Computer System Architecture

Course Code IT2101

Subject Guide: Dr. Aprna Tripathi

Course Objectives:

- The core objective of this course is to describe the general organization and architecture of a computer system.
- It covers in detail the description and design of basic computer, functional units, machine instructions, control unit and memory hierarchy design.
- It provides a detailed coverage of logic circuits to perform various arithmetic operations and use of pipelining to achieve parallelism to achieve at different levels using hardware and software techniques to yield high-performance processors.

Course Outcomes:

- A student who successfully fulfills the course requirements will be able to:
- [2101.1]. Understand the concepts of different combinational and sequential circuits.
- [2101.2]. Explain various microoperations and different components of computer organization.
- [2101.3]. Describe the operations of control unit and design of various arithmetic circuits.
- [2101.4]. Analysis the concepts of I/O organization and memory organization.
- [2101.5]. Illustrate the working of pipelining, its interconnection structure and architecture of different processors.

Student Outcomes (SO):

a) An ability to apply the knowledge of mathematics, science and computing appropriate to the discipline

b) An ability to analyze a problem, identify and define the computing requirements appropriate to its solution.

c) An ability to design, implement and evaluate a system / computer-based system, process, component or program to meet desired needs

Assessment Pattern:

- At least four assignment
- Three quizzes out of which best two will be considered
- Quiz can be conducted before announcement
- In final assessment
  - Assignment- 10
  - Quiz – 10

Syllabus

Manipal University Jaipur

- **Basics of Digital Electronics:** Codes, Logic Gates, Flip-Flops, Registers, Counters, Multiplexer, De-multiplexer, Encoder, Decoder;
- **RTL and Micro Operations:** Register Transfer, Bus and Memory Transfer, Logic Micro Operations, Shift Micro Operations;
- **Basic Computer Organization:** Complete Computer Description & Design of Basic Computer, Instruction Codes, Computer Instructions, Timing & Control, Instruction Cycles, Memory Reference Instructions, Input/output & Interrupts;
- **Control Unit:** Hardwired vs. Micro Programmed Control Unit, Central Processing Unit, General Register Organization, Stack Organization, Instruction Format, Data Transfer & Manipulation, Program Control, RISC, CISC;
- **Computer Arithmetic:** Addition & Subtraction, Multiplication Algorithms, Division Algorithms;
- **Input-Output Organization:** Peripheral devices, I/O interface, Data Transfer Schemes, Program Control, Interrupt, DMA Transfer, I/O Processor;
- **Memory Unit:** Memory Hierarchy, Processor vs. Memory Speed, High-speed Memory, Cache Memory, Associative Memory, Interleave, Virtual Memory, Memory Management;
- **Introduction to Parallel Processing:** Pipelining, Characteristics of Multiprocessors, Interconnection Structures, Inter-processor Arbitration, Inter-processor Communication & Synchronization;
- **Case Studies:** Case Studies of some Contemporary Advanced Architecture for Processors of Families like Intel, AMD, IBM.

Text Books:

The main textbook is:

1. T1. M. Morris Mano, “Computer System Architecture”, Pearson, 3rd Edition Revised, 2017.

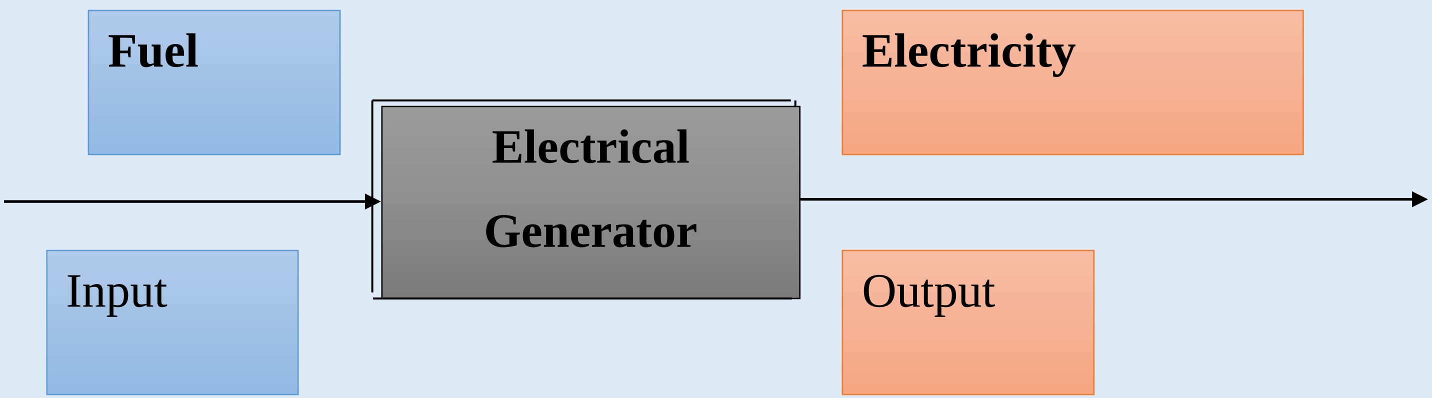
Reference Books:

