

Assignment 4

Samaksh Gupta
2019200

Question 1

```
192.168.3.57
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
File 5 192.168.3.57
smaksh19200@edcserver1:~/Assy0/Outs$0$5 cd ..
smaksh19200@edcserver1:~/Assy$0$5 ls
fast.lib Ques2 Ques3a Ques3b Ques4 Ques5
smaksh19200@edcserver1:~/Assy$4 cd Ques2
smaksh19200@edcserver1:~/Assy4/Ques2$ ls
fast.lib Nogate15m_0Cl_worst_low_conditional_nldm.lib Q2_submod.v Q2.v synth.v yosys_script.tcl
smaksh19200@edcserver1:~/Assy4/Ques2$ 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 0.11+10 (git sha1 4871d0f19, clang 6.0.0-lubuntu2 -fPIC -Os)

yosys> 
```

Yosys Script

192.168.3.57 Terminal Sessions View X server Tools Games Settings Macros Help Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help Quick connect... 192.168.3.21 11:02 192.168.3.57

Reading the technology library specified in the given liberty file.
Read liberty lib_fest.lib
Read liberty lib_Nandate_15nm_OCL_fast_conditional_ccs.lib
Reading the conditional compilation file abc_constraint.sdc
Reading the conditional compilation file abc_constraint.sdc
Warning: In case of multiple abc files each file should be read individually
abc: In case of multiple abc files each file should be read individually
read_verilog MyRst.v
convert high level behavioral parts ("processes") to d-type flip-flops and muxes

perform some simple optimizations
opt
Resource Sharing
shares -force
show
convert high level memory constructs to d-type flip-flops and multiplexers
memory
opt
convert design to (logical) gate-level netlists
techmap
top.v
technology mapping of flip-flops
#Map internal flip-flop calls to the flip-flop cells in the technology library specified in the given liberty file.
#opt techmap -liberty Newgate15nm_OCL_fast_conditional_ccs.lib
#opt techmap -liberty Nandate15nm_OCL_fast_conditional_ccs.lib
opt
tse -o reports/top_timing.txt abc -D -constr constraint.sdc -liberty Nandate15nm_OCL_fast_conditional_ccs.lib -showtmp

use ABC to map remaining logic to cells from the library
abc -liberty Test.lib
abc -liberty Nandate15nm_OCL_fast_conditional_ccs.lib
opt
Print statistics
stat -liberty Test.lib
stat -liberty Nandate15nm_OCL_fast_conditional_ccs.lib
write synthesized design
write_verilog synth.v
create a graphviz DOT file
show

< >

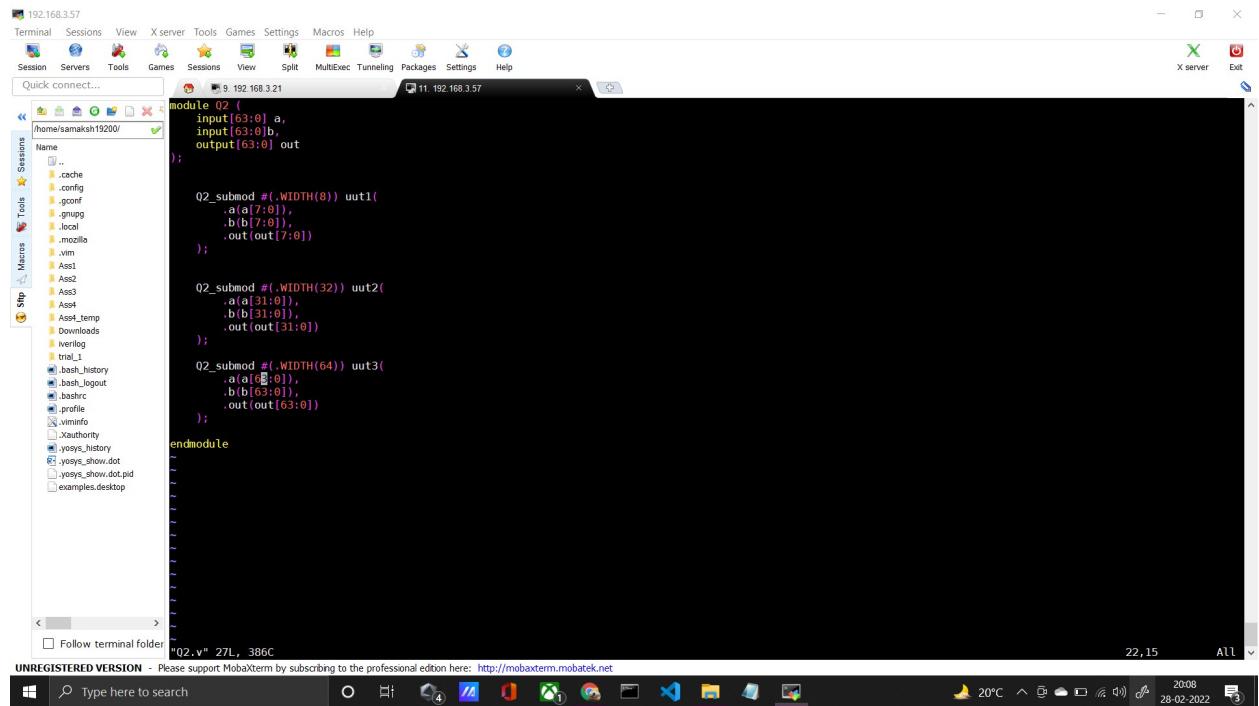
Follow terminal folder

UNREGISTERED VERSION Please support MobaXterm by subscribing to the professional edition here: <http://mobaxterm.mobatek.net>

36,0-1 All

(This is for Q4, minute changes were done for different parts)

Question-2



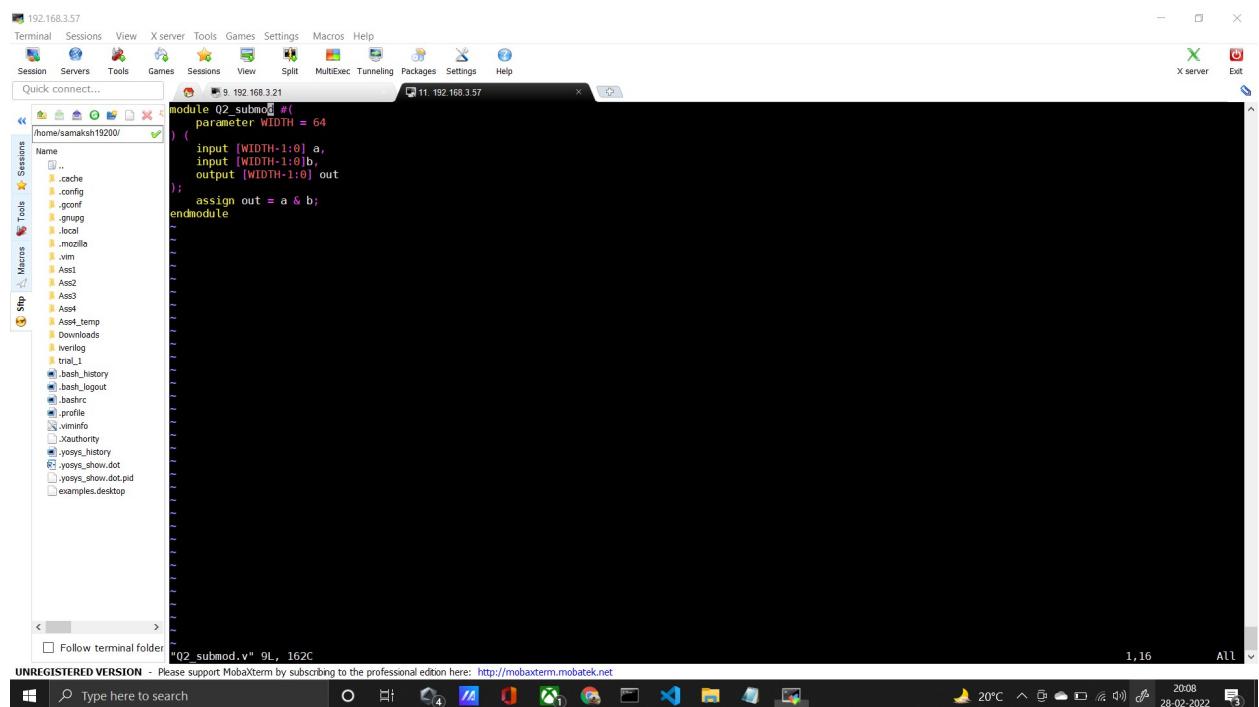
The screenshot shows the MobaXterm interface running on a Windows host. The terminal window displays Verilog code for a module named 02. The code defines three submodules: uut1, uut2, and uut3, each with specific input and output widths. The terminal window also shows the current file path as /home/samaksh19200/ and the command "02.v" at the bottom.

```
module 02 (
    input[63:0] a,
    input[63:0] b,
    output[63:0] out
);

    02_submod #(WIDTH(8)) uut1(
        .a(a[7:0]),
        .b(b[7:0]),
        .out(out[7:0])
    );

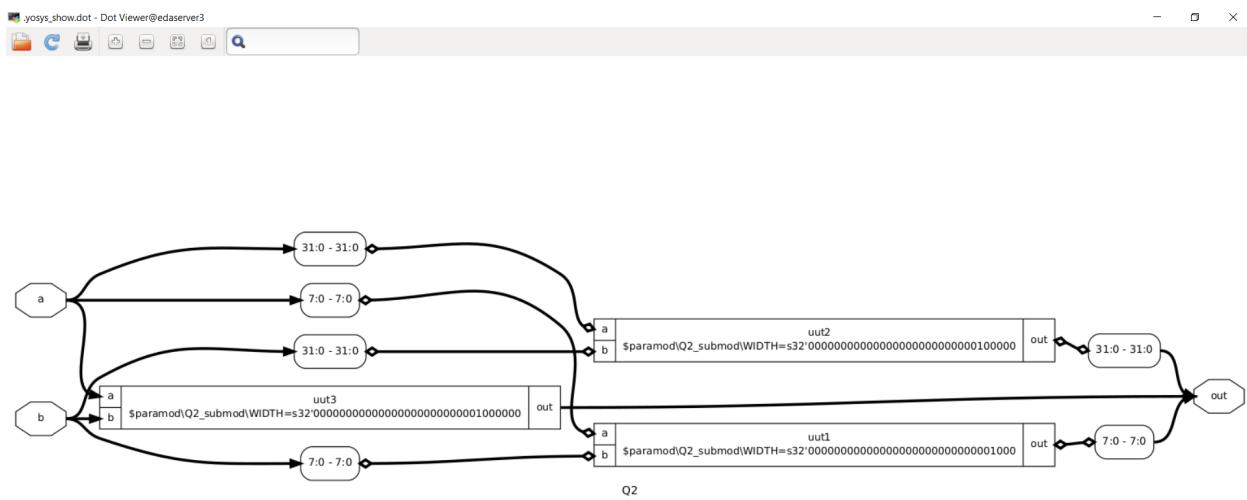
    02_submod #(WIDTH(32)) uut2(
        .a(a[31:0]),
        .b(b[31:0]),
        .out(out[31:0])
    );

    02_submod #(WIDTH(64)) uut3(
        .a(a[63:0]),
        .b(b[63:0]),
        .out(out[63:0])
    );
endmodule
```



The screenshot shows the MobaXterm interface running on a Windows host. The terminal window displays modified Verilog code for module 02_submod. The code includes a parameter WIDTH = 64 and an assign statement to set the output to the logical AND of inputs a and b. The terminal window also shows the current file path as /home/samaksh19200/ and the command "02_submod.v" at the bottom.

```
module 02_submod #(
    parameter WIDTH = 64
) (
    input [WIDTH-1:0] a,
    input [WIDTH-1:0] b,
    output [WIDTH-1:0] out
);
    assign out = a & b;
endmodule
```



Parameter ‘WIDTH’ is 64 by default. We are instantiating this module three times. In each instance, we are changing the value of WIDTH to some other values (8,32,64). Each instance calculates ‘and’ operation (bitwise) between the two operands.

