

## **Day 1 - Introduction to Verilog RTL design and Synthesis**

### **SKY130RTL D1SK1 L1 Introduction to iverilog design test bench**

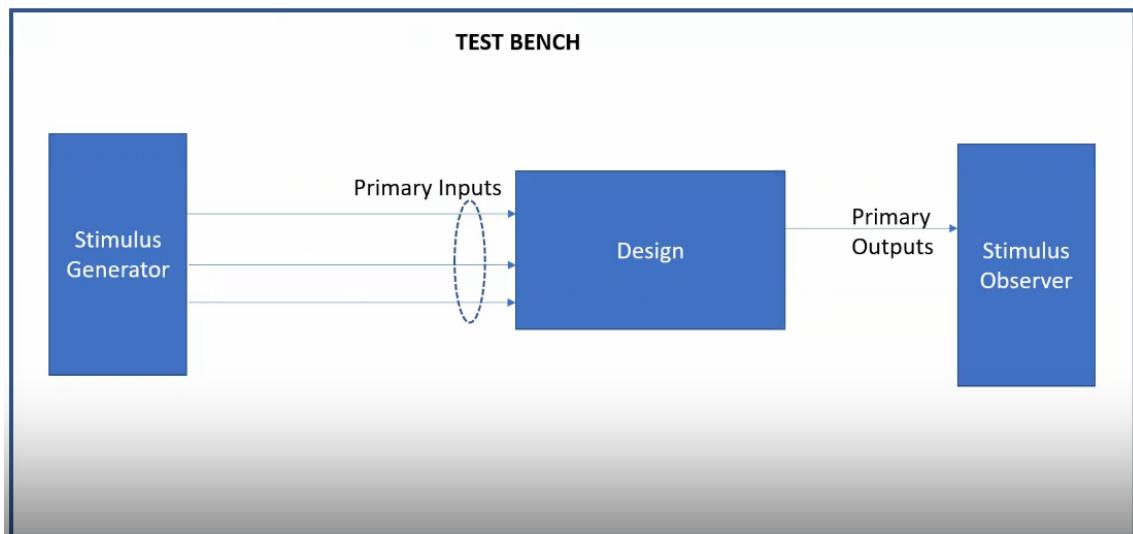
In the first lecture of the course, we learnt about the definitions of the RTL Design, Simulator and Test Bench. The simulator we are going to use in this workshop is iverilog. After the definitions of the terms mentioned above, we learnt the workings of Simulator and Test Bench.

#### **Simulator Working:**

Simulator looks for the changes in the input and for every change in the input, we observe changes in the output.

#### **Test Bench:**

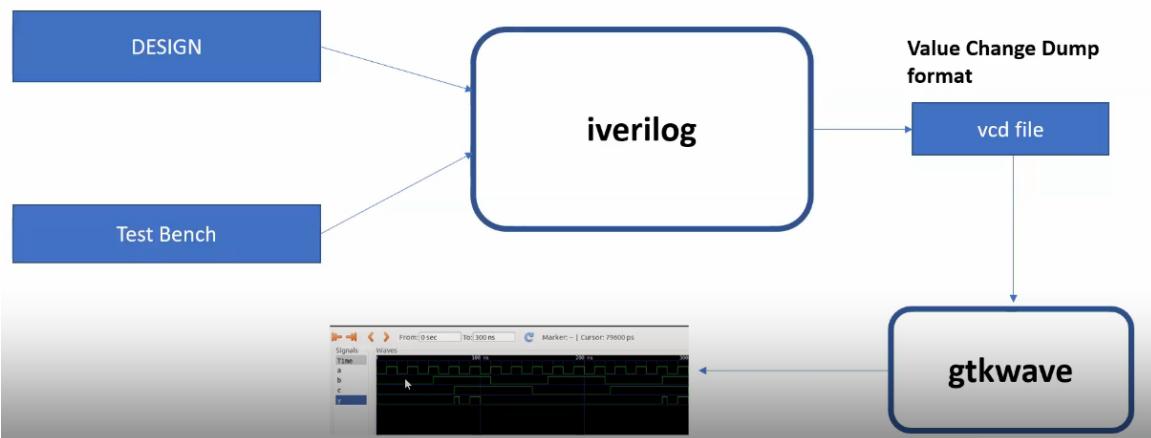
Test Bench generates the stimulus for the design and observes the output.



#### **iverilog based simulator flow:**

iverilog simulator takes the design and corresponding test bench of the design and generates the vcd( Value Change Dump format) file. The vcd file can be viewed using gtkwave waveform viewer.

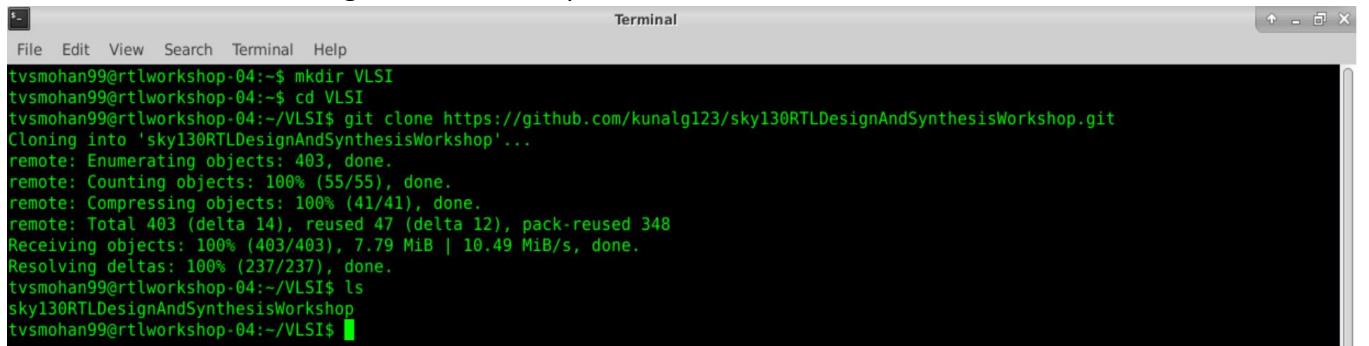
### **Iverilog based Simulation Flow**



## SKY130RTL D1SK2 - Labs using iverilog and gtkwave

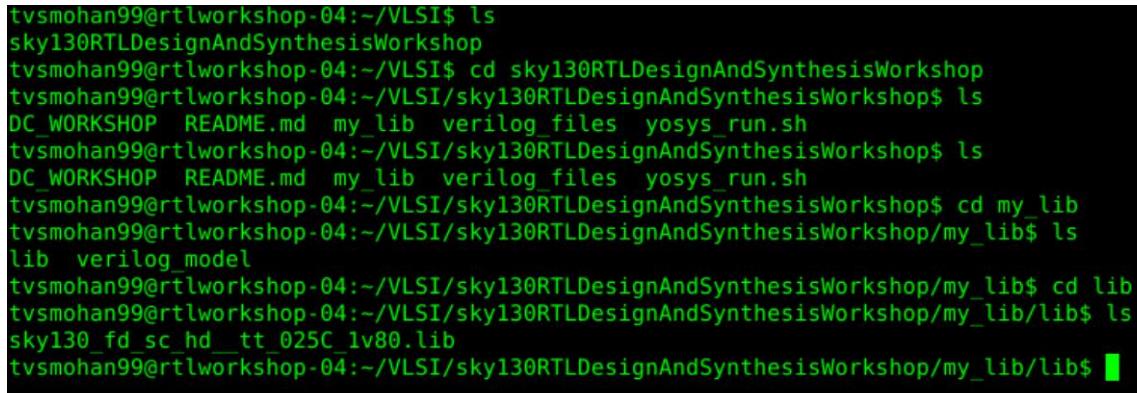
### **SKY130RTL D1SK2 L1 Lab1 introduction to lab:**

In this session we git cloned the required modules from GitHub to VLSI foldr



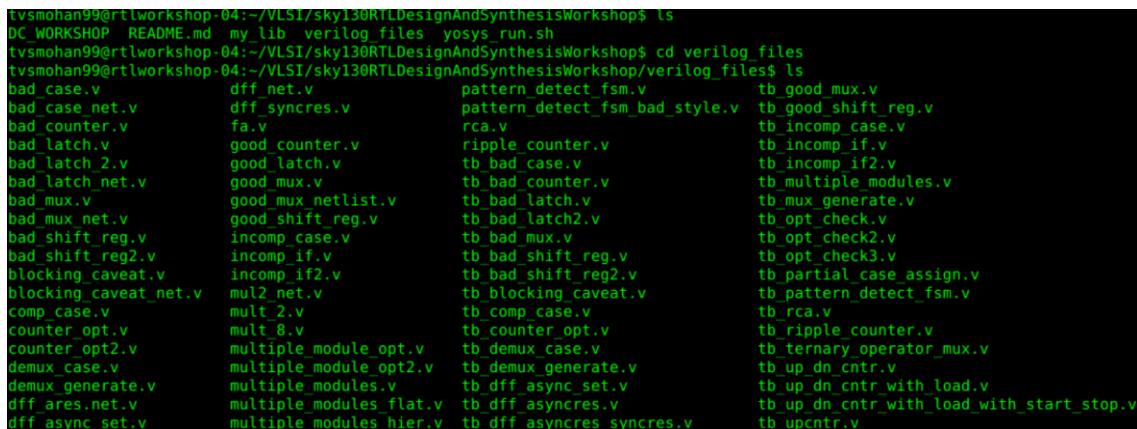
```
File Edit View Search Terminal Help
Terminal
tvsmohan99@rtlworkshop-04:~$ mkdir VLSI
tvsmohan99@rtlworkshop-04:~$ cd VLSI
tvsmohan99@rtlworkshop-04:~/VLSI$ git clone https://github.com/kunalg123/sky130RTLDesignAndSynthesisWorkshop.git
Cloning into 'sky130RTLDesignAndSynthesisWorkshop'...
remote: Enumerating objects: 403, done.
remote: Counting objects: 100% (55/55), done.
remote: Compressing objects: 100% (41/41), done.
remote: Total 403 (delta 14), reused 47 (delta 12), pack-reused 348
Receiving objects: 100% (403/403), 7.79 MiB | 10.49 MiB/s, done.
Resolving deltas: 100% (237/237), done.
tvsmohan99@rtlworkshop-04:~/VLSI$ ls
sky130RTLDesignAndSynthesisWorkshop
tvsmohan99@rtlworkshop-04:~/VLSI$
```

Inside the sky130 folder we have my lib which contains the standard cell library  
sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib



```
tvsmohan99@rtlworkshop-04:~/VLSI$ ls
sky130RTLDesignAndSynthesisWorkshop
tvsmohan99@rtlworkshop-04:~/VLSI$ cd sky130RTLDesignAndSynthesisWorkshop
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop$ ls
DC_WORKSHOP README.md my_lib verilog_files yosys_run.sh
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop$ ls
DC_WORKSHOP README.md my_lib verilog_files yosys_run.sh
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop$ cd my_lib
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/my_lib$ ls
lib verilog_model
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop$ cd lib
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/my_lib/lib$ ls
sky130_fd_sc_hd_tt_025C_1v80.lib
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/my_lib/lib$
```

All the required Verilog files and Test benches are present in Verilog files folder.



```
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop$ ls
DC_WORKSHOP README.md my_lib verilog_files yosys_run.sh
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop$ cd verilog_files
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ ls
bad_case.v          dff.net.v           pattern_detect_fsm.v      tb_good_mux.v
bad_case_net.v      dff_syncres.v     pattern_detect_fsm_bad_style.v  tb_good_shift_reg.v
bad_counter.v       fa.v               rca.v                  tb_incomp_case.v
bad_latch.v         good_counter.v   ripple_counter.v        tb_incomp_if.v
bad_latch_2.v       good_latch.v    tb_bad_case.v        tb_incomp_if2.v
bad_latch_net.v    good_mux.v       tb_bad_latch.v       tb_multiple_modules.v
bad_mux.v           good_mux_netlist.v tb_bad_latch2.v      tb_mux_generate.v
bad_mux_net.v       good_shift_reg.v  tb_bad_latch2.v      tb_opt_check.v
bad_shift_reg.v     incomp_case.v   tb_bad_mux.v        tb_opt_check2.v
bad_shift_reg2.v    incomp_if.v     tb_bad_shift_reg.v   tb_opt_check3.v
blocking_caveat.v  incomp_if2.v    tb_bad_shift_reg2.v  tb_partial_case_assign.v
blocking_caveat_net.v mul2.net.v     tb_blocking_caveat.v  tb_pattern_detect_fsm.v
comp_case.v         mult_2.v        tb_comp_case.v      tb_rca.v
counter_opt.v       mult_8.v        tb_counter_opt.v   tb_ripple_counter.v
counter_opt2.v      multiple_module_opt.v tb_demux_case.v  tb_ternary_operator_mux.v
demux_case.v        multiple_module_opt2.v tb_demux_generate.v tb_up_dn_cntr.v
demux_generate.v    multiple_modules.v  tb_dff_async_set.v tb_up_dn_cntr_with_load.v
dff_ares.net.v      multiple_modules.flat.v tb_dff_asyncres.v tb_up_dn_cntr_with_load_with_start_stop.v
dff_async_set.v     multiple_modules.hier.v tb_dff_asyncres_syncres.v tb_upcntr.v
```

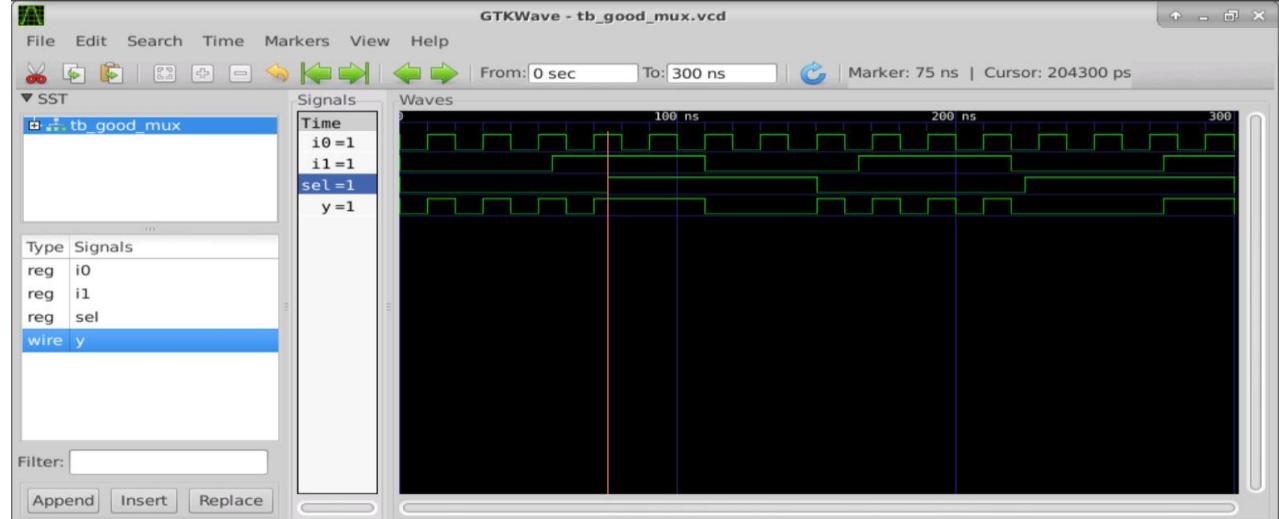
## SKY130RTL D1SK2 L2 Lab2 Introduction iverilog gtkwave part1:

In this lab, we learnt how to simulate Verilog files using iverilog. After executing the a.out file we get the dumpfile, which can be used using GTKWAVE waveform user.

```
dff_async_set.v      multiple_modules_hier.v   tb_dff_asyncre_synthres.v    tb_upcntr.v
dff_asyncre.v        mux_generate.v          tb_dff_const1.v           ternary_operator_mux.v
dff_asyncre_net.v   mux_spice.v            tb_dff_const2.v           ternary_operator_mux_net.v
dff_asyncre_synthres.v net.v                tb_dff_const3.v           up_dn_cntr.v
dff_const1.v         opt_check.v            tb_dff_const4.v           up_dn_cntr_with_load.v
dff_const2.v         opt_check2.v           tb_dff_const5.v           up_dn_cntr_with_load_with_start_stop.v
dff_const3.v         opt_check3.v           tb_dff_syncres.v          upcntr.v
dff_const4.v         opt_check4.v           tb_good_counter.v
dff_const5.v         partial_case_assign.v  tb_good_latch.v
tvsmohan99@rtlworkshop-04:/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ iverilog good_mux.v tb_good_mux.v
tvsmohan99@rtlworkshop-04:/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ ./a.out
VCD info: dumpfile tb_good_mux.vcd opened for output.
tvsmohan99@rtlworkshop-04:/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_good_mux.vcd

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

[0] start time.
[300000] end time.
```



## SKY130RTL D1SK2 L3 Lab2 Introduction iverilog gtkwave part2:

In this module, we have seen how to open verilog file using gvim command.

```
File Edit Tools Syntax Buffers Window Help
good_mux.v (~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files) ((1 of 2) - GVIM
timescale 1ns / 1ps
module tb_good_mux;
  // Inputs
  reg i0,i1,sel;
  // Outputs
  wire y;

  // Instantiate the Unit Under Test (UUT)
  good_mux uut (
    .sel(sel),
    .i0(i0),
    .i1(i1),
    .y(y)
  );

  initial begin
    $dumpfile("tb_good_mux.vcd");
    $dumpvars(0,tb_good_mux);
    // Initialize Inputs
    sel = 0;
    i0 = 0;
  end
endmodule
```

## SKY130RTL D1SK3 - Introduction to Yosys and Logic synthesis

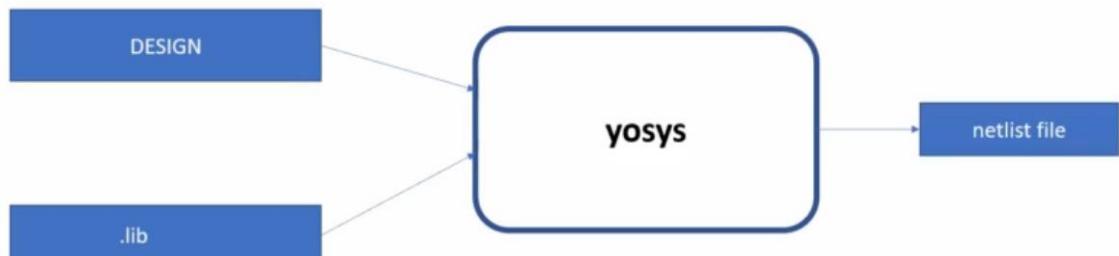
### SKY130RTL D1SK3 L1 Introduction to yosys:

#### Synthesizer:

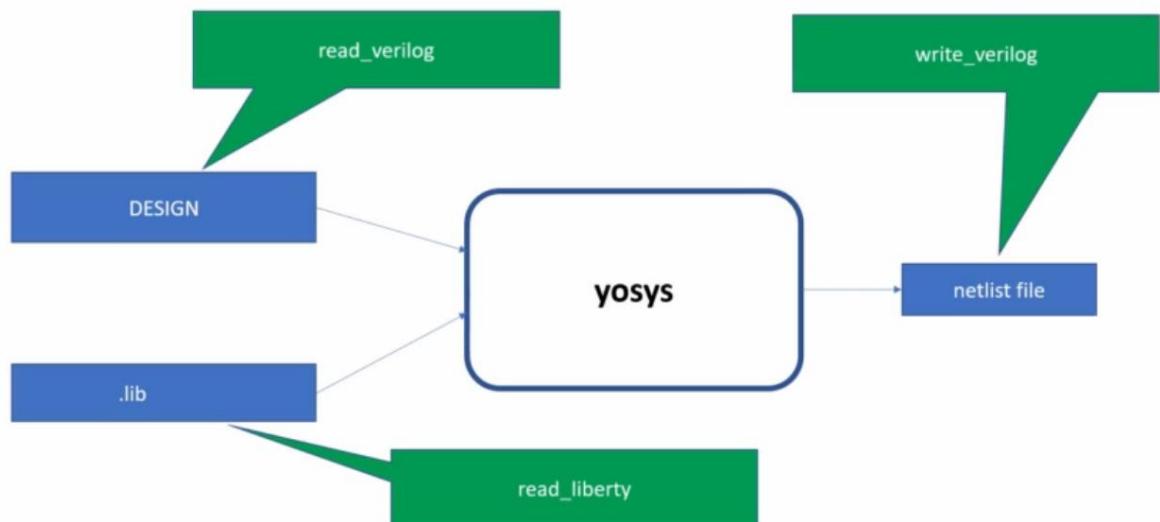
It is a tool used for converting the RTL into netlist. In this workshop we are going to use yosys synthesizer.

## Synthesizer

- Tool used for converting the RTL to netlist
- **Yosys** is the synthesizer used in this course



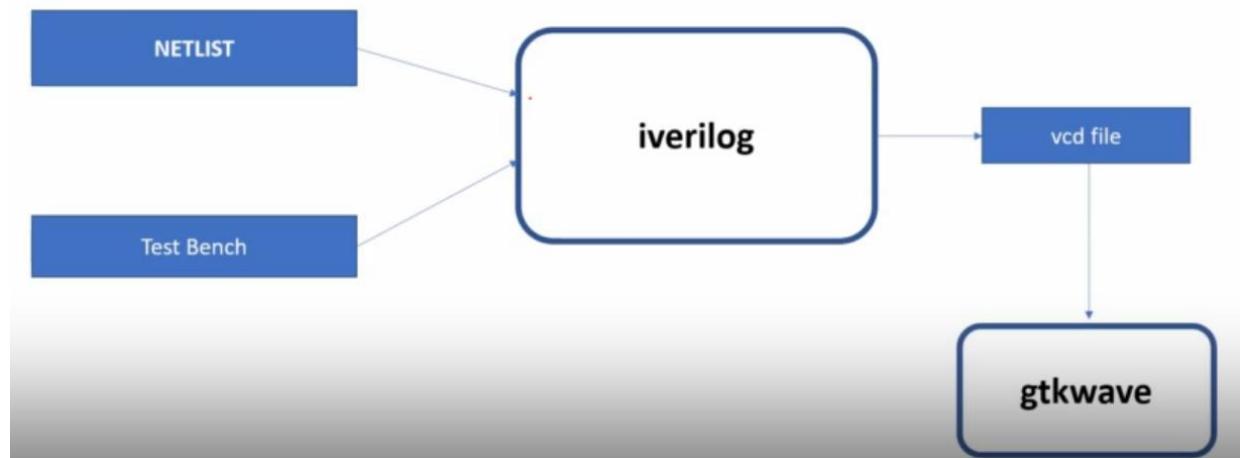
#### Yosys setup:



### **Verification of the synthesis:**

To verify the generated netlist, we are going to use the same test bench and iverilog simulator.

## Verify the synthesis



### **SKY130RTL D1SK3 L2 introduction to logic synthesis part1**

#### **RTL Design:**

## RTL Design

- Behavioral representation of the required specification

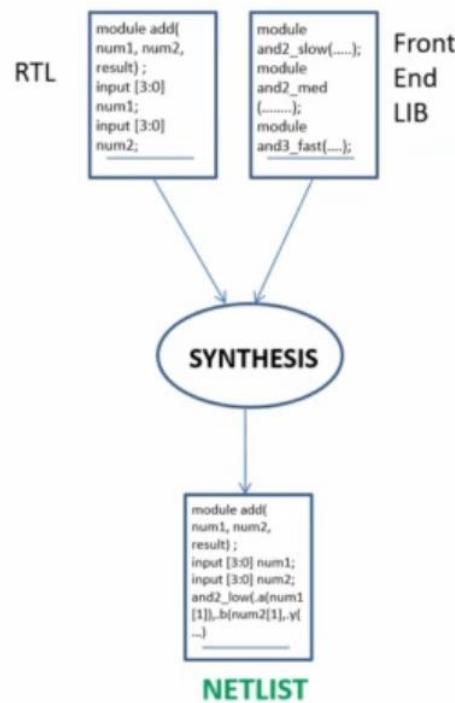
```
module sample_code (
    input clk,rst ,
    output result,done);

    always @ (posedge clk ,posedge rst)
        if(rst)
            ---
        else
            ---
    endmodule
```

## SYNTHESIS:

# Synthesis

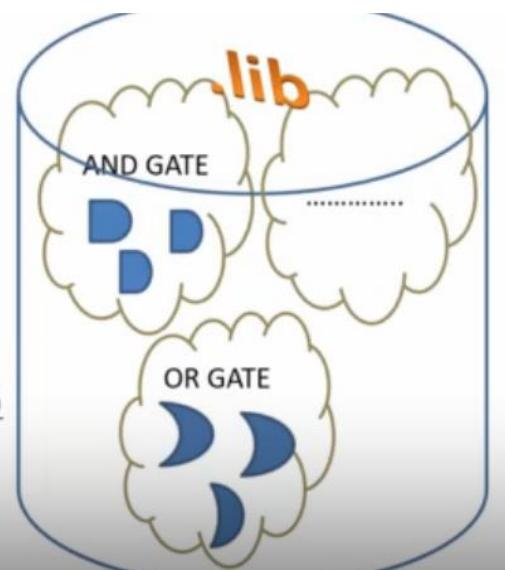
- RTL to Gate level translation
- The design is converted into gates and the connections are made between the gates
- This is given out as a file called netlist



## .lib:

.lib is a collection of logic modules, which include basic logic gates and different flavours of same gates.

- .lib
  - Collection of logical modules.
  - Includes basic logic gates like And, Or , Not, etc...
  - Different flavors of same gate
    - 2 input And gate
      - Slow
      - Medium
      - Fast
    - 3 input And gate
      - Slow
      - Medium
      - Fast
    - 4 input And gate

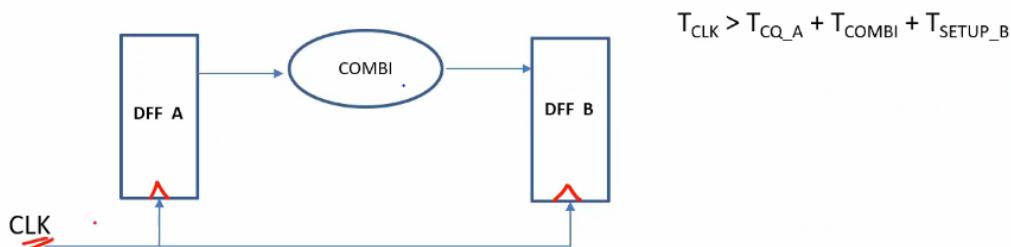


## Why do we have different flavours of gates?

In order to have time for the logic to be executed properly and for setting up and holding of the current and previous inputs and outputs, we need different flavours of gates.

For example, consider this situation:

- Combinational delay in logic path determines the maximum speed of operation of digital logic circuit



In this situation the clock should accommodate the propagation delay, combinational logic delay and setup time of the B D flip flop. As the time is inversely proportional to frequency the smaller the time the faster the frequency. If we have faster cells, the time taken for the operation will be less. Less time implies maximum frequency.

## SKY130RTL D1SK3 L3 introduction to logic synthesis part2:

### Why do we need slower cells?

In order to have proper hold time for the newly generated values in the design. If we go too fast, the logic may miss few values and we won't get the intended results.

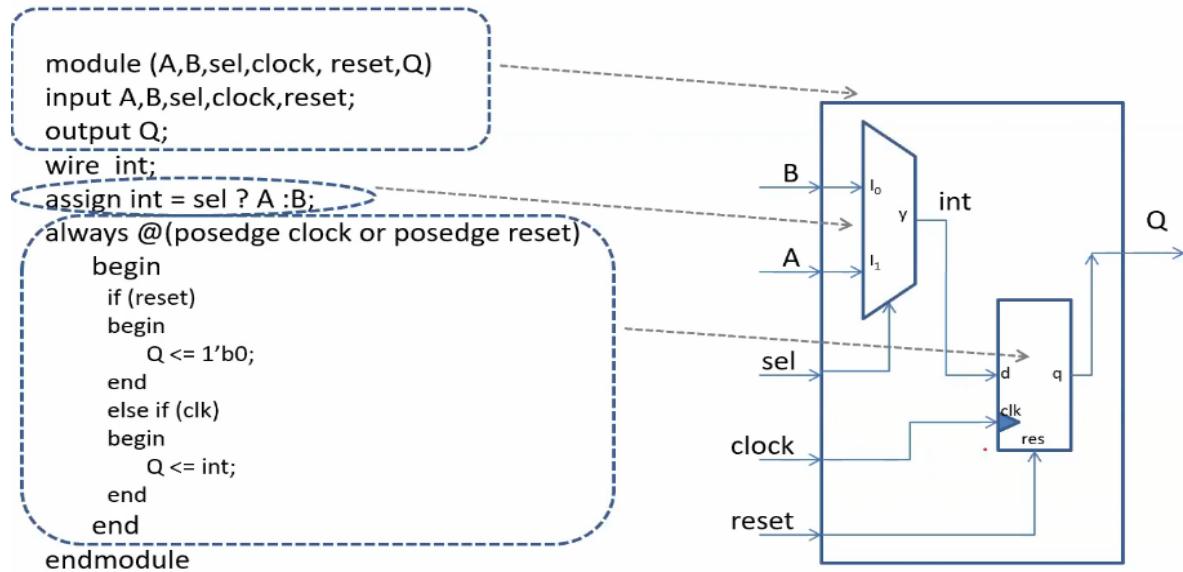
### FASTER CELLS VS SLOWER CELLS:

- Load in Digital Logic circuit → Capacitance
- Faster the charging / discharging of capacitance → Lesser the cell delay
  - To charge / discharge the capacitance fast , we need transistors capable of sourcing more current
  - Wider transistors -> Low Delay -> More Area and Power as well !!
  - Narrow transistors -> More Delay -> Less Area and Power
  - Faster cells donot come free , they come at penalty of area and power

## SYNTHESIS:

After syntactical check, the synthesizer will map different types of statements in the code to standard cells as shown below.

