Jordan Wu

U0900517

Homework Assignment 5

- **Start with this equation:**

$$\boldsymbol{p = a \cdot \bar{b} \cdot (\bar{f} \cdot \overline{(\bar{a} \cdot d + b)} \cdot c + \bar{a} \cdot \bar{d})}$$

Do the following manipulations on the equation:

- **Reduce it to sum-of-products form, and**
- **eliminate any redundant terms or product terms that will always be false.**

Steps:

1) $DeMorgan's\ Theorem \stackrel{\text{def}}{=} \overline{x \cdot y} = \bar{x} + \bar{y}$
   - $\overline{((\bar{a} \cdot d) + b)} \Rightarrow (\overline{\bar{a} \cdot d} \cdot \bar{b})$
2) $DeMorgan's\ Theorem \stackrel{\text{def}}{=} \overline{x \cdot y} = \bar{x} + \bar{y}$
   - $(\overline{\bar{a} \cdot d} \cdot \bar{b}) \Rightarrow ((a + \bar{d}) \cdot \bar{b})$
3) $Distributive\ Property \stackrel{\text{def}}{=} x \cdot (y + z) = x \cdot y + x \cdot z$
   - $((a + \bar{d}) \cdot \bar{b}) \Rightarrow (a \cdot \bar{b} + \bar{d} \cdot \bar{b})$
4) $Distributive\ Property \stackrel{\text{def}}{=} x \cdot (y + z) = x \cdot y + x \cdot z$
   - $\bar{f} \cdot (a \cdot \bar{b} + \bar{d} \cdot \bar{b}) \cdot c \Rightarrow \bar{f} \cdot a \cdot \bar{b} \cdot c + \bar{f} \cdot \bar{d} \cdot \bar{b} \cdot c$
5) $Distributive\ Property \stackrel{\text{def}}{=} x \cdot (y + z) = x \cdot y + x \cdot z$
   - $a \cdot \bar{b} \cdot (\bar{f} \cdot a \cdot \bar{b} \cdot c + \bar{f} \cdot \bar{d} \cdot \bar{b} \cdot c + \bar{a} \cdot \bar{d}) \Rightarrow a \cdot \bar{b} \cdot \bar{f} \cdot a \cdot \bar{b} \cdot c + a \cdot \bar{b} \cdot \bar{f} \cdot \bar{d} \cdot \bar{b} \cdot c + a \cdot \bar{b} \cdot \bar{a} \cdot \bar{d}$
6) $Single\ Variable\ Theorems \stackrel{\text{def}}{=} x \cdot x = x\ and\ x \cdot \bar{x} = 0$
   - $a \cdot \bar{b} \cdot \bar{f} \cdot a \cdot \bar{b} \cdot c \Rightarrow a \cdot \bar{b} \cdot \bar{f} \cdot c$
   - $a \cdot \bar{b} \cdot \bar{f} \cdot \bar{d} \cdot \bar{b} \cdot c \Rightarrow a \cdot \bar{b} \cdot \bar{f} \cdot \bar{d} \cdot c$
   - $a \cdot \bar{b} \cdot \bar{a} \cdot \bar{d} \Rightarrow 0$
7) $Absorption \stackrel{\text{def}}{=} x + x \cdot y = x$
   - $a \cdot \bar{b} \cdot \bar{f} \cdot c + a \cdot \bar{b} \cdot \bar{f} \cdot \bar{d} \cdot c \Rightarrow a \cdot \bar{b} \cdot \bar{f} \cdot c$

