Digital Electronics Verilog Lab 2

Submitted By:
Saurabh Kumar
SC ID: SC22B146
Course: DIGITAL ELECTRONICS AND VLSI DESIGN LAB (AV232)

LAB SHEET

Question No. 1 Design a Three Input majority gate circuit in verilog coding by predefined gates or by Boolean expression.
Sol: **Truth Table:**

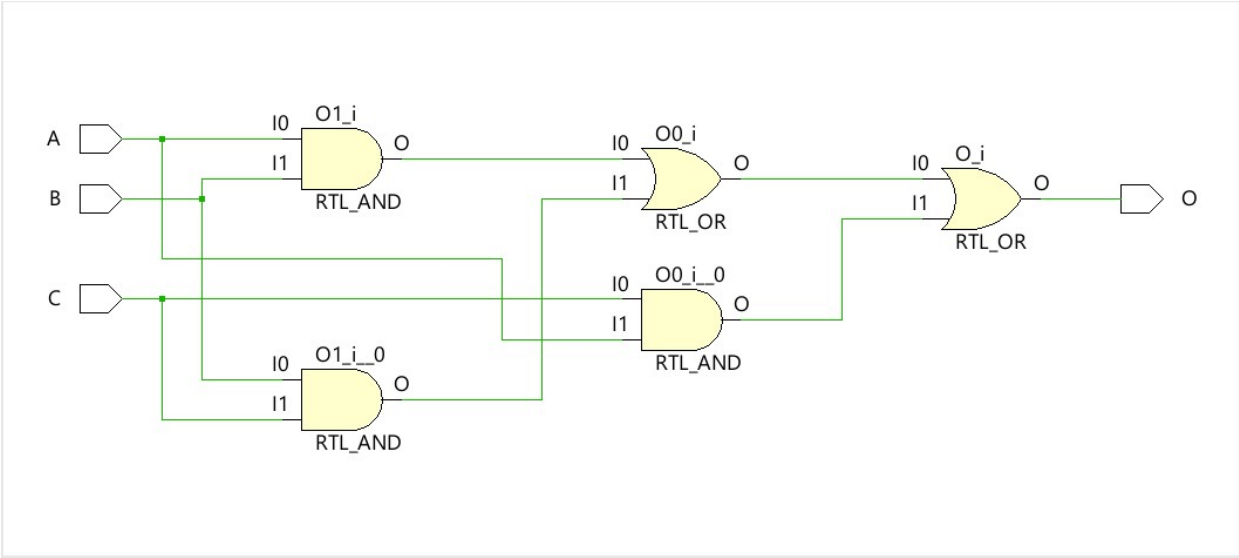
A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

The reduced **boolean** expression using K-Map should be:
F = (A and B) or (B and C) or (C and A)

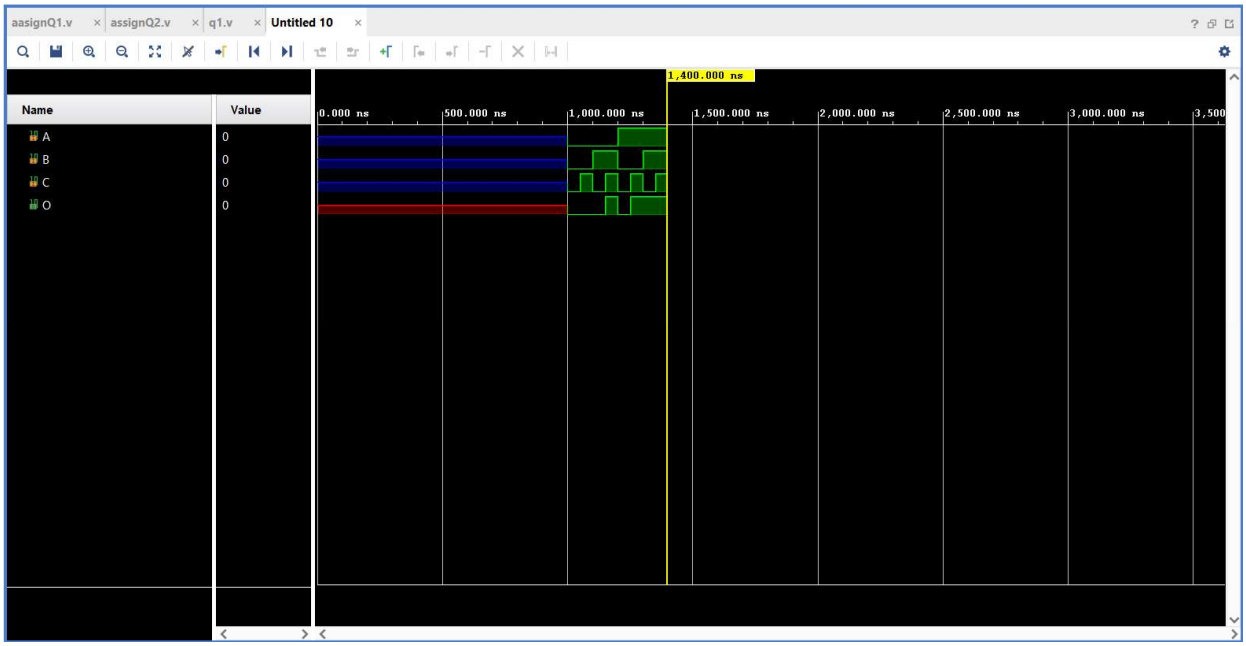
Code:

```
module majorityGate(
input A,B,C,
output O);
  assign O = (A&B)|(B&C)|(C&A);
endmodule
```