1. R1. W. Stallings, “Computer Organization and Architecture –Designing for Performance”, PHI, 2009.
2. R2. S. Salivahanan & S. Arivazhagan, Digital Circuits and Design, Oxford University Press; Fifth edition, 2018.
3. R3. David A. Patterson, John L. Hennessy, “Computer Organization and Design: The Hardware/Software Interface”, Morgan Kauffmann, 4th Edition, 2010.
4. R4. John P. Hayes, “Computer Architecture and Organization”, TMH, 3rd Edition, 1999

Introduction

- **What is a System?**

System has wide variety of meanings in various contexts, but here a system is something that takes input and does some processing work and gives output.

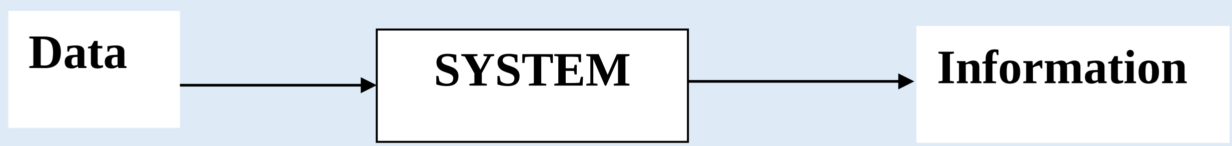


Examples: Electrical Generator
Educational Institute is a system
Doctor, etc.,

Questions: Is there any system which takes input but never produces output and Vice-versa

Cont'd...

- If any system takes data as input and producing information as output, such a processing is called "Data processing system".



What is Data?

What is Information?

Example for Data Processing Systems is Computer

What is Computer ?

Collins English Dictionary: 'A device, usually electronic, that processes data according to a set of instructions'

Que's Computer Users Dictionary: 'A machine capable of following instructions to alter data in a desirable way and to perform at least some of these operations without human intervention'.

Microsoft Press Computer Dictionary, 3rd Ed.: 'any machine that does three things: accepts structured input, processes it according to prescribed rules, and produces the results as output'.

Oxford dictionary: as "an automatic electronic apparatus for making calculations of controlling operations prescribed in numerical or logical terms.

What is a computer?

- a computer is a sophisticated electronic calculating machine that:
  - Accepts input information,
  - Processes the information according to a list of internally stored instructions and
  - Produces the resulting output information.
- Functions performed by a computer are:
  - Accepting information to be processed as input.
  - Storing a list of instructions to process the information.
  - Processing the information according to the list of instructions.
  - Providing the results of the processing as output.
- What are the functional units of a computer?

Computer Definition

Manipal University Jaipur

"A computer is a complex system incorporating diverse technologies. Typically, electronic technology is used for computation, magnetic for long-term storage, and electromechanical for input and output."

Manipal University Jaipur

Architecture & Organization

- Computer architecture deals with the functional behavior of a computer system as viewed by a programmer (like the size of a data type – 32 bits to an integer).
- Computer organization deals with structural relationships that are not visible to the programmer (like clock frequency or the size of the physical memory).

Computer Architecture & Organization

Computer Architecture & Organization

Manipal University Jaipur

- Computer Organization:
  - Design of the components and functional blocks using which computer systems are built.
  - *Analogy*: civil engineer's task during building construction (cement, bricks, iron rods, and other building materials).
- Computer Architecture:
  - How to integrate the components to build a computer system to achieve a desired level of performance.
  - *Analogy*: architect's task during the planning of a building (overall layout, floorplan, etc.).

