TRƯỜNG ĐẠI HỌC BÁCH KHOA

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

KHOA ĐIỆN – ĐIỆN TỬ

LOGIC SYNTHESIS AND VLSI FRONT-END FLOW

Môn: THIẾT KẾ VI MẠCH

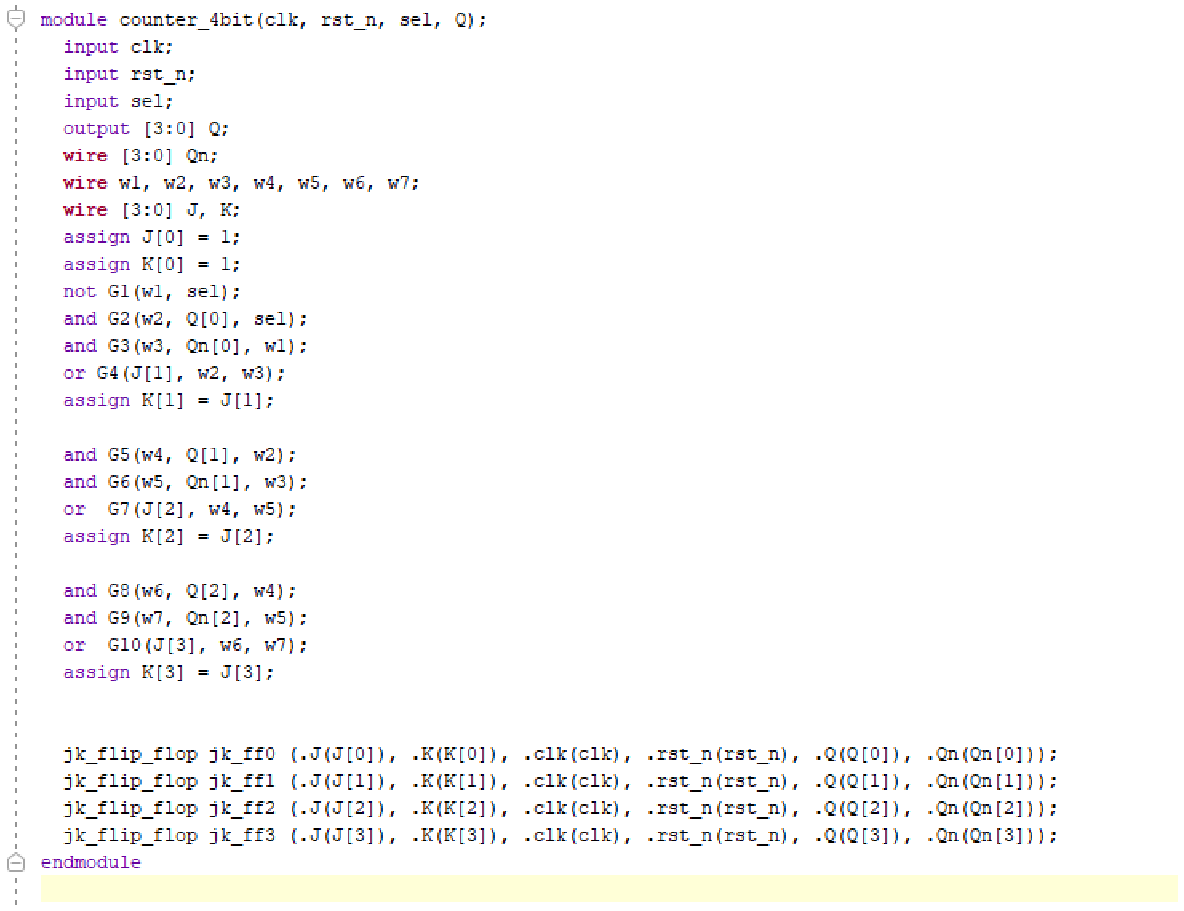
Giảng Viên Hướng Dẫn: TRẦN HOÀNG QUÂN

I. LOGIC SYNTHESIS

Lab 2:

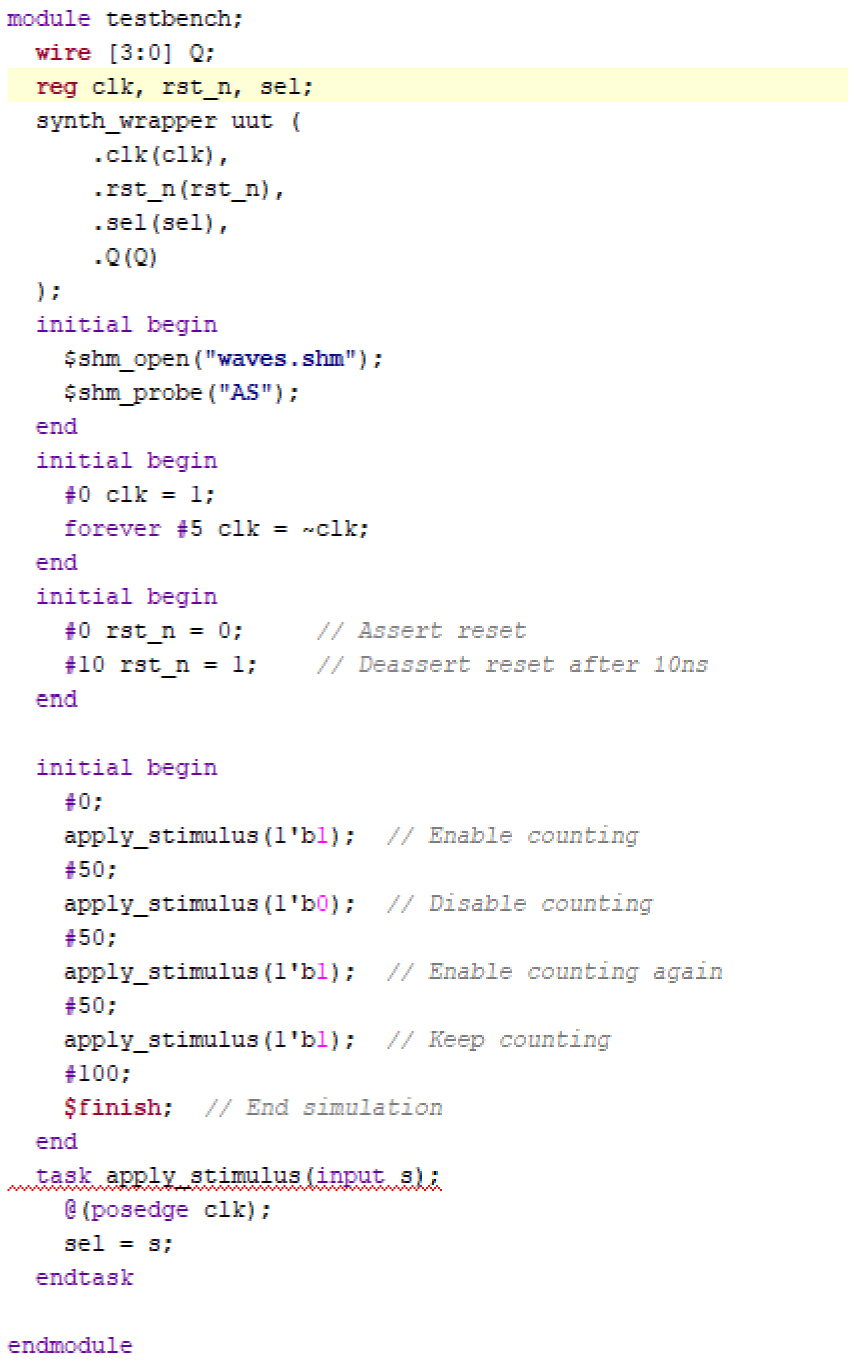
Design and synthesize "4-bit counter" specified in the previous RTL and verification lab, use verification plan in previous lab.

