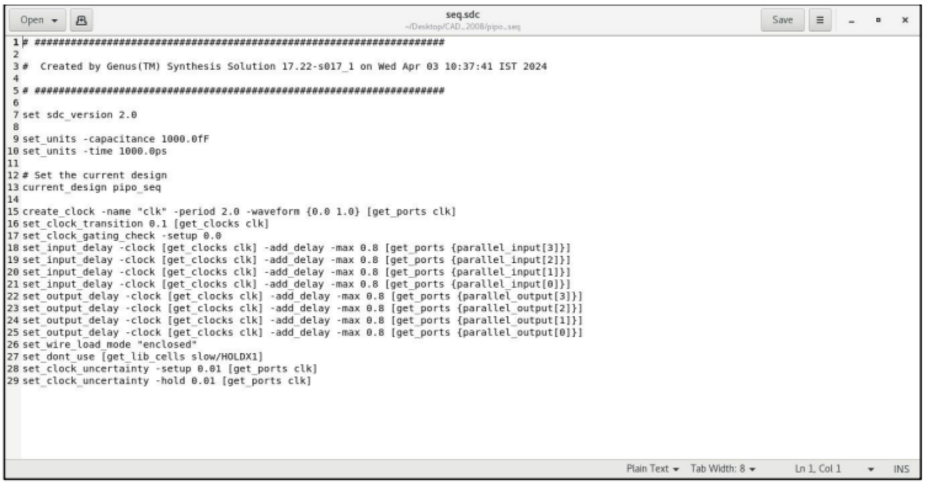
```
// Verification Directory fv/pipo_seq

module pipo_seq(clk, reset, parallel_input, parallel_output);
  input clk, reset;
  input [3:0] parallel_input;
  output [3:0] parallel_output;
  wire clk, reset;
  wire [3:0] parallel_input;
  wire [3:0] parallel_output;
  wire n_0;
  DFFRHQX1 \register_reg[3] (.RN (n_0), .CK (clk), .D
       (parallel_input[3]), .Q (parallel_output[3]));
  DFFRHQX1 \register_reg[2] (.RN (n_0), .CK (clk), .D
       (parallel_input[2]), .Q (parallel_output[2]));
  DFFRHQX1 \register_reg[0] (.RN (n_0), .CK (clk), .D
       (parallel_input[0]), .Q (parallel_output[0]));
  DFFRHQX1 \register_reg[1] (.RN (n_0), .CK (clk), .D
       (parallel_input[1]), .Q (parallel_output[1]));
  INVXL g7(.A (reset), .Y (n_0));
endmodule

@genus:root: 16> write_sdc > seq.sdc
Finished SDC export (command execution time mm:ss (real) = 00:01).
@genus:root: 17>
```

Fig. 47: Generate block level constraint file



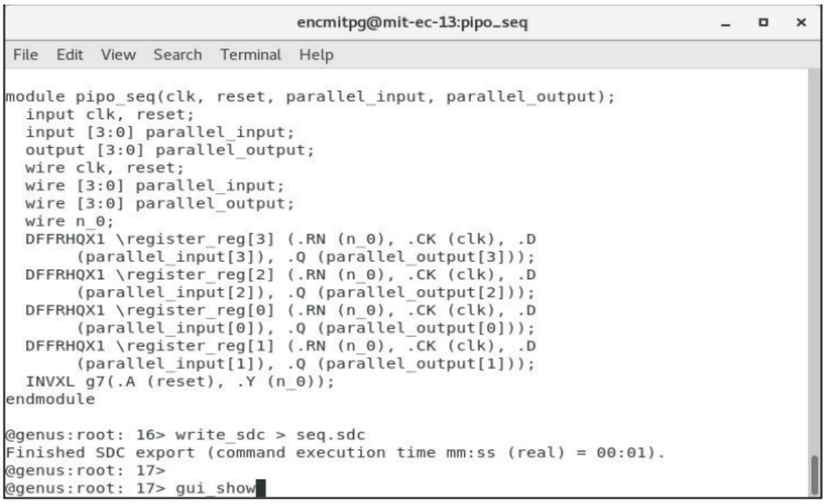
```
// Generated by Cadence Genus(TM) Synthesis Solution 17.22-s017_1
// Generated on: Apr  5 2024 14:08:04 IST (Apr  5 2024 08:38:04 UTC)

// Verification Directory fv/pipo_seq

module pipo_seq(clk, rst, pi, po);
  input clk, rst;
  input [3:0] pi;
  output [3:0] po;
  wire clk, rst;
  wire [3:0] pi;
  wire [3:0] po;
  wire n_0, n_1, n_2, n_3;
  DFFQXL \po_reg[3] (.CK (clk), .D (n_3), .Q (po[3]));
  DFFQXL \po_reg[2] (.CK (clk), .D (n_0), .Q (po[2]));
  DFFQXL \po_reg[0] (.CK (clk), .D (n_1), .Q (po[0]));
  DFFQXL \po_reg[1] (.CK (clk), .D (n_2), .Q (po[1]));
  NOR2BX1 g7__8780(.AN (pi[3]), .B (rst), .Y (n_3));
  NOR2BX1 g9__4296(.AN (pi[1]), .B (rst), .Y (n_2));
  NOR2BX1 g8__3772(.AN (pi[0]), .B (rst), .Y (n_1));
  NOR2BX1 g10__1474(.AN (pi[2]), .B (rst), .Y (n_0));
endmodule