# Synthesis (Illustration)



## SKY130RTL D1SK4 L1 Lab3 Yosys 1 good mux Part1:

### yosys:

yosys can be invoked by using the command yosys in the Verilog files module.

```
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDdesignAndSynthesisWorkshop/verilog_files$ yosys
-----
| yosys -- Yosys Open SYnthesis Suite
|
| Copyright (C) 2012 - 2020 Claire Xenia Wolf <claire@yosyshq.com>
|
| Permission to use, copy, modify, and/or distribute this software for any
| purpose with or without fee is hereby granted, provided that the above
| copyright notice and this permission notice appear in all copies.
|
| THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
| WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
| MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
| ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
| WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
```

**Reading the liberty:**

**Command: read\_liberty -lib ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib**

```
| WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
| ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
| OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
```

```
\-----/
```

```
Yosys 0.9+4081 (git sha1 862e84eb, gcc 7.5.0-3ubuntu1~18.04 -fPIC -Os)
```

```
yosys> read_liberty -lib ..../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.
```

```
yosys> █
```

**Command: read\_verilog good\_mux.v**

Reads the good\_mux verilog file.

```
yosys> read_verilog good_mux.v
2. Executing Verilog-2005 frontend: good_mux.v
Parsing Verilog input from `good_mux.v' to AST representation.
Generating RTLIL representation for module `good_mux'.
Successfully finished Verilog frontend.
```

```
yosys> █
```

**Command: synth-top good\_mux**

Specifies the modules to be synthesized and synthesizes the module.

```
3.25. Printing statistics.

==== good_mux ===

Number of wires: 4
Number of wire bits: 4
Number of public wires: 4
Number of public wire bits: 4
Number of memories: 0
Number of memory bits: 0
Number of processes: 0
Number of cells: 1
$_MUX_ 1

3.26. Executing CHECK pass (checking for obvious problems).
Checking module good_mux...
Found and reported 0 problems.
```

```
yosys> █
```

Commnad: abc -liberty ../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

This command generates netlist based on the standard cells available in the library.

```
ABC: Library "sky130_fd_sc_hd_tt_025C_1v80" from "/home/tvsmohan99/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files/..../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib" has 334 cells (94 skipped: 63 seq; 13 tri-state; 18 no func; 0 dont_use). Time = 0.24 sec
ABC: Memory = 16.00 MB. Time = 0.24 sec
ABC: Warning: Detected 9 multi-output gates (for example, "sky130_fd_sc_hd_fa_1").
ABC: + strash
ABC: + ifraig
ABC: + scorr
ABC: Warning: The network is combinational (run "fraig" or "fraig_sweep").
ABC: + dc2
ABC: + dretime
ABC: + strash
ABC: + &get -n
ABC: + &dch -f
ABC: + &nf
ABC: + &put
ABC: + write_bif <abc-temp-dir>/output.blif

5.1.2. Re-integrating ABC results.
ABC RESULTS: sky130_fd_sc_hd_mux2_i cells: 1
ABC RESULTS: internal signals: 0
ABC RESULTS: input signals: 3
ABC RESULTS: output signals: 1
Removing temp directory.

yosys> █
```

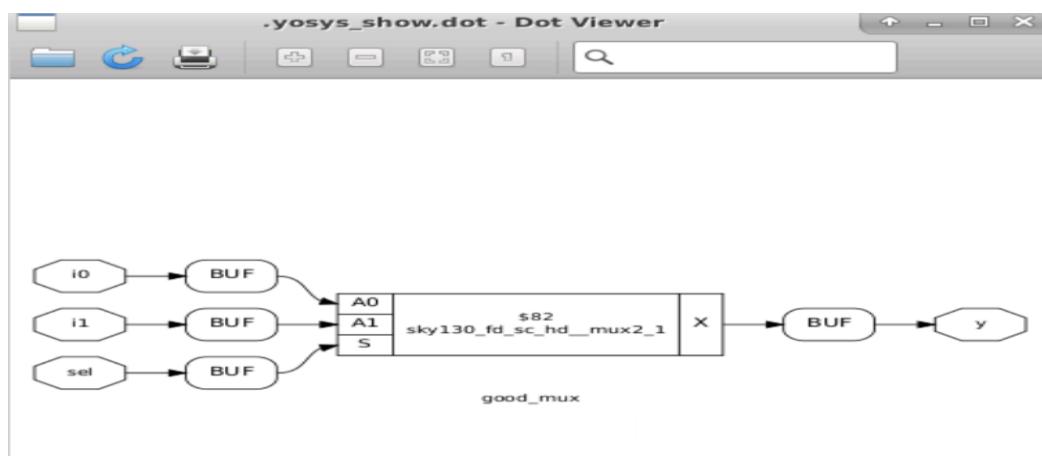
Command: show

Displays the graphical version of the netlist in dot viewer.

```
yosys> show
6. Generating Graphviz representation of design.
Writing dot description to '/home/tvsmohan99/.yosys_show.dot'.
Dumping module good_mux to page 1.
Exec: { test -f '/home/tvsmohan99/.yosys_show.dot.pid' && fuser -s '/home/tvsmohan99/.yosys_show.dot.pid' 2> /dev/null; } || ( echo $$ >&3; exec xdot '/home/tvsmohan99/.yosys_show.dot'; ) 3> '/home/tvsmohan99/.yosys_show.dot.pid' &
yosys> █
```

### SKY130RTL D1SK4 L2 Lab3 Yosys 1 good mux Part2:

In older version of yosys the above synthesis will generate two inverse gates and Nand for the function of and. But in the latest version directly a mux is used as standard cell.



### SKY130RTL D1SK4 L3 Lab3 Yosys 1 good mux Part3:

Command: write\_verilog good\_mux\_netlist.v

```
yosys> write_verilog good_mux_netlist.v

7. Executing Verilog backend.
Dumping module `good_mux'.

yosys>
```

Command: !gvim good\_mux\_netlist.v

Reads the netlist.



The screenshot shows a GVIM window displaying the Verilog code for the 'good\_mux' module. The window title is 'good\_mux\_netlist.v (~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog\_files) - GVIM1'. The code is generated by Yosys 0.9+4081. It defines a module 'good\_mux' with inputs i0, i1, and sel, and output y. The module uses four wires (0, 1, 2, 3) and a sky130\_fd\_sc\_hd\_mux2\_1 component. The component has inputs A0, A1, and S, and output Z. The Verilog code includes comments indicating the source of each component and wire definition.

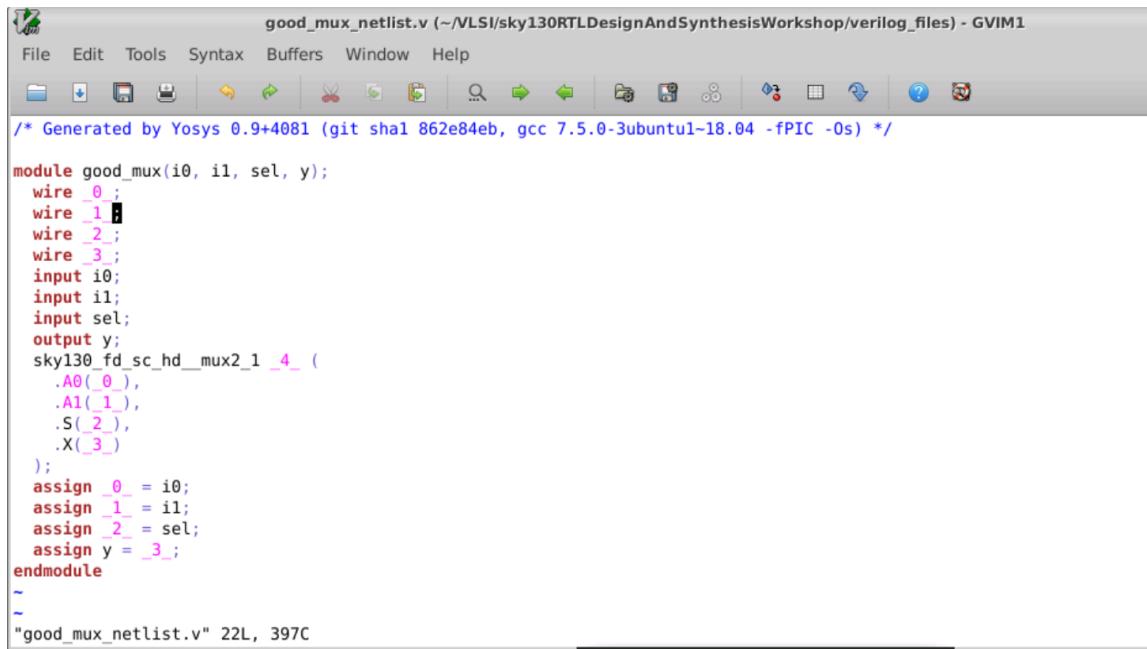
```
/* Generated by Yosys 0.9+4081 (git sha1 862e84eb, gcc 7.5.0-3ubuntul-18.04 -fPIC -Os) */

(* top = 1 *)
(* src = "good_mux.v:2.1-10.10" *)
module good_mux(i0, i1, sel, y);
    (* src = "good_mux.v:2.24-2.26" *)
    wire _0;
    (* src = "good_mux.v:2.35-2.37" *)
    wire _1;
    (* src = "good_mux.v:2.46-2.49" *)
    wire _2;
    (* src = "good_mux.v:2.63-2.64" *)
    wire _3;
    (* src = "good_mux.v:2.24-2.26" *)
    input i0;
    (* src = "good_mux.v:2.35-2.37" *)
    input i1;
    (* src = "good_mux.v:2.46-2.49" *)
    input sel;
    (* src = "good_mux.v:2.63-2.64" *)
    output y;
    sky130_fd_sc_hd_mux2_1 _4_ (
        .A0(_0),
        .A1(_1),
        .Z(y)
    );
endmodule
"good_mux_netlist.v" 32L, 744C
```

Command: write\_verilog -noattr good\_mux\_netlist.v

!gvim goof\_mux\_netlist.v

To make the netlist readable.



The screenshot shows a GVIM editor window with the title "good\_mux\_netlist.v (~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog\_files) - GVIM1". The menu bar includes File, Edit, Tools, Syntax, Buffers, Window, and Help. The toolbar below has icons for file operations like Open, Save, Find, and Print. The code in the buffer is a Verilog module definition:

```
/* Generated by Yosys 0.9+4081 (git sha1 862e84eb, gcc 7.5.0-3ubuntu1-18.04 -fPIC -Os) */

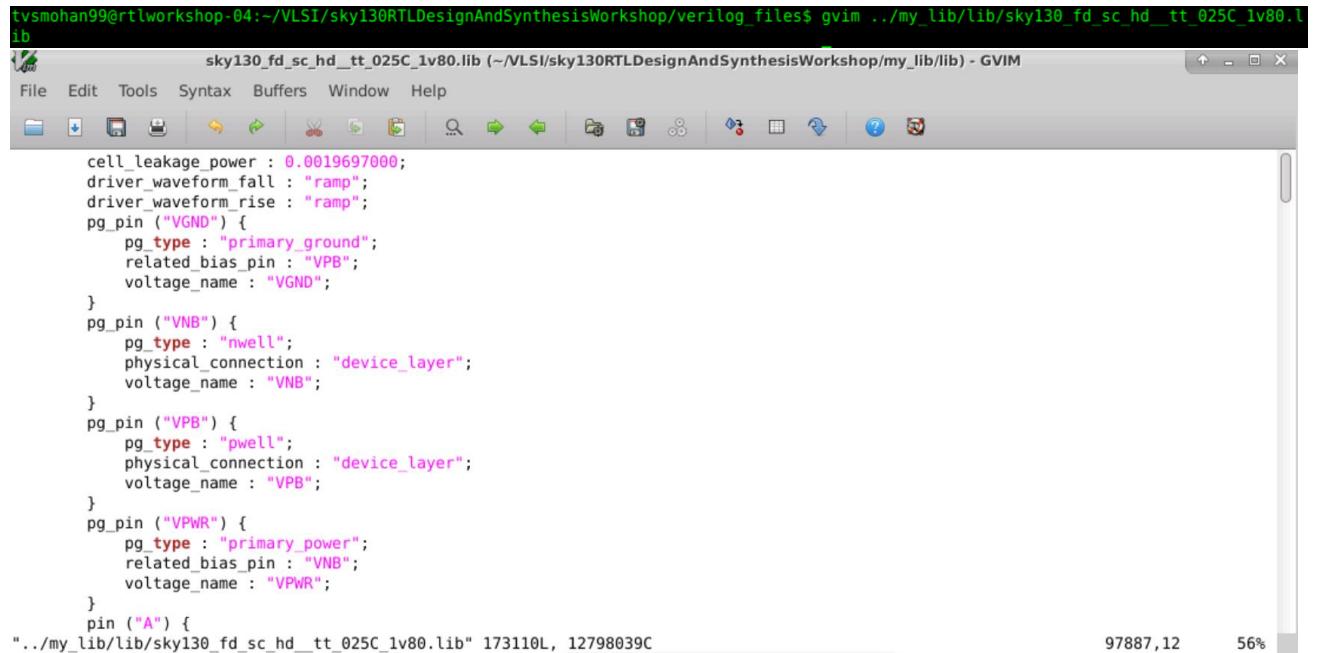
module good_mux(i0, i1, sel, y);
    wire _0;
    wire _1;
    wire _2;
    wire _3;
    input i0;
    input i1;
    input sel;
    output y;
    sky130_fd_sc_hd__mux2_1 _4_ (
        .A0(_0),
        .A1(_1),
        .S(_2),
        .X(_3)
    );
    assign _0 = i0;
    assign _1 = i1;
    assign _2 = sel;
    assign y = _3;
endmodule
~
~

"good_mux_netlist.v" 22L, 397C
```

## DAY 2:

### **SKY130RTL D2SK1 L1 Lab4 Introduction to dot :**

Command: gvim ..\my\_lib\lib\sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib



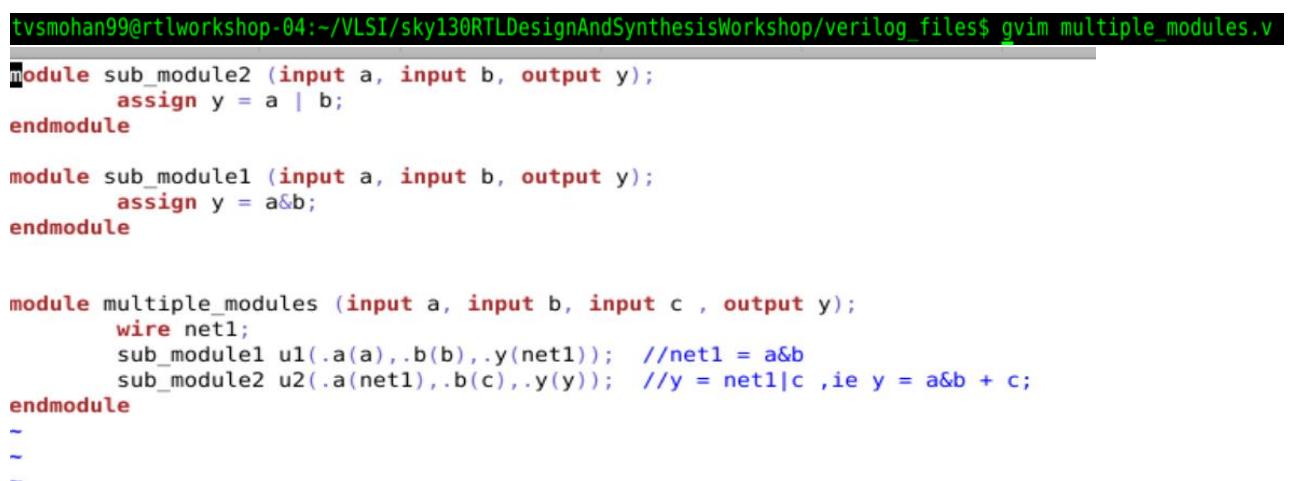
```
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gvim ..\my_lib\lib\sky130_fd_sc_hd_tt_025C_1v80.lib
File Edit Tools Syntax Buffers Window Help
cell_leakage_power : 0.0019697000;
driver_waveform_fall : "ramp";
driver_waveform_rise : "ramp";
pg_pin ("VGND") {
    pg_type : "primary_ground";
    related_bias_pin : "VPB";
    voltage_name : "VGND";
}
pg_pin ("VNB") {
    pg_type : "nwell";
    physical_connection : "device_layer";
    voltage_name : "VNB";
}
pg_pin ("VPB") {
    pg_type : "pwell";
    physical_connection : "device_layer";
    voltage_name : "VPB";
}
pg_pin ("VPWR") {
    pg_type : "primary_power";
    related_bias_pin : "VNB";
    voltage_name : "VPWR";
}
pin ("A") {
    assign y = a | b;
}
"..\my_lib\lib\sky130_fd_sc_hd_tt_025C_1v80.lib" 173110L, 12798039C
97887,12      56%
```

The various terms of the opened library's name are tt-> typical process ,025C-> Temperature, 1v80->voltage.

### **SKY130RTL D2SK2 L1 Lab05 Hier synthesis flat synthesis part1:**

#### **HEIRARCHIAL SYNTHESIS OF MULTIPLE\_MODULES**

Command: gvim multiple\_modules.v



```
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gvim multiple_modules.v
module sub_module2 (input a, input b, output y);
    assign y = a | b;
endmodule

module sub_module1 (input a, input b, output y);
    assign y = a&b;
endmodule

module multiple_modules (input a, input b, input c , output y);
    wire net1;
    sub_module1 u1(.a(a),.b(b),.y(net1)); //net1 = a&b
    sub_module2 u2(.a(net1),.b(c),.y(y)); //y = net1|c ,ie y = a&b + c;
endmodule
~
```

Command:yosys

Command:read\_liberty –lib ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:read\_verilog multiple\_modules.v

```
yosys> read_liberty -lib ..../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
2. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog multiple_modules.v
3. Executing Verilog-2005 frontend: multiple_modules.v
Parsing Verilog input from `multiple_modules.v' to AST representation.
Generating RTLIL representation for module `\sub_module2'.
Generating RTLIL representation for module `\sub_module1'.
Generating RTLIL representation for module `\multiple_modules'.
Successfully finished Verilog frontend.

yosys> █
```

Command:synth–top multiple\_modules

```
== design hierarchy ==
multiple_modules          1
  sub_module1              1
  sub_module2              1

Number of wires:           11
Number of wire bits:       11
Number of public wires:    11
Number of public wire bits: 11
Number of memories:        0
Number of memory bits:     0
Number of processes:       0
Number of cells:           2
  $_AND_                   1
  $_OR_                     1

4.26. Executing CHECK pass (checking for obvious problems).
Checking module multiple_modules...
Checking module sub_module1...
Checking module sub_module2...
Found and reported 0 problems.

yosys> █
```

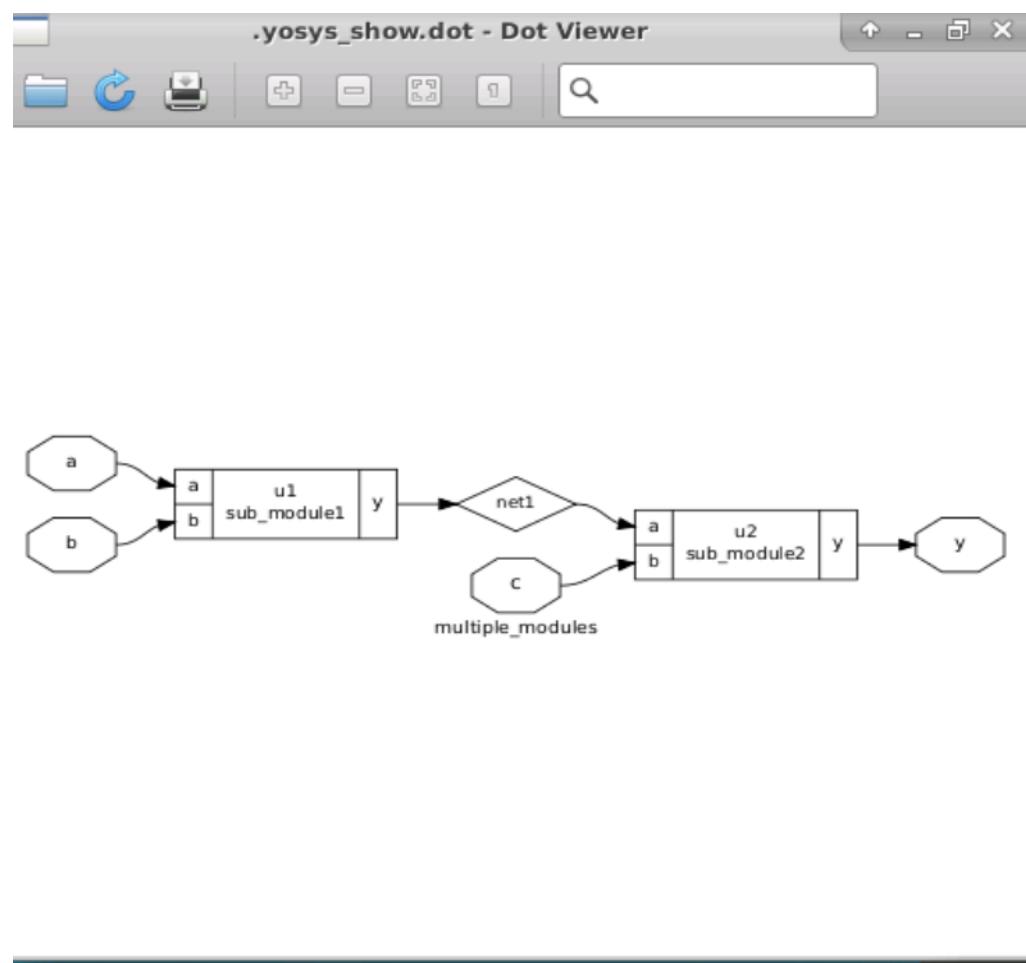
Commnad:abc -liberty ..//my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

```
5.3.2. Re-integrating ABC results.  
ABC RESULTS: sky130_fd_sc_hd_lpflow_inputisop_1 cells:  
ABC RESULTS: internal signals: 0  
ABC RESULTS: input signals: 2  
ABC RESULTS: output signals: 1  
Removing temp directory.
```

yosys> █

Command:show multiple\_modules

HEIRARICHALSYNTHESIS



Command: write\_verilog -noattr multiple\_modules\_netlist.v

Command: !gvim multiple\_modules\_netlist.v

```
yosys> write_verilog -noattr multiple_modules_netlist.v

8. Executing Verilog backend.
Dumping module `multiple_modules'.
Dumping module `sub_module1'.
Dumping module `sub_module2'.

yosys> !gvim multiple_modules_netlist.v

9. Shell command: gvim multiple_modules_netlist.v

yosys> █
```

```
/* Generated by Yosys 0.9+4081 (git sha1 862e84eb, gcc 7.5.0-3ubuntu1~18.04 -fPIC -Os) */

module multiple_modules(a, b, c, y);
    input a;
    input b;
    input c;
    wire net1;
    output y;
    sub_module1 u1 (
        .a(a),
        .b(b),
        .y(net1)
    );
    sub_module2 u2 (
        .a(net1),
        .b(c),
        .y(y)
    );
endmodule

module sub_module1(a, b, y);
    wire _0;
    wire _1;
    wire _2;
```

```
module sub_module1(a, b, y);
    wire _0_;
    wire _1_;
    wire _2_;
    input a;
    input b;
    output y;
    sky130_fd_sc_hd_and2_0 _3_ (
        .A(_1_),
        .B(_0_),
        .X(_2_)
    );
    assign _1_ = b;
    assign _0_ = a;
    assign y = _2_;
endmodule
```

```
module sub_module2(a, b, y);
    wire _0_;
    wire _1_;
    wire _2_;
    input a;
    input b;
    output y;
    sky130_fd_sc_hd_lpfflow_inputisop_1 _3_ (
        .A(_1_),
        .SLEEP(_0_),
        .X(_2_)
    );
    assign _1_ = b;
    assign _0_ = a;
    assign y = _2_;
endmodule
```

## FLAT SYNTHESIS OF MULTIPLE\_MODULES:

Command: flatten

Command: write\_verilog -noattr multiple\_modules\_flat.v

Command: !gvim multiple\_modules\_flat.v

```
yosys> flatten

8. Executing FLATTEN pass (flatten design).
Deleting now unused module sub_module1.
Deleting now unused module sub_module2.
<suppressed ~2 debug messages>

yosys> write_verilog -noattr multiple_modules_flat.v

9. Executing Verilog backend.
Dumping module `multiple_modules'.

yosys> !gvim multiple_modules_flat.v

10. Shell command: gvim multiple_modules_flat.v
```

```
/* Generated by Yosys 0.9+4081 (git sha1 862e84eb, gcc 7.5.0-3ubuntu1-18.04 -fPIC -O0) */
/* Generated by Yosys 0.9+4081 (git sha1 862e84eb, gcc 7.5.0-3ubuntu1-18.04 -fPIC -O0) */

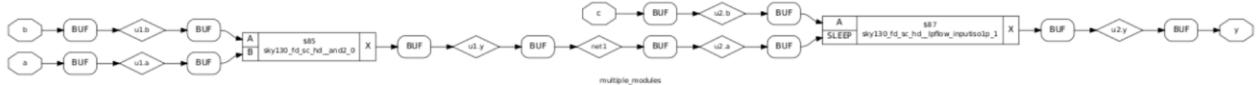
module multiple_modules(a, b, c, y);
    input a;
    input b;
    input c;
    wire net1;
    output y;
    sub_module1 u1 (
        .a(a),
        .b(b),
        .y(net1)
    );
    sub_module2 u2 (
        .a(net1),
        .b(c),
        .y(y)
    );
endmodule

module sub_module1(a, b, y);
    wire 0;
endmodule
module sub_module2(a, b, y);
    wire 1;
    wire 2;
    wire 3;
    wire 4;
    wire 5;
    input a;
    input b;
    input c;
    wire net1;
    wire \u1.a ;
    wire \u1.b ;
    wire \u1.y ;
    wire \u2.a ;
    wire \u2.b ;
    wire \u2.y ;
    output y;
    sky130_fd_sc_hd__and2_0 _6_ (
        .A( 1),
        .B( 2),
        .C( 3),
        .D( 4),
        .Y(y)
    );
endmodule

multiple_modules_heir.v [R0]          17,1      Top multiple_modules_flat.v          17,1      Top
```

<pre> module sub_module1(a, b, y);     wire _0_;     wire _1_;     wire _2_;     input a;     input b;     output y;     sky130_fd_sc_hd_and2_0 _3_ (         .A(_1_),         .B(_0_),         .X(_2_)     );     assign _1_ = b;     assign _0_ = a;     assign y = _2_; endmodule  module sub_module2(a, b, y);     wire _0_;     wire _1_;     wire _2_;     input a;     input b;     output y;     sky130_fd_sc_hd_lpfflow_inputisop1 _7_ (         .A(_4_),         .SLEEP(_3_),         .X(_5_)     );     assign _4_ = \u2.b ;     assign _3_ = \u2.a ;     assign \u2.y = _5_ ;     assign \u2.a = net1;     assign \u2.b = c;     assign y = \u2.y ;     assign _1_ = \u1.b ;     assign _0_ = \u1.a ;     assign \u1.y = _2_ ;     assign \u1.a = a;     assign \u1.b = b;     assign net1 = \u1.y ; endmodule </pre>	multiple modules heir.v [R0]	26,1	66%	multiple_modules_flat.v	38,22	Bot
<pre>     .X(_2_); ); assign _1_ = b; assign _0_ = a; assign y = _2_; endmodule  module sub_module2(a, b, y);     wire _0_;     wire _1_;     wire _2_;     input a;     input b;     output y;     sky130_fd_sc_hd_lpfflow_inputisop1 _3_ (         .A(_1_),         .SLEEP(_0_),         .X(_2_)     );     assign _1_ = b;     assign _0_ = a;     assign y = _2_; endmodule </pre>	multiple_modules_heir.v [R0]	53,9	Bot	multiple_modules_flat.v	43,9	Bot

Command: show multiple\_modules



## SYNTHESIS OF INDIVIDUAL MODULES:

Command:yosys

Command:read\_liberty –lib ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:read\_verilog multiple\_modules.v

```
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ yosys
-----
|   yosys -- Yosys Open SYnthesis Suite
|   Copyright (C) 2012 - 2020 Claire Xenia Wolf <claire@yosyshq.com>
|   Permission to use, copy, modify, and/or distribute this software for any
|   purpose with or without fee is hereby granted, provided that the above
|   copyright notice and this permission notice appear in all copies.

yosys> read_liberty -lib ..../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog multiple_modules.v

2. Executing Verilog-2005 frontend: multiple_modules.v
Parsing Verilog input from 'multiple_modules.v' to AST representation.
Generating RTLIL representation for module '\sub_module2'.
Generating RTLIL representation for module '\sub_module1'.
Generating RTLIL representation for module '\multiple_modules'.
Successfully finished Verilog frontend.

yosys> █
```

Command:synth –top sub\_module1

```
3.25. Printing statistics.

==== sub_module1 ===

Number of wires: 3
Number of wire bits: 3
Number of public wires: 3
Number of public wire bits: 3
Number of memories: 0
Number of memory bits: 0
Number of processes: 0
Number of cells: 1
$_AND_ 1

3.26. Executing CHECK pass (checking for obvious problems).
Checking module sub_module1...
Found and reported 0 problems.

yosys> █
```

```
Command: abc -liberty ..//my-lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
```

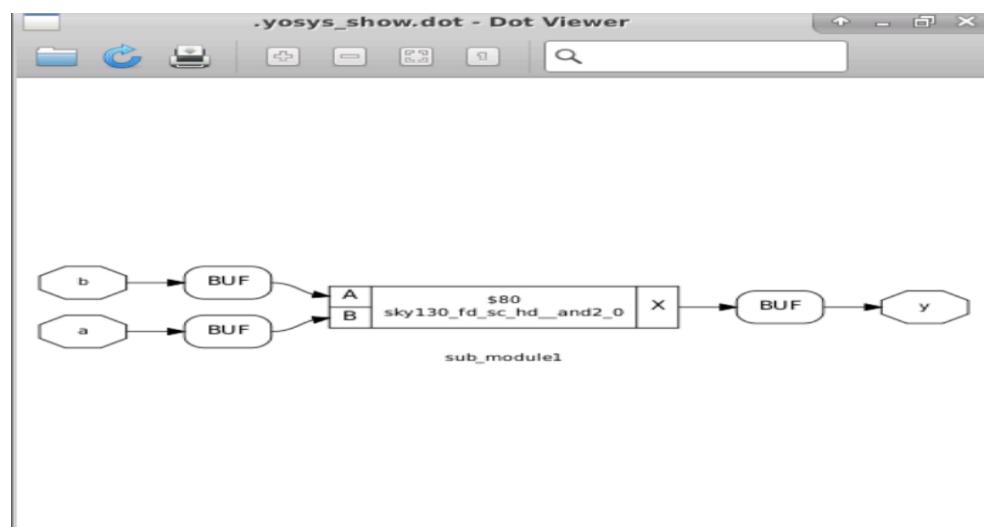
```
Command: show sub_module1
```

```
Command: write -noattr sub_moudle_netlist.v
```

```
Command: !gvim sub_module1_netlist.v
```

```
yosys> show sub_module1
5. Generating Graphviz representation of design.
Writing dot description to '/home/tvsmohan99/.yosys_show.dot'.
Dumping module sub_module1 to page 1.
Exec: { test -f '/home/tvsmohan99/.yosys_show.dot.pid' && fuser -s '/home/tvsmohan99/.yosys_show.dot.pid' 2> /dev/null; } || ( echo $$ >&2; exec xdot '/home/tvsmohan99/.yosys_show.dot'; ) 3> '/home/tvsmohan99/.yosys_show.dot.pid' &
yosys> write_verilog -noattr sub_module1_netlist.v
6. Executing Verilog backend.
Dumping module '\sub_module1'.
yosys> !gvim sub_module1_netlist.v
7. Shell command: gvim sub_module1_netlist.v
yosys>
```

```
File Edit Tools Syntax Buffers Window Help
File Edit Tools Syntax Buffers Window Help
/* Generated by Yosys 0.9+4081 (git sha1 862e84eb, gcc 7.5.0-3ubuntu1-18.04 -fPIC -Os) */
module sub_module1(a, b, y);
    wire _0_;
    wire _1_;
    wire _2_;
    input a;
    input b;
    output y;
    sky130_fd_sc_hd_and2_0 _3_ (
        .A(_1_),
        .B(_0_),
        .X(_2_)
    );
    assign _1_ = b;
    assign _0_ = a;
    assign y = _2_;
endmodule
```



## SKY130RTL D2SK3 L3 Lab flop synthesis simulations part1:

### DFF\_ASYNCRES IVERILOG SIMULATION

Command: iverilog dff\_asyncres.v tb\_dff\_asyncres.v

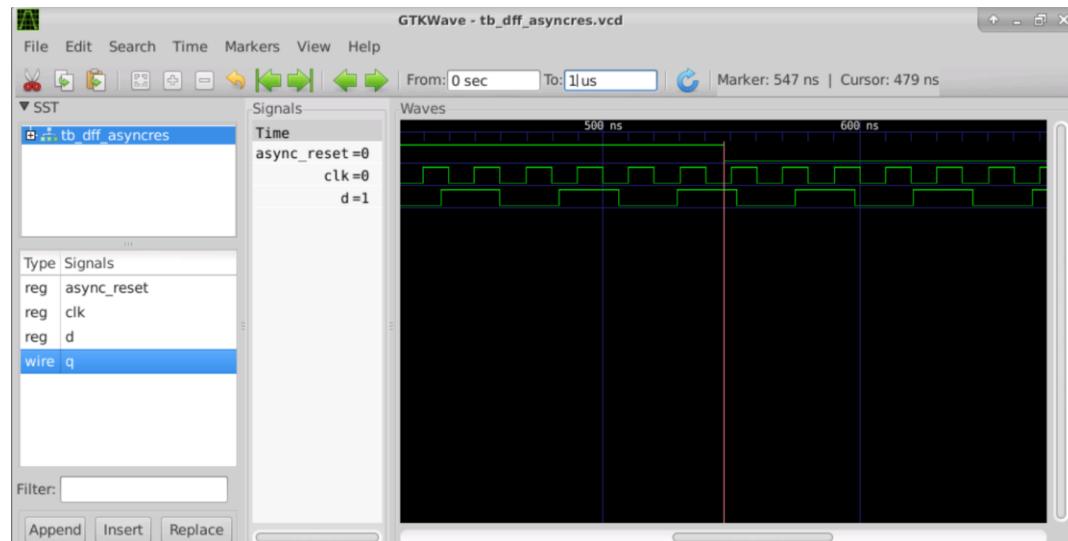
Command: ./a.out

Command: gtkwave tb\_dff\_asyncres.vcd

```
tvsimohan99@rtlworkshop-04:/VLSI/sky130RTLDdesignAndSynthesisWorkshop/verilog_files$ iverilog dff_asyncres.v tb_dff_asyncres.v
tvsimohan99@rtlworkshop-04:/VLSI/sky130RTLDdesignAndSynthesisWorkshop/verilog_files$ ./a.out
VCD info: dumpfile tb_dff_asyncres.vcd opened for output.
tvsimohan99@rtlworkshop-04:/VLSI/sky130RTLDdesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_dff_asyncres.vcd

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI
Error opening .vcd file 'tb_dff_asyncres.vcd'.
Why: No such file or directory
tvsimohan99@rtlworkshop-04:/VLSI/sky130RTLDdesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_dff_asyncres.vcd

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI
[0] start time,
[3000000] end time.
```



OBSERVATIONS: If there is no reset, the state changes only with posedge of clock.

