NAME / ID#

Quiz 2
Fall 2019

Even

Question 1 (20 points) 7-bit registers are used in this question.

(a) **Convert the following two signed numbers to binary, employing the 2's complement representation:**

M = (+ 34)$_{10}$

	sgn	2^5	2^4	2^3	2^2	2^1	2^0
M represented on 7 bits: 34 = 32 + 2 =	0	1	0	0	0	1	0

N = (- 49)$_{10}$; N's magnitude |N| **= (+ 49)$_{10}$** = 32+16+1, represented on 7 bits = 011 0001 => **N = -49** = *2's complement of* (011 0001) = **100 1111**

OR:

34	2		49	2
17	0	2^0	24	1
8	1	2^1	12	0
4	0	2^2	6	0
2	0	2^3	3	0
1	1	2^4	1	1
0	1	2^5	0	1

(b) **Find the 2's complement of the signed binary number N, and** verify your calculations by converting your result to decimal**.**

2's complement of N (i.e., -N) = *2's complement of* (**100 1111**) = **011 0001**, which in decimal is 32+16+1=+49 => the result **011 0001** is **correct**, because its decimal value +49=-(-49)=-N; it makes sense, since by complementing -49, we negate its value and get +49.

(c) **Perform the following arithmetic operations in signed 2's complement representation, considering that a 7-bit adder is used to perform these operations; show the operations and the results (including intermediary steps), both in binary and, for verification, in decimal.**

S=M+N = (+ 34)$_{10}$ + (-49)$_{10}$

Operations for	verification in
decimal, but not	+34
converted from	-49
registers binary	-15

Carry:	0	0	0	1	1	1	0
M →	0	1	0	0	0	1	0
N →	1	0	0	1	1	1	1
S →	1	1	1	0	0	0	1

The result S = 111 0001 is a negative number, which in decimal is = $-2^6 + 2^5 + 2^4 + 2^0$ = -15,

OR, say – *x* = 111 0001. To convert 111 0001 (i.e., -*x*) to decimal, first we find |S|= *x*, by complementing 111 0001, i.e., *x* = - (- *x*) = *2's complement* (-x) = *2's complement* (111 0001) = 000 1110 +1 = 000 1111 = +15, so, S = -15 => it coincides with the result of operation performed in decimal -15 … which verifies S

Comment if 7 bits would be enough to correctly carry out that operation? Justify your answer

All 3 numbers M, N and S ⊂ [-64, +63], which is the range of signed numbers with 7 bits, => no overflow ➔ **7 bits are sufficient to perform S=M+N** with these numbers.

D = M - N = (34)$_{10}$ - (-49)$_{10}$ = (+34)$_{10}$ + (49)$_{10}$

Operations for	verification in
decimal, but not	34+
converted from	49
registers binary	83

Carry:	0	1	0	0	0	0	0
M →	0	1	0	0	0	1	0
-N →	0	1	1	0	0	0	1
D →	1	0	1	0	0	1	1

The result D = **101 0011** is a negative number, say D = – *x* = **101 0011**.

To convert **101 0011** (i.e., -*x*) to decimal, first we find |D|= *x*, by complementing 101 0011, i.e., *x* = - (- *x*) = *2's complement* (-x) = *2's complement* (101 0011) = 010 1100 +1 = 010 1101 = +45,

=> the result D in decimal - *x* = **-45** which is wrong since a negative number cannot result from the addition of 2 positive numbers!

Comment if 7 bits would be enough to correctly carry out that operation? Justify your answer

M - N = 83 > 63 i.e. it does not fit in [-64, +63], so **it cannot be represented with 7 bits only**.

Question 2 (5 points)

An 8-bit register contains the binary value 00011100. What is the BCD representation of this number?

		2^6	2^5	2^4	2^3	2^2	2^1	2^0	
Binary:	**0**	0	0	1	1	1	0	0	Decimal 16+8+4 = (2 8)$_{10}$
									BCD: 010 1000