1. Code



```
module counter_4bit(clk, rst_n, sel, Q);
  input clk;
  input rst_n;
  input sel;
  output [3:0] Q;
  wire [3:0] Qn;
  wire w1, w2, w3, w4, w5, w6, w7;
  wire [3:0] J, K;
  assign J[0] = 1;
  assign K[0] = 1;
  not G1(w1, sel);
  and G2(w2, Q[0], sel);
  and G3(w3, Qn[0], w1);
  or G4(J[1], w2, w3);
  assign K[1] = J[1];

  and G5(w4, Q[1], w2);
  and G6(w5, Qn[1], w3);
  or  G7(J[2], w4, w5);
  assign K[2] = J[2];

  and G8(w6, Q[2], w4);
  and G9(w7, Qn[2], w5);
  or  G10(J[3], w6, w7);
  assign K[3] = J[3];


  jk_flip_flop jk_ff0 (.J(J[0]), .K(K[0]), .clk(clk), .rst_n(rst_n), .Q(Q[0]), .Qn(Qn[0]));
  jk_flip_flop jk_ff1 (.J(J[1]), .K(K[1]), .clk(clk), .rst_n(rst_n), .Q(Q[1]), .Qn(Qn[1]));
  jk_flip_flop jk_ff2 (.J(J[2]), .K(K[2]), .clk(clk), .rst_n(rst_n), .Q(Q[2]), .Qn(Qn[2]));
  jk_flip_flop jk_ff3 (.J(J[3]), .K(K[3]), .clk(clk), .rst_n(rst_n), .Q(Q[3]), .Qn(Qn[3]));
endmodule
```

2. Code testbench



```verilog
module testbench;
  wire [3:0] Q;
  reg clk, rst_n, sel;
  synth_wrapper uut (
      .clk(clk),
      .rst_n(rst_n),
      .sel(sel),
      .Q(Q)
  );
  initial begin
    $shm_open("waves.shm");
    $shm_probe("AS");
  end
  initial begin
    #0 clk = 1;
    forever #5 clk = ~clk;
  end
  initial begin
    #0 rst_n = 0;      // Assert reset
    #10 rst_n = 1;     // Deassert reset after 10ns
  end

  initial begin
    #0;
    apply_stimulus(1'b1);   // Enable counting
    #50;
    apply_stimulus(1'b0);   // Disable counting
    #50;
    apply_stimulus(1'b1);   // Enable counting again
    #50;
    apply_stimulus(1'b1);   // Keep counting
    #100;
    $finish;   // End simulation
  end
  task apply_stimulus(input s);
    @(posedge clk);
    sel = s;
  endtask

endmodule
```

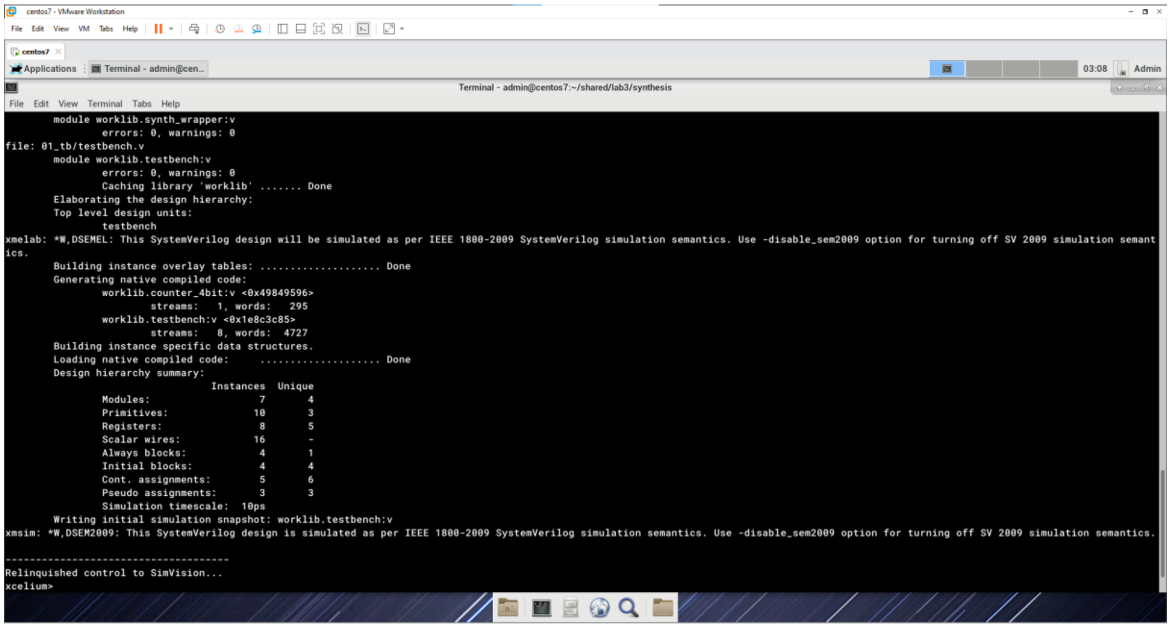
Dưới đây là tập lệnh Makefile trong bài lab này:

```
##################################################
#
#Tuan Hung
#HCMUT LAB3 counter_4bit
#
##################################################
PHONY: syn
syn: clean
	genus -f ./03_synth/genus_shell.tcl
##################################################
#GUI = -gui
tb_file := 01_tb/testbench.v

PHONY: sim
sim:
	xrun $(GUI) +xm64bit -sv \
	-f 00_src/flist.f \
	$(tb_file) \
	-timescale 1ns/10ps \
	+access+rcw
##################################################
#GUI = -gui
delay_mode := -delay_mode punit
#sdf_file := -sdf_cmd_file 03_synth/sdf.cmd
netlist := 03_synth/*_gate.v
test_file := 01_tb/testbench.v

PHONY: verify
verify:
	xrun $(GUI) +xm64bit -sv \
	$(netlist) \
	$(test_file) \
	-vlogext .sv \
	-f 02_sim/flist.f \
	$(sdf_file) \
	$(delay_mode) \
	-timescale 1ns/10ps \
	+access+rcw

###################################################
PHONY: report
report:
	@echo "with Frequency at: "
	@grep "set FREQ_GHz" 03_synth/genus_shell.tcl
	@grep "Path " 04_reports/*.timing.rpt

###################################################
PHONY: clean
clean:
	@rm -rf genus.cmd genus.log xrun.history xrun.key xcelium.d *.shm

###################################################
PHONY: clean-all
clean-all: clean-rpt clean-syn
	@rm -rf fv genus.cmd genus.log xrun.history xrun.log xrun.key
counter_4bit.sdf.X xcelium.d *.shm

###################################################
PHONY: clean-rpt
```

```
clean-rpt:
	@rm -rf 04_reports/*.rpt

#####################################################
PHONY: clean-syn
clean-syn:
	@rm -rf 03_synth/*.v 03_synth/*.sdf

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

3. Simulation

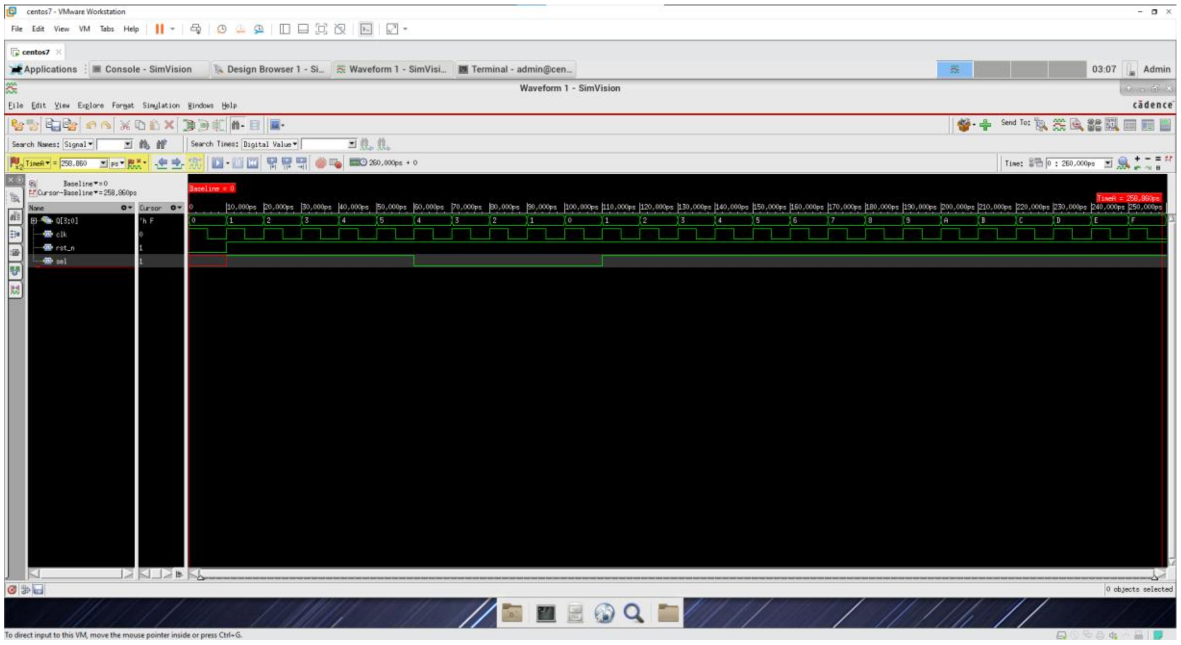
- Result make sim:



centos7 - VMware Workstation
File Edit View VM Tabs Help
centos7
Applications
Terminal - admin@cen...
03:08
Admin
Terminal - admin@centos7:~/shared/lab3/synthesis
File Edit View Terminal Tabs Help

```
        module worklib.synth_wrapper:v
                errors: 0, warnings: 0
file: 01_tb/testbench.v
        module worklib.testbench:v
                errors: 0, warnings: 0
                Caching library 'worklib' ....... Done
        Elaborating the design hierarchy:
        Top level design units:
                testbench
xmelab: *W,DSEMEL: This SystemVerilog design will be simulated as per IEEE 1800-2009 SystemVerilog simulation semant
ics.
        Building instance overlay tables: .................... Done
        Generating native compiled code:
                worklib.counter_4bit:v <0x49849596>
                        streams:   1, words:   295
                worklib.testbench:v <0x1e8c3c85>
                        streams:   8, words:  4727
        Building instance specific data structures.
        Loading native compiled code:     .................... Done
        Design hierarchy summary:
                                  Instances  Unique
                Modules:                  7       4
                Primitives:              10       3
                Registers:                8       5
                Scalar wires:            16       -
                Always blocks:            4       1
                Initial blocks:           4       4
                Cont. assignments:        5       6
                Pseudo assignments:       3       3
                Simulation timescale:  10ps
        Writing initial simulation snapshot: worklib.testbench:v
xmsim: *W,DSEM2009: This SystemVerilog design is simulated as per IEEE 1800-2009 SystemVerilog simulation semantics.

