FIT1047 Tutorial 5

Topics

- Revision of weeks 1 to 4

Instructions

- The tasks are supposed to be done in groups. In some tasks, it might be useful to have different roles for people in the group.

Task 1: Data Representation

1.a Given the following two binary numbers: 11111100 and 01110000.

(i) Which of the two numbers is the larger unsigned binary number? **Solution:** Assume, A = 11111100, and B = 01110000. For A, we can calculate as follows:

Positional Value	128	64	32	16	8	4	2	1
Bits	1	1	1	1	1	1	0	0

So, A_2 = 128+64+32+16+8+4 = 252_{10}.
Then, for B

Positional Value	128	64	32	16	8	4	2	1
Bits	0	1	1	1	0	0	0	0

So, B_2 = 64+32+16 = 112_{10}. Thus, $A > B$.

(ii) Which of these two numbers is larger when it is being interpreted on a computer using signed two's complement representation?

Solution: In two's complement representation, A is a negative number (as the left most bit is 1). To know the decimal value it represents, we need to inverse the bits and "add 1".

Bits	1	1	1	1	1	1	0	0
Inverse	0	0	0	0	0	0	1	1
Add 1	0	0	0	0	0	0	0	1
The Number	0	0	0	0	0	1	0	0

$(0000\ 0100)_2 = 4_{10}$. So, in two's complement representation, $A_2 = -4$. Now, B is a positive number (as the left most bit is 0).

Positional Value	128	64	32	16	8	4	2	1
Bits	0	1	1	1	0	0	0	0

So, $B_2 = +112_{10}$. Thus, $A < B$.

(iii) Which of these two numbers is smaller when it is being interpreted on a computer using signed-magnitude representation?

Solution: Assume, A = 11111100, and B = 01110000. Now, A is a negative number (as the left most bit is 1). For A, we can calculate the value it represents as follows:

Positional Value	Sign Bit	64	32	16	8	4	2	1
Bits	1	1	1	1	1	1	0	0

So, A_2 = -(64+32+16+8+4) = -124_{10}. Now, B is a positive number (as the left most bit is 0).

Positional Value	Sign Bit	64	32	16	8	4	2	1
Bits	0	1	1	1	0	0	0	0

So, $B_2 = +(64 + 32 + 16) = +112_{10}$. Thus, $A < B$.

1.b Add the following unsigned binary numbers:

01000100
10111011

Carry								
Number A	0	1	0	0	0	1	0	0
Number B	1	0	1	1	1	0	1	1
A + B	1	1	1	1	1	1	1	1

01011011
00011111

Carry			1	1	1	1	1	
Number A	0	1	0	1	1	0	1	1
Number B	0	0	0	1	1	1	1	1
A + B	0	1	1	1	1	0	1	0

10101100
00100100

Carry		1		1	1			
Number A	1	0	1	0	1	1	0	0
Number B	0	0	1	0	0	1	0	0
A + B	1	1	0	1	0	0	0	0

1.c Subtract the following signed binary numbers using two's complement arithmetic (Note: These numbers use 2's complement notation).

11000100	01011011	10101100
-00111011	-00011111	-00100100

Solution:

In 2's complement notation, subtraction of a positive number is replaced by adding the 2's complement. Then, if there is no overflow, the left-most carry bit is just discarded.

11000100
-00111011

Carry	1	1			1			
A	1	1	0	0	0	1	0	0
2's compl. of B	1	1	0	0	0	1	0	1
A - B	1	0	0	0	1	0	0	1

01011011
-00011111

Carry	1					1	1	
A	0	1	0	1	1	0	1	1
2's compl. of B	1	1	1	0	0	0	0	1
A - B	0	0	1	1	1	1	0	0

10101100
-00100100

10101100 11011100 10001000

Carry	1	1	1	1	1			
A	1	0	1	0	1	1	0	0
2's compl. of B	1	1	0	1	1	1	0	0
A - B	1	0	0	0	1	0	0	0

1.e Assume a 24-bit word in a computer. In these 24 bits, you should represent the string B95. If your computer uses 8-bit ASCII with even parity (i.e. from left to right 7 bit for the character and 1 parity bit), how would the computer represent the string B95.

