# Computer Organization and Architecture

Under Graduate Course
(B. Tech-Information Technology, 2nd Semester)
Jan 2020- June 2020

By


**Dr. Satish Kumar Singh**

Associate Professor
Indian Institute of Information Technology, Allahabad
Email:satish432002@gmail.com


## Contents

- Data Types
- Classification of data types
- Number System
- Conversion
- Coded Number System
- Alphanumeric Codes
- Complements
- Complement Arithmetic
- Problem of overflow and detection
- Normalization of floating point number
- Gray codes and conversion

## Data types

Binary information in digital computers is stored in memory or processor registers.

Data are numbers and other binary-coded information that are operated on to achieve required computational results.



## Classification of data types

Numbers used in arithmetic computations

Letters of the alphabet used in data processing

Other discrete symbols used for specific purposes

Binary number system is the most natural numbers systems to use in a digital computer.

# NUMBER SYSTEM

- A number system of base or radix r is a system that uses distinct symbols for r digits.

- Decimal
  - Uses the radix 10
  - The 10 symbols are 0,1,2,3,4,……,9
  - The decimal no 724.5 can be interpreted as

$$7 \times 10^2 + 2 \times 10^1 + 4 \times 10^0 + 5 \times 10^{-1}$$

- Binary
  - Uses the radix 2
  - The 2 digit symbols used are 0 and 1
  - The string of digit 101101 is interpreted as

$$1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 45$$



# CONTINUED….

- Octal
  - Uses the radix 8
  - The 8 symbols are 0,1,2,3,4,……,7
  - The octal no 736.4 can be interpreted as

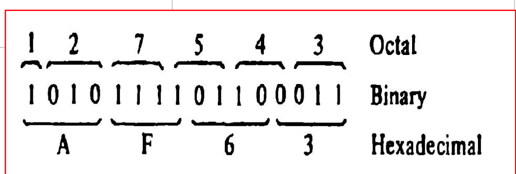
$$(736.4)_8 = 7 \times 8^2 + 3 \times 8^1 + 6 \times 8^0 + 4 \times 8^{-1}$$
$$= 7 \times 64 + 3 \times 8 + 6 \times 1 + 4/8 = (478.5)_{10}$$

- Hexadecimal
  - Uses the radix 16
  - The 16 symbols are 0,1,2,3,4,……,9,A,B,C,D,E and F
  - The equivalent decimal of F3 is

$$(F3)_{16} = F \times 16 + 3 = 15 \times 16 + 3 = (243)_{10}$$

## CONVERSIONS

- Conversion from decimal to its equivalent representation in the radix r system is carried out by separating the number into its integer and fraction parts and converting each separately.
- Binary, octal and hexadecimal conversion.






## DECIMAL TO BINARY

- Technique
  - Divide by two, keep track of the remainder
  - First remainder is bit 0 (LSB, least-significant bit)
  - $2^{nd}$ remainder is bit 0
  - Etc.

# CONTINUED..

- For example Conversion of decimal no 41.687 into binary.

**Integer = 41**

| | |
|---|---|
| 41 | |
| 20 | 1 |
| 10 | 0 |
| 5 | 0 |
| 2 | 1 |
| 1 | 0 |
| 0 | 1 |

**Fraction = 0.6875**

```
 0.6875
      2
 1.3750
    x 2
 0.7500
    x 2
 1.5000
    x 2
 1.0000
```

$(41)_{10} = (101001)_2$ $\quad$ $(0.6875)_{10} = (0.1011)_2$

$(41.6875)_{10} = (101001.1011)_2$



# OCTAL TO BINARY

- Technique
  - Convert each octal digit to a 3-bit equivalent binary representation
  - For example $705_8 = ?_2$

```
 7    0    5
 ↓    ↓    ↓
111  000  101
```

$705_8 = 111000101_2$

## HEXADECIMAL TO BINARY

- Technique
  - Convert each hexadecimal digit to a 4-bit equivalent binary representation
  - For example

$10AF_{16} = ?_2$

1 0 A F

0001 0000 1010 1111

$10AF_{16} = 0001000010101111_2$



## BINARY TO DECIMAL

- Technique
  - Multiply each bit by $2^n$, where $n$ is the "weight" of the bit
  - The weight is the position of the bit, starting from 0 on the right
  - Add the results
  - For example

$101011_2$ =>

$1 \times 2^0 = 1$
$1 \times 2^1 = 2$
$0 \times 2^2 = 0$
$1 \times 2^3 = 8$
$0 \times 2^4 = 0$
$1 \times 2^5 = 32$