-------------------------------------
Relinquished control to SimVision...
xcelium>
```

- Kết quả mô phỏng:

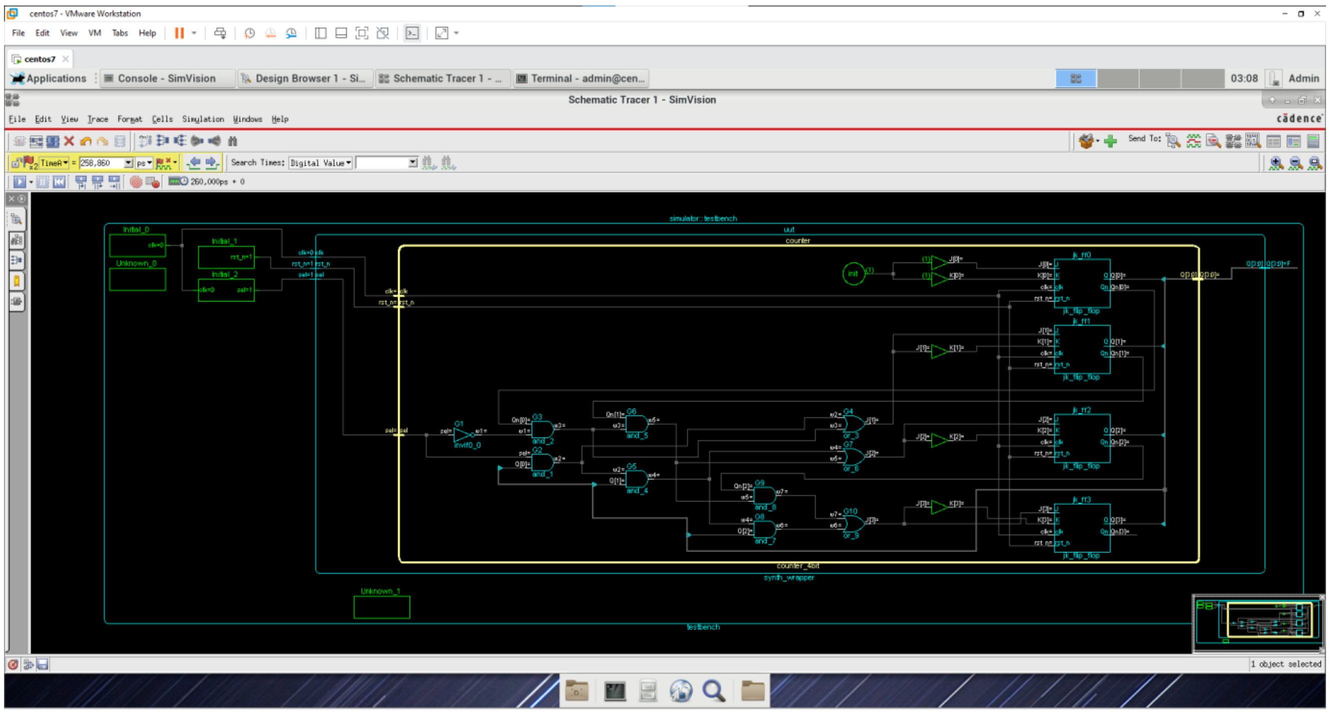


Verification plan:

Section	Item	Descreption	Status
1	Reset	When rst_n is asserted, the output is 0, when it de-asserts, the output start to count up each positive edge of clock	PASS
2	Max count	When output is 4’b1111 and the counter is counting up, the next positive edge clock output will be 4’b0000	PASS
3	Min count	When output is 4’b0000 and the counter is counting down, the next positive edge clock output will be 4’b1111	PASS

4. High level Arichitecture of the design

- Schematic của netlist:



- Mô tả ngõ vào/ra:

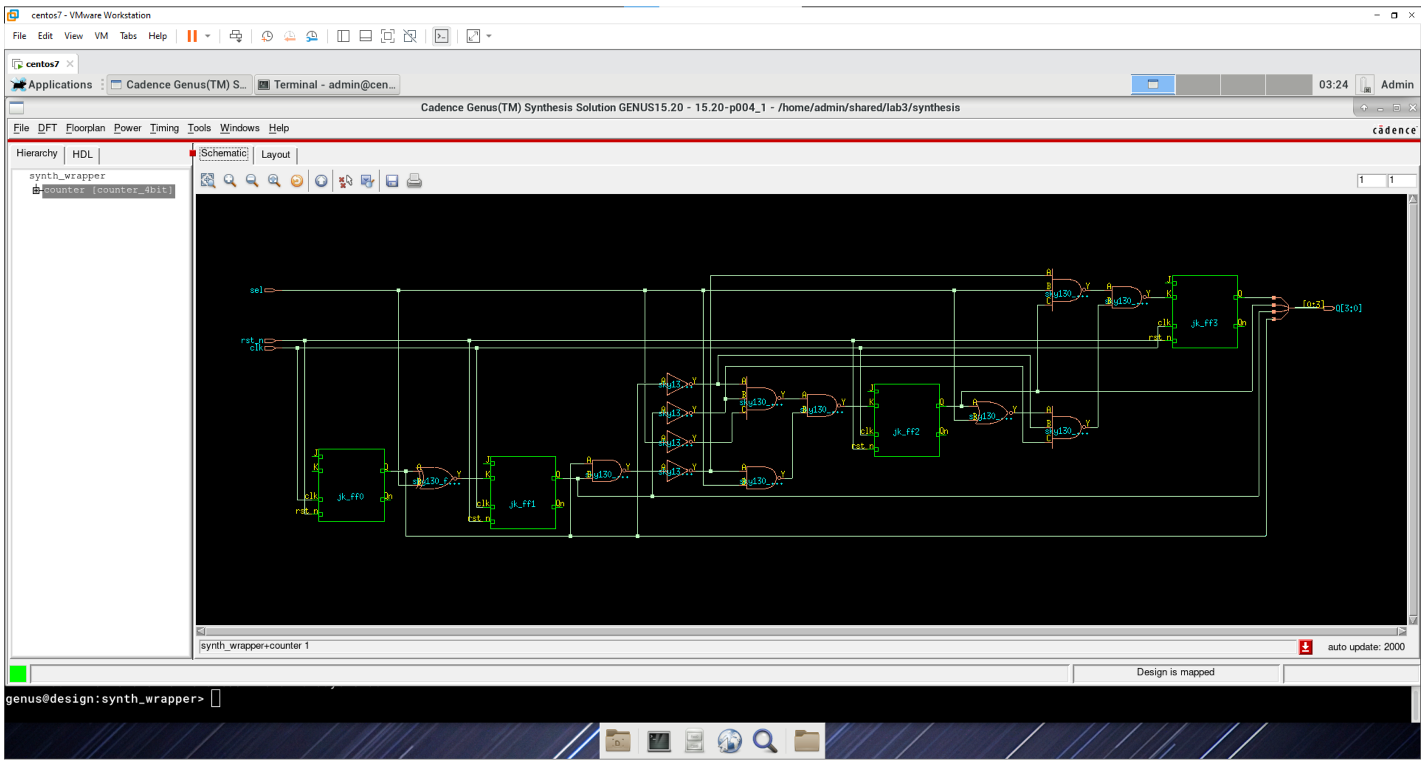
Signal	Width	Type	Description
Clk	1	input	Clock signal

Rst_n	1	input	Negative edge reset. If rst_n = 0, output will be set to 0. Else, it will start the normal operation.
Sel	1	input	Mode selection signal. If sel = 1, the design will start counting up. Else, it will start counting down from the current output value.
Out	4	output	Result of the counter.

- Sử dụng Flip Flop JK và các cổng Not, And, Or để thực hiện đếm lên hoặc đếm xuống trong bộ counter_4bit.

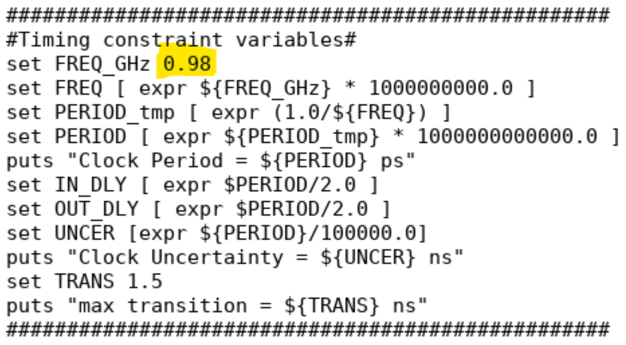
5. Report

- Schematic Viewer (main)



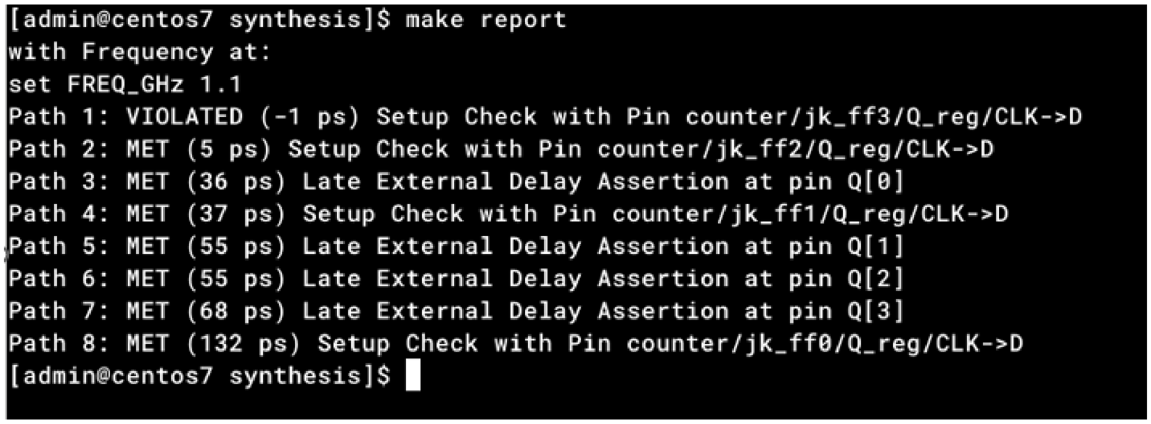
```
genus@design:synth_wrapper>
```

- Tần số cao nhất mà thiết kế có thể đạt được là 0.98 GHz



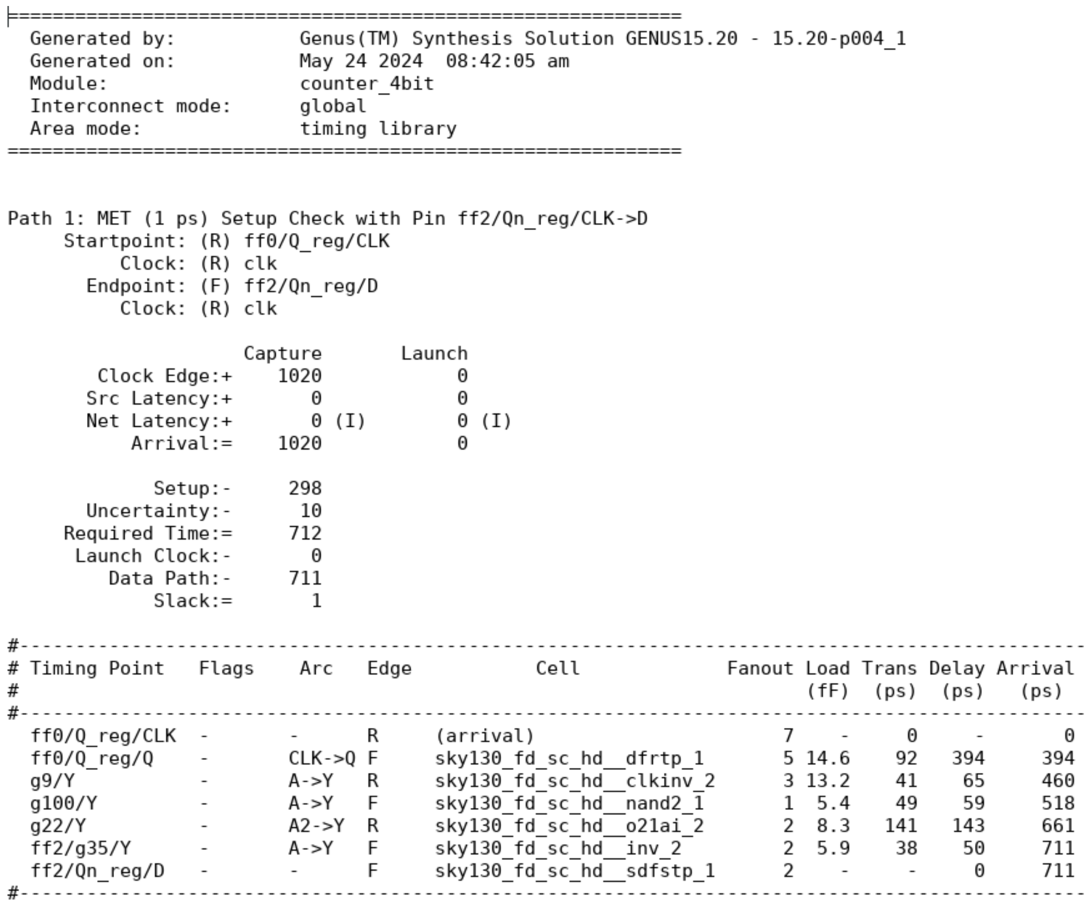
```
##################################################
#Timing constraint variables#
set FREQ_GHz 0.98
set FREQ [ expr ${FREQ_GHz} * 1000000000.0 ]
