Department of Computer Science and Engineering

RGUKT – SRIKAKULAM

COMPUTER ORGANIZATION AND ARCHITECTURE
LECTURE NOTES

Prepared by
***Mr. Dileep Kumar Koda** M. Tech., (Ph. D)*
Asst. Professor
Computer Science & Engineering

Rajiv Gandhi University of Knowledge Technologies
Srikakulam

Catering to the Educational Needs of Gifted Rural Youth of Andhra Pradesh

(Established by the Govt. of Andhra Pradesh and recognized as per Section 2(f) of UGC Act, 1956)

COMPUTER ORGANIZATION AND ARCHITECTURE

Course code	Course name Course	Category L-T-P	Credits
20CS2201	COMPUTER ORGANIZATION AND ARCHITECTURE	PCC 3-0-0	3

Course Content

UNIT-I
Basic Functional blocks of a computer: CPU, Memory, Input -Output Subsystems, Control Unit. **Data Representation:** Number Systems, Signed Number Representation, Fixed and Floating Point Representations, Character Representation.

UNIT-II
ALU: Computer Integer Arithmetic: addition, subtraction, multiplication, division, floating point arithmetic: Addition, subtraction, multiplication, division.
Instruction set architecture of a CPU registers, instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set. RISC and CISC architecture. Case study instruction sets of some common CPUs.

UNIT-III
CPU control unit design: Introduction to CPU design, Processor Organization, Execution of Complete Execution, Design of Control Unit: hardwired and micro-programmed control, Case study design of a simple hypothetical CPU.

UNIT-IV
Memory system design: Concept of memory: Memory hierarchy, SRAM vs DRAM, Internal organization of memory chips, cache memory: Mapping functions, replacement algorithms, Memory management, virtual memory.

UNIT-V
Input -output subsystems, I/O transfers: programmed I/O, interrupt driven and DMA. I/O Buses, Peripheral devices and their characteristics, Disk Performance

UNIT-VI
Performance enhancement techniques: Pipelining: Basic concepts of pipelining, through put and speedup, pipeline hazards.
Parallel processing: Introduction to parallel processing, Introduction to Network, Cache coherence

Text Books:

1. V. C. Hamacher, Z. G. Vranesic and S. G. Zaky, "Computer Organization," 5/e, McGraw Hill, 2002.
2. William Stallings, "Computer Organization and Architecture": Designing for Performance, 8/e, Pearson Education India. 2010.
3. **Morris Mano**, "Computer System Architecture", Pearson Education India, Third edition. References: A. S. Tanenbaum, "Structured Computer Organization", 5/e, Prentice Hall of India, 2009.
4. D. A. Patterson and J. L. Hennessy, "Computer Organization and Design," 4/e, Morgan Kaufmann, 2008.
5. J. L. Hennessy and D. A. Patterson," Computer Architecture: A Quantitative Approach",4/e, Morgan Kaufmann, 2006.

6. D. V. Hall, “Microprocessors and Interfacing”, 2/e, McGraw Hall, 2006 “8086 Assembler Tutorial for Beginners “By Prof. Emerson Giovani Carati.

UNIT - I

Module – 1: Functional Blocks

● CPU
● Memory
● Input-Output Subsystems
● Control Unit
● Data Representation

Module – 2: Number Systems

● Signed Number representations
● Fixed and Floating Point Representations
● Character Representation

COMPUTER

What is a computer?

A computer is an electronic device that manipulates information, or data. It has the ability to store, retrieve and process data.

(or)

A computer is an electronic device for storing and processing data, typically in binary form, according to instructions given to it in a variable program

What are the Basic Parts of a Computer?

Every computer comprises 5 basic parts, namely, a motherboard, a central processing unit, a graphics processing unit, a random access memory, and a hard disk or solid-state drive.

Basic parts of a Computer

Computer Organization and Architecture (COA):

- In the early part of the computer evolution, there were no stored-program computer
- The computational power was less and on the top of it the size of the computational was a very huge one.
- The task that the computer designer handles is a complex one:
    - Determine what attributes are important for a new machine
    - Design a machine to maximize performance while staying within cost constraints
    - Including instruction set design, functional organization, logic design, an implementation.
- While looking for the task for computer design, both the terms computer organization and architecture

Why we need to study Computer Organization and Architecture (COA)?

- Without computers, the field of computer science does not exist.
- Its helps to design better programs, including system software such as compilers, operating systems, and device drivers. –Optimize program behaviour.
- The main objective of the computer organizations is to understand the various computer hardware components and the interaction between these components.
- **Computer Architecture** helps us to understand functionalities of a system.
- **Computer Organization** tells us how exactly all the units in the system are arranged and interconnected.