Reducing the equation using the steps, starting from the original equation:

$$p = a \cdot \bar{b} \cdot \left(\bar{f} \cdot \overline{((\bar{a} \cdot d) + b)} \cdot c + \bar{a} \cdot \bar{d}\right) \quad (1)$$

$$p = a \cdot \bar{b} \cdot \left(\bar{f} \cdot \left(\overline{\bar{a} \cdot d} \cdot \bar{b}\right) \cdot c + \bar{a} \cdot \bar{d}\right) \quad (2)$$

$$p = a \cdot \bar{b} \cdot (\bar{f} \cdot ((a + \bar{d}) \cdot \bar{b}) \cdot c + \bar{a} \cdot \bar{d}) \quad (3)$$

$$p = a \cdot \bar{b} \cdot (\bar{f} \cdot \left(a \cdot \bar{b} + \bar{d} \cdot \bar{b}\right) \cdot c + \bar{a} \cdot \bar{d}\,) \quad (4)$$

$$p = a \cdot \bar{b} \cdot (\bar{f} \cdot a \cdot \bar{b} \cdot c + \bar{f} \cdot \bar{d} \cdot \bar{b} \cdot c + \bar{a} \cdot \bar{d}\,) \quad (5)$$

$$p = a \cdot \bar{b} \cdot \bar{f} \cdot a \cdot \bar{b} \cdot c + a \cdot \bar{b} \cdot \bar{f} \cdot \bar{d} \cdot \bar{b} \cdot c + a \cdot \bar{b} \cdot \bar{a} \cdot \bar{d} \quad (6)$$

$$p = a \cdot \bar{b} \cdot \bar{f} \cdot c + a \cdot \bar{b} \cdot \bar{f} \cdot \bar{d} \cdot c \quad (7)$$

$$p = a \cdot \bar{b} \cdot \bar{f} \cdot c \quad (8)$$

I was able to reduce the following output p, into a sum-of-products form. Then using some properties to simplify the equation.

- **Do problem B.4 from the end of appendix B.**
    - **First, draw the truth table. (See page B-5 for an example.)**
    - **Next, write a sum-of-products representation for the function described by the truth table. (This is required.) Important note: The final formula in the example on page B-12 is wrong, but the solution procedure is correct.**
    - **Finally, draw the logic diagram using only AND gates, OR gates, and NOT gates (do not use extra inversion bubbles, but NOT gates are allowed).**

B.5

One logic function that is used for a variety of purpose (including within adders and to compute parity) is exclusive OR. The output of a two-input exclusive OR function is true only if exactly one of the inputs is true. Show the truth table for a two-input exclusive OR function and implement this function using AND gates, OR gates, and inverters.

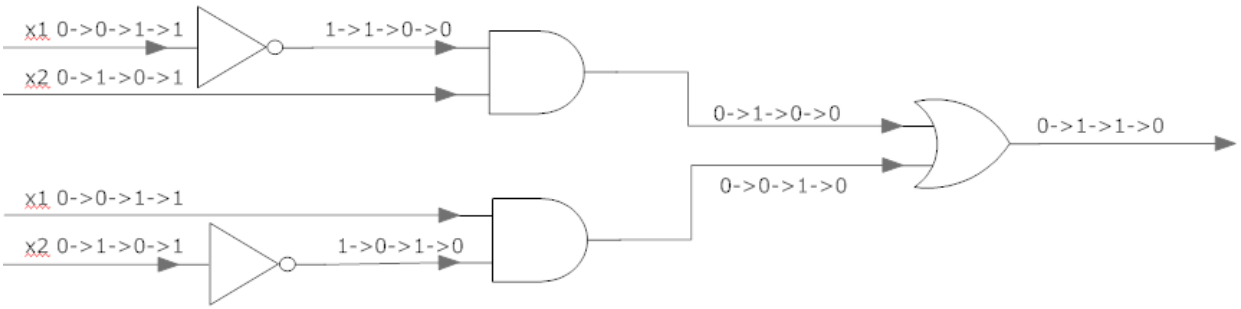
The following truth table for XOR gate

XOR Gate		
input		output
X_1	X_2	f
0	0	0
0	1	1
1	0	1
1	1	0

Just looking at the output on the truth table, the output will be 1 when $x_1 = 0$ AND $x_2 = 1$ OR $x_1 = 1$ AND $x_2 = 0$.

$$\xrightarrow[\text{we will get the following}]{} f = \overline{x_1} \cdot x_2 + x_1 \cdot \overline{x_2}$$

The following output can be represented by using AND gates, OR gates and inverters. (note, that the output is the same as the XOR output)



- **Do problem B.7 from the end of appendix B. Your table should have a single output column labeled 'ODD'. Then, using your truth table from this question, write a sum-of-products representation for the four-bit ODD parity function. Finally, do problem B.8 from the end of appendix B. Use inversion bubbles where appropriate (no NOT gates).**

  There are several definitions of odd parity - for this problem consider all four inputs together with the output (all five bits). Odd parity guarantees an odd number of 'true' or '1' bits. So, for inputs '0', '1', '1', and '0', the output will be '1' to make it such that an odd number of the input/output bits are true.