set PERIOD_tmp [ expr (1.0/${FREQ}) ]
set PERIOD [ expr ${PERIOD_tmp} * 1000000000000.0 ]
puts "Clock Period = ${PERIOD} ps"
set IN_DLY [ expr $PERIOD/2.0 ]
set OUT_DLY [ expr $PERIOD/2.0 ]
set UNCER [expr ${PERIOD}/100000.0]
puts "Clock Uncertainty = ${UNCER} ns"
set TRANS 1.5
puts "max transition = ${TRANS} ns"
##################################################
```

- Báo cáo tiến độ thiết kế:



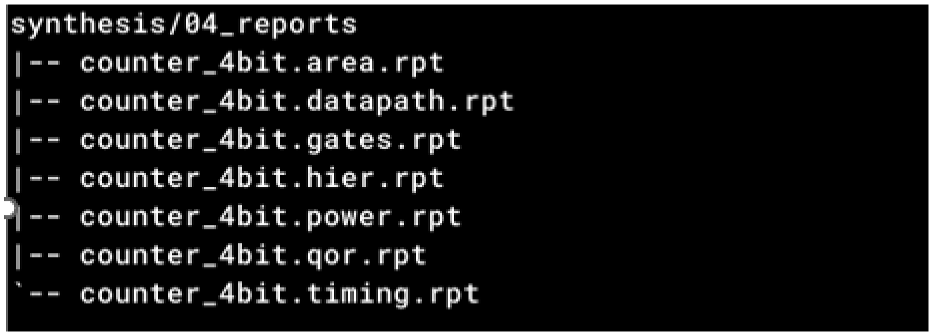
```
[admin@centos7 synthesis]$ make report
with Frequency at:
set FREQ_GHz 1.1
Path 1: VIOLATED (-1 ps) Setup Check with Pin counter/jk_ff3/Q_reg/CLK->D
Path 2: MET (5 ps) Setup Check with Pin counter/jk_ff2/Q_reg/CLK->D
Path 3: MET (36 ps) Late External Delay Assertion at pin Q[0]
Path 4: MET (37 ps) Setup Check with Pin counter/jk_ff1/Q_reg/CLK->D
Path 5: MET (55 ps) Late External Delay Assertion at pin Q[1]
Path 6: MET (55 ps) Late External Delay Assertion at pin Q[2]
Path 7: MET (68 ps) Late External Delay Assertion at pin Q[3]
Path 8: MET (132 ps) Setup Check with Pin counter/jk_ff0/Q_reg/CLK->D
[admin@centos7 synthesis]$
```

- Báo cáo cũng có thể được tìm thấy trong 04_report/counter_4bit.timing.rpt



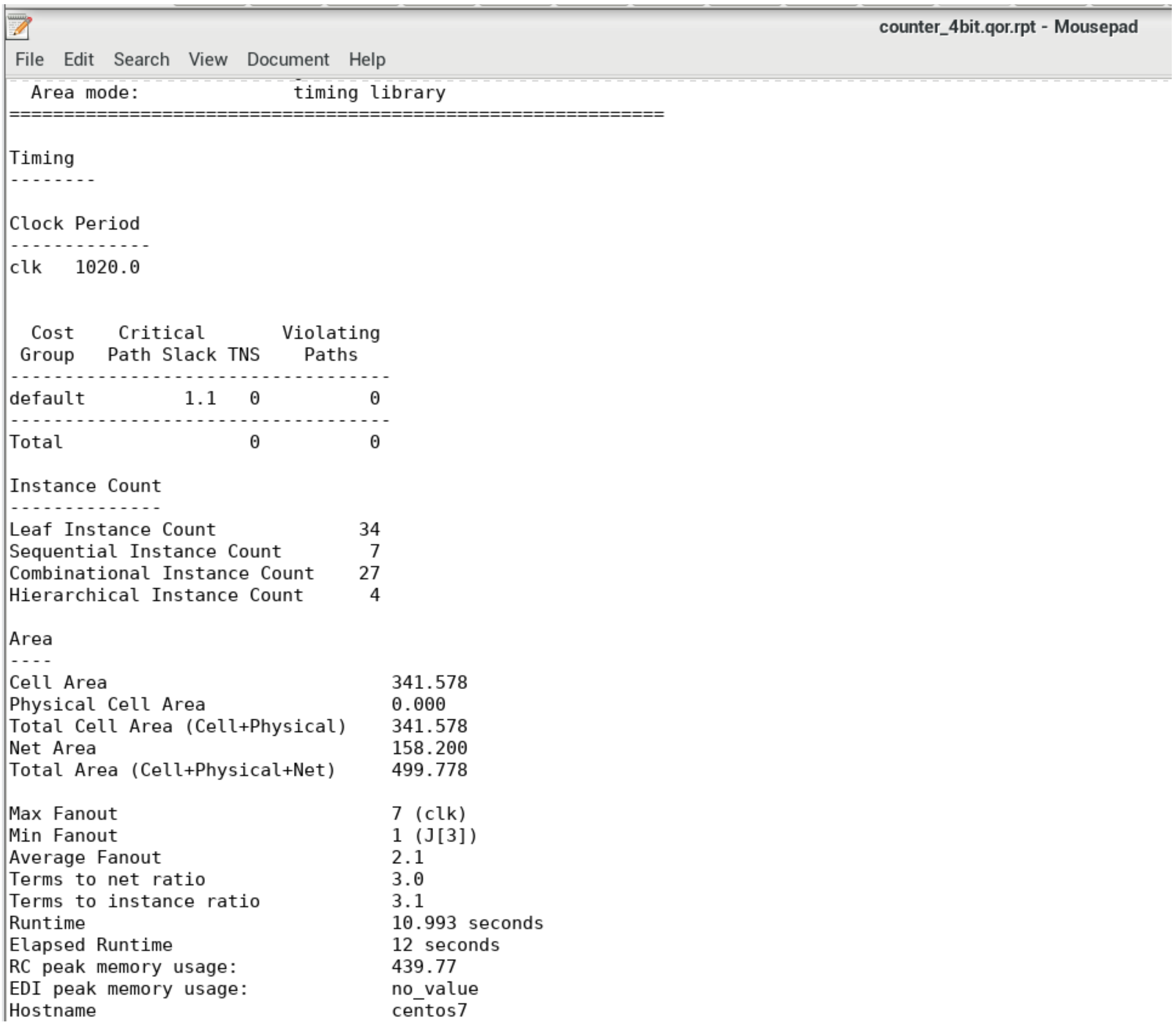
```
============================================================
  Generated by:           Genus(TM) Synthesis Solution GENUS15.20 - 15.20-p004_1
  Generated on:           May 24 2024  08:42:05 am
  Module:                 counter_4bit
  Interconnect mode:      global
  Area mode:              timing library
============================================================