```

Fig. 48: Generated netlist



```
# ######################################################################

#  Created by Genus(TM) Synthesis Solution 17.22-s017_1 on Wed Apr 03 10:37:41 IST 2024

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

set sdc_version 2.0

set_units -capacitance 1000.0fF
set_units -time 1000.0ps

# Set the current design
current_design pipo_seq

create_clock -name "clk" -period 2.0 -waveform {0.0 1.0} [get_ports clk]
set_clock_transition 0.1 [get_clocks clk]
set_clock_gating_check -setup 0.0
set_input_delay -clock [get_clocks clk] -add_delay -max 0.8 [get_ports {parallel_input[3]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 0.8 [get_ports {parallel_input[2]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 0.8 [get_ports {parallel_input[1]}]
set_input_delay -clock [get_clocks clk] -add_delay -max 0.8 [get_ports {parallel_input[0]}]
set_output_delay -clock [get_clocks clk] -add_delay -max 0.8 [get_ports {parallel_output[3]}]
set_output_delay -clock [get_clocks clk] -add_delay -max 0.8 [get_ports {parallel_output[2]}]
set_output_delay -clock [get_clocks clk] -add_delay -max 0.8 [get_ports {parallel_output[1]}]
set_output_delay -clock [get_clocks clk] -add_delay -max 0.8 [get_ports {parallel_output[0]}]
set_wire_load_mode "enclosed"
set_dont_use [get_lib_cells slow/HOLDX1]
set_clock_uncertainty -setup 0.01 [get_ports clk]
set_clock_uncertainty -hold 0.01 [get_ports clk]
```

Fig. 49: Generated block level constraint file



encmitpg@mit-ec-13:pipo_seq
File Edit View Search Terminal Help

```
module pipo_seq(clk, reset, parallel_input, parallel_output);
  input clk, reset;
  input [3:0] parallel_input;
  output [3:0] parallel_output;
  wire clk, reset;
  wire [3:0] parallel_input;
  wire [3:0] parallel_output;
  wire n_0;
  DFFRHQX1 \register_reg[3] (.RN (n_0), .CK (clk), .D
       (parallel_input[3]), .Q (parallel_output[3]));
  DFFRHQX1 \register_reg[2] (.RN (n_0), .CK (clk), .D
       (parallel_input[2]), .Q (parallel_output[2]));
  DFFRHQX1 \register_reg[0] (.RN (n_0), .CK (clk), .D
       (parallel_input[0]), .Q (parallel_output[0]));
  DFFRHQX1 \register_reg[1] (.RN (n_0), .CK (clk), .D
       (parallel_input[1]), .Q (parallel_output[1]));
  INVXL g7(.A (reset), .Y (n_0));
endmodule

@genus:root: 16> write_sdc > seq.sdc
Finished SDC export (command execution time mm:ss (real) = 00:01).
@genus:root: 17>
@genus:root: 17> gui_show
```