Else, the output changes with reset.

### DFF\_ASYNC\_SET IVERILOG SIMULATION:

Command: iverilog dff\_async\_set.v tb\_dff\_async\_set.v

Command: ./a.out

Command: gtkwave tb\_dff\_async\_set.vcd

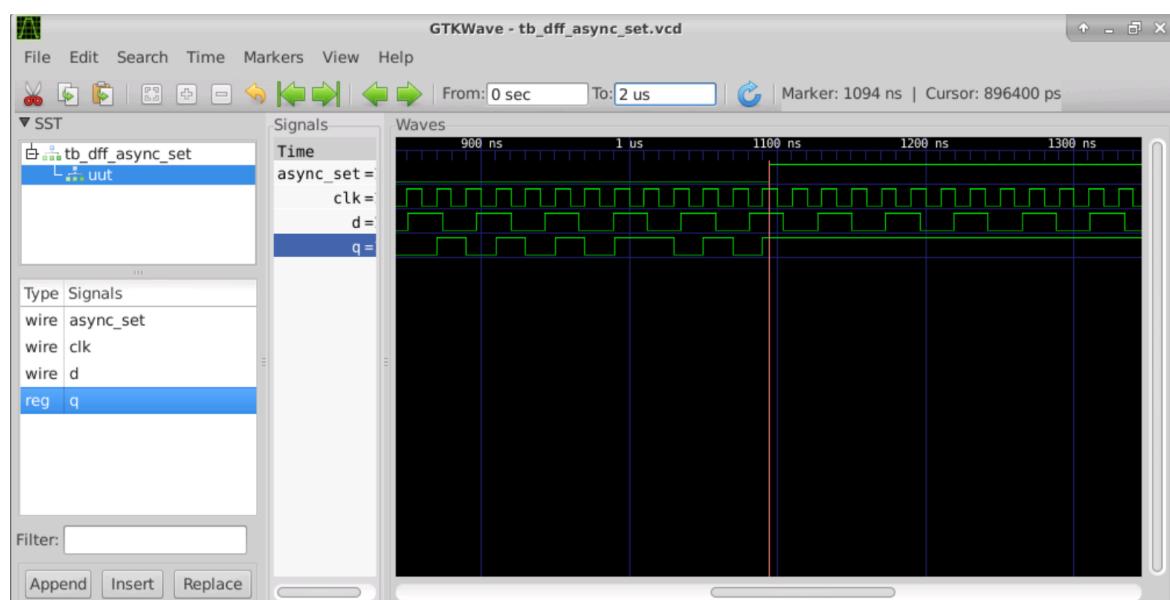
```

tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ iverilog dff_async_set.v tb_dff_async_set.v
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ ./a.out
VCD info: dumpfile tb_dff_async_set.vcd opened for output.
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_dff_async_set.vcd

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

[0] start time,
[3000000] end time.

```



OBSERVATIONS: The q will be set without waiting for the clock.

The q value will wait till the posedge of clock.

#### **DFF\_SYNCRES IVERILOG SIMULATION:**

Command: iverilog dff\_syncres.v tb\_dff\_syncres.v

Command: ./a.out

Command: gtkwave tb\_dff\_syncres.vcd

OBSERVATIONS: The q will change only with the posedge of the clock even when the sync\_reset is on.

```

tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ iverilog dff_syncres.v tb_dff_syncres.v
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ ./a.out
VCD info: dumpfile tb_dff_syncres.vcd opened for output.
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_dff_syncres.vcd

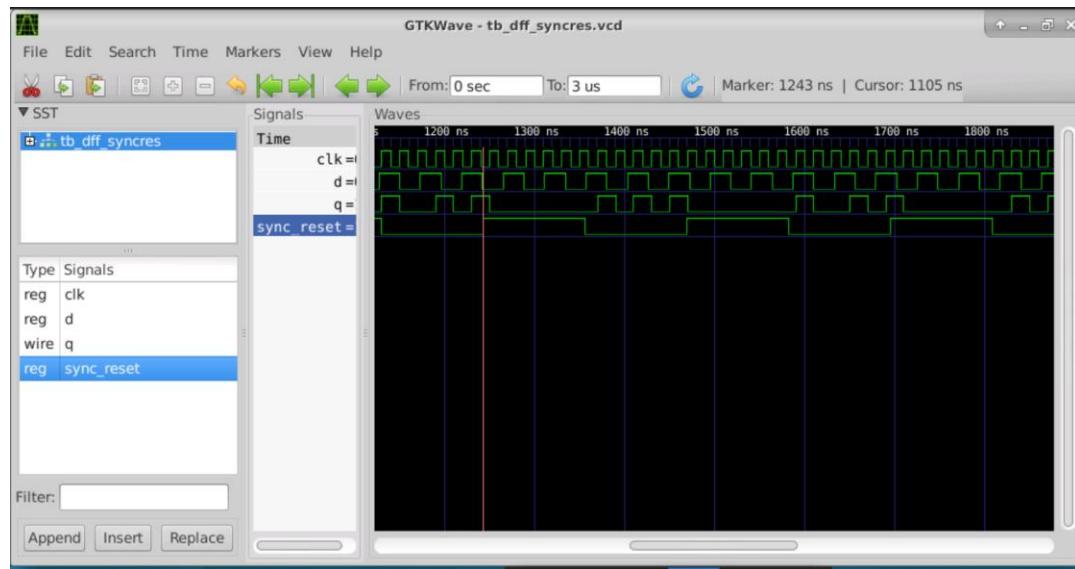
GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

Near byte 16, VCD search table NULL..is this a VCD file?
Near byte 16, Unknown VCD identifier: 'ns'
No symbols in VCD file..is it malformed? Exiting!
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_dff_syncres.vcd

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

[0] start time.
[3000000] end time.

```



## SKY130RTL D2SK3 L4 Lab flop synthesis simulations part2:

### DFF\_ASYNCRES SYNTHESIS:

Command: `read_liberty -lib ..//my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib`

Command: `read_verilog dff_asyncres.v`

Command: `synth -top dff_asyncres`

```
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ yosys
-----
| yosys -- Yosys Open SYnthesis Suite
| Copyright (C) 2012 - 2020 Claire Xenia Wolf <claire@yosyshq.com>
|
| Permission to use, copy, modify, and/or distribute this software for any
| purpose with or without fee is hereby granted, provided that the above
| copyright notice and this permission notice appear in all copies.
|
| THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
| WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
| MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
| ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
| WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
| ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
| OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
|-----\

\\-----/
Yosys 0.9+4081 (git sha1 862e84eb, gcc 7.5.0-3ubuntu1-18.04 -fPIC -Os)

yosys> read_liberty -lib ../my_lib/lib/sky130_fd_sc_hd_tt_025C_lv80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog dff_asyncres.v

2. Executing Verilog-2005 frontend: dff_asyncres.v
Parsing Verilog input from 'dff_asyncres.v' to AST representation.
Generating RTLIL representation for module `\\dff_asyncres'.
Successfully finished Verilog frontend.

yosys> █
Top module: \dff_asyncres
3.24.2. Analyzing design hierarchy..
Top module: \dff_asyncres
Removed 0 unused modules.

3.25. Printing statistics.

==== dff_asyncres ====
Number of wires: 4
Number of wire bits: 4
Number of public wires: 4
Number of public wire bits: 4
Number of memories: 0
Number of memory bits: 0
Number of processes: 0
Number of cells: 1
$_DFF_PP0_

3.26. Executing CHECK pass (checking for obvious problems).
Checking module dff_asyncres...
Found and reported 0 problems.

yosys>
```

Command: dfflibmap -liberty ./my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

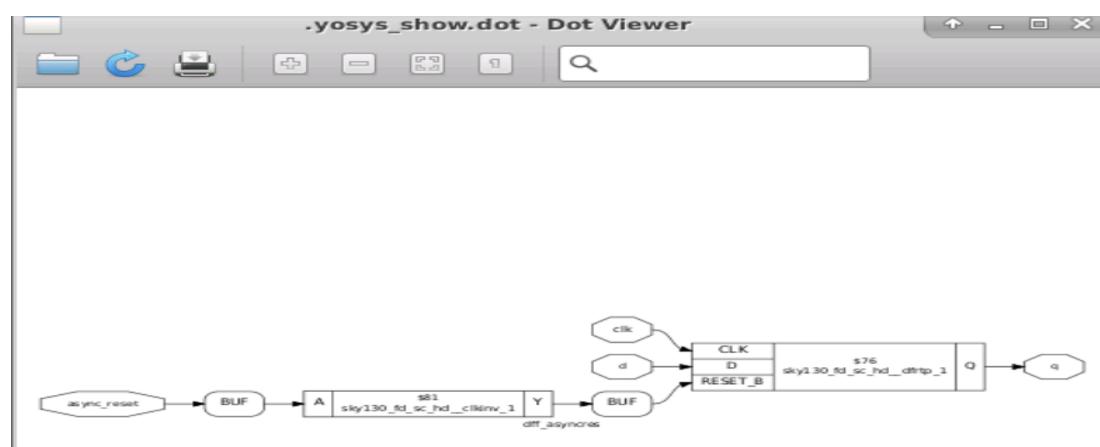
This command helps if the flop library and standard cell library are different.

```
finaldff cell mappings:  
unmapped dff cell: $_DFF_N  
\sky130_fd_sc_hd_dfrtp1_DFF_P (.CLK( C ), .D( D ), .Q( Q ));  
\sky130_fd_sc_hd_dfrtn1_DFF_NN0 (.CLK_N( C ), .D( D ), .Q( Q ), .RESET_B( R ));  
unmapped dff cell: $_DFF_NN1  
unmapped dff cell: $_DFF_NP0  
unmapped dff cell: $_DFF_NP1  
\sky130_fd_sc_hd_dfrtp1_DFF_PN0 (.CLK( C ), .D( D ), .Q( Q ), .RESET_B( R ));  
\sky130_fd_sc_hd_dfstp2_DFF_PN1 (.CLK( C ), .D( D ), .Q( Q ), .SET_B( R ));  
unmapped dff cell: $_DFF_PPO  
unmapped dff cell: $_DFF_PPI  
\sky130_fd_sc_hd_dfbnn1_DFFSR_NNN (.CLK_N( C ), .D( D ), .Q( Q ), .Q_N(~Q), .RESET_B( R ), .SET_B( S ));  
unmapped dff cell: $_DFFSR_NNP  
unmapped dff cell: $_DFFSR_NPN  
unmapped dff cell: $_DFFSR_NPP  
\sky130_fd_sc_hd_dfbbp1_DFFSR_PNN (.CLK( C ), .D( D ), .Q( Q ), .Q_N(~Q), .RESET_B( R ), .SET_B( S ));  
unmapped dff cell: $_DFFSR_PNP  
unmapped dff cell: $_DFFSR_PPN  
unmapped dff cell: $_DFFSR_PPP  
  
4.1. Executing DFFLEGALIZE pass (convert FFs to types supported by the target).  
Mapping DFF cells in module '\dff.asynres':  
    mapped 1 $_DFF_PN0 cells to \sky130_fd_sc_hd_dfrtp1 cells.  
  
yosys> █
```

Command: abc -liberty ./my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

```
5.1.2. Re-integrating ABC results.  
ABC RESULTS: sky130_fd_sc_hd_clkinv1 cells: 1  
ABC RESULTS: internal signals: 0  
ABC RESULTS: input signals: 1  
ABC RESULTS: output signals: 1  
Removing temp directory.  
  
yosys> █
```

Command: show



OBSERVATIONS: The use of inverse gate in the synthesis indicates that the flop, in the library, has active low reset as input.

#### DFF\_ASYNC\_SET SYNTHESIS:

Command: read\_verilog dff\_async\_set.v

Command: synth -top dff\_async\_set

Command: dfflibmap -liberty ../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: abc -liberty ../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: show

#### 8.25. Printing statistics.

==== dff\_async\_set ===

Number of wires:	4
Number of wire bits:	4
Number of public wires:	4
Number of public wire bits:	4
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	1
\$_DFF_PP1_	1

#### 8.26. Executing CHECK pass (checking for obvious problems).

Checking module dff\_async\_set...

Found and reported 0 problems.

yosys> █

#### 9.1. Executing DFFLEGALIZE pass (convert FFs to types supported by the target).

Mapping DFF cells in module '\dff\_async\_set':

    mapped 1 \$\_DFF\_PP1\_ cells to \sky130\_fd\_sc\_hd\_dfstp\_2 cells.

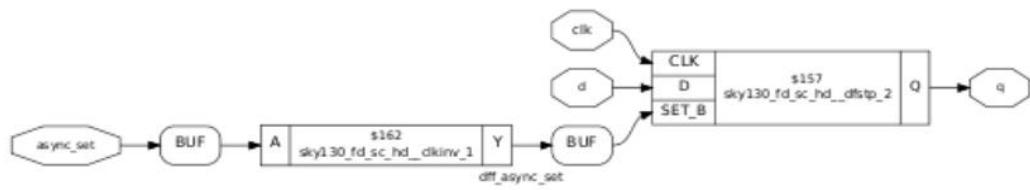
yosys>

11.1.2. Re-integrating ABC results.  
ABC RESULTS: sky130\_fd\_sc\_hd\_clkinv\_1 cells: 1  
ABC RESULTS: internal signals: 0  
ABC RESULTS: input signals: 1  
ABC RESULTS: output signals: 1  
Removing temp directory.

yosys> show

12. Generating Graphviz representation of design.  
Writing dot description to '/home/tvsmohan99/yosys\_show.dot'.  
Dumping module dff\_async\_set to page 1.  
Exec: { test -f '/home/tvsmohan99/yosys\_show.dot.pid' && fuser -s '/home/tvsmohan99/yosys\_show.dot.pid' 2> /dev/null; } || ( echo \$\$ >&3; exec xdot '/home/tvsmohan99/yosys\_show.dot'; ) 3> '/home/tvsmohan99/yosys\_show.dot.pid' &

yosys>



### **DFF\_SYNCRES SYNTHESIS:**

Command: read\_liberty –lib ..//my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: read\_verilog dff\_syncres.v

Command: synth –top dff\_syncres

Command: dfflibmap –liberty ..//my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: abc –liberty ..//my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: show

```
15.25. Printing statistics.
```

```
== dff_syncres ==
```

```
Number of wires:      5
Number of wire bits:  5
Number of public wires: 5
Number of public wire bits: 5
Number of memories:   0
Number of memory bits: 0
Number of processes:   0
Number of cells:      1
$_SDFF_PP0_           1
```

```
15.26. Executing CHECK pass (checking for obvious problems).
```

```
Checking module dff_syncres...
```

```
Found and reported 0 problems.
```

```
yosys>
```

```
6.1.2. Re-integrating ABC results.
```

```
ABC RESULTS: sky130_fd_sc_hd_lpfw_isobufsrc_1 cells:      1
```

```
ABC RESULTS: internal signals:      1
```

```
ABC RESULTS: input signals:        2
```

```
ABC RESULTS: output signals:       1
```

```
Removing temp directory.
```

```
yosys> show
```

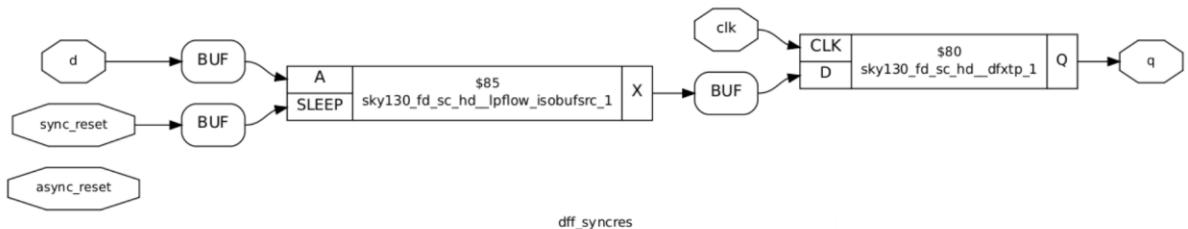
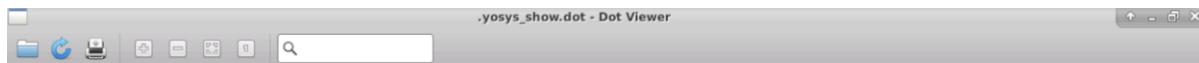
```
7. Generating Graphviz representation of design.
```

```
Writing dot description to '/home/tvsmohan99/.yosys_show.dot'.
```

```
Dumping module dff_syncres to page 1.
```

```
Exec: { test -f '/home/tvsmohan99/.yosys_show.dot.pid' && fuser -s '/home/tvsmohan99/.yosys_show.dot.pid' 2> /dev/null; } || ( echo $$ >&3; exec xdot '/home/tvsmohan99/.yosys_show.dot'; ) 3> '/home/tvsmohan99/.yosys_show.dot.pid' &
```

```
yosys>
```



## **SKY130RTL D2SK3 L5 Interesting optimisations part1:**

Command: gvim mult\_\*.v -o

This command helps in opening multiple files, with mult as starting four letters.

```
tvs mohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gvim mult_*.v -o
2 files to edit
tvs mohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$
```



```
mult_2.v (~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files) (1 of 2) - GVIM
File Edit Tools Syntax Buffers Window Help
File Open Save All Save As Find Replace Go To Insert Run Macros Plugins
mult_2.v
module mul2 (input [2:0] a, output [3:0] y);
    assign y = a * 2;
endmodule
~
~
~
~
~
~
~
mult_2.v
module mult8 (input [2:0] a, output [5:0] y);
    assign y = a * 9;
endmodule
~
~
~
~
~
~
mult_8.v
"mult_8.v" 4L, 77C
1,1      All
1,1      All
```

### **OBSERVATIONS:**

1) The first file mult\_2.v gives 4-bit output after multiplication of 2 and 3-bit input.

The output is nothing but appending a '0' at the right side of the input.

2) The second file mult\_8.v gives 6-bit output after multiplication of 9 and 3-bit input.

The output can be generated by appending the input with itself.

### **MULT\_2 SYNTHESIS:**

Command: yosys

Command: read\_liberty -lib ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: read\_verilog mult\_2.v

Command: synth -top mul2

Command: abc -liberty ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: show

```
yosys> read_liberty -lib ../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog mult_2.v

2. Executing Verilog-2005 frontend: mult_2.v
Parsing Verilog input from 'mult_2.v' to AST representation.
Generating RTLIL representation for module '\mul2'.
Successfully finished Verilog frontend.

yosys> █
```

### 3.25. Printing statistics.

```
==== mul2 ===
```

Number of wires:	2
Number of wire bits:	7
Number of public wires:	2
Number of public wire bits:	7
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	0

### 3.26. Executing CHECK pass (checking for obvious problems).

Checking module mul2...

Found and reported 0 problems.

```
yosys>
```

## OBSERVATIONS:

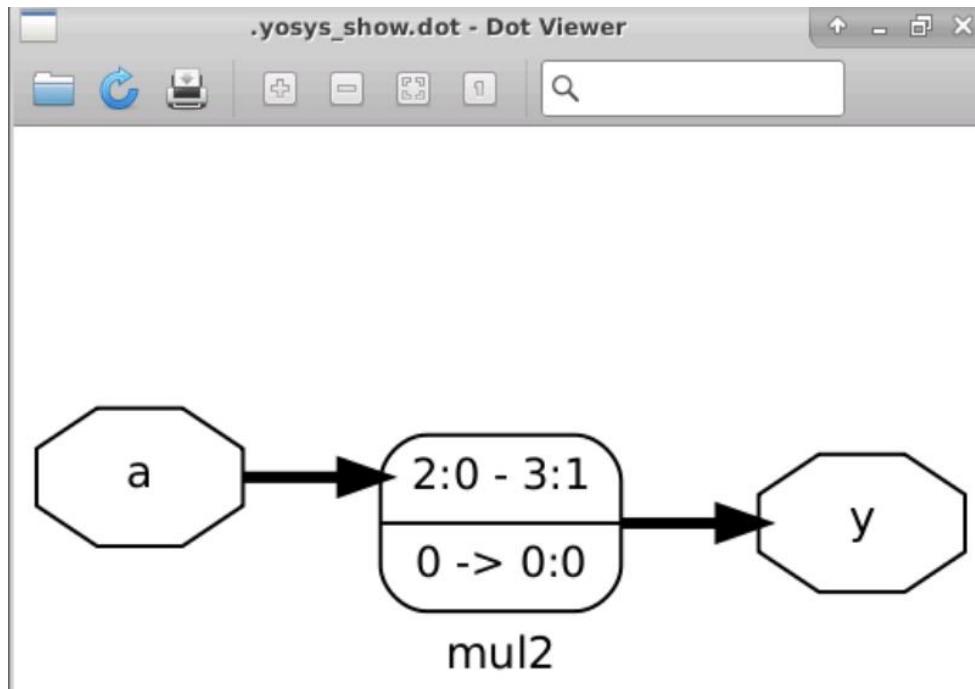
- 1) The print statistics indicates that no cells are required.
- 2) The netlist generation command shows “there is nothing to map”.

```
yosys> abc -liberty ../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
4. Executing ABC pass (technology mapping using ABC).

4.1. Extracting gate netlist of module '\mul2' to '<abc-temp-dir>/input.blif'..
Extracted 0 gates and 0 wires to a netlist network with 0 inputs and 0 outputs.
Don't call ABC as there is nothing to map.
Removing temp directory.

yosys> █
```

Command: show



OBSERVATION: As discussed before, the result is generated by appending the zero at right side.

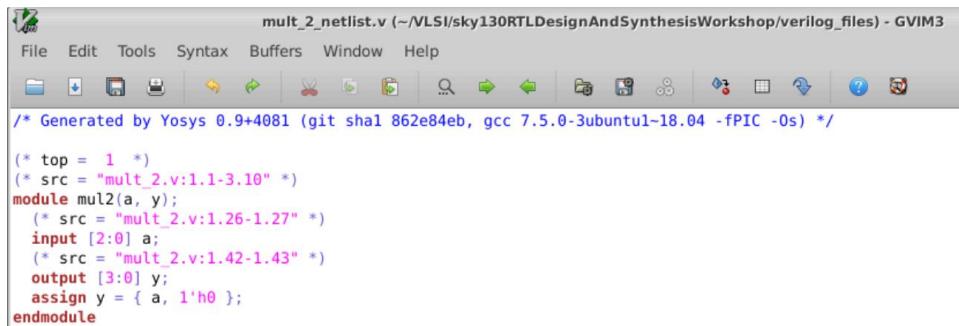
Command: write\_verilog –noattr mult\_2\_netlist.v

Command: !gvim mult\_2\_netlist.v

```
yosys> write_verilog mult_2_netlist.v
6. Executing Verilog backend.
Dumping module `mul2'.

yosys> !gvim mult_2_netlist.v
7. Shell command: gvim mult_2_netlist.v

yosys>
```



```

/* Generated by Yosys 0.9+4081 (git sha1 862e84eb, gcc 7.5.0-3ubuntu1-18.04 -fPIC -Os) */

(* top = 1 *)
(* src = "mult_2.v:1.1-3.10" *)
module mul2(a, y);
  (* src = "mult_2.v:1.26-1.27" *)
  input [2:0] a;
  (* src = "mult_2.v:1.42-1.43" *)
  output [3:0] y;
  assign y = { a, 1'h0 };
endmodule

```

### MULT\_8 SYNTHESIS:

Command:yosys

Command:read\_liberty–lib ..//my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:read\_verilog mult\_8.v

Command:synth–top mult8

Command:abc–liberty ..//my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:show

```

yosys> read_verilog mult_8.v
8. Executing Verilog-2005 frontend: mult_8.v
Parsing Verilog input from `mult_8.v' to AST representation.
Generating RTLIL representation for module `\mult8'.
Successfully finished Verilog frontend.

```

9.25. Printing statistics.

==== mult8 ===

Number of wires:	2
Number of wire bits:	9
Number of public wires:	2
Number of public wire bits:	9
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	0

9.26. Executing CHECK pass (checking for obvious problems).
Checking module mult8...
Found and reported 0 problems.

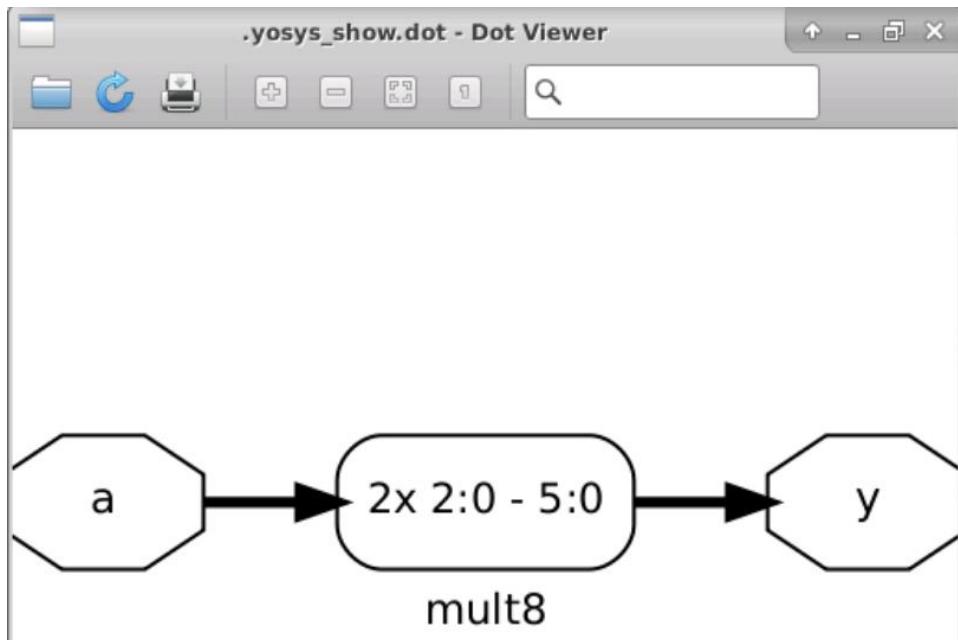
yosys> █

OBSERVATIONS: There are no standard cells used in this synthesis.

```
yosys> abc -liberty ../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
10. Executing ABC pass (technology mapping using ABC).

10.1. Extracting gate netlist of module `\'mult8' to `<abc-temp-dir>/input.blif'..
Extracted 0 gates and 0 wires to a netlist network with 0 inputs and 0 outputs.
Don't call ABC as there is nothing to map.
Removing temp directory.
```

OBSERVATIONS: The abc command doesn't want to be called without standard cells.



OBSERVATIONS: The show command indicates that the input is appended to itself and given as output.

Command : write\_verilog mult\_8\_netlist.v

Command: !gvim mult\_8\_netlist.v

```
yosys> write_verilog mult_8_netlist.v

12. Executing Verilog backend.
Dumping module `\'mult8'.

yosys> !gvim mult_8_netlist.v

13. Shell command: gvim mult_8_netlist.v

yosys>
```

```
/* Generated by Yosys 0.9+4081 (git sha1 862e84eb, gcc 7.5.0-3ubuntu1~18.04 -fPIC -Os) */

(* top = 1 *)
(* src = "mult_8.v:1.1-3.10" *)
module mult8(a, y);
    (* src = "mult_8.v:1.27-1.28" *)
    input [2:0] a;
    (* src = "mult_8.v:1.44-1.45" *)
    output [5:0] y;
    assign y = { a, a };
endmodule
```

### **DAY 3:**

#### **SKY130RTL D3SK2 L1 Lab06 Combinational Logic Optimisations part1:**

#### **OPT\_CHECK OPTIMISATION AND SYNTHESIS:**

Command: yosys

Command: read\_liberty -lib ./my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: read\_verilog opt\_check.v

Command: synth -top opt\_check

```
yosys> read_liberty -lib ./my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
2. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog opt_check.v
3. Executing Verilog-2005 frontend: opt_check.v
Parsing Verilog input from `opt_check.v' to AST representation.
Generating RTLIL representation for module `opt_check'.
Successfully finished Verilog frontend.
```

4.25. Printing statistics.

```
==== opt_check ====
Number of wires: 3
Number of wire bits: 3
Number of public wires: 3
Number of public wire bits: 3
Number of memories: 0
Number of memory bits: 0
Number of processes: 0
Number of cells: 1
$_AND_ 1
```

4.26. Executing CHECK pass (checking for obvious problems).
Checking module opt\_check...
Found and reported 0 problems.

```
yosys> █
```

Command: opt\_clean -purge

This command helps in identifying and removing the unused cells in the design.

