# ASIC Homework – 2

Name : J. Prashanth

Roll number : IMT2019507

Note : All the .tcl, verilog, excel files and data folders can be found along with this pdf submission. Please refer to the file in case the screenshots are too small / not readable. All the script files are well commented, please go through them for code explanations. The questions are in no way altered to get a positive slack.

## (A) sel_sense :

### Description :

"sel_sense" is a simple combinational circuit with two inputs and one output. A plain-English behavioral description of the logic is as follows:

1. The circuit has one output, B, and two inputs, A and S

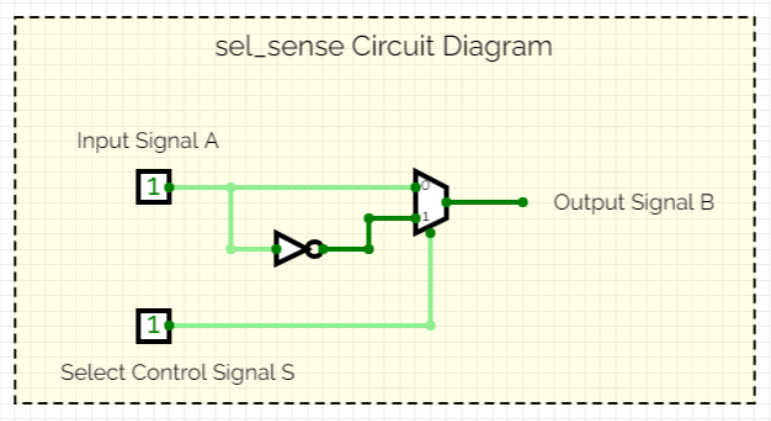
2. The output B is equal to input A if S=0 or is equal to the inverse of A if S = 1.

### (1) Behavioral Verilog code for sel_sense :

change.log | sel_sense.v

```
1  module sel_sense (A, S, B);
2
3      input A, S;
4      output B;
5
6      assign B = S?~A:A;
7
8  endmodule
9
```

(2) Schematic of circuit that implements sel_sense :

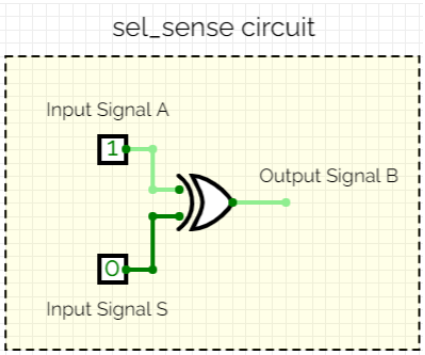



(3) Truth table of sel_sense :

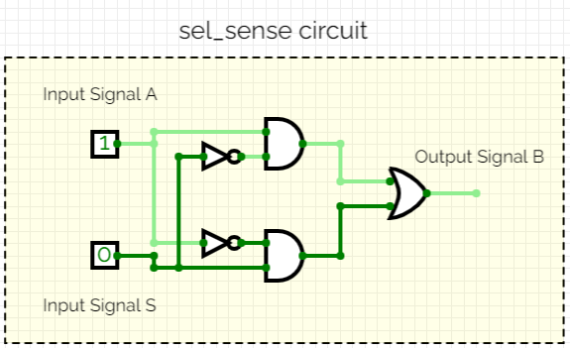
| S | A | B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

| S \ A | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 0 |

Making a K-map, we can see that, B = A(~S) + S(~A) = A ⊕ S,




( or )



(4) Synthesize sel_sense using Skywater HS and MS libraries, netlist file results and synthesis scripts :

(ss, 150C, 1.6V - pvt)

The sel_sense_hs_c.tcl file : (Synthesis file for HS library)

sel_sense_ms_c.tcl | sel_sense_hs_c.tcl

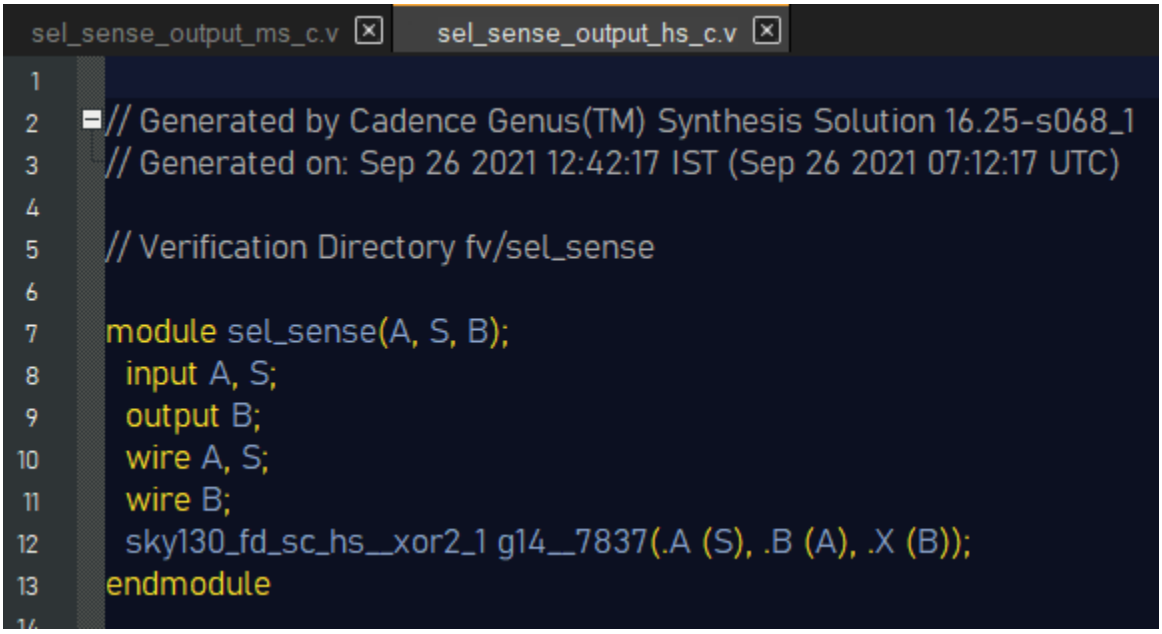
```
#rm designs/*

# reading the sel_sense.v verilog file
read_hdl /home/cad5/Hw-2/sel_sense.v

# setting the .lib file to be used
set_attr library /home/cad5/PDK/sky130_fd_sc_hs/Liberty/sky130_fd_sc_hs__ss_150C_1v60.lib

elaborate
set_attr syn_generic_effort high
syn_gen
syn_map

# setting the max delay from input pins A, S to output pin B to be 50 ps
set_max_delay 0.05 -from A -to B
set_max_delay 0.05 -from S -to B

# setting the load capacitance of output pin B to 0.00277 pF
set_load 0.0027700000 B

set_attr syn_opt_effort high
syn_opt

# writing the obtained verilog file into sel_sense_output_hs_c.v file
write_hdl > sel_sense_output_hs_c.v