Fig. 50: Schematic generation

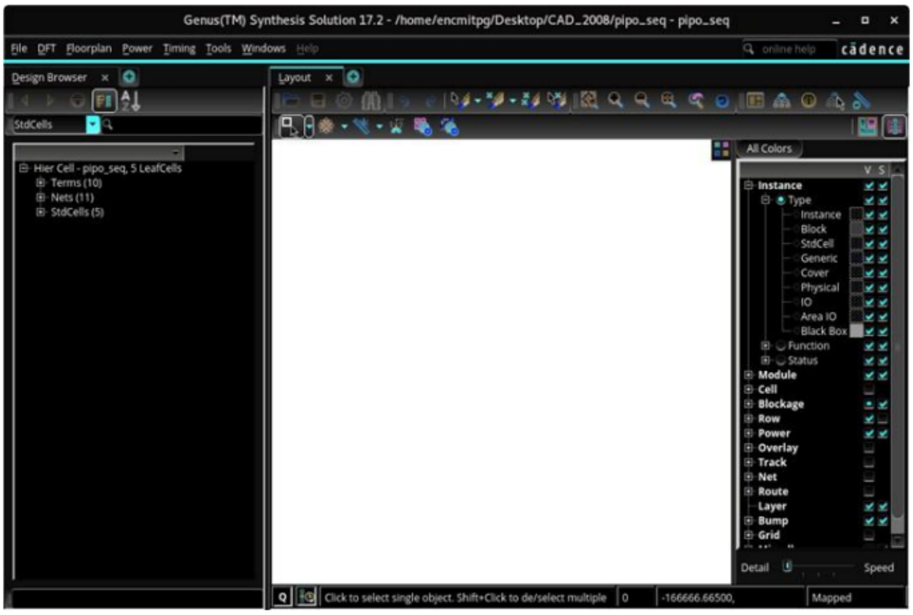
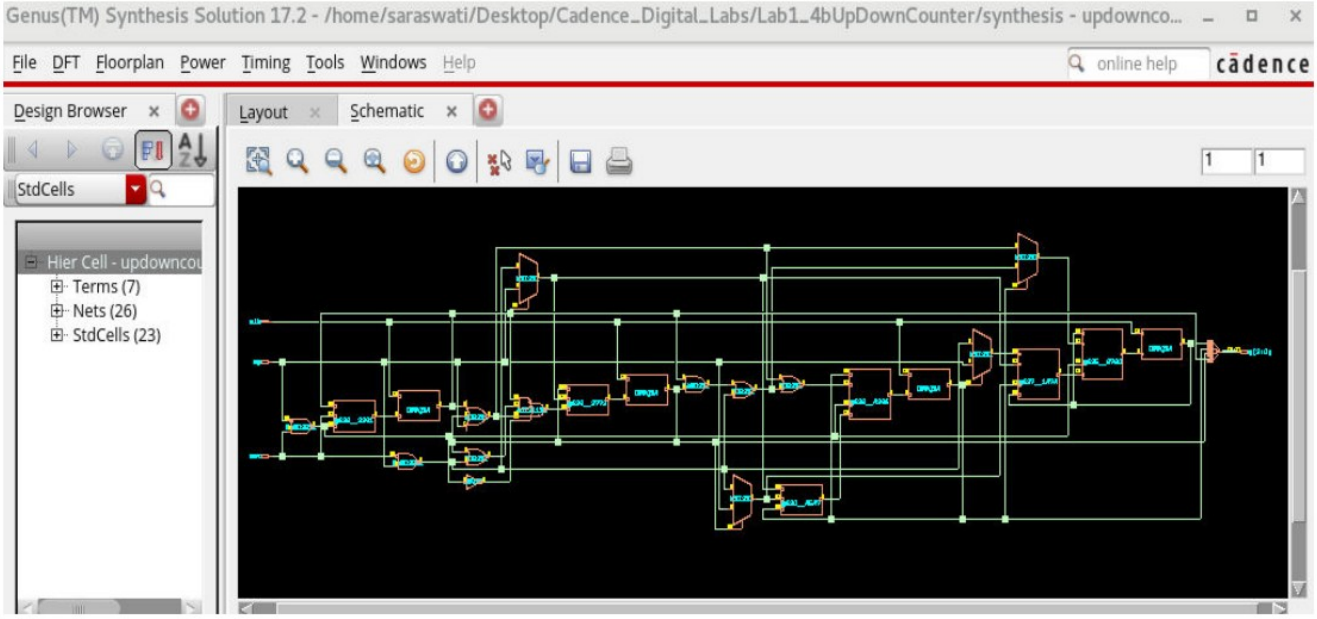


Fig. 51: Genus synthesis solution window appears

Synthesis RTL Schematic : Schematic Capture of 4bit Up-Down Counter


Genus(TM) Synthesis Solution 17.2 - /home/saraswati/Desktop/Cadence_Digital_Labs/Lab1_4bUpDownCounter/synthesis - updownco...
File DFT Floorplan Power Timing Tools Windows Help
online help
cādence
Design Browser
Layout
Schematic
StdCells
Hier Cell - updowncou
Terms (7)
Nets (26)
StdCells (23)

Commands 1-5 are intended for Synthesis process while 11-15 for Generating reports and Outputs.

Note :-
1) report_timing gives you the path with highest failing slack where
Setup Slack = Required Time – Arrival Time.

2) Worst Setup Slack ==> Highest Arrival time ==> Highest Propagation Delay.