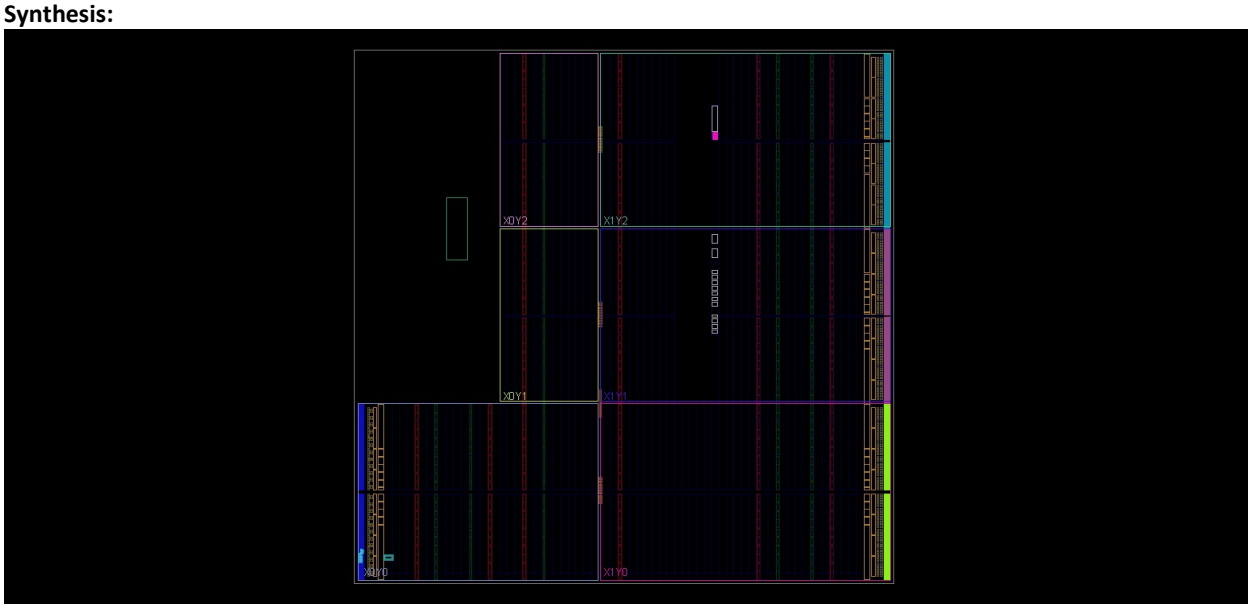
Schematic:



Simulation:



Synthesis:



Question No. 2 Make a BCD to seven segment display circuit.