Computer Architecture:

Computer architecture refers to those parameters of a computer system that are visible to a programmer.

(or)

Those parameters that have a direct impact on the logical execution of a program.

Examples of architectural attributes include The instruction set, the number of bits used for represent different data types**,** I/O Mechanisms, Techniques for addressing Memory.

Computer organization:

Computer organization refers to the operational units and their interconnections that realize the architectural specifications.

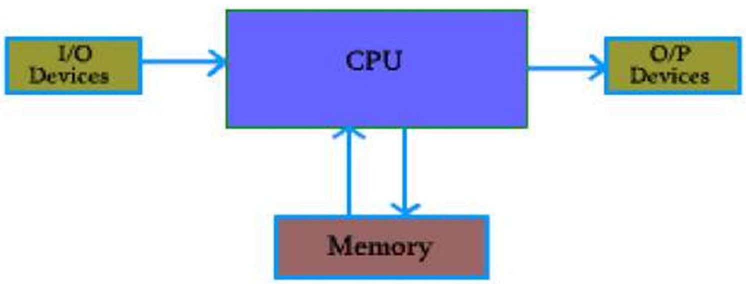
Examples of organizational attributes include the hardware details transparent to the programmer such as control signals, Interface between the computer and peripherals the memory technology used.

Basic Computer Model & Basic functional blocks of a Computer:

The models of a computer can be described by four basic units in the high level of abstraction and they are as follows.

1. Central Processor Unit
2. Input unit
3. Output unit
4. Memory unit

Units of a computer:



(Functional blocks of a basic Computer)

1. **Central Processing Unit (CPU):** A typical CPU has three major components:
   A. Control Unit(CU)
   B. Execution unit or Data Processing unit

A. Control Unit:

- Control Unit is the part of the computer central processing unit (cpu), which directs the operations of the processor.
- It is the responsibility of the Control Unit to tell the computer's memory, arithmetic/logic unit and input and output devices how to respond to the instructions that have been sent to the processor.
- Control unit has a set of registers and control circuit to generate control signals.

  Note : Control Unit is a hardware part.
  Note : Execution Unit is a software part.

Register set:

- Generally, register consists of a group of flip-flops and gates that effect their transition. The flip-flops hold the binary information and the gates control when and how new information is transferred into the register.
- Registers are extremely fast memory locations within the CPU that are used to create and store the results of CPU operations and other calculations as a temporary data.

- Different computers have different register sets. They differ in the number of registers, register types, and the length of each register.
- Eg.: Data Register(DR), Instruction Register (IR), etc..

B. Execution unit (or) Data processing unit:

- An execution unit performs the operations and calculations as instructed by the computer program.
- It may have some registers and other internal units such as Arithmetic and Logic Unit(ALU).

Arithmetic and Logic Unit (ALU):

- It carries arithmetic and logic operations on the operands by the computer instruction.

2. Input Unit:

- With the help of input unit data from outside can be
- supplied to the computer.
- Program or data is read into main storage from input device or secondary storage under the control of CPU input instruction.
- **Ex:** Keyboard, Mouse, Hard disk, Floppy disk, CD-ROM drive etc.

3. Output Unit:

- With the help of output unit computer results can be provided to the user or it can be stored in storage device permanently for future use.
- Output data from main storage go to output device under the control of CPU output instructions.
- **Ex:** Printer, Monitor, Plotter, Hard Disk, Floppy Disk etc.

4. Memory Unit

- Memory is basically a device that has the capacity to store information. A memory unit is the amount of data that the memory can hold.
- It is used to store the data and program. CPU can work with the information stored in memory unit. This memory unit is termed as **primary memory or main memory** module.

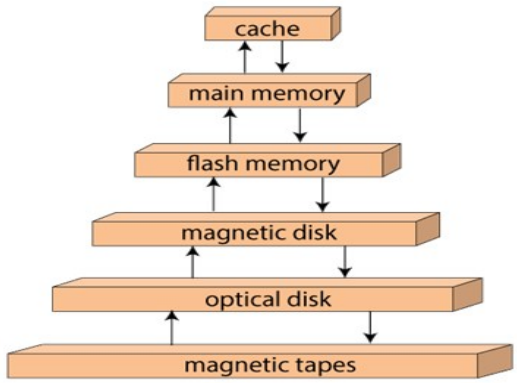
Primary Memory /Volatile Memory:

- It is the one that is responsible for operating the data that is available by the storage medium.
- The main memory handles each instruction of a computer machine.
- The memory can store gigabytes of data on a system but is small enough to carry the entire database.
- The main memory loses the whole content if the system shuts down because of power failure or other reasons.