3) Maximum Clock Frequency = 1/ (Max Data Path Delay – Min Clock Path Delay + Tsetup)

All the Information can be gathered from report_timing.

4) The Cells given in the netlist can be checked in the .lib files for their properties.

c) Compilation, Simulation and Synthesis of 32-bit Up/Down Counter.

Source Code :

```
 `timescale 1ns/1ps                          //Defining a Timescale for Precision
module counter(clk,rst,m,count);             //Defining Module and Port List
input clk,rst,m;                                      //Defining Inputs
output reg [31:0]count;                      //Defining 4-bit Output as Reg type
always@(posedge clk or negedge rst)          //The Block is executed when
begin                                        //EITHER of positive edge of clock
if(!rst)                                     //or Neg Edge of Rst arrives
count=0;                                              // Both are independent events
if(m)
count=count+1;
else
count=count-1;
end
endmodule
```

Test Bench :

```
`timescale 1ns/1ps           //Creating Time Scale as in Source Code
module counter_test;           //Defining Module Name without Port List
reg clk, rst,m;                //Defining I/P as Registers [to Hold Values]
wire [31:0] count;             //Defining O/P as Wires [To Probe Waveforms]
initial
begin
clk=0;                         //Initializing Clock and Reset
rst=0;#25;                              //All O/P is 4'b0000 from t=0 to t=25ns.
rst=1;                         //Up-Down counting is allowed at posedge clk
end
20

initial
begin
m=1;                                  //Condition for Up-Count
#600 m=0;                             //Condition for Down-Count
rst=0;#25;
rst=1;
#500 m=0;
end
counter counter1(clk,m,rst, count);   //Instantiation of Source Code
always #5 clk=~clk;                       //Inverting Clk every 5ns
initial
#1400 $finish;                           //Finishing Simulation at t=1400ns
endmodule
```

Waveform :

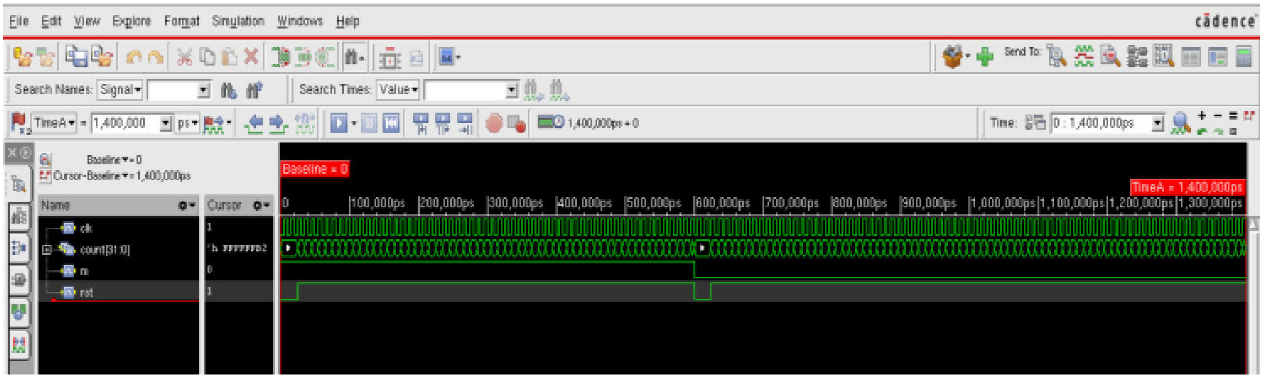


Figure No1.21: Simulation Waveform Window

The procedure for Simulation and Synthesis remains same as mentioned earlier.

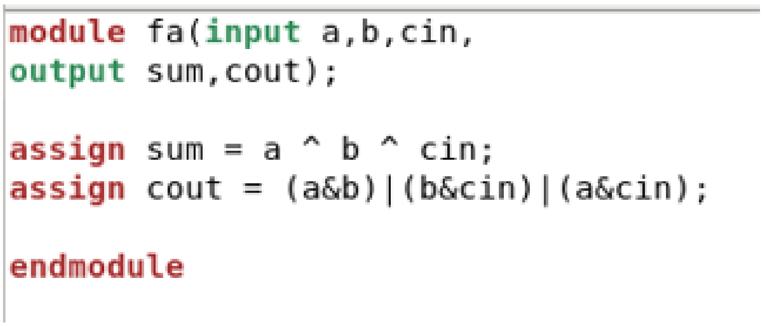
- **Post Lab Task**
  - Can synthesis be possible without Constraints?
  - How constraints are important for timing analysis.