The above image shows this instance being called (three times).

The netlist (At the end) instantiates ‘and’ logic to calculate the operation for the three instances in which the submodule is called. One and operation is for 1 bit... hence in total we need:-
 $64 + 32 + 8 = 104$ ‘and’ instances broken into 3 parts (as 3 instances are called).

Question-3 (a)

The screenshot shows a terminal window in MobaXterm with the IP address 192.168.3.21 and port 3.57. The window title is "module 03a". The code displayed is:

```
module 03a (
    input sel, input [3:0] a, input [3:0]b, input [3:0]x, input [3:0]y, output reg [7:0]z
);
    always @(*) begin
        if (sel == 1'b0) begin
            z = a + b;
        end
        else begin
            z = x * y;
        end
    end
endmodule
```

The terminal window also shows a file tree on the left and a status bar at the bottom.

I am using the share-force command in order to invoke optimization for resource sharing. Resource sharing can happen in the following way: We put MUX on Inputs rather than outputs. This way we only need 1-multiplier and 2 MUX, rather than the unshared design which will use 1 MUX on the output but calculate 2 multiplication results, thus using 2 multipliers in the design.

With resource sharing optimization, we have managed to reduce the area by almost two times the amount. The area statistics and the design schematic can be seen in the next two pages for both designs.

Without Optimization

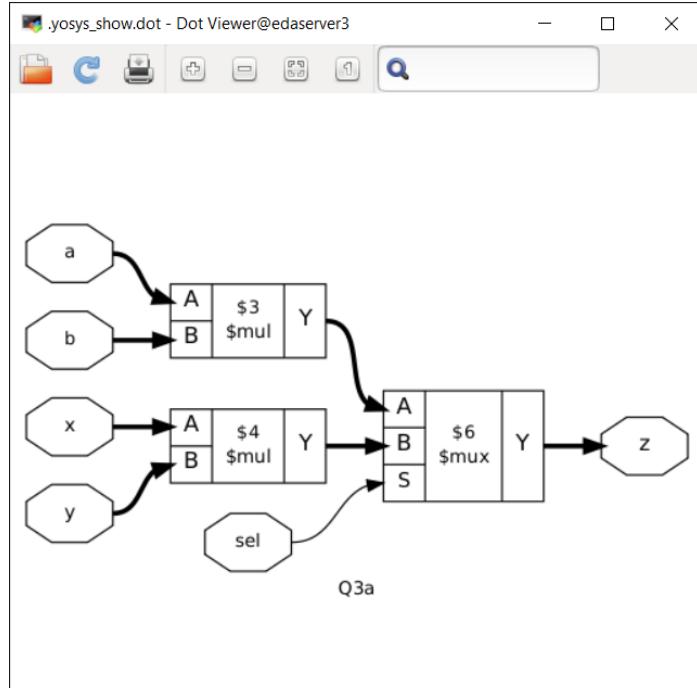
15. Printing statistics.

==== Q3a ===

```
Number of wires:          146
Number of wire bits:      165
Number of public wires:    6
Number of public wire bits: 25
Number of memories:       0
Number of memory bits:    0
Number of processes:      0
Number of cells:          148
  AND2_X1                  6
  AND3_X1                  3
  AND4_X1                  2
  AOI21_X1                 15
  AOI22_X1                 6
  INV_X1                   3
  NAND2_X1                 38
  NAND3_X1                 9
  NAND4_X1                 1
  NOR2_X1                  8
  NOR3_X1                  2
  OAI21_X1                 17
  OAI22_X1                 9
  OR2_X1                   3
  OR3_X1                   4
  XNOR2_X1                 15
  XOR2_X1                  7
```

Chip area for module '\Q3a': 43.696128

16. Executing Verilog backend.
Dumping module '\Q3a'.

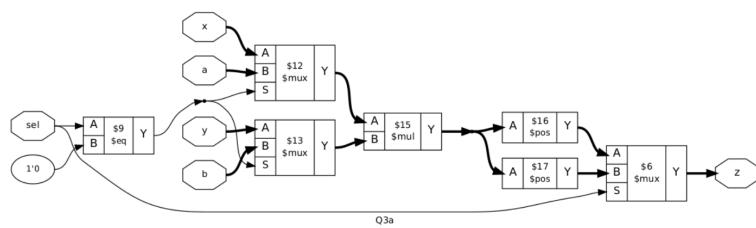
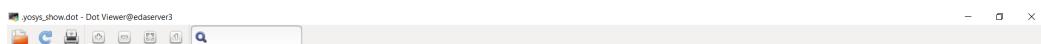


WITH OPTIMIZATION

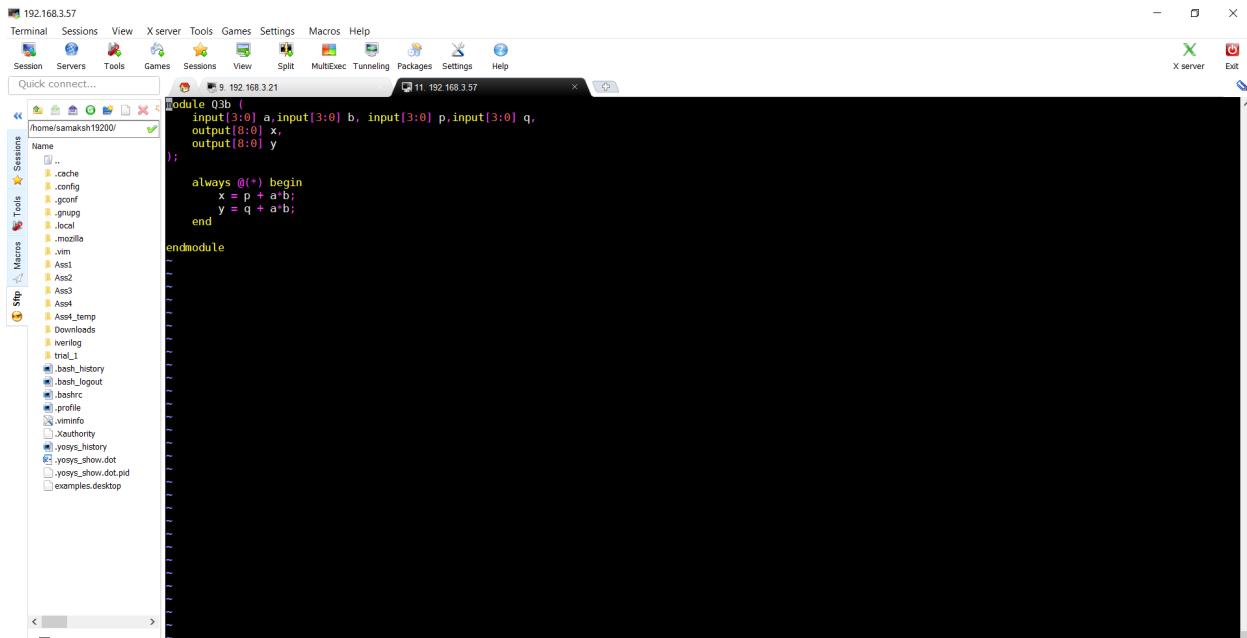
```
==== Q3a ====
Number of wires: 66
Number of wire bits: 85
Number of public wires: 6
Number of public wire bits: 25
Number of memories: 0
Number of memory bits: 0
Number of processes: 0
Number of cells: 68
AND2_X1 2
AND3_X1 3
AND4_X1 2
AOI21_X1 4
AOI22_X1 4
MUX2_X1 8
NAND2_X1 12
NAND3_X1 3
NAND4_X1 2
NOR2_X1 6
OAI21_X1 8
OR2_X1 1
OR3_X1 2
XNOR2_X1 7
XOR2_X1 4

Chip area for module '\Q3a': 23.740416

17. Executing Verilog backend.
Dumping module '\Q3a'.
```



Question 3-b)



The screenshot shows a terminal window titled "192.168.3.57" with the command "xterm" in the title bar. The window contains Verilog code for a multiplier module:

```
module Q3b (
    input[3:0] a, input[3:0] b, input[3:0] p, input[3:0] q,
    output[8:0] x,
    output[8:0] y
);
    always @(*) begin
        x = p + a*b;
        y = q + a*b;
    end
endmodule
```

The terminal window has a menu bar with "Terminal", "Sessions", "View", "Xserver", "Tools", "Games", "Settings", "Macros", "Help". The "Sessions" menu is open, showing a list of sessions including "home/samaksh19200" (selected), ".cache", ".config", ".gnupg", ".local", ".mozilla", ".vim", "Ass1", "Ass2", "Ass3", "Ass4", "Ass5_temp", "Downloads", "iTerm2", "Ass1_1", ".bash_history", ".bash_logout", ".bashrc", ".profile", ".vminfo", ".Xauthority", ".yesys_history", ".yesys_show_dot", ".yesys_show_dot.pid", and "examples.desktop".

Here, we can again remove 1 multiplier by saving the value of ' $a*b$ ' as some constant already. Since, this ' $a*b$ ' is common in both the outputs, we only need to calculate it ones. This optimization is achieved and the area reduces for the optimized design.

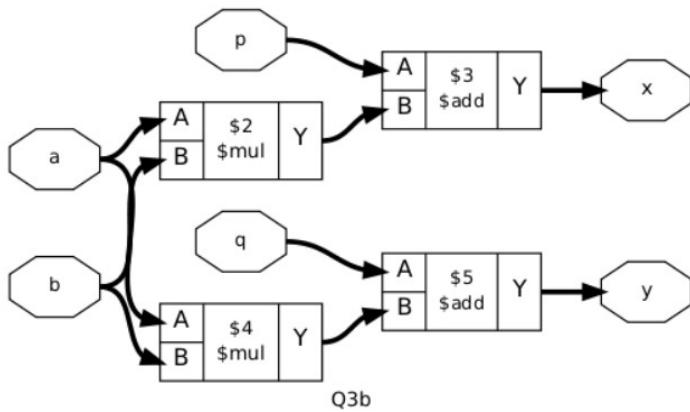
Without Optimization

```
== Q3b ==

Number of wires: 106
Number of wire bits: 134
Number of public wires: 6
Number of public wire bits: 34
Number of memories: 0
Number of memory bits: 0
Number of processes: 0
Number of cells: 116
AND2_X1 1
AND3_X1 5
AND4_X1 3
AOI21_X1 8
AOI22_X1 3
INV_X1 5
NAND2_X1 20
NAND3_X1 6
NOR2_X1 9
NOR3_X1 5
OAI21_X1 15
OAI22_X1 2
OR2_X1 1
OR3_X1 5
OR4_X1 2
XNOR2_X1 20
XOR2_X1 6

Chip area for module '\Q3b': 36.421632

16. Executing Verilog backend.
Dumping module '\Q3b'.
```