Secondary Memory /Non-Volatile Memory:

- There is another kind of storage device, apart from primary or main memory, which is known as secondary memory.
- Secondary memories are **non-volatile memory** and it is used for permanent storage of data and program.

Block diagram of Storage Device Hierarchy:



Storage device hierarchy

Basic Working Principle of a Computer:

- To distinguish between arithmetic and logical operation, we may use a signal line,
    - 0 - in that signal, represents an **arithmetic operation**
    - 1 - in that signal, represents a **logical operation.**
- In the similar manner, we need another two signal lines to distinguish between four arithmetic operations. The different operations and their binary code is as follows:
- Let us, Consider the part of Control Unit, its task is to generate the appropriate signal at right moment.
- There is an instruction decoder in CPU which decodes this information in such a way that computer can perform the desired task

Arithmetic		Logical	
000	ADD	100	OR
001	SUB	101	AND
010	MULT	110	NAND
011	DIV	111	ADD

- In simple model, we use three storage units in CPU,

    Two -- for storing the operand and

    One -- for storing the results.

These storage units are known as **register.**

- But in computer, we need more storage space for proper functioning of the Computer.
- Some of them are inside CPU, which are known as register. Other bigger junk of storage space is known as
- primary memory or main memory. The CPU can work with the information available in main memory only.
- Data and program is stored in main memory. While executing a program, CPU brings instruction and data from main memory, performs the tasks as per the instruction fetch from the memory. After completion of operation, CPU stores the result back into the memory.
- Registers have limited storage capacity; hence we are using main memory.
- For proper functioning we require more storage in computer, this memory is known as **'Main Memory**'.
- To transfer the data/program from memory to CPU we require two special registers.

  1**. MAR** -**M**emory **A**ddress **R**egister
  2.**MDR**-**M**emory **D**ata **R**egister

- MAR can store the location of data in terms of address and MDR can stores the Actual data
- While Execution the data and instructions will be stored in the registers.
- Data and programs are stored in the main memory
- Before Execution, the data and program will be fetched from the main memory, after that completion of operation the CPU will stores the result back into the memory.

MAIN MEMORY ORGANIZATION

Main Memory Organization:

- Main memory unit is the storage unit, there are several locations for storing information in the main memory Module
- The capacity of a memory module is specified by the number of memory location and the information stored in each location.
- A memory module of capacity 16 X 4 indicates that, there are 16 locations in the memory module and in each location, we can store 4 bit of information.
- We have to know how to indicate or point to a specific memory location. This is done by address of the memory location.
- We need two operations to work with memory.
  - **READ Operation:** This operation is to retrieve the data from memory and bring it to CPU register
  - **WRITE Operation:** This operation is to store the data to a memory location from CPU register
- We need some mechanism to distinguish these two operations **READ** and **WRITE**

- With the help of one signal line, we can differentiate these two operations. If the content of this signal line is 0, we say that we will do a READ operation; and if it is 1, then it is a WRITE operation.
- To transfer the data from CPU to memory module and vice-versa, we need some connection. This is termed as DATA BUS.
- The size of the data bus indicate how many bit we can transfer at a time. Size of data bus is mainly specified by the data storage capacity of each location of memory module.
- We have to resolve the issues how to specify a particular memory location where we want to store our data or from where we want to retrieve the data.
- This can be done by the memory address. Each location can be specified with the help of a binary address.
- If we use 4 signal lines, we have 16 different combinations in these four lines, provided we use two signal values only (say 0 and 1).
- To distinguish 16 locations, we need four signal lines. These signal lines use to identify a memory location is termed as ADDRESS BUS. Size of address bus depends on the memory size.
- For a memory module of capacity of 2^n location, we need 'n' address lines, that is, an address bus of size 'n'.
- We use an address decoder to decode the address that are present in address bus
- As for example, consider a memory module of 16 locations and each location can store 4 bit of information
    - The size of address bus is 4 bit and the size of the data bus is 4 bit
    - The size of address decoder is 4 X 16.

There is a control signal named R/W.

If R/W = 0, we perform a READ operation and

if R/W = 1, we perform a WRITE operation

MEMORY INSTRUCTION

- We need some more instruction to work with the computer. Apart from the instruction needed to perform task inside CPU, we need some more instructions for data transfer from main memory to CPU and vice versa.
- In hypothetical machine, we use three signal lines to identify a particular instruction. If we want to include more instruction, we need additional signal lines.