Path 1: MET (1 ps) Setup Check with Pin ff2/Qn_reg/CLK->D
     Startpoint: (R) ff0/Q_reg/CLK
          Clock: (R) clk
       Endpoint: (F) ff2/Qn_reg/D
          Clock: (R) clk

                     Capture       Launch
        Clock Edge:+    1020            0
       Src Latency:+       0            0
       Net Latency:+       0 (I)        0 (I)
           Arrival:=    1020            0

             Setup:-     298
       Uncertainty:-      10
     Required Time:=     712
      Launch Clock:-       0
         Data Path:-     711
             Slack:=       1

#-----------------------------------------------------------------------------------------------
# Timing Point   Flags    Arc   Edge           Cell             Fanout Load Trans Delay Arrival
#                                                                       (fF)  (ps)  (ps)   (ps)
#-----------------------------------------------------------------------------------------------
  ff0/Q_reg/CLK  -       -      R     (arrival)                     7    -     0     -       0
  ff0/Q_reg/Q    -       CLK->Q F     sky130_fd_sc_hd__dfrtp_1      5 14.6    92   394     394
  g9/Y           -       A->Y   R     sky130_fd_sc_hd__clkinv_2     3 13.2    41    65     460
  g100/Y         -       A->Y   F     sky130_fd_sc_hd__nand2_1      1  5.4    49    59     518
  g22/Y          -       A2->Y  R     sky130_fd_sc_hd__o21ai_2      2  8.3   141   143     661
  ff2/g35/Y      -       A->Y   F     sky130_fd_sc_hd__inv_2        2  5.9    38    50     711
  ff2/Qn_reg/D   -       -      F     sky130_fd_sc_hd__sdfstp_1     2    -     -     0     711
#-----------------------------------------------------------------------------------------------
```

- We can also see other aspects of the synthesized design in other reports inside the 04_reports folder.



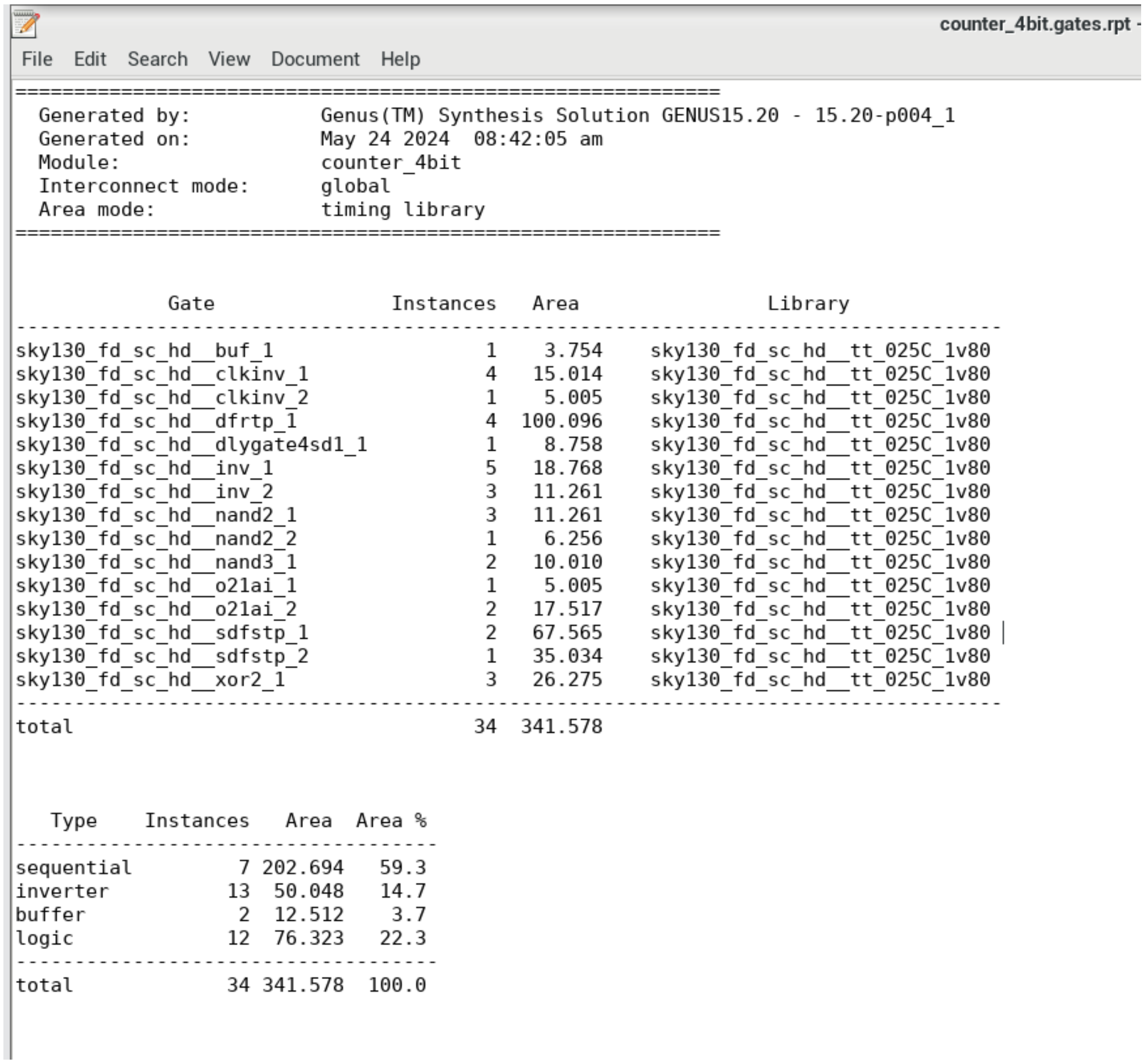
```
synthesis/04_reports
|-- counter_4bit.area.rpt
|-- counter_4bit.datapath.rpt
|-- counter_4bit.gates.rpt
|-- counter_4bit.hier.rpt
|-- counter_4bit.power.rpt
|-- counter_4bit.qor.rpt
`-- counter_4bit.timing.rpt
```

- Tổng diện tích thiết kế và tài nguyên cổng được sử dụng trong thiết kế:


counter_4bit.qor.rpt - Mousepad
File Edit Search View Document Help