```

The sel_sense_ms_c.tcl file : (Synthesis file for MS library)

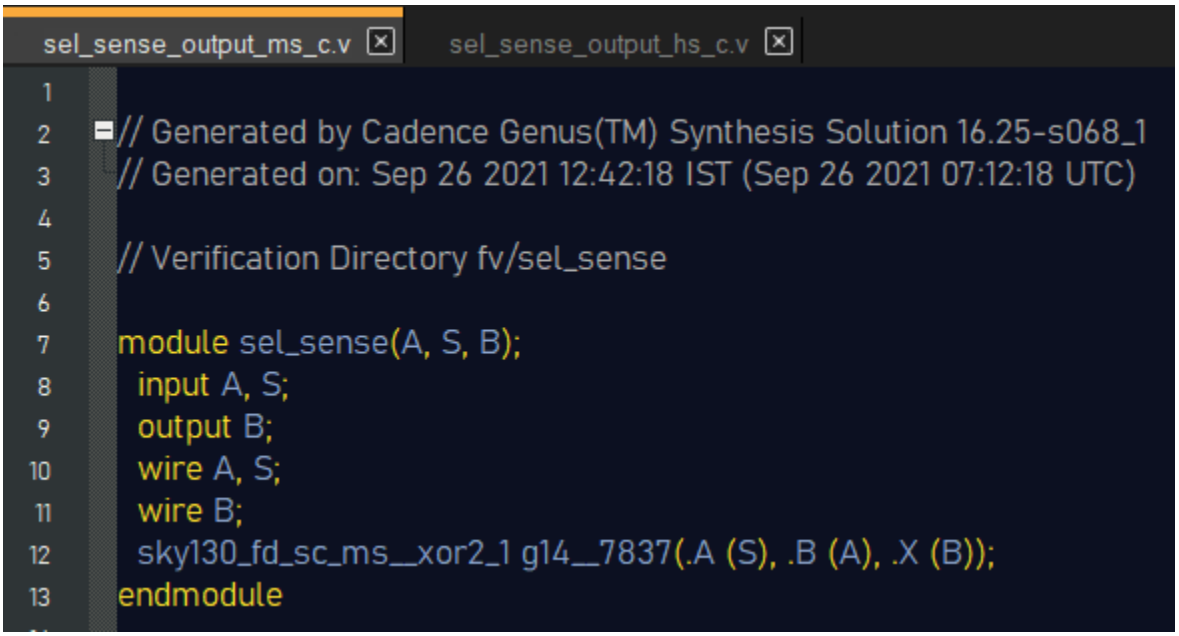
sel_sense_ms_c.tcl | sel_sense_hs_c.tcl

```
#rm designs/*

# reading the sel_sense.v verilog file
read_hdl /home/cad5/Hw-2/sel_sense.v

# setting the .lib file to be used
set_attr library /home/cad5/PDK/sky130_fd_sc_ms/Liberty/sky130_fd_sc_ms__ss_150C_1v60.lib

elaborate
set_attr syn_generic_effort high
syn_gen
syn_map

# setting the max delay from input pins A, S to output pin B to be 50 ps
set_max_delay 0.05 -from A -to B
set_max_delay 0.05 -from S -to B

# setting the load capacitance of output pin B to 0.00297 pF
set_load 0.0029700000 B

set_attr syn_opt_effort high
syn_opt

# writing the obtained verilog file into sel_sense_output_ms_c.v file
write_hdl > sel_sense_output_ms_c.v

```

The sel_sense_output_hs_c.v file : (The output HDL file for HS lib)




The sel_sense_output_ms_c.v file : (The output HDL file for MS lib)



(5) Synthesize sel_sense using Skywater HS and MS libraries, netlist file results and synthesis scripts (without using XOR and XNOR gates) :

(ss, 150C, 1.6V - pvt)

The sel_sense_hs_d.tcl file : (Synthesis file for HS library)

sel_sense_hs_d.tcl | sel_sense_ms_d.tcl

```
#rm designs/*

# reading the sel_sense.v verilog file
read_hdl /home/cad5/Hw-2/sel_sense.v

# setting the .lib file to be used
set_attr library /home/cad5/PDK/sky130_fd_sc_hs/Liberty/sky130_fd_sc_hs__ss_150C_1v60.lib

elaborate
set_attr syn_generic_effort high
syn_gen

# setting the don't use condition on the xor and xnor gates obtained after using the grep commands
set_dont_use sky130_fd_sc_hs__xor2_1 sky130_fd_sc_hs__xor2_2 sky130_fd_sc_hs__xor2_4 sky130_fd_sc_hs__xnor2_1 sky130_fd_sc_hs__xnor2_2 sky130_fd_sc_hs__xor2_4

syn_map

# setting the max delay from input pins A, S to output pin B to be 50 ps
set_max_delay 0.05 -from A -to B
set_max_delay 0.05 -from S -to B

# setting the load capacitance of output pin B to 0.00277 pF
set_load 0.0027700000 B

set_attr syn_opt_effort high
syn_opt

# writing the obtained verilog file into sel_sense_output_hs_d.v file
write_hdl > sel_sense_output_hs_d.v
```

The sel_sense_ms_d.tcl file : (Synthesis file for MS library)

sel_sense_hs_d.tcl | sel_sense_ms_d.tcl

```
#rm designs/*

# reading the sel_sense.v verilog file
read_hdl /home/cad5/Hw-2/sel_sense.v

# setting the .lib file to be used
set_attr library /home/cad5/PDK/sky130_fd_sc_ms/Liberty/sky130_fd_sc_ms__ss_150C_1v60.lib

elaborate
set_attr syn_generic_effort high
syn_gen

# setting the don't use condition on the xor and xnor gates obtained after using the grep commands
set_dont_use sky130_fd_sc_ms__xor2_1 sky130_fd_sc_ms__xor2_2 sky130_fd_sc_ms__xor2_4 sky130_fd_sc_ms__xnor2_1 sky130_fd_sc_ms__xnor2_2 sky130_fd_sc_ms__xor2_4

syn_map

# setting the max delay from input pins A, S to output pin B to be 50 ps
set_max_delay 0.05 -from A -to B
set_max_delay 0.05 -from S -to B

# setting the load capacitance of output pin B to 0.00297 pF
set_load 0.0029700000 B

set_attr syn_opt_effort high
syn_opt

# writing the obtained verilog file into sel_sense_output_ms_d.v file
write_hdl > sel_sense_output_ms_d.v
```