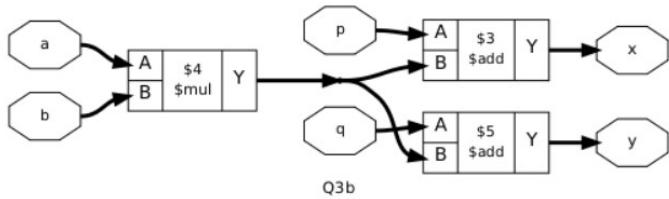
With Optimization

```
== Q3b ==

Number of wires:          100
Number of wire bits:      128
Number of public wires:    6
Number of public wire bits: 34
Number of memories:       0
Number of memory bits:    0
Number of processes:      0
Number of cells:          110
  AND2_X1                  3
  AND3_X1                  4
  AND4_X1                  2
  A0I2I_X1                 15
  A0I22_X1                 4
  INV_X1                   1
  NAND2_X1                 15
  NAND3_X1                 10
  NOR2_X1                  10
  NOR3_X1                  3
  OAI2I_X1                 8
  OR2_X1                   3
  OR3_X1                   4
  XNOR2_X1                 15
  XOR2_X1                  13

Chip area for module '\Q3b': 35.241984

17. Executing Verilog backend.
Dumping module '\Q3b'.
```



Question-4)

```

Module MyFSM(in,clk,rst,out);
Input in,clk,rst;
Output out;
reg [2:0] cs; //state elements of the FSM
parameter S0=3'b000,S1=3'b010,S2=3'b100;
begin
    assign out = cs;
end
always @(*) begin
    case (ns)
        S0: if (in==1'b1) ns=S1;
              else ns=S0;
        S1: if (in==1'b0) ns=S2;
              else ns=S1;
        S2: if (in==1'b0) ns=S0;
              else ns=S2;
    endcase
end
always @(*) begin
    case (cs)
        S0: out=1'b0;
        S1: out=1'b0;
        S2: if (in==1'b0) out=1'b1;
              else out=1'b0;
    endcase
end
endmodule

```

UNREGISTERED VERSION Please support NoobsTerm by subscribe to the professional version. <http://noobsterm.mobi/share>

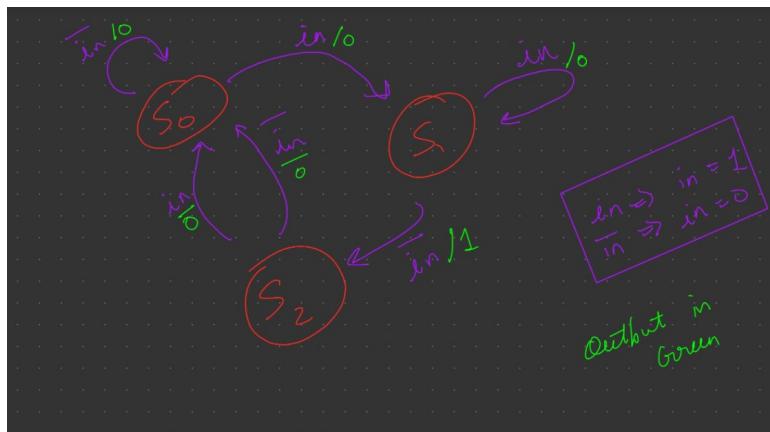
Constraints

```

set_driving_cell CLKBUF
set_load 100

```

FSM TABLE (Mealy Machine)



The FSM is working on the following transition logic and at the positive edge, the current state is assigned as the next state which is decided as per the inputs.

16. Printing statistics.

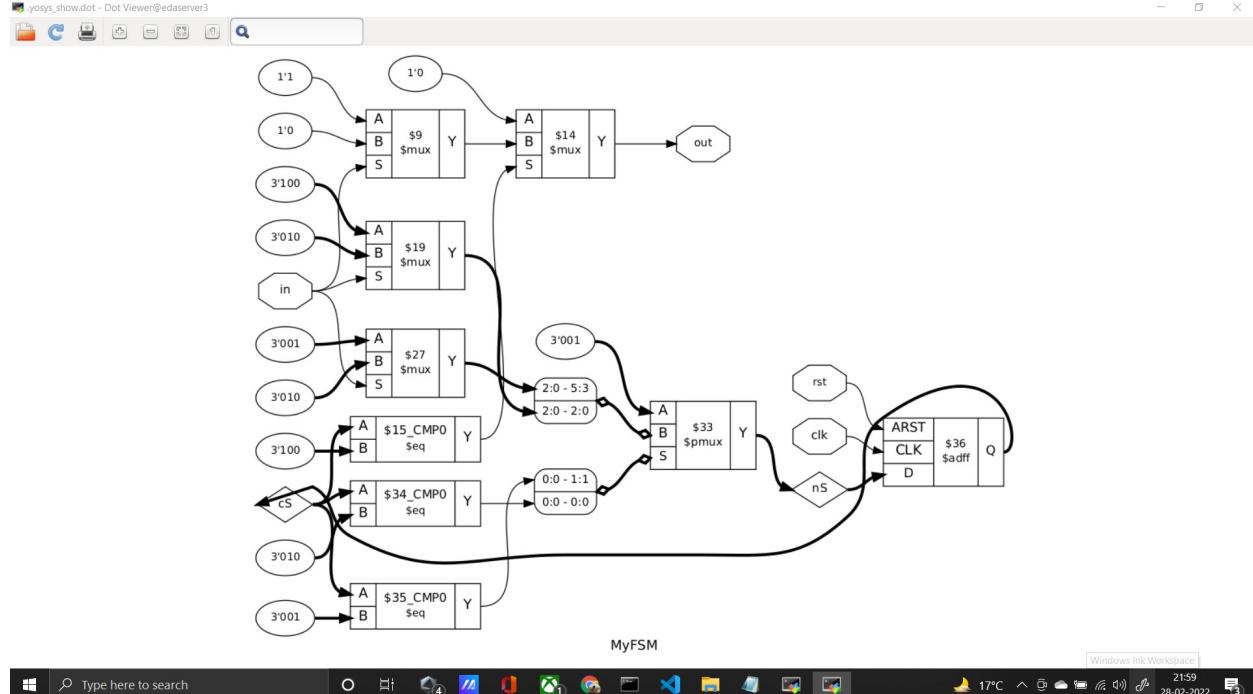
== MyFSM ==

Number of wires:	18
Number of wire bits:	22
Number of public wires:	6
Number of public wire bits:	10
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	19
AND2_X2	1
AND4_X1	1
BUF_X12	2
BUF_X16	1
BUF_X8	2
DFFRNQ_X1	2
DFFSNQ_X1	1
INV_X1	2
INV_X16	1
INV_X2	1
NOR2_X2	1
NOR3_X2	1
NOR4_X1	1
NOR4_X2	1
OR2_X2	1

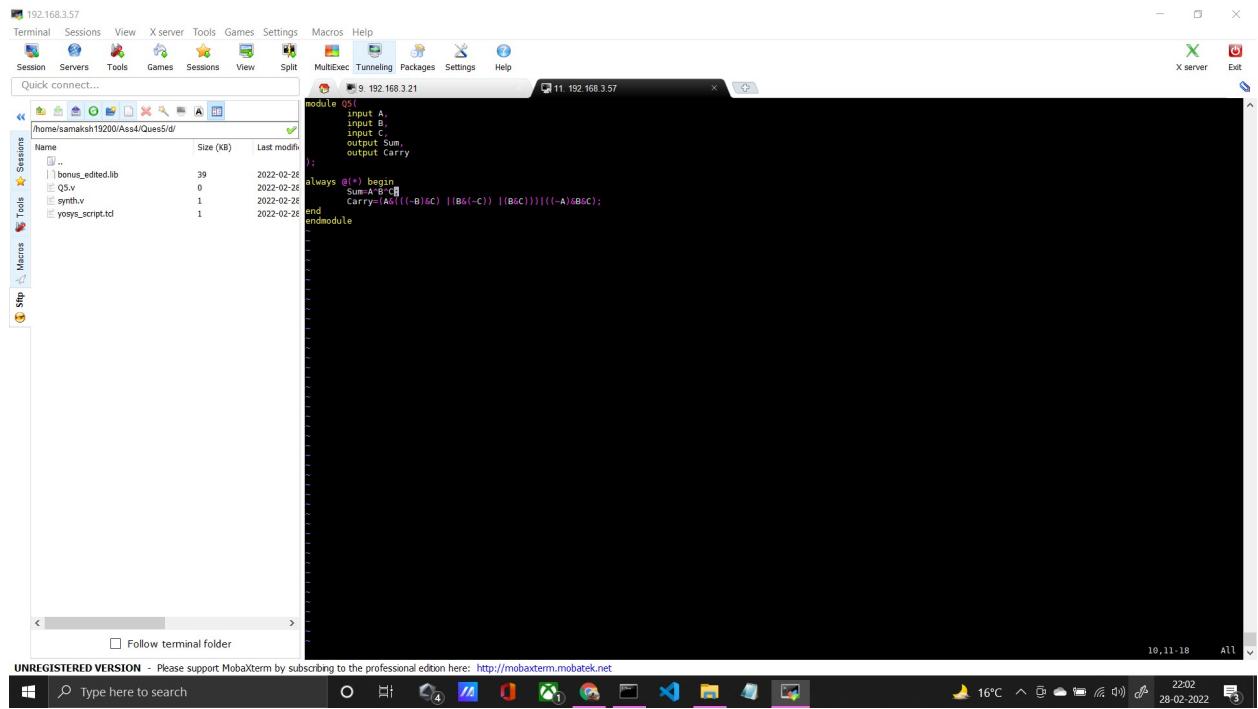
Chip area for module '\MyFSM': 12.582912

17. Executing Verilog backend.

Dumping module '\MyFSM'.



Question-5)



The screenshot shows a terminal window in MobaXterm connected to IP 192.168.3.57. The window title is "9. 192.168.3.57". The terminal displays the following Verilog code:

```
module QS1
    input A;
    input B;
    input C;
    output Sum;
    output Carry;
endmodule

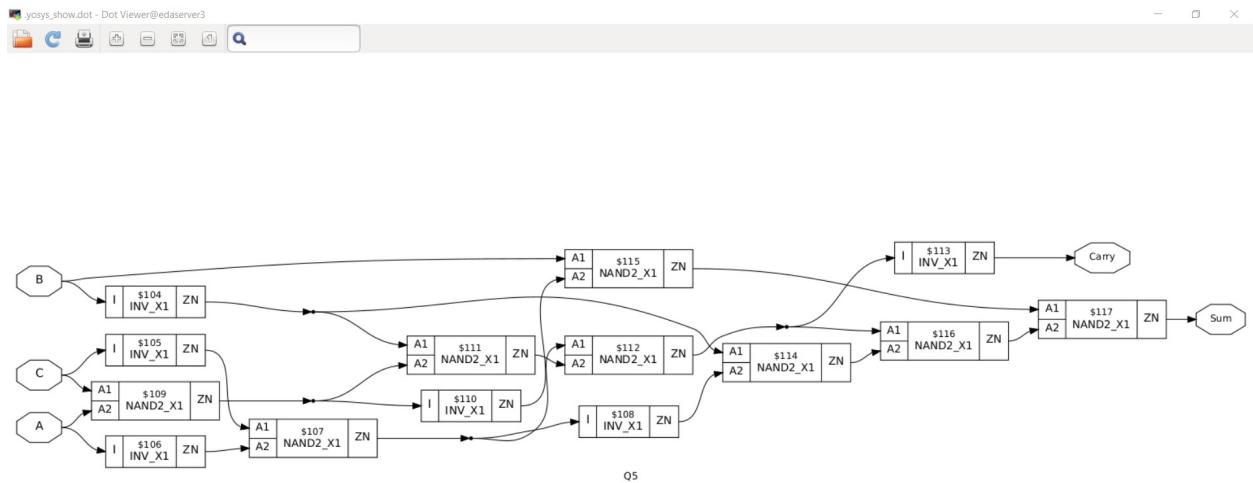
always @(*) begin
    Sum=A'B'C;
    Carry=(A&((~B)&C)) | (B&(~C)) | ((B&C)) | ((~A)&B&C);
end
```

The terminal interface includes a file browser on the left showing files in "/home/samaksh19200/Ass4/Ques5/d/", and a status bar at the bottom indicating "UNREGISTERED VERSION - Please support MobaTerm by subscribing to the professional edition here: <http://mobaterm.mobatek.net>". The taskbar at the bottom of the screen shows various application icons.