```
  Area mode:              timing library
============================================================

Timing
--------

Clock Period
-------------
clk   1020.0


  Cost    Critical       Violating
 Group   Path Slack TNS    Paths
-----------------------------------
default         1.1   0          0
-----------------------------------
Total                 0          0

Instance Count
--------------
Leaf Instance Count             34
Sequential Instance Count        7
Combinational Instance Count    27
Hierarchical Instance Count      4

Area
----
Cell Area                          341.578
Physical Cell Area                 0.000
Total Cell Area (Cell+Physical)    341.578
Net Area                           158.200
Total Area (Cell+Physical+Net)     499.778

Max Fanout                         7 (clk)
Min Fanout                         1 (J[3])
Average Fanout                     2.1
Terms to net ratio                 3.0
Terms to instance ratio            3.1
Runtime                            10.993 seconds
Elapsed Runtime                    12 seconds
RC peak memory usage:              439.77
EDI peak memory usage:             no_value
Hostname                           centos7
```

Quality of Results report (QoR)



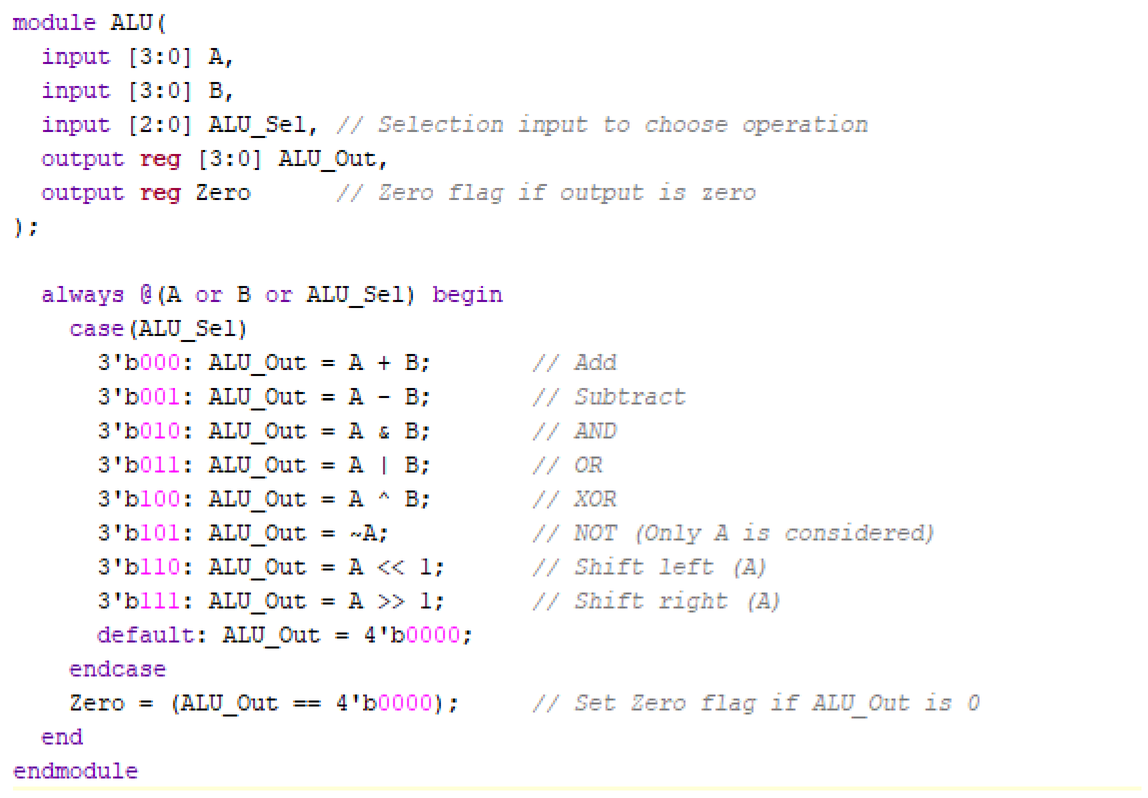
```
============================================================
  Generated by:           Genus(TM) Synthesis Solution GENUS15.20 - 15.20-p004_1
  Generated on:           May 24 2024  08:42:05 am
  Module:                 counter_4bit
  Interconnect mode:      global
  Area mode:              timing library
============================================================


             Gate              Instances   Area             Library
-------------------------------------------------------------------------------------
sky130_fd_sc_hd__buf_1                 1    3.754    sky130_fd_sc_hd__tt_025C_1v80
sky130_fd_sc_hd__clkinv_1              4   15.014    sky130_fd_sc_hd__tt_025C_1v80
sky130_fd_sc_hd__clkinv_2              1    5.005    sky130_fd_sc_hd__tt_025C_1v80
sky130_fd_sc_hd__dfrtp_1               4  100.096    sky130_fd_sc_hd__tt_025C_1v80
sky130_fd_sc_hd__dlygate4sd1_1         1    8.758    sky130_fd_sc_hd__tt_025C_1v80
sky130_fd_sc_hd__inv_1                 5   18.768    sky130_fd_sc_hd__tt_025C_1v80
sky130_fd_sc_hd__inv_2                 3   11.261    sky130_fd_sc_hd__tt_025C_1v80
sky130_fd_sc_hd__nand2_1               3   11.261    sky130_fd_sc_hd__tt_025C_1v80
sky130_fd_sc_hd__nand2_2               1    6.256    sky130_fd_sc_hd__tt_025C_1v80
sky130_fd_sc_hd__nand3_1               2   10.010    sky130_fd_sc_hd__tt_025C_1v80
sky130_fd_sc_hd__o21ai_1               1    5.005    sky130_fd_sc_hd__tt_025C_1v80
sky130_fd_sc_hd__o21ai_2               2   17.517    sky130_fd_sc_hd__tt_025C_1v80
sky130_fd_sc_hd__sdfstp_1              2   67.565    sky130_fd_sc_hd__tt_025C_1v80
sky130_fd_sc_hd__sdfstp_2              1   35.034    sky130_fd_sc_hd__tt_025C_1v80
sky130_fd_sc_hd__xor2_1                3   26.275    sky130_fd_sc_hd__tt_025C_1v80
-------------------------------------------------------------------------------------
total                                 34  341.578