$43_{10}$

## Octal to decimal

- Technique
  - Multiply each bit by $8^n$, where $n$ is the "weight" of the bit
  - The weight is the position of the bit, starting from 0 on the right
  - Add the results
  - For example

$724_8$ =>

| | |
|---|---|
| $4 \times 8^0$ = | 4 |
| $2 \times 8^1$ = | 16 |
| $7 \times 8^2$ = | 448 |
| | $468_{10}$ |



## Hexadecimal to decimal

- Technique
  - Multiply each bit by $16^n$, where $n$ is the "weight" of the bit
  - The weight is the position of the bit, starting from 0 on the right
  - Add the results
  - For example

$ABC_{16}$ =>

| | |
|---|---|
| $C \times 16^0 = 12 \times 1$ = | 12 |
| $B \times 16^1 = 11 \times 16$ = | 176 |
| $A \times 16^2 = 10 \times 256$ = | 2560 |
| | $2748_{10}$ |

# CONTINUED…

- Following table shows binary and decimal equivalent of few octal numbers

| Octal number | Binary-coded octal | Decimal equivalent | |
|---|---|---|---|
| 0 | 000 | 0 | |
| 1 | 001 | 1 | |
| 2 | 010 | 2 | Code |
| 3 | 011 | 3 | for one |
| 4 | 100 | 4 | octal |
| 5 | 101 | 5 | digit |
| 6 | 110 | 6 | |
| 7 | 111 | 7 | |
| 10 | 001 000 | 8 | |
| 11 | 001 001 | 9 | |
| 12 | 001 010 | 10 | |
| 24 | 010 100 | 20 | |
| 62 | 110 010 | 50 | |
| 143 | 001 100 011 | 99 | |
| 370 | 011 111 000 | 248 | |



# CONTINUED…

- Following table shows binary and decimal equivalent of few hexadecimal numbers

| Hexadecimal number | Binary-coded hexadecimal | Decimal equivalent | |
|---|---|---|---|
| 0 | 0000 | 0 | |
| 1 | 0001 | 1 | |
| 2 | 0010 | 2 | |
| 3 | 0011 | 3 | |
| 4 | 0100 | 4 | |
| 5 | 0101 | 5 | |
| 6 | 0110 | 6 | Code |
| 7 | 0111 | 7 | for one |
| 8 | 1000 | 8 | hexadecimal |
| 9 | 1001 | 9 | digit |
| A | 1010 | 10 | |
| B | 1011 | 11 | |
| C | 1100 | 12 | |
| D | 1101 | 13 | |
| E | 1110 | 14 | |
| F | 1111 | 15 | |
| 14 | 0001 0100 | 20 | |
| 32 | 0011 0010 | 50 | |
| 63 | 0110 0011 | 99 | |
| F8 | 1111 1000 | 248 | |

# Coded Number System

- Binary code is a group of n bits that assume up to 2^n distinct combinations of 1's and 0's with each combination representing one element of the set that is being coded.
- Hence set of 8 elements requires 3-bit code ,a set of 16 elements requires a 4-bit code and so on.
- One of the coded number system is Binary Coded Decimal (BCD).
- For example decimal no 99 is represented as 1100011 in binary but in BCD form it becomes 1001 1001.



# Continued….

- Few decimal no and there BCD equivalent is as shown in figure below..

| Decimal number | Binary-coded decimal (BCD) number | |
|---|---|---|
| 0 | 0000 | ↑ |
| 1 | 0001 | |
| 2 | 0010 | |
| 3 | 0011 | Code |
| 4 | 0100 | for one |
| 5 | 0101 | decimal |
| 6 | 0110 | digit |
| 7 | 0111 | |
| 8 | 1000 | |
| 9 | 1001 | ↓ |
| 10 | 0001 0000 | |
| 20 | 0010 0000 | |
| 50 | 0101 0000 | |
| 99 | 1001 1001 | |
| 248 | 0010 0100 1000 | |

## ALPHANUMERIC CODES

- An alphanumeric character is a set of elements that includes the 10 decimal digits, the 26 letters of the alphabet and a number of special characters such as +,= etc.
- The standard alphanumeric binary code is the ASCII(American Standard code for Information Interchange).
- Which uses 7 bits to code 128 characters.