B.7

Construct the truth table for a four-input odd-parity function.

Parity is the concept to detect errors. A single bit error is detected by it. Odd parity if the number of 1s is odd, even otherwise. In the Odd parity we want the 4-bits input to have odd numbers of 1s, when adding the parity bit we want to satisfy this condition. Even parity we want the opposite, an even number of 1s when adding the parity bit.

Truth Table					
Transmission bits				Odd	Minterm
X_1	X_2	X_3	X_4	f	m
0	0	0	0	1	m_0
0	0	0	1	0	m_1
0	0	1	0	0	m_2
0	0	1	1	1	m_3
0	1	0	0	0	m_4
0	1	0	1	1	m_5
0	1	1	0	1	m_6
0	1	1	1	0	m_7
1	0	0	0	0	m_8
1	0	0	1	1	m_9
1	0	1	0	1	m_{10}
1	0	1	1	0	m_{11}
1	1	0	0	1	m_{12}
1	1	0	1	0	m_{13}
1	1	1	0	0	m_{14}
1	1	1	1	1	m_{15}

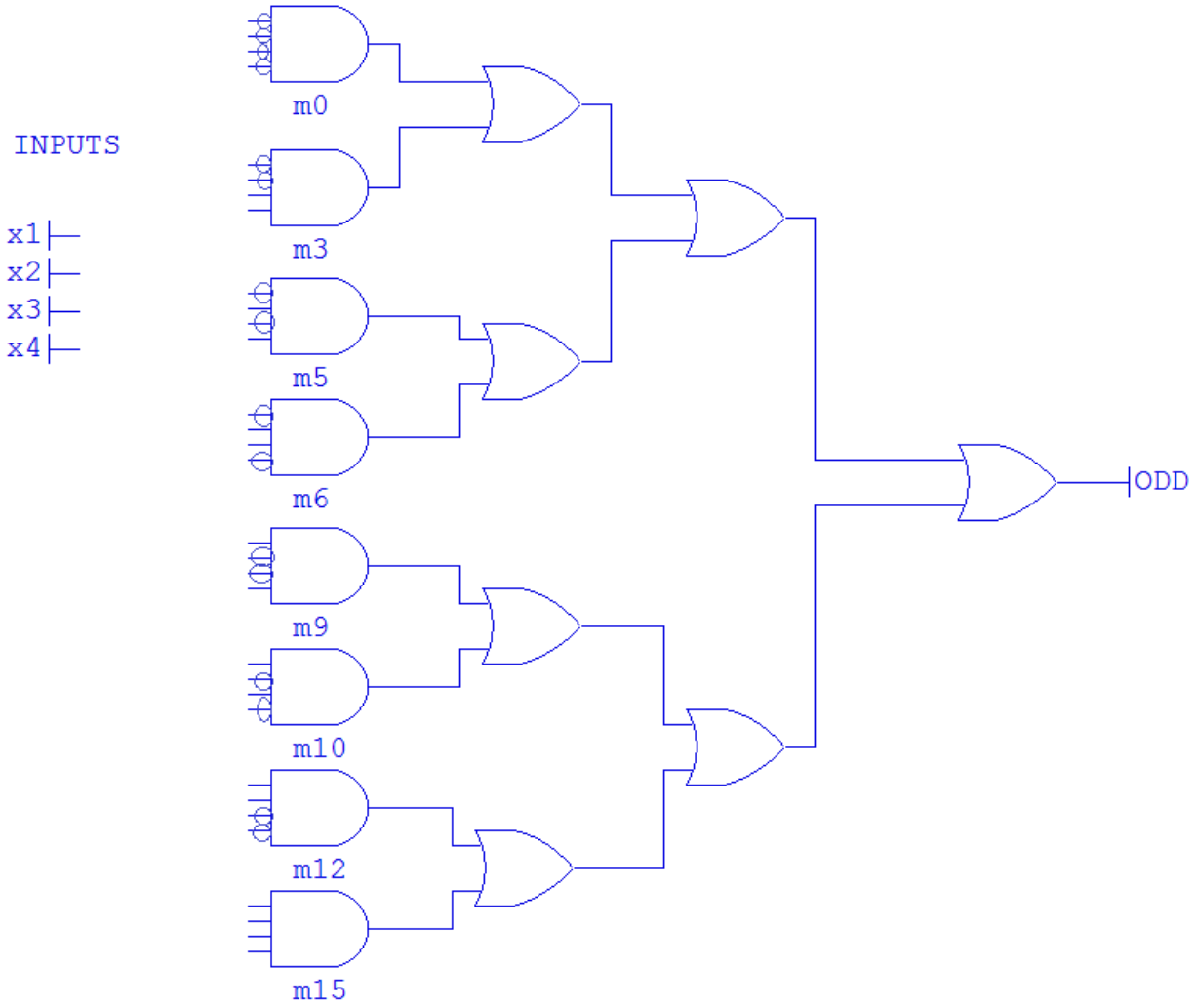
Looking at this truth table, the output is true (1) with the following minterms: m_0, m_3, m_5, m_6, m_9, m_{10}, m_{12}, and m_{15}. The sum of these minterm will give us the sum of products.