Instruction	Code	Meaning
1000	LDAI imm	To enable Register A
1001	LDAA addr	Load register A,with data from memory
1010	LDBI imm	To enable Register A
1011	LDBA addr	Load register A,with data from memory

1100	STC addr	Store the value of register c in memory
1101	HALT	Stop the execution
1110	NOP	No operation
1111	NOP	No operation

- With this additional signal line, we can go up to 16 instructions. When the signal of this new line is 0, it will indicate the ALU operation. For signal value equal to 1, it will indicate 8 new instructions. So, we can design 8 new memory access instructions
- We have added 6 new instructions. Still two codes are unused, which can be used for other purposes. We show it as NOP means No Operation.
- We have seen that for ALU operation, instruction decoder generated the signal for appropriate ALU operation.
- Apart from that we need many more signals for proper functioning of the computer. Therefore, we need a module, which is known as control unit, and it is a part of CPU. The control unit is responsible to generate the appropriate signal.
- As for example, for LDAI instruction, control unit must generate a signal which enables the register A to store in data into register A.
- One major task is to design the control unit to generate the appropriate signal at appropriate time for the proper functioning of the computer.
- Consider a simple problem to add two numbers and store the result in memory, say we want to add 7 to 5.
- To solve this problem in computer, we have to write a computer program. The program is machine specific, and it is related to the instruction set of the machine.
- For hypothetical machine, the program is as follows

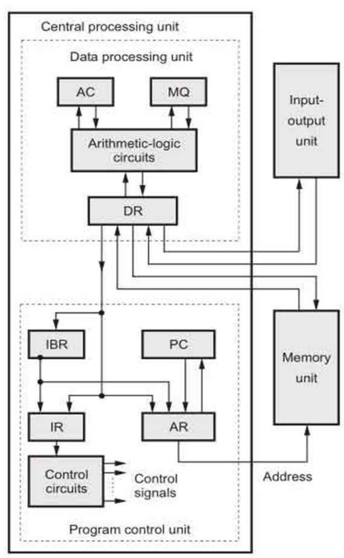
Instruction	Binary	Hex	Memory location
LDAI 5	1000 0101	8 5	(0,1)
LDBI 7	1010 0111	A 7	(2,3)
ADD	0000	0	(4)
STC 12	1100 1100	C C	(5,6)
HALT	1101	D	(7)

NOTE: Number of instructions that can be provided in a computer depends on the signal lines that are used to provide the instruction.

EX: Computer memories may range from 1024 words, requiring an address of 10 bits, 2^{32} words requiring 32 address bits.

Memory Organization: Stored Program

- The present digital computers are based on stored-program concept introduced by Von Neumann. In this stored-program concept, programs and data are stored in separate storage unit called memories.
- Central Processing Unit, the main component of computer can work with the information stored in storage unit only.
- In **1946**, **Von Neumann** and his colleagues began the design of a stored-program computer at the Institute for Advanced Studies in Princeton. This computer is referred as the IAS computer.



Register	Abbreviation
Accumulator	AC
Multiplier Quotient	MQ
Data Register	DR
Program Counter	PC
Instruction Buffer Register	IBR
Address Register	AR
Instruction Register	IR

Structure of a first generation computer: IAS

- The IAS computer is having three basic units:
    - The Central Processing Unit (CPU).
    - The Main Memory Unit.
    - The Input/Output Device.

Central Processing Unit:

- This is the main unit of computer, which is responsible to perform all the operations. The CPU of the IAS computer consists of a data processing unit and a program control unit.
- The data processing unit contains a high speed registers intended for temporary storage of instructions, memory addresses and data. The main action specified by instructions are performed by the arithmetic-logic circuits of the data processing unit.

- The control circuits in the program control unit are responsible for fetching instructions, decoding opcodes, controlling the information movements correctly through the system, and providing proper control signals for all CPU actions.

The Main Memory Unit:

- It is used for storing programs and data. The memory locations of memory unit is uniquely specified by the memory address of the location. M(X) is used to indicate the location of the memory unit M with address X.
- The data transfer between memory unit and CPU takes place with the help of data register DR. When CPU wants to read some information from memory unit, the information first brings to DR, and after that it goes to appropriate position. Similarly, data to be stored to memory must put into DR first, and then it is stored to appropriate location in the memory unit.
- The address of the memory location that is used during memory read and memory write operations are stored in the memory register AR.
- The information fetched from the memory is an operand of an instruction, then it is moved from DR to data processing unit (either to AC or MQ). If it is an operand, then it is moved to program control unit (either to IR or IBR).
- Two additional registers for the temporary storage of operands and results are included in data processing units: the accumulator AC and the multiplier-quotient register MQ.
- Two instructions are fetch simultaneously from M and transferred to the program control unit. The instruction that is not to be executed immediately is placed in the instruction buffer register IBR. The opcode of the other instruction is placed in the instruction register IR where it is decoded.
- In the decoding phase, the control circuits generate the required control signals to perform the specified operation in the instruction.
- The program counter PC is used to store the address of the next instruction to be fetched from memory.