## CONTINUED..

- ASCII codes for the diff characters are listed below

| Character | Binary code | Character | Binary code |
|---|---|---|---|
| A | 100 0001 | 0 | 011 0000 |
| B | 100 0010 | 1 | 011 0001 |
| C | 100 0011 | 2 | 011 0010 |
| D | 100 0100 | 3 | 011 0011 |
| E | 100 0101 | 4 | 011 0100 |
| F | 100 0110 | 5 | 011 0101 |
| G | 100 0111 | 6 | 011 0110 |
| H | 100 1000 | 7 | 011 0111 |
| I | 100 1001 | 8 | 011 1000 |
| J | 100 1010 | 9 | 011 1001 |
| K | 100 1011 | | |
| L | 100 1100 | | |
| M | 100 1101 | space | 010 0000 |
| N | 100 1110 | . | 010 1110 |
| O | 100 1111 | ( | 010 1000 |
| P | 101 0000 | + | 010 1011 |
| Q | 101 0001 | $ | 010 0100 |
| R | 101 0010 | * | 010 1010 |
| S | 101 0011 | ) | 010 1001 |
| T | 101 0100 | − | 010 1101 |
| U | 101 0101 | / | 010 1111 |
| V | 101 0110 | , | 010 1100 |
| W | 101 0111 | = | 011 1101 |
| X | 101 1000 | | |
| Y | 101 1001 | | |
| Z | 101 1010 | | |

## COMPLEMENT

- Used for simplifying the subtraction and logical multiplication operation.
- Two types for each base r
  - r's complement
  - (r-1)'s complement
- For binary no system r is 2 hence there is 1's complement and 2's complement
- In decimal system base is 10 so there is 10's complement and 9's complement



## CONTINUED..

- 1's complement
  - 1's complement of N is =(2^n-1)-N.
  - It is also obtained by complementing each digit in binary no.
- 2's complement
  - 2'somplement = 1's complement + 1.
  - It can also be obtained by leaving all least significant 0's and first 1 unchanged and then replacing all 1's by 0's and all 0's by 1's in all higher significant bits.
- 9's complement
  - 9's complement of a number N is given by (10^n-1)-N
  - It can also obtained by subtracting each digit form 9.

## CONTINUED…

- 10’s complement
  - 10’s complement= 9’s complement +1 i.e. it is equal to [(10^n-1)-N]+1.



## COMPLEMENT ARITHMETIC

- Subtraction of Unsigned Numbers.
  - Subtraction of two n-digit unsigned numbers M-N in base r can be done as follows.

1. Add the minuend $M$ to the $r$'s complement of the subtrahend $N$. This performs $M + (r^n - N) = M - N + r^n$.
2. If $M \geq N$, the sum will produce an end carry $r^n$ which is discarded, and what is left is the result $M - N$.
3. If $M < N$, the sum does not produce an end carry and is equal to $r^n - (N - M)$, which is the $r$'s complement of $(N - M)$. To obtain the answer in a familiar form, take the $r$'s complement of the sum and place a negative sign in front.

## CONTINUED…

- Consider , e.g. , the subtraction 72532-13250=5982. The 10's complement of 13250 is 86750.Therefore:

| | |
|---|---|
| $M$ = | 72532 |
| 10's complement of $N$ = | +86750 |
| Sum = | 159282 |
| Discard end carry $10^5$ = | −100000 |
| Answer = | 59282 |

Now consider an example with $M < N$. The subtraction 13250 − 72532 produces negative 59282. Using the procedure with complements, we have

| | |
|---|---|
| $M$ = | 13250 |
| 10's complement of $N$ = | +27468 |
| Sum = | 40718 |



## CONTINUED…

- As there is no end carry, answer is negative 59282= 10's complement of 40718
- Same way subtraction of binary numbers can be done using same procedure.
- For example X=1010100 and Y=1000011, the subtraction X-Y and Y-X using 2's complement can be performed as bellow:

## CONTINUED..

$$\begin{aligned}
X &= \quad 1010100 \\
\text{2's complement of } Y &= +0111101 \\
\text{Sum} &= 10010001 \\
\text{Discard end carry } 2^7 &= -10000000 \\
\text{Answer: } X - Y &= \quad 0010001
\end{aligned}$$

$$\begin{aligned}
Y &= \quad 1000011 \\
\text{2's complement of } X &= +0101100 \\
\text{Sum} &= \quad 1101111
\end{aligned}$$

There is no end carry

Answer is negative 0010001 = 2's complement of 1101111



## INTEGER REPRESENTATION

- For binary integer no is positive sign is represented by 0 and negative sign is represented by 1.
- The negative number can be represented in one of the following three possible ways.
  1. Signed-magnitude representation
  2. Signed-1's complement representation
  3. Signed-2's complement representation
- For example no +14 is represented as 0 bit for sign followed by binary equivalent of 14 i.e. 00001110.
- And number -14 is represented as

| | |
|---|---|
| In signed-magnitude representation | 1 0001110 |
| In signed-1's complement representation | 1 1110001 |
| In signed-2's complement representation | 1 1110010 |

## Problem of overflow and detection

- Overflow occurs when a result of binary arithmetic operation exceeds the size of register.
- When two numbers of n digits each are added and the sum occupies n+1 digits then we say overflow occurred.
- The detection of overflow after the addition of two binary numbers depends on whether the numbers are signed or unsigned.
- When numbers added are unsigned the overflow is detected from end carry
- However if numbers are signed, sign bit is treated as part of the number and end carry does not indicate an overflow.