$$Odd = \sum m(0,3,5,6,9,10,12,15)$$

$$Odd = \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} + \overline{x_1} \cdot \overline{x_2} \cdot x_3 \cdot x_4 + \overline{x_1} \cdot x_2 \cdot \overline{x_3} \cdot x_4 + \overline{x_1} \cdot x_2 \cdot x_3 \cdot \overline{x_4} + x_1 \cdot \overline{x_2} \cdot \overline{x_3} \cdot x_4 + x_1 \cdot \overline{x_2} \cdot x_3 \cdot \overline{x_4} + x_1 \cdot x_2 \cdot \overline{x_3} \cdot \overline{x_4} + x_1 \cdot x_2 \cdot x_3 \cdot x_4$$

B.8

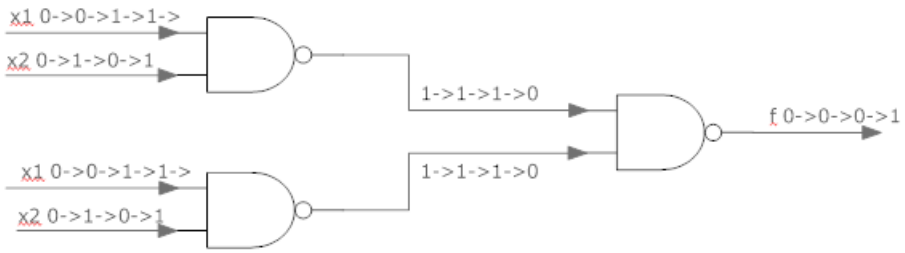
Implement the four-input odd-parity function with AND and OR gates using bubbled inputs and outputs. (program did not have inversion bubbles, I quickly added them)



- **The NAND gate is universal -- this means that AND gates, OR gates, and NOT gates can be constructed through clever arrangements of NAND gates. Show this is true by drawing circuits using only two-input NAND gates that compute NOT, AND, and OR logic functions. Label the inputs and outputs with reasonable labels.**

