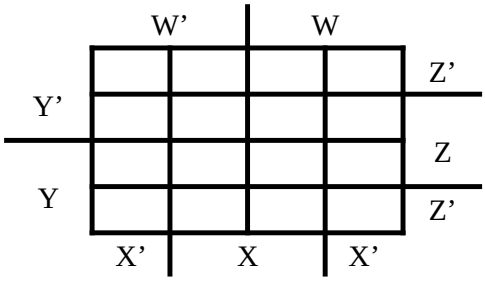
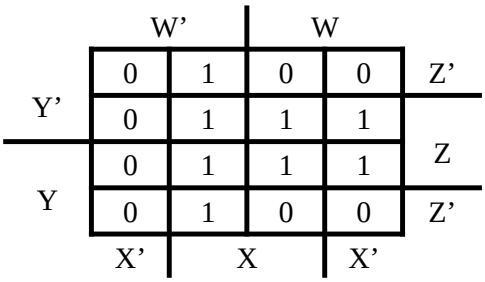
1) For a function F(W,X,Y,Z) that has the following truth table, fill in the equivalent Karnaugh map for it, below.

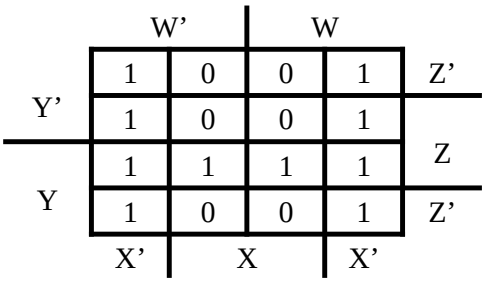
W	X	Y	Z	F(W,X,Y,Z)
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



2) For the following Karnaugh map, **circle** the essential prime implicants. Draw a **dotted circle** around any non-essential prime implicants.



3) For the Boolean function F(W,X,Y,Z) represented by the following Karnaugh map, find a **minimal sum-of-products** expression.



F(W,X,Y,Z) = _____

4) For the Karnaugh map, below, for F(W,X,Y,Z), we don't care about some states because their input values are not possible. Create a **minimal sum-of-products** expression for F(W,X,Y,Z).

		W'		W	
Y'		1	d	0	1	Z'
		d	1	0	d	
Y		d	1	0	0	Z
		d	d	d	1	Z'
		X'	X		X'	

F(W,X,Y,Z) = _____

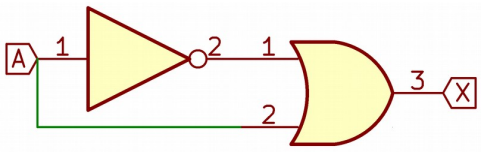
5) For the Karnaugh map, below, for F(W,X,Y,Z), we don't care about some states because their input values are not possible. Use the OFF set of the function to create a **minimal sum-of-products** expression for the **complement**, F'(W,X,Y,Z).

		W'		W	
Y'		1	d	0	1	Z'
		d	1	0	d	
Y		d	1	0	0	Z
		d	d	d	1	Z'
		X'	X		X'	

F'(W,X,Y,Z) = _____

6) If the complement of a Boolean function is expressed as a sum of products like this: F'(X,Y,Z) = X'·Y·Z + X·Y'·Z, then use DeMorgan's law to transform it into a **product of sums** expression for F(X,Y,Z).

Use the following circuit for questions 7 and 8.



7) What kind of hazard exists? (i.e., Static-0 or Static-1?)

8) When does the glitch occur?

Use the Karnaugh map, below, for questions 7, 8, and 9. Assume a minimal sum-of-products expression is used to realize the Boolean function F(W,X,Y,Z) with gates.

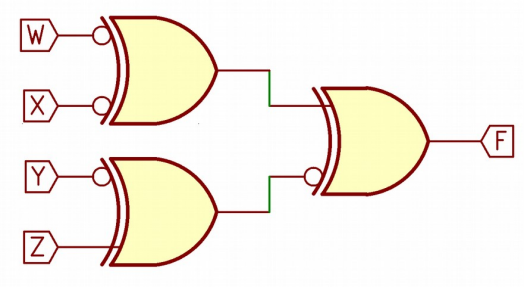
	W'		W		
Y'	0	0	1	1	Z'
	0	0	1	1	
	0	1	1	1	Z
Y	0	1	0	0	Z'
	X'	X		X'	

9) What kind of logic hazards exist? (i.e., static -1 or static-0)

10) Describe W, X, Y, and Z when the glitch occurs.

11) Write a complete sum-of-products expression for F that is glitch-free. Do so by adding a single term to the minimal sum-of-product expression.

12) For the Boolean function realized by the circuit, below, complete the truth table. (Reminder: You should not need to evaluate each case if you remember how a network of XOR and XNOR gates can be reduced.)

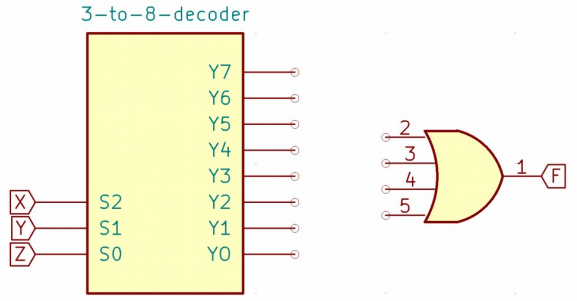


W	X	Y	Z	F
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

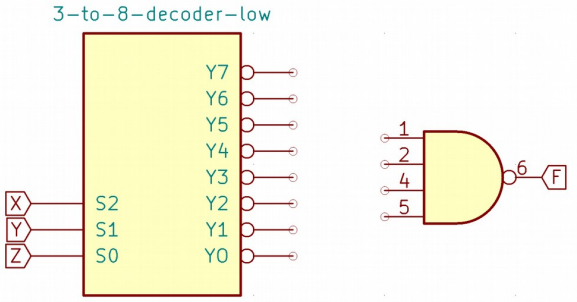
Refer to the truth table, below, for questions 13, 14, 15, and 16.

