# Number System

ESC108A Elements of Computer Science and Engineering

B. Tech. 2017


## Course Leaders:

**Roopa G.**

**Ami Rai E.**

**Chaitra S.**

# Objectives

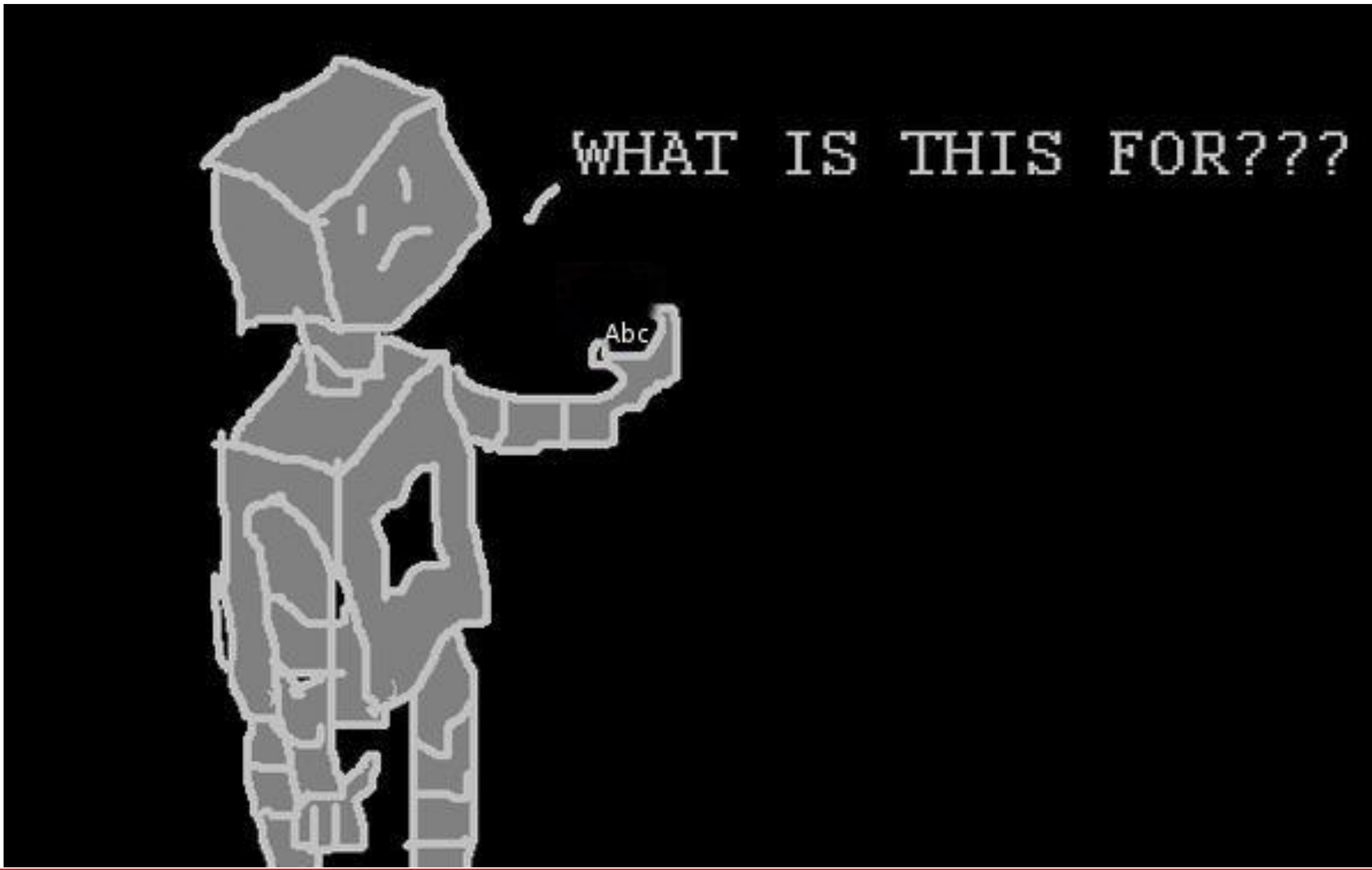
- At the end of this lecture, student will be able to
  - Explain binary and hexadecimal number systems
  - Apply method for conversion between binary, hexadecimal and decimal number systems
  - Represent a binary number in 2's Complement notation

# Contents

- Number Systems
- Conversions
- Binary subtraction and addition
- 1’s and 2’s Complement Notation

# Question

Computers understand numbers only… How can it work with other data such as strings, real values, etc?



# Number Systems and Conversion

- Decimal Number System – What we generally use
- **1028**
  - Units place: 8
  - Tenths place:2
  - Hundreds place:0
  - Thousands place:1

- Actual meaning
  - $8x10^0+2x10^1+0x10^2+1x10^3$
  - 10 is the ***'Base'*** as there are ***10 possible symbols {0-9}***
  - Hence we call the number system as ***'Decimal'***

# Numeral Systems Conversion Table

| Decimal Numbers | Binary Numbers | Octal Numbers | Hexa-decimal Numbers |
|---|---|---|---|
| 0 | 0000 | 0 | 0 |
| 1 | 0001 | 1 | 1 |
| 2 | 0010 | 2 | 2 |
| 3 | 0011 | 3 | 3 |
| 4 | 0100 | 4 | 4 |
| 5 | 0101 | 5 | 5 |
| 6 | 0110 | 6 | 6 |
| 7 | 0111 | 7 | 7 |
| 8 | 1000 | - | 8 |
| 9 | 1001 | - | 9 |
| 10 | 1010 | - | A |
| 11 | 1011 | - | B |
| 12 | 1100 | - | C |
| 13 | 1101 | - | D |
| 14 | 1110 | - | E |
| 15 | 1111 | - | F |