Solution: 7-bits ASCII characters representing 2, 5 and 9 are:

'B' is represented by	1000010
'9' is represented by	0111001
'5' is represented by	0110101

Using the right-most bit as a parity bit. We will have 8-bit ASCII characters.

7-Bit ASCII	Count of 1 bit	Parity Bit	8-Bit ASCII
1000010	2	0	10000100
0111001	4	0	01110010
0110101	4	0	01101010

So, in computer the string 'B95' will be represented as → 100001000111001001101010

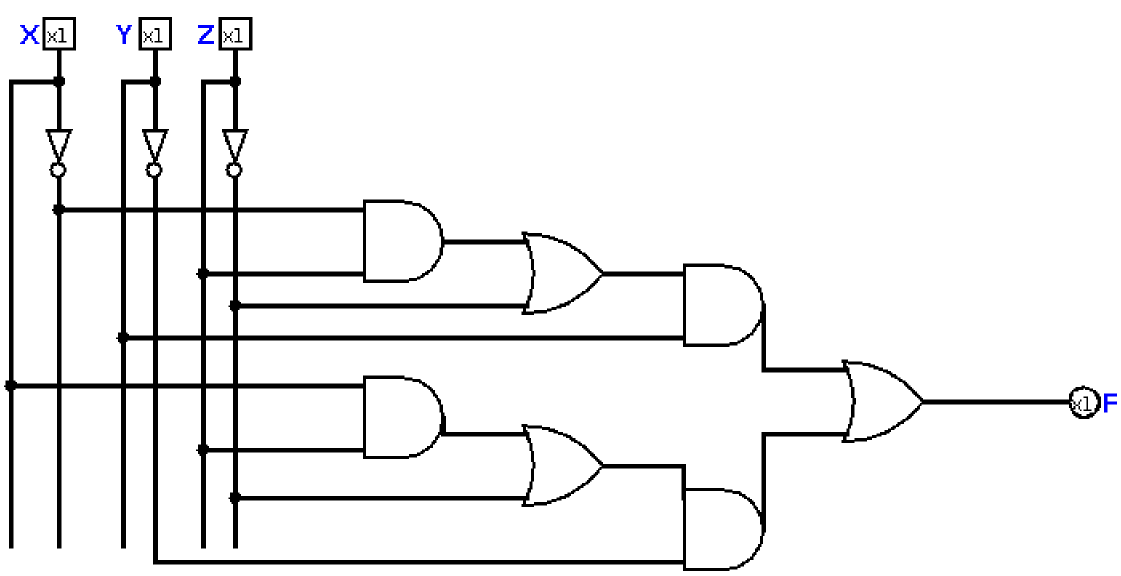
Task 2: Boolean Algebra

Given the function $F(X, Y, Z) = Y(\overline{X}Z + \overline{Z}) + \overline{Y}(XZ + \overline{Z})$

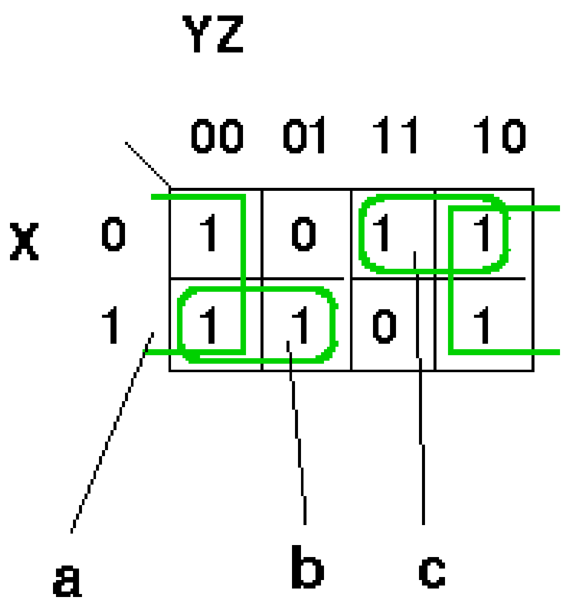
2.a List the truth table for F.