[You can use behavioural or procedural statement]

[You can arrange the 7 LEDs as your requirement]

Sol: The reduced **boolean** expression should be:

a = A or C or (B and D) or (not B and not D)

b = (not B) or (not C and not D) or (C and D)

c = B or (not C) or D

d = (not B and not D) or (C and not D) or (B and not C and D) or (not B and C) or A

e = (not B and D) or (C and not D)

f = A or (not C and not D) or (B and not C) or (B or not D)

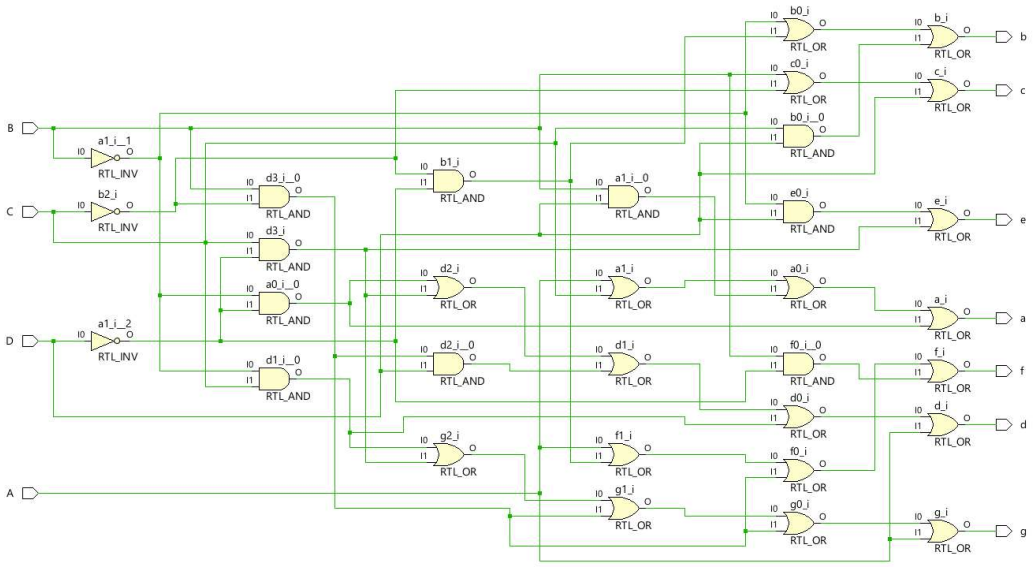
g = (not B and C) or (C and not D) or (B and not C) or (B and not C) or A

Code:

```
module BCDto7Segment(
   input A,B,C,D,
   output a,b,c,d,e,f,g);
   assign a = A|C|(B&D)|((~B)&(~D));
   assign b = (~B)|((~C)&(~D))|(C&D);
   assign c = B|(~C)|D;
   assign d = ((~B)&(~D))|(C&(~D))|(B&(~C)&D)|((~B)&C)|A;
   assign e = ((~B)&D)|(C&(~D));
```

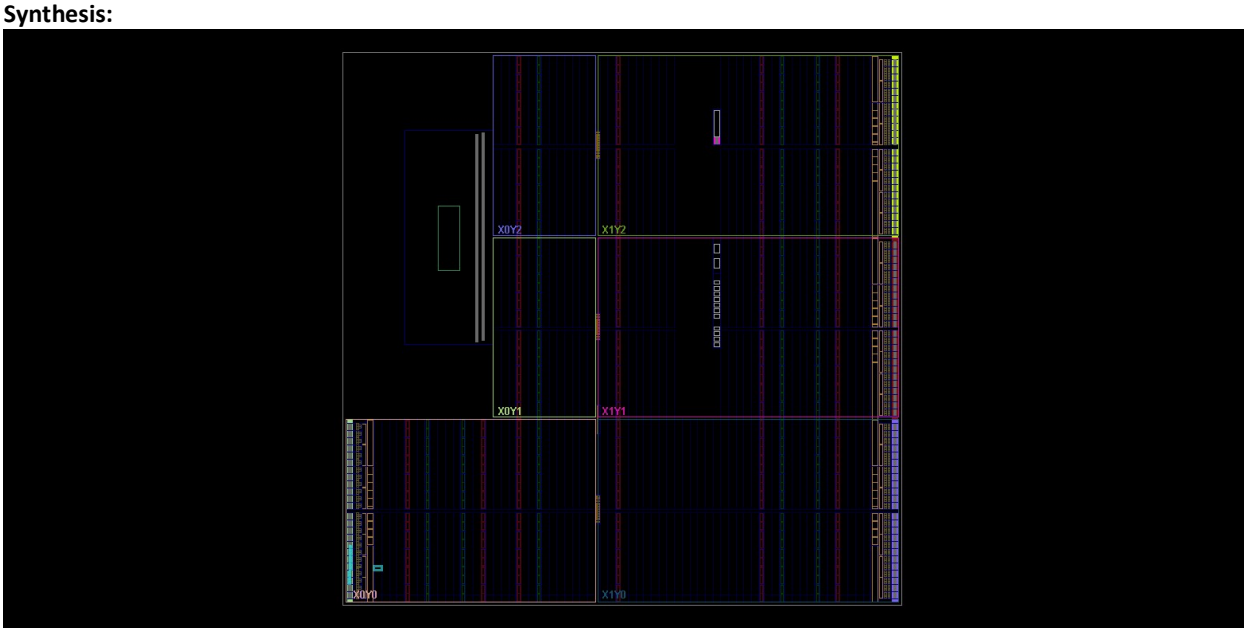
```
    assign f = A|((~C)&(~D))|(B&(~C))|(B&(~D));
    assign g = ((~B)&C)|(C&(~D))|(B&(~C))|(B&(~C))|A;
endmodule
```