Exercise-1

- Q1. List out core functions of computer
- Q2. Give an example of data and related process and final information generated after processing
- Set of numbers, addition, sum
- Q3. Difference between Computer Architecture and Computer organization

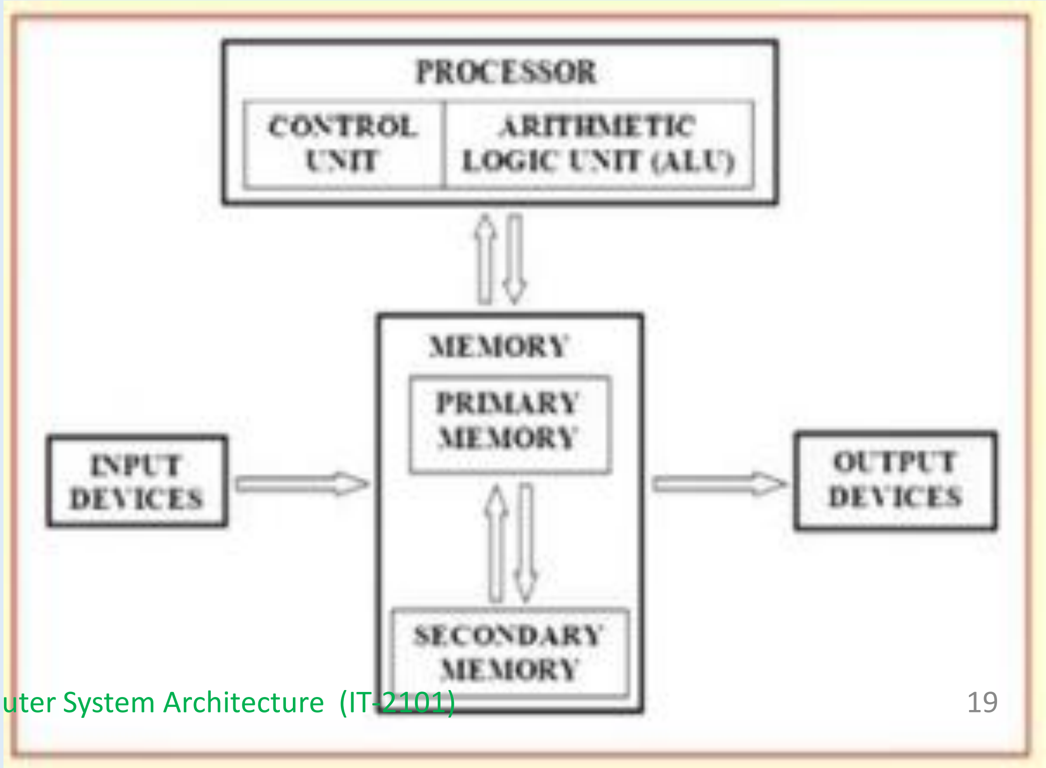
Core Functions of Computer

Functions performed by a computer are:

- Accepting information to be processed as input.
- Storing a list of instructions to process the information.
- Processing the information according to the list of instructions.
- Providing the results of the processing as output.

Von Neumann Architecture

- All most all computer designs based on concept developed by John Von Neumann referred to as "Von Neumann architecture".
- Virtually all-contemporary computers are based on concepts developed by John von Neumann who designed the IAS computer.

