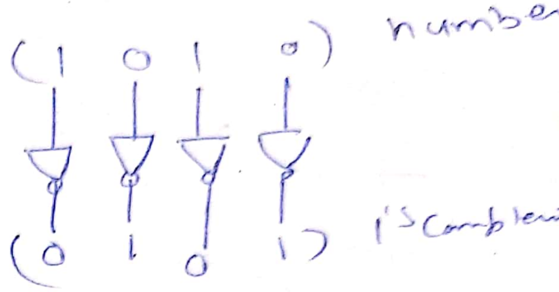
KIET Group of Institutions
Department of Information Technology
COURSE B.Tech., 3rd SEM,
Computer Organization and Architecture (KCS-302)
Session 2020-21

1's and 2's Complement

1's Complement can be easily obtained by just flipping each digit of number.

$1010 \xrightarrow{\text{1's Complement}} 0101$



2's Complement = 1's Complement +1

Ex:- $(10111010)_2 \xrightarrow{\text{2's Complement}} \overbrace{01000101}^{\text{1's Compliment}} + 1$

$= 01000110$ Ans

Use of complements → (i) Simplifying Subtraction operation
(ii) for logical manipulation.

r's Complement → radix Complement
(r-1)'s Complement → diminished Radix Complement

Signed Binary Numbers :-

→ Positive numbers including zero can be represented as unsigned numbers.

→ Due to hardware limitations, minus sign can't be used in computer instructions.

→ Everything should be represented in bits.

There are 3 different ways to represent it :-

(i) Signed-Magnitude Representation :-

→ Binary digit at left most position (MSB) or Most Significant bit shows the sign of number.

→ (+) → 0 & (−) is 1.

For example :- Representation of (−9) in 8-bits.

(+9) = 00001001 → magnitude

So, (−9) = 10001001 → MSB (Sign bit)

(ii) (Signed) 1's Complement Representation :-

→ Flip all the bits of positive number including sign bit.

(+9) = 00001001 → flip all bits

So, (−9) = 11110110

(iii) (Signed) 2's Complement Representation :-

→ 2's Complement of positive number including sign bit.

(+9) = 00001001

(−9) = 11110111

Data Representation

- Magnitude
  - ~~+ve~~ Unsigned
    (+ve binary no.)
    (drawback :- -ve rep.)
  - Signed
    (+ve & -ve both)
    [(drawback :- (-0))]
- Complement
  - 1's Complement
    (+ve & -ve both)
  - 2's Complement
    (+ve & -ve both)

⇒ In all 4 representation, +ve numbers are represented in same way.

Unsigned → ((-6) → Can't Present)

Range :- Unsigned → 0 to $2^n - 1$

Signed → $-(2^{n-1}-1)$ to $+(2^{n-1}+1)$

Ex → $n=4$ → -7 to $+7$.

1's Complement → $-(2^{n-1}-1)$ to $+(2^{n-1}-1)$

2's Complement → -2^{n-1} to $(2^{n-1}-1)$

where n = no. of bits to represent.

Values in different systems

$b_3 b_2 b_1 b_1$	Sign & Magnitude	1's Complement	2's Complement
0 111	+7	+7	+7
0 110	+6	+6	+6
0 101	+5	+5	+5
0 100	+4	+4	+4
0 011	+3	+3	+3
0 010	+2	+2	+2
0 001	+1	+1	+1
0 000	+0	+0	+0
1 000	−0	−7	−8
1 001	−1	−6	−7
1 010	−2	−5	−6
1 011	−3	−4	−5
1 100	−4	−3	−4
1 101	−5	−2	−3
1 110	−6	−1	−2
1 111	−7	−0	−1

Addition:-

① Simple binary Addition:- $0+0=0$ $1+0=0+1=1$

$1+1=10$

$1+1+1=11$

$$\begin{array}{rl} 7 = & 0111 \\ +\ 3 = & 0011 \\ \hline & 1010 \\ \hline \end{array} = (10)_{10}$$

$$\begin{array}{rl} 9 = & 1001 \\ +\ 7 = & 0111 \\ \hline & 10000 \\ \hline \end{array} = (16)_{10}$$

② Signed Addition:- (In case of overflow, here the register ~~will~~ may not be able to store all bits in result.)