Schematic:



Simulation:

Synthesis:



Question No. 3 (i) Make one 4:2 Encoder unit.

Sol: The reduced **boolean** expression should be:

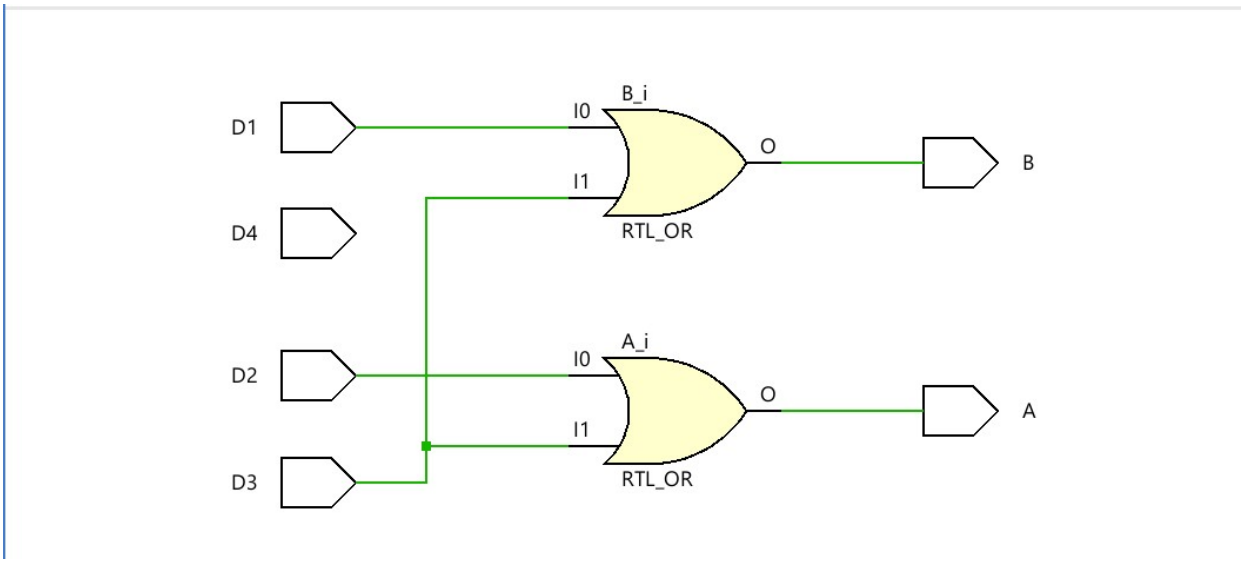
A = D2 or D3

B = D1 or D3