Verilog code for Full adder:

You must perform the function simulation using NCLaunch (Already discuss in previous experiments). This involves three stages :

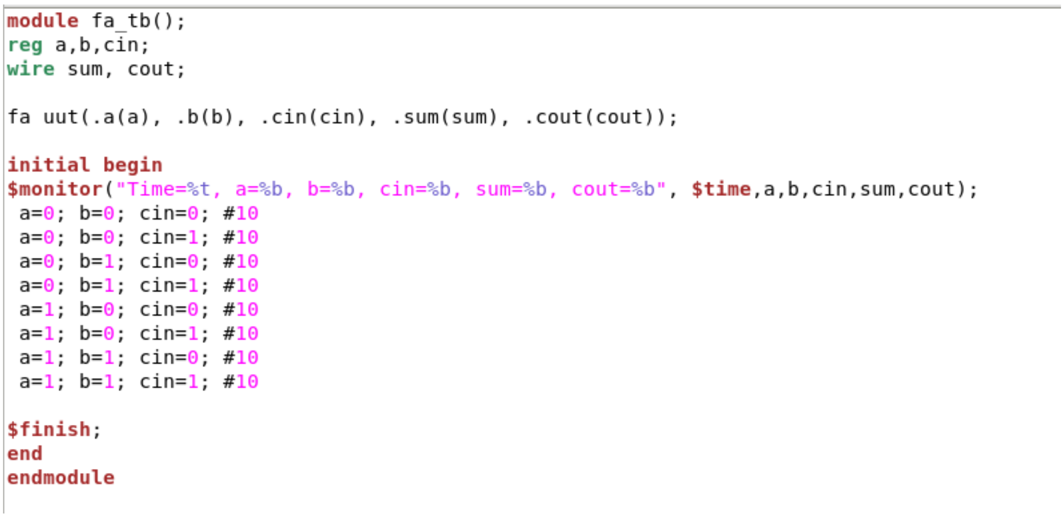
- Step 1: Compilation- Process to check the correct Verilog language syntax and usage
- Step 2: Elaboration- To check the port connections in hierarchical design
- Step 3: Simulation- Simulate with the given test vectors over a period of time to observe the output behavior.

  Full Adder Verilog Code:



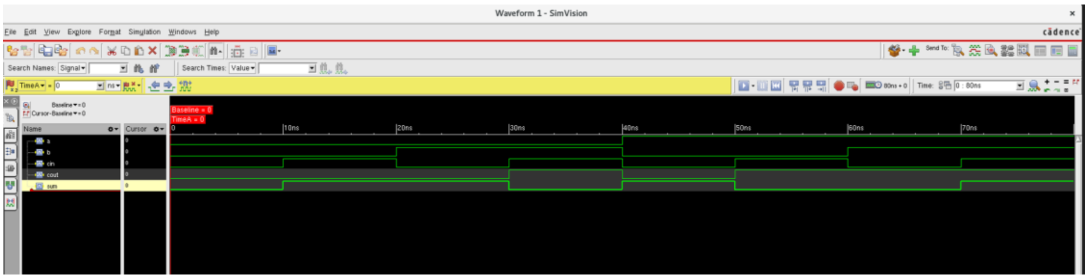
```
module fa(input a,b,cin,
output sum,cout);

assign sum = a ^ b ^ cin;
assign cout = (a&b)|(b&cin)|(a&cin);

endmodule
```

Test Bench for Full adder:



```
module fa_tb();
reg a,b,cin;
wire sum, cout;

fa uut(.a(a), .b(b), .cin(cin), .sum(sum), .cout(cout));

initial begin
$monitor("Time=%t, a=%b, b=%b, cin=%b, sum=%b, cout=%b", $time,a,b,cin,sum,cout);
 a=0; b=0; cin=0; #10
 a=0; b=0; cin=1; #10
 a=0; b=1; cin=0; #10
 a=0; b=1; cin=1; #10
 a=1; b=0; cin=0; #10
 a=1; b=0; cin=1; #10
 a=1; b=1; cin=0; #10
 a=1; b=1; cin=1; #10

$finish;
end
endmodule
```

Waveform:



Repeat the above steps for synthesis for full adder.

Exercise problem:

1. Perform the above exercises with corner analysis includes fast.
2. Write a verilog code and performed synthesis for MOD 10 counter.
3. To write a verilog code for 4bit Adder and verify the functionality using Test bench and synthesize, Analyse Reports and Netlist, Critical Path and Max Operating Frequency.
   - Please note that the adder does not contain any clock, SDC can be skipped as an Input, but you have to consider the SDC during the synthesis.