Input Output Device:

- Input devise are used to put the information into computer. With the help of input devices, we can store information in memory so that CPU can use it. Program or data is read into main memory from input device or secondary storage under the control of CPU input instruction.
- Output devices are used to output the information from computer. If some results are evaluated by computer and it is stored in computer, then with the help of output devices, we can present it to the user. Output data from the main memory go to output device under the control of CPU output instruction.

Registers & their functionalities

1. **DR:** The Data Register which hold the operand read from the memory location and Data transfer between memory unit and CPU takes place with the help of data register.
2. **PC: The** Program Counter (PC) which hold the address of the next instruction to be read from memory after the current instruction is executed.
3. **AC: The** Accumulator (AC) register is a general purpose processing register.

4. **IR: The** instruction read from memory is placed in the Instruction register (IR).
5. **MQ: Multiplier** quotient (MQ), Employed to hold temporarily operands and results of ALU operations.
6. **IBR:** The instruction that is not to be executed immediately is placed in the instruction buffer register
7. **AR: The** address of the memory location that is used during memory read and memory write operations are stored in the memory register AR.

Number System

Depending on the base or radix, number systems can be classified into the following four major types –

1. Decimal Number System (radix/base=10)
2. Binary Number System (radix/base =2)
3. Octal Number System (radix/base=8)
4. Hexadecimal Number System (radix/base=16)

1. Decimal Number System:

The system of numbers which has base or radix 10, i.e. uses total 10 symbols to represent numbers of the system is called decimal number system. The symbols used in the decimal number system are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9; where each of these symbols assigned a specific value.

Eg:

$(1\times1000) + (2\times1000) + (3\times1000) + (4\times1)$

$1000+200+30+4=1234$

2. Binary Number system:

A number system with base or radix 2 is called binary number system. The binary number system uses only 2 symbols (0 and 1) to represent binary numbers. All modern digital devices like computers, combinational circuits, sequential circuits, etc. use the binary number system to operate.

- In computer to represent any information we have to take values with the help of two signals.
- These two signals will correspond to the two level of signals.
- **DEFINITION:** It uses two digits and coefficients are multiplied by the power of 2.
- For an n-bit number the coefficient is a_j multiplied by 2^j

  where, $(0<=j<n)$

Eg:

Let a binary number 1101 and we have to convert it into an equivalent decimal number, then

$(1\times2^3) + (1\times2^2) + (0\times2^1) + (1\times2^0)$

$8+4+0+1= 13$

$\therefore (1101)_2 = (13)_{10}$

3. Octal Number System:

A number system which has base 8 is called an octal number system. Therefore, the octal number system uses 8 symbols, (0, 1, 2, 3, 4, 5, 6, 7) to represent the number.

EX: An octal number can be converted into an equivalent decimal number as follows. Let an octal number 124 and we need to find its equivalent in decimal, then

$(1\times8^2) + (2\times8^1) + (4\times8^0)$

$64+16+4=84$

$\therefore (124)_8 = (84)_{10}$

Hexa Decimal Number System:

The number system with base or radix 16 is called as hexadecimal number system. Thus, the hexadecimal number system uses 16 symbol to represent numbers. These symbols are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Where, A = 10; B = 11; C = 12; D = 13; E = 14; F = 15.

Eg.: Let 1AF is a hexadecimal number, we can convert it into the equivalent decimal number as follows −

$(1\times16^2) + (A\times16^1) + (F\times16^0)$

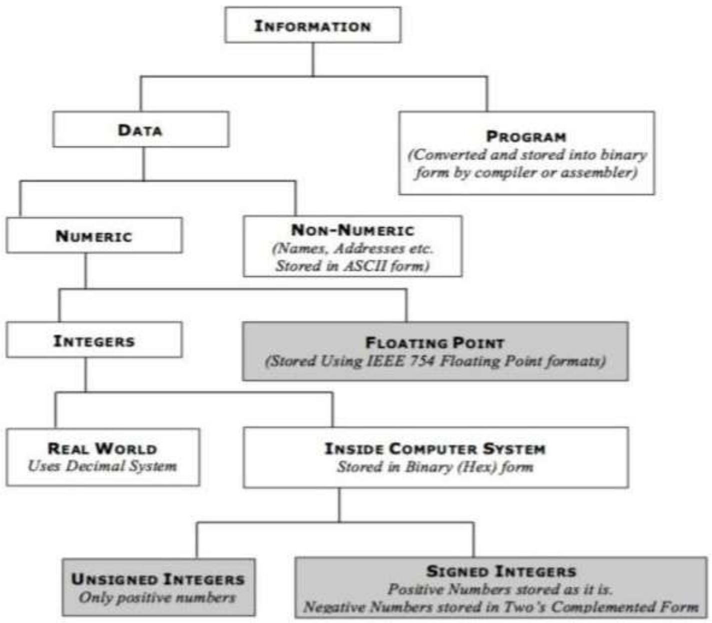
$(1\times16^2) + (10\times16^1) + (15\times16^0)$