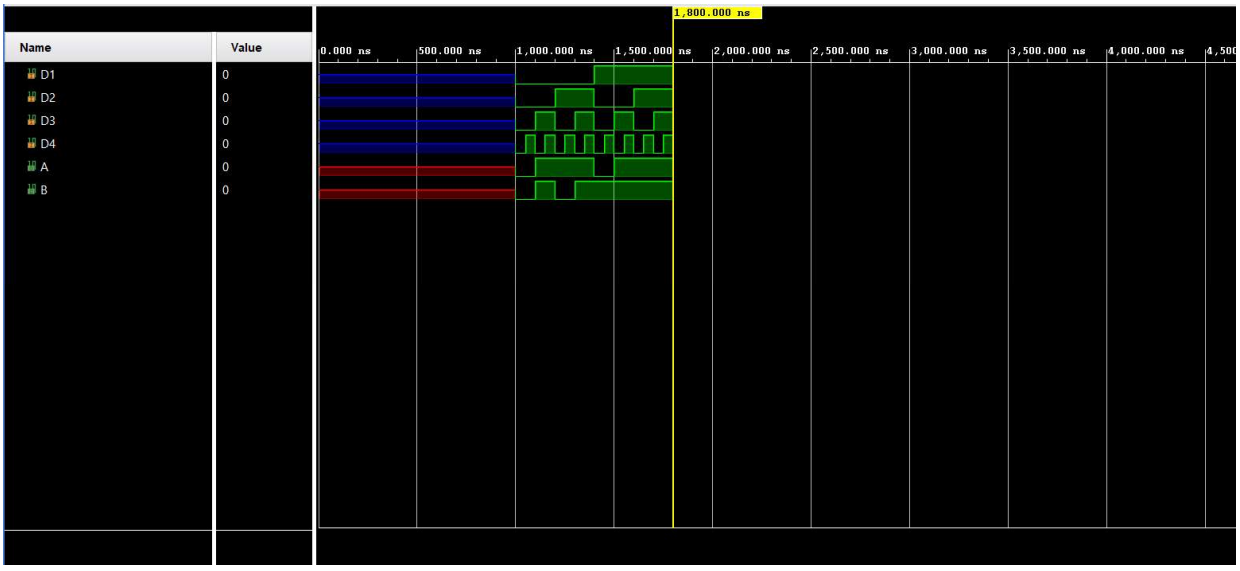
Code:

```
module encoder(
   input D1,D2,D3,D4,
   output A,B);
assign A = D2|D3;
assign B = D1|D3;
endmodule
```

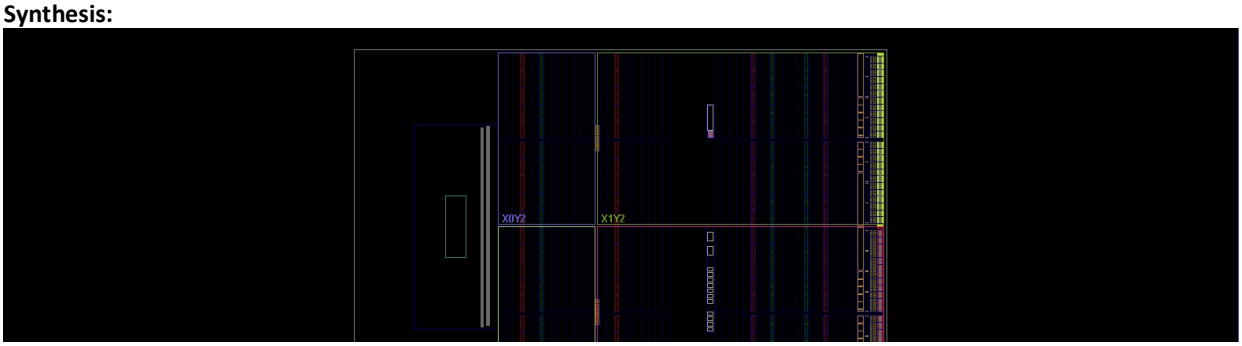
Schematic:

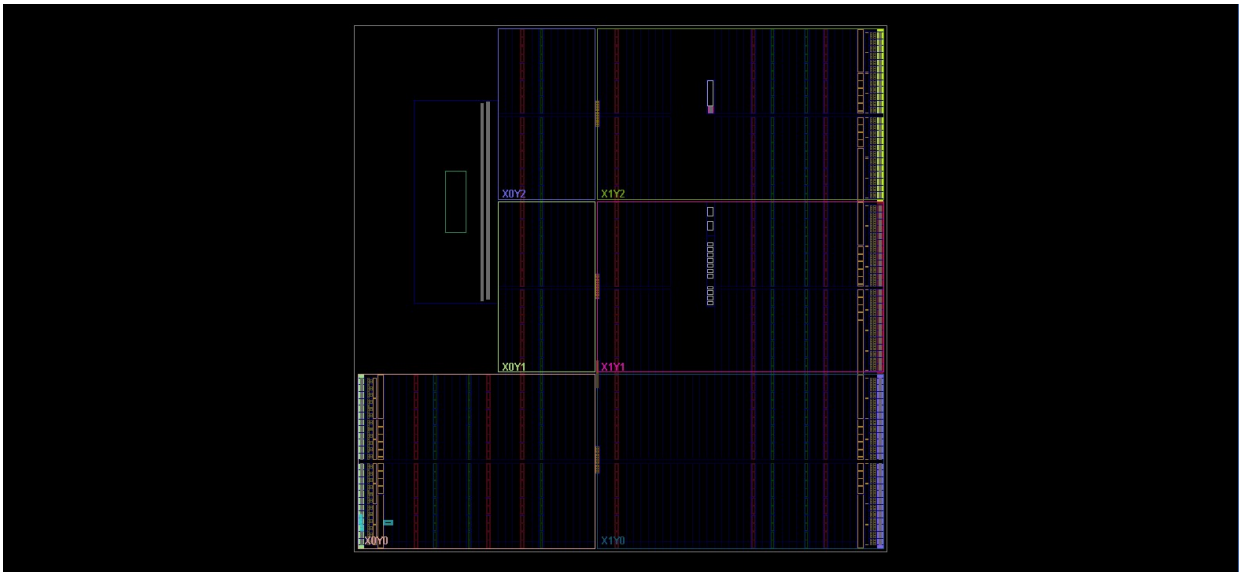


Simulation:



Synthesis:





Question No. 3 (ii) Make a MUX from that Encoder unit.

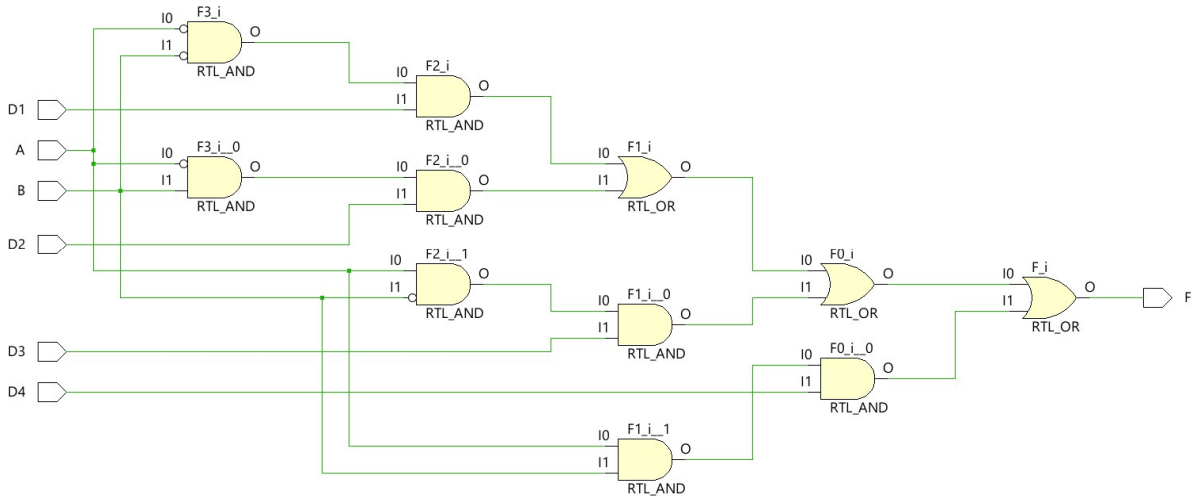
Sol: The reduced **boolean** expression should be:

F = (not A and not B and D1) or (not A and B and D2) or (A and not B and D3) or (A and B and D4)