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

2.b Draw the logic diagram using the original Boolean expression.



2.c Simplify the expression using a Karnaugh map.



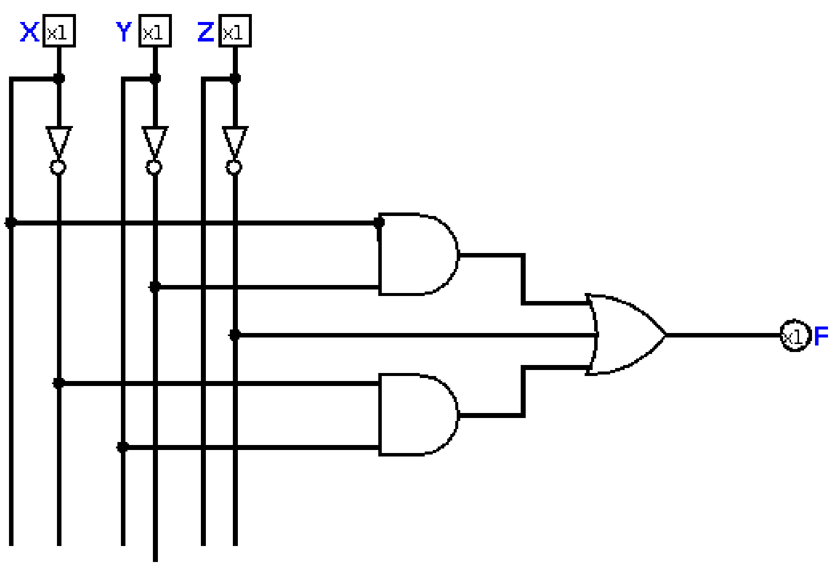
Section a is $\overline{Z}$, section b is $X\overline{Y}$ and section c is $\overline{X}Y$.

Thus, $F(X,Y,Z) = \overline{Z} + X\overline{Y} + \overline{X}Y$

2.d List the truth table for your answer in part 2.c.

Solution: Same as 2.a

2.e Draw the logic diagram for the simplified expression in part 2.c.



Task3: CPUs and MARIE

3.a Explain why, in MARIE the MAR is only 12 bits wide and the AC and MBR are 16 bits wide (Hint: Consider the difference between data and addresses).

3.b Write the following code segment in MARIE assembly language:

```
X = 1
while X < 10 do
  X = X + 1
endwhile
```

Solution:

Label	Instruction	Operand
Begin,	Load	var_X
	Subt	Ten
	skipcond	000
	Jump	Done
	Load	var_X
	Add	One
	Store	var_X
	Jump	Begin
Done,	Halt	
var_X	Dec	1
One,	Dec	1
Ten,	Dec	10

3.c The following table shows MARIE's datapath control signals:

	Memory	MAR	PC	MBR	AC	IN	OUT	IR
P_2,P_1,P_0 (Read)	000	001	010	011	100	101	110	111
P_5,P_4,P_3 (Write)	000	001	010	011	100	101	110	111

In addition to these signals there are special signals such as IncrPC to increment the PC and M_R and M_W for read from memory and write into memory.

Provide RTL and control signals for the 6 steps of the `JumpI` instruction:

	Description	Operation	Time	Data Movement	Signals
1.	Read the program counter and write into the memory address register	Fetch	t_0	MAR $\leftarrow$ PC	t_0, P_1, P_3
2.	Read the instruction at this address and write into instruction register	Fetch	t_1	IR $\leftarrow M[MAR]$	t_1, P_3, P_4, P_5, M_R
3.	Increment the program counter	Decode	t_2	PC $\leftarrow$ PC + 1	$t_2, IncrPC$
4.	Read operand from instruction register and write into memory address register	Execute	t_3	MAR $\leftarrow$ IR	t_3, P_0, P_1, P_2, P_3
5.	Read memory at this address and write into memory buffer register	Execute	t_4	MBR $\leftarrow$ M[MAR]	t_4, P_3, P_4, M_R
6.	Read memory buffer register and write into program counter	Execute	t_5	PC $\leftarrow$ MBR	t_5, P_0, P_1, P_4