$256+160+15=431$

$\therefore(1AF)_{16} = (431)_{10}$

$\therefore(1AF)_{16} = (431)_{10}$

DATA REPRESENTATION



Representation of Unsigned Integers:

- Any integer can be stored in computer in binary form. As for example:
- The binary equivalent of integer 107 is 1101011, so 1101011 are stored to represent 107.
- These are binary numbers that are always assumed to be positive.
  E.g.: Roll Numbers, Memory addresses etc.

  What is the size of Integer that can be stored in a Computer?
- It depends on the word size of the Computer. If we are working with 8-bit computer, then we can use only 8 bits to represent the number. The eight-bit computer means the storage organization for data is 8 bits.
- In case of 8-bit numbers, the minimum number that can be stored in computer is 00000000 (0) and maximum number is 11111111 (255) (if we are working with natural numbers).
- So, the domain of number is restricted by the storage capacity of the computer. Also it is related to number system; above range is for natural numbers.
- In general, for n-bit number, the range for natural number is from ***0 to $2^n - 1$.***

- Any arithmetic operation can be performed with the help of binary number system. Consider the following two examples, where decimal and binary additions are shown side by side.

0 1 1 0 1 0 0 0	104
0 0 1 1 0 0 0 1	49
10011001	153

- In the above example, the result is an 8-bit number, as it can be stored in the 8-bit computer, so we get the correct results

1 0 0 0 0 0 0 1	129
1 0 1 0 1 0 1 0	178
1 0 0 1 0 1 0 1 1	307

- In the above example, the result is a 9-bit number, but we can store only 8 bits, and the most significant bit (msb) cannot be stored.
- The result of this addition will be stored as (00101011) which is 43 and it is not the desired result. Since we cannot store the complete result of an operation, and it is known as the overflow case.
- The smallest unit of information is known as BIT (Binary digit).
- The binary number 110011 consists of 6 bits and it represents:

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

- For an n-bit number the coefficient is – a_j multiplied by 2^j where, ($0 \leq j < n$)
- The coefficient ***a(n-1)*** is multiplied by $2^{(n-1)}$ and it is known as most significant bit (***MSB***).
- For our convenient, while writing in paper, we may take help of other number systems like octal and hexadecimal. It will reduce the burden of writing long strings of 0s and 1s.

Octal Number: The octal number system is said to be of base, or radix 8, because it uses 8 digits and the coefficients are multiplied by power of 8.

Eight digits used in octal system are: 0, 1, 2, 3, 4, 5, 6 and 7.

Hexadecimal number: The hexadecimal number system is said to be of base, or radix 16, because it uses 16 symbols and the coefficients are multiplied by power of 16.

Sixteen digits used in hexadecimal system are: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E and F.
Consider the following addition example:

Binary	Octal	Hexadecimal	Decimal
0 1 1 0 1 0 0 0	150	68	104
0 0 1 1 1 0 1 0	072	3A	58
---------------	------	------	-----
1 0 1 0 0 0 1 0	242	A2	162

Signed Integers:

- For n-bit number, the range for natural number is from ***0 to*** $2^n - 1$

- For n-bit, we have all together 2^n different combination, and we use these different combinations to represent 2^n numbers, which ranges from ***0 to $2^n - 1$.***
- If we want to include the negative number, naturally, the range will decrease. Half of the combinations are used for positive number and other half is used for negative number.
- For n-bit representation, the range is from $-2^{n-1} -1$ *to* $+ 2^{n-1} - 1$
- For example, if we consider 8-bit number, then range
  for natural number is from 0 to 255 but
  for signed integer the range is from -127 to +127.

● These are binary numbers that can be either positive or negative.

● The MSB of the number indicates whether it is positive or negative.

  - If MSB is 0 then the number is Positive.
  - If MSB is 1 then the number is Negative.

● Negative numbers are always stored in signed format.

Representation of signed numbers:

- For n-bit number, the range for natural number is from 0 to $2^n - 1$.
- There are three different schemes to represent negative number:
  A. Signed-Magnitude form.
  B. 1's complement form.
  C. 2's complement form.