The following code for sum and carry will be run for 4 parts as per the given constraints of elements in the libraries. Please find them in the following pages (Schematic and Statistic Report).

An interesting thing to notice was that the Area in part (a and b) was the same, and for parts (c and d) it was the same. I think, that it doesn't take into constraint the sizing issues to skew the gates. Since we are talking at the top level so maybe it only counts the number of PMOS and NMOS and not the orientation which leads to size constraints. Since, both NOR and NAND consist of 2 NMOS and 2 PMOS, both have the same area. However, in part 'c' and 'd' since we have 'and' and 'or' gates, we need extra inverter and hence the area has increased in comparison to part (a and b).

A.



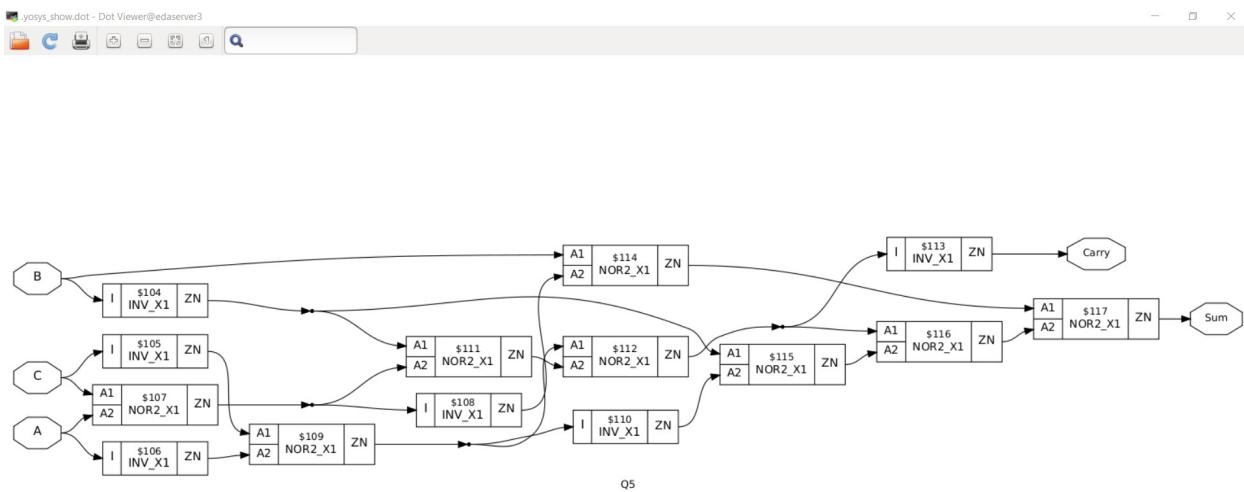
— 05 —

Number of wires:	17
Number of wire bits:	17
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:	14
INV_X1	6
NAND2_X1	2

chip area for module '05': 3.457600

15. Executing Verilog backend.
Dumping module `v05'.

B.



14. Printing statistics.

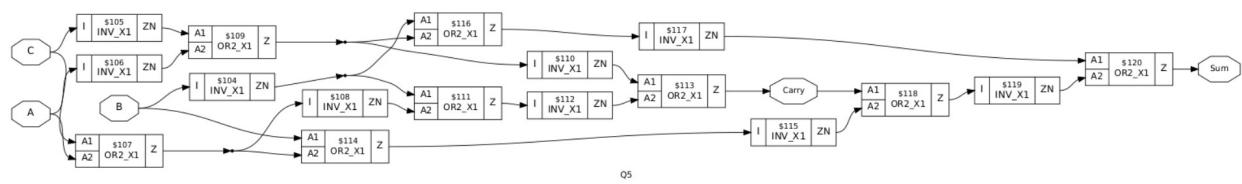
== Q5 ==

Number of wires:	17
Number of wire bits:	17
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:	14
INV_X1	6
NOR2_X1	8

Chip area for module '\Q5': 2.457600

15. Executing Verilog backend.
Dumping module `Q5'.

C.



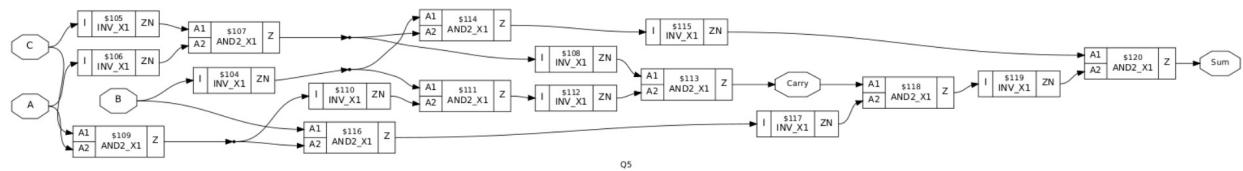
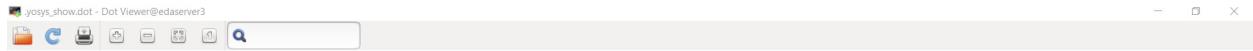
14. Printing statistics.

==== Q5 ===

Number of wires:	20
Number of wire bits:	20
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:	17
INV_X1	9
OR2_X1	8

Chip area for module '\Q5': 3.686400

D.



```
14. Printing statistics.

==== Q5 ===

Number of wires: 20
Number of wire bits: 20
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: 17
    AND2_X1 8
    INV_X1 9

Chip area for module '\Q5': 3.686400

15. Executing Verilog backend.
Dumping module '\Q5'.
```

NETLISTS

RTL files along with .lib files and constraints files are taken as input by the tool. Then this tool generates the netlist file. This netlist file shows the instantiations of various modules at the gate level. The .lib file contains the gates we are allowed to use. This fact is important in the last Question, where we are putting constraints on the type of Gates we can use in different parts. The constraint files lay down rules about various constraints (Area, delays etc). This .sdc file contains information about the bounding box of the chip, we shouldn't go past a finite area that is declared in this constraint file, or timing constraint (as changes in Question-4).

The flow of all netlists (of all questions) answers the same questions. They all break down each operation at the gate level and then implement the logic. As commented for Q2 (where images for Q2 were) that 'and' module was being called to perform 'and' bitwise operation for each unit under test called (Width= 8,32,64).

Similarly, for other questions too, we can see in the netlists that first the logic is broken into gates and these gates are simply being called. MUXs are directly called and at places(Q3a), we can see complex gates (AOI) being used.

In Question 5: Depending on the constraints of the .lib files, only those gates would be called in the netlist created.

In Question 4: We have designed an FSM (transition diagram attached before), this gives rise to numerous possibilities of it being implemented. I feel this is creating an instance of the FSM for different possibilities and depending on the state being reached and input received, it is calling that FSM instance which is in relation to that possibility. The netlist is quite difficult to understand as it is not only calling gates but some instances of the FSM module itself. I am guessing it is some sort of recursive relationship which differs as per different possibilities of the transition logic.

Question-2

```
/* Generated by Yosys 0.11+10 (git sha1 4871d8f19, clang 6.0.0-1ubuntu2
-fPIC -Os) */

(* dynports = 1 *)
(* hdlname = "\Q2_submod" *)
(* src = "Q2_submod.v:1.1-9.10" *)
module \$paramod\Q2_submod\WIDTH=s32'000000000000000000000000000000001000 (a,
b, out);
(* src = "Q2_submod.v:4.23-4.24" *)
input [7:0] a;
(* src = "Q2_submod.v:5.22-5.23" *)
input [7:0] b;
(* src = "Q2_submod.v:6.24-6.27" *)
output [7:0] out;
AND2_X1 _0_ (
    .A1(a[0]),
    .A2(b[0]),
    .Z(out[0])
);
AND2_X1 _1_ (
    .A1(a[1]),
    .A2(b[1]),
    .Z(out[1])
);
AND2_X1 _2_ (
    .A1(a[2]),
    .A2(b[2]),
    .Z(out[2])
);
AND2_X1 _3_ (
    .A1(a[3]),
    .A2(b[3]),
    .Z(out[3])
);
AND2_X1 _4_ (
    .A1(a[4]),
    .A2(b[4]),
    .Z(out[4])
);
```

```

) ;
AND2_X1 _5_ (
    .A1(a[5]),
    .A2(b[5]),
    .Z(out[5])
) ;
AND2_X1 _6_ (
    .A1(a[6]),
    .A2(b[6]),
    .Z(out[6])
) ;
AND2_X1 _7_ (
    .A1(a[7]),
    .A2(b[7]),
    .Z(out[7])
) ;
endmodule

(* dynports = 1 *)
(* hdlname = "\Q2_submod" *)
(* src = "Q2_submod.v:1.1-9.10" *)
module $paramod\Q2_submod\WIDTH=s32'00000000000000000000000000000000100000 (a,
b, out);
(* src = "Q2_submod.v:4.23-4.24" *)
input [31:0] a;
(* src = "Q2_submod.v:5.22-5.23" *)
input [31:0] b;
(* src = "Q2_submod.v:6.24-6.27" *)
output [31:0] out;
AND2_X1 _00_ (
    .A1(a[0]),
    .A2(b[0]),
    .Z(out[0])
) ;
AND2_X1 _01_ (
    .A1(a[1]),
    .A2(b[1]),
    .Z(out[1])
) ;
AND2_X1 _02_ (

```

```
.A1(a[2]),  
.A2(b[2]),  
.Z(out[2])  
);  
AND2_X1 _03_ (  
.A1(a[3]),  
.A2(b[3]),  
.Z(out[3])  
);  
AND2_X1 _04_ (  
.A1(a[4]),  
.A2(b[4]),  
.Z(out[4])  
);  
AND2_X1 _05_ (  
.A1(a[5]),  
.A2(b[5]),  
.Z(out[5])  
);  
AND2_X1 _06_ (  
.A1(a[6]),  
.A2(b[6]),  
.Z(out[6])  
);  
AND2_X1 _07_ (  
.A1(a[7]),  
.A2(b[7]),  
.Z(out[7])  
);  
AND2_X1 _08_ (  
.A1(a[8]),  
.A2(b[8]),  
.Z(out[8])  
);  
AND2_X1 _09_ (  
.A1(a[9]),  
.A2(b[9]),  
.Z(out[9])  
);  
AND2_X1 _10_ (  
);
```

```
.A1(a[10]),
.A2(b[10]),
.Z(out[10])
) ;
AND2_X1 _11_ (
.A1(a[11]),
.A2(b[11]),
.Z(out[11])
) ;
AND2_X1 _12_ (
.A1(a[12]),
.A2(b[12]),
.Z(out[12])
) ;
AND2_X1 _13_ (
.A1(a[13]),
.A2(b[13]),
.Z(out[13])
) ;
AND2_X1 _14_ (
.A1(a[14]),
.A2(b[14]),
.Z(out[14])
) ;
AND2_X1 _15_ (
.A1(a[15]),
.A2(b[15]),
.Z(out[15])
) ;
AND2_X1 _16_ (
.A1(a[16]),
.A2(b[16]),
.Z(out[16])
) ;
AND2_X1 _17_ (
.A1(a[17]),
.A2(b[17]),
.Z(out[17])
) ;
AND2_X1 _18_ (
```

```

    .A1(a[18]),
    .A2(b[18]),
    .Z(out[18])
) ;
AND2_X1 _19_ (
    .A1(a[19]),
    .A2(b[19]),
    .Z(out[19])
) ;
AND2_X1 _20_ (
    .A1(a[20]),
    .A2(b[20]),
    .Z(out[20])
) ;
AND2_X1 _21_ (
    .A1(a[21]),
    .A2(b[21]),
    .Z(out[21])
) ;
AND2_X1 _22_ (
    .A1(a[22]),
    .A2(b[22]),
    .Z(out[22])
) ;
AND2_X1 _23_ (
    .A1(a[23]),
    .A2(b[23]),
    .Z(out[23])
) ;
AND2_X1 _24_ (
    .A1(a[24]),
    .A2(b[24]),
    .Z(out[24])
) ;
AND2_X1 _25_ (
    .A1(a[25]),
    .A2(b[25]),
    .Z(out[25])
) ;
AND2_X1 _26_ (