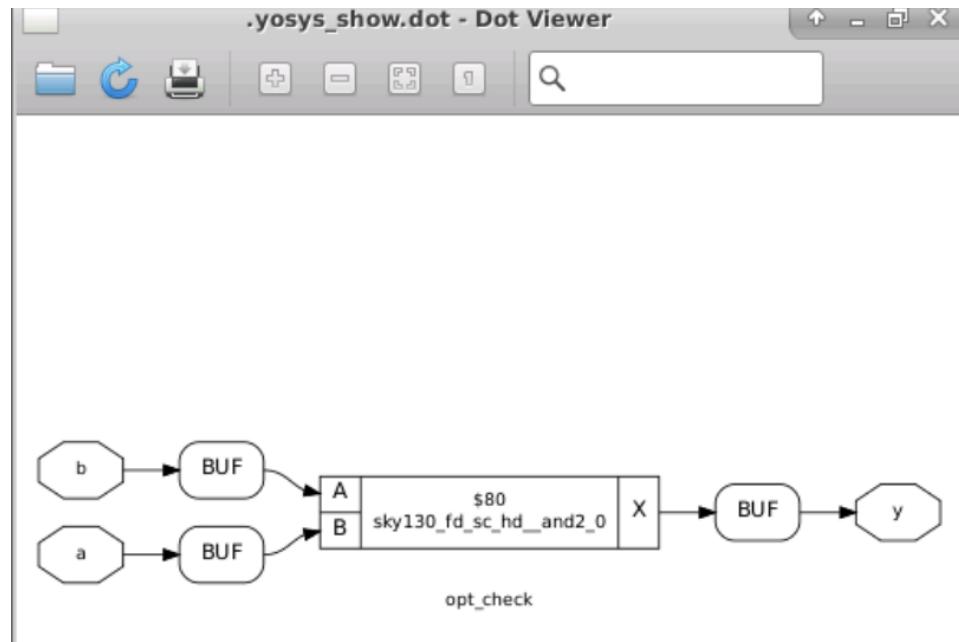
```
yosys> opt_clean -purge
5. Executing OPT_CLEAN pass (remove unused cells and wires).
Finding unused cells or wires in module \opt_check..
```

Command: abc -liberty ../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: show

```
6.1.2. Re-integrating ABC results.
ABC RESULTS: sky130_fd_sc_hd_and2_0 cells:      1
ABC RESULTS:   internal signals:      0
ABC RESULTS:   input signals:       2
ABC RESULTS:   output signals:      1
Removing temp directory.

yosys> show
7. Generating Graphviz representation of design.
Writing dot description to '/home/tvsmohan99/.yosys_show.dot'.
Dumping module opt check to page 1.
Exec: { test -f '/home/tvsmohan99/.yosys_show.dot.pid' && fuser -s '/home/tvsmohan99/.yosys_show.dot.pid' 2> /dev/null; } || ( echo $$
>&3; exec xdot '/home/tvsmohan99/.yosys_show.dot'; ) 3> '/home/tvsmohan99/.yosys_show.dot.pid' &
yosys>
```



#### OBSERVATIONS:

- 1) When one of the inputs of a multiplexer is zero, the design can be optimised as an AND gate.

## **OPT\_CHECK2 OPTIMISZATION AND SYNTHESIS:**

Command: yosys

Command: read\_liberty –lib ..//my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: read\_verilog opt\_check2.v

Command: synth –top opt\_check2

```
yosys> read_verilog opt_check2.v
8. Executing Verilog-2005 frontend: opt_check2.v
Parsing Verilog input from `opt_check2.v' to AST representation.
Generating RTLIL representation for module `\opt_check2'.
Successfully finished Verilog frontend.
```

9.25. Printing statistics.

```
== opt_check2 ==
Number of wires:          3
Number of wire bits:      3
Number of public wires:    3
Number of public wire bits: 3
Number of memories:       0
Number of memory bits:    0
Number of processes:      0
Number of cells:          1
$_OR_                   1
```

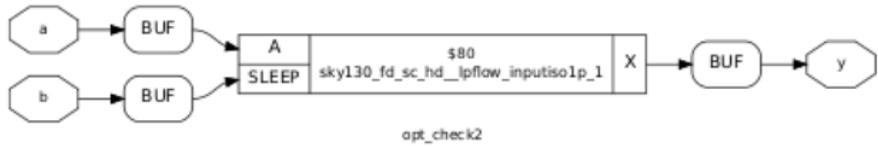
9.26. Executing CHECK pass (checking for obvious problems).
Checking module opt\_check2...
Found and reported 0 problems.

Command: opt\_clean –purge

Command: abc –liberty ..//my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: show

```
5.1.2. Re-integrating ABC results.
ABC RESULTS: sky130_fd_sc_hd_lpflow_inputisop_1 cells:      1
ABC RESULTS:      internal signals:      0
ABC RESULTS:      input signals:        2
ABC RESULTS:      output signals:       1
Removing temp directory.
```



#### OBSERVATIONS:

- 1) When one of the inputs of multiplexer is one, the design can be optimised as a OR gate.
- 2) Instead of generating OR gate from Nand gates, the updated standard cell library had an OR gate.

#### **SKY130RTL D3SK2 L2 Lab06 Combinational Logic Optimisations part2:**

##### **OPT\_CHECK3 OPTIMIZATION AND SYNTHESIS :**

Command:yosys

Command:read\_liberty–lib ..//my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:read\_verilog opt\_check3.v

Command:synth–top opt\_check3

```
yosys> read_verilog opt_check3.v
7. Executing Verilog-2005 frontend: opt_check3.v
Parsing Verilog input from `opt_check3.v' to AST representation.
Generating RTLIL representation for module `\opt_check3'.
Successfully finished Verilog frontend.
```

Command:opt\_clean–purge

Command:abc–liberty ..//my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:show

### 8.25. Printing statistics.

```
==== opt_check3 ====  
  
Number of wires: 5  
Number of wire bits: 5  
Number of public wires: 4  
Number of public wire bits: 4  
Number of memories: 0  
Number of memory bits: 0  
Number of processes: 0  
Number of cells: 2  
$_ANDNOT_ 1  
$_NAND_ 1
```

### 8.26. Executing CHECK pass (checking for obvious problems).

Checking module opt\_check3...  
Found and reported 0 problems.

```
yosys> █
```

```
yosys> opt_clean -purge
```

9. Executing OPT\_CLEAN pass (remove unused cells and wires).  
Finding unused cells or wires in module \opt\_check3..

```
yosys> abc -liberty ../../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
```

### 10.1.2. Re-integrating ABC results.

```
ABC RESULTS: sky130_fd_sc_hd_and3_1 cells: 1  
ABC RESULTS: internal signals: 1  
ABC RESULTS: input signals: 3  
ABC RESULTS: output signals: 1
```

Removing temp directory.

```
yosys> █
```

```
yosys> show
```

11. Generating Graphviz representation of design.  
Writing dot description to '/home/tvsmohan99/.yosys\_show.dot'.  
Dumping module opt\_check3 to page 1.  
Exec: { test -f '/home/tvsmohan99/.yosys\_show.dot.pid' && fuser -s '/home/tvsmohan99/.yosys\_show.dot.pid' 2> /dev/null; } || ( echo \$\$ >&3; exec xdot '/home/tvsmohan99/.yosys\_show.dot'; ) 3> '/home/tvsmohan99/.yosys\_show.dot.pid' &

```
yosys>
```

OBSERVATIONS: The two multiplexers in the opt\_check3.v are optimised as single 3 input AND gate. Because, one the inputs of the two multiplexers are zero they are optimised to AND gates as in opt\_check.v.



#### **OPT\_CHECK4.V OPTIMISATION AND SYNTHESIS:**

Command: yosys

Command: read\_liberty -lib ./my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: read\_verilog opt\_check4.v

Command: synth -top opt\_check4

Command: opt\_clean -purge

Command: abc -liberty ./my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: show

```

yosys> read_liberty -lib ./my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog opt_check4.v
2. Executing Verilog-2005 frontend: opt_check4.v
Parsing Verilog input from `opt_check4.v' to AST representation.
Generating RTLIL representation for module `opt_check4'.
Successfully finished Verilog frontend.

yosys> █

```

```
== opt_check4 ==
```

```
Number of wires:          7
Number of wire bits:      7
Number of public wires:   4
Number of public wire bits: 4
Number of memories:      0
Number of memory bits:    0
Number of processes:      0
Number of cells:          4
  $_AND_                  1
  $_MUX_                  2
  $_NOT_                  1
```

4.26. Executing CHECK pass (checking for obvious problems).

```
Checking module opt_check4...
Found and reported 0 problems.
```

```
yosys> opt_clean -purge
```

5. Executing OPT\_CLEAN pass (remove unused cells and wires).  
Finding unused cells or wires in module \opt\_check4...

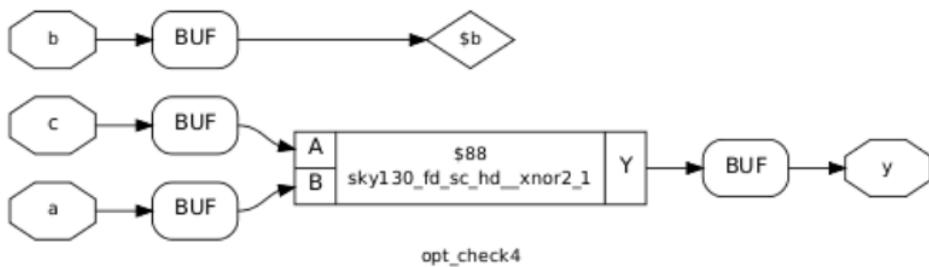
```
yosys> [REDACTED]
```

```
6.1.2. Re-integrating ABC results.
ABC RESULTS:  sky130_fd_sc_hd_xnor2_1 cells:      1
ABC RESULTS:  internal signals:      3
ABC RESULTS:  input signals:        3
ABC RESULTS:  output signals:       1
Removing temp directory.
```

```
yosys> show
```

```
7. Generating Graphviz representation of design.
Writing dot description to '/home/tvsmohan99/.yosys_show.dot'.
Dumping module opt_check4 to page 1.
Exec: { test -f '/home/tvsmohan99/.yosys_show.dot.pid' && fuser -s '/home/tvsmohan99/.yosys_show.dot.pid' 2> /dev/null; } || ( echo $$
>&3; exec xdot '/home/tvsmohan99/.yosys_show.dot'; ) 3> '/home/tvsmohan99/.yosys_show.dot.pid' &
```

```
yosys>
```



OBSERVATIONS: The opt\_check4.v design is simplified as an XNOR gate.

**MULTIPLE\_MODULE\_OPT.V OPTIMIZATION AND SYNTHESIS:**

Command:yosys

Command:read\_liberty –lib ../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:read\_verilog multiple\_module\_opt.v

Command:synth –top multiple\_module\_opt

Command:flatten

Command:opt\_clean –purge

Command:abc –liberty ../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:show

**9.25. Printing statistics.**

==== sub\_module1 ===

Number of wires:	3
Number of wire bits:	3
Number of public wires:	3
Number of public wire bits:	3
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	1
\$AND_	1

```
==== multiple_module_opt ===
```

Number of wires:	7
Number of wire bits:	7
Number of public wires:	6
Number of public wire bits:	6
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	3
\$_AND_	1
\$_OR_	1
sub_module1	1

```
==== design hierarchy ===
```

multiple_module_opt	1
sub_module1	1
Number of wires:	10
Number of wire bits:	10
Number of public wires:	9
Number of public wire bits:	9
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	3
\$_AND_	2
\$_OR_	1

```
9.26. Executing CHECK pass (checking for obvious problems).  
Checking module sub_module1...  
Checking module multiple_module_opt...  
Found and reported 0 problems.
```

```
yosys> flatten
```

```
10. Executing FLATTEN pass (flatten design).  
Deleting now unused module sub_module1.  
<suppressed ~1 debug messages>
```

```

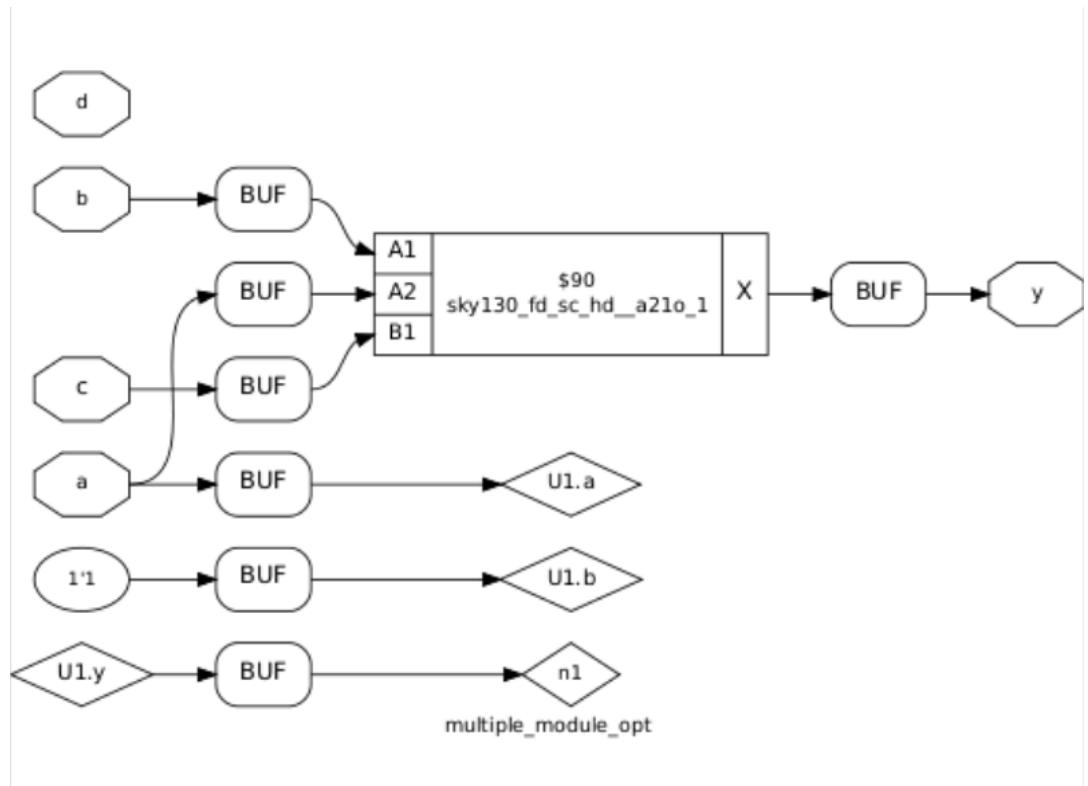
5.1.2. Re-integrating ABC results.
ABC RESULTS: sky130_fd_sc_hd_a2lo_1 cells:      1
ABC RESULTS:      internal signals:      3
ABC RESULTS:      input signals:      3
ABC RESULTS:      output signals:      1
Removing temp directory.

yosys> show

6. Generating Graphviz representation of design.
Writing dot description to '/home/tvsmohan99/.yosys_show.dot'.
Dumping module multiple_module_opt to page 1.
Exec: { test -f '/home/tvsmohan99/.yosys_show.dot.pid' && fuser -s '/home/tvsmohan99/.yosys_show.dot.pid' 2> /dev/null; } || ( echo $$
>&3; exec xdot '/home/tvsmohan99/.yosys_show.dot'; ) 3> '/home/tvsmohan99/.yosys_show.dot.pid' &

yosys> █

```



Command: write\_verilog –noattr multiple\_module\_opt\_netlist.v

Command: !gvim multiple\_module\_opt\_netlist.v

```

yosys> write_verilog -noattr multiple_module_opt_netlist.v

7. Executing Verilog backend.
Dumping module '\multiple_module_opt'.

yosys> !gvim multiple_module_opt.v

8. Shell command: gvim multiple_module_opt.v

yosys> !gvim multiple_module_opt_netlist.v

9. Shell command: gvim multiple_module_opt_netlist.v

```

```

multiple_module_opt_netlist.v (~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files) - GVIM1
File Edit Tools Syntax Buffers Window Help
File Edit Tools Syntax Buffers Window Help
/* Generated by Yosys 0.9+4081 (git sha1 862e84eb, gcc 7.5.0-3ubuntu1~18.04 -fPIC -Os) */

module multiple_module_opt(a, b, c, d, y);
    wire _0;
    wire _1;
    wire _2;
    wire _3;
    wire _4;
    wire \U1.a;
    wire \U1.b;
    wire \U1.y;
    input a;
    input b;
    input c;
    input d;
    wire n1;
    output y;
    sky130_fd_sc_hd_a2lo_1 _5_ (
        .A1(_2_),
        .A2(_1_),
        .B1(_3_),
        .X(_4_)
    );
    assign \U1.a = a;
    assign \U1.b = 1'h1;
    assign n1 = \U1.y ;
    assign _2_ = b;
    assign _3_ = c;
    assign y = _4_;
    assign _1_ = a;
endmodule

```

#### OBSERVATIONS:

- 1)The multiple\_module\_opt.v is optimised to a two input AND gate and two input OR gate using boolean logic optimization.
- 2)The U1 module is simplified to a.

## **MULTIPLE\_MODULE\_OPT2.V OPTIMIZATION AND SYNTHESIS:**

Command:yosys

Command:read\_liberty–lib ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:read\_verilog multiple\_module\_opt.v

Command:synth–top multiple\_module\_opt2

Command:flatten

Command:opt\_clean–purge

Command:abc–liberty ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:show

Command:write\_verilog–noattr multiple\_module\_opt2\_netlist.v

Command:!gvim multiple\_module\_opt2\_netlist.v

```
yosys> read_liberty -lib ..../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog multiple_module_opt2.v

2. Executing Verilog-2005 frontend: multiple_module_opt2.v
Parsing Verilog input from `multiple_module_opt2.v' to AST representation.
Generating RTLIL representation for module `\sub_module'.
Generating RTLIL representation for module `\multiple_module_opt2'.
Successfully finished Verilog frontend.
```

3.25. Printing statistics.

```
== multiple_module_opt2 ==
```

Number of wires:	8
Number of wire bits:	8
Number of public wires:	8
Number of public wire bits:	8
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	4
sub_module	4

```
== sub_module ==

Number of wires:          3
Number of wire bits:      3
Number of public wires:    3
Number of public wire bits: 3
Number of memories:       0
Number of memory bits:    0
Number of processes:      0
Number of cells:          1
$_AND_                   1
```

```
== design hierarchy ==

multiple_module_opt2           1
  sub_module                     4

Number of wires:              20
Number of wire bits:          20
Number of public wires:        20
Number of public wire bits:    20
Number of memories:           0
Number of memory bits:         0
Number of processes:          0
Number of cells:              4
$_AND_                          4
```

```
yosys> flatten

4. Executing FLATTEN pass (flatten design).
Deleting now unused module sub_module.
<suppressed ~4 debug messages>

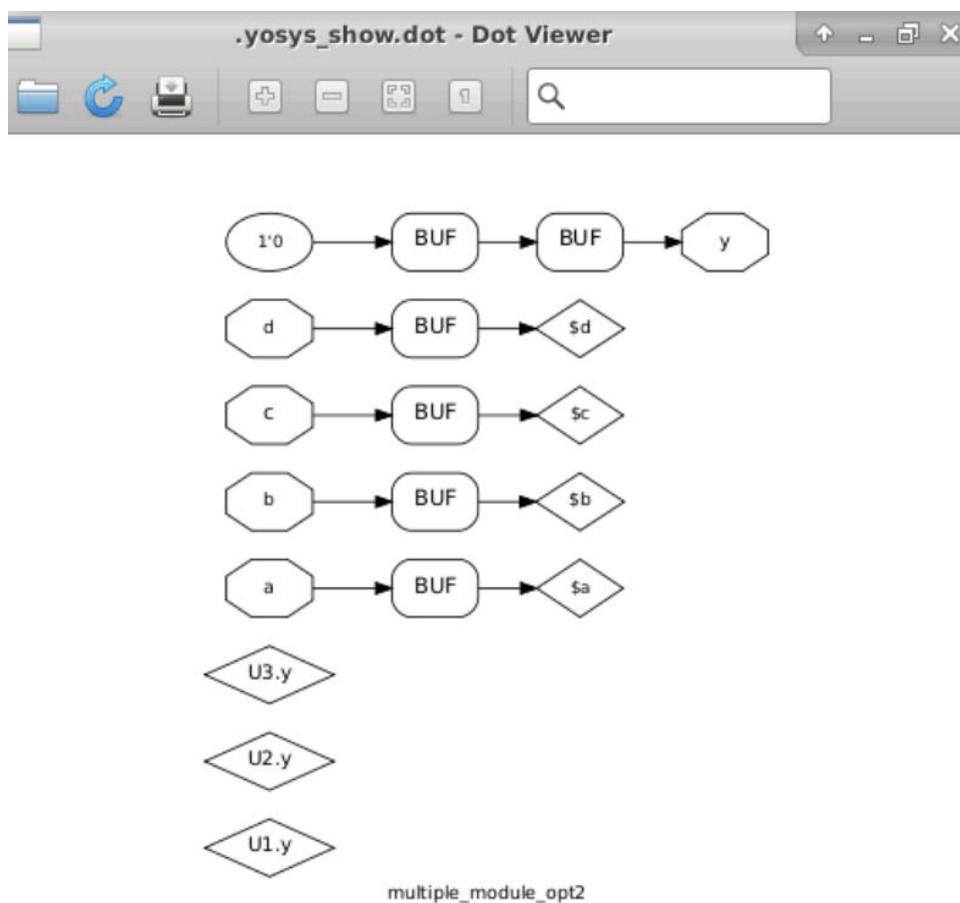
yosys> opt_clean -purge

5. Executing OPT_CLEAN pass (remove unused cells and wires).
Finding unused cells or wires in module \multiple_module_opt2..
Removed 0 unused cells and 12 unused wires.
<suppressed ~12 debug messages>
```

```
6.1.2. Re-integrating ABC results.
ABC RESULTS:      const0 cells:      1
ABC RESULTS:      internal signals:   4
ABC RESULTS:      input signals:     4
ABC RESULTS:      output signals:    1
Removing temp directory.

yosys> show

7. Generating Graphviz representation of design.
Writing dot description to '/home/tvsmohan99/.yosys_show.dot'.
Dumping module multiple_module_opt2 to page 1.
Exec: { test -f '/home/tvsmohan99/.yosys_show.dot.pid' && fuser -s '/home/tvsmohan99/.yosys_show.dot.pid' 2> /dev/null; } || ( echo $$ >&3; exec xdot '/home/tvsmohan99/.yosys_show.dot'; ) 3> '/home/tvsmohan99/.yosys_show.dot.pid' &
```



```

yosys> write_verilog -noattr multiple_module_opt2_netlist.v
8. Executing Verilog backend.
Dumping module `\\multiple_module_opt2'.
yosys> !gvim multiple_module_opt2_netlist.v
9. Shell command: gvim multiple_module_opt2_netlist.v

```

```

multiple_module_opt2_netlist.v (~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files) - GVIM5
File Edit Tools Syntax Buffers Window Help
/* Generated by Yosys 0.9+4081 (git sha1 862e84eb, gcc 7.5.0-3ubuntu1-18.04 -fPIC -Os) */

module multiple_module_opt2(a, b, c, d, y);
    wire _0;
    wire _1;
    wire _2;
    wire _3;
    wire _4;
    wire \U1.y ;
    wire \U2.y ;
    wire \U3.y ;
    input a;
    input b;
    input c;
    input d;
    output y;
    assign _4_ = 1'h0;
    assign _0_ = a;
    assign _2_ = c;
    assign _1_ = b;
    assign _3_ = d;
    assign y = _4_;
endmodule

```

#### OBSERVATIONS:

- 1) The synthesizer optimised the multiple\_module\_opt2.v using constant propagation optimization.
- 2) The synthesizer optimized the output to zero.

#### SKY130RTL D3SK3 L1 Lab07 Sequential Logic Optimisations part1:

##### DFF\_CONST1SIMULATION:

Command:iverilog dff\_const1.v tb\_dff\_const1.v

Command:./a.out

Command:gtkwave tb\_dff\_const1.vcd

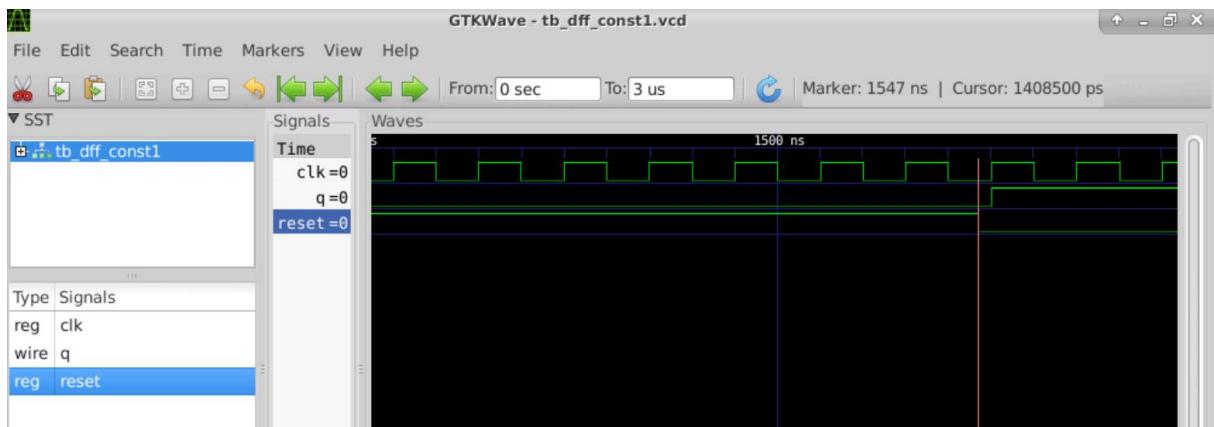
```

tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ iverilog dff_const1.v tb_const1.v
tb_const1.v: No such file or directory
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ iverilog dff_const1.v tb_dff_const1.v
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ ./a.out
VCD info: dumpfile tb_dff_const1.vcd opened for output.
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_dff_const1.vcd

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

[0] start time.
[3000000] end time.

```



**OBSERVATIONS:** The q doesn't change value when reset is low. Q will change value only with the posedge of the clock.

#### DFF\_CONST2SIMULATION:

Command: iverilog dff\_const1.v tb\_dff\_const2.v

Command: ./a.out

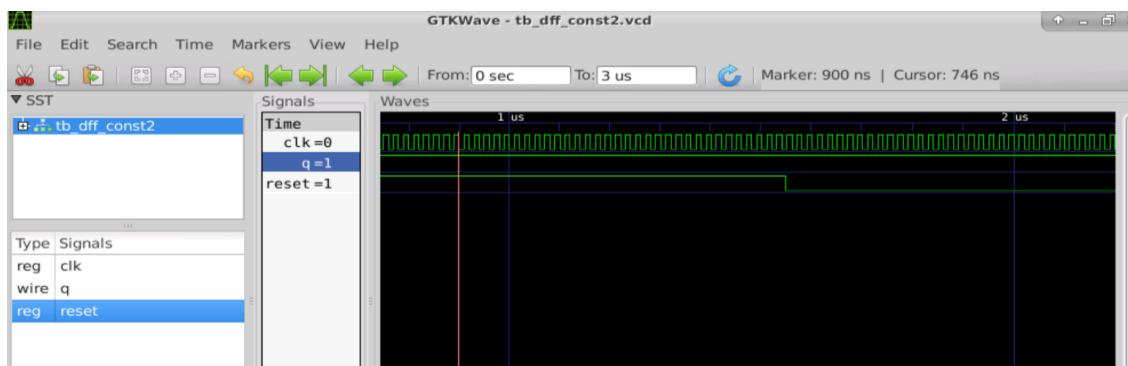
Command: gtkwave tb\_dff\_const2.vcd

```
tvsimohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ iverilog dff_const2.v tb_dff_const2.v
tvsimohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ ./a.out
VCD info: dumpfile tb_dff_const2.vcd opened for output.
tvsimohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_dff_const2.vcd

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

[0] start time.
[3000000] end time.

```



**OBSERVATIONS:** The q value is always one.

## DFF\_CONST1.V SYNTHESIS:

Command:yosys

Command:read\_liberty–lib ..//my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:read\_verilog dff\_const1.v

Command:synth–top dff\_const1

Command:dfflibmap–liberty ..//my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:abc–liberty ..//my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:show

```
yosys> read_liberty -lib ..//my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog dff_const1.v

2. Executing Verilog-2005 frontend: dff_const1.v
Parsing Verilog input from 'dff_const1.v' to AST representation.
Generating RTLIL representation for module '\dff_const1'.
Successfully finished Verilog frontend.
```

### 3.25. Printing statistics.

```
==== dff_const1 ===
```

Number of wires:	3
Number of wire bits:	3
Number of public wires:	3
Number of public wire bits:	3
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	1
\$_DFF_PP0_	1

### 3.26. Executing CHECK pass (checking for obvious problems).

Checking module dff\_const1...

Found and reported 0 problems.

5.1. Executing DFFLEGALIZE pass (convert FFs to types supported by the target).  
Mapping DFF cells in module `\\dff\_const1':

mapped 1 \$\_DFF\_PN0\_ cells to \\sky130\_fd\_sc\_hd\_dfrtp\_1 cells.

```
yosys> █
```

```

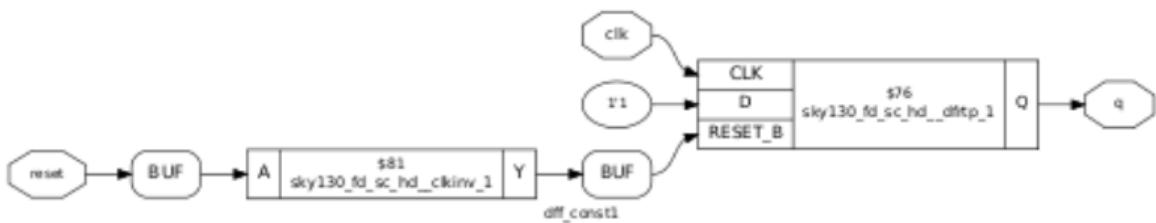
6.1.2. Re-integrating ABC results.
ABC RESULTS: sky130_fd_sc_hd_clkinv_1 cells:      1
ABC RESULTS:   internal signals:      0
ABC RESULTS:   input signals:       1
ABC RESULTS:   output signals:      1
Removing temp directory.

yosys> show

7. Generating Graphviz representation of design.
Writing dot description to '/home/tvsmohan99/.yosys_show.dot'.
Dumping module dff_const1 to page 1.
Exec: { test -f '/home/tvsmohan99/.yosys_show.dot.pid' && fuser -s '/home/tvsmohan99/.yosys_show.dot.pid' 2> /dev/null; } || ( echo $$ >&3; exec xdot '/home/tvsmohan99/.yosys_show.dot'; ) 3> '/home/tvsmohan99/.yosys_show.dot.pid' &

yosys>

```



**OBSERVATIONS:** As stated before, the D\_flip\_flop in the netlist is active\_low. So, the synthesizer used an inverter to change rest from active\_high to active\_low.