Note: Negative numbers are stored in 2's complement form due to two reasons.

1) Two's complement gives a unique representation for zero.

- Any other system gives a separate representation for +0 and for -0. his is absurd.
- In two's complement system, -(x) is stored as two's complement of (x).
- Applying the same rule for 0, -(0) should be stored as two's complement of 0. 0 is stored as 000.
- So (0) should be stored as two's complement of 000, which again is 000. Hence two's complement gives a unique representation for 0.

2) It produces an additional number on the negative side.

- As two's complement system produces a unique combination for 0, it has a spare combination '100' in the above case, and can be used to represent $(4)_{10}$

3-Bit Representation of unsigned numbers:

Number	Sign Magnitude	One's Complement	Two's Complement
3	011	011	011
2	010	010	010
1	001	001	001
0	000	000	000
-0	100	111	000
-1	101	110	011
-2	110	101	110
-3	111	100	101
-4	---	---	100

3 Bit Integer	
$2^3 = 8$ therefore 8 combinations	
Unsigned	Signed
0 ... 7	-4 ... -1 0 1 ... 3

4 Bit Integer	
$2^4 = 16$ therefore 16 combinations	
Unsigned	Signed
0 ... 15	-8 ... -1 0 1 ... 7

5 Bit Integer	
$2^5 = 32$ therefore 32 combinations	
Unsigned	Signed
0 ... 31	-16 ... -1 0 1 ... 15

8 Bit Integer	
2^8 = 256 therefore 256 combinations	
Unsigned	Signed
0 … 255	-128 … -1 0 1 … 127

16 Bit Integer	
2^{16} = 65536 therefore 65536 combinations	
Unsigned	Signed
0 … 65535	-32768 … -1 0 1 … 32767

Signed Number Examples		
+5	0101	
-5	1011	← *2's complement*
+9	01001	
-9	10111	← *2's complement*
+23	010111	
-23	101001	← *2's complement*

4-Bit Representation of unsigned numbers:

Decimal	2's Complement	1's complement	Signed Magnitude
+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	0001	0001	0001
+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	------	-------

Complements:

which species the subtraction operations (or) subtraction can be achieved by addition operation.

r's complement: $\mathbf{r^n-N}$

(r-1)'s complement: $\mathbf{(r^n-1)-N}$

complement of a complement: $\mathbf{r^n-(r^n-N)}$

Representation of a real numbers

Fixed point representation:

- Any real number can be converted to binary number system. There are two schemes to represent real number:
  - Fixed-point representation
  - Floating-point representation
- Before getting into representation of real numbers first, convert the given decimal number into a binary number.

Eg : 48.687

Step-1: successive division for integers, results 110000

Step-2: successive multiplication for decimal part results 10101111111

Result $=(48.627)_{10c}$=c$(110000.10101111111)_2$

● To store this number, we have to store two information,

i. the part before decimal point and

ii. the part after decimal point.

● This is known as fixed-point representation where the position of decimal point is fixed and number of bits before and after decimal point are also predefined.

If we use 16 bits before decimal point and 8 bits after decimal point, in signed magnitude form, the range is

$$-2^{16} - 1 \ \textit{to} \ +2^{16} - 1 \ \text{ and the precision is } 2^{(-8)}$$

One bit is required for sign information, so the total size of the number is 25 bits

(1(sign) + 16(before decimal point) + 8(after decimal point)).

Floating point representation :

- Numbers are represented by a mantissa comprising the significant digits and an exponent part of Radix R.

The format is: Mantissa **X Base $^{\textbf{Exponent}}$** **(M X B^E)**

(Or)

Mantissa X Radix $^{\text{Exponent}}$ (M X R^E)

Numbers are often normalized, such that the decimal point is placed to the right of the first non zero digit.

Eg: the decimal number,

5236 is equivalent to 5.236×10^3

(To store this number in floating point representation, we store 5236 in mantissa part and 3 in exponent part.)

IEEE standard floating point format:

IEEE has proposed two standards for representing floating-

point number:

A. Single precision

B. Double precision

A. Single precision:

S	E	M

S - 1 bit for Sign bit (0 denotes + and 1 denotes)

E - 8 bits for Exponent

M - 23 bits for mantissa

B. Double Precision:

S	E	M

S - 1 bit for Sign bit (0 denotes+ and 1 denotes)

E - 11 bits for Exponent

M - 52 bits for mantissa

Representation of Character:

- Since we are working with 0's and 1's only, to represent character in computer we use strings of 0's and 1's only

- To represent character, we are using some coding scheme, which is nothing but a mapping function.
- Some of standard coding schemes are: ASCII, EBCDIC, UNICODE.