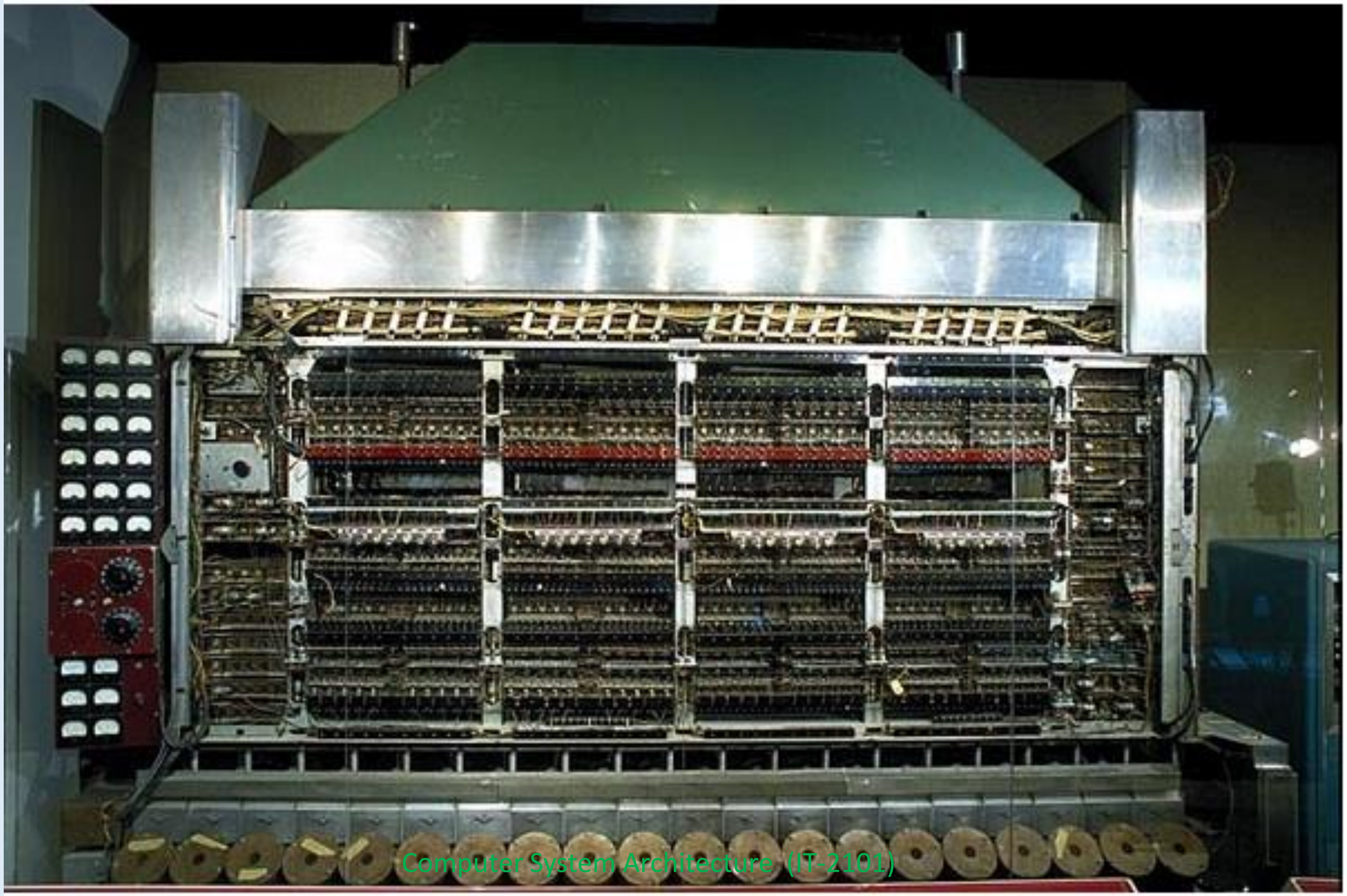


Inside the Processor

- Also called *Central Processing Unit* (CPU).
- Consists of a *Control Unit* and an *Arithmetic Logic Unit* (ALU).
  - All calculations happen inside the ALU.
  - The Control Unit generates sequence of control signals to carry out all operations.
- The processor fetches an instruction from memory for execution.
  - An instruction specifies the exact operation to be carried out.
  - It also specifies the data that are to be operated on.
  - A program refers to a set of instructions that are required to carry out some specific task (e.g. sorting a set of numbers).