```

```

    .A1(a[26]),
    .A2(b[26]),
    .Z(out[26])
);
AND2_X1 _27_ (
    .A1(a[27]),
    .A2(b[27]),
    .Z(out[27])
);
AND2_X1 _28_ (
    .A1(a[28]),
    .A2(b[28]),
    .Z(out[28])
);
AND2_X1 _29_ (
    .A1(a[29]),
    .A2(b[29]),
    .Z(out[29])
);
AND2_X1 _30_ (
    .A1(a[30]),
    .A2(b[30]),
    .Z(out[30])
);
AND2_X1 _31_ (
    .A1(a[31]),
    .A2(b[31]),
    .Z(out[31])
);
endmodule

(* dynports = 1 *)
(* hdlname = "\Q2_submod" *)
(* src = "Q2_submod.v:1.1-9.10" *)
module \$paramod\Q2_submod\WIDTH=s32'000000000000000000000000000000001000000 (a,
b, out);
(* src = "Q2_submod.v:4.23-4.24" *)
input [63:0] a;
(* src = "Q2_submod.v:5.22-5.23" *)
input [63:0] b;

```

```

(* src = "Q2_submod.v:6.24-6.27" *)
output [63:0] out;
AND2_X1 _00_ (
    .A1(a[22]),
    .A2(b[22]),
    .Z(out[22])
);
AND2_X1 _01_ (
    .A1(a[23]),
    .A2(b[23]),
    .Z(out[23])
);
AND2_X1 _02_ (
    .A1(a[24]),
    .A2(b[24]),
    .Z(out[24])
);
AND2_X1 _03_ (
    .A1(a[25]),
    .A2(b[25]),
    .Z(out[25])
);
AND2_X1 _04_ (
    .A1(a[26]),
    .A2(b[26]),
    .Z(out[26])
);
AND2_X1 _05_ (
    .A1(a[27]),
    .A2(b[27]),
    .Z(out[27])
);
AND2_X1 _06_ (
    .A1(a[28]),
    .A2(b[28]),
    .Z(out[28])
);
AND2_X1 _07_ (
    .A1(a[29]),
    .A2(b[29]),

```

```
.Z(out[29])
) ;
AND2_X1 _08_ (
.A1(a[30]),
.A2(b[30]),
.Z(out[30])
) ;
AND2_X1 _09_ (
.A1(a[31]),
.A2(b[31]),
.Z(out[31])
) ;
AND2_X1 _10_ (
.A1(a[32]),
.A2(b[32]),
.Z(out[32])
) ;
AND2_X1 _11_ (
.A1(a[33]),
.A2(b[33]),
.Z(out[33])
) ;
AND2_X1 _12_ (
.A1(a[34]),
.A2(b[34]),
.Z(out[34])
) ;
AND2_X1 _13_ (
.A1(a[35]),
.A2(b[35]),
.Z(out[35])
) ;
AND2_X1 _14_ (
.A1(a[36]),
.A2(b[36]),
.Z(out[36])
) ;
AND2_X1 _15_ (
.A1(a[37]),
.A2(b[37]),
```

```
.Z(out[37])
) ;
AND2_X1 _16_ (
.A1(a[38]),
.A2(b[38]),
.Z(out[38])
) ;
AND2_X1 _17_ (
.A1(a[39]),
.A2(b[39]),
.Z(out[39])
) ;
AND2_X1 _18_ (
.A1(a[40]),
.A2(b[40]),
.Z(out[40])
) ;
AND2_X1 _19_ (
.A1(a[41]),
.A2(b[41]),
.Z(out[41])
) ;
AND2_X1 _20_ (
.A1(a[42]),
.A2(b[42]),
.Z(out[42])
) ;
AND2_X1 _21_ (
.A1(a[43]),
.A2(b[43]),
.Z(out[43])
) ;
AND2_X1 _22_ (
.A1(a[44]),
.A2(b[44]),
.Z(out[44])
) ;
AND2_X1 _23_ (
.A1(a[45]),
.A2(b[45]),
```

```
.Z(out[45])
) ;
AND2_X1 _24_ (
.A1(a[46]),
.A2(b[46]),
.Z(out[46])
) ;
AND2_X1 _25_ (
.A1(a[47]),
.A2(b[47]),
.Z(out[47])
) ;
AND2_X1 _26_ (
.A1(a[48]),
.A2(b[48]),
.Z(out[48])
) ;
AND2_X1 _27_ (
.A1(a[49]),
.A2(b[49]),
.Z(out[49])
) ;
AND2_X1 _28_ (
.A1(a[50]),
.A2(b[50]),
.Z(out[50])
) ;
AND2_X1 _29_ (
.A1(a[51]),
.A2(b[51]),
.Z(out[51])
) ;
AND2_X1 _30_ (
.A1(a[52]),
.A2(b[52]),
.Z(out[52])
) ;
AND2_X1 _31_ (
.A1(a[53]),
.A2(b[53]),
```

```
.Z(out[53])
) ;
AND2_X1 _32_ (
.A1(a[54]),
.A2(b[54]),
.Z(out[54])
) ;
AND2_X1 _33_ (
.A1(a[55]),
.A2(b[55]),
.Z(out[55])
) ;
AND2_X1 _34_ (
.A1(a[56]),
.A2(b[56]),
.Z(out[56])
) ;
AND2_X1 _35_ (
.A1(a[57]),
.A2(b[57]),
.Z(out[57])
) ;
AND2_X1 _36_ (
.A1(a[58]),
.A2(b[58]),
.Z(out[58])
) ;
AND2_X1 _37_ (
.A1(a[59]),
.A2(b[59]),
.Z(out[59])
) ;
AND2_X1 _38_ (
.A1(a[60]),
.A2(b[60]),
.Z(out[60])
) ;
AND2_X1 _39_ (
.A1(a[61]),
.A2(b[61]),
```

```
.Z(out[61])
) ;
AND2_X1 _40_ (
.A1(a[62]),
.A2(b[62]),
.Z(out[62])
) ;
AND2_X1 _41_ (
.A1(a[63]),
.A2(b[63]),
.Z(out[63])
) ;
AND2_X1 _42_ (
.A1(a[0]),
.A2(b[0]),
.Z(out[0])
) ;
AND2_X1 _43_ (
.A1(a[1]),
.A2(b[1]),
.Z(out[1])
) ;
AND2_X1 _44_ (
.A1(a[2]),
.A2(b[2]),
.Z(out[2])
) ;
AND2_X1 _45_ (
.A1(a[3]),
.A2(b[3]),
.Z(out[3])
) ;
AND2_X1 _46_ (
.A1(a[4]),
.A2(b[4]),
.Z(out[4])
) ;
AND2_X1 _47_ (
.A1(a[5]),
.A2(b[5]),
.Z(out[5])
)
```

```
.Z(out[5])
) ;
AND2_X1 _48_ (
.A1(a[6]),
.A2(b[6]),
.Z(out[6])
) ;
AND2_X1 _49_ (
.A1(a[7]),
.A2(b[7]),
.Z(out[7])
) ;
AND2_X1 _50_ (
.A1(a[8]),
.A2(b[8]),
.Z(out[8])
) ;
AND2_X1 _51_ (
.A1(a[9]),
.A2(b[9]),
.Z(out[9])
) ;
AND2_X1 _52_ (
.A1(a[10]),
.A2(b[10]),
.Z(out[10])
) ;
AND2_X1 _53_ (
.A1(a[11]),
.A2(b[11]),
.Z(out[11])
) ;
AND2_X1 _54_ (
.A1(a[12]),
.A2(b[12]),
.Z(out[12])
) ;
AND2_X1 _55_ (
.A1(a[13]),
.A2(b[13]),
```

```
.Z(out[13])
) ;
AND2_X1 _56_ (
.A1(a[14]),
.A2(b[14]),
.Z(out[14])
) ;
AND2_X1 _57_ (
.A1(a[15]),
.A2(b[15]),
.Z(out[15])
) ;
AND2_X1 _58_ (
.A1(a[16]),
.A2(b[16]),
.Z(out[16])
) ;
AND2_X1 _59_ (
.A1(a[17]),
.A2(b[17]),
.Z(out[17])
) ;
AND2_X1 _60_ (
.A1(a[18]),
.A2(b[18]),
.Z(out[18])
) ;
AND2_X1 _61_ (
.A1(a[19]),
.A2(b[19]),
.Z(out[19])
) ;
AND2_X1 _62_ (
.A1(a[20]),
.A2(b[20]),
.Z(out[20])
) ;
AND2_X1 _63_ (
.A1(a[21]),
.A2(b[21]),
```

```

.Z(out[21])
);
endmodule

(* top = 1 *)
(* src = "Q2.v:1.1-27.10" *)
module Q2(a, b, out);
(* src = "Q2.v:2.17-2.18" *)
input [63:0] a;
(* src = "Q2.v:3.16-3.17" *)
input [63:0] b;
(* src = "Q2.v:4.18-4.21" *)
output [63:0] out;
(* src = "Q2.v:8.28-12.6" *)
\$paramod\Q2_submod\WIDTH=s32'000000000000000000000000000000001000 uut1 (
.a(a[7:0]),
.b(b[7:0]),
.out(out[7:0])
);
(* src = "Q2.v:15.29-19.6" *)
\$paramod\Q2_submod\WIDTH=s32'00000000000000000000000000000000100000 uut2 (
.a(a[31:0]),
.b(b[31:0]),
.out(out[31:0])
);
(* src = "Q2.v:21.29-25.6" *)
\$paramod\Q2_submod\WIDTH=s32'000000000000000000000000000000001000000 uut3 (
.a(a),
.b(b),
.out(out)
);
endmodule