The sel_sense_output_hs_d.v file : (The output HDL file for HS lib)

sel_sense_output_ms_d.v | sel_sense_output_hs_d.v

```
1
2   // Generated by Cadence Genus(TM) Synthesis Solution 16.25-s068_1
3   // Generated on: Sep 26 2021 12:42:19 IST (Sep 26 2021 07:12:19 UTC)
4
5   // Verification Directory fv/sel_sense
6
7   module sel_sense(A, S, B);
8    input A, S;
9    output B;
10   wire A, S;
11   wire B;
12   wire n_13, n_14, n_15, n_16;
13   sky130_fd_sc_hs__nand2_1 g16(.A (n_14), .B (n_16), .Y (B));
14   sky130_fd_sc_hs__nand2_2 g20(.A (n_13), .B (S), .Y (n_14));
15   sky130_fd_sc_hs__inv_8 g21(.A (A), .Y (n_13));
16   sky130_fd_sc_hs__nand2_2 g17(.A (n_15), .B (A), .Y (n_16));
17   sky130_fd_sc_hs__inv_16 g19(.A (S), .Y (n_15));
18  endmodule
19
```

The sel_sense_output_ms_d.v file : (The output HDL file for MS lib)

sel_sense_output_ms_d.v | sel_sense_output_hs_d.v

```
1
2   // Generated by Cadence Genus(TM) Synthesis Solution 16.25-s068_1
3   // Generated on: Sep 26 2021 12:42:20 IST (Sep 26 2021 07:12:20 UTC)
4
5   // Verification Directory fv/sel_sense
6
7   module sel_sense(A, S, B);
8    input A, S;
9    output B;
10   wire A, S;
11   wire B;
12   wire n_17, n_18, n_19, n_20;
13   sky130_fd_sc_ms__nand2_1 g16(.A (n_18), .B (n_20), .Y (B));
14   sky130_fd_sc_ms__nand2_2 g17(.A (n_17), .B (A), .Y (n_18));
15   sky130_fd_sc_ms__inv_8 g19(.A (S), .Y (n_17));
16   sky130_fd_sc_ms__nand2_2 g18(.A (n_19), .B (S), .Y (n_20));
17   sky130_fd_sc_ms__inv_16 g20(.A (A), .Y (n_19));
18  endmodule
19
```

(6) For this question, first we write a synthesis file for each HS and MS libraries, the tcl files written for those libraries are “sel_sense_hs_6.tcl” and “sel_sense_ms_6.tcl” respectively. All the data acquired from the tcl scripts are in the “hs_S0_data”, “hs_S1_data”, “ms_S0_data” and “ms_S1_data” folders. I have gathered the data and put them in an excel sheet “Graph_data.xlsx”. The screenshots of the graphs and data from the excel sheet are also given below.

Synthesis file used for getting data for HS library :

sel_sense_ms_6.tcl | sel_sense_hs_6.tcl

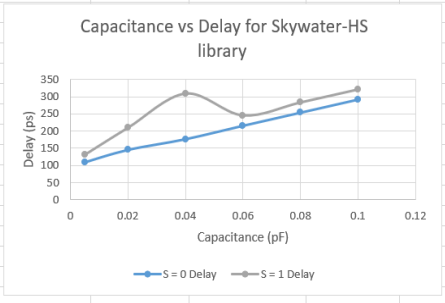
```
#rm designs/*

# reading the sel_sense.v verilog file
read_hdl /home/cad5/Hw-2/sel_sense.v

# setting the .lib file to be used
set_attr library /home/cad5/PDK/sky130_fd_sc_hs/Liberty/sky130_fd_sc_hs__ss_150C_1v60.lib

elaborate
set_attr syn_generic_effort high
syn_gen
syn_map

# setting the max delay from input pins A, S to output pin B to be 50 ps
set_max_delay 0.05 -from A -to B
set_max_delay 0.05 -from S -to B

# for the value of S = 0
set_case_analysis 0 S

# seting the value of input's transition to 40 ps
set_input_transition 0.04 A

# the load capacitance of output pin B is incremented in values of 0.02 from 0.005 to 0.1
# started with 0.02 as it was easier for the file naming and representation

foreach in collection itr [list 0.005 0.02 0.04 0.06 0.08 0.1] {
set_load $itr -pin_load B

set_attr syn_opt_effort high
syn_opt

report_timing -from A -to B > /home/cad5/Hw-2/hs_S0_data/$itr+_hs_s0_report.txt
}


# for the value of S = 1

set_case_analysis 1 S

# the load capacitance of output pin B is incremented in values of 0.02 from 0.005 to 0.1
# started with 0.02 as it was easier for the file naming and representation

foreach in collection itr [list 0.005 0.02 0.04 0.06 0.08 0.1] {
set_load $itr -pin_load B

set_attr syn_opt_effort high
syn_opt

report_timing -from A -to B > /home/cad5/Hw-2/hs_S1_data/$itr+_hs_s1_report.txt
}
```

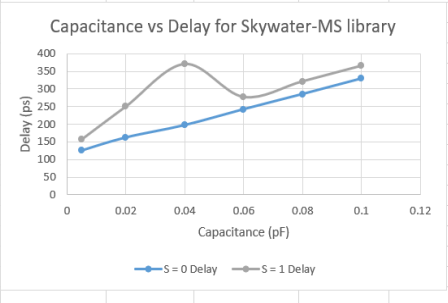
Synthesis file used for getting data for MS library :

sel_sense_ms_6.tcl

```
#rm designs/*

# reading the sel_sense.v verilog file
read_hdl /home/cad5/Hw-2/sel_sense.v

# setting the .lib file to be used
set_attr library /home/cad5/PDK/sky130_fd_sc_ms/Liberty/sky130_fd_sc_ms__ss_150C_1v60.lib

elaborate
set_attr syn_generic_effort high
syn_gen
syn_map

# setting the max delay from input pins A, S to output pin B to be 50 ps
set_max_delay 0.05 -from A -to B
set_max_delay 0.05 -from S -to B

# for the value of S = 0
set_case_analysis 0 S

# seting the value of input's transition to 40 ps
set_input_transition 0.04 A

# the load capacitance of output pin B is incremented in values of 0.02 from 0.005 to 0.1
# started with 0.02 as it was easier for the file naming and representation

foreach in collection itr [list 0.005 0.02 0.04 0.06 0.08 0.1] {
set_load $itr -pin_load B

set_attr syn_opt_effort high
syn_opt

report_timing -from A -to B > /home/cad5/Hw-2/ms_S0_data/$itr+_ms_s0_report.txt
}


# for the value of S = 1

set_case_analysis 1 S

# the load capacitance of output pin B is incremented in values of 0.02 from 0.005 to 0.1
# started with 0.02 as it was easier for the file naming and representation

foreach in collection itr [list 0.005 0.02 0.04 0.06 0.08 0.1] {
set_load $itr -pin_load B

set_attr syn_opt_effort high
syn_opt

report_timing -from A -to B > /home/cad5/Hw-2/ms_S1_data/$itr+_ms_s1_report.txt
}
```

| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | HS Library Data from hs_S0_data and hs_S1_data files | | | | | MS Library Data from ms_S0_data and hs_S1_data files | | | | |
| 2 | S = 0 | | S = 1 | | | S = 0 | | S = 1 | | |
| 3 | Capacitance | S = 0 Delay | Capacitance | S = 1 Delay | | Capacitance | S = 0 Delay | Capacitance | S = 1 Delay | |
| 4 | | | | | | | | | | |
| 5 | 0.005 | 110 | 0.005 | 131 | | 0.005 | 125 | 0.005 | 157 | |
| 6 | 0.02 | 146 | 0.02 | 210 | | 0.02 | 162 | 0.02 | 250 | |
| 7 | 0.04 | 177 | 0.04 | 310 | | 0.04 | 197 | 0.04 | 370 | |
| 8 | 0.06 | 216 | 0.06 | 246 | | 0.06 | 242 | 0.06 | 277 | |
| 9 | 0.08 | 254 | 0.08 | 284 | | 0.08 | 285 | 0.08 | 321 | |
| 10 | 0.1 | 292 | 0.1 | 322 | | 0.1 | 329 | 0.1 | 365 | |
| 11 | | | | | | | | | | |