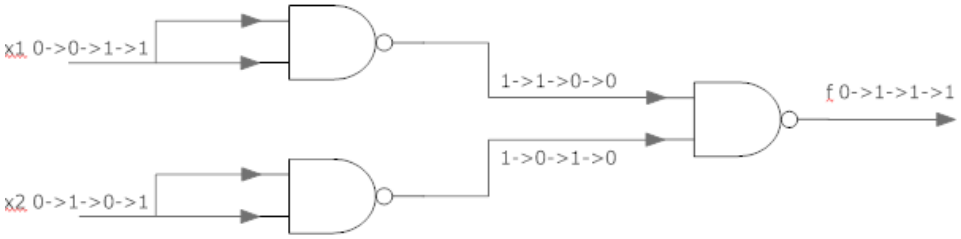
  Finally, show that your NAND gate construction of an AND gate is correct by writing a Boolean algebra equation for it, then simplify the algebra. The NAND gate input terms and the NAND operation should be clearly visible in your first equation, and the AND term should be clearly visible in the simplified equation.

AND GATE



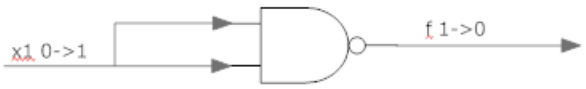
AND		
X_1	X_2	f
0	0	0
0	1	0
1	0	0
1	1	1

OR GATE



OR		
X_1	X_2	f
0	0	0
0	1	1
1	0	1
1	1	1

NOT GATE



NOT	
X_1	f
0	1
1	0

PROOF OF NAND GATE INTO AND GATE

Steps:

1. $DeMorgan's\ Theorem \stackrel{\text{def}}{=} \overline{x \cdot y} = \bar{x} + \bar{y}$
2. $Single\ Variable\ Theorems \stackrel{\text{def}}{=} \bar{\bar{x}} = x$
3. $Single\ Variable\ Theorems \stackrel{\text{def}}{=} x + x = x$

Now using the steps, we get the following

$$\overline{\overline{x_1 \cdot x_2} \cdot \overline{x_1 \cdot x_2}} \qquad (1)$$

$$\overline{\overline{x_1 \cdot x_2}} + \overline{\overline{x_1 \cdot x_2}} \qquad (2)$$

$$x_1 \cdot x_2 + x_1 \cdot x_2 \qquad (2)$$

$$x_1 \cdot x_2 \qquad (4)$$

- ***(Extra credit - 20 homework points)*** **Do problem B.13. Write your Boolean equations in any form. Explain how the terms of your equations guarantee the result. Truth tables and logic circuits are not required or expected for this problem. You should directly write and explain Boolean equations that are true only if the corresponding mathematical comparisons are true.**

  Extra credit will be applied only after other students' grades have been determined. Extra credit will only be awarded if your work on this problem is correct and professional (excellent).

B.13

Assume that X consists of 3 bits, x2 x1 x0, and Y consists of 3 bits, y2 y1 y0. Write logic functions that are true if and only if

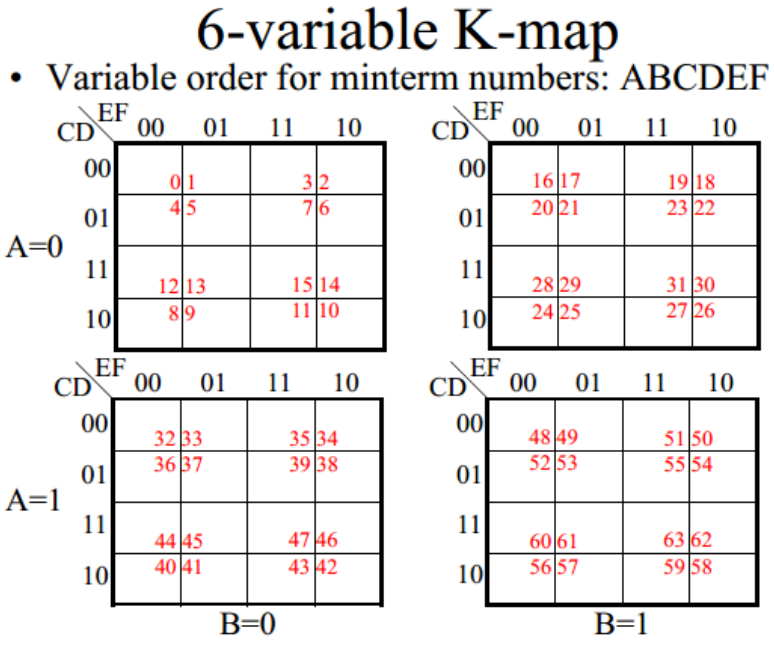
- X < Y, where X and Y are thought of as unsigned binary numbers
- X < Y, where X and Y are thought of as signed (two's complement numbers)
- X = Y