```

QUESTION-3a

```
/* Generated by Yosys 0.11+10 (git sha1 4871d8f19, clang 6.0.0-1ubuntu2
-fPIC -Os) */
```

```
(* top = 1 *)
(* src = "Q3a.v:1.1-14.10" *)
module Q3a(sel, a, b, x, y, z);
    wire _000_;
    wire _001_;
    wire _002_;
    wire _003_;
    wire _004_;
    wire _005_;
    wire _006_;
    wire _007_;
    wire _008_;
    wire _009_;
    wire _010_;
    wire _011_;
    wire _012_;
    wire _013_;
    wire _014_;
    wire _015_;
    wire _016_;
    wire _017_;
    wire _018_;
    wire _019_;
    wire _020_;
    wire _021_;
    wire _022_;
    wire _023_;
    wire _024_;
    wire _025_;
    wire _026_;
    wire _027_;
    wire _028_;
    wire _029_;
    wire _030_;
    wire _031_;
    wire _032_;
    wire _033_;
    wire _034_;
    wire _035_;
```

```

wire _036_;
wire _037_;
wire _038_;
wire _039_;
wire _040_;
wire _041_;
wire _042_;
wire _043_;
wire _044_;
wire _045_;
wire _046_;
wire _047_;
wire _048_;
wire _049_;
wire _050_;
wire _051_;
wire _052_;
wire _053_;
wire _054_;
wire _055_;
wire _056_;
wire _057_;
wire _058_;
wire _059_;

(* src = "Q3a.v:2.27-2.28" *)
input [3:0] a;
(* src = "Q3a.v:2.40-2.41" *)
input [3:0] b;
(* src = "Q3a.v:2.11-2.14" *)
input sel;
(* src = "Q3a.v:2.53-2.54" *)
input [3:0] x;
(* src = "Q3a.v:2.67-2.68" *)
input [3:0] y;
(* src = "Q3a.v:2.86-2.87" *)
output [7:0] z;
MUX2_X1_060_(
    .I0(a[0]),
    .I1(x[0]),
    .S(sel),

```

```
.Z(_000_)
);
MUX2_X1_061(
.I0(b[0]),
.I1(y[0]),
.S(sel),
.Z(_001_)
);
AND2_X1_062(
.A1(_000_),
.A2(_001_),
.Z(z[0])
);
MUX2_X1_063(
.I0(b[1]),
.I1(y[1]),
.S(sel),
.Z(_002_)
);
MUX2_X1_064(
.I0(a[1]),
.I1(x[1]),
.S(sel),
.Z(_003_)
);
AND3_X1_065(
.A1(z[0]),
.A2(_002_),
.A3(_003_),
.Z(_004_)
);
AOI22_X1_066(
.A1(_000_),
.A2(_002_),
.B1(_003_),
.B2(_001_),
.ZN(_005_)
);
NOR2_X1_067(
.A1(_004_),

```

```
.A2(_005_),  
.ZN(z[1])  
);  
MUX2_X1_068_(  
.I0(b[2]),  
.I1(y[2]),  
.S(sel),  
.Z(_006_)  
);  
NAND2_X1_069_(  
.A1(_000_),  
.A2(_006_),  
.ZN(_007_)  
);  
MUX2_X1_070_(  
.I0(a[2]),  
.I1(x[2]),  
.S(sel),  
.Z(_008_)  
);  
NAND2_X1_071_(  
.A1(_002_),  
.A2(_008_),  
.ZN(_009_)  
);  
AND4_X1_072_(  
.A1(_001_),  
.A2(_002_),  
.A3(_003_),  
.A4(_008_),  
.Z(_010_)  
);  
NAND4_X1_073_(  
.A1(_001_),  
.A2(_002_),  
.A3(_003_),  
.A4(_008_),  
.ZN(_011_)  
);  
AOI22_X1_074_(
```

```
.A1(_002_),  
.A2(_003_),  
.B1(_008_),  
.B2(_001_),  
.ZN(_012_)  
);  
NOR2_X1_075_(  
.A1(_010_),  
.A2(_012_),  
.ZN(_013_)  
);  
XNOR2_X1_076_(  
.A1(_007_),  
.A2(_013_),  
.ZN(_014_)  
);  
AND2_X1_077_(  
.A1(_004_),  
.A2(_014_),  
.Z(_015_)  
);  
NOR2_X1_078_(  
.A1(_004_),  
.A2(_014_),  
.ZN(_016_)  
);  
NOR2_X1_079_(  
.A1(_015_),  
.A2(_016_),  
.ZN(z[2])  
);  
MUX2_X1_080_(  
.I0(b[3]),  
.I1(y[3]),  
.S(sel),  
.Z(_017_)  
);  
NAND2_X1_081_(  
.A1(_000_),  
.A2(_017_),
```

```
.ZN(_018_)
);
OAI21_X1_082_(
.A1(_007_),
.A2(_012_),
.B(_011_),
.ZN(_019_)
);
NAND2_X1_083_(
.A1(_003_),
.A2(_006_),
.ZN(_020_)
);
MUX2_X1_084_(
.I0(a[3]),
.I1(x[3]),
.S(sel),
.Z(_021_)
);
NAND2_X1_085_(
.A1(_002_),
.A2(_021_),
.ZN(_022_)
);
AND4_X1_086_(
.A1(_001_),
.A2(_002_),
.A3(_008_),
.A4(_021_),
.Z(_023_)
);
NAND4_X1_087_(
.A1(_001_),
.A2(_002_),
.A3(_008_),
.A4(_021_),
.ZN(_024_)
);
AOI22_X1_088_(
.A1(_002_),
```

```
.A2(008) ,
.B1(021) ,
.B2(001) ,
.ZN(025)
);
OR3_X1_089_(
.A1(020) ,
.A2(023) ,
.A3(025) ,
.Z(026)
);
OAI21_X1_090_(
.A1(023) ,
.A2(025) ,
.B(020) ,
.ZN(027)
);
AND3_X1_091_(
.A1(019) ,
.A2(026) ,
.A3(027) ,
.Z(028)
);
NAND3_X1_092_(
.A1(019) ,
.A2(026) ,
.A3(027) ,
.ZN(029)
);
AOI21_X1_093_(
.A1(026) ,
.A2(027) ,
.B(019) ,
.ZN(030)
);
OR3_X1_094_(
.A1(018) ,
.A2(028) ,
.A3(030) ,
.Z(031)
);
```

```
) ;  
OAI21_X1_095_(  
    .A1(_028_),  
    .A2(_030_),  
    .B(_018_),  
    .ZN(_032_)  
) ;  
AND3_X1_096_(  
    .A1(_015_),  
    .A2(_031_),  
    .A3(_032_),  
    .Z(_033_)  
) ;  
NAND3_X1_097_(  
    .A1(_015_),  
    .A2(_031_),  
    .A3(_032_),  
    .ZN(_034_)  
) ;  
OAI21_X1_098_(  
    .A1(_018_),  
    .A2(_030_),  
    .B(_029_),  
    .ZN(_035_)  
) ;  
NAND2_X1_099_(  
    .A1(_003_),  
    .A2(_017_),  
    .ZN(_036_)  
) ;  
OAI21_X1_100_(  
    .A1(_020_),  
    .A2(_025_),  
    .B(_024_),  
    .ZN(_037_)  
) ;  
NAND2_X1_101_(  
    .A1(_006_),  
    .A2(_008_),  
    .ZN(_038_)
```

```
) ;  
NAND2_X1_102_(  
.A1(_006_),  
.A2(_021_),  
.ZN(_039_)  
) ;  
XOR2_X1_103_(  
.A1(_022_),  
.A2(_038_),  
.Z(_040_)  

```

```
) ;  
OAI21_X1_110_(  
    .A1(_034_),  
    .A2(_045_),  
    .B(_044_),  
    .ZN(_046_)  
) ;  
OAI21_X1_111_(  
    .A1(_036_),  
    .A2(_042_),  
    .B(_041_),  
    .ZN(_047_)  
) ;  
AOI21_X1_112_(  
    .A1(_002_),  
    .A2(_008_),  
    .B(_039_),  
    .ZN(_048_)  
) ;  
NAND2_X1_113_(  
    .A1(_008_),  
    .A2(_017_),  
    .ZN(_049_)  
) ;  
XNOR2_X1_114_(  
    .A1(_048_),  
    .A2(_049_),  
    .ZN(_050_)  
) ;  
NOR2_X1_115_(  
    .A1(_047_),  
    .A2(_050_),  
    .ZN(_051_)  
) ;  
XNOR2_X1_116_(  
    .A1(_047_),  
    .A2(_050_),  
    .ZN(_052_)  
) ;  
XNOR2_X1_117_(
```

```
.A1(_046_),  
.A2(_052_),  
.ZN(z[5])NAND2_X1_118_(  
.A1(_017_),  
.A2(_021_),  
.ZN(_053_)AOI21_X1_119_(  
.A1(_009_),  
.A2(_049_),  
.B(_039_),  
.ZN(_054_)NAND3_X1_120_(  
.A1(_017_),  
.A2(_021_),  
.A3(_054_),  
.ZN(_055_)XOR2_X1_121_(  
.A1(_053_),  
.A2(_054_),  
.Z(_056_)AOI22_X1_122_(  
.A1(_035_),  
.A2(_043_),  
.B1(_047_),  
.B2(_050_),  
.ZN(_057_)OR2_X1_123_(  
.A1(_051_),  
.A2(_057_),  
.Z(_058_)XOR2_X1_124_(  
.A1(_056_),
```

```

    .A2(_058_),
    .Z(z[6])
);
AOI21_X1_125_(
    .A1(_031_),
    .A2(_032_),
    .B(_015_),
    .ZN(_059_)
);
NOR2_X1_126_(
    .A1(_033_),
    .A2(_059_),
    .ZN(z[31])
);
OAI21_X1_127_(
    .A1(_056_),
    .A2(_058_),
    .B(_055_),
    .ZN(z[7])
);
endmodule

```

QUESTION-3b)

```

/* Generated by Yosys 0.11+10 (git sha1 4871d8f19, clang 6.0.0-1ubuntu2
-fPIC -Os) */
(* top = 1 *)

```

```
(* src = "Q3b.v:1.1-12.10" *)
module Q3b(a, b, p, q, x, y);
  wire _000 ;
  wire _001 ;
  wire _002 ;
  wire _003 ;
  wire _004 ;
  wire _005 ;
  wire _006 ;
  wire _007 ;
  wire _008 ;
  wire _009 ;
  wire _010 ;
  wire _011 ;
  wire _012 ;
  wire _013 ;
  wire _014 ;
  wire _015 ;
  wire _016 ;
  wire _017 ;
  wire _018 ;
  wire _019 ;
  wire _020 ;
  wire _021 ;
  wire _022 ;
  wire _023 ;
  wire _024 ;
  wire _025 ;
  wire _026 ;
  wire _027 ;
  wire _028 ;
  wire _029 ;
  wire _030 ;
  wire _031 ;
  wire _032 ;
  wire _033 ;
  wire _034 ;
  wire _035 ;
  wire _036 ;
  wire _037 ;
```

```
wire _038_;  
wire _039_;  
wire _040_;  
wire _041_;  
wire _042_;  
wire _043_;  
wire _044_;  
wire _045_;  
wire _046_;  
wire _047_;  
wire _048_;  
wire _049_;  
wire _050_;  
wire _051_;  
wire _052_;  
wire _053_;  
wire _054_;  
wire _055_;  
wire _056_;  
wire _057_;  
wire _058_;  
wire _059_;  
wire _060_;  
wire _061_;  
wire _062_;  
wire _063_;  
wire _064_;  
wire _065_;  
wire _066_;  
wire _067_;  
wire _068_;  
wire _069_;  
wire _070_;  
wire _071_;  
wire _072_;  
wire _073_;  
wire _074_;  
wire _075_;  
wire _076_;  
wire _077_;
```

```
wire _078_;
wire _079;
wire _080;
wire _081;
wire _082;
wire _083;
wire _084;
wire _085;
wire _086;
wire _087;
wire _088;
wire _089;
wire _090;
wire _091;
wire _092;
wire _093;

(* src = "Q3b.v:2.16-2.17" *)
input [3:0] a;
(* src = "Q3b.v:2.29-2.30" *)
input [3:0] b;
(* src = "Q3b.v:2.43-2.44" *)
input [3:0] p;
(* src = "Q3b.v:2.56-2.57" *)
input [3:0] q;
(* src = "Q3b.v:3.17-3.18" *)
output [8:0] x;
(* src = "Q3b.v:4.17-4.18" *)
output [8:0] y;
NAND2_X1_094_(
    .A1(a[0]),
    .A2(b[0]),
    .ZN(_034)
);
NAND3_X1_095_(
    .A1(a[0]),
    .A2(b[0]),
    .A3(p[0]),
    .ZN(_035)
);
NAND2_X1_096_(<
```

```
.A1(a[1]),
.A2(b[1]),
.ZN(_036_)

);
NOR2_X1_097_(
.A1(_034_),
.A2(_036_),
.ZN(_037_)

);
AOI22_X1_098_(
.A1(a[0]),
.A2(b[1]),
.B1(b[0]),
.B2(a[1]),
.ZN(_038_)

);
NOR2_X1_099_(
.A1(_037_),
.A2(_038_),
.ZN(_039_)

);
NAND2_X1_100_(
.A1(p[1]),
.A2(_039_),
.ZN(_040_)

);
XNOR2_X1_101_(
.A1(p[1]),
.A2(_039_),
.ZN(_041_)

);
OR2_X1_102_(
.A1(_035_),
.A2(_041_),
.Z(_042_)

);
XOR2_X1_103_(
.A1(_035_),
.A2(_041_),
.Z(x[1])
```

```
) ;  
NAND2_X1_104_(  
.A1(a[2]),  
.A2(b[0]),  
.ZN(_043_)  
) ;  
AND4_X1_105_(  
.A1(a[0]),  
.A2(a[1]),  
.A3(b[2]),  