(B) Buffers in Standard Cell Library :

| Cell Name | Name of Input Pin | Capacitance of Input Pin (pF) |
|---|---|---|
| BUFX2 (from Cadence RAK, /max/slow.lib) | A | 0.000601595 |
| BUF_X2 (from Nangate_15nm_OCL, NanGate_15nm_OCL_slow_conditional_nldm.lib) | I | 0.000839939 |
| BUF_X2 (from Nangate_15nm_OCL, NanGate_15nm_OCL_slow_conditional_ecsm.lib) | I | 0.000839939 |
| BUF_X2 (from Nangate_15nm_OCL, NanGate_15nm_OCL_slow_conditional_ccs.lib) | I | 0.000839939 |

Screenshots :

First, we identify the naming in the .lib files, after trial and error we find out that buffer files of strength “n” are named like BUF_X{n}. So now we use the grep command to find out the capacitances.

```
prashanth@DESKTOP-C1PFNS2:/mnt/c/Users/jonna/Documents/PDK/Cadence_RAK_1
8.1_blockImplementation/LIBS/lib/max$ grep -A 10 "cell (BUFX2" slow.lib
cell (BUFX2) {
  area : 0.0;
  cell_leakage_power : 0.0852604;
  rail_connection( VDD, RAIL_VDD );
  rail_connection( VSS, RAIL_VSS );
  pin(A)  {
    direction : input;
    input_signal_level : RAIL_VDD;
    capacitance : 0.000601595;
    rise_capacitance : 0.000601052;
    fall_capacitance : 0.000601595;
--
cell (BUFX20) {
  area : 0.0;
  cell_leakage_power : 0.840344;
  rail_connection( VDD, RAIL_VDD );
  rail_connection( VSS, RAIL_VSS );
  pin(A)  {
    direction : input;
    input_signal_level : RAIL_VDD;
    capacitance : 0.00359149;
    rise_capacitance : 0.00358933;
    fall_capacitance : 0.00359149;
prashanth@DESKTOP-C1PFNS2:/mnt/c/Users/jonna/Documents/PDK/Cadence_RAK_1
8.1_blockImplementation/LIBS/lib/max$
```

```
prashanth@DESKTOP-C1PFNS2:/mnt/c/Users/jonna/Documents/PDK/NanGate_15nm_
OCL_v0.1_2014_06_Apache.A/front_end/timing_power_noise/NLDM$ ls
NanGate_15nm_OCL_fast_conditional_nldm.lib
NanGate_15nm_OCL_low_temp_conditional_nldm.lib
NanGate_15nm_OCL_slow_conditional_nldm.lib
NanGate_15nm_OCL_typical_conditional_nldm.lib
NanGate_15nm_OCL_worst_low_conditional_nldm.lib
prashanth@DESKTOP-C1PFNS2:/mnt/c/Users/jonna/Documents/PDK/NanGate_15nm_
OCL_v0.1_2014_06_Apache.A/front_end/timing_power_noise/NLDM$ grep -A 10
"  cell (BUF" NanGate_15nm_OCL_slow_conditional_nldm.lib
  cell (BUF_X1) {

        drive_strength  : 1;

        area                    : 0.245760;
        pg_pin(VDD) {
                voltage_name : VDD;
                pg_type      : primary_power;
        }
        pg_pin(VNW) {
                voltage_name : VNW;
--
  cell (BUF_X2) {

        drive_strength  : 2;

        area                    : 0.245760;
        pg_pin(VDD) {
                voltage_name : VDD;
                pg_type      : primary_power;
        }
        pg_pin(VNW) {
                voltage_name : VNW;
--
  cell (BUF_X4) {
```

```
prashanth@DESKTOP-C1PFNS2:/mnt/c/Users/jonna/Documents/PDK/NanGate_15nm_OCL_v0.1_2014_0
6_Apache.A/front_end/timing_power_noise/NLDM$ grep -A 40
"  cell (BUF_X2)" NanGate_15nm_OCL_slow_conditional_nldm.lib
  cell (BUF_X2) {

       drive_strength  : 2;

       area                    : 0.245760;
       pg_pin(VDD) {
               voltage_name : VDD;
               pg_type      : primary_power;
       }
       pg_pin(VNW) {
               voltage_name : VNW;
               pg_type      : nwell;
       }
       pg_pin(VPW) {
               voltage_name : VPW;
               pg_type      : pwell;
       }
       pg_pin(VSS) {
               voltage_name : VSS;
               pg_type      : primary_ground;
       }


       cell_leakage_power      : 819445.222351;

       leakage_power () {
               when           : "!I";
               value          : 884643.328180;
       }
       leakage_power () {
               when           : "I";
               value          : 754247.116522;
       }

       pin (I) {

               direction              : input;
               related_power_pin      : "VDD";
               related_ground_pin     : "VSS";
               capacitance            : 0.839939;
               fall_capacitance       : 0.841850;
```

(C) mux_comparator :

(1) The verilog code for mux_comparator is given below :

mux_comparator.v

```
1  module mux_comparator (
2     input [2:0] x,
3     input [2:0] y,
4     output reg [2:0] z
5     );
6
7     always @(x or y)
8     begin
9
10     if (x < y)
11       z = x;
12
13     else
14       z = y;
15
16    end
17
18 endmodule
19
```

I have written the test bench for the above verilog file and obtained the following results to validate the working of the mux_comparator by taking few random cases, (testbench file : ”mux_comparator_tb.v”)

```
F:\Semester Courses\Physical Design of ASICs\Assignments\Homework-2\Hw-2i
verilog -o result mux_comparator_tb.v

F:\Semester Courses\Physical Design of ASICs\Assignments\Homework-2\Hw-2v
vp result
VCD info: dumpfile mux_comparator.vcd opened for output.
 x value = 3 , y value = 0 --> z value = 0
 x value = 4 , y value = 1 --> z value = 1
 x value = 1 , y value = 2 --> z value = 1
 x value = 4 , y value = 4 --> z value = 4
 x value = 1 , y value = 0 --> z value = 0
 x value = 6 , y value = 5 --> z value = 5
 x value = 4 , y value = 6 --> z value = 4
 x value = 6 , y value = 3 --> z value = 3

F:\Semester Courses\Physical Design of ASICs\Assignments\Homework-2\Hw-2>
```