X	Y	Z	F(X,Y,Z)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

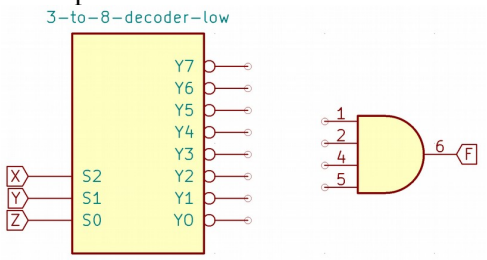
13) Wire a simple 3-to-8 decoder, with active-high outputs, to a 4-input OR gate to implement the Boolean function.



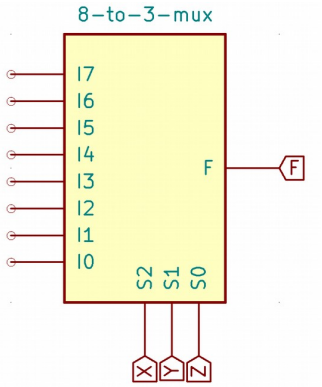
14) Wire a 3-to-8 decoder, with active-low outputs, to a 4-input NAND gate to implement the Boolean function.



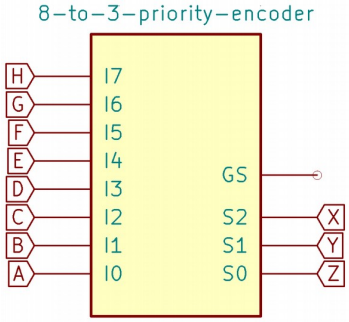
15) Wire a 3-to-8 decoder, with active-low outputs, to a 4-input AND gate to implement the Boolean function.



16) Wire the inputs of a simple 8-to-3 multiplexer, with active-high inputs, to either 1 (logic high, or V_{DD}) or 0 (Gnd) to implement the Boolean function.



17) Write a Boolean expression for the Y output of an 8-to-3 priority encoder with active-high inputs and outputs. Express your answer in terms of A, B, C, D, E, F, G, H, or their negations.



Y=_____

Here are two example Verilog modules:

```
module xor2(a,b,c);
    input wire a,b;
    output wire [1:0] c;
    assign c = {a,b} ^ 2'b11;
endmodule

module upper(d1,d2,x);
    input wire d1, d2;
    output wire [1:0] x;
    wire [1:0] result;
    xor2 u1(.a(d1), .b(d2), .c(result));
    assign x = ~result;
endmodule
```

18) Write a Boolean expression for x[0] in terms of d1 and d2.
Make it as simple as possible.

19) Write a Verilog module called 'SoP' that implements the Boolean function **OUT = W·X'·Y·Z + W'·X·Y'·Z' + W·X·Y·Z**
List the ports in any order.

20) Write a Verilog module named ‘invertif’ with 1-bit input wires named ‘a’ and ‘doinvert’, and a 1-bit output wire named ‘out’. ‘out’ should be the inverse of ‘a’ if ‘doinvert’ is high. Otherwise, ‘out’ should be the value of ‘a’.

21) For signals defined in a Verilog module as ‘`wire [31:0]X;`’ and ‘`wire [64:1]Y;`’, what is the size of the vector, `{X[12:3],Y[14:10],X[30:28],X[22:15]}`

22) For the following Verilog module,

```
module decode(x,a);
       input wire [2:0]a;
       output wire [7:0]x;
       wire [7:0] map [7:0];
       assign map[0] = 8'h12;
       assign map[1] = 8'h34;
       assign map[3] = 8'h58;
       assign map[6] = 8'hac;
       assign map[7] = 8'hbd;
       assign x = map[a];
endmodule
```

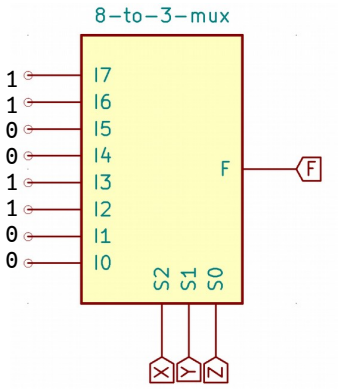
What is the value of x[4] when a == 3’b011 ?

23) For the following Verilog module

```
module priority42(in,s);
    input wire [3:0] in;
    output wire [1:0] s;
    assign s = in[3] == 1 ? 2'b11 :
               in[2] == 1 ? 2'b10 :
               in[1] == 1 ? 2'b01 :
               in[0] == 1 ? 2'b00 :
               2'b00;
endmodule
```

What is the value of s when in == 4'b0110 ?

24) For the circuit below, write a minimal sum-of-products expression for F(X,Y,Z).



25) For the following Verilog code, what is the value of f when x=1, y=0, z=1?

```
module mux8to1(in,s,out);
    input wire [7:0] in;
    input wire [2:0] s;
    output wire out;
    assign out = (s== 3'b111) ? in[7] :
                 (s== 3'b110) ? in[6] :
                 (s== 3'b101) ? in[5] :
                 (s== 3'b100) ? in[4] :
                 (s== 3'b011) ? in[3] :
                 (s== 3'b010) ? in[2] :
                 (s== 3'b001) ? in[1] :
                                in[0];
endmodule

module higher(x,y,z,f);
    input x,y,z;
    output f;
    mux8to1 u1(8'b11001100,{x,y,z},f);
endmodule
```