```

```
) ;  
XOR2_X1_111_(  
.A1(_037_),  
.A2(_048_),  
.Z(_050_)  
);  
XNOR2_X1_112_(  
.A1(p[2]),  
.A2(_050_),  
.ZN(_051_)  
);  
AOI21_X1_113_(  
.A1(_040_),  
.A2(_042_),  
.B(_051_),  
.ZN(_052_)  
);  
AND3_X1_114_(  
.A1(_040_),  
.A2(_042_),  
.A3(_051_),  
.Z(_053_)  
);  
NOR2_X1_115_(  
.A1(_052_),  
.A2(_053_),  
.ZN(x[2])  
);  
AOI21_X1_116_(  
.A1(p[2]),  
.A2(_050_),  
.B(_052_),  
.ZN(_054_)  
);  
NAND2_X1_117_(  
.A1(a[3]),  
.A2(b[0]),  
.ZN(_055_)  
);  
NAND2_X1_118_(
```

```
.A1(a[2]),  
.A2(b[1]),  
.ZN(_056_ )  
);  
NAND2_X1_119_(  
.A1(b[3]),  
.A2(a[1]),  
.ZN(_057_ )  
);  
AND4_X1_120_(  
.A1(a[0]),  
.A2(b[3]),  
.A3(a[1]),  
.A4(b[2]),  
.Z(_058_ )  
);  
AOI22_X1_121_(  
.A1(a[0]),  
.A2(b[3]),  
.B1(a[1]),  
.B2(b[2]),  
.ZN(_059_ )  
);  
NOR3_X1_122_(  
.A1(_056_ ),  
.A2(_058_ ),  
.A3(_059_ ),  
.ZN(_060_ )  
);  
OR3_X1_123_(  
.A1(_056_ ),  
.A2(_058_ ),  
.A3(_059_ ),  
.Z(_061_ )  
);  
OAI21_X1_124_(  
.A1(_058_ ),  
.A2(_059_ ),  
.B(_056_ ),  
.ZN(_062_ )
```

```
) ;  
AND3_X1_125_(  
.A1(_044_),  
.A2(_061_),  
.A3(_062_),  
.Z(_063_)  
) ;  
NAND3_X1_126_(  
.A1(_044_),  
.A2(_061_),  
.A3(_062_),  

```

```
.A3(067) ,  
.ZN(069)  
);  
AOI21_X1_132_(  
.A1(066) ,  
.A2(067) ,  
.B(047) ,  
.ZN(070)  
);  
OR3_X1_133_(  
.A1(049) ,  
.A2(068) ,  
.A3(070) ,  
.Z(071)  
);  
OAI21_X1_134_(  
.A1(068) ,  
.A2(070) ,  
.B(049) ,  
.ZN(072)  
);  
AND2_X1_135_(  
.A1(071) ,  
.A2(072) ,  
.Z(073)  
);  
NAND3_X1_136_(  
.A1(p[31]) ,  
.A2(071) ,  
.A3(072) ,  
.ZN(074)  
);  
AOI21_X1_137_(  
.A1(071) ,  
.A2(072) ,  
.B(p[31]) ,  
.ZN(075)  
);  
XOR2_X1_138_(  
.A1(p[31]) ,
```

```
.A2(_073_),  
.Z(_076_)XNOR2_X1_139_(  
.A1(_054_),  
.A2(_076_),  
.ZN(x[3])OAI21_X1_140_(  
.A1(_049_),  
.A2(_070_),  
.B(_069_),  
.ZN(_077_)NAND2_X1_141_(  
.A1(_064_),  
.A2(_066_),  
.ZN(_078_)AND2_X1_142_(  
.A1(a[3]),  
.A2(b[1]),  
.Z(_079_)NAND2_X1_143_(  
.A1(a[2]),  
.A2(b[2]),  
.ZN(_080_)NOR2_X1_144_(  
.A1(_057_),  
.A2(_080_),  
.ZN(_081_)XOR2_X1_145_(  
.A1(_057_),  
.A2(_080_),  
.Z(_082_)XOR2_X1_146_(
```

```
.A1(079) ,
.A2(082) ,
.Z(083)
);
OAI21_X1_147_(
.A1(058) ,
.A2(060) ,
.B(083) ,
.ZN(084)
);
OR3_X1_148_(
.A1(058) ,
.A2(060) ,
.A3(083) ,
.Z(085)
);
NAND2_X1_149_(
.A1(084) ,
.A2(085) ,
.ZN(086)
);
AOI21_X1_150_(
.A1(064) ,
.A2(066) ,
.B(086) ,
.ZN(087)
);
NAND3_X1_151_(
.A1(064) ,
.A2(066) ,
.A3(086) ,
.ZN(088)
);
XOR2_X1_152_(
.A1(078) ,
.A2(086) ,
.Z(089)
);
XNOR2_X1_153_(
.A1(077) ,
```

```
.A2(_089_),  
.ZN(_090_)  
);  
AOI21_X1_154_(  
.A1(_054_),  
.A2(_074_),  
.B(_075_),  
.ZN(_091_)  
);  
NAND2_X1_155_(  
.A1(_090_),  
.A2(_091_),  
.ZN(_092_)  
);  
XOR2_X1_156_(  
.A1(_090_),  
.A2(_091_),  
.Z(x[4])  
);  
AOI21_X1_157_(  
.A1(_077_),  
.A2(_088_),  
.B(_087_),  
.ZN(_093_)  
);  
AOI21_X1_158_(  
.A1(_079_),  
.A2(_082_),  
.B(_081_),  
.ZN(_000_)  
);  
NAND2_X1_159_(  
.A1(b[3]),  
.A2(a[3]),  
.ZN(_001_)  
);  
NOR2_X1_160_(  
.A1(_080_),  
.A2(_001_),  
.ZN(_002_)
```

```
) ;  
AOI22_X1_161_(  
.A1(b[3]),  
.A2(a[2]),  
.B1(a[3]),  
.B2(b[2]),  
.ZN(_003)  
) ;  
NOR2_X1_162_(  
.A1(_002),  
.A2(_003),  
.ZN(_004)  

```

```
.A2(_007_),  
.Z(_010_)NOR2_X1_169_(  
.A1(_008_),  
.A2(_010_),  
.ZN(_011_)XNOR2_X1_170_(  
.A1(_093_),  
.A2(_011_),  
.ZN(_012_)NAND3_X1_171_(  
.A1(_090_),  
.A2(_091_),  
.A3(_012_),  
.ZN(_013_)XNOR2_X1_172_(  
.A1(_092_),  
.A2(_012_),  
.ZN(x[5])OAI21_X1_173_(  
.A1(_093_),  
.A2(_010_),  
.B(_009_),  
.ZN(_014_)NAND3_X1_174_(  
.A1(b[3]),  
.A2(a[3]),  
.A3(_080_),  
.ZN(_015_)XNOR2_X1_175_(  
.A1(_005_),  
.A2(_015_),  
.ZN(_016_)
```

```
) ;  
XNOR2_X1_176_(  
.A1(_014_),  
.A2(_016_),  
.ZN(_017_)  
) ;  
XOR2_X1_177_(  
.A1(_013_),  
.A2(_017_),  
.Z(x[6])  

```

```
NAND2_X1_183_(
    .A1(q[1]),
    .A2(_039_),
    .ZN(_021_)
);
XNOR2_X1_184_(
    .A1(q[1]),
    .A2(_039_),
    .ZN(_022_)
);
OR2_X1_185_(
    .A1(_020_),
    .A2(_022_),
    .Z(_023_)
);
XOR2_X1_186_(
    .A1(_020_),
    .A2(_022_),
    .Z(y[1])
);
XNOR2_X1_187_(
    .A1(q[2]),
    .A2(_050_),
    .ZN(_024_)
);
AOI21_X1_188_(
    .A1(_021_),
    .A2(_023_),
    .B(_024_),
    .ZN(_025_)
);
AND3_X1_189_(
    .A1(_021_),
    .A2(_023_),
    .A3(_024_),
    .Z(_026_)
);
NOR2_X1_190_(
    .A1(_025_),
    .A2(_026_),
```

```
.ZN(y[2])
);
AOI21_X1_191_(
.A1(q[2]),
.A2(_050_),
.B(_025_),
.ZN(_027_)
);
NAND3_X1_192_(
.A1(q[3]),
.A2(_071_),
.A3(_072_),
.ZN(_028_)
);
AOI21_X1_193_(
.A1(_071_),
.A2(_072_),
.B(q[3]),
.ZN(_029_)
);
XOR2_X1_194_(
.A1(q[3]),
.A2(_073_),
.Z(_030_)
);
XNOR2_X1_195_(
.A1(_027_),
.A2(_030_),
.ZN(y[3])
);
AOI21_X1_196_(
.A1(_027_),
.A2(_028_),
.B(_029_),
.ZN(_031_)
);
NAND2_X1_197_(
.A1(_090_),
.A2(_031_),
.ZN(_032_)
```

```

) ;
XOR2_X1_198_(
.A1(_090_) ,
.A2(_031_) ,
.Z(y[4])
) ;
NAND3_X1_199_(
.A1(_090_) ,
.A2(_012_) ,
.A3(_031_) ,
.ZN(_033_)
) ;
XNOR2_X1_200_(
.A1(_012_) ,
.A2(_032_) ,
.ZN(y[5])
) ;
XOR2_X1_201_(
.A1(_017_) ,
.A2(_033_) ,
.Z(y[6])
) ;
OAI21_X1_202_(
.A1(_017_) ,
.A2(_033_) ,
.B(_019_) ,
.ZN(y[7])
) ;
XNOR2_X1_203_(
.A1(a[0]) ,
.A2(_034_) ,
.ZN(y[0])
) ;
assign x[8] = 1'h0;
assign y[8] = 1'h0;
endmodule

