Number Systems

By

P. K. Roy

What is the purpose?

- A set of values used to represent different quantities is known as ***Number System***.
- Computer represents all kinds of data and information in terms of numbers (Low Level Language).
- Are of 2 types –
    1. Non-positional number systems – ancient technique

       e.g. Roman number system
    2. Positional number systems – finite digits represented by base or radix.

       e.g. Decimal, binary, etc.

Positional Number System

Can be classified into two categories -

1. High Level Numbers – Decimal
2. Low Level Numbers – Binary, Octal, Hexadecimal

Conversion :

High level to Low level => Division by Radix of Low level number

Low level to High level => Multiplication Radix of Low level number

(***See The Board…….***)

Binary Numbers

1. ***Signed –***

MSB		LSB
Sign Bit 0 – positive 1 - negative	Magnitude (all bits except sign bit)	

e.g. +15 => 0000 1111 (considering 8-bit representation)

-15 => 1000 1111

2. ***Unsigned –***

MSB		LSB
Magnitude (all bits including sign bit)		

Cntd…

Two additional representations:

1. 1’s complement:

    e.g. +15 => 0000 1111

    - 15 => 1111 0000 (complement of +15)

2. 2’s complement :

    e.g. +15 => 0000 1111

    - 15 => 1111 0000 (1’s complement of +15)

    + 0000 0001

    1111 0001 (2’s complement)

Range of Binary Numbers

- Considering 4 bit numbers,

  2^n = 16 representations are possible

  Where ***n*** is the no. of bits

For ***unsigned*** nos. 0 to 15

For ***signed*** nos. 0 to 7 => (+)ve

-1 to -8 => (-)ve 2's complement form

-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7

Hence the range of signed nos. can be represented by

$$-2^{n-1} \text{ to } +2^{n-1}-1$$

Where ***n*** is the no. of bits

Binary Arithmetic

Addition

=> 1+1 = 0 with carry 1; 1+ 0 =1 ; 0 + 1 = 1 ; 0 + 0 = 0

Subtraction

=> 1-1 = 0 ; 1- 0 = 1; 0 – 0 = 0; 10 - 1 = 1 with borrow 1

Multiplication

=> 1 × 1 = 1 ; 1 × 0 = 0 ; 0 × 1 = 0; 0 × 0 = 0

Division

=> 1 ÷ 1 = 1 ; 1 ÷ 0 => not allowed ; 0 ÷ 0 => not allowed ; 0 ÷ 1= 0

(Self Review)

Overflow

- Overflow occurs when two numbers are added or subtracted and the correct result is a number that is outside of the range of allowable numbers(or beyond the capacity of storage registers).

Example:

- 254 + 10 = 264 (>255); overflow in unsigned 8-bit.
- -100 – 30 = -130(<-128); overflow in signed 8-bit.

Overflow detection

Addition:

Adding same sign numbers and the result with different sign => overflow.

No overflow in case if the two numbers have different sign.

e.g.   +120   =>   0111 1000

   +65   =>   0100 0001

   + 185   =>   1011 1001

As the sign bit is 1, i.e. negative => overflow

Cntd…

Subtraction:

Minuend – subtrahend = difference

If sign(difference) == sign(minuend) => no overflow

If sign(difference) == sign(subtrahend) => overflow.

```
e.g.    -77     =>   1011 0011
        -122    =>   1000 0110
      ------------------------------
        -199    =>   1 0011 1001  => 0011 1001 (for 8-bt register)
```

As the sign bit is 0, i.e. positive => overflow

Floating Point Number Representation

Are of 2 types-

1. IEEE 754 Single-precision (32 bits)
2. IEEE 754 Double-precision (64 bits)

1. Single – precision : $\pm 1.Mx2^{(e' = e + 127)}$

Sign bit	Exponent	Mantissa
Sign bit (31)	8 bit exponent in Excess -127 Representation (30 - 23)	Significand/Mantissa (22 - 0)

Cntd…

- **Double – Precision :** $\pm 1.Mx2^{(e'=e+1023)}$

Sign	Exponent	Mantissa
Sign bit (63) 0 – positive 1 - Negative	11 bit Exponent in Excess – 1023 form (62 – 52)	52 bit Mantissa (51 - 0) or Significand

Normalized Floating – Point Number

- MSB of the mantissa is not ‘0’

  e.g. $(11001)_2$

- Non-normalized numbers can be converted into normalized form by removing the leading 0s of the mantissa.

  e.g. 0.000110×2^{-6} => 0.110×2^{-9}

  0.00345×10^{4} => 0.345×10^{2}

Q+A

1. For signed nos. which bit represents the sign?

   a) LSB   b) MSB   c) Carry bit   d) None of these

2. Binary negative nos. can be represented by –

   a) Sign - Magnitude   b) 1’s complement

   c) 2’s complement   d) all of these

3. For 8-bit binary unsigned no. system, total no. of representations -

   a) 253   b) 254   c) 255   d) 256

4. For 8-bit unsigned binary no. system, the largest no. is –

   a) 253   b) 254   c) 255   d) 256

5. For 8-bit unsigned binary no. system, the smallest no. is –

   a) 253   b) 254   c) 255   d) none of these

6. For 8-bit binary signed no. system, total no. of representations –

   a) 253   b) 254   c) 255   d) 256

7. For 8-bit signed binary no. system, the largest positive no. is –

a) 125 b) 126 c) 127 d) none of these

8. For 8-bit signed binary no. system, the smallest negative no. is –

a) -125 b) -126 c) -127 d) -128

9. Overflow may occur in addition when –

a) the two numbers have different sign

b) the two numbers have same sign

c) none of these

d) both a & b

10. Decimal nos. are positional

a) True b) False

11. Binary nos. can be considered as high level nos.

a) True b) False

12. In sign magnitude representation $(1111)_2$ means –

a) + 15 b) -15 c) +7 d) -7

13. In 2's complement representation $(1111)_2$ means –

a) + 15 b) -15 c) -1 d) +1

14. Today's computers only understand –

a) binary nos. b) hexadecimal nos.

c) decimal nos. d) octal nos.

15. . 9 is an octal number

a) True b) False

16. In binary arithmetic 1+1 =

a) 0 0 b) 0 1 c) 1 0 d) 1 1

17. In sign magnitude representation magnitude represents –

a) all bits   b) MSB   c) LSB   d) none of these

18. A borrow is generated when -

a) 1 - 1   b) 1 - 0   c) 10 - 1   d) 0 – 0

19. $(1010)_2$ + $(1010)_2$ for 4 bit binary no. system generates overflow

a) True   b) False

20. For a Hexadecimal no. system $(15)_{10}$ =

a) A   b) B   c) C   d) none of these

21. EAC should be added to the LSB of the result in –

a) 1' complement   b) 2's complement

c) both   d) none

22. A carry is generated when –

a) 0 + 0   b) 0 + 1   c) 1 + 0   d) 1 + 1

References

- Digital Circuits & Design – Salivahanan
- Digital Logic & Computer Design – Morris Mano