(i) For Unsigned numbers (representation):-

⇒ Range:- 0 to $(2^n - 1)$ where

n = no. of bits (Size)

if $n=4$; range ⇒ 0 to +15

⇒ -ve numbers can't be represented in this case.

$$\begin{array}{rl} +6 = & 0110 \\ +3 = & 0011 \\ \hline & 1001 \\ \hline \end{array} = 9 \Rightarrow \text{(within Range)}$$

⇒ Correct output

$$\begin{array}{rl} +8 = & 1000 \\ +9 = & 1001 \\ \hline & \boxed{1}\underbrace{0001}_{\text{(Stored)}} \\ \hline \end{array}$$

(overflow) (Stored)

⇒ Actual Answer should be 17, but is not in Range.

∵ 17 > 15, So, overflow occurs

Track ⇒ Overflow will occur if ⇒

(i) Actual Answer is not in Range.

(ii) End carry is generated.

(17) Sign & Magnitude :-

Range :- $-(2^{(n-1)}-1)$ to $+(2^{(n-1)}-1)$

if $n=4$; -7 to $+7$

```
+6 = 0 110
+1 = 0 001
-----------
     0 111   => (+7) ✓
```

```
+6 = 0 110
+7 = 0 111
-----------
    0[1]101  => +13 (out of Range)
                      ↓
                   Overflow
```

↓ Overflown bit

```
+3 = 0 011
-2 = 1 010
-----------
     0 001   = (+1) ✓
```

```
-3 = 1 011
+2 = 0 010
-----------
     1 001   = (-1) ✓
```

```
-6 = 1 110
-1 = 1 001
-----------
     1 111   = (-7) ✓
     ↑
    Sign
    bit
```

```
-6 = 1 110
-7 = 1 111
-----------
   1[1]101   = -13 (out of Range)
                       ↓
                    Overflow
                    /Underflow
```

Overflown bit (Can't be stored)

Rules:- (i) Sign bit is MSB (Most Significant bit).

(ii) 0 → +ve, 1 → -ve] → if Sign bits are

Sign bit will not be added.

Same ↓ Result Sign will be Same. The numbers (magnitude) will be added.

different ↓ Result's sign will be the sign of larger number. Magnitude will be subtracted using binary subtraction.

Track:- in sdm
(i) If Result is out of Range.
(ii) If there is an extra bit in result after the addition of magnitudes.
Overflow.

(iii) 1's Complement:-

Range:- $(-(2^{(n-1)}-1)$ to $+(2^{(n-1)}-1))$

if n=4; ⇒ (+7 to −7)

$\boxed{0\ 0}$
+5 = 0101
+2 = 0010
0111 = (+7) ✓

$\boxed{0\ 1}$
+5 = 0101
+6 = 0110
1011 ⇒ (−4) ✗
⇓
Actual Result should be +11; but it will be out of Range. (Overflow

$\boxed{1\ 1}$
+5 = 0101
−1 = 1110
10011
→ 1
0100 ⇒ (+4) ✓

$\boxed{0\ 0}$
−5 = 1010
+1 = 0001
1011
⇒ −(0100) = (−4) ✓

10
−6 = 1001
−1 = 1110
10111
→ 1
1000
= (−7) ✓

−6 = 1001
−5 = 1010
10011
→ 1
0100 = (+4) ✗
(Stored)
⇓
Overflow / (Underflow)

⇒ Actual Result (−11) is Out of Range.

In case of 1's Complement:-

(i) MSB is taken with -ve magnitude.

(ii) -ve no. are represented with the 1's comple. of +ve equivalent.

(iii) If ~~Same sig~~ the result is out of Range,

or

If 2 +ve no. (MSB=0) $\xrightarrow{\text{result}}$ -ve no. (MSB=1)

or

If 2 -ve no. (MSB=1) $\xrightarrow{\text{result}}$ +ve no. (MSB=0) (underflow)

∴ then there will be overflow.

(iv) 2's Complement:-

Range:- $-2^{(n-1)}$ to $+(2^{(n-1)}-1)$

if $n=4$; $(-8 \text{ to } +7)$.

$$\begin{array}{rl} & \boxed{0\,0} \\ +5 = & \boxed{0}101 \\ +2 = & \boxed{0}010 \\ \hline & \boxed{0}111 \end{array} \quad (+7)$$

$$\begin{array}{rl} & \boxed{0\,1} \\ +5 = & \boxed{0}101 \\ +6 = & \boxed{0}110 \\ \hline & \boxed{1}011 \end{array} \quad = (-5) \times \text{ (Overflow)}$$

$$\begin{array}{rl} & \boxed{0\,0} \\ +5 = & 0101 \\ -6 = & 1010 \\ \hline & 1111 \end{array}$$