### SKY130RTL D3SK3 L2 Lab07 Sequential Logic Optimisations part2:

#### DFF\_CONST2.V SYNTHESIS

Command:yosys

Command:read\_liberty–lib ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:read\_verilog dff\_const2.v

Command:synth–top dff\_const2

Command:dfflibmap–liberty ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:abc–liberty ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:show

```
yosys> read_liberty -lib ../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog dff_const2.v

2. Executing Verilog-2005 frontend: dff_const2.v
Parsing Verilog input from `dff_const2.v' to AST representation.
Generating RTLIL representation for module `\'dff_const2'.
Successfully finished Verilog frontend.

yosys> █
```

### 3.25. Printing statistics.

```
==== dff_const2 ====

Number of wires: 3
Number of wire bits: 3
Number of public wires: 3
Number of public wire bits: 3
Number of memories: 0
Number of memory bits: 0
Number of processes: 0
Number of cells: 0
```

```
3.26. Executing CHECK pass (checking for obvious problems).
Checking module dff_const2...
Found and reported 0 problems.
```

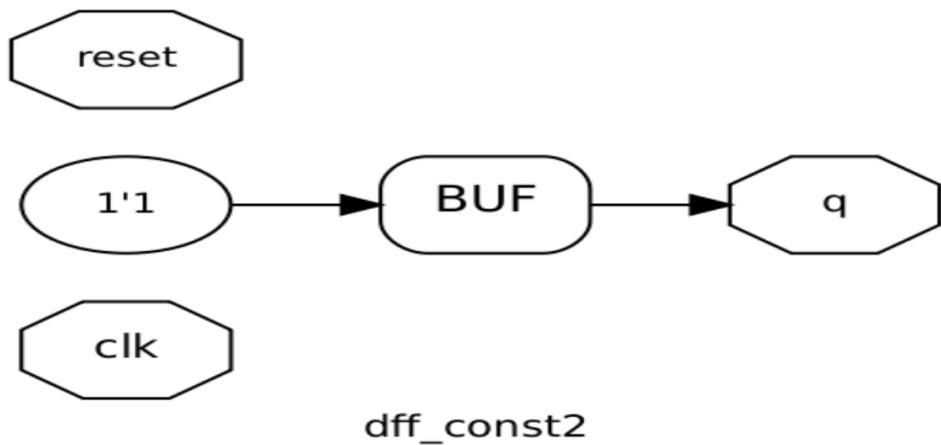
```
yosys> █
```

```
yosys> abc -liberty ../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
5. Executing ABC pass (technology mapping using ABC).

5.1. Extracting gate netlist of module `\'dff_const2' to `<abc-temp-dir>/input.blif'..
Extracted 0 gates and 0 wires to a netlist network with 0 inputs and 0 outputs.
Don't call ABC as there is nothing to map.
Removing temp directory.

yosys> █
```

OBSERVATION: ABC command's result is indicating that there is no standard cell.



### SKY130RTL D3SK3 L2 Lab07 Sequential Logic Optimisations part3:

#### DFF\_CONST3.V SIMULATION, OPTIMIZATION AND SYNTHESIS:

Command: iverilog dff\_const3.v tb\_dff\_const3.v

Command: ./a.out

Command: gtkwave tb\_dff\_const3.vcd

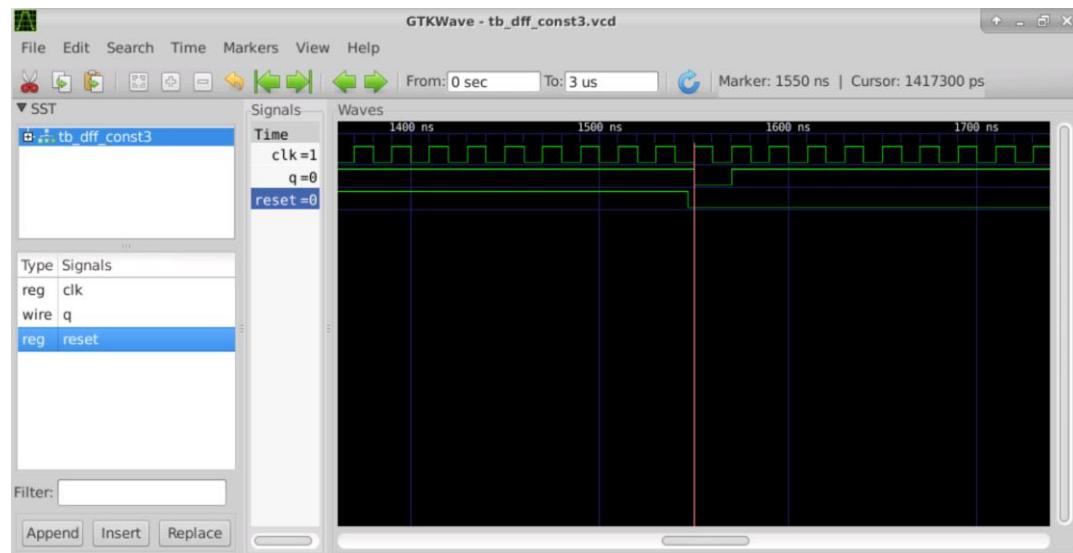
```

tvsmohan99@rtlworkshop-04:/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ iverilog dff_const3.v tb_dff_const3.v
tvsmohan99@rtlworkshop-04:/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ ./a.out
VCD info: dumpfile tb_dff_const3.vcd opened for output.
tvsmohan99@rtlworkshop-04:/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_dff_const3.vcd

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

[0] start time.
[3000000] end time.

```



OBSERVATIONS:  $q=1$  at every instant except at the moment in the above figure. So, the flipflop can't be optimised.

Command:yosys

Command:read\_liberty–lib ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:read\_verilog dff\_const3.v

Command:synth–top dff\_const3

Command:dfflibmap–liberty ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:abc–liberty ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:show

```
yosys> read_liberty -lib ..../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog dff_const3.v

2. Executing Verilog-2005 frontend: dff_const3.v
Parsing Verilog input from `dff_const3.v' to AST representation.
Generating RTLIL representation for module `\\dff_const3'.
Successfully finished Verilog frontend.

yosys>
```

3.25. Printing statistics.

```
== dff_const3 ==
```

Number of wires:	4
Number of wire bits:	4
Number of public wires:	4
Number of public wire bits:	4
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	2
\$_DFF_PP0	1
\$_DFF_PP1	1

3.26. Executing CHECK pass (checking for obvious problems).

Checking module dff\_const3...

Found and reported 0 problems.

```
yosys> █
```

```

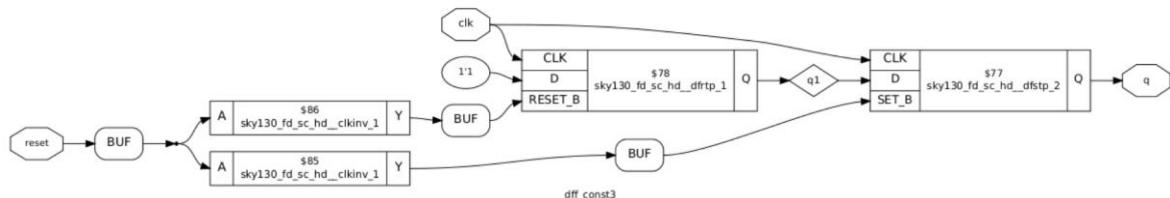
5.1.2. Re-integrating ABC results.
ABC RESULTS: sky130_fd_sc_hd_clkinv_1 cells:      2
ABC RESULTS: internal signals:          0
ABC RESULTS: input signals:           1
ABC RESULTS: output signals:          2
Removing temp directory.

yosys> show

6. Generating Graphviz representation of design.
Writing dot description to '/home/tvsmohan99/.yosys_show.dot'.
Dumping module dff_const3 to page 1.
Exec: { test -f '/home/tvsmohan99/.yosys_show.dot.pid' && fuser -s '/home/tvsmohan99/.yosys_show.dot.pid' 2> /dev/null; } || ( echo $$ >&3; exec xdot '/home/tvsmohan99/.yosys_show.dot'; ) 3> '/home/tvsmohan99/.yosys_show.dot.pid' &

yosys> █

```



OBSERVATIONS: Because q cannot be optimised, the circuit remains unoptimized.

#### DFF\_CONST4.V SIMULATION, OPTIMIZATION AND SYNTHESIS:

Command: iverilog dff\_const4.v tb\_dff\_const4.v

Command: ./a.out

Command: gtkwave tb\_dff\_const4.vcd

```

tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ iverilog dff_const4.v tb_dff_const4.v
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ ./a.out
VCD info: dumpfile tb_dff_const4.vcd opened for output.
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_dff_const4.vcd

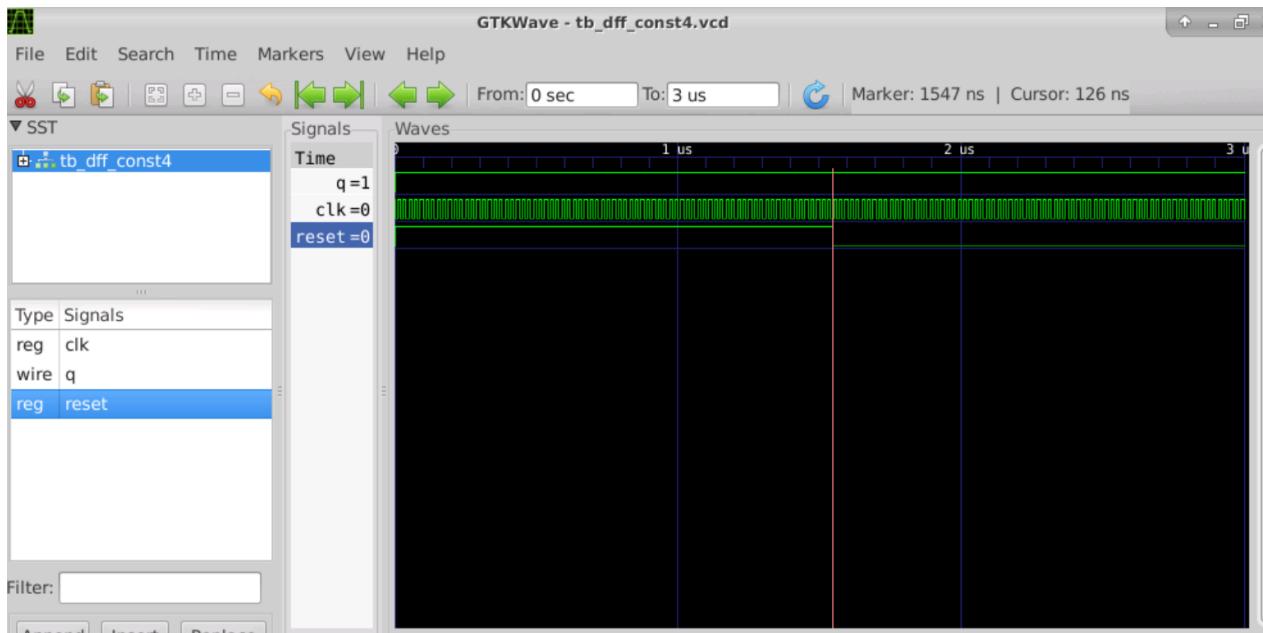
```

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

```

[0] start time.
[30000000] end time.
█

```



OBSERVATIONS: The q is always one.

Command: yosys

Command: read\_liberty -lib ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: read\_verilog dff\_const4.v

Command: synth -top dff\_const4

Command:dfflibmap -liberty ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: abc -liberty ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: show

```
yosys> read_liberty -lib ..../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog dff_const4.v

2. Executing Verilog-2005 frontend: dff_const4.v
Parsing Verilog input from `dff_const4.v' to AST representation.
Generating RTLIL representation for module `\'dff_const4'.
Successfully finished Verilog frontend.
```

```
3.25. Printing statistics.
```

```
==== dff_const4 ===
```

```
Number of wires: 4
Number of wire bits: 4
Number of public wires: 4
Number of public wire bits: 4
Number of memories: 0
Number of memory bits: 0
Number of processes: 0
Number of cells: 0
```

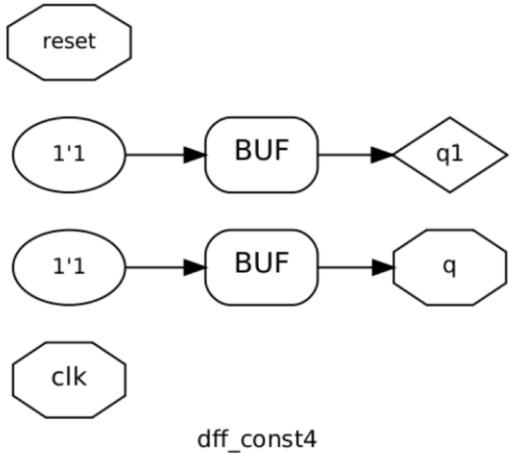
```
3.26. Executing CHECK pass (checking for obvious problems).
```

```
Checking module dff_const4...
Found and reported 0 problems.
```

```
yosys> abc -liberty ../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
```

```
5. Executing ABC pass (technology mapping using ABC).
```

```
5.1. Extracting gate netlist of module `\\dff_const4` to `<abc-temp-dir>/input.blif'..
Extracted 0 gates and 0 wires to a netlist network with 0 inputs and 0 outputs.
Don't call ABC as there is nothing to map.
Removing temp directory.
```



OBSERVATIONS: As the q is always one in this synthesis, there is no standard cell in the netlist.

## DFF\_CONST5.V SIMULATION, OPTIMIZATION AND SYNTHESIS:

Command: iverilog dff\_const5.v tb\_dff\_const5.v

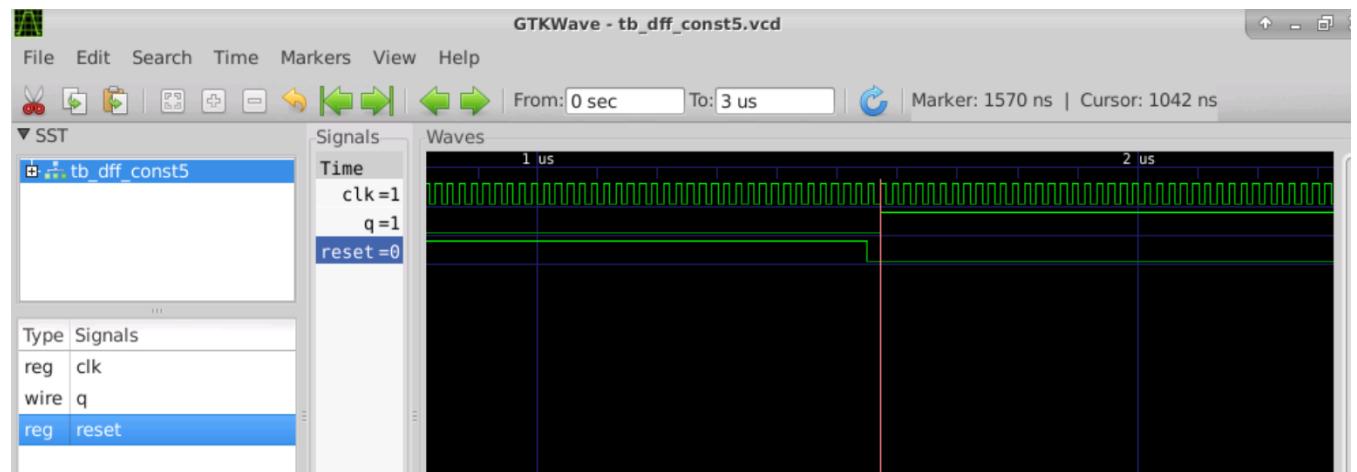
Command: ./a.out

Command: gtkwave tb\_dff\_const5.vcd

```
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ iverilog dff_const5.v tb_dff_const5.v
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ ./a.out
VCD info: dumpfile tb_dff_const5.vcd opened for output.
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_dff_const5.vcd

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

[0] start time.
[3000000] end time.
```



Command: yosys

Command: read\_liberty –lib ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: read\_verilog dff\_const4.v

Command: synth –top dff\_const4

Command: dfflibmap –liberty ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: abc –liberty ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: show

```

yosys> read_liberty -lib ../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog dff_const5.v

2. Executing Verilog-2005 frontend: dff_const5.v
Parsing Verilog input from `dff_const5.v' to AST representation.
Generating RTLIL representation for module `\'dff_const5'.
Successfully finished Verilog frontend.

```

### 3.25. Printing statistics.

```

==== dff_const5 ====

Number of wires: 4
Number of wire bits: 4
Number of public wires: 4
Number of public wire bits: 4
Number of memories: 0
Number of memory bits: 0
Number of processes: 0
Number of cells: 2
    $_DFF_PP0_ 2

```

### 3.26. Executing CHECK pass (checking for obvious problems).

Checking module dff\_const5...

Found and reported 0 problems.

```

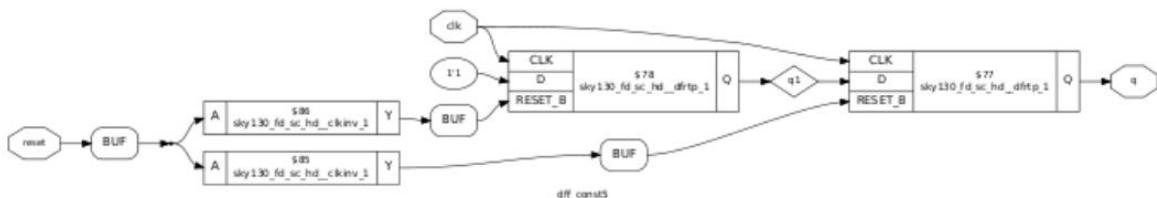
5.1.2. Re-integrating ABC results.
ABC RESULTS: sky130_fd_sc_hd_ckinv_1 cells: 2
ABC RESULTS: internal signals: 0
ABC RESULTS: input signals: 1
ABC RESULTS: output signals: 2
Removing temp directory.

yosys> show

6. Generating Graphviz representation of design.
Writing dot description to '/home/tvsmohan99/.yosys_show.dot'.
Dumping module dff_const5 to page 1.
Exec: { test -f '/home/tvsmohan99/.yosys_show.dot.pid' && fuser -s '/home/tvsmohan99/.yosys_show.dot.pid' 2> /dev/null; } || ( echo $$
>&3; exec xdot '/home/tvsmohan99/.yosys_show.dot'; ) 3> '/home/tvsmohan99/.yosys_show.dot.pid' &

yosys>

```



OBSERVATIONS: There is no optimization in this file synthesis.

## SKY130RTL D3SK4 L1 Seq optimisation unused outputs part1:

### COUNTER\_OPT.V SYNTHESIS:

Command: yosys

Command: read\_liberty -lib ../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: read\_verilog counter\_opt.v

Command: synth -top counter\_opt

Command: dfflibmap -liberty ../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: abc -liberty ../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: show

```
yosys> read_liberty -lib ../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog counter_opt.v

2. Executing Verilog-2005 frontend: counter_opt.v
Parsing Verilog input from `counter_opt.v' to AST representation.
Generating RTLIL representation for module `\counter_opt'.
Successfully finished Verilog frontend.

yosys>
```

### 3.25. Printing statistics.

```
==== counter_opt ===
```

Number of wires:	5
Number of wire bits:	9
Number of public wires:	4
Number of public wire bits:	6
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	2
\$_DFF_PP0_	1
\$_NOT_	1

### 3.26. Executing CHECK pass (checking for obvious problems).

Checking module counter\_opt...

Found and reported 0 problems.

```

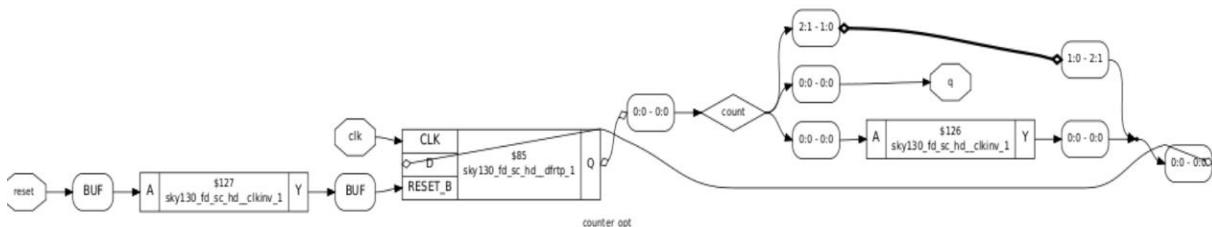
5.1.2. Re-integrating ABC results.
ABC RESULTS: sky130_fd_sc_hd_clkinv_1 cells:      2
ABC RESULTS:   internal signals:     0
ABC RESULTS:   input signals:      2
ABC RESULTS:   output signals:     2
Removing temp directory.

yosys> show

6. Generating Graphviz representation of design.
Writing dot description to '/home/tvsmohan99/.yosys_show.dot'.
Dumping module counter_opt to page 1.
Exec: { test -f '/home/tvsmohan99/.yosys_show.dot.pid' && fuser -s '/home/tvsmohan99/.yosys_show.dot.pid' 2> /dev/null; } || ( echo $$
>&3; exec xdot '/home/tvsmohan99/.yosys_show.dot'; ) 3> '/home/tvsmohan99/.yosys_show.dot.pid' &

yosys>

```



**OBSERVATIONS:** The synthesis shows that q is connected to D through an inverter. So, for every posedge, they q will change. The synthesizer will ignore the counting of the remaining two bits.

### SKY130RTL D3SK4 L2 Seq optimisation unused outputs part2:

#### COUNTER\_OPT2.V SYNTHESIS:

Command:yosys

Command:read\_liberty –lib ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:read\_verilog counter\_opt2.v

Command:synth –top counter\_opt

Command:dfflibmap –liberty ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:abc –liberty ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:show

```

yosys> read_liberty -lib ..../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog counter_opt2.v
2. Executing Verilog-2005 frontend: counter_opt2.v
Parsing Verilog input from 'counter_opt2.v' to AST representation.
Generating RTLIL representation for module '\counter_opt'.
Successfully finished Verilog frontend.

```

```
4.25. Printing statistics.
```

```
==== counter_opt ===
```

```
Number of wires: 8
Number of wire bits: 14
Number of public wires: 4
Number of public wire bits: 6
Number of memories: 0
Number of memory bits: 0
Number of processes: 0
Number of cells: 9
$ _ANDNOT_ 1
$ _AND_ 1
$ _DFP_PP0_ 3
$ _NOT_ 1
$ _OR_ 1
$ _XOR_ 2
```

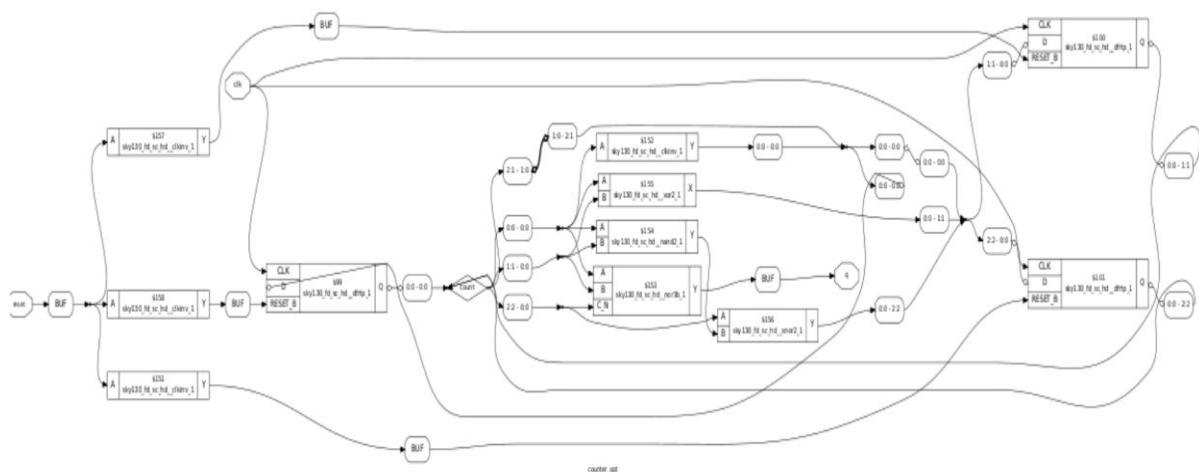
```
4.26. Executing CHECK pass (checking for obvious problems).
```

```
Checking module counter_opt...
Found and reported 0 problems.
```

### 6.1.2. Re-integrating ABC results.

```
ABC RESULTS: sky130_fd_sc_hd_clkinv_1 cells: 4
ABC RESULTS: sky130_fd_sc_hd_nand2_1 cells: 1
ABC RESULTS: sky130_fd_sc_hd_nor3b_1 cells: 1
ABC RESULTS: sky130_fd_sc_hd_xnor2_1 cells: 1
ABC RESULTS: sky130_fd_sc_hd_xor2_1 cells: 1
ABC RESULTS: internal signals: 2
ABC RESULTS: input signals: 4
ABC RESULTS: output signals: 7
```

```
Removing temp directory.
```



## COUNTER\_OPT3.V SYNTHESIS FOR ASSESSMENT:

Command:yosys

Command:read\_liberty –lib ..//my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:read\_verilog counter\_opt3.v

Command:synth –top counter\_opt

Command:dfflibmap –liberty ..//my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:abc –liberty ..//my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:show

```
yosys> read_liberty -lib ..//my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog counter_opt3.v

2. Executing Verilog-2005 frontend: counter_opt3.v
Parsing Verilog input from `counter_opt3.v' to AST representation.
Generating RTLIL representation for module `\counter_opt'.
Successfully finished Verilog frontend.
```

3.25. Printing statistics.

==== counter\_opt ===

Number of wires:	8
Number of wire bits:	14
Number of public wires:	4
Number of public wire bits:	6
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	9
\$_ANDNOT_	1
\$_AND_	1
\$_DFF_PP0_	3
\$_NOT_	1
\$_ORN0T_	1
\$_XOR_	2

3.26. Executing CHECK pass (checking for obvious problems).

Checking module counter\_opt...

Found and reported 0 problems.

```

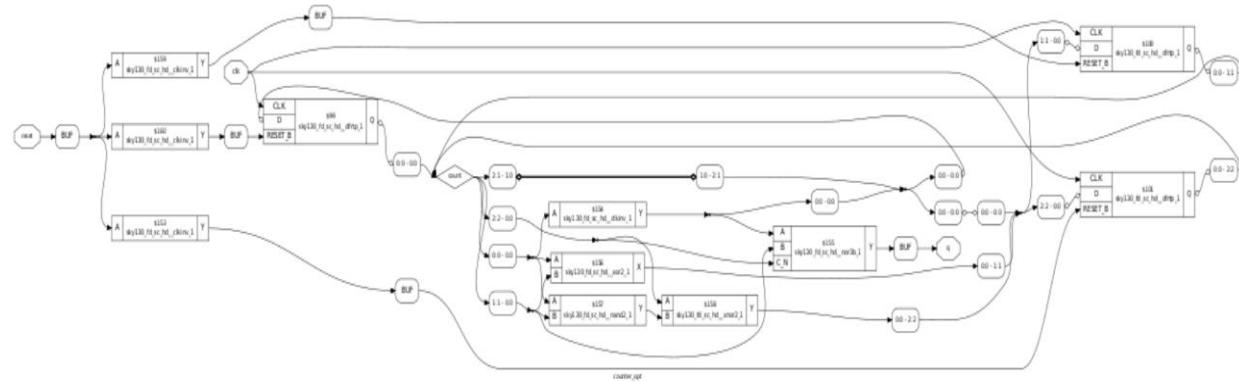
5.1.2. Re-integrating ABC results.
ABC RESULTS: sky130_fd_sc_hd_ckinv_1 cells:      4
ABC RESULTS: sky130_fd_sc_hd_nand2_1 cells:      1
ABC RESULTS: sky130_fd_sc_hd_nor3b_1 cells:      1
ABC RESULTS: sky130_fd_sc_hd_xnor2_1 cells:      1
ABC RESULTS: sky130_fd_sc_hd_xor2_1 cells:      1
ABC RESULTS: internal signals:      2
ABC RESULTS: input signals:      4
ABC RESULTS: output signals:      7
Removing temp directory.

yosys> show

6. Generating Graphviz representation of design.
Writing dot description to '/home/tvsmohan99/.yosys_show.dot'.
Dumping module counter_opt to page 1.
Exec: { test -f '/home/tvsmohan99/.yosys_show.dot.pid' && fuser -s '/home/tvsmohan99/.yosys_show.dot.pid' 2> /dev/null; } || ( echo $$ >&3; exec xdot '/home/tvsmohan99/.yosys_show.dot'; ) 3> '/home/tvsmohan99/.yosys_show.dot.pid' &

yosys>

```



## **DAY 4:**

### **SKY130RTL D4SK2 L1 Lab GLS Synth Sim Mismatch part1:**

#### **TERENARY\_OPERATOR GLS :**

Command: iverilog ternary\_operator\_mux.v tb\_ternary\_operator\_mux.v

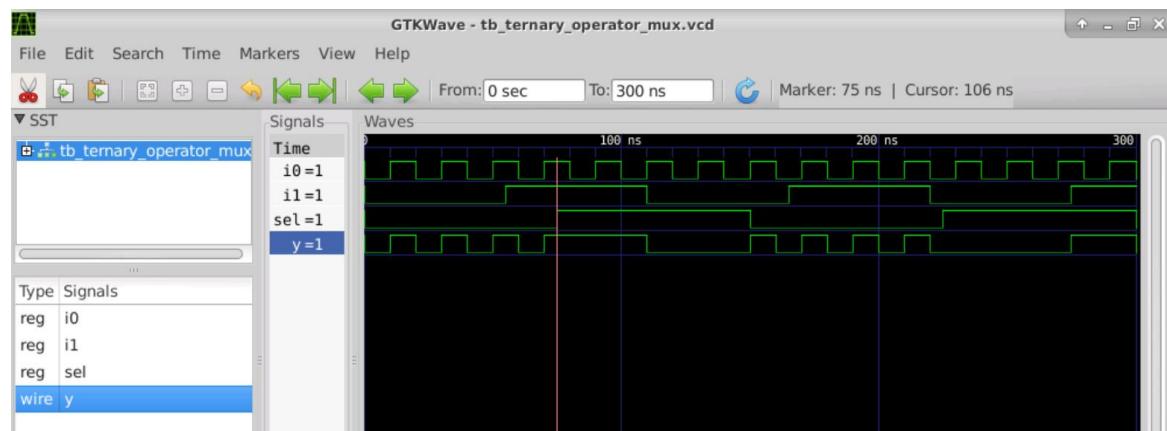
Command: ./a.out

Command: gtkwave tb\_ternary\_operator\_mux.vcd

```
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDdesignAndSynthesisWorkshop/verilog_files$ iverilog ternary_operator_mux.v tb_ternary_operator_mux.v
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDdesignAndSynthesisWorkshop/verilog_files$ ./a.out
VCD info: dumpfile tb_ternary_operator_mux.vcd opened for output.
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDdesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_ternary_operator_mux.vcd

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

[0] start time.
[300000] end time.
```



Command: yosys

Command: read\_liberty –lib ..//my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: read\_verilog ternary\_operator\_mux.v

Command: synth-top ternary\_operator\_mux

Command: abc-liberty ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: show

Command: write\_verilog -noattr ternary\_operator\_mux\_net.v

```
yosys> read_liberty -lib ..../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog ternary_operator_mux.v
ERROR: Can't open input file `ternary_operator_mux.v' for reading: No such file or directory

yosys> read_verilog ternary_operator_mux.v

2. Executing Verilog-2005 frontend: ternary_operator_mux.v
Parsing Verilog input from `ternary_operator_mux.v' to AST representation.
Generating RTLIL representation for module `\ternary_operator_mux'.
Successfully finished Verilog frontend.
```

### 3.25. Printing statistics.

```
==== ternary_operator_mux ===
```

Number of wires:	4
Number of wire bits:	4
Number of public wires:	4
Number of public wire bits:	4
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	1
\$_MUX_	1