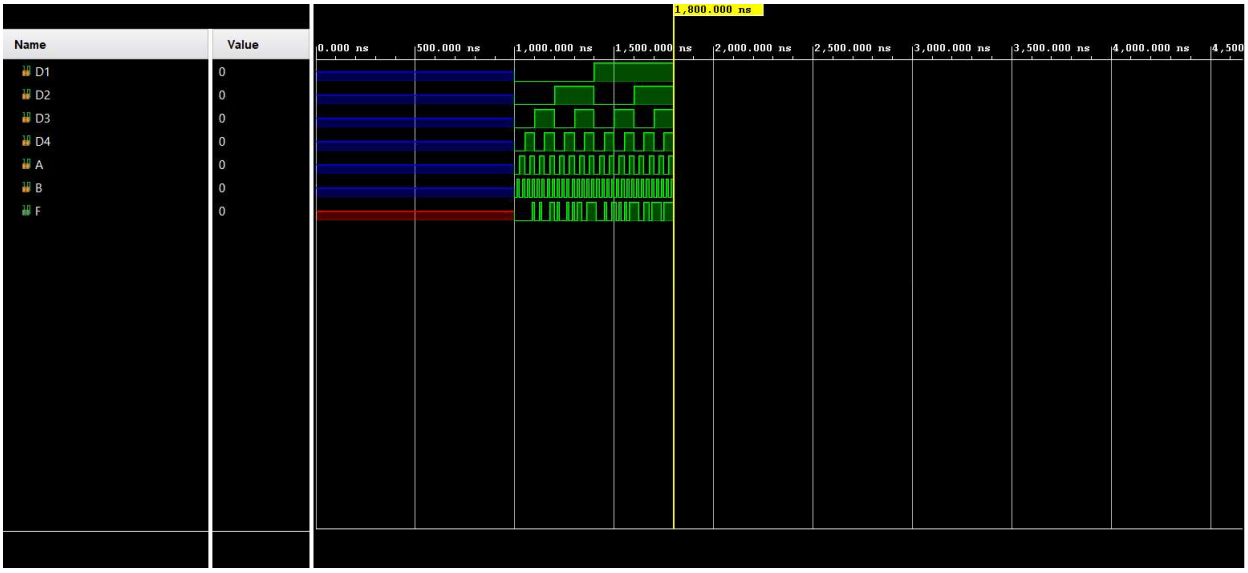
Code:

```
module mux(
   input D1,D2,D3,D4,A,B,
   output F);
assign F = ((~A)&(~B)&(D1))|((~A)&(B)&(D2))|((A)&(~B)&(D3))|((A)&(B)&(D4));
endmodule
```

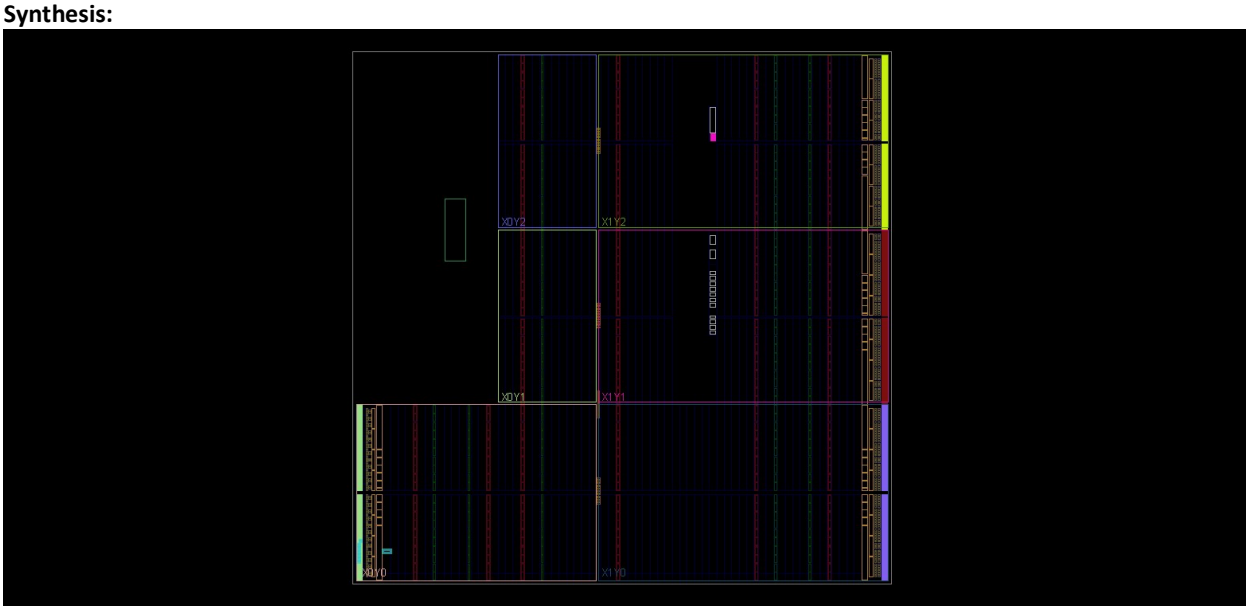
Schematic:



Simulation:



Synthesis:



ASSIGNMENT

Question No. 1 Design a Full Adder circuit in verilog by nand gate only.