IAS Computer, 1952

The IAS Computer, 1952

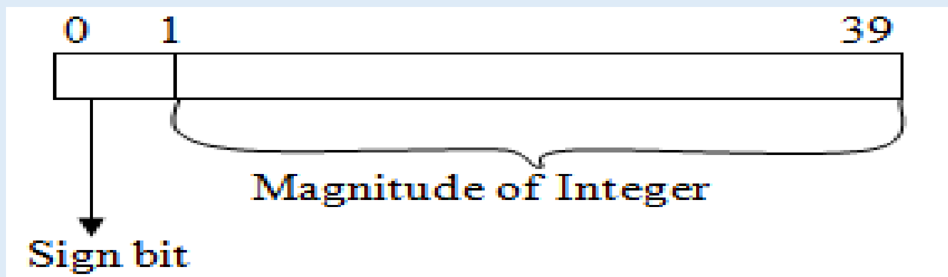


IAS Computer

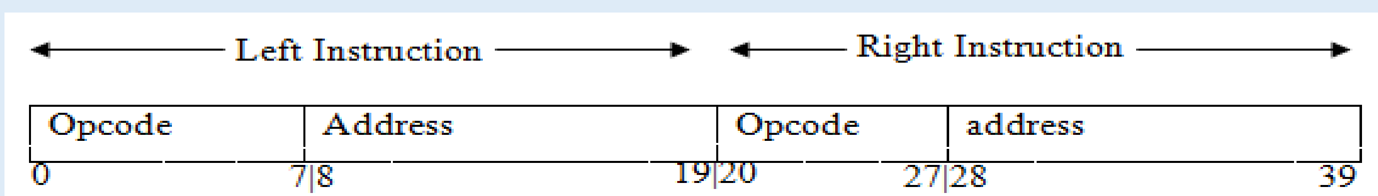
- Developed by John Von Neumann in 1940 at Princeton University.
- In IAS computer, IAS stands for Institute for Advanced Studies.

Characteristics of IAS Computer

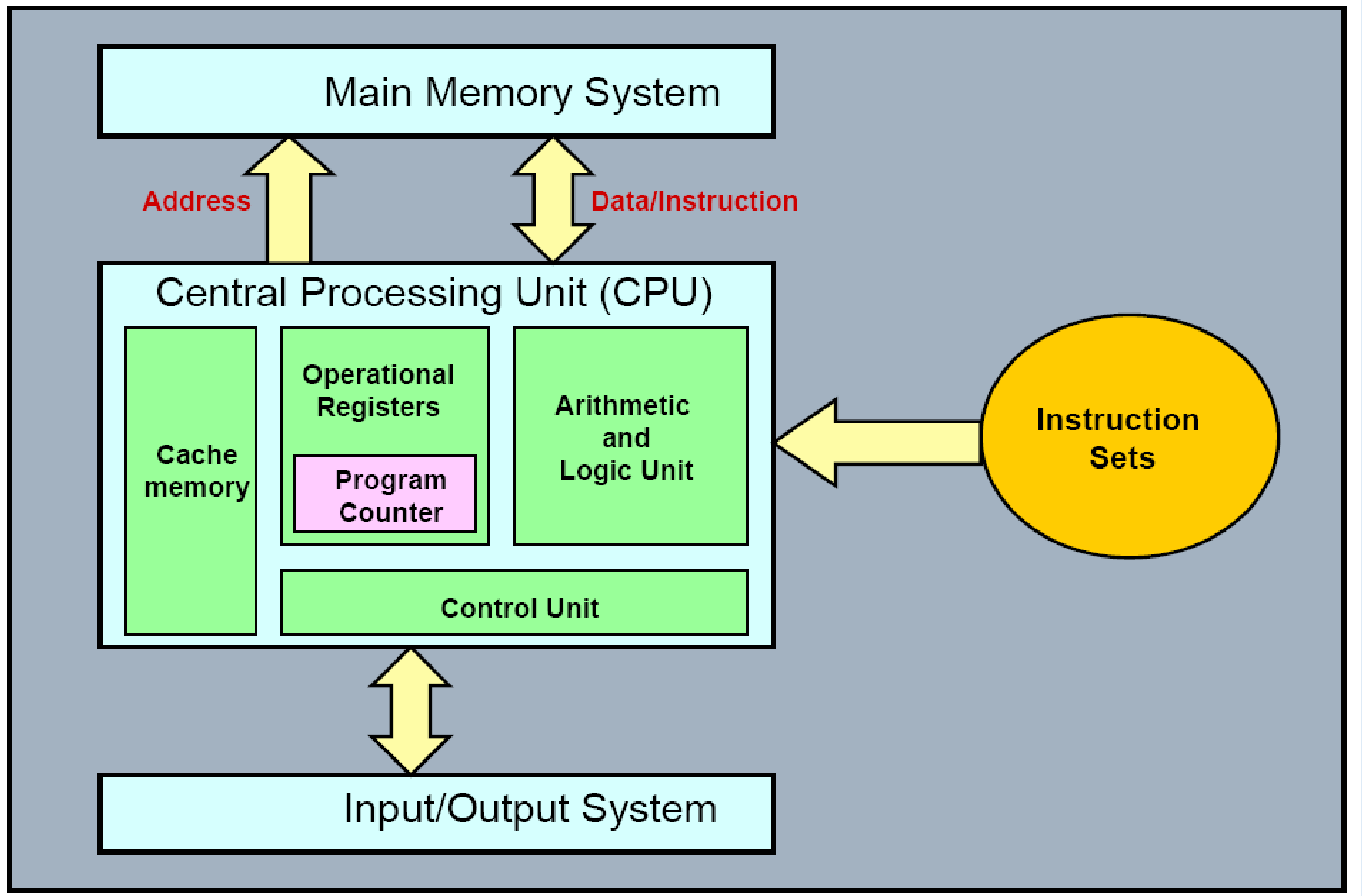
- 1000 words
- 40 bit word length
- A sign bit and a 39-bit value represent each number.