### 3.26. Executing CHECK pass (checking for obvious problems).

```
Checking module ternary_operator_mux...
```

```
Found and reported 0 problems.
```

```

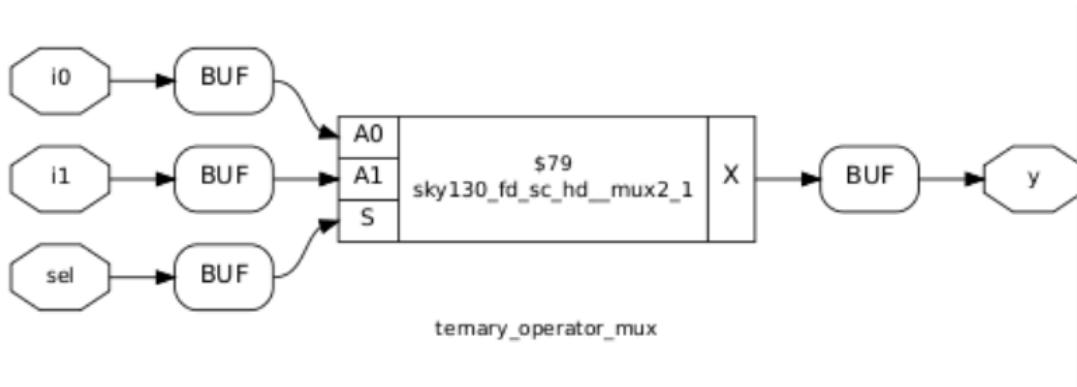
4.1.2. Re-integrating ABC results.
ABC RESULTS: sky130_fd_sc_hd_mux2_1 cells:      1
ABC RESULTS:      internal signals:      0
ABC RESULTS:      input signals:      3
ABC RESULTS:      output signals:      1
Removing temp directory.

yosys> show

5. Generating Graphviz representation of design.
Writing dot description to '/home/tvsmohan99/.yosys_show.dot'.
Dumping module ternary operator mux to page 1.
Exec: { test -f '/home/tvsmohan99/.yosys_show.dot.pid' && fuser -s '/home/tvsmohan99/.yosys_show.dot.pid' 2> /dev/null; } || ( echo $$ >&3; exec xdot '/home/tvsmohan99/.yosys_show.dot'; ) 3> '/home/tvsmohan99/.yosys_show.dot.pid' &

yosys>

```



Command: iverilog ..../my\_lib/verilog\_model/primitives.v ..../my\_lib/verilog\_model/sky130\_fd\_sc\_hd.v  
ternary\_operator\_mux\_net.v tb\_ternary\_operator\_mux.v

Command: ./a.out

Command: gtkwave tb\_ternary\_operator\_mux.v

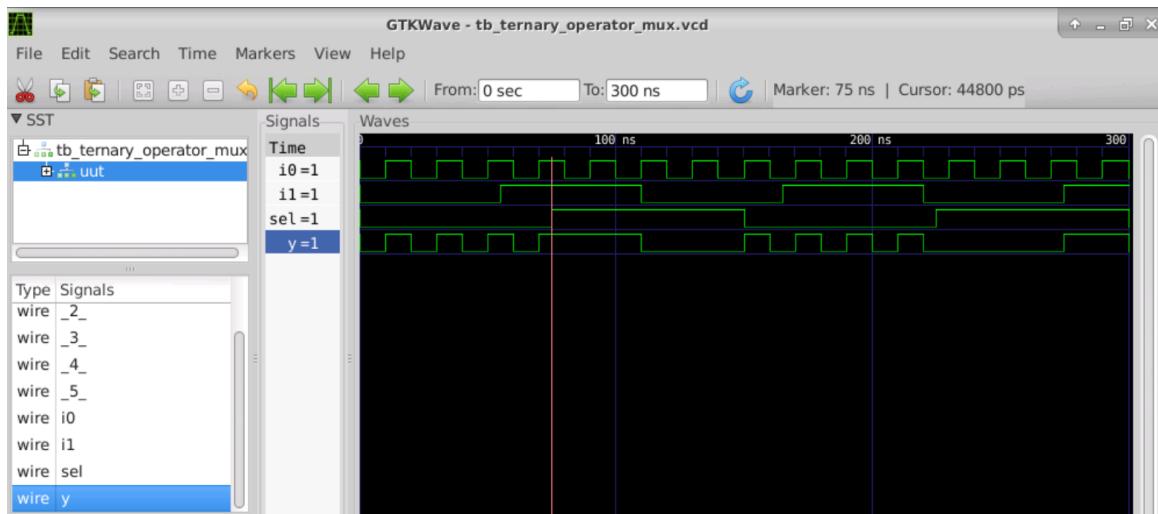
```

tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ iverilog ..../my_lib/verilog_model/primitives.v ..../my_lib/verilog_model/sky130_fd_sc_hd.v ternary_operator_mux_net.v tb_ternary_operator_mux.v
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ ./a.out
VCD info: dumpfile tb_ternary_operator_mux.vcd opened for output.
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_ternary_operator_mux.vcd

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

[0] start time.
[300000] end time.

```



## SKY130RTL D4SK2 L2 Lab GLS Sim Mismatch part2:

### BAD\_MUX.V SIMULATION:

Command: iverilog bad\_mux.v tb\_bad\_mux.v

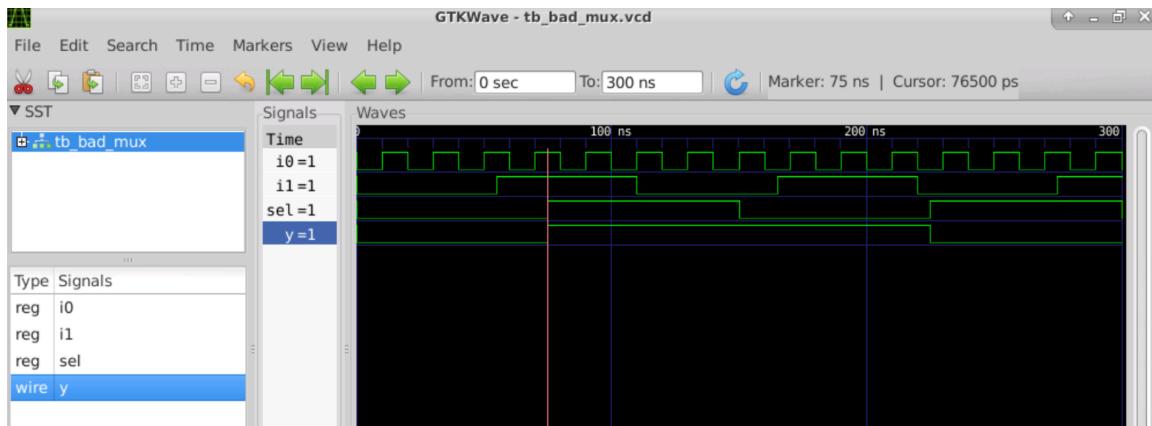
Command: ./a.out

Command: gtkwave tb\_bad\_mux.vcd

```
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDdesignAndSynthesisWorkshop/verilog_files$ iverilog bad_mux.v tb_bad_mux.v
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDdesignAndSynthesisWorkshop/verilog_files$ ./a.out
VCD info: dumpfile tb_bad_mux.vcd opened for output.
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDdesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_bad_mux.vcd
```

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

```
[0] start time.
[3000000] end time.
```



OBSERVATIONS: Clearly, the waveform shows that the verilog file is not working as a multiplexer.

#### BAD\_MUX.V Synthesis:

Command: yosys

Command: read\_liberty -lib ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: read\_verilog bad\_mux.v

Command: synth -top bad\_mux

Command: abc -liberty ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: write\_verilog -noattr bad\_mux\_net.v

```
yosys> read_liberty -lib ..../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog bad_mux.v

2. Executing Verilog-2005 frontend: bad_mux.v
Parsing Verilog input from `bad_mux.v' to AST representation.
Generating RTLIL representation for module `bad_mux'.
Note: Assuming pure combinatorial block at bad_mux.v:4.1-10.4 in
compliance with IEC 62142(E):2005 / IEEE Std. 1364.1(E):2002. Recommending
use of @* instead of @(...) for better match of synthesis and simulation.
Successfully finished Verilog frontend.
```

### 3.25. Printing statistics.

```
==== bad_mux ====  
  
Number of wires: 4  
Number of wire bits: 4  
Number of public wires: 4  
Number of public wire bits: 4  
Number of memories: 0  
Number of memory bits: 0  
Number of processes: 0  
Number of cells: 1  
$_MUX_ 1
```

### 3.26. Executing CHECK pass (checking for obvious problems).

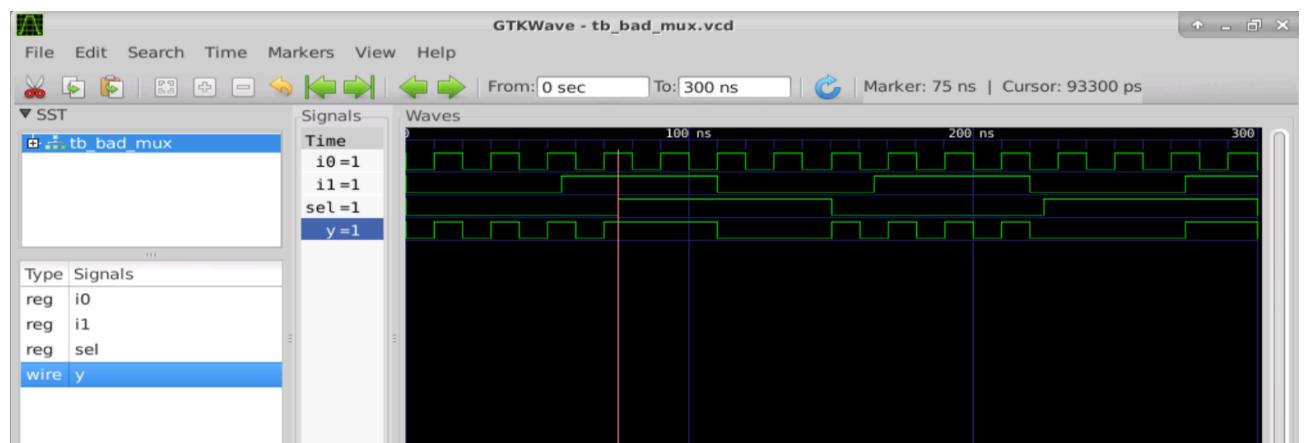
```
Checking module bad_mux...  
Found and reported 0 problems.
```

Command:iverilog ..../my\_lib/verilog\_module/primitives.v  
..../my\_lib/verilog\_model/sky130\_fd\_sc\_hd.v bad\_mux.net.v tb\_bad\_mux.v

Command:./a.out

Command:gtkwave tb\_bad\_mux.vcd

```
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ iverilog ..../my_lib/verilog_model/primitives.v ..../my_lib/verilog model/sky130_fd_sc_hd.v bad_mux.net.v tb_bad_mux.v  
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ ./a.out  
VCD info: dumpfile tb_bad_mux.vcd opened for output.  
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_bad_mux.vcd  
  
GTKWave Analyzer v3.3.86 (w)1999-2017 BSI  
  
[0] start time.  
[300000] end time.
```



OBSERVATION: THE ABOVE WAVEFORM CLEARLY INDICATES SIMULATION – SYNTHESIS MISMATCH CAUSED BY MISSING SENSITIVITY LIST.

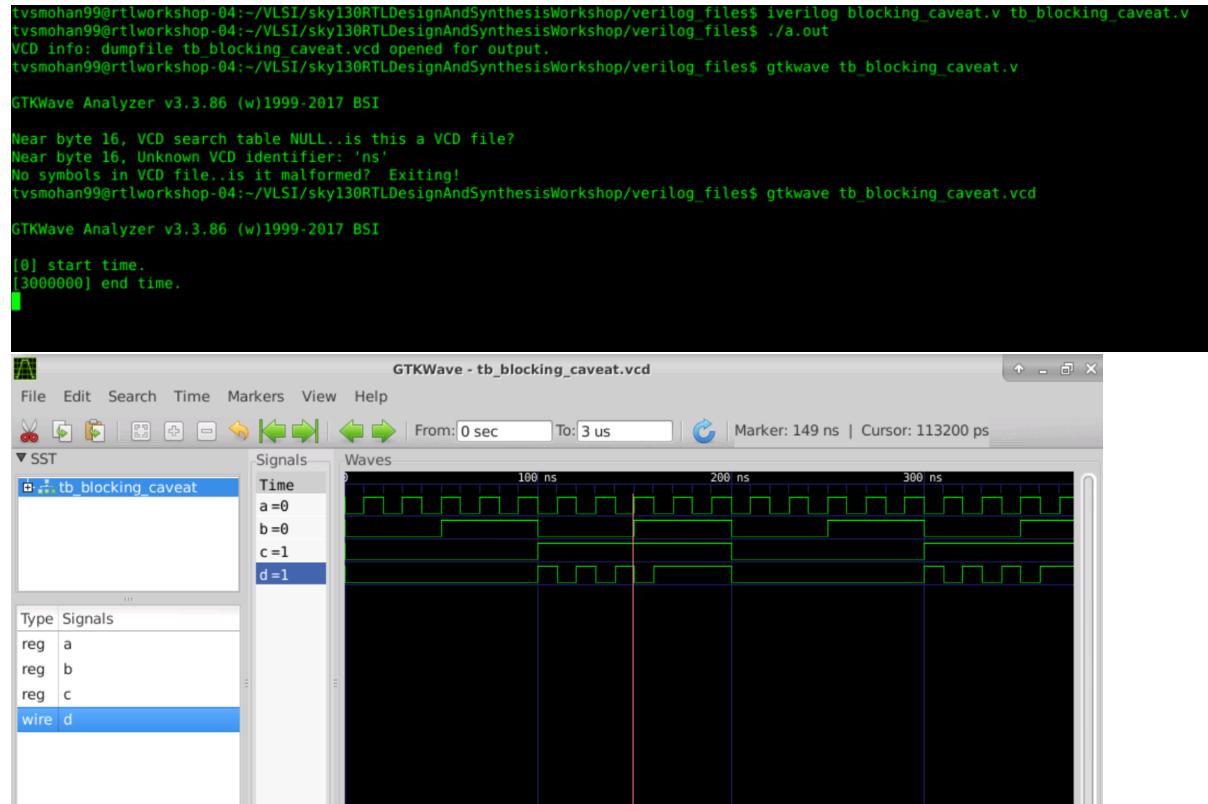
## SKY130RTL D4SK3 L1 Lab Synth sim mismatch blocking statement part1:

### BLOCKING CAVEAT.V SIMULATION

Command: iverilog blocking\_caveat.v tb\_blocking\_caveat.v

Command: ./a.out

Command: gtkwave tb\_blocking\_caveat.vcd



OBSERVATIONS: Clearly, the output of the aORb is taking previous values to generate d.

## SKY130RTL D4SK3 L1 Lab Synth sim mismatch blocking statement part2:

### BLOCKING CAVEAT.V SYNTHESIS:

Command: yosys

Command: read\_liberty -lib ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: read\_verilog blocking\_caveat.v

Command: synth -top blocking\_caveat

Command: abc -liberty ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: write\_verilog -noattr bad\_mux\_net.v

```
yosys> read_liberty -lib ../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog blocking_caveat.v

2. Executing Verilog-2005 frontend: blocking_caveat.v
Parsing Verilog input from `blocking_caveat.v' to AST representation.
Generating RTLIL representation for module `\blocking_caveat'.
Successfully finished Verilog frontend.

yosys>
```

### 3.25. Printing statistics.

```
== blocking_caveat ==
```

Number of wires:	5
Number of wire bits:	5
Number of public wires:	4
Number of public wire bits:	4
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	2
\$ ANDNOT	1
\$ NOR	1

### 3.26. Executing CHECK pass (checking for obvious problems).

```
Checking module blocking_caveat...
Found and reported 0 problems.
```

### 4.1.2. Re-integrating ABC results.

ABC RESULTS: sky130_fd_sc_hd_o21a_1 cells:	1
ABC RESULTS: internal signals:	1
ABC RESULTS: input signals:	3
ABC RESULTS: output signals:	1

```
Removing temp directory.
```

```
yosys> write_verilog -noattr blocking_caveat_net.v
```

### 5. Executing Verilog backend.

```
Dumping module `\blocking_caveat'.
```

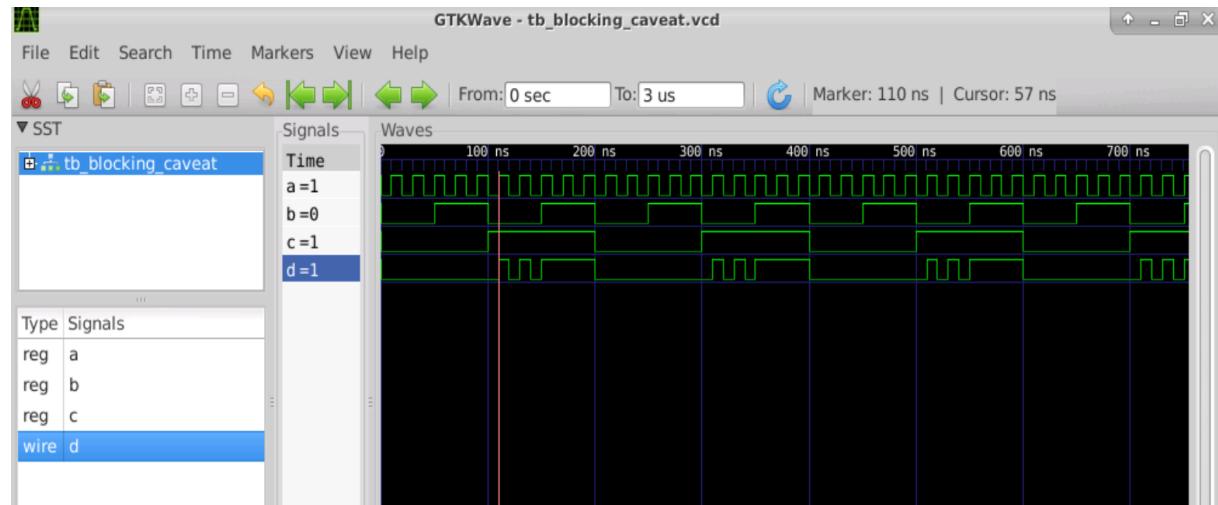
```
yosys> █
```

```
Command: iverilog ..//my_lib/verilog_module/primitives.v  
..//my_lib/verilog_model/sky130_fd_sc_hd.v blocking_caveat_net.v tb_blocking_caveat.v
```

Command: ./a.out

Command: gtkwave tb\_blocking\_caveat.vcd

```
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ iverilog ..//my_lib/verilog_module/primitives.v ..//my_lib/verilog_model/sky130_fd_sc_hd.v blocking_caveat_net.v tb_blocking_caveat.v  
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ ./a.out  
VCD info: dumpfile tb_blocking_caveat.vcd opened for output.  
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_blocking_caveat.vcd  
  
GTKWave Analyzer v3.3.86 (w)1999-2017 BSI  
  
[0] start time.  
[3000000] end time.
```



OBSERVATION: Clearly, the waveform after synthesis indicates the simulation-synthesis mismatch caused by blocking-caveat.

**DAY 5:**

**SKY130RTL D5SK2 L1 Lab Incomplete IF part1:**

**INCOMPLETE\_IF SIMULATION:**

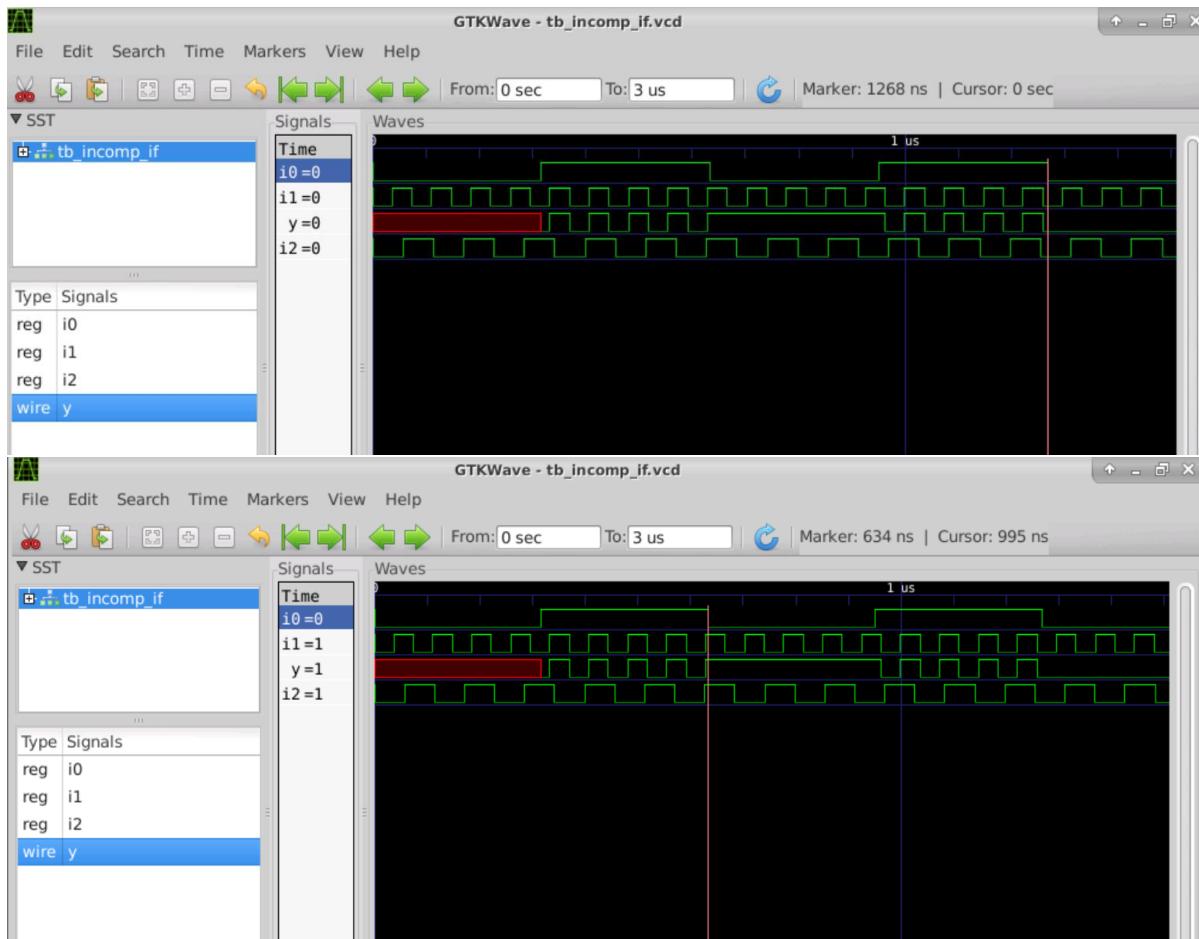
Command: iverilog incomp\_if.v tb\_incomp\_if.v

Command: ./a.out

Command: gtkwave tb\_incomp\_if.vcd

```
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLD...$ iverilog incomp_if.v tb_incomp_if.v
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLD...$ ./a.out
VCD info: dumpfile tb_incomp_if.vcd opened for output.
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLD...$ gtkwave tb_incomp_if.vcd

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI
[0] start time.
[3000000] end time.
```



#### OBSERVATIONS:

- 1) Whenever i0 is high, the y is following i1.
- 2) Whenever i0 is low, the output y is latching to its previous value as shown in above two images.
- 3) This behaviour is due to inferred latch.

#### INCOMP\_IF SYNTHESIS:

Command: yosys

Command: read\_liberty –lib ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: read\_verilog incomp\_if.v

Command: synth –top incomp\_if

Command: abc –liberty ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: show

```
yosys> read_liberty -lib ../my_lib/lib/skyl30_fd_sc_hd_tt_025C_1v80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog incomp_if.v

2. Executing Verilog-2005 frontend: incomp_if.v
Parsing Verilog input from `incomp_if.v' to AST representation.
Generating RTLIL representation for module `incomp_if'.
Successfully finished Verilog frontend.

yosys> █
```

### 3.25. Printing statistics.

```
== incomp_if ==
Number of wires: 4
Number of wire bits: 4
Number of public wires: 4
Number of public wire bits: 4
Number of memories: 0
Number of memory bits: 0
Number of processes: 0
Number of cells: 1
$_DLATCH_P_ 1
```

### 3.26. Executing CHECK pass (checking for obvious problems).

```
Checking module incomp_if...
Found and reported 0 problems.
```

```

yosys> abc -liberty ../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
4. Executing ABC pass (technology mapping using ABC).

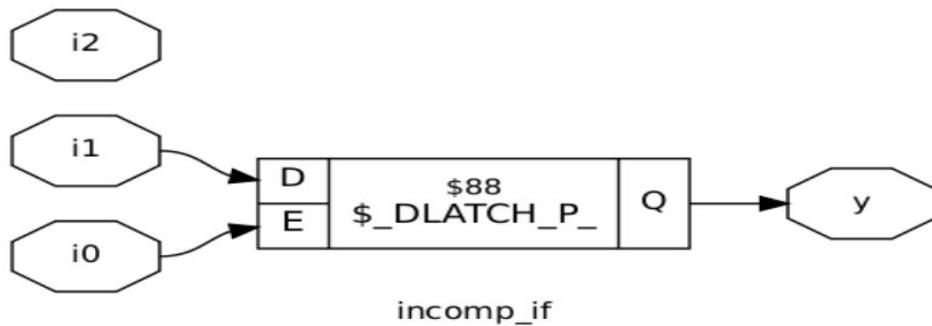
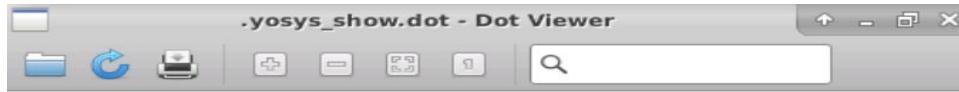
4.1. Extracting gate netlist of module `incomp_if` to `<abc-temp-dir>/input.blif`...
Extracted 0 gates and 0 wires to a netlist network with 0 inputs and 0 outputs.
Don't call ABC as there is nothing to map.
Removing temp directory.

yosys> show

5. Generating Graphviz representation of design.
Writing dot description to '/home/tvsmohan99/.yosys_show.dot'.
Dumping module incomp_if to page 1.
Exec: { test -f '/home/tvsmohan99/.yosys_show.dot.pid' && fuser -s '/home/tvsmohan99/.yosys_show.dot.pid' 2> /dev/null; } || ( echo $$
>>63; exec xdot '/home/tvsmohan99/.yosys_show.dot'; ) 3> '/home/tvsmohan99/.yosys_show.dot.pid' &

yosys>

```



**OBSERVATIONS:** Both the simulation and synthesis are inferring a latch, when we designed a multiplexer.

### SKY130RTL D5SK2 L2 Lab Incomplete IF part2:

#### INCOMPLETE\_IF SIMULATION2:

Command: iverilog incomp\_if2.v tb\_incomp\_if2.v

Command: ./a.out

Command: gtkwave tb\_incomp\_if2.vcd

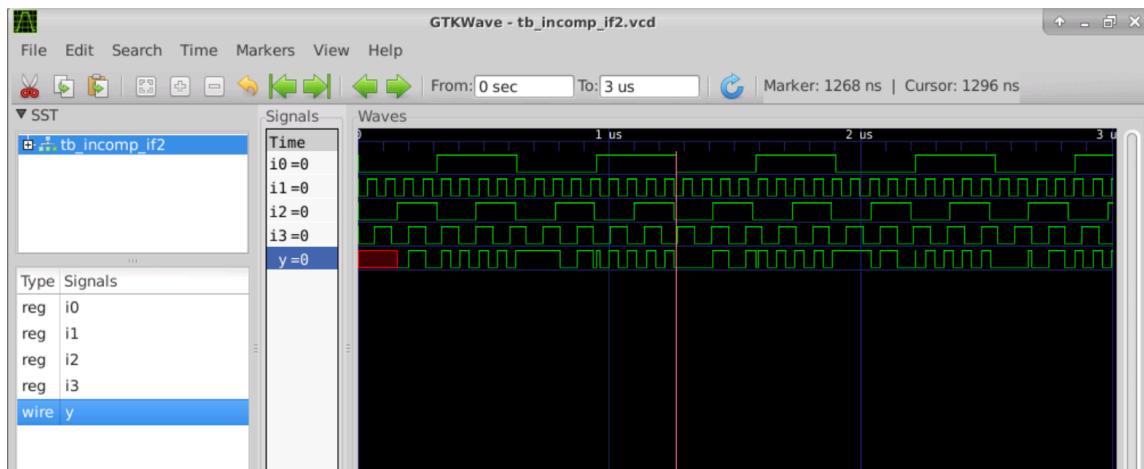
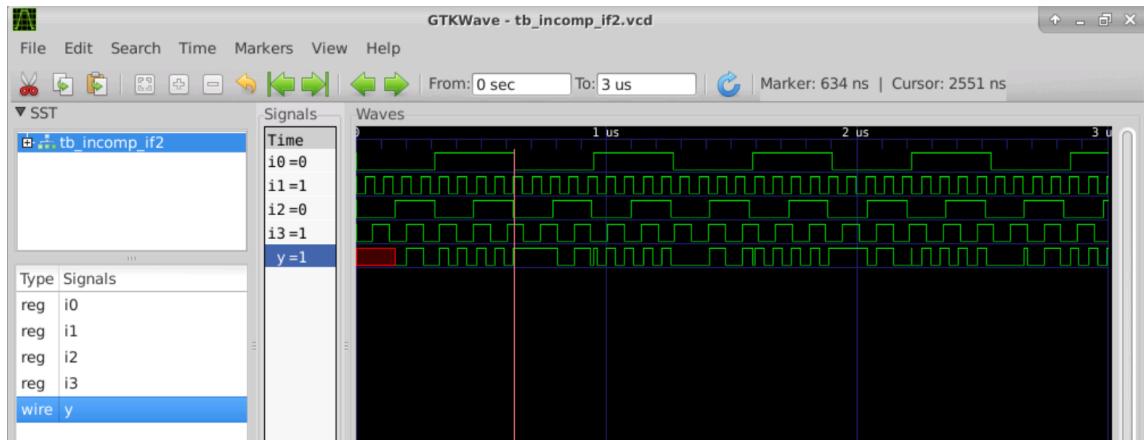
```

tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ iverilog incomp_if2.v tb_incomp_if2.v
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ ./a.out
VCD info: dumpfile tb_incomp_if2.vcd opened for output.
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_incomp_if2.vcd

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

[0] start time.
[3000000] end time.

```



## OBSERVATIONS:

- 1) When  $i_0$  is high, the output is following  $i_1$ .
- 2) When  $i_2$  is high and  $i_1$  is low, the output is following  $i_2$ .
- 3) When both  $i_2$  and  $i_0$  are low, the output is latching to  $y$ .

## INCOMP\_IF2 SYNTHESIS:

Command: yosys

Command: read\_liberty -lib ./my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: read\_verilog incomp\_if2.v

Command: synth -top incomp\_if2

Command: abc -liberty ./my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: show

```
yosys> read_liberty -lib ./my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog incomp_if2.v

2. Executing Verilog-2005 frontend: incomp_if2.v
Parsing Verilog input from `incomp_if2.v' to AST representation.
Generating RTLIL representation for module `incomp_if2'.
Successfully finished Verilog frontend.
```

3.25. Printing statistics.

==== incomp\_if2 ===

Number of wires:	7
Number of wire bits:	7
Number of public wires:	5
Number of public wire bits:	5
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	3
\$_DLATCH_N_	1
\$_MUX_	1
\$_NOR_	1

3.26. Executing CHECK pass (checking for obvious problems).

Checking module incomp\_if2...

