113 計算機組織 Homework 3-1 Due date: YYYY/MM/DD.

- You must write down / type the calculations, or you won't get the scores.
- 請用手寫於 A4 紙上,並掃描上傳至 i 學園
- 請勿抄襲，抄襲者與被抄襲者單一分數取平均

1.(15%) Let the decimal numbers A = 57, B = -84, C = 0.8125

(a) Convert the decimal number C to binary format

(b) Compute A – B in 8-bit 2's complement, is it overflow? Why or why not?

(c) Compute A + B in 8-bit 2's complement, is it overflow? Why or why not?

2.(10%) When performing arithmetic addition and subtraction, overflow might occur. Fill in the blanks in the following table of overflow conditions for addition and subtraction.

Operation	Operand A	Operand B	Result indicating overflow
_____	≥ 0	≥ 0	(a) _____
_____	< 0	< 0	(b) _____
_____	≥ 0	< 0	(c) _____
_____	< 0	≥ 0	(d) _____

3.(20%) Prove that the overflow condition can be determined simply by checking to see if the CarryIn to the most significant bit of the result is not the same as the CarryOut of the most significant bit of the result. (Apply true table)

4.(5%) Assume -41 and 87 are signed 8-bit decimal integers stored in sign-magnitude format. Calculate -41 - 87. Is there overflow, underflow, or neither?

5.(5%) What decimal number does the bit pattern 0×AC000000 represent if it is a two's complement integer? An unsigned integer?

6.(10%) Sketch a 4-bits Adder-Subtractor-Nor-Zero_detection Circuit using 4 Full-Adder (FA) and 4 XOR gates. The input are ($B_3B_2B_1B_0$), ($A_3A_2A_1A_0$) and the signal ctrl controls the add/sub operation (ctrl = 00, 01, 10 for Adder/Subtractor/Nor). In your design try to use minimum number of the following basic logic gates (1-bit adders, AND, OR, INV, and XOR, MUX).

7.(15%) Using 4-bit carry-lookahead blocks to design an efficient 12-bit carry-lookahead adder.

(a)(5%) Draw the block diagram.

(b)(10%) Describe how signals are generated/processed step by step to determine the last Carryout (CarryOut11, $C_{in}12$) in the following example:

x: 1010 0011 0011

y: 0101 1110 1011

8.(10%) Booth's algorithm is an elegant approach to multiply signed numbers. It starts with the observation that with the ability to both add and subtract there are multiple ways to compute a product. The key to Booth's insight is in his classifying groups of bits into the beginning, the middle, or the end of a run of 1s. Is Booth's algorithm always better? Why does the Booth's algorithm work for multiplication of two's complement

signed integers?

9.(10%) Using a table similar to that shown in below, calculate the product of the decimal unsigned 8- bit integers 78 and 27 using the hardware described in Figure 1. You should show the contents of each register on each step.

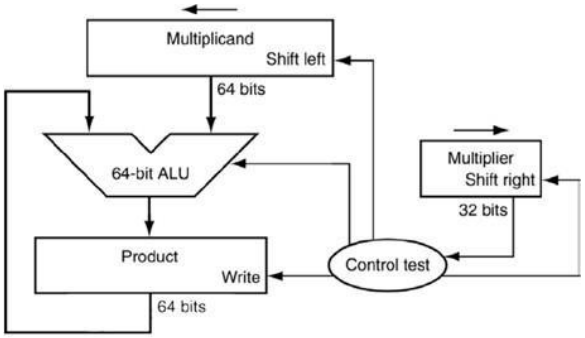

Figure 1. Multiplication hardware