- 20-bit instruction, 8-bit operation code (opcode)
  12-bit address



Functional components of a computer



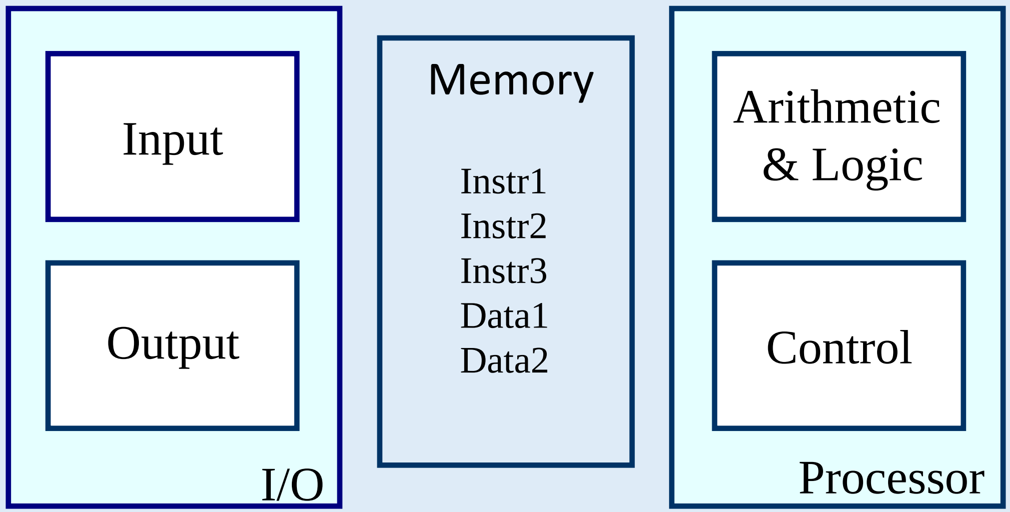
Functional units of a computer

Input unit accepts information:

- Human operators,
- Electromechanical devices (keyboard)
- Other computers

Arithmetic and logic unit(ALU):

- Performs the desired operations on the input information as determined by instructions in the memory



Output unit sends results of processing:

- To a monitor display,
- To a printer

Stores information:

- Instructions,
- Data

Control unit coordinates various actions

- Input,
- Output
- Processing

Information in a computer -- *Instructions*

Manipal University Jaipur

- Instructions specify commands to:
  - Transfer information within a computer (e.g., from memory to ALU)
  - Transfer of information between the computer and I/O devices (e.g., from keyboard to computer, or computer to printer)
  - Perform arithmetic and logic operations (e.g., Add two numbers, Perform a logical AND).
- A sequence of instructions to perform a task is called a program, which is stored in the memory.
- Processor fetches instructions that make up a program from the memory and performs the operations stated in those instructions.
- What do the instructions operate upon?