Found and reported 0 problems.

yosys>

```

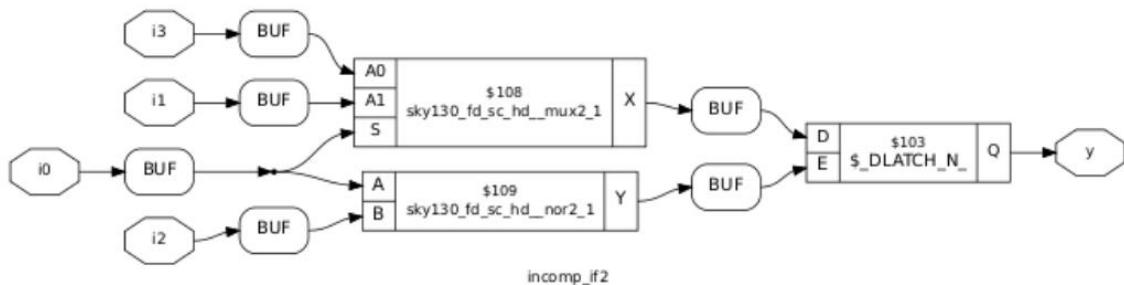
4.1.2. Re-integrating ABC results.
ABC RESULTS: sky130_fd_sc_hd_mux2_1 cells:      1
ABC RESULTS: sky130_fd_sc_hd_nor2_1 cells:      1
ABC RESULTS: internal signals:          0
ABC RESULTS: input signals:           4
ABC RESULTS: output signals:          2
Removing temp directory.

yosys> show

5. Generating Graphviz representation of design.
Writing dot description to '/home/tvsmohan99/.yosys_show.dot'.
Dumping module incom_if2 to page 1.
Exec: { test -f '/home/tvsmohan99/.yosys_show.dot.pid' && fuser -s '/home/tvsmohan99/.yosys_show.dot.pid' 2> /dev/null; } || ( echo $$
>&3; exec xdot '/home/tvsmohan99/.yosys_show.dot'; ) 3> '/home/tvsmohan99/.yosys_show.dot.pid' &

yosys> █

```



## OBSERVATIONS:

- 1) Clearly, we can see the inferred latch on dot viewer.
- 2) The D\_latch has active high enable. So, when `i0` and `i1` are low, the output of the nor gate becomes high. Then the latch becomes active.

## **SKY130RTL D5SK3 L1 Lab incomplete overlapping Case part1:**

### **INCOMP\_CASE SIMUALTION:**

Command: iverilog incom\_case.v tb\_incomp\_case.v

Command: ./a.out

Command: gtkwave tb\_incomp\_case.vcd

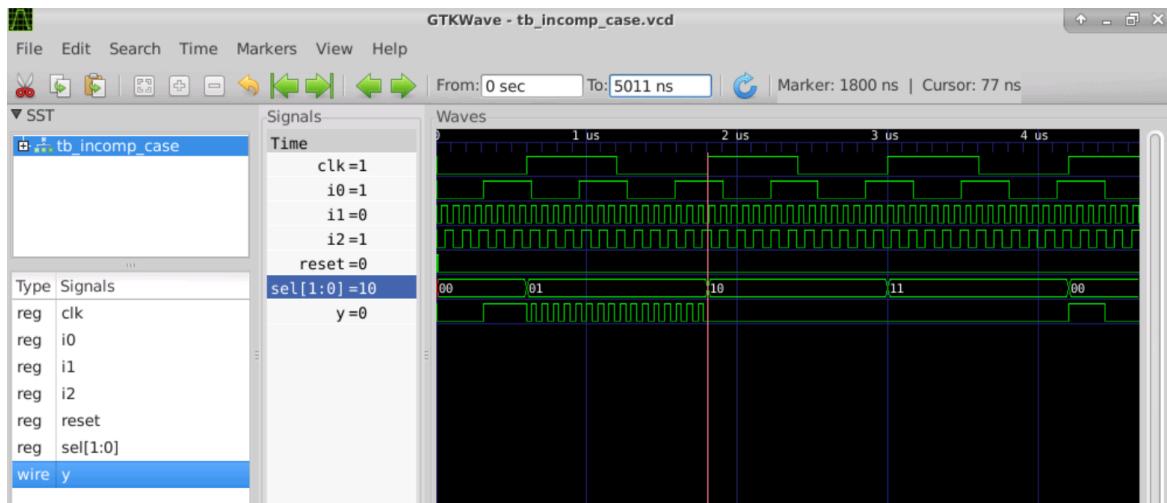
```

tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ iverilog incom_case.v tb_incomp_case.v
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ ./a.out
VCD info: dumpfile tb_incomp_case.vcd opened for output.
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_incomp_case.vcd

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

[0] start time.
[5011000] end time.

```



## OBSERVATIONS:

- 1) The output  $y$  is  $i_0$  when select is 00. When the select is 01, the output  $y$  is  $i_1$ .
- 2) When the select bit [1] is high, the circuit latches to  $y$  value. In this  $y$  is zero.

## INCOMP\_CASE SYNTHESIS:

Command: yosys

Command: read\_liberty -lib ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: read\_verilog incomp\_case.v

Command: synth -top incomp\_case

Command: abc -liberty ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: show

```
yosys> read_liberty -lib ..../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog incomp_case.v
2. Executing Verilog-2005 frontend: incomp_case.v
Parsing Verilog input from 'incomp_case.v' to AST representation.
Generating RTLIL representation for module '\incomp_case'.
Successfully finished Verilog frontend.
```

```
==== incomp_case ===
```

```
Number of wires: 9
Number of wire bits: 10
Number of public wires: 5
Number of public wire bits: 6
Number of memories: 0
Number of memory bits: 0
Number of processes: 0
Number of cells: 5
$_ANDNOT_ 1
$_DLATCH_N_ 1
$_MUX_ 1
$_NOR_ 1
$_ORNOT_ 1
```

3.26. Executing CHECK pass (checking for obvious problems).

Checking module incomp\_case...

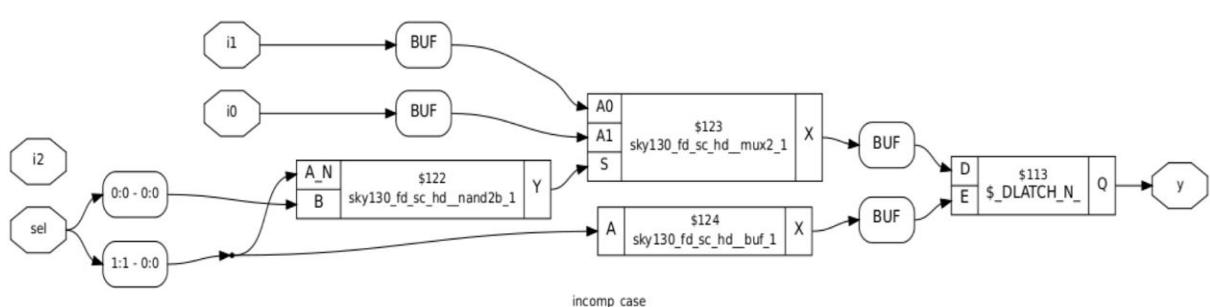
Found and reported 0 problems.

```
yosys>
```

```
4.1.2. Re-integrating ABC results.
ABC RESULTS: sky130_fd_sc_hd_buf_1 cells: 1
ABC RESULTS: sky130_fd_sc_hd_mux2_1 cells: 1
ABC RESULTS: sky130_fd_sc_hd_nand2b_1 cells: 1
ABC RESULTS: internal signals: 2
ABC RESULTS: input signals: 4
ABC RESULTS: output signals: 2
Removing temp directory.
```

```
yosys> show
```

```
5. Generating Graphviz representation of design.
Writing dot description to '/home/tvsmohan99/.yosys_show.dot'.
Dumping module incomp_case to page 1.
Exec: { test -f '/home/tvsmohan99/.yosys_show.dot.pid' && fuser -s '/home/tvsmohan99/.yosys_show.dot.pid' 2> /dev/null; } || ( echo $$
>&3; exec xdot '/home/tvsmohan99/.yosys_show.dot'; ) 3> '/home/tvsmohan99/.yosys_show.dot.pid' &
yosys>
```



#### OBSERVATIONS:

- 1)The dot viewer is showing the latch with enable tied to select [1].
- 2)The remaining part is the mux logic.

## SKY130RTL D5SK3 L2 Lab incomplete overlapping Case part2:

### COMP\_CASE SIMUALTION:

Command: iverilog comp\_case.v tb\_comp\_case.v

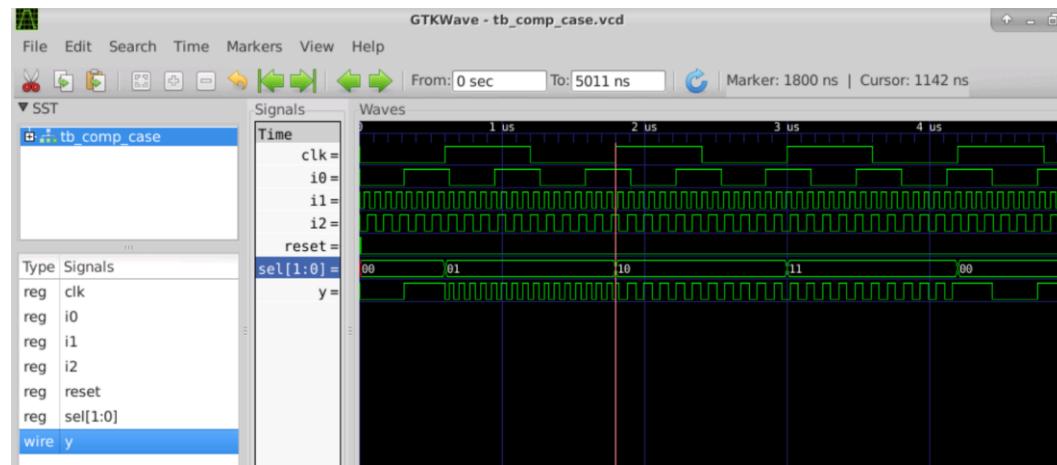
Command: ./a.out

Command: gtkwave tb\_comp\_case.vcd

```
tvsman99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ iverilog comp_case.v tb_comp_case.v
tvsman99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ ./a.out
VCD info: dumpfile tb_comp_case.vcd opened for output.
tvsman99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_comp_case.vcd

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

[0] start time.
[5011000] end time.
```



OBSERVATIONS: The output y is following y when select is 10 or 11.

### COMP\_CASE SYNTHESIS:

Command: yosys

Command: read\_liberty -lib ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: read\_verilog comp\_case.v

Command: synth -top comp\_case

Command: abc -liberty ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: show

```

yosys> read_liberty -lib ../my_lib/lib/sky130_fd_sc_hd_tt_025C_lv80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog comp_case.v

2. Executing Verilog-2005 frontend: comp_case.v
Parsing Verilog input from `comp_case.v' to AST representation.
Generating RTLIL representation for module `\comp_case'.
Successfully finished Verilog frontend.

yosys> 

```

### 3.25. Printing statistics.

```

==== comp_case ====

  Number of wires:          11
  Number of wire bits:      12
  Number of public wires:   5
  Number of public wire bits: 6
  Number of memories:      0
  Number of memory bits:    0
  Number of processes:      0
  Number of cells:          7
    $_ANDNOT_              2
    $_AND_                  1
    $_MUX_                  1
    $_ORNOT_                1
    $_OR_                   2

```

### 3.26. Executing CHECK pass (checking for obvious problems).

Checking module comp\_case...

Found and reported 0 problems.

```

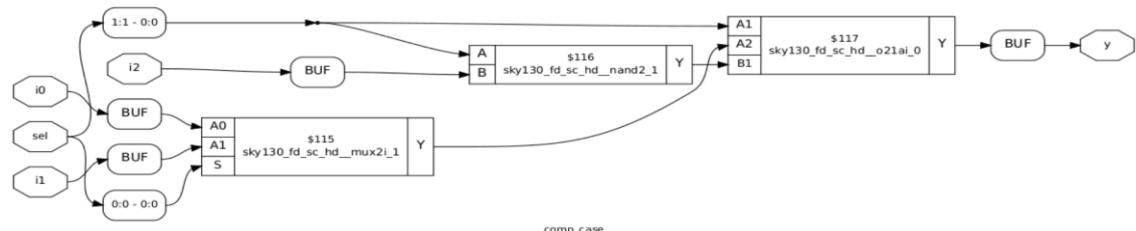
4.1.2. Re-integrating ABC results.
ABC RESULTS: sky130_fd_sc_hd_mux2i_1 cells:      1
ABC RESULTS: sky130_fd_sc_hd_nand2_1 cells:      1
ABC RESULTS: sky130_fd_sc_hd_o2lai_0 cells:      1
ABC RESULTS:     internal signals:        6
ABC RESULTS:     input signals:         5
ABC RESULTS:     output signals:        1
Removing temp directory.

yosys> show

5. Generating Graphviz representation of design.
Writing dot description to '/home/tvsmohan99/.yosys_show.dot'.
Dumping module comp_case to page 1.
Exec: { test -f '/home/tvsmohan99/.yosys_show.dot.pid' && fuser -s '/home/tvsmohan99/.yosys_show.dot.pid' 2> /dev/null; } || ( echo $$
>&3; exec xdot '/home/tvsmohan99/.yosys_show.dot'; ) 3> '/home/tvsmohan99/.yosys_show.dot.pid' &

yosys> 

```



**OBSERVATIONS:** Clearly, there is no latch and circuit is working fine in both simulation and synthesis.

## SKY130RTL D5SK3 L3 Lab incomplete overlapping Case part3:

### PARTIAL\_CASE\_ASSIGN SIMULATION:

Command: iverilog partial\_case\_assign.v tb\_partial\_case\_assign.v

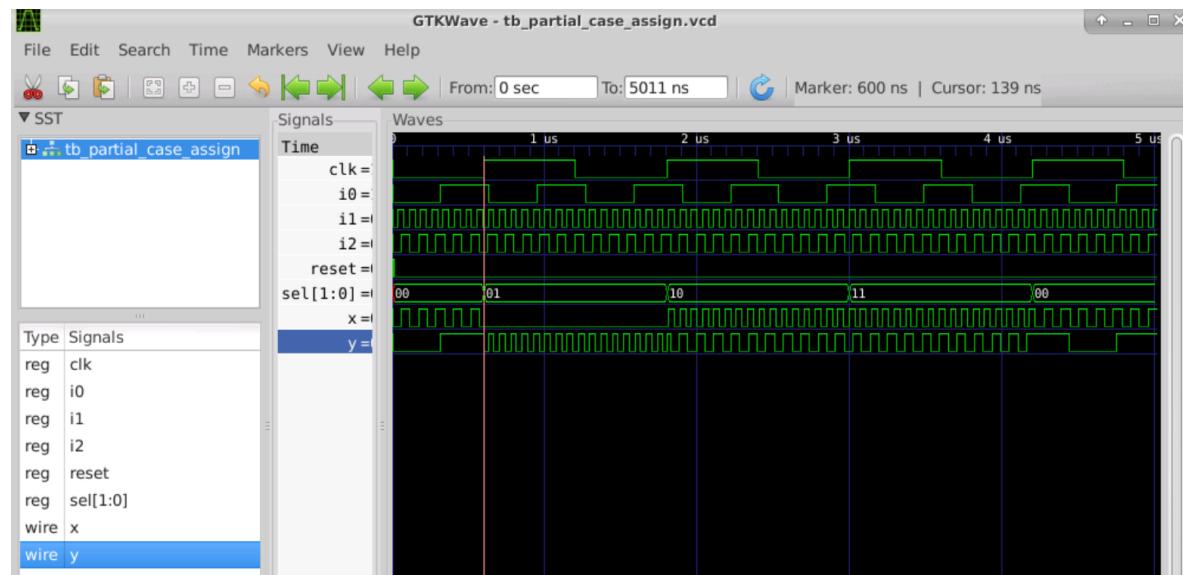
Command: ./a.out

Command: gtkwave tb\_partial\_case\_assign.vcd

```
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ iverilog partial_case_assign.v tb_partial_case_assign.v
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ ./a.out
VCD info: dumpfile tb_partial_case_assign.vcd opened for output.
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_partial_case_assign.vcd

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

[0] start time.
[5011000] end time.
[
```



OBSERVATION: Clearly, when select is 01, the x is latching to its value.

### PARTIAL\_CASE\_ASSIGN SYNTHESIS:

Command: yosys

Command: read\_liberty -lib ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: read\_verilog partial\_case\_assign.v

Command: synth -top partial\_case\_assign

Command: abc -liberty ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: show

```

yosys> read_liberty -lib ../my_lib/lib/sky130_fd_sc_hd_tt_025C_lv80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog partial_case_assign.v

2. Executing Verilog-2005 frontend: partial_case_assign.v
Parsing Verilog input from 'partial_case_assign.v' to AST representation.
Generating RTLIL representation for module '\partial_case_assign'.
Successfully finished Verilog frontend.

yosys> █

```

### 3.25. Printing statistics.

```

==== partial_case_assign ===

Number of wires: 13
Number of wire bits: 14
Number of public wires: 6
Number of public wire bits: 7
Number of memories: 0
Number of memory bits: 0
Number of processes: 0
Number of cells: 9
    $_ANDNOT_ 2
    $_AND_ 1
    $_DLATCH_P_ 1
    $_MUX_ 2
    $_NOR_ 1
    $_ORNOD_ 1
    $_OR_ 1

```

### 3.26. Executing CHECK pass (checking for obvious problems).

Checking module partial\_case\_assign...

Found and reported 0 problems.

```

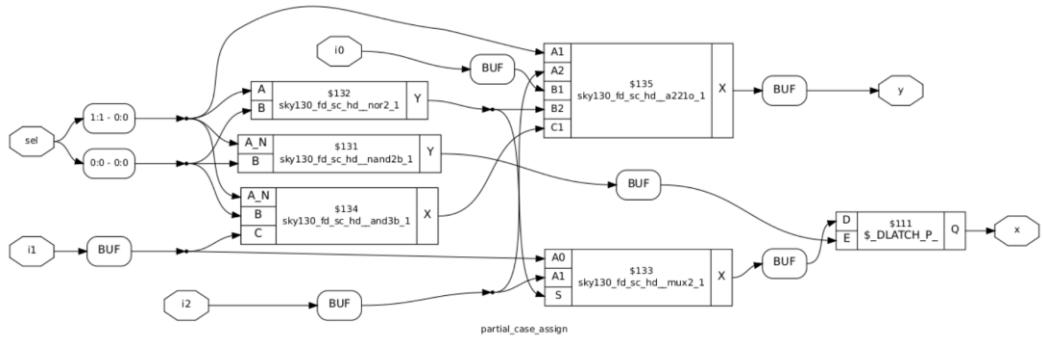
4.1.2. Re-integrating ABC results.
ABC RESULTS: sky130_fd_sc_hd_a221o_1 cells: 1
ABC RESULTS: sky130_fd_sc_hd_and3b_1 cells: 1
ABC RESULTS: sky130_fd_sc_hd_mux2_1 cells: 1
ABC RESULTS: sky130_fd_sc_hd_nand2b_1 cells: 1
ABC RESULTS: sky130_fd_sc_hd_nor2_1 cells: 1
ABC RESULTS: internal signals: 5
ABC RESULTS: input signals: 5
ABC RESULTS: output signals: 3
Removing temp directory.

yosys> show

5. Generating Graphviz representation of design.
Writing dot description to '/home/tvsmohan99/.yosys_show.dot'.
Dumping module partial_case_assign to page 1.
Exec: { test -f '/home/tvsmohan99/.yosys_show.dot.pid' && fuser -s '/home/tvsmohan99/.yosys_show.dot.pid' 2> /dev/null; } || ( echo $$ >&3; exec xdot '/home/tvsmohan99/.yosys_show.dot'; ) 3> '/home/tvsmohan99/.yosys_show.dot.pid' &

yosys> █

```



## OBSERVATIONS:

- 1) The dot viewer clearly shows the latch for x and there is no latch for y.

## BAD\_CASE SIMULATION:

Command: iverilog bad\_case.v tb\_bad\_case.v

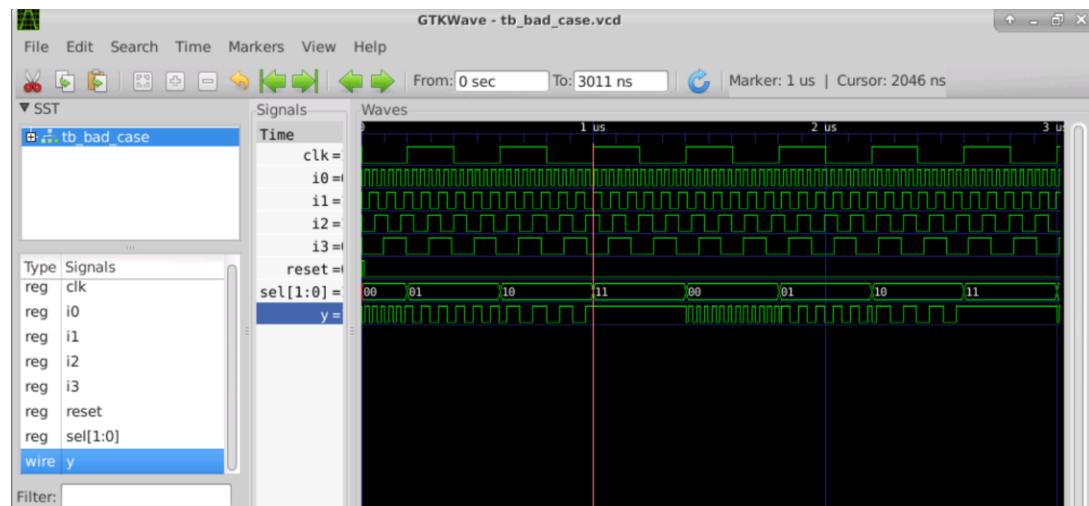
Command: ./a.out

Command: gtkwave tb\_bad\_case.vcd

```
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ iverilog bad_case.v tb_bad_case.v
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ ./a.out
VCD info: dumpfile tb_bad_case.vcd opened for output.
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_bad_case.vcd

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

[0] start time.
[3011000] end time.
```



OBSERVATION: Clearly, when the select is 11, the output y is becoming 1.

## SKY130RTL D5SK3 L4 Lab incomplete overlapping Case part4:

### BAD\_CASE SYNTHESIS:

Command:yosys

Command:read\_liberty–lib ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:read\_verilog bad\_case.v

Command:synth–top bad\_case

Command:abc–liberty ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:show

```
yosys> read_liberty -lib ..../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog bad_case.v

2. Executing Verilog-2005 frontend: bad_case.v
Parsing Verilog input from 'bad_case.v' to AST representation.
Warning: Yosys has only limited support for tri-state logic at the moment. (bad_case.v:8)
Generating RTLIL representation for module '\bad_case'.
Successfully finished Verilog frontend.

yosys> █
```

### 3.25. Printing statistics.

```
==== bad_case ===
```

Number of wires:	15
Number of wire bits:	16
Number of public wires:	6
Number of public wire bits:	7
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	10
\$_ANDNOT_	3
\$_AND_	1
\$_MUX_	1
\$_ORN0T_	2
\$_OR_	3

### 3.26. Executing CHECK pass (checking for obvious problems).

Checking module bad\_case...

Found and reported 0 problems.

Observation:There are no latches in bad\_case.

```

4.1.2. Re-integrating ABC results.
ABC RESULTS: sky130_fd_sc_hd_mux4_2 cells:      1
ABC RESULTS: internal signals: 9
ABC RESULTS: input signals: 6
ABC RESULTS: output signals: 1
Removing temp directory.

yosys> show

5. Generating Graphviz representation of design.
Writing dot description to '/home/tvsmohan99/.yosys_show.dot'.
Dumping module bad_case to page 1.
Exec: { test -f '/home/tvsmohan99/.yosys_show.dot.pid' && fuser -s '/home/tvsmohan99/.yosys_show.dot.pid' 2> /dev/null; } || ( echo $$
>&3; exec xdot '/home/tvsmohan99/.yosys_show.dot'; ) 3> '/home/tvsmohan99/.yosys_show.dot.pid' &

yosys>

```

### BAD\_MUXNETLSIT SIMULATION:

Command: write\_verilog –noattrb \_bad\_case\_net.v

Command: iverilog ..//my\_lib/verilog\_model/primitives.v ..//my\_lib/verilog\_model/sky130\_fd\_sc\_hd.v  
bad\_case\_net.v tb \_bad\_case.v

Command: ./a.out

Command: gtkwave tb \_bad\_case.vcd

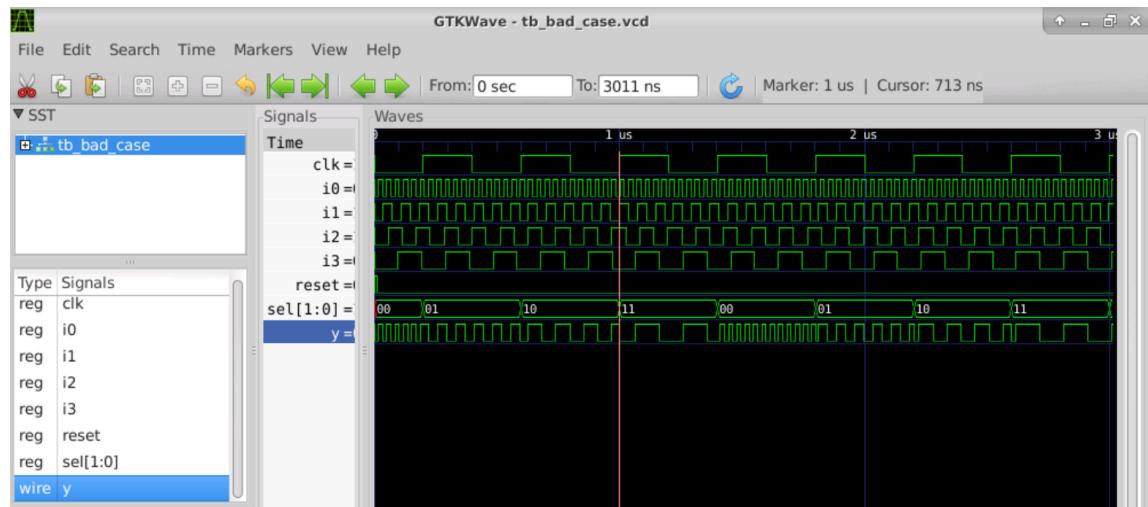
```

tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ iverilog ..//my_lib/verilog_model/primitives.v ..//my_lib/verilog_model/sky130_fd_sc_hd.v bad_case_net.v tb _bad_case.v
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ ./a.out
VCD info: dumpfile tb _bad_case.vcd opened for output.
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gtkwave tb _bad_case.vcd

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

[0] start time.
[3011000] end time.

```



### OBSERVATION:

- 1)The netlist simulations shows that the output y is taking i3, when the select is 11.
- 2) This is a simulation-synthesis mismatch. So, we need to avoid overlapping case while VERILOG coding.

## SKY130RTL D5SK5 L1 Lab For and For Generate part1:

### MUX\_GENERATE.V SIMULATION:

Command: iverilog mux\_generate.v tb\_mux\_generate.v

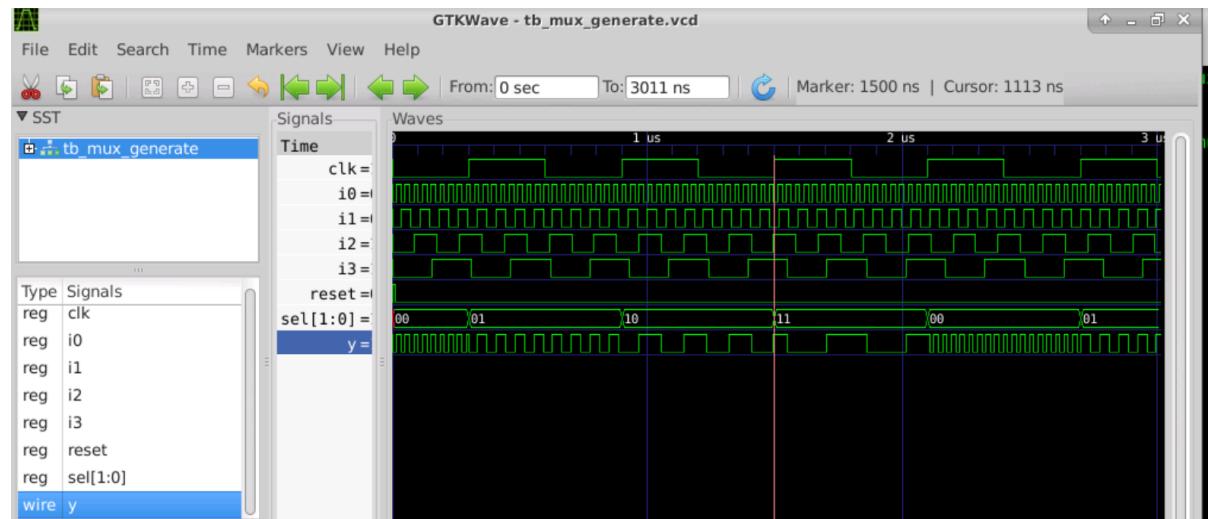
Command: ./a.out

Command: gtkwave tb\_mux\_generate.vcd

```
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ iverilog mux_genrate.v tb_mux_genrate.v
mux_genrate.v: No such file or directory
No top level modules, and no -s option.
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ iverilog mux_generate.v tb_mux_generate.v
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ ./a.out
VCD info: dumpfile tb_mux_generate.vcd opened for output.
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_mux_generate.vcd

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

[0] start time.
[3011000] end time.
```



OBSERVATION: We can see the waveforms of a 4:1 multiplexer designed using for loop.

