Representation of signed no.s binary arithmetic in computers:

- Two ways of rep signed no.s
  1. Sign Magnitude form
  2. Complemented form
- Two complimented forms
  1. 1's compliment form
  2. 2's compliment form

Advantage of performing subtraction by the compliment method is reduction in the hardware.(instead of addition & subtraction only adding ckt's are needed.)
i.e, subtraction is also performed by adders only.

Instead of subtracting one no. from other the compliment of the subtrahend is added to minuend. In sign magnitude form, an additional bit called the sign bit is placed in front of the no. If the sign bit is 0, the no. is +ve, If it is a 1, the no is _ve.

Ex:

0	1	0	1	0	0	1

↓

Sign bit =+41 magnitude

↑

1	1	0	1	0	0	1

= -41

Note: manipulation is necessary to add a +ve no to a –ve no

Representation of signed no.s using 2's or 1's complement method:

If the no. is +ve, the magnitude is rep in its true binary form & a sign bit 0 is placed in front of the MSB.I f the no is _ve , the magnitude is rep in its 2's or 1's compliment form &a sign bit 1 is placed in front of the MSB.

Ex:

Given no.	Sign mag form	2's comp form	1's comp form
01101	+13	+13	+13
010111	+23	+23	+23
10111	-7	-7	-8
1101010	-42	-22	-21

Special case in 2's comp representation:

Whenever a signed no. has a 1 in the sign bit & all 0's for the magnitude bits, the decimal equivalent is -2^n , where n is the no of bits in the magnitude .
Ex: 1000= -8 & 10000=-16

Characteristics of 2's compliment no.s:

Properties:

1. There is one unique zero
2. 2's comp of 0 is 0
3. The leftmost bit can't be used to express a quantity . it is a 0 no. is +ve.
4. For an n-bit word which includes the sign bit there are (2^{n-1}-1) +ve integers, 2^{n-1} –ve integers & one 0 , for a total of 2^n uniquestates.
5. Significant information is containd in the 1's of the +ve no.s & 0's of the _ve no.s
6. A _ve no. may be converted into a +ve no. by finding its 2's comp.

Signed binary numbers:

Decimal	Sign 2's comp form	Sign 1's comp form	Sign mag form
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0011	0011	0011
+0	0000	0000	0000

-0	--	1111	1000
-1	1111	1110	1001
-2	1110	1101	1010
-3	1101	1100	1011
-4	1100	1011	1100
-5	1011	1010	1101
-6	1010	1001	1110
-7	1001	1000	1111
8	1000	--	--

Methods of obtaining 2's comp of a no:

- In 3 ways

1. By obtaining the 1's comp of the given no. (by changing all 0's to 1's & 1's to 0's) & then adding 1.
2. By subtracting the given n bit no N from 2^n
3. Starting at the LSB , copying down each bit upto & including the first 1 bit encountered , and complimenting the remaining bits.

Ex: Express -45 in 8 bit 2's comp form

+45 in 8 bit form is 00101101

I method:

1's comp of 00101101 & the add 1

00101101

11010010

+1

__ – – – – – – – – – –

11010011 is 2's comp form

II method:

Subtract the given no. N from 2^n

2^n = 100000000

Subtract 45= -00101101

+1

__ – – –

11010011 is 2's comp

III method:

Original no: 00101101

Copy up to First 1 bit 1

Compliment remaining : 1101001

____

bits 11010011

Ex:

-73.75 in 12 bit 2‘compform

I method

01001001.1100

10110110.0011

+1

____

10110110.0100 is 2‘s

II method:

2^8 = 100000000.0000

Sub 73.75=-01001001.1100

____

10110110.0100 is 2‘s comp

III method :

Orginalno : 01001001.1100

Copy up to 1‘st bit 100

Comp the remaining bits: 10110110.0

____

10110110.0100

2’s compliment Arithmetic:

- The 2‘s comp system is used to rep –ve no.s using modulus arithmetic . The word length of a computer is fixed. i.e, if a 4 bit no. is added to another 4 bit no . the result will be only of 4 bits. Carry if any , from the fourth bit will overflow called the Modulus arithmetic.

