Assignment 2

CMPT 215
Name: Yinsheng Dong
Student Number: 11148648
NSID: yid164
Due: July 27^{th}, 2017
Total: 65 marks

Problem 1.
(8 marks) Add the following, indicate if there is a type of overflow or carry over for a 8 bit binary.

i $12 + 10$

Solution:
12 in the 8-bit binary: 0000 1100
10 in the 8-bit binary: 0000 1010
0000 1100 (12)
+
0000 1010 (10)
=
0001 0110
It should be an unsigned overflow

ii $01001100 + 00111110$

Solution:
0100 1100
+
0011 1110
=
1 0000 1010
It should be a signed overflow

iii $10010000 + 11111111$

Solution:
1001 0000
+
1111 1111
=
1 1000 1111
It should be a singed overflow

iv $4 + -13$

Solution:
4 in the 8-bit binary: 0000 0100
13 in the 8 bit binary: 0000 1101
-13 in the binary: 1111 0011
0000 0100 (4)
+
1111 0011(-13)
=
1111 0111
It should be neither kind overflow

Problem 2.
(8 marks) Subtract the following, indicate if there is a type of overflow or carry over for a 8 bit binary.

i $00111010 - 00011111$

Solution:
0011 1010
-
0001 1111
=
0001 1011
It should be neither carry over or overflow

ii $9 - 10$

Solution:
9 in 8-bit binary: 0000 1001
10 in 8-bit binary: 0000 1010
-10 in 8-bit binary: 1111 0110
9 - 10 = 9 + (-10)
0000 1001 (9)
+
1111 0110 (-10)
=
1111 1111
It should be the signed overflow

iii $00001010 - 11111111$

Solution:
0000 1010 - 1111 1111 =
0000 1010
+
0000 0001 (by 2s complement)
=
0000 1011
It should be a carry over

iv $4 - 13$

Solution:
4 in 8-bit binary: 0000 0100
13 in 8-bit binary: 0000 1101
-13 in 8-bit binary: 1111 0011
4 - 13 = 4 + (-13)
0000 0100
+
1111 0011
=
1111 0111
It should be signed overflow

Problem 3.
(12 marks) Multiply the following, indicate if there is a type of overflow or carry over for a 8 bit binary.

i 00111010×00011111

> **Solution:**
> 0011 1010
> $\times$
> 0001 1111
> $=$
>   0011 1010
> $+$
>   0011 1010
> $+$
>   0011 1010
> $+$
> 0011 1010
> $+$
> 0011 1010
> $=$
> 111 0000 0110

ii 9×10

Solution:
9 in 8-bit binary: 0000 1001
10 in 8-bit binary: 0000 1010

0000 1001
$\times$
0000 1010
=
0000 0000
+
0000 1001
+
0000 0000
+
0000 1001
=
0101 1010

iii 01100111 $\times$ 00001111

Solution:
0110 0111
$\times$
0000 1111
=
0110 0111
+
0110 0111
+
0110 0111
+
0110 0111
=
110 0000 1001

iv 11010100 × 11110101

Solution:
1101 0100
×
1111 0101
1101 0100
+
0000 0000
+
1101 0100
+
0000 0000
+
1101 0100
+
0000 0000
+
1101 0100
+
1101 0100
=
1100 1010 1110 0100

Problem 4.
(6 marks) Divide the following, indicate if there is a type of overflow or carry over for a 8 bit binary.

i 01001100/00000100

Solution:
The result should be 0001 0011

ii 10101111/00001111

Solution:
The result should be 0000 1011 r 1010

iii 11010100/11110101

Solution:
The result should be 0000 0000 r 1101 0100
because 1101 0100 is larger than 1101 0100, so it only could be divided by 0, and left the 1101 0100

Problem 5.
(8 marks) Convert the following to IEEE 754 single precision binary floating point representation for each of the following numbers.

i -3.96875

Solution:
Sign: 1
exponent: $127 + 1 = 128$ (1000 0000)
fraction: 111111
the final answer is 1100 0000 0111 1110 0000 0000 0000 0000

ii -1.5

Solution:
Sign: 1
exponent: $127 + 0 = 127$ (0111 1111)
fraction: 1
the final answer is 1011 1111 1100 0000 0000 0000 0000 0000

iii 1.1×10^{-126}

Solution:
Sign: 0
exponent: 0 (0000 0000)
fraction: 0000
the final answer is 0000 0000 0000 0000 0000 0000 0000 0000
This should be an underflow case

iv 2.8×10^{6}

Solution:
Sign: 0
exponent: 148 (1001 0010)
2.8×10^{6} should be 0100 1010 0010 1010 1011 1001 1000 0000

Problem 6.
(8 marks) Convert the following IEEE 754 single precision binary floating point values to $base_{10}$ number.

i 0100 0000 0100 0000 0000 0000 0000 0000

Solution:
Sign: 0 is positive
exponent: 1000 0000 is 128-127 = 1
fraction should be 0011
the final answer should be 3

ii 0100 0001 1010 0000 0000 0000 0000 0000

Solution:
Sign: 0 is positive
exponent: 1000 0011 is 131-127 = 4
fraction should be 10100.000...
the final answer should be 20

iii 1111 1111 1000 0000 0000 0000 0000 0000

Solution:
This question should be negative infinity because all exponent digit is 1, and the sign is 1

iv 1100 0001 0101 1010 0000 0000 0000 0000

Solution:
Sign: 1 is negative
exponent: 1000 0010 is 130 - 127 = 3
fraction should be 1101.1010...
the final answer is -13.625

Problem 7.

(10 marks) Fill out the following table for the following MIPS instructions, assume that it starting at address 4000.

```
loop:       ben   $s0,$s1,out
            sw    $s2,4($s1)
            addu  $s1,$s1,$t0
            j  loop
out:        ori  $t2,$s7,3
```

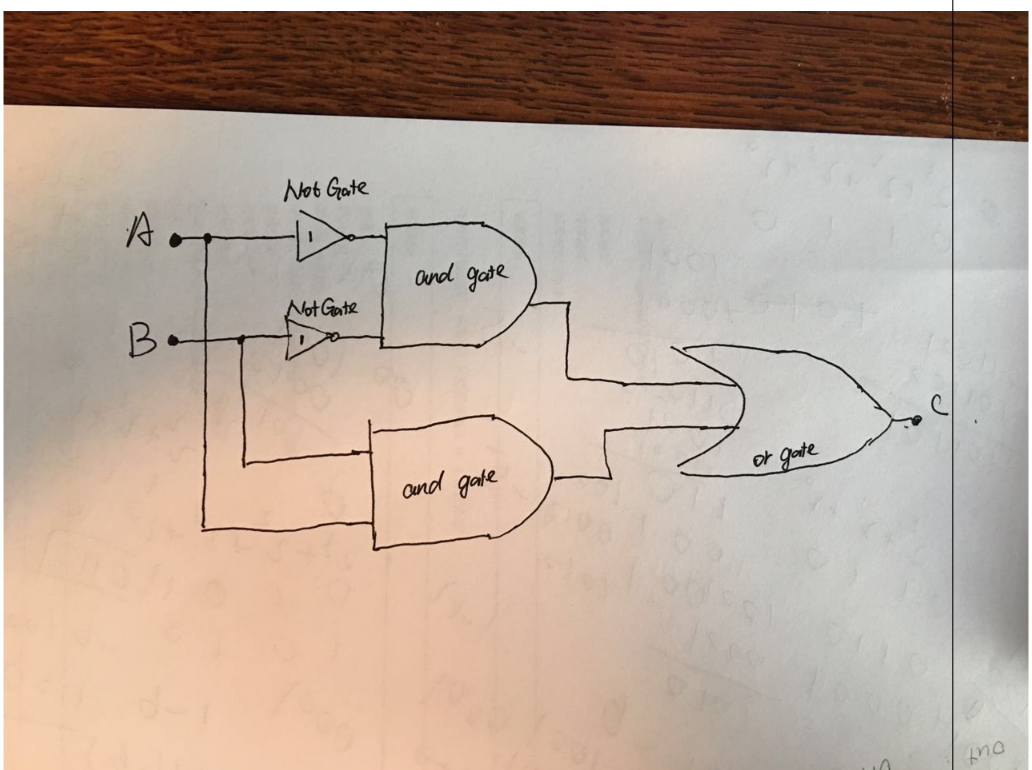
Instruction	Format	Fields					
		6 bits	5 bits	5 bits	5 bits	5 bits	6 bits
loop: bne $\$s0$,$\$s1$,out	I	000101	10000	10001	00000	00000	000010
sw $\$s2$,4($\$s1$),out	I	101011	10001	10010	00100	00000	000000
addu $\$s1$,$\$s1$,$\$t0$	R	000000	10001	01000	10001	00000	100001
j loop	J	000010	00000	00000	00000	01111	101000
out: ori $\$t2$,$\$s7$,3	I	001101	10111	01010	00000	00000	000011

Problem 8.

(5 marks) Design and show the truth table for Ex-Nor Gate using only NOT, AND and OR gates. EX-Nor game is a digital logic gate that is the reverse or complementary form of the Exclusive-OR function.

Solution:
The truth table is below, A and B is input, C is true if and only if A and B is same input to make EX-Nor gate

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



This is the flow gate that I designed
If A and B both 1, it will be (A * B) + (-A * -B) = (1 * 1) + (0 * 0) = 1
If A is 1 and B is 0, it will be (1 * 0) + (1 * 0) = 0
If A is 0 and B is 1, it will be (0 * 1) + (0 * 1) = 0
If A and B both 0, it will be (0 * 0) + (1 * 1) = 1
the result is if and only if the A and B is same, and C will be the true

Bonus:

Problem 9.
(3 marks) Name two universal Quantum circuits and one error correction gate used in Quantum computers.

Solution:
1. Depth-universal Quantum circuits
construct universal circuits whose depth is the same order as the circuits being simulated

2. Almost-size universal Quantum circuits
there is a log factor blow-up in the universal circuits constructed here

3. Controlled NOT (or C-NOT or CNOT) gate
is a quantum gate that is an important part of a quantum computer.