## Continued..

- Overflow can not occur if one positive and one negative numbers are added
- Overflow occurs if both the numbers are either positive or both are negative.
- Overflow can be detected by observing carry in sign bit and carry out of sign bit.
- Consider following example.

| | | | | | |
|---|---|---|---|---|---|
| carries: | 0 1 | | carries: | 1 0 | |
| +70 | 0 | 1000110 | −70 | 1 | 0111010 |
| +80 | 0 | 1010000 | −80 | 1 | 0110000 |
| +150 | 1 | 0010110 | −150 | 0 | 1101010 |

## Normalization of floating point numbers

- The floating point representation of a number has two parts such as mantissa and exponent.
- Mantissa represents a signed, fixed-point number and exponent represents the position of decimal point.
- For example consider decimal no +6132.789
  - Fraction : +0.6132789
  - Exponent : +04
- Exponent indicates that actual position of decimal point is 4 position to the right.
- Floating point representation is as follow

$$m \times r^e$$

- M is mantissa, e exponent and r is radix.



## Continued..

- Floating point binary number :
  - Consider +1001.11
  - Fraction : 0100111
  - Exponent: 000100
- Fraction has 0 in left most position to indicate positive
- A floating point number is said to be normalized, if the most significant digit of the mantissa is nonzero.
- For example 8 bit binary no 00011010 is not normalized because of leading zero.

## GRAY CODE AND CONVERSION

- Also called as reflected binary code
- Advantage over binary code is that Gray code changes by only one bit i.e. from 1 to 0 or from 0 to 1 as it sequences from one number to the next.
- Typical application of Gray code occurs when the analog data are represented by the continuous change of shaft position.



## CONTINUED

- 4-bit Gray code is as shown in table below.

| Binary code | Decimal equivalent | Binary code | Decimal equivalent |
|---|---|---|---|
| 0000 | 0 | 1100 | 8 |
| 0001 | 1 | 1101 | 9 |
| 0011 | 2 | 1111 | 10 |
| 0010 | 3 | 1110 | 11 |
| 0110 | 4 | 1010 | 12 |
| 0111 | 5 | 1011 | 13 |
| 0101 | 6 | 1001 | 14 |
| 0100 | 7 | 1000 | 15 |

## OTHER DECIMAL CODES

- Self complementing codes:
  - 2421-code and exess-3 codes are example of self complementing codes.
- Excess-3 codes:
  - This is one of the unweighted code
  - It is obtained by adding binary 3 (0011) to the corresponding BCD equivalent binary number.
- Weighted codes:
  - In weighted code, the bits are multiplied by the weights indicated and the sum o the weighted bits gives decimal digit.
  - for example 1101 in 2421 weighted form gives (2 x 1+4 x 1+ 2 x 0 +1 x 1)=7.



## CONTINUED…

- Different binary codes for the Decimal Digits are as shown in table below.

| Decimal digit | BCD 8421 | 2421 | Excess-3 | Excess-3 gray |
|---|---|---|---|---|
| 0 | 0000 | 0000 | 0011 | 0010 |
| 1 | 0001 | 0001 | 0100 | 0110 |
| 2 | 0010 | 0010 | 0101 | 0111 |
| 3 | 0011 | 0011 | 0110 | 0101 |
| 4 | 0100 | 0100 | 0111 | 0100 |
| 5 | 0101 | 1011 | 1000 | 1100 |
| 6 | 0110 | 1100 | 1001 | 1101 |
| 7 | 0111 | 1101 | 1010 | 1111 |
| 8 | 1000 | 1110 | 1011 | 1110 |
| 9 | 1001 | 1111 | 1100 | 1010 |
| Unused bit combi-nations | 1010 | 0101 | 0000 | 0000 |
| | 1011 | 0110 | 0001 | 0001 |
| | 1100 | 0111 | 0010 | 0011 |
| | 1101 | 1000 | 1101 | 1000 |
| | 1110 | 1001 | 1110 | 1001 |
| | 1111 | 1010 | 1111 | 1011 |