   Type    Instances    Area  Area %
------------------------------------
sequential         7 202.694   59.3
inverter          13  50.048   14.7
buffer             2  12.512    3.7
logic             12  76.323   22.3
------------------------------------
total             34 341.578  100.0
```

Gate usage of the design

II. VLSI FRONT-END FLOW

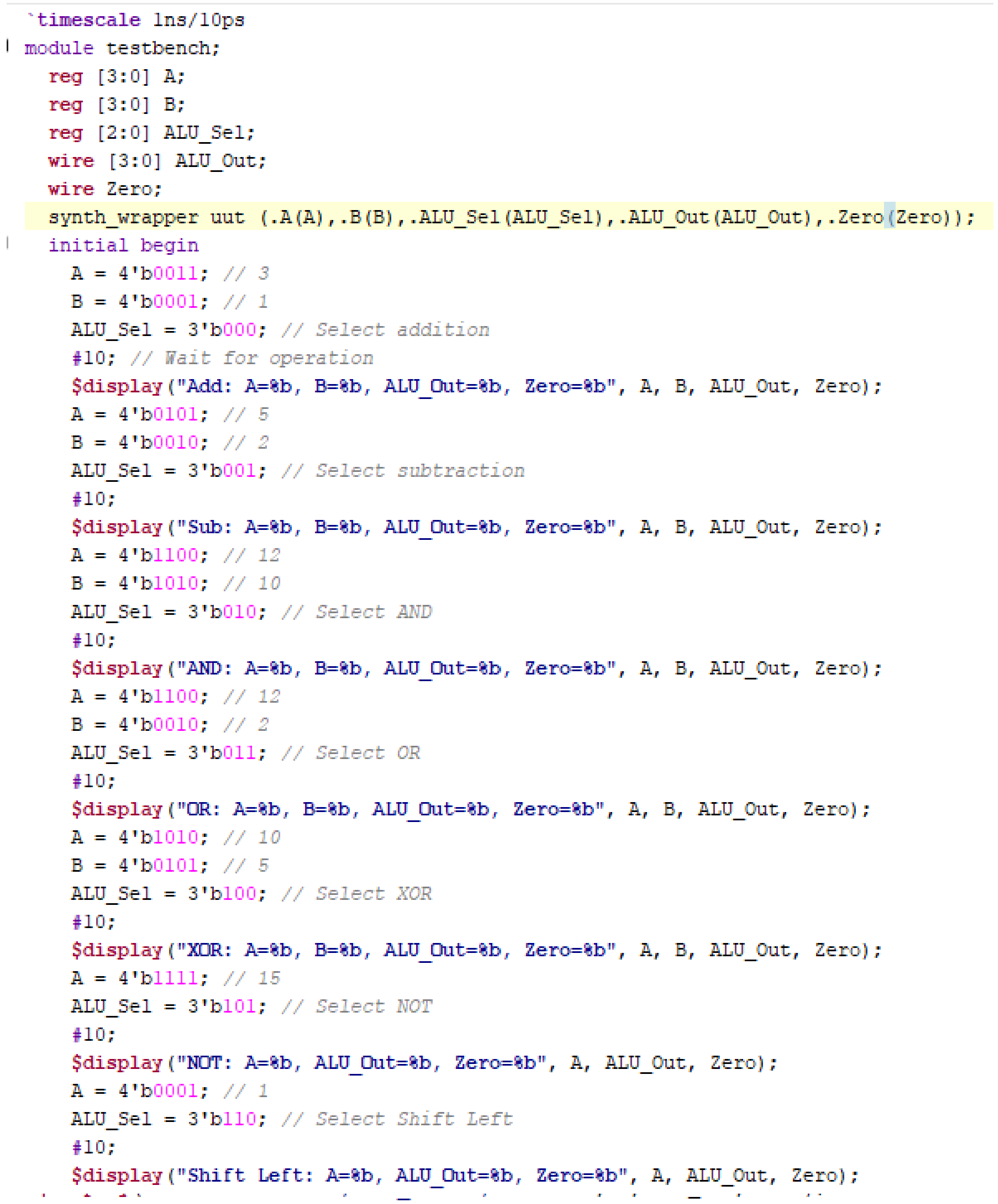
Lab: ALU

1. Code

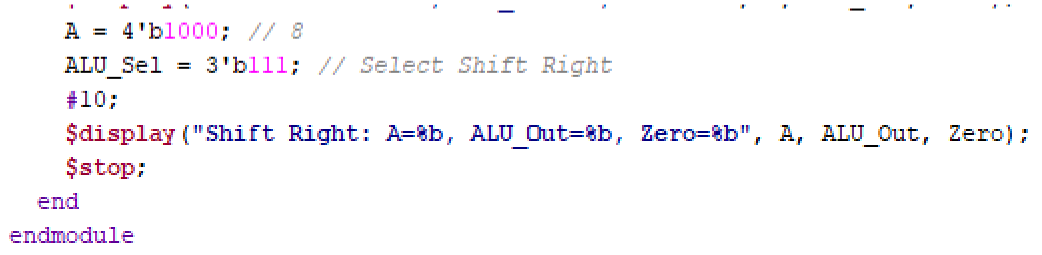


```
module ALU(
  input [3:0] A,
  input [3:0] B,
  input [2:0] ALU_Sel, // Selection input to choose operation
  output reg [3:0] ALU_Out,
  output reg Zero      // Zero flag if output is zero
);

  always @(A or B or ALU_Sel) begin
    case(ALU_Sel)
      3'b000: ALU_Out = A + B;        // Add
      3'b001: ALU_Out = A - B;        // Subtract
      3'b010: ALU_Out = A & B;        // AND
      3'b011: ALU_Out = A | B;        // OR
      3'b100: ALU_Out = A ^ B;        // XOR
      3'b101: ALU_Out = ~A;           // NOT (Only A is considered)
      3'b110: ALU_Out = A << 1;       // Shift left (A)
      3'b111: ALU_Out = A >> 1;       // Shift right (A)
      default: ALU_Out = 4'b0000;
    endcase
    Zero = (ALU_Out == 4'b0000);      // Set Zero flag if ALU_Out is 0
  end
endmodule
```

2. Code testbench