### **MUX\_GENERATE.V SYNTHESIS:**

Command:yosys

Command:read\_liberty–lib ..//my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:read\_verilog mux\_generate.v

Command:synth–top mux\_generate

Command:abc–liberty ..//my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:show

```
yosys> read_liberty -lib ..//my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog mux_generate.v

2. Executing Verilog-2005 frontend: mux_generate.v
Parsing Verilog input from 'mux_generate.v' to AST representation.
Generating RTLIL representation for module '\mux_generate'.
Successfully finished Verilog frontend.
```

### **3.25. Printing statistics.**

```
==== mux_generate ===
```

Number of wires:	15
Number of wire bits:	50
Number of public wires:	8
Number of public wire bits:	43
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	8
\$_ANDNOT_	3
\$_DLATCH_N_	1
\$_MUX_	3
\$_NAND_	1

```

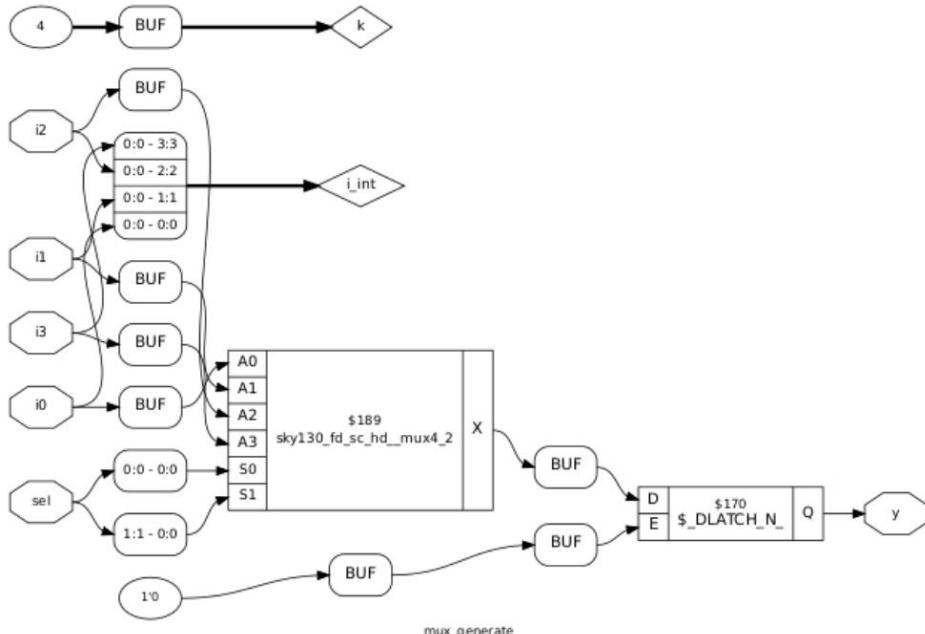
4.1.2. Re-integrating ABC results.
ABC RESULTS: const0 cells: 1
ABC RESULTS: sky130_fd_sc_hd_mux4_2 cells: 1
ABC RESULTS: internal signals: 5
ABC RESULTS: input signals: 6
ABC RESULTS: output signals: 2
Removing temp directory.

yosys> show

5. Generating Graphviz representation of design.
Writing dot description to '/home/tvsmohan99/yosys_show.dot'.
Dumping module mux generate to page 1.
Exec: { test -f '/home/tvsmohan99/yosys_show.dot.pid' && fuser -s '/home/tvsmohan99/.yosys_show.dot.pid' 2> /dev/null; } || ( echo $$ >63; exec xdot '/home/tvsmohan99/.yosys_show.dot'; ) 3> '/home/tvsmohan99/.yosys_show.dot.pid' &

yosys>

```



OBSERVATION: We can see that the synthesis has a latch. The latch just acts as a buffer.

### SKY130RTL D5SK5 L2 Lab For and For Generate part2:

#### DEMUX\_CASE SIMULATION:

Command: iverilog demux\_case.v tb\_demux\_case.v

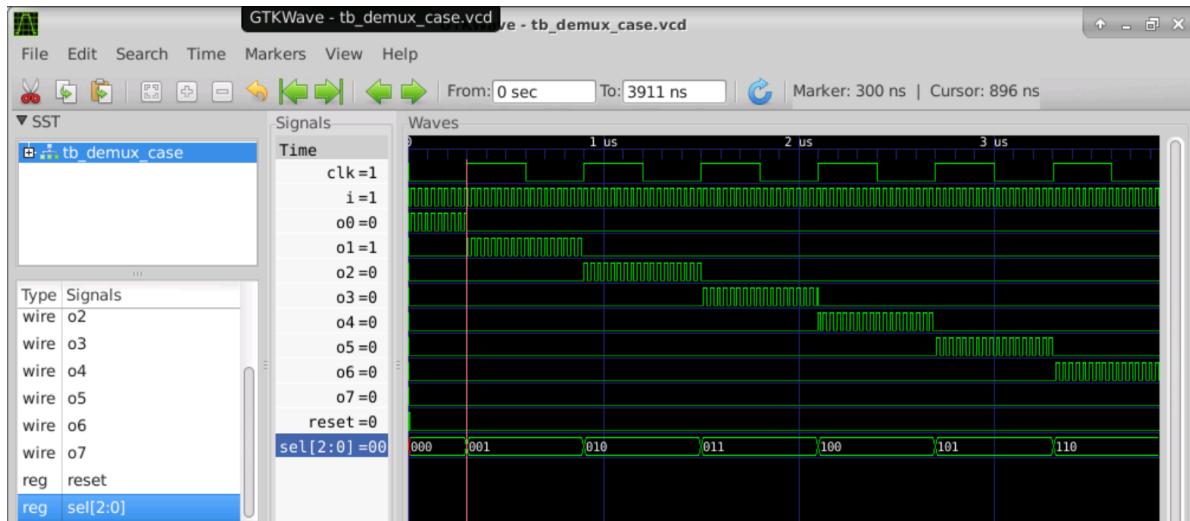
Command: ./a.out

Command: gtkwave tb\_demux\_case.vcd

```
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ iverilog demux_case.v tb_demux_case.v
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ ./a.out
VCD info: dumpfile tb_demux_case.vcd opened for output.
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_demux_case.vcd

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

[0] start time.
[3911000] end time.
```



OBSERVATION: We can see that output is being selected based on the select line.

### **DEMUX\_CASE SYNTHESIS:**

Command: yosys

Command: read\_liberty -lib ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: read\_verilog demux\_case.v

Command: synth -top demux\_case

Command: abc -liberty ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: show

```
yosys> read_liberty -lib ..../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog demux_case.v

2. Executing Verilog-2005 frontend: demux_case.v
Parsing Verilog input from `demux_case.v' to AST representation.
Generating RTLIL representation for module `demux_case'.
Successfully finished Verilog frontend.
```

```
==== demux_case ====

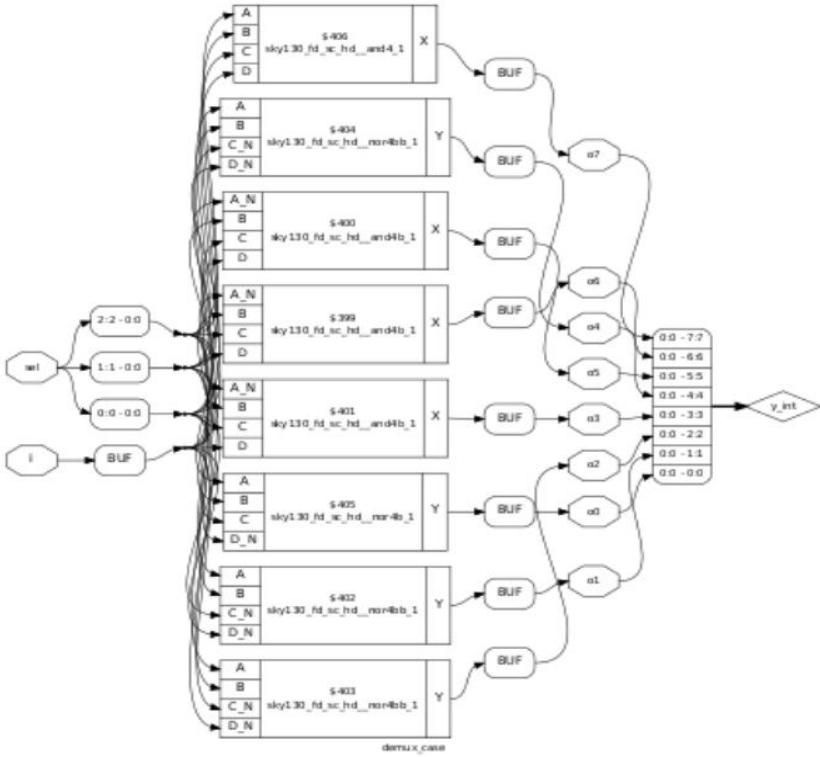
Number of wires:          46
Number of wire bits:      55
Number of public wires:   11
Number of public wire bits: 20
Number of memories:       0
Number of memory bits:    0
Number of processes:      0
Number of cells:          43
$_ANDNOT_                 14
$_AND_                      1
$_MUX_                      3
$_NAND_                     1
$_NOR_                      1
$_NOT_                      1
$_ORNOD_                    4
$_OR_                       18
```

3.26. Executing CHECK pass (checking for obvious problems).  
 Checking module demux\_case...  
 Found and reported 0 problems.

```
4.1.2. Re-integrating ABC results.
ABC RESULTS: sky130_fd_sc_hd_and4_1 cells:      1
ABC RESULTS: sky130_fd_sc_hd_and4b_1 cells:      3
ABC RESULTS: sky130_fd_sc_hd_nor4b_1 cells:      1
ABC RESULTS: sky130_fd_sc_hd_nor4bb_1 cells:     3
ABC RESULTS:      internal signals:      35
ABC RESULTS:      input signals:        4
ABC RESULTS:      output signals:       8
Removing temp directory.

yosys> show

5. Generating Graphviz representation of design.
Writing dot description to '/home/tvsmohan99/.yosys_show.dot'.
Dumping module demux_case to page 1.
Exec: { test -f '/home/tvsmohan99/.yosys_show.dot.pid' && fuser -s '/home/tvsmohan99/.yosys_show.dot.pid' 2> /dev/null; } || ( echo $$ >&3; exec xdot '/home/tvsmohan99/.yosys_show.dot'; ) 3> '/home/tvsmohan99/.yosys_show.dot.pid'
```



### DEMUX\_GENERATE.VSIMULATION:

Command: iverilog demux\_generate.v tb\_demux\_generate.v

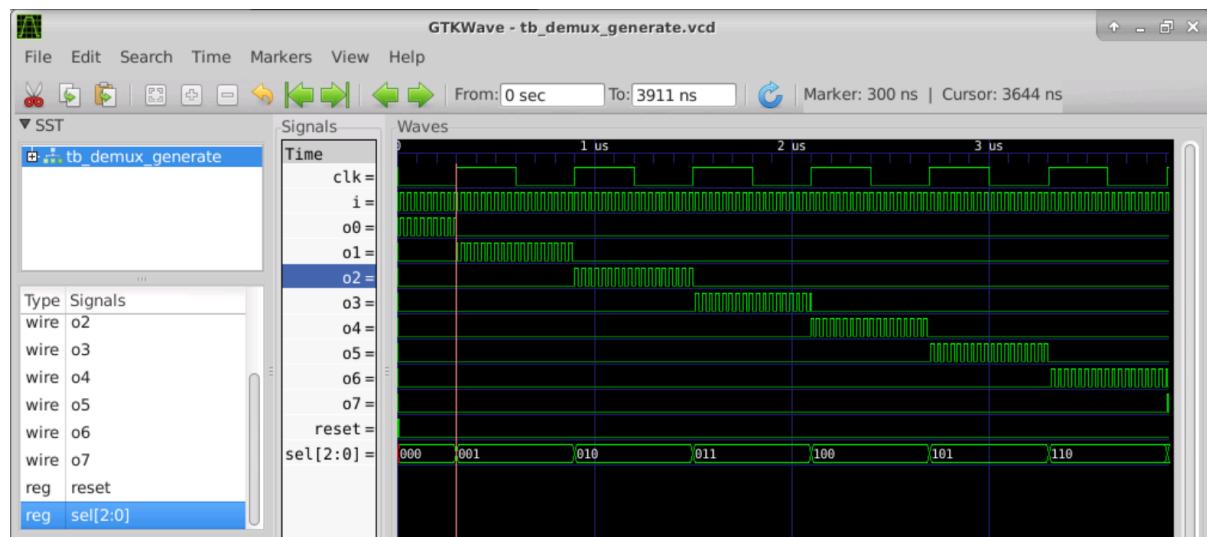
Command: ./a.out

Command: gtkwave tb\_demux\_generate.vcd

```
tvsmanoh99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ iverilog demux_generate.v tb_demux_generate.v
tvsmanoh99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ ./a.out
VCD info: dumpfile tb_demux_generate.vcd opened for output.
tvsmanoh99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_demux_generate.vcd

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

[0] start time.
[3911000] end time.
```



## **DEMUX\_GENERATE SYNTHESIS:**

Command:yosys

Command:read\_liberty –lib ..//my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:read\_verilog demux\_generate.v

Command:synth –top demux\_generate

Command:abc –liberty ..//my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command:show

```
yosys> read_liberty -lib ..//my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog demux_generate.v
2. Executing Verilog-2005 frontend: demux_generate.v
Parsing Verilog input from `demux_generate.v' to AST representation.
Generating RTLIL representation for module `demux_generate'.
Successfully finished Verilog frontend.

yosys>
```

## **3.25. Printing statistics.**

==== demux\_generate ===

Number of wires:	25
Number of wire bits:	65
Number of public wires:	12
Number of public wire bits:	52
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	21
\$_ANDNOT_	8
\$_NAND_	1
\$_NOT_	1
\$_ORNOD_	2
\$_OR_	9

## **3.26. Executing CHECK pass (checking for obvious problems).**

Checking module demux\_generate...

Found and reported 0 problems.

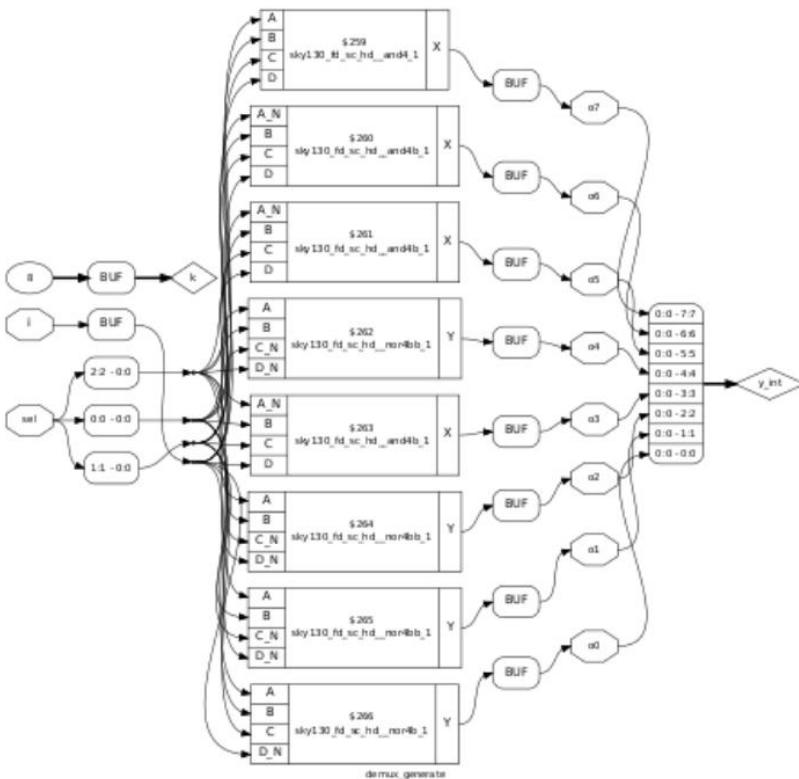
```

4.1.2. Re-integrating ABC results.
ABC RESULTS: sky130_fd_sc_hd_and4_1 cells:      1
ABC RESULTS: sky130_fd_sc_hd_and4b_1 cells:      3
ABC RESULTS: sky130_fd_sc_hd_nor4b_1 cells:      1
ABC RESULTS: sky130_fd_sc_hd_nor4bb_1 cells:     3
ABC RESULTS: internal signals:      13
ABC RESULTS: input signals:        4
ABC RESULTS: output signals:       8
Removing temp directory.

yosys> show

5. Generating Graphviz representation of design.
Writing dot description to '/home/tvsmohan99/.yosys_show.dot'.
Dumping module demux_generate to page 1.
Exec: { test -f '/home/tvsmohan99/.yosys_show.dot.pid' && fuser -s '/home/tvsmohan99/.yosys_show.dot.pid' 2> /dev/null; } || ( echo $$
>&3; exec xdot '/home/tvsmohan99/.yosys_show.dot'; ) 3> '/home/tvsmohan99/.yosys_show.dot.pid' &

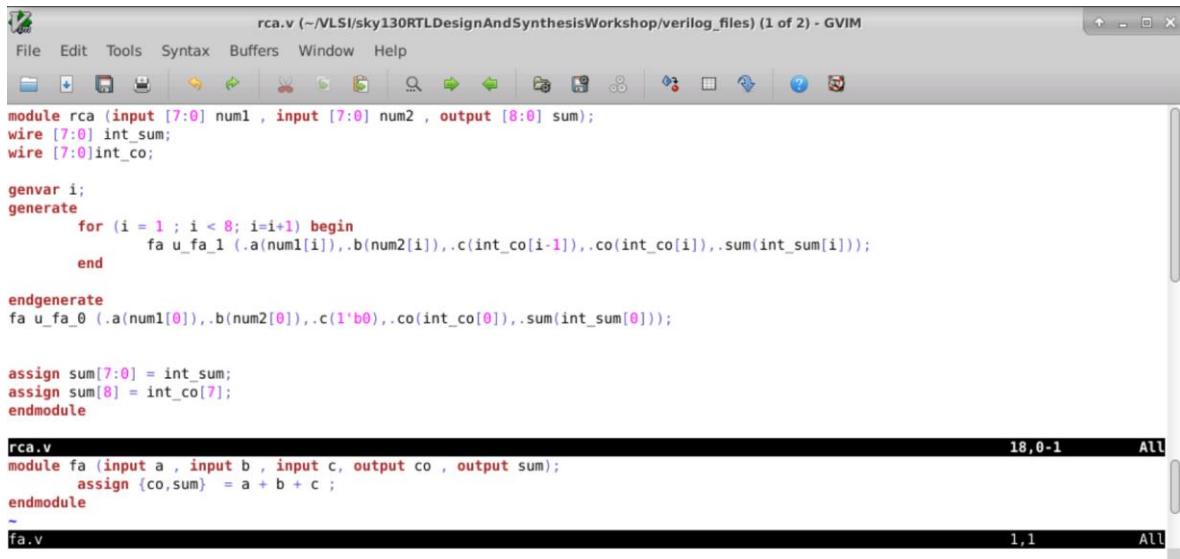
```



OBSERVATION: In this lab, both demux\_case.v and demux\_generate.v has similar synthesis. However, the demux\_generate.v can be scalable.

### SKY130RTL D5SK5 L3 Lab For and For Generate part3:

Command: gvim rca.v -o fa.v



The screenshot shows a GVIM window with two tabs open. The current tab, 'rca.v', contains the following Verilog code:

```
module rca (input [7:0] num1 , input [7:0] num2 , output [8:0] sum);
wire [7:0] int_sum;
wire [7:0]int_co;

genvar i;
generate
    for (i = 1 ; i < 8; i=i+1) begin
        fa u_fa_1 (.a(num1[i]),.b(num2[i]),.c(int_co[i-1]),.co(int_co[i]),.sum(int_sum[i]));
    end
endgenerate
fa u_fa_0 (.a(num1[0]),.b(num2[0]),.c(1'b0),.co(int_co[0]),.sum(int_sum[0]));

assign sum[7:0] = int_sum;
assign sum[8] = int_co[7];
endmodule

rca.v                                         18.0-1      All
module fa (input a , input b , input c, output co , output sum);
    assign {co,sum} = a + b + c ;
endmodule
~
fa.v                                           1,1       All
```

### SKY130RTL D5SK5 L4 Lab For and For Generate part4:

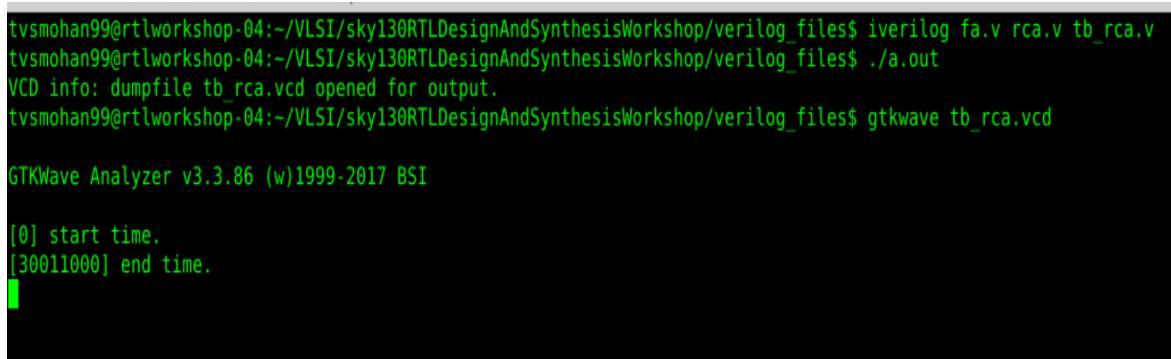
#### RCA SIMULATION:

Command: iverilog fa.v rca.v tb\_rca.v

This command tells the simulator that fa.v is separate module but is called in rca.v.

Command: ./a.out

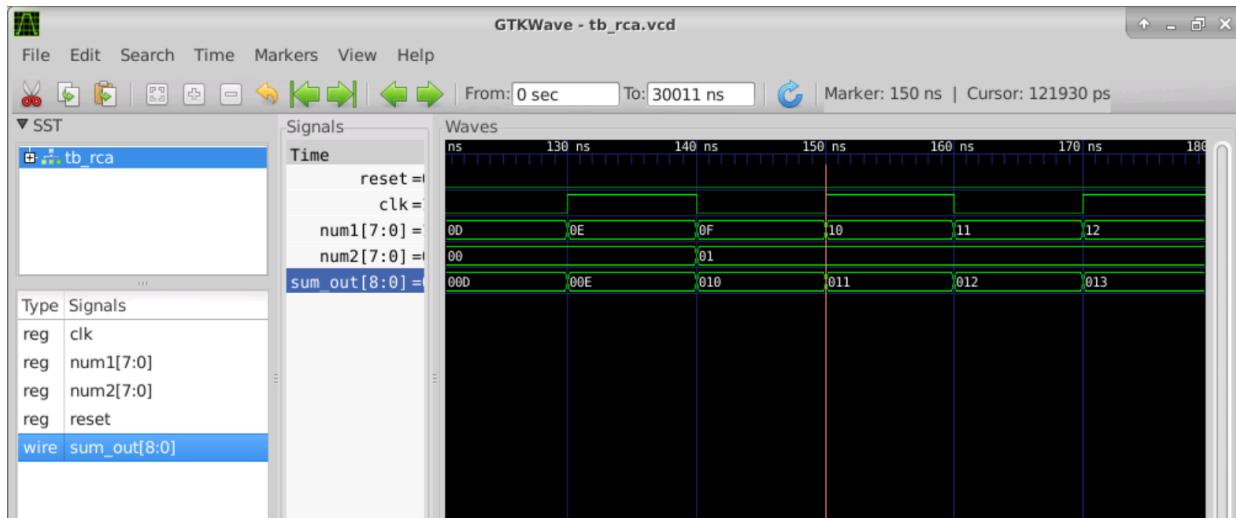
Command: gtkwave tb\_rca.vcd



```
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDdesignAndSynthesisWorkshop/verilog_files$ iverilog fa.v rca.v tb_rca.v
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDdesignAndSynthesisWorkshop/verilog_files$ ./a.out
VCD info: dumpfile tb_rca.vcd opened for output.
tvsomohan99@rtlworkshop-04:~/VLSI/sky130RTLDdesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_rca.vcd

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

[0] start time.
[30011000] end time.
```



### RCS SYNTHESIS:

Command: yosys

Command: read\_liberty -lib ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: read\_verilog fa.v

Command: read\_verilog rca.v

Command: synth -top rca

Command: abc -liberty ..../my\_lib/lib/sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

Command: show rca

```
yosys> read_liberty -lib ..../my_lib/lib/sky130_fd_sc_hd_tt_025C_1v80.lib
1. Executing Liberty frontend.
Imported 428 cell types from liberty file.

yosys> read_verilog fa.v

2. Executing Verilog-2005 frontend: fa.v
Parsing Verilog input from `fa.v' to AST representation.
Generating RTLIL representation for module `\fa'.
Successfully finished Verilog frontend.

yosys> read_verilog rca.v

3. Executing Verilog-2005 frontend: rca.v
Parsing Verilog input from `rca.v' to AST representation.
Generating RTLIL representation for module `\rca'.
Successfully finished Verilog frontend.
```

```

==== design hierarchy ===

rca                      1
  fa                      8

Number of wires:          69
Number of wire bits:     105
Number of public wires:   45
Number of public wire bits: 81
Number of memories:      0
Number of memory bits:   0
Number of processes:     0
Number of cells:          40
  $_ANDNOT_                 8
  $_AND_                     8
  $_OR_                      8
  $_XNOR_                    16

4.26. Executing CHECK pass (checking for obvious problems).
Checking module fa...
Checking module rca...
Found and reported 0 problems.

```

```

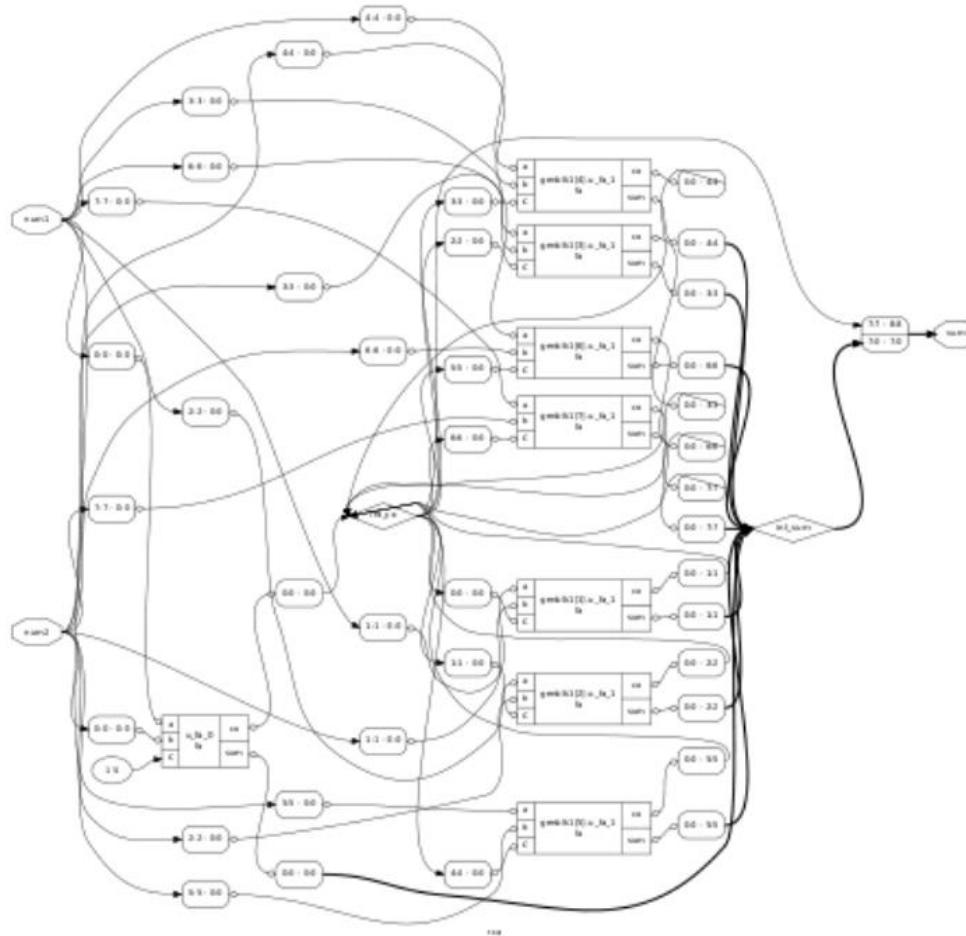
5.1.2. Re-integrating ABC results.
ABC RESULTS: sky130_fd_sc_hd_maj3_1 cells:      1
ABC RESULTS: sky130_fd_sc_hd_xor3_1 cells:      1
ABC RESULTS: internal signals:                  3
ABC RESULTS: input signals:                     3
ABC RESULTS: output signals:                   2
Removing temp directory.

5.2. Extracting gate netlist of module '\rca' to '<abc-temp-dir>/input.blif'..
Extracted 0 gates and 0 wires to a netlist network with 0 inputs and 0 outputs.
Don't call ABC as there is nothing to map.
Removing temp directory.

yosys> show rca

6. Generating Graphviz representation of design.
Writing dot description to '/home/tvsmohan99/.yosys_show.dot'.
Dumping module rca to page 1.
Exec: { test -f '/home/tvsmohan99/.yosys_show.dot.pid' && fuser -s '/home/tvsmohan99/.yosys_show.dot.pid' 2> /dev/null; } || ( echo $$
>63; exec xdot '/home/tvsmohan99/.yosys_show.dot'; ) 3> '/home/tvsmohan99/.yosys_show.dot.pid' &

```



OBSERVATION: 8 FA MODULES ARE INSTANTIATED.

#### RCA GLS SYNTHESIS:

Command: write\_verilog -noattr tb\_rca\_net.v

```
yosys> write_verilog -noattr tb_rca_net.v

8. Executing Verilog backend.
Dumping module `fa'.
Dumping module `rca'.
```

Command: iverilog ..//my\_lib/verilog\_module/primitives.v  
..//my\_lib/verilog\_model/sky130\_fd\_sc\_hd.v tb\_rca\_net.v tb\_rca.v

Command: ./a.out

Command: gtkwave tb\_rca.vcd

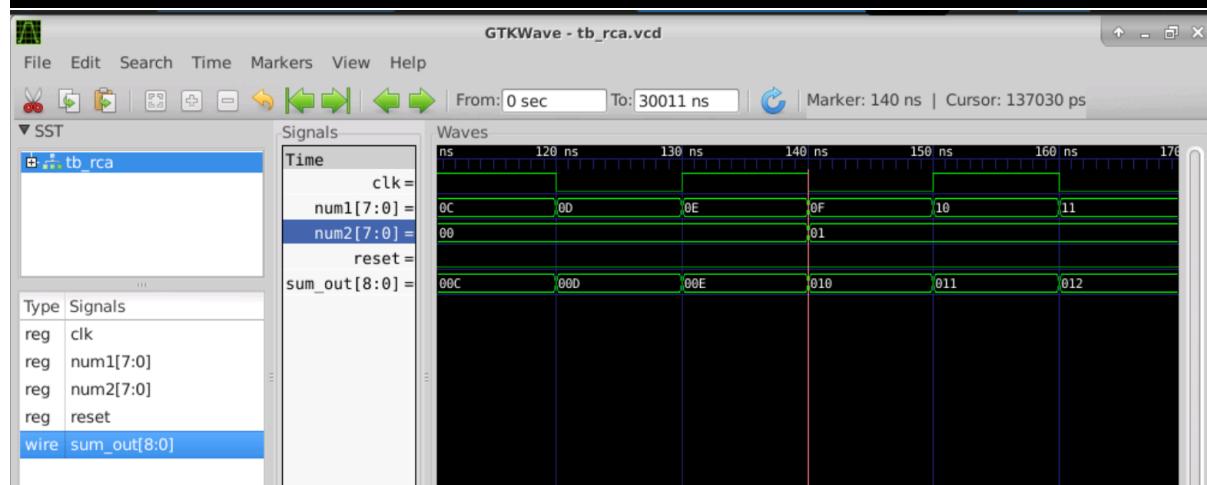
```

tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ iverilog ..//my_lib/verilog_model/primitives.v ..//my_lib/verilog model/sky130_fd_sc_hd.v tb_rca_net.v tb_rca.v
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ ./a.out
VCD info: dumpfile tb_rca.vcd opened for output.
tvsmohan99@rtlworkshop-04:~/VLSI/sky130RTLDesignAndSynthesisWorkshop/verilog_files$ gtkwave tb_rca.vcd

```

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

[0] start time.  
[30011000] end time.



OBSERVATIONS: There is no mismatch between simulation and synthesis.