

7-bit registers are used in this question.

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

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

M represented on 7 bits: $34 = 32 + 2 =$

0	1	0	0	0	1	0
---	---	---	---	---	---	---

N's magnitude represented on 7 bits = $(+49)_{10} = 32+16+1 =$

0	1	1	0	0	0	1
---	---	---	---	---	---	---

$\Rightarrow N = -49 = 2's \text{ complement } (011\ 0001) = 100\ 1111$

- (b) Find the 2's complement of the signed binary numbers M and N, and VERIFY YOUR CALCULATIONS BY CONVERTING YOUR TWO RESULTS TO DECIMAL.

2's complement of M (i.e., $-M$) = 2's complement (010 0010) = **101 1110**, which is a negative number, say $-x$. To convert $-x$ to decimal, we have to find the magnitude x of this number ($-x$) by complementing it: $x = -(-x) = 2's \text{ complement } (-x) = 2's \text{ complement } (101\ 1110) = 010\ 0010 = +34$
 \Rightarrow 2's complement of M in decimal = $-x = -34$, which makes sense, since complementing M = +34, we negate its value to -34.

2's complement of N (i.e., $-N$) = 2's complement (100 1111) = **011 0001**, which in decimal is +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.

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

Verification:	Carry:	0	0	0	1	1	1	0
+34	M \rightarrow	0	1	0	0	0	1	0
-49	N \rightarrow	1	0	0	1	1	1	1
-15	S \rightarrow	1	1	1	0	0	0	1

The result **S = 111 0001** is a negative number, say $-x = 111\ 0001$.

To convert **111 0001** (i.e., $-x$) to decimal, first we find $|S| = x$, by complementing **111 0001**, i.e., $x = -(-x) = 2's \text{ complement } (-x) = 2's \text{ complement } (111\ 0001) = 000\ 1110 + 1 = 000\ 1111 = +15$,
 \Rightarrow the result in decimal $-x = -15$... which was expected.

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

Verification	Carry:	0	1	0	0	0	0	0
34+	M \rightarrow	0	1	0	0	0	1	0
49	-N \rightarrow	0	1	1	0	0	0	1
83	D \rightarrow	1	0	1	0	0	1	1

The result **D = 101 0011** is a negative number, say $-x = 101\ 0011$.

To convert **101 0011** (i.e., $-x$) to decimal, first we find $|D| = x$, by complementing **101 0011**, i.e., $x = -(-x) = 2's \text{ complement } (-x) = 2's \text{ complement } (101\ 0011) = 010\ 1100 + 1 = 010\ 1101 = +43$,
 \Rightarrow the result in decimal $-x = -43$ instead of 83 as it was expected.

- d) Are there any overflows?

No overflow for S, but yes for D since the expected result +83 is out of the range of 7-bit signed numbers [-64,+64]

How can a computer detect overflow?

1. Overflow = Carry₇ xor Carry₆ or
2. Overflow = Sign A & Sign B & (Sign result)' + (Sign A)' & (Sign B)' & Sign result