# C335 Homework #5

| | |
|---|---|
| Points: | : 40 points |
| Due Date: | : Mar. 24th |
| Submissions: | : For F-2-F students, hardcopy (type or write your solution clearly)<br>For online students, e-copy to Canvas |

## PART I (8 POINTS)

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

(A) X contains only one 0
(B) X contains an even number of 0s
(C) X when interpreted as an unsigned binary number is less than 4
(D) X when interpreted as a signed (two's complement) number is negative

## PART II (8 POINTS)

With x = 01011011 (bin) and y = 00001101(bin) representing two's complement signed integers, perform, showing all work:

(A) x + y
(B) x – y
(C) x * y
(D) x/y

## PART III (6 POINTS)

Do the unsigned multiply for 0011 * 0111, using the *Multiply Algorithm Version 3* (check the lecture notes). Show the contents of registers for multiplicand (4 bits) and product (8 bits) step by step.

## PART IV (6 POINTS)

Do the unsigned multiply for 0011 * 0111, using the *Booth's Algorithm* (check the lecture notes). Show the contents of registers for multiplicand (4 bits) and product (8 bits) step by step.

## PART V (6 POINTS)

In Lecture Notes c335-ALU-1.pdf, we discussed how we could modify the 4-bit ALU to support detecting overflow but did not give the proof. Now you are expected to prove you can detect overflow by simply check if Carry into MSB ! = Carry out of MSB.
**Hint:** you could use a table (like a truth table) to list all situations, then identify the cases that cause overflows

## PART VI (6 POINTS)

Do necessary modifications on the following 1 bit ALU, then use it to construct a 4-bit ALU that supports the SLT instruction.
**Hint:** read text appendix **B** section **B.5** to find clue.