My approach will be to write the truth table (this will give us all the possible inputs), which should give me all the possible outcome for a given event/output (should give me correct logic function). I will treat the 3-bits X and Y as the input and X < Y unsigned/signed and X=Y as the output. For my truth table the MSB is X_2/Y_2 (input variable sequence is for example: $X_2\ X_1\ X_0\ Y_2\ Y_1\ Y_0$). I will also solve the outputs by hand on the truth table

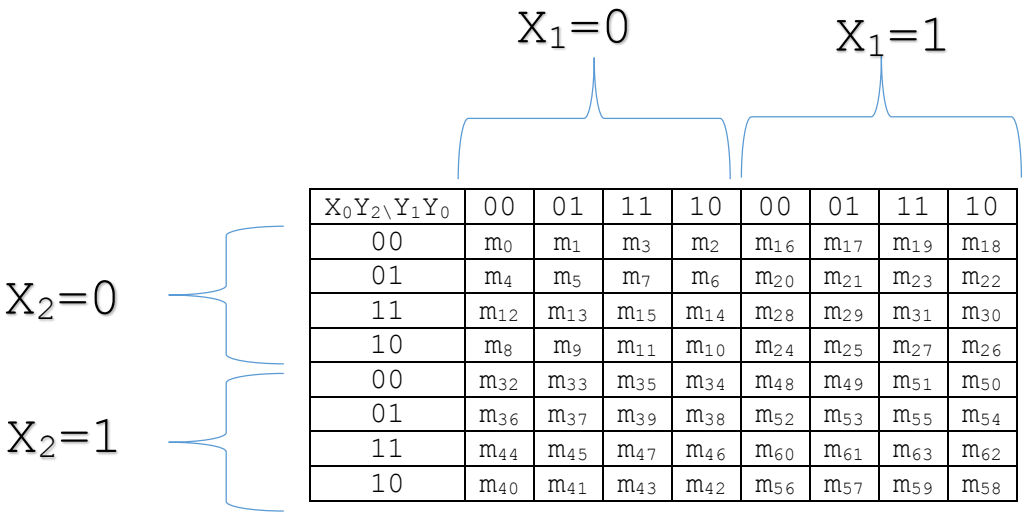
Inputs						outputs			minterm
X_2	X_1	X_1	Y_2	Y_1	Y_0	X<Y (unsigned)	X<Y (signed)	X=Y	$m_{\#}$
0	0	0	0	0	0	0	0	1	m_0
0	0	0	0	0	1	1	1	0	m_1
0	0	0	0	1	0	1	1	0	m_2
0	0	0	0	1	1	1	1	0	m_3
0	0	0	1	0	0	1	0	0	m_4
0	0	0	1	0	1	1	0	0	m_5
0	0	0	1	1	0	1	0	0	m_6
0	0	0	1	1	1	1	0	0	m_7
0	0	1	0	0	0	0	0	0	m_8
0	0	1	0	0	1	0	0	1	m_9
0	0	1	0	1	0	1	1	0	m_{10}
0	0	1	0	1	1	1	1	0	m_{11}
0	0	1	1	0	0	1	0	0	m_{12}
0	0	1	1	0	1	1	0	0	m_{13}
0	0	1	1	1	0	1	0	0	m_{14}
0	0	1	1	1	1	1	0	0	m_{15}
0	1	0	0	0	0	0	0	0	m_{16}
0	1	0	0	0	1	0	0	0	m_{17}
0	1	0	0	1	0	0	0	1	m_{18}
0	1	0	0	1	1	1	1	0	m_{19}
0	1	0	1	0	0	1	0	0	m_{20}
0	1	0	1	0	1	1	0	0	m_{21}
0	1	0	1	1	0	1	0	0	m_{22}
0	1	0	1	1	1	1	0	0	m_{23}
0	1	1	0	0	0	0	0	0	m_{24}
0	1	1	0	0	1	0	0	0	m_{25}
0	1	1	0	1	0	0	0	0	m_{26}
0	1	1	0	1	1	0	0	1	m_{27}
0	1	1	1	0	0	1	0	0	m_{28}
0	1	1	1	0	1	1	0	0	m_{29}
0	1	1	1	1	0	1	0	0	m_{30}
0	1	1	1	1	1	1	0	0	m_{31}
1	0	0	0	0	0	0	1	0	m_{32}
1	0	0	0	0	1	0	1	0	m_{33}
1	0	0	0	1	0	0	1	0	m_{34}
1	0	0	0	1	1	0	1	0	m_{35}
1	0	0	1	0	0	0	0	1	m_{36}
1	0	0	1	0	1	1	1	0	m_{37}
1	0	0	1	1	0	1	1	0	m_{38}
1	0	0	1	1	1	1	1	0	m_{39}
1	0	1	0	0	0	0	1	0	m_{40}
1	0	1	0	0	1	0	1	0	m_{41}
1	0	1	0	1	0	0	1	0	m_{42}