(2) .sdc files for,

Cadence RAK lib : (file : “mux_comparator_cadence_RAK.sdc”)

mux_comparator_cadence_RAK.sdc | mux_comparator_nangate_OCL.sdc

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

#  Created by Genus(TM) Synthesis Solution 16.25-s068_1 on Sun Sep 26 12:42:36 +0530 2021

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

set sdc_version 1.7

set_units -capacitance 1000.0fF
set_units -time 1000.0ps

# Set the current design
current_design mux_comparator

set_load -pin_load 0.0006 [get_ports {z[2]}]
set_load -pin_load 0.0006 [get_ports {z[1]}]
set_load -pin_load 0.0006 [get_ports {z[0]}]
set_max_delay 0.25 -from [list \
 [get_ports {x[2]}] \
 [get_ports {x[1]}] \
 [get_ports {x[0]}] \
 [get_ports {y[2]}] \
 [get_ports {y[1]}] \
 [get_ports {y[0]}] ] -to [list \
 [get_ports {z[2]}] \
 [get_ports {z[1]}] \
 [get_ports {z[0]}] ]
set_clock_gating_check -setup 0.0
set_input_transition 0.1 [get_ports {x[2]}]
set_input_transition 0.1 [get_ports {x[1]}]
set_input_transition 0.1 [get_ports {x[0]}]
set_input_transition 0.1 [get_ports {y[2]}]
set_input_transition 0.1 [get_ports {y[1]}]
set_input_transition 0.1 [get_ports {y[0]}]
set_wire_load_mode "enclosed"
set_dont_use [get_lib_cells gpdk045bc/CLKAND2X2]
set_dont_use [get_lib_cells gpdk045bc/CLKAND2X6]
set_dont_use [get_lib_cells gpdk045bc/CLKBUFX12]
set_dont_use [get_lib_cells gpdk045bc/CLKBUFX16]
set_dont_use [get_lib_cells gpdk045bc/CLKBUFX2]
set_dont_use [get_lib_cells gpdk045bc/CLKBUFX20]
set_dont_use [get_lib_cells gpdk045bc/CLKBUFX4]
set_dont_use [get_lib_cells gpdk045bc/CLKBUFX6]
set_dont_use [get_lib_cells gpdk045bc/CLKBUFX8]
set_dont_use [get_lib_cells gpdk045bc/CLKINVX1]
set_dont_use [get_lib_cells gpdk045bc/CLKINVX12]
set_dont_use [get_lib_cells gpdk045bc/CLKINVX16]
set_dont_use [get_lib_cells gpdk045bc/CLKINVX2]
set_dont_use [get_lib_cells gpdk045bc/CLKINVX20]
set_dont_use [get_lib_cells gpdk045bc/CLKINVX3]
set_dont_use [get_lib_cells gpdk045bc/CLKINVX4]
set_dont_use [get_lib_cells gpdk045bc/CLKINVX6]
set_dont_use [get_lib_cells gpdk045bc/CLKMX2X12]
set_dont_use [get_lib_cells gpdk045bc/CLKXOR2X1]
set_dont_use [get_lib_cells gpdk045bc/CLKXOR2X4]
set_dont_use [get_lib_cells gpdk045bc/HOLDX1]
set_dont_use [get_lib_cells gpdk045bc/CLKAND2X12]
set_dont_use [get_lib_cells gpdk045bc/CLKAND2X3]
```

```
set_dont_use [get_lib_cells gpdk045bc/CLKAND2X3]
set_dont_use [get_lib_cells gpdk045bc/CLKBUFX3]
set_dont_use [get_lib_cells gpdk045bc/CLKINVX8]
set_dont_use [get_lib_cells gpdk045bc/CLKMX2X6]
set_dont_use [get_lib_cells gpdk045bc/CLKXOR2X2]
set_dont_use [get_lib_cells gpdk045bc/CLKAND2X8]
set_dont_use [get_lib_cells gpdk045bc/CLKMX2X2]
set_dont_use [get_lib_cells gpdk045bc/CLKMX2X3]
set_dont_use [get_lib_cells gpdk045bc/CLKMX2X4]
set_dont_use [get_lib_cells gpdk045bc/CLKXOR2X8]
set_dont_use [get_lib_cells gpdk045bc/CLKMX2X8]
set_dont_use [get_lib_cells gpdk045bc/CLKAND2X4]
```

For Nangate OCL lib : (file : “mux_comparator_nangate_OCL.sdc”)

mux_comparator_cadence_RAK.sdc | mux_comparator_nangate_OCL.sdc

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

#  Created by Genus(TM) Synthesis Solution 16.25-s068_1 on Sun Sep 26 12:42:32 +0530 2021

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

set sdc_version 1.7

set_units -capacitance 1.0fF
set_units -time 1.0ps

# Set the current design
current_design mux_comparator

set_load -pin_load 0.0 [get_ports {z[2]}]
set_load -pin_load 0.0 [get_ports {z[1]}]
set_load -pin_load 0.0 [get_ports {z[0]}]
set_max_delay 0 -from [list \
 [get_ports {x[2]}] \
 [get_ports {x[1]}] \
 [get_ports {x[0]}] \
 [get_ports {y[2]}] \
 [get_ports {y[1]}] \
 [get_ports {y[0]}] ] -to [list \
 [get_ports {z[2]}] \
 [get_ports {z[1]}] \
 [get_ports {z[0]}] ]
set_clock_gating_check -setup 0.0
set_input_transition 0.1 [get_ports {x[2]}]
set_input_transition 0.1 [get_ports {x[1]}]
set_input_transition 0.1 [get_ports {x[0]}]
set_input_transition 0.1 [get_ports {y[2]}]
set_input_transition 0.1 [get_ports {y[1]}]
set_input_transition 0.1 [get_ports {y[0]}]
set_wire_load_mode "enclosed"
set_dont_use [get_lib_cells NanGate_15nm_OCL/ANTENNA]
set_dont_use [get_lib_cells NanGate_15nm_OCL/FILLTIE]
set_dont_use [get_lib_cells NanGate_15nm_OCL/FILL_X1]
set_dont_use [get_lib_cells NanGate_15nm_OCL/FILL_X2]
set_dont_use [get_lib_cells NanGate_15nm_OCL/FILL_X4]
set_dont_use [get_lib_cells NanGate_15nm_OCL/FILL_X8]
set_dont_use [get_lib_cells NanGate_15nm_OCL/FILL_X16]
set_dont_use [get_lib_cells NanGate_15nm_OCL/TIEH]
set_dont_use [get_lib_cells NanGate_15nm_OCL/TIEL]

