COMPUTER ORGANIZATION AND ARCHITECTURE

ASSIGNMENT –3

1. Perform the arithmetic operations below with binary numbers and with negative numbers in signed 2's complement representation. Use seven bits to accommodate each number together with its sign. In each case, determine if there is an overflow by checking the carries into and out of the sign bit position.
    a. (+35) + (+40)
    b. (-35) + (-40)
    c. (-35) - (+40)

When two numbers of n digits each are added and the sum occupies n + 1 digits, we say that an overflow occurred.
A result that contains n + 1 bits cannot be accommodated in a register with a standard length of n bits.
The detection of an overflow after the addition of two binary numbers depends on whether the numbers are considered to be signed or unsigned. When two unsigned numbers are added, an overflow is detected from the end carry out of the most significant position.
In the case of signed numbers, the leftmost bit always represents the sign, and negative numbers are in 2' s complement form. When two signed numbers are added, the sign bit is treated as part of the number and the end carry does not indicate an overflow.
An overflow cannot occur after an addition if one number is positive and the other is negative, since adding a positive number to a negative number produces a result that is smaller than the larger of the two original numbers. An overflow may occur if the two numbers added are both positive or both negative.
An overflow condition can be detected by observing the carry into the sign bit position and the carry out of the sign bit position. If these two carries are not equal, an overflow condition is produced. If the two carries are applied to an exclusive-OR gate, an overflow will be detected when the output of the gate is equal to 1 .

a)
+35	0	100011
+40	0	101000
+75	1	001011

Last two carries are F=0 and E=1. F XOR E= 1.Hence overflow had occurred.

b)
-35	1	011101
-40	1	011000
-75	1	110101

Last two carries are F=1 and E=0. F XOR E= 1.Hence overflow had occurred.

c) –(35)-(+40) . Add 2's complement of 40 to 2' complement of 35. Same as above

2. Show the contents of registers E, A, Q, and SC during the process of multiplication of two binary numbers, 11111 (multiplicand) and 10101 (multiplier). The signs are not included.

Multiplicand B = 1 1 111 = $(31)_{10}$ $31 \times 21 = 651$

	E	A	Q	SC	
Multiplier in Q - -	0	00000	10101	101	Q = $(21)_{10}$
Q_n = 1, add B - - -		11111			
	0	11111			
shr EAQ - - - -		01111	11010	100	
Q_n = 0, shr EAQ - -		00111	11101	011	
Q_n = 1, add B - -		11111			
	1	00110			
shr EAQ - - - -	0	10011	01110	010	
Q_n = 0, shr EAQ - -		01001	10111	001	
Q_n = 1, add B - -		11111			
	1	01000			
shr EAQ - - - -		1010001011		000	
		$(651)_{10}$			

3. Show the step-by-step multiplication process using Booth algorithm when the following binary numbers are multiplied. Assume 5-bit registers that hold signed numbers. The multiplicand in both cases is + 15.

a. (+ 15) * (+ 13) b. (+ 15) * (- 13)

$(+15) \times (+13) = +195 = (0\ 011000011)_2$

BR = 01111 (+15); $\overline{BR}$ + 1 = 10001 (–15); QR = 01101 (+13)

$Q_n Q_{n+1}$		AC	QR	Q_{n+1}	SC
	Initial	00000	01101	0	101
1 0	Subtract BR	10001			
		10001			
	ashr ———	11000	10110	1	100
0 1	Add BR	01111			
		00111			
	ashr ———	00011	11011	0	011
1 0	Subtract BR	10001			
		10100			
	ashr ———	11010	01101	1	010
1 1	ashr ———	11101	00110	1	001
0 1	Add BR	01111			
		01100			
	ashr ———	00110	00011	0	000
		+195			

$(+15) \times (-13) = -195$ $= (1100\ 111101)_{2\text{'s comp.}}$

BR = 0 11111 (+15); $\overline{BR}$ + 1 = 10001 (–15); QR = 10011 (–13)

$Q_n Q_{n+1}$		AC	QR	Q_{n+1}	SC
	Initial	00000	10011	0	101
1 0	Subtract BR	10001			
		10001			
	ashr ———	11000	11001	1	100
1 1	ashr ———	11100	01100	1	011
0 1	add BR	01111			
		01011			
	ashr ———	00101	10110	0	010
0 0	ashr ———	00010	11011	0	001
1 0	Subtract BR	10001			
		10011			
	ashr ———	11001	11101	1	000
		–195			

4. Show the contents of registers E, A, Q, and SC during the process of division of
(a) 10100011 by 1011; (b) 00001111 by 0011. (Use a dividend of eight bits.)

a)

$$\frac{10100011}{1011} = 1110 + \frac{1001}{1011} \qquad \frac{163}{11} = 14 + \frac{9}{11}$$

B = 1011 $\bar{B} + 1 = 0101$ DVF = 0

	E	A	Q	SC
Dividend in AQ - - - -	0	1010	0011	100
shl EAQ - - - - -	1	0100	0110	
add $\bar{B} + 1$, suppress carry - -		0101		
$E = 1$, set Q_n to 1 - - - -	1	1001	0111	011
shl EAQ - - - - - -	1	0010	1110	
add $\bar{B} + 1$, suppress carry -		0101		
$E = 1$, set Q_n to 1 - - - -	1	0111	1111	010
shl EAQ - - - - - -	0	1111	1110	
add $\bar{B} + 1$, carry to E - -		0101		
$E = 1$, set Q_n to 1 - - - -	1	0100	1111	001
shl EAQ - - - - - -	0	1001	1110	
add $\bar{B} + 1$, carry to E - - -		0101		
$E = 0$, leave $Q_n = 0$ - - - -	0	1110	1110	
add B - - - - - - -		1011		
restore remainder - -	1	1001 (remainder)	1110 (quotient)	000

b)

$$\frac{1111}{0011} = 0101$$

B = 0011 $\quad \bar{B} + 1 = 1101$

	E	A	Q	SC
Dividend in Q, A = 0 - - - -		0000	1111	100
shl EAQ - - - - - -	0	0001	1110	
add $\bar{B} + 1$ - - - -		1101		
E = 0, leave $Q_n = 0$ - - - -	0	1110	1110	
add B - - - - - -		0011		
restore partial remainder - -	1	0001		011
shl EAQ - - - - - -	0	0011	1100	
add $\bar{B} + 1$ - - - - -		1101		
E = 1, set Q_n to 1 - - - - -	1	0000	1101	010
shl EAQ - - - - - - -	0	0001	1010	
add $\bar{B} + 1$ - - - - -		1101		
E = 0, leave $Q_n = 0$ - - - - -	0	1110	1010	
add B - - - - - -		0011		
restore partial remainder - - -	1	0001		001
shl EAQ - - - - - - - -	0	0011	0100	
add $\bar{B} + 1$ - - - - - -	1101			
E = 1, set Q_n to 1 - - - - -	1	0000	0101	000
		remainder	quotient	