1	0	1	0	1	1	0	1	0	m_{43}
1	0	1	1	0	0	0	0	0	m_{44}
1	0	1	1	0	1	0	0	1	m_{45}
1	0	1	1	1	0	1	1	0	m_{46}
1	0	1	1	1	1	1	1	0	m_{47}
1	1	0	0	0	0	0	1	0	m_{48}
1	1	0	0	0	1	0	1	0	m_{49}
1	1	0	0	1	0	0	1	0	m_{50}
1	1	0	0	1	1	0	1	0	m_{51}
1	1	0	1	0	0	0	0	0	m_{52}
1	1	0	1	0	1	0	0	0	m_{53}
1	1	0	1	1	0	0	0	1	m_{54}
1	1	0	1	1	1	1	1	0	m_{55}
1	1	1	0	0	0	0	1	0	m_{56}
1	1	1	0	0	1	0	1	0	m_{57}
1	1	1	0	1	0	0	1	0	m_{58}
1	1	1	0	1	1	0	1	0	m_{59}
1	1	1	1	0	0	0	0	0	m_{60}
1	1	1	1	0	1	0	0	0	m_{61}
1	1	1	1	1	0	0	0	0	m_{62}
1	1	1	1	1	1	0	0	1	m_{63}

We can use the minterms to find the logic function. I will be using a Karnaugh map to reduce the equation for the six variables. Below an example of a K-map with 6-variable input



We get the following



$X_0Y_2\backslash Y_1Y_0$	00	01	11	10	00	01	11	10
00	m_0	m_1	m_3	m_2	m_{16}	m_{17}	m_{19}	m_{18}
01	m_4	m_5	m_7	m_6	m_{20}	m_{21}	m_{23}	m_{22}
11	m_{12}	m_{13}	m_{15}	m_{14}	m_{28}	m_{29}	m_{31}	m_{30}
10	m_8	m_9	m_{11}	m_{10}	m_{24}	m_{25}	m_{27}	m_{26}
00	m_{32}	m_{33}	m_{35}	m_{34}	m_{48}	m_{49}	m_{51}	m_{50}
01	m_{36}	m_{37}	m_{39}	m_{38}	m_{52}	m_{53}	m_{55}	m_{54}
11	m_{44}	m_{45}	m_{47}	m_{46}	m_{60}	m_{61}	m_{63}	m_{62}
10	m_{40}	m_{41}	m_{43}	m_{42}	m_{56}	m_{57}	m_{59}	m_{58}

X < Y (unsigned binary number)

$$X < Y = \sum m(1,2,3,4,5,6,7,10,11,12,13,14,15,19,20,21,22,23,28,29,30,31,37,38,39,46,47,55)$$