```
 `timescale 1ns/10ps
module testbench;
  reg [3:0] A;
  reg [3:0] B;
  reg [2:0] ALU_Sel;
  wire [3:0] ALU_Out;
  wire Zero;
  synth_wrapper uut (.A(A),.B(B),.ALU_Sel(ALU_Sel),.ALU_Out(ALU_Out),.Zero(Zero));
  initial begin
    A = 4'b0011; // 3
    B = 4'b0001; // 1
    ALU_Sel = 3'b000; // Select addition
    #10; // Wait for operation
    $display("Add: A=%b, B=%b, ALU_Out=%b, Zero=%b", A, B, ALU_Out, Zero);
    A = 4'b0101; // 5
    B = 4'b0010; // 2
    ALU_Sel = 3'b001; // Select subtraction
    #10;
    $display("Sub: A=%b, B=%b, ALU_Out=%b, Zero=%b", A, B, ALU_Out, Zero);
    A = 4'b1100; // 12
    B = 4'b1010; // 10
    ALU_Sel = 3'b010; // Select AND
    #10;
    $display("AND: A=%b, B=%b, ALU_Out=%b, Zero=%b", A, B, ALU_Out, Zero);
    A = 4'b1100; // 12
    B = 4'b0010; // 2
    ALU_Sel = 3'b011; // Select OR
    #10;
    $display("OR: A=%b, B=%b, ALU_Out=%b, Zero=%b", A, B, ALU_Out, Zero);
    A = 4'b1010; // 10
    B = 4'b0101; // 5
    ALU_Sel = 3'b100; // Select XOR
    #10;
    $display("XOR: A=%b, B=%b, ALU_Out=%b, Zero=%b", A, B, ALU_Out, Zero);
    A = 4'b1111; // 15
    ALU_Sel = 3'b101; // Select NOT
    #10;
    $display("NOT: A=%b, ALU_Out=%b, Zero=%b", A, ALU_Out, Zero);
    A = 4'b0001; // 1
    ALU_Sel = 3'b110; // Select Shift Left
    #10;
    $display("Shift Left: A=%b, ALU_Out=%b, Zero=%b", A, ALU_Out, Zero);
```



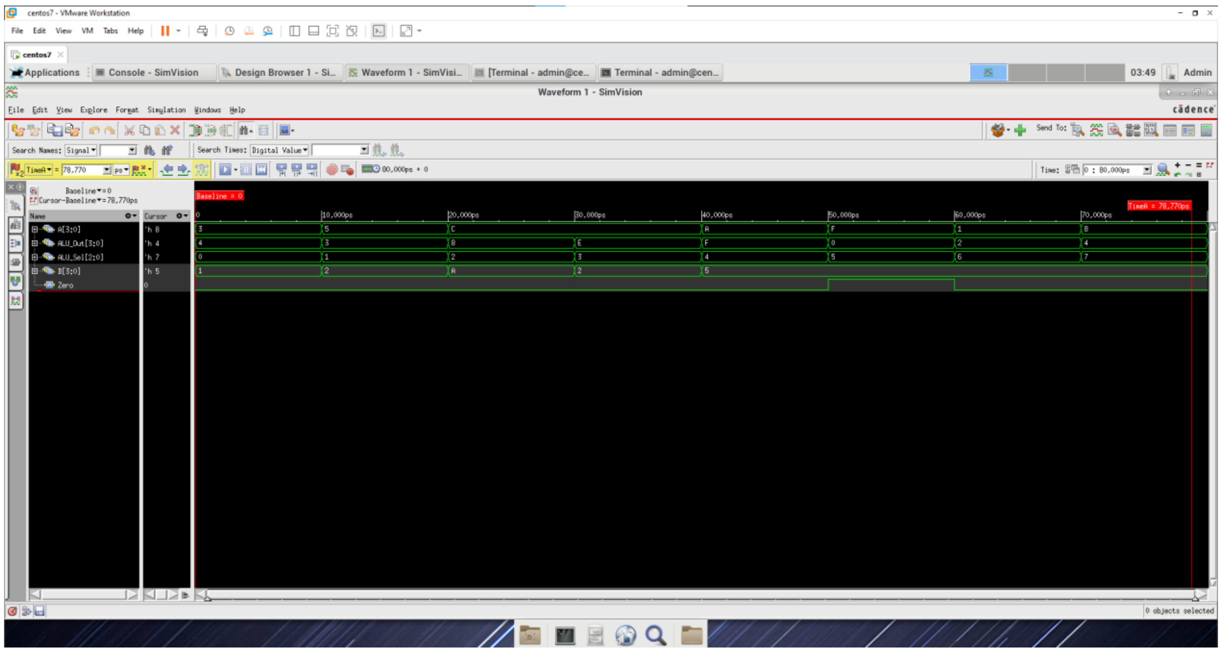
```
    A = 4'b1000; // 8
    ALU_Sel = 3'b111; // Select Shift Right
    #10;
    $display("Shift Right: A=%b, ALU_Out=%b, Zero=%b", A, ALU_Out, Zero);
    $stop;
  end
endmodule
```

3. Verification plan

Section	Item	Decription	Status
1	Reset	Khi rst_n = 0, thì giá trị ngõ ra và cờ carry = 0, khi rst_1 = 1 thì mạch hoạtđộng bình thường.	PASS
2	Max add	Khi thực hiện việc cộng 2 số 4 bit, thìgiá trị ngõ ra có thể bị tràn bit. Lúc bị tràn thì cờ carry sẽ lên 1.	PASS
3	Min sub	Khi thực hiện việc trừ 2 số 4 bit, thì giá trị ngõ ra có thể bị âm. Lúc giá trịâm thì cờ carry sẽ lên 1	PASS
4	And, or, xor, not	Thực hiện and, or, xor, not hai số 4 bitbất kì	PASS
5	Shift	Thực hiện shift right hoặc left hai số 4bit bất kì	PASS

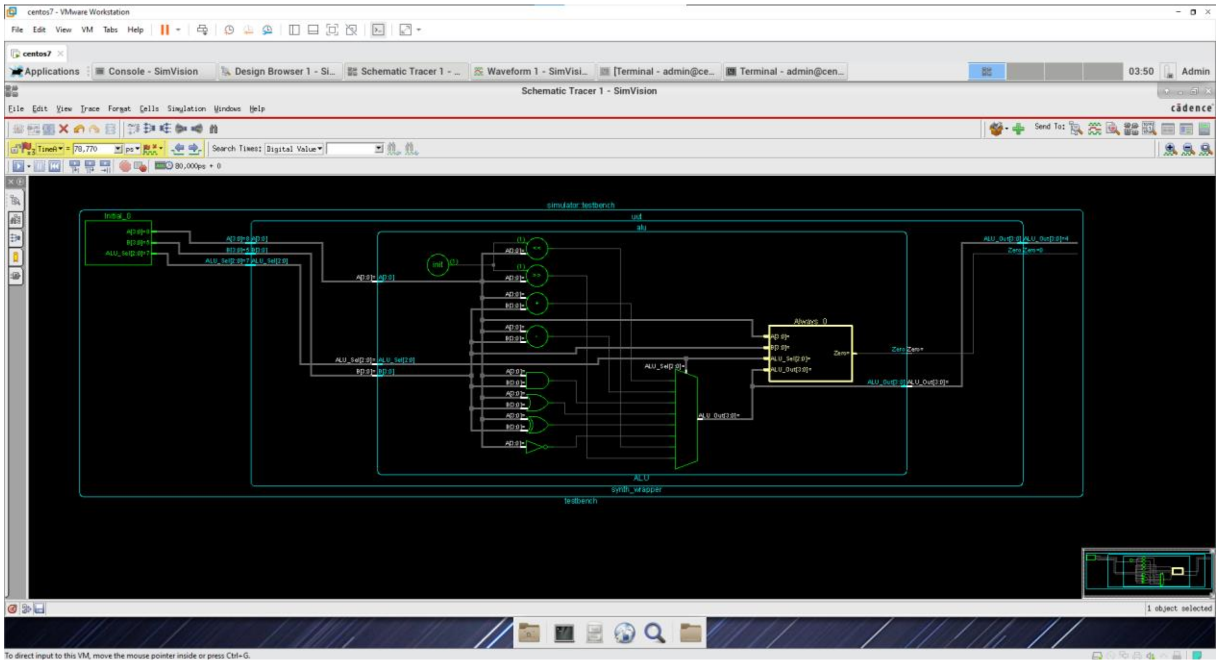
4. Simulation

Simulation (test add, sub, and, or, xor, not)



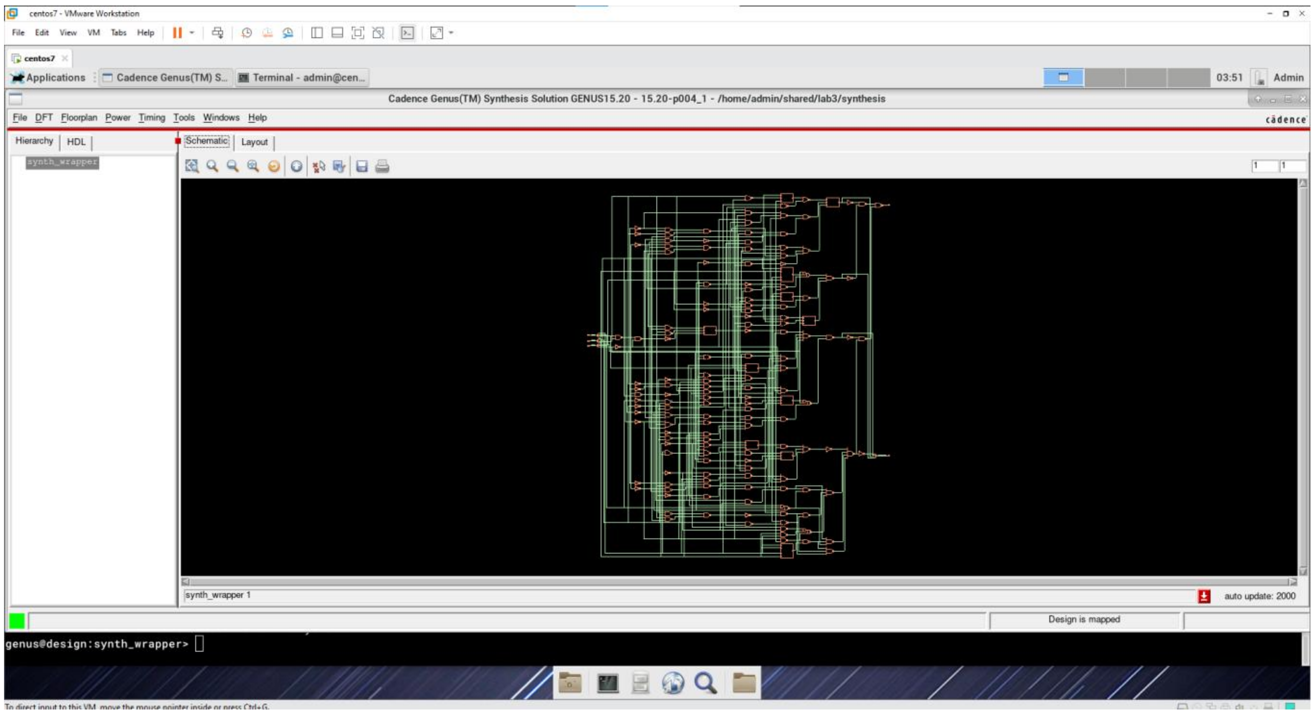
5. High level Arichitecture of the design

- Schematic của netlist



6. Report

- Schematic Viewer (main)


centos7 - VMware Workstation
File Edit View VM Tabs Help
centos7
Applications
Cadence Genus(TM) S...
Terminal - admin@cen...
03:51
Admin
Cadence Genus(TM) Synthesis Solution GENUS15.20 - 15.20-p004_1 - /home/admin/shared/lab3/synthesis
File DFT Floorplan Power Timing Tools Windows Help
cādence
Hierarchy
HDL
synth_wrapper
Schematic
Layout
synth_wrapper 1
auto update: 2000
Design is mapped

```
genus@design:synth_wrapper>
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.