```

Question-4)

```
/* Generated by Yosys 0.11+10 (git sha1 4871d8f19, clang 6.0.0-1ubuntu2
-fPIC -Os) */

(* top = 1 *)
(* src = "MyFSM.v:1.1-43.10" *)
module MyFSM(in, clk, rst, out);
    wire _00_;
    wire _01_;
    wire _02_;
    wire _03_;
    wire _04_;
    wire _05_;
    wire _06_;
(* src = "MyFSM.v:5.11-5.13" *)
```

```

wire [2:0] cS;
(* src = "MyFSM.v:2.11-2.14" *)
input clk;
(* src = "MyFSM.v:2.7-2.9" *)
input in;
(* src = "MyFSM.v:6.11-6.13" *)
wire [2:0] nS;
(* src = "MyFSM.v:3.8-3.11" *)
output out;
(* src = "MyFSM.v:2.16-2.19" *)
input rst;
INV_X1_07 (
    .I(rst),
    .ZN(_00_)
);
INV_X1_08 (
    .I(cS[2]),
    .ZN(_01_)
);
INV_X1_09 (
    .I(cS[1]),
    .ZN(_02_)
);
INV_X1_10 (
    .I(in),
    .ZN(_03_)
);
NOR4_X1_11 (
    .A1(_01_),
    .A2(cS[1]),
    .A3(cS[0]),
    .A4(in),
    .ZN(out)
);
NOR3_X1_12 (
    .A1(cS[2]),
    .A2(_02_),
    .A3(cS[0]),
    .ZN(_04_)
);

```

```

AND4_X1_13_(
    .A1(_01_),
    .A2(_02_),
    .A3(cS[0]),
    .A4(in),
    .Z(_05_)
);
NOR2_X1_14_(
    .A1(_04_),
    .A2(_05_),
    .ZN(nS[0])
);
NOR4_X1_15_(
    .A1(cS[2]),
    .A2(_02_),
    .A3(cS[0]),
    .A4(_03_),
    .ZN(_06_)
);
OR2_X1_16_(
    .A1(_05_),
    .A2(_06_),
    .Z(nS[1])
);
AND2_X1_17_(
    .A1(_03_),
    .A2(_04_),
    .Z(nS[2])
);
(* src = "MyFSM.v:37.1-42.4" *)
DFFSNQ_X1_18_(
    .CLK(clk),
    .D(nS[0]),
    .O(cS[0]),
    .SN(_00_)
);
(* src = "MyFSM.v:37.1-42.4" *)
DFFRNQ_X1_19_(
    .CLK(clk),
    .D(nS[1]),

```

```

    .O(cS[1]),
    .RN(_00_)
);
(* src = "MyFSM.v:37.1-42.4" *)
DFFRNO_X1_20_(
    .CLK(clk),
    .D(nS[2]),
    .O(cS[2]),
    .RN(_00_)
);
endmodule

```

Question 5 (All parts)

```

/* Generated by Yosys 0.11+10 (git sha1 4871d8f19, clang 6.0.0-1ubuntu2
-fPIC -Os) */

(* top = 1 *)
(* src = "Q5.v:1.1-14.10" *)
module Q5(A, B, C, Sum, Carry);
    wire _00_;
    wire _01_;
    wire _02_;
    wire _03_;
    wire _04_;
    wire _05_;
    wire _06_;

```

```

wire _07_;
wire _08;
wire _09;
wire _10;
wire _11;
(* src = "Q5.v:2.8-2.9" *)
input A;
(* src = "Q5.v:3.8-3.9" *)
input B;
(* src = "Q5.v:4.8-4.9" *)
input C;
(* src = "Q5.v:6.9-6.14" *)
output Carry;
(* src = "Q5.v:5.9-5.12" *)
output Sum;
INV_X1_12_(
    .I(B),
    .ZN(_08_)
);
INV_X1_13_(
    .I(C),
    .ZN(_09_)
);
INV_X1_14_(
    .I(A),
    .ZN(_10_)
);
NAND2_X1_15_(
    .A1(_09_),
    .A2(_10_),
    .ZN(_11_)
);
INV_X1_16_(
    .I(_11_),
    .ZN(_00_)
);
NAND2_X1_17_(
    .A1(C),
    .A2(A),
    .ZN(_01_)
);
```

```
) ;  
INV_X1_18_()  
.I(_01_),  
.ZN(_02_)  
) ;  
NAND2_X1_19_()  
.A1(_08_),  
.A2(_01_),  
.ZN(_03_)  

```

```
/* Generated by Yosys 0.11+10 (git sha1 4871d8f19, clang 6.0.0-1ubuntu2
-fPIC -Os) */

(* top = 1 *)
(* src = "Q5.v:1.1-15.10" *)
module Q5(A, B, C, Sum, Carry);
    wire _00_;
    wire _01_;
    wire _02_;
    wire _03_;
    wire _04_;
    wire _05_;
    wire _06_;
    wire _07_;
    wire _08_;
    wire _09_;
    wire _10_;
    wire _11_;

    (* src = "Q5.v:2.13-2.14" *)
    input A;
    (* src = "Q5.v:3.13-3.14" *)
    input B;
    (* src = "Q5.v:4.13-4.14" *)
    input C;
    (* src = "Q5.v:6.13-6.18" *)
    output Carry;
    (* src = "Q5.v:5.13-5.16" *)
    output Sum;

    INV_X1_12_(
        .I(B),
        .ZN(_08_)
    );
    INV_X1_13_(
        .I(C),
        .ZN(_09_)
    );

```

```
INV_X1_14_(  
    .I(A),  
    .ZN(_10_)  
) ;  
NOR2_X1_15_(  
    .A1(C),  
    .A2(A),  
    .ZN(_11_)  
) ;  
INV_X1_16_(  
    .I(_11_),  
    .ZN(_00_)  
) ;  
NOR2_X1_17_(  
    .A1(_09_),  
    .A2(_10_),  
    .ZN(_01_)  
) ;  
INV_X1_18_(  
    .I(_01_),  
    .ZN(_02_)  
) ;  
NOR2_X1_19_(  
    .A1(_08_),  
    .A2(_11_),  
    .ZN(_03_)  
) ;  
NOR2_X1_20_(  
    .A1(_01_),  
    .A2(_03_),  
    .ZN(_04_)  
) ;  
INV_X1_21_(  
    .I(_04_),  
    .ZN(Carry)  
) ;  
NOR2_X1_22_(  
    .A1(B),  
    .A2(_00_),  
    .ZN(_05_)
```

```

);
NOR2_X1_23(
.A1(08),
.A2(02),
.ZN(06)
);
NOR2_X1_24(
.A1(04),
.A2(06),
.ZN(07)
);
NOR2_X1_25(
.A1(05),
.A2(07),
.ZN(Sum)
);
endmodule

```

```

/* Generated by Yosys 0.11+10 (git sha1 4871d8f19, clang 6.0.0-1ubuntu2
-fPIC -Os) */

(* top = 1 *)
(* src = "Q5.v:1.1-13.10" *)
module Q5(A, B, C, Sum, Carry);
    wire _00;
    wire _01;
    wire _02;
    wire _03;
    wire _04;
    wire _05;
    wire _06;
    wire _07;
    wire _08;
    wire _09;
    wire _10;
    wire _11;
    wire _12;

```

```

wire _13_;
wire _14;
(* src = "Q5.v:2.13-2.14" *)
input A;
(* src = "Q5.v:3.13-3.14" *)
input B;
(* src = "Q5.v:4.13-4.14" *)
input C;
(* src = "Q5.v:6.13-6.18" *)
output Carry;
(* src = "Q5.v:5.13-5.16" *)
output Sum;
INV_X1_15_(
    .I(B),
    .ZN(_11_)
);
INV_X1_16_(
    .I(C),
    .ZN(_12_)
);
INV_X1_17_(
    .I(A),
    .ZN(_13_)
);
OR2_X1_18_(
    .A1(C),
    .A2(A),
    .Z(_14_)
);
INV_X1_19_(
    .I(_14_),
    .ZN(_00_)
);
OR2_X1_20_(
    .A1(_12_),
    .A2(_13_),
    .Z(_01_)
);
INV_X1_21_(
    .I(_01_),

```

```
.ZN(_02_)
);
OR2_X1_22_(
.A1(_11_),
.A2(_00_),
.Z(_03_)
);
INV_X1_23_(
.I(_03_),
.ZN(_04_)
);
OR2_X1_24_(
.A1(_02_),
.A2(_04_),
.Z(Carry)
);
OR2_X1_25_(
.A1(B),
.A2(_14_),
.Z(_05_)
);
INV_X1_26_(
.I(_05_),
.ZN(_06_)
);
OR2_X1_27_(
.A1(_11_),
.A2(_01_),
.Z(_07_)
);
INV_X1_28_(
.I(_07_),
.ZN(_08_)
);
OR2_X1_29_(
.A1(Carry),
.A2(_06_),
.Z(_09_)
);
INV_X1_30_()
```

```

        .I(_09_),
        .ZN(_10_)
    );
OR2_X1_31_(
    .A1(_08_),
    .A2(_10_),
    .Z(Sum)
);
endmodule

```

```

/* Generated by Yosys 0.11+10 (git sha1 4871d8f19, clang 6.0.0-1ubuntu2
-fPIC -Os) */

(* top = 1 *)
(* src = "Q5.v:1.1-14.10" *)
module Q5(A, B, C, Sum, Carry);
    wire _00_;
    wire _01_;
    wire _02_;
    wire _03_;
    wire _04_;
    wire _05_;
    wire _06_;
    wire _07_;
    wire _08_;
    wire _09_;
    wire _10_;
    wire _11_;
    wire _12_;
    wire _13_;
    wire _14_;
(* src = "Q5.v:2.13-2.14" *)
    input A;
(* src = "Q5.v:3.13-3.14" *)
    input B;
(* src = "Q5.v:4.13-4.14" *)
    input C;

```

```

(* src = "Q5.v:6.13-6.18" *)
output Carry;
(* src = "Q5.v:5.13-5.16" *)
output Sum;
INV_X1_15_(
.I(B),
.ZN(_11_));
);
INV_X1_16_(
.I(C),
.ZN(_12_));
);
INV_X1_17_(
.I(A),
.ZN(_13_));
);
AND2_X1_18_(
.A1(_12_),
.A2(_13_),
.Z(_14_));
);
INV_X1_19_(
.I(_14_),
.ZN(_00_));
);
AND2_X1_20_(
.A1(C),
.A2(A),
.Z(_01_));
);
INV_X1_21_(
.I(_01_),
.ZN(_02_));
);
AND2_X1_22_(
.A1(_11_),
.A2(_02_),
.Z(_03_));
);
INV_X1_23_(

```

```
.I(_03_),
.ZN(_04_)

);
AND2_X1_24_(
.A1(_00_),
.A2(_04_),
.Z(Carry)

);
AND2_X1_25_(
.A1(_11_),
.A2(_14_),
.Z(_05_)

);
INV_X1_26_(
.I(_05_),
.ZN(_06_)

);
AND2_X1_27_(
.A1(B),
.A2(_01_),
.Z(_07_)

);
INV_X1_28_(
.I(_07_),
.ZN(_08_)

);
AND2_X1_29_(
.A1(Carry),
.A2(_08_),
.Z(_09_)

);
INV_X1_30_(
.I(_09_),
.ZN(_10_)

);
AND2_X1_31_(
.A1(_06_),
.A2(_10_),
.Z(Sum)

);
```

```
endmodule
```