Sol: Code:

```
module aasignQ1(
   input A,B,Cin,
   output S, Cout);
//   S = ((A xor B) nand ((A xor B) nand Cin)) nand (Cin nand ((A xor B) nand Cin))
//         -wire 1-            -wire1-                                  -wire1-
//                              ------wire2------            -------wire2-------
//        ----------wire3-----------------        ------------wire4-----------
//        ---------------------------------Sum----------------------------------
   wire wire1, wire2, wire3, wire4;
   wire w1,w2,w3;
   nand(w1,A,B);  //w1 = ~(AB)
   nand(w2,A,w1);  //w2 = ~(Aw1)
   nand(w3,B,w1);  //w3 = ~(Bw1)

   nand(wire1,w2,w3);  //wire1 = A xor B
   nand(wire2,wire1,Cin);
   nand(wire3,wire1,wire2);
   nand(wire4,wire2,Cin);
```

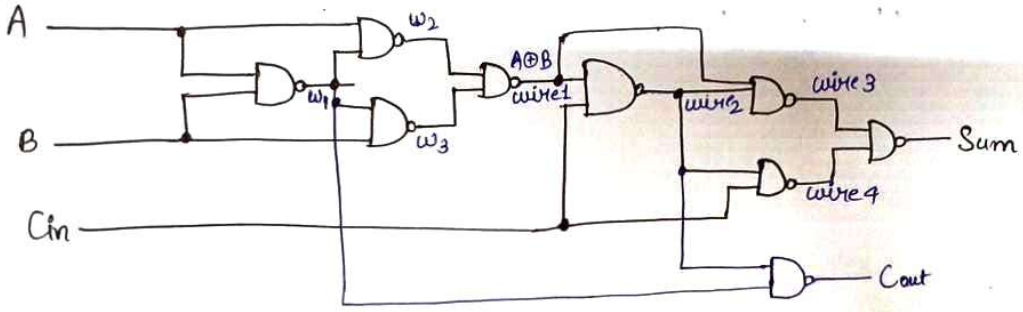
```
nand(S,wire3,wire4);
nand(Cout,w1,wire2);

endmodule
```

Schematic and Boolean Expression:

Verilog Lab Assignment -2

① Design a Full Adder circuit in verilog by nand gate only.

Sol^n:



Boolean Expression:

By reduction,

$$Sum = A \oplus B \oplus Cin$$

$$= \overline{(A \oplus B).\overline{(A \oplus B)\,Cin}.\ \overline{Cin.\overline{(A \oplus B)\,Cin}}},$$

where $A \oplus B = \overline{\overline{A \cdot \overline{AB}}.\ \overline{B \cdot \overline{AB}}}$

$$Cout = AB + (A \oplus B)Cin$$

$$= \overline{\overline{Cin\,(A \oplus B)}.\ \overline{AB}}$$

Question No. 2 Design a 2:1 Mux circuit in verilog by nor gate only.

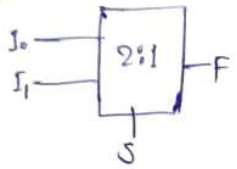
Sol: Code:

```
module muxUsingNor(
  input I0,I1,S,
  output F);

wire w1,w2,w3,w4,w5;
nor(w1,I0,I1);
nor(w2,I0,S);
nor(w3,S,S);
nor(w4,w3,I1);
nor(w5,w1,w2);
nor(F,w4,w5);
endmodule
```

Boolean Expression:

② Design a 2:1 Mux Circuit in Verilog by nor gate only.

Soln:



$F = I_0\bar{S} + I_1 S$

$= (\bar{S}I_0 + S)(\bar{S}I_0 + I_1)$ [Using $A + BC = (A+B)(A+C)$]

$= (\bar{S} + S)(I_0 + S)(\bar{S} + I_1)(I_0 + I_1)$

$= (I_0 + S)(\bar{S} + I_1)(I_0 + I_1)$

$= \overline{(S + I_0)(\bar{S} + I_1)(I_0 + I_1)}$

$\therefore F = \overline{(S + I_0) + (\bar{S} + I_1) + (I_0 + I_1)}$

Schematic: