CS-2110 Quiz 2A

Alan Chiang

TOTAL POINTS

95 / 100

QUESTION 1

1 2's Complement 10 / 10

+ 10 **Sample Answer: No. You need 6 bits to represent 32 in unsigned binary and you need one more bit to represent the sign - totaling 7 bits.**

+ 0 **Graded**

QUESTION 2

2 Division with Bitwise Operators 10 / 10

+ 10 **Rightshift by 3**

+ 0 **Graded**

QUESTION 3

3 Toggling Bits 10 / 10

- 10 Sample Answer: Addition could produce an incorrect result if the contents of the desired bit is unknown this is why we must use XOR.

- 0 **Graded**

QUESTION 4

4 Bitwise Operators and Hexadecimal 35 / 40

+ 5 a) Unsigned Binary 00011100

+ 5 **a) Hexadecimal 0x1C**

+ 5 **b) Unsigned Binary 11111011**

+ 5 **b) Hexadecimal 0xFB**

+ 5 **c) Unsigned Binary 01001000**

+ 5 **c) Hexadecimal 0x48**

+ 5 **d) Unsigned Binary 11111111**

+ 5 **d) Hexadecimal 0xFF**

+ 0 Graded

Partial credit was given for the correct hex representations of incorrect binary answers.

QUESTION 5

5 Bitmasking 30 / 30

+ 10 **a) 4**

+ 10 **b) 1**

+ 10 **c) 1**

+ 0 Graded

CS-2110 Spring 2016 Quiz 02a

Name: _____ Section: _____

Short Answer: (___ / 30)

a) Using 2's Complement, can we express **35** with **6** bits? Explain why or why not.

No, because using 2's complement w/ 6 bits can express the range of numbers -2^5 to $+2^5-1$.

Positive $2^5-1 = 31$, and $35 > 31$, so no, 35 can't be expressed.

b) How can we use bitwise operators and constants to divide a number by **8**?

Rightshift the number by 3

unsigned binary → 1000 >> 3 = 0001

8/8 = 1

c) Why should we use bitwise operators instead of addition to toggle bits?

Because using addition to toggle bits can sometimes give unwanted/unexpected results.

If 0010 represents a steak dinner and 0100 represents lobster, then if a person accidentally double-orders and we are using addition to toggle, that person will get lobster instead of 2 steaks.

But using bitwise operators 0010 & 0010 will just give 0010.

CS-2110 Spring 2016 Quiz 02a

Name: Alan Chiang Section: 3:00 PM Klaus

Bitwise Operators and Hexadecimal: (____ / 40)

1. Evaluate the following bitwise operations and express your answer in **unsigned binary** and **hexadecimal**. All right shifts are **signed shifts.** **PLEASE EXPRESS BINARY ANSWERS WITH 8 BITS AND HEXADECIMAL NUMBERS SHOULD HAVE 0x FOLLOWED BY YOUR ANSWER.**

A 10
B 11
C 12
D 13
E 14
F 15

a) `~(11001000 ^ 00101011)`

11001000
11100011

Unsigned Binary: 11100011

Hexadecimal: 0xE3

b) `(00001001 << 3) | 11111011`

01001000
01001000
11111011

Unsigned Binary: 11111011

Hexadecimal: 0xFB

c) `01001011 & (10001110 >> 4)`

11111000
01001000

Unsigned Binary: 01001000

Hexadecimal: 0x48

d) `(~10100011 ^ 10100011) >> 2`

01011100
01011100
11111111

Unsigned Binary: 11111111

Hexadecimal: 0xFF

Bitmasking: (____ / 30)

2. Fill in the blanks that would make the following expressions true. All hexadecimal values are **8 bits**. All right shifts are **signed shifts.** You may express your answers in **decimal**.

00000001 << 4
00010000

a) `~(0x1 << __?__ ) & 0xBA = 10101010`

11101111
10111010

Answer: 4

00001110
11100011

b) `(0xE << __?__ ) | 0xE3 = 11111111`

Answer: 1

01110000
00111000
01101010

c) `(0x70 >> __?__ ) & 0x6A = 00101000`

Answer: 1

01101010