  Ex:1100+1111=1011
- In the 2‘s compl subtraction, add the 2‘s comp of the subtrahend to the minuend . If there is a carry out , ignore it , look at the sign bit I,e, MSB of the sum term .If the MSB is a 0, the result is positive.& it is in true binary form. If the MSB is a ` (carry in or no carry at all) the result is negative.& is in its 2‘s comp form. Take its 2‘s comp to find its magnitude in binary.

Ex: Subtract 14 from 46 using 8 bit 2‘s comp arithmetic:

+14 = 00001110

-14 = 11110010 2‘s comp

+46 = 00101110

-14 =+11110010 2‘s comp form of -14

____ _____________

-32 (1)00100000 ignore carry

Ignore carry , The MSB is 0 . so the result is +ve. & is in normal binary form. So the result is +00100000=+32.

EX: Add -75 to +26 using 8 bit 2's comp arithmetic

+75 = 01001011
-75 =10110101 2's comp

+26 = 00011010
-75 =+10110101 2's comp form of -75

-49 11001111 No carry

No carry , MSB is a 1, result is _ve & is in 2's comp. The magnitude is 2's comp of 11001111. i.e, 00110001 = 49. so result is -49

Ex: add -45.75 to +87.5 using 12 bit arithmetic

+87.5 = 01010111.1000
-45.75=+11010010.0100

-41.75 (1)00101001.1100 ignore carry
MSB is 0, result is +ve. =+41.75

1's compliment of n number:

- It is obtained by simply complimenting each bit of the no,.& also , 1's comp of a no, is subtracting each bit of the no. form 1.This complemented value rep the –ve of the original no. One of the difficulties of using 1's comp is its rep o f zero. Both 00000000 & its 1's comp 11111111 rep zero.
- The 00000000 called +ve zero& 11111111 called –ve zero.

Ex: -99 & -77.25 in 8 bit 1's comp

+99 = 01100011
-99 = 10011100

+77.25 = 01001101.0100
-77.25 = 10110010.1011

1's compliment arithmetic:

In 1's comp subtraction, add the 1's comp of the subtrahend to the minuend. If there is a carryout , bring the carry around & add it to the LSB called the **end around carry.** Look at the sign bit (MSB) . If this is a 0, the result is +ve & is in true binary. If the MSB is a 1 (carry or no carry), the result is –ve & is in its is comp form .Take its 1's comp to get the magnitude inn binary.

Ex: Subtract 14 from 25 using 8 bit 1's EX: ADD -25 to +14

25	=	00011001	+14	= 00001110
-45	=	11110001	-25	=+11100110
+11		(1)00001010	-11	11110100
+1				
			No carry	MSB =1
		00001011		result=-ve=-11_{10}

MSB is a 0 so result is +ve (binary)

=+11_{10}

Binary codes

Binary codes are codes which are represented in binary system with modification from the original ones.

- Weighted Binary codes
- Non Weighted Codes

Weighted binary codes are those which obey the positional weighting principles, each position of the number represents a specific weight. The binary counting sequence is an example.

Decimal	BCD 8421	Excess-3	84-2-1	2421	5211	Bi-Quinary 5043210			5	0	4	3	2	1	0
0	0000	0011	0000	0000	0000	0100001		0		X					X
1	0001	0100	0111	0001	0001	0100010		1		X				X	
2	0010	0101	0110	0010	0011	0100100		2		X			X		
3	0011	0110	0101	0011	0101	0101000		3		X		X			
4	0100	0111	0100	0100	0111	0110000		4		X	X				
5	0101	1000	1011	1011	1000	1000001		5	X						X
6	0110	1001	1010	1100	1010	1000010		6	X					X	
7	0111	1010	1001	1101	1100	1000100		7	X				X		
8	1000	1011	1000	1110	1110	1001000		8	X			X			
9	1001	1111	1111	1111	1111	1010000		9	X		X				

Reflective Code

A code is said to be reflective when code for 9 is complement for the code for 0, and