```

(3) The resulting netlists are,

For Cadence RAK library, (file : mux_comparator_cadence_RAK.v)

mux_comparator_cadence_RAK.v | mux_comparator_nangate_OCL.v

```
1
2  // Generated by Cadence Genus(TM) Synthesis Solution 16.25-s068_1
3  // Generated on: Sep 26 2021 12:42:36 IST (Sep 26 2021 07:12:36 UTC)
4
5  // Verification Directory fv/mux_comparator
6
7  module mux_comparator(x, y, z);
8    input [2:0] x, y;
9    output [2:0] z;
10   wire [2:0] x, y;
11   wire [2:0] z;
12   wire n_28, n_60, n_86, n_87, n_88, n_89, n_93, n_97;
13   wire n_98, n_99, n_100, n_101, n_102, n_105, n_106, n_107;
14   wire n_108, n_111, n_112, n_118;
15   INVXL g185(.A (y[0]), .Y (n_28));
16   AND2X6 g194(.A (x[2]), .B (y[2]), .Y (z[2]));
17   OAI21XL g1067(.A0 (n_88), .A1 (n_89), .B0 (n_118), .Y (z[1]));
18   NOR2X6 g1070(.A (y[1]), .B (n_87), .Y (n_88));
19   NOR2X8 g1073(.A (x[2]), .B (n_86), .Y (n_87));
20   INVX20 g1079(.A (y[2]), .Y (n_86));
21   INVX20 g1080(.A (x[1]), .Y (n_89));
22   NOR2X8 g1076(.A (x[0]), .B (x[1]), .Y (n_93));
23   AOI21X4 g1069(.A0 (n_97), .A1 (n_99), .B0 (n_101), .Y (n_102));
24   NOR2X8 g1074(.A (n_89), .B (n_60), .Y (n_97));
25   INVX20 g1078(.A (x[0]), .Y (n_60));
26   NAND2X8 g1072(.A (y[2]), .B (n_98), .Y (n_99));
27   INVX20 g1081_dup(.A (x[2]), .Y (n_98));
28   NOR2X2 g1077(.A (y[2]), .B (n_100), .Y (n_101));
29   INVX4 g1081(.A (x[2]), .Y (n_100));
30   NAND2X2 g92(.A (n_105), .B (n_102), .Y (n_106));
31   INVX2 g95(.A (n_60), .Y (n_105));
32   NOR2X4 g91(.A (n_112), .B (n_107), .Y (n_108));
33   INVX3 g93(.A (n_102), .Y (n_107));
34   OAI22X2 g33(.A0 (n_112), .A1 (n_106), .B0 (n_28), .B1 (n_108), .Y
35     (z[0]));
36   NOR2X8 g34(.A (n_93), .B (n_111), .Y (n_112));
37   INVX8 g36(.A (n_88), .Y (n_111));
38   NAND3BX4 g2(.AN (y[2]), .B (y[1]), .C (x[2]), .Y (n_118));
39 endmodule
40
```

For Nangate OCL library, (file : mux_comparator_nangate_OCL.v)

mux_comparator_cadence_RAK.v | mux_comparator_nangate_OCL.v

```
// Generated by Cadence Genus(TM) Synthesis Solution 16.25-s068_1
// Generated on: Sep 26 2021 12:42:32 IST (Sep 26 2021 07:12:32 UTC)

// Verification Directory fv/mux_comparator

module mux_comparator(x, y, z);
 input [2:0] x, y;
 output [2:0] z;
 wire [2:0] x, y;
 wire [2:0] z;
 wire n_156, n_231, n_292, n_296, n_297, n_298, n_305, n_306;
 wire n_307, n_313, n_314, n_315, n_318, n_320, n_321, n_322;
 wire n_340, n_341, n_348, n_349, n_350, n_368, n_369, n_370;
 wire n_371, n_372, n_373, n_374, n_375, n_376;
 NOR2_X1 g4857(.A1 (n_374), .A2 (y[1]), .ZN (n_156));
 INV_X2 g96(.I (y[1]), .ZN (n_231));
 OAI21_X1 g23(.A1 (n_348), .A2 (n_318), .B (n_341), .ZN (z[1]));
 OAI21_X1 g106(.A1 (n_231), .A2 (x[0]), .B (x[2]), .ZN (n_292));
 NAND2_X2 g103(.A1 (n_297), .A2 (n_370), .ZN (n_298));
 OAI21_X2 g272(.A1 (n_296), .A2 (x[1]), .B (x[0]), .ZN (n_297));
 INV_X4 g108(.I (y[1]), .ZN (n_296));
 OAI21_X2 g276(.A1 (x[0]), .A2 (x[1]), .B (n_368), .ZN (n_305));
 NOR2_X2 g71(.A1 (n_374), .A2 (n_373), .ZN (n_306));
 INV_X4 g72(.I (y[2]), .ZN (n_307));
 NAND2_X1 g70(.A1 (n_307), .A2 (x[2]), .ZN (n_313));
 NOR2_X1 g36(.A1 (n_314), .A2 (n_315), .ZN (z[2]));
 NOR2_X1 g38(.A1 (n_298), .A2 (n_156), .ZN (n_314));
 NAND2_X1 g37(.A1 (x[2]), .A2 (y[2]), .ZN (n_315));
 OAI21_X1 g35(.A1 (n_298), .A2 (n_156), .B (y[1]), .ZN (n_318));
 NAND2_X1 g227(.A1 (n_322), .A2 (n_349), .ZN (z[0]));
 OAI21_X1 g228(.A1 (n_350), .A2 (n_321), .B (y[0]), .ZN (n_322));
 INV_X1 g230(.I (n_320), .ZN (n_321));
 OAI21_X2 g231(.A1 (n_306), .A2 (x[2]), .B (n_307), .ZN (n_320));
 NAND4_X1 g82(.A1 (n_369), .A2 (n_292), .A3 (n_340), .A4 (n_313), .ZN
   (n_341));
 NOR2_X1 g83(.A1 (n_368), .A2 (n_374), .ZN (n_340));
 AND2_X1 g46(.A1 (n_376), .A2 (y[2]), .Z (n_348));
 NAND4_X1 g45(.A1 (n_320), .A2 (n_376), .A3 (n_305), .A4 (x[0]), .ZN
   (n_349));
 NAND2_X1 g47(.A1 (n_376), .A2 (n_305), .ZN (n_350));
 NOR2_X2 g289(.A1 (y[1]), .A2 (y[2]), .ZN (n_368));
 OR3_X1 g2(.A1 (n_373), .A2 (n_374), .A3 (y[2]), .Z (n_369));
 NAND2_X2 g94(.A1 (n_372), .A2 (n_375), .ZN (n_376));
 NOR2_X2 g290(.A1 (n_370), .A2 (n_371), .ZN (n_372));
 INV_X4 g291(.I (x[2]), .ZN (n_370));
 NOR2_X2 g292(.A1 (x[1]), .A2 (x[0]), .ZN (n_371));
 OAI21_X2 g293(.A1 (n_373), .A2 (n_374), .B (y[1]), .ZN (n_375));
 INV_X8 g98(.I (x[0]), .ZN (n_373));
 INV_X16 g294(.I (x[1]), .ZN (n_374));
endmodule
```

For part (4) and (5), refer to,

“mux_comparator_RAK_timing_report.txt” file for Cadence RAK library

“mux_comparator_OCL_timing_report.txt” file for Nangate OCL library

(The script files used for generating these are “mux_comparator_RAK.tcl” and “mux_comparator_OCL.tcl”)

I haven’t pasted their screenshots here as there is a lot of data in those files and it would take 4-5 pages for a single library’s timing report file.