ASCII: American Standard Code for Information Interchange.

- It uses a 7-bit code. All together we have 128 combinations of 7 bits and we can represent 128 character.

For example, 65 = 1000001 represents character 'A'.

EBCDIC: Extended Binary Coded Decimal Interchange Code.

- It uses 8-bit code and we can represent 256 character.

UNICODE: It is used to capture most of the languages of the world. It uses 16-bit

- Unicode provides a unique number for every character, no matter what the platform, no matter what the program, no matter what the language.
- The Unicode Standard has been adopted by such industry leaders as Apple, HP, IBM, JustSystem, Microsoft, Oracle, SAP, Sun, Sybase, Unisys and many others.

KEYWORDS DISCUSSED IN THIS CHAPTER

Instruction

- An instruction is a set of codes that the computer processor can understand, the code is usually in 1s and 0s, or machine language.

(or)

- Set of commands / programs that the computer processor can understand.

Processor

- A small chip that resides on the computer and other electronic devices. Its basic job is to receive input and provide appropriate output.

Program

- Program, as name suggest, are simply set of instructions developed by programmers in programming language that automate, collect, manage, calculate, analyze processing of data and information accurately.

Data

- Data, as name suggests, are information processed and stored in files or folders by computer, translated into form that is efficient for movement or processing and can be in form of text documents, images, audio clips, etc

Analog Signal

- Analog signal is a continues signal that contains time-varying quantities.

Digital Signal

- A Digital signal having discrete values at each point.
  Now we are using only digital signals which comprises of digital computers.

Control Unit

- CU is a circuitry within a computer's processor that directs operations. It instructs the memory, logic unit, and both output and input devices of the computer on how to respond to the program's instructions.

CPU

- Also called as a brain of the computer
- It performs the all types of data processing operations.

Volatile Memory

- Data will be removed when there is a power loss or system may crash.
- Eg: RAM,etc..

Non-Volatile Memory

- Data will not be removed when there is a power loss or system may crash.
  Eg; ROM,etc..

Register

- Used to store less amount of data as a temporary storage with fast access.

Table -1:

1 bit	Fundamental unit of information ,either 0 or 1
4 bits	1 Nibble
8 bits	1 Byte
1024 Bytes	1 Kilo Byte
1024 KB	1 Mega Byte
1024 MB	1 Giga Byte

Table -2:

Difference between Program and Data

Program	Data
1.Programs are collection of software instructions understandable by CPU.	1.Data are information stored in computer hard disc.
2.It is used to provide expected output whenever executed by computer.	2.It is used to help organization in effectively determining cause of problems that occurred.
3.Programs helps in decision making purpose.	3.Data does not help in any decision making purpose.
4.Programs are not simple and hard to understand.	4.Data are simple and easy to understand.
5.Different languages of program can be C, C++, Java, Pascal, Python, etc.	5.Different forms of data can be images, text documents, audio clips, software programs, Boolean data, etc.

Table -3:

Difference between fixed point and floating point representation:

S. No.	**Fixed point representation**	**Floating point representation**
1	Less computational power.	More computational power.
2	Addition of two numbers does not affect the precision.	Addition of two number, usually affect the precision.
3	Rounding and truncation must be a part of program.	It is not necessary to specify rounding and truncation.
4	Overflow error occurs because the size of intermediate register is comparatively small.	The size of intermediate register is around 80-Bits, so overflow error does not occur.
5	Requires less registers and less number of input-output pins.	Requires large registers and more number of input-output pins.

EXERCISE-1:

1. Convert the following binary numbers to decimal: 101110; 1110101;
2. Convert the following numbers with the indicated bases to decimal: $(12121)_3$, $(4310)_5$,$(50)_7$
3. Convert the following decimal numbers to binary: 1231,673 and 1998.
4. Convert the following decimal numbers to the bases indicated.

    a. 7562 to octal

b. 1938 to hexadecimal

c.175 to binary

EXERCISE-2:

1. What is the radix of the number if the solution to quadratic equation x^2 − 10x + 31 = 0 is x = 5 and x = 8? (gate 2018)
2. Obtain the 9's complement & 10’s of the following eight-digit decimal numbers: 12349876; 00980100; 90009951; and 00000000.
3. Perform the arithmetic operations (+42) + (- 13) and (-42) - (- 13) in binary using signed-2's complement representation for negative numbers.
4. covert $(293.687)_{10}$ into equivalent binary form
5. Represent the fixed point notation of unsigned binary number 1100101011 using 5-bit integer and 5-bit fraction.
6. convert into floating point representation (181.8125)

***