Using the karnaugh map we can reduce this equation into this

$$X < Y = \overline{X_2} \cdot Y_2 \cdot \overline{Y_0} \cdot \overline{Y_1} + \overline{X_2} \cdot X_0 \cdot Y_2 \cdot \overline{Y_0} + \overline{X_2} \cdot X_1 \cdot Y_2 \cdot \overline{Y_1} + \overline{X_2} \cdot X_1 \cdot X_0 \cdot Y_2 + X_0 \cdot \overline{Y_0} \cdot \overline{Y_1} + X_1 \cdot X_0 \cdot \overline{Y_1} + X_1 \cdot \overline{Y_0}$$

The sum-of-product tells us that if one or more of the products are 1 then the results will be 1.

For example the first product for this results is $\overline{X_2} \cdot Y_2 \cdot \overline{Y_0} \cdot \overline{Y_1}$ if this is 1 then the results will also be one. For this to be 1 the following input must be the following.

X_2	X_1	X_0	Y_2	Y_1	Y_0
0	Don't care	Don't care	1	0	0

Looking at the table above, we do not care about X_1 or X_2 since product $\overline{X_2} \cdot Y_2 \cdot \overline{Y_0} \cdot \overline{Y_1}$ will still be 1 (will satisfy X < Y unsigned). We can do the same thing for each of the following products. Here are all the following products that must be satisfy to guarantee that X < Y

- $\overline{X_2} \cdot Y_2 \cdot \overline{Y_0} \cdot \overline{Y_1}$
- $\overline{X_2} \cdot X_0 \cdot Y_2 \cdot \overline{Y_0}$
- $\overline{X_2} \cdot X_1 \cdot Y_2 \cdot \overline{Y_1}$
- $\overline{X_2} \cdot X_1 \cdot X_0 \cdot Y_2$
- $X_0 \cdot \overline{Y_0} \cdot \overline{Y_1}$
- $X_1 \cdot X_0 \cdot \overline{Y_1}$
- $X_1 \cdot \overline{Y_0}$

IF AND ONLY IF, one or more of the products are 1 then it is guarantee that X < Y for unsigned binary number

X < Y (signed binary number)

$$X < Y = \sum m(1,2,3,10,11,19,32,33,34,35,37,38,39,40,41,42,43,46,47,48,49,50,51,55,56,57,58,59)$$

Using the karnaugh map we can reduce this equation into this

$$X < Y = \overline{X_2} \cdot \overline{X_1} \cdot Y_2 \cdot \overline{Y_1} + \overline{X_2} \cdot \overline{X_1} \cdot X_0 \cdot Y_2 + \overline{X_2} \cdot Y_2 \cdot Y_0 \cdot \overline{Y_1} + \overline{X_2} \cdot X_0 \cdot Y_2 \cdot Y_0 + \overline{X_1} \cdot X_0 \cdot \overline{Y_1} + X_0 \cdot Y_0 \cdot \overline{Y_1} + \overline{X_1} \cdot Y_0$$

The following products for this case will be the following

- $\overline{X_2} \cdot \overline{X_1} \cdot Y_2 \cdot \overline{Y_1}$
- $\overline{X_2} \cdot \overline{X_1} \cdot X_0 \cdot Y_2$
- $\overline{X_2} \cdot Y_2 \cdot Y_0 \cdot \overline{Y_1}$
- $\overline{X_2} \cdot X_0 \cdot Y_2 \cdot Y_0$
- $\overline{X_1} \cdot X_0 \cdot \overline{Y_1}$
- $X_0 \cdot Y_0 \cdot \overline{Y_1}$
- $\overline{X_1} \cdot Y_0$

IF AND ONLY IF, one or more of the products are 1 then it is guarantee that X < Y for signed binary number

X = Y

$$X = Y = \sum m(9,18,27,36,45,54,63)$$

This cannot be reduced using karnaugh map, since the minterms are not grouped or paired up in the K-map. One interesting thing is that the minterms are multiple of nine. For this case, the following products are the minterms that are multiple of nine. If one of the minterms are satisfy, then we can guarantee that X = Y

NOTE: The following reduced equation might be off (there are programs that solves k-map which I used for this problem)