$\Rightarrow -(0001)$

$= (-1)$

$$\begin{array}{rl} & \boxed{1\,1} \\ +6 = & 0110 \\ -5 = & 1011 \\ \hline & \boxed{1}0001 \end{array}$$

$\boxed{1}$ → Discarded

$0001 = (+1)$

	1 1 1	
-6 =	1 0 1 0	
-1 =	1 1 1 1	
	$\cancel{1}$ 1 0 0 1	⇒ $-8+1$ ⇒ (-7)

	1 0	
-6 =	1 0 1 0	
-7 =	1 0 1 1	
	$\cancel{1}$ 0 1 0 1	⇒ $(+5)$ ✗ Overflow/(Underflow)

Rules:- (i) Sign bit is MSB, i.e., it's magnitude is added with -ve sign.

(ii) -ve no. are represented by 2's complement of their +ve equivalent.

(iii) Overflow :-

(i) if the result is not in range.

(ii) If 2 same signe no.s. result in differen sign.

(iii) If carry in & carry out of final bit is not same.

01 — Overflow

10 — (Overflow). Underflow.

Observation:- (i) Underflow is special case of overflow, where the ~~mag~~ results ~~was~~ falls below the minimum value, a no. system can represent.

⇒ (ii) When 2 different sign numbers are added in any representation, then no overflow / underflow occurs.

⇒ (iii) In case of unsigned number addition, no underflow occurs, only overflow can be seen.

Subtraction:-

(i) Simple Binary Subtraction:-

1 − 1 = 0   0 − 0 = 0
1 − 0 = 1   0 − 1 = 1 (with 1 borrow)
10 − 1 = 1

```
  1011
- 1001
  ----
  0010
```

```
  1010
- 0111
  ----
  0011
```

⇒ If 0 is subtracted by 1, then we have to take borrow from next bit.

⇒ If that next bit is 0; then we have to go to next from that bit & so on.

⇒ If we get borrow from any bit, that bit becomes 0 & borrow is propagated to previous bit & the previous bit, i.e., 0 becomes 10.

⇒ If borrow is propagated from 10, then 1 is remained in that bit & 1 is propagated to previous bit for ~~carr~~ borrow purpose.

Extend borrow

```
      1110   ⇒ 14
(−)   1111   ⇒ 15
    ------
    1 1111 b
```

⇓ Or

(Answer in 2's Complement) = (−1).

(ii) Sign & Magnitude:- (Same as Addition)

```
+6 = 0 : 110
−1 = 1 : 001
    ---------
     0 : 101
    ---------
  = (+5)
```

```
+5 = 0  101
−7 = 1  111
-----------
     ⇓
−7 = 1 : 111
+5 = 0 : 101
------------
     1  010   = (−2)
```

Subtraction using 1's Complement →

(a.) $X = 1010100$ Perform $(X - Y)$ & $(Y - X)$

$Y = 1000011$

$X - Y = X + (-Y)$

$$
\begin{array}{rr}
X = & 1010100 \\
\text{1's Com. of } Y = & 0111100 \\
\hline
\text{SUM} = & \boxed{1}0010000 \\
\text{(End-around Carry)} & +1 \\
\hline
X - Y = & 0010001
\end{array}
$$

(Answer)

$Y - X = Y + (-X)$

$Y = 1000011$
1's Comp. of $X = 0101011$
Sum $= 1101110$ → no end carry generated.

Answer: $(Y-X) = -$(1's Complement of 1101110) $= -0010001$

Subtraction using 2's Complement :-

$X = 1010100$ $Y = 1000011$

$X = 1010100$
2's Comp. of $Y = 0111101$
Sum ⇒ 10010001
Discarded carry 2^7 ⇒ (−) 10000000
$X - Y = 00010001$ (Answer)

$Y - X = ?$ $Y = 1000011$
2's Complement of $X = 0101100$
Sum ⇒ 1101111

There is no end carry.

Ans:- $Y - X = -$(2's Complement of 1101111) $= -0010001$

Step-1:- Find 2's complement of number to be subtracted.

Step-2:- Perform the addition.

Step-3:- If end carry is generated, then the result is positive & in its true form.

The answer will be The number after simply neglecting or discarding the carry.

8

If carry is not generated, the result is negative & in its 2's complement form.

In case of 1's Complement →

(i) find 1's Complement of the number to be subtracted.

(ii) Perform addition.

(iii) If final carry is generated:
- → result is positive.
- → Add this carry to the sum.

If end carry is not generated:
- → result is negative.
- → Answer will be its one's complement.