What it contains :

It contains timing reports for all the possible paths, (ie, x[0] -> z[0], x[0] -> z[2], y[2] -> z[1], etc.). By looking at each of the timing delays and other data, we can determine which of the paths have the maximum delay (critical path) and the minimum delay, and can hence list out the cells in those paths.

(4) The critical paths are the first generated timing reports in each of the above mentioned .txt files as “report_timing” in general gives max_delay (critical) path.

For Nangate OCL : max delay = 14 ps

Number of cells = 6 cells

For Cadence RAK : max delay = 249 ps

Number of cells = 5 cells

(the screenshots below show the timings and cells in the critical path)

## For Nangate OCL library,

mux_comparator_OCL_timing_report - Notepad

File Edit Format View Help

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 16.25-s068_1
  Generated on:           Sep 26 2021  12:42:32 pm
  Module:                 mux_comparator
  Technology library:     NanGate_15nm_OCL revision 1.0
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

 Pin          Type     Fanout Load Slew Delay Arrival
                            (fF) (ps)  (ps)   (ps)
-------------------------------------------------------------
x[0]    <<<  in port        6 15.9    0    +0       0 R
g292/A2                                   +0       0
g292/ZN      NOR2_X2        1  2.0    2    +2       2 F
g290/A2                                   +0       2
g290/ZN      NOR2_X2        1  1.7    4    +2       4 R
g94/A1                                    +0       4
g94/ZN       NAND2_X2       3  3.6    5    +4       8 F
g47/A1                                    +0       8
g47/ZN       NAND2_X1       1  1.0    3    +2      10 R
g228/A1                                   +0      10
g228/ZN      OAI21_X1       1  1.2    5    +3      13 F
g227/A1                                   +0      13
g227/ZN      NAND2_X1       1  0.0    3    +1      14 R
z[0]    <<<  out port                     +0      14 R
-------------------------------------------------------------
Exception    : 'path_delays/del_1'       0ps
Timing slack :     -14ps (TIMING VIOLATION)
Start-point  : x[0]
End-point    : z[0]
```

## For Cadence RAK library,

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 16.25-s068_1
  Generated on:           Sep 26 2021  12:42:36 pm
  Module:                 mux_comparator
  Technology library:     gpdk045bc
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

  Pin          Type     Fanout Load Slew Delay Arrival
                            (fF) (ps)  (ps)   (ps)
-------------------------------------------------------------
x[0]     <<<  in port        2 20.0  100    +0       0 R
g1078/A                                   +0       0
g1078/Y       INVX20         2  7.6   33   +44      44 F
g1074/B                                   +0      44
g1074/Y       NOR2X8         1  3.2   39   +32      76 R
g1069/A0                                  +0      76
g1069/Y       AOI21X4        2  3.7   74   +67     142 F
g92/B                                     +0     142
g92/Y         NAND2X2        1  1.6   30   +42     184 R
g33/A1                                    +0     184
g33/Y         OAI22X2        1  0.6   71   +65     249 F
z[0]     <<<  out port                    +0     249 F
-------------------------------------------------------------
Exception    : 'path_delays/del_1'     250ps
Timing slack :       1ps
Start-point  : x[0]
End-point    : z[0]
```

(5) For the min delay paths,

| For Cadence RAK, input -> output | Timing delay (in pico seconds) |
|---|---|
| x[0] –> z[0] | 249 ps |
| x[1] –> z[0] | 256 ps |
| x[1] –> z[1] | 109 ps |
| x[2] –> z[0] | 256 ps |
| x[2] –> z[1] | 189 ps |
| x[2] –> z[2] | 128 ps |
| y[0] –> z[0] | 131 ps |
| y[1] –> z[0] | 217 ps |
| y[1] –> z[1] | 161 ps |
| y[2] –> z[0] | 254 ps |
| y[2] –> z[1] | 199 ps |
| y[2] –> z[2] | 122 ps |

The remaining paths do not exist in the circuit. The min delay path is from x[1] -> z[1] as per the table. The number of standard cells are 2.

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 16.25-s068_1
  Generated on:           Sep 26 2021  12:42:36 pm
  Module:                 mux_comparator
  Technology library:     gpdk045bc
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

  Pin          Type     Fanout Load Slew Delay Arrival
                             (fF) (ps)  (ps)   (ps)
------------------------------------------------------------
x[1]     <<<  in port        2 20.3  100    +0       0 F
g1080/A                                    +0       0
g1080/Y       INVX20        2  6.4   32   +39      39 R
g1067/A1                                   +0      39
g1067/Y       OAI21XL       1  0.6   92   +69     109 F
z[1]     <<<  out port                     +0     109 F
------------------------------------------------------------
Exception    : 'path_delays/del_1'     250ps
Timing slack :     141ps
Start-point  : x[1]
End-point    : z[1]
```

| For Nangate OCL, input -> output | Timing delay (in pico seconds) |
|---|---|
| x[0] –> z[0] | 14 ps |
| x[0] -> z[1] | 13 ps |
| x[0] -> z[2] | 9 ps |
| x[1] –> z[0] | 14 ps |
| x[1] –> z[1] | 13 ps |
| x[1] -> z[2] | 10 ps |
| x[2] –> z[0] | 14 ps |
| x[2] –> z[1] | 12 ps |
| x[2] –> z[2] | 8 ps |
| y[0] –> z[0] | 4 ps |
| y[1] –> z[0] | 14 ps |
| y[1] –> z[1] | 12 ps |
| y[1] –> z[2] | 10 ps |
| y[2] –> z[0] | 14 ps |
| y[2] –> z[1] | 12 ps |
| Y[2] -> z[2] | 4 ps |

The remaining paths do not exist in the circuit. There are two min delay paths,

y[0] -> z[0] and y[2] -> z[2],

both of which have 4 ps timing delay. There are 2 cells in both the paths.

(6) For the 8-bit mux_comparator, the following Verilog file has been created, (file : “mux_comparator_8bit.v”)

mux_comparator_8bit_RAK_timing_report.txt | mux_comparator_8bit.v