# Binary Number System

- Base ***2***
- Number of ***symbols = 2*** i.e. ***{0, 1}***

- Hence 1101 in Binary number system would mean
  - $1x2^0+0x2^1+1x2^2+1x2^3$
  - 1x1+0x2+1x4+1+8
  - 1+0+4+8
  - 13

# Octal Number System

– Base ***8***

– Number of ***symbols  = 8*** i.e. ***{0-7}***

– Hence 23 in Octal number system would mean

- $3x8^0$+$2x8^1$
- 3x1+2x8
- 3+16
- 19

# Hexadecimal Number System

– Base ***16***
– Number of ***symbols = 16*** i.e. ***{0-9,a-f,A-F}***

– Hence 1f in Hexadecimal number system would mean
  - $f \times 16^0 + 1 \times 16^1$
  - $15 \times 1 + 1 \times 16$
  - $15 + 16$
  - 31

# Number Systems and Conversion

$(1028)_{10}$ is 1028 in *Decimal* number system

$(1101)_2$ is 13 in *Binary* number system

$(23)_8$ is 19 in *Octal* number system

$(1f)_{16}$ is 31 in *Hexadecimal* number system

- Conversion from any base system to Decimal number system
  - Multiply with respective base powers for that location
    - $(1101)_2 = 1x2^0+0x2^1+1x2^2+1x2^3 = (13)_{10}$
    - $(1f)_{16} = fx16^1+1x16^1 =(241)_{10}$

# Number Systems and Conversion

- Conversion from Decimal number system to another base number system
  - Divide with the base of the number system recursively and find remainder
  - Reverse the order of remainder

# Decimal to Binary

- Decimal number system to Binary number system
  - $(13)_{10}$
  - 13/2 = 6 , remainder 1
  - 6 /2 = 3, remainder 0
  - 3 /2 = 1, remainder 1
  - 1 /2 = 0, remainder 1

  - Hence $(13)_{10}$ is $(1101)_2$

# Decimal to Hexadecimal

- Decimal number system to Hexadecimal number system
  - $(241)_{10}$
  - 241/16 = 15 , remainder 1
  - 15/16 = 0 , remainder 15 ('f')
  - Hence $(241)_{10}$ is $(f1)_{16}$

# Binary Addition and Subtraction

| | |
|---|---|
| 0110 | 0110 |
| +0101 | - 0101 |
| 1011 | 0001 |

# Rules for Binary Addition

- 0 + 0 = 0
- 0 + 1 = 1
- 1 + 0 = 1
- 1 + 1 = 0, and carry 1 to the next more significant bit

- Example

$$0\ 0\ 0\ 1\ 1\ 0\ 1\ 0 \quad = \quad 26_{(\text{base } 10)} +$$

$$0\ 0\ 0\ 0\ 1\ 1\ 0\ 0 \quad = \quad 12_{(\text{base } 10)}$$

$$0\ 0\ 1\ 0\ 0\ 1\ 1\ 0 \quad = \quad 38_{(\text{base } 10)}$$

# Rules for Binary Subtraction

- 0 - 0 = 0
- 0 - 1 = 1, and borrow 1 from the next more significant bit
- 1 - 0 = 1
- 1 - 1 = 0

- Example

$$\begin{array}{rcl} 0\ 0\ 1\ {}^{1}0\ 0\ 1\ 0\ 1 & = & 37_{(\text{base } 10)} \\ -\ 0\ 0\ 0\ 1\ \ 0\ 0\ 0\ 1 & = & 17_{(\text{base } 10)} \\ \\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0 & = & 20_{(\text{base } 10)} \end{array}$$

# Logical Primitives

- And (&)
- Or (|)
- Complement(!)
- Xor(^)

| A | B | A&B | A\|B | A^B | !A |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 |

# 1’s Complement

- 1’s Complement: Inversion of bits
  - 1101
  - 0010

# 2's Complement

- 2's Complement
  - Method: 1's complement of number + 1
  - Actual meaning :
    - If there are N bits in a number 'x', 2's complement of 'x' is $2^N - x$ where $2^N$ is a binary number that has 1 followed by N zeros

  - 2's Complement of 1101
  - 0010+0001 = 0011 = 10000 - 1101

# MSB and LSB

- Most Significant Bit (MSB)
- Least Significant Bit (LSB)

  – Consider a binary number, say 10101100
  – 1 is called the MSB
    - Changes in MSB varies the value by a huge margin
  – 0 is called the LSB
    - Changes in LSB varies the value by a the smallest margin possible

# Data Sizes

- 1 Bit - 1 or 0 (bit is short for BInary digiT)
- 1 Nibble – 4 bits
- 1 Byte – 8 bits or 2 nibbles
- 1 Word – a sequence of 16 bits or 2 bytes
- 1 Kilobyte – 1024 bytes
- 1 Megabyte – 1024 Kilobytes
- 1 Gigabyte – 1024 Megabytes
- 1 Terabyte – 1024 Gigabytes
- 1 Petabyte – 1024 Terabytes

# Summary

- There are different number systems created with different symbols
- Number systems are represented with base that specifies the number of symbols used
- A number's value in decimal system can be found by calculating the sum of the products of each digit and the base raised to the power of the digit's position from right corner
- Binary number system is used in computers
- The complement of a number is the number obtained with inversion of its bits
- The 2's complement of a number is the number obtained by subtracting the number from $2^N$, where N is number of bits in the number

# Further Reading

Kernighan, B. W. and Richie, D. (1992) *The C Programming Language.* 2nd ed., New Delhi:PHI.