```
module comparator (
  input reg [7:0] x,
  input reg [7:0] y,
  output reg [7:0] z
  );

  always @* begin

   if (x < y)
     z = x;

   else
     z = y;

  end

endmodule
```

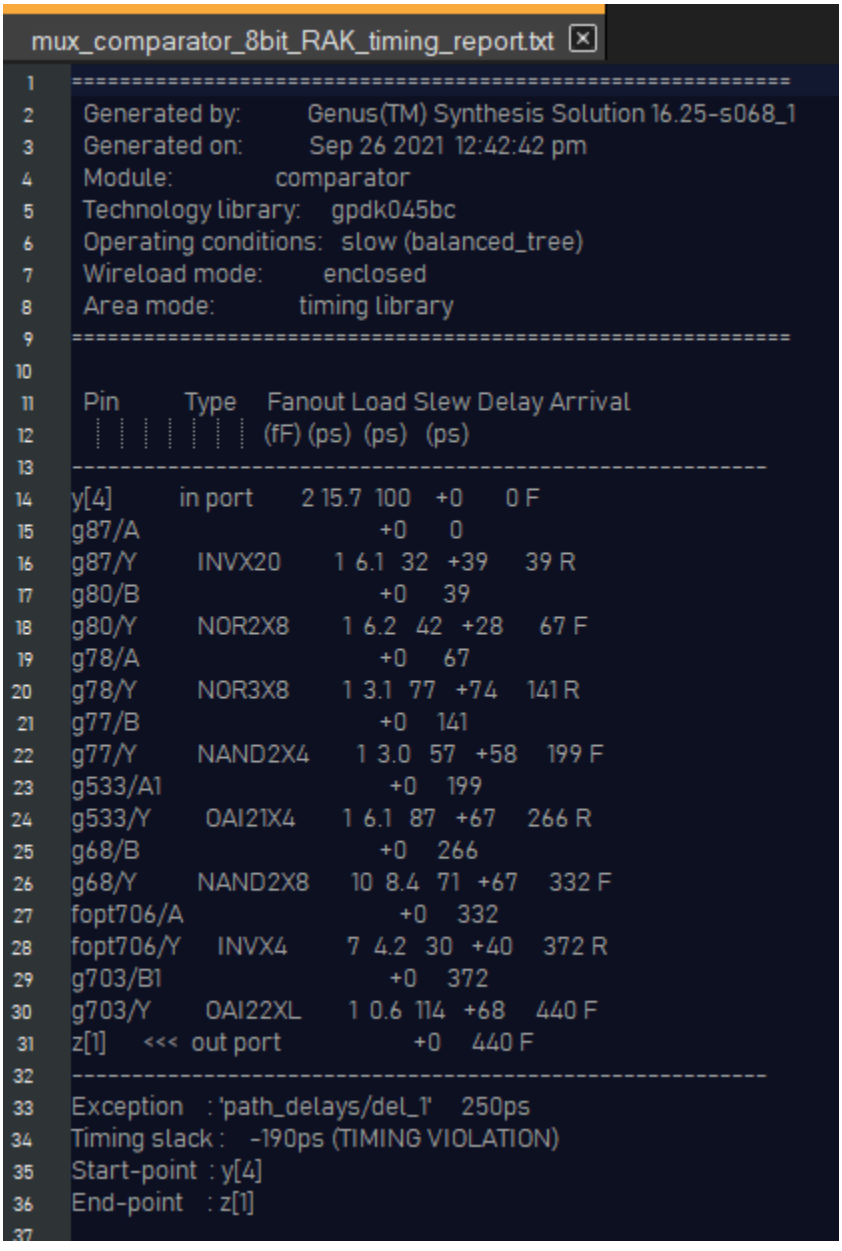
The Tcl file used to generate critical path info is, (file : “mux_comparator_8bit_RAK.tcl”)

mux_comparator_8bit_RAK.tcl

```
#rm designs/*

# reading the verilog file mux_comparator_8bit.v
read_hdl /home/cad5/Hw-2/mux_comparator_8bit.v

# reading the required .lib file in Cadence RAK
set_attr library /home/cad5/PDK/Cadence_RAK_18.1_blockImplementation/LIBS/lib/max/slow.lib

elaborate

set_attr syn_generic_effort high
syn_gen

syn_map

# setting the max delays from all inputs to all outputs to be 250 ps
set_max_delay 0.25 -from [all_inputs] -to [all_outputs]

# setting the load capacitance value of all output pins to be 0.00060 pF
set_load 0.000601595 [all_outputs]
# z[0] z[1] z[2] z[3] z[4] z[5] z[6] z[7]

# setting the input transition of all input pins to be 100 ps
set_input_transition 0.1  [all_inputs]
# x[0] x[1] x[2] x[3] x[4] x[5] x[6] x[7] y[0] y[1] y[2] y[3] y[4] y[5] y[6] y[7]

set_attr syn_opt_effort high
syn_opt

# we will get the critical path and the cells used in the critical path by using report_timing
report_timing > mux_comparator_8bit_RAK_timing_report.txt
```

The generated timing report is,

(file : “mux_comparator_8bit_RAK_timing_report.txt”)



mux_comparator_8bit_RAK_timing_report.txt

```
1  ============================================================
2   Generated by:        Genus(TM) Synthesis Solution 16.25-s068_1
3   Generated on:        Sep 26 2021 12:42:42 pm
4   Module:              comparator
5   Technology library:  gpdk045bc
6   Operating conditions:  slow (balanced_tree)
7   Wireload mode:       enclosed
8   Area mode:           timing library
9  ============================================================
10
11  Pin        Type    Fanout Load Slew Delay Arrival
12                     (fF) (ps)  (ps)   (ps)
13 ------------------------------------------------------------
14 y[4]        in port      2 15.7  100   +0     0 F
15 g87/A                                +0     0
16 g87/Y       INVX20       1  6.1   32  +39    39 R
17 g80/B                                +0    39
18 g80/Y       NOR2X8       1  6.2   42  +28    67 F
19 g78/A                                +0    67
20 g78/Y       NOR3X8       1  3.1   77  +74   141 R
21 g77/B                                +0   141
22 g77/Y       NAND2X4      1  3.0   57  +58   199 F
23 g533/A1                              +0   199
24 g533/Y      OAI21X4      1  6.1   87  +67   266 R
25 g68/B                                +0   266
26 g68/Y       NAND2X8     10  8.4   71  +67   332 F
27 fopt706/A                            +0   332
28 fopt706/Y    INVX4       7  4.2   30  +40   372 R
29 g703/B1                              +0   372
30 g703/Y      OAI22XL      1  0.6  114  +68   440 F
31 z[1]    <<<  out port                +0   440 F
32 ------------------------------------------------------------
33 Exception   : 'path_delays/del_1'   250ps
34 Timing slack :   -190ps (TIMING VIOLATION)
35 Start-point  : y[4]
36 End-point    : z[1]
37
```

So the path from y[4] to z[1] is the critical path and the cells used in that path are INVX20, NOR2X8, NOR3X8, NAND2X4, OAI21X4, NAND2X8, INVX4, OAI22XL. Number of cells used in critical path are 8.