Information in a computer -- **Data**

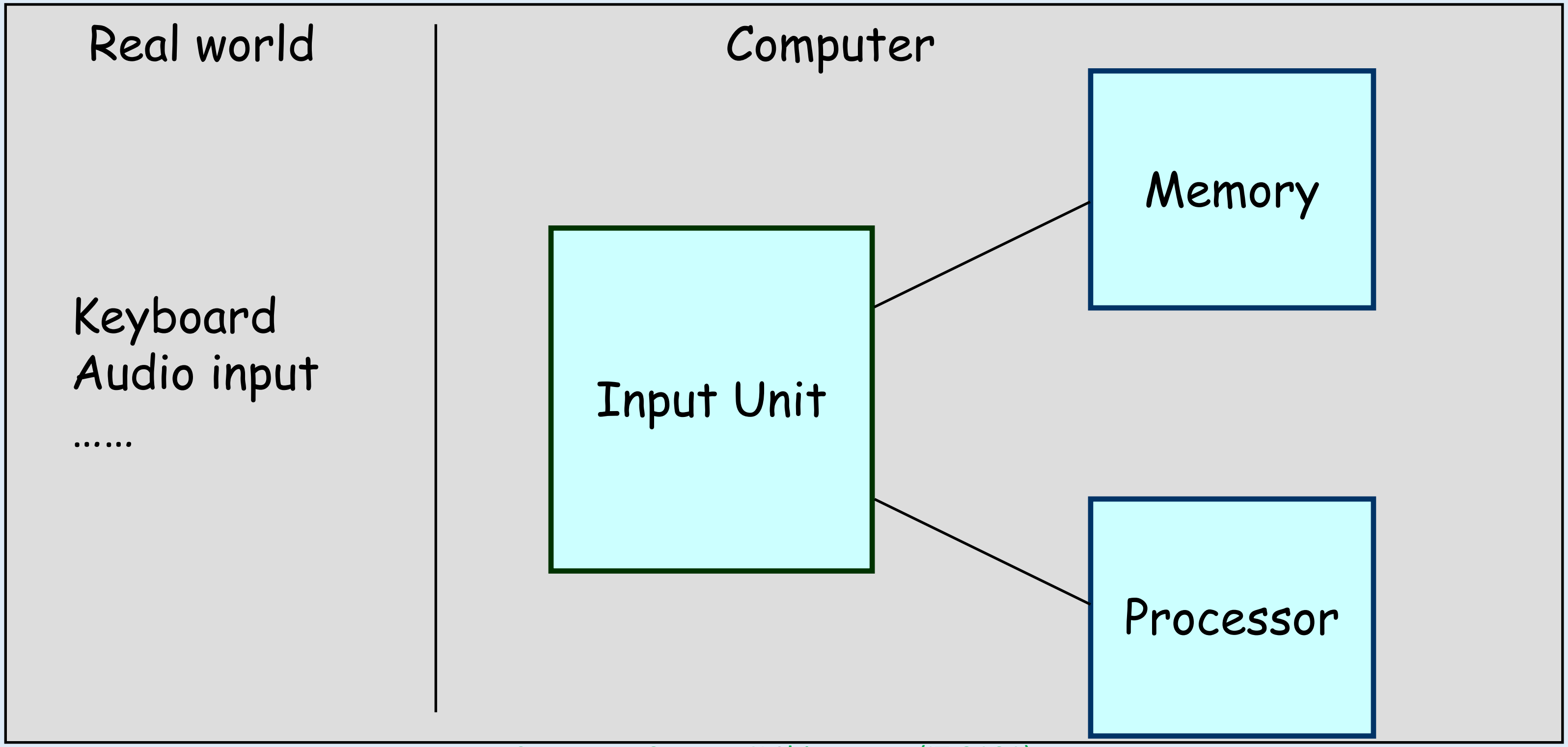
Manipal University Jaipur

- Data are the “operands” upon which instructions operate.
- Data could be:
  - Numbers,
  - Encoded characters.
- Data, in a broad sense means any digital information.
- Computers use data that is encoded as a string of binary digits called bits. Signal – on or off (logical representation 0 or 1)
- 1 Byte= 8 bits
- Operation--- operator and operands.
- -- unary operator ; ++, i++= > i= i+1
- ---Binary operator s= a+b

Input unit

Binary information must be presented to a computer in a specific format. This task is performed by the input unit:

- Interfaces with input devices.
- Accepts binary information from the input devices.
- Presents this binary information in a format expected by the computer.
- Transfers this information to the memory or processor.



Memory unit

- Memory unit stores instructions and data.
  - Recall, data is represented as a series of bits.
  - To store data, memory unit thus stores bits.
- Processor reads instructions and reads/writes data from/to the memory during the execution of a program.
  - In theory, instructions and data could be fetched one bit at a time.
  - In practice, a group of bits is fetched at a time.
  - Group of bits stored or retrieved at a time is termed as "word"
  - Number of bits in a word is termed as the "word length" of a computer.
- In order to read/write to and from memory, a processor should know where to look:
  - "Address" is associated with each word location.

Memory unit (contd..)

- Processor reads/writes to/from memory based on the memory address:
  - Access any word location in a short and fixed amount of time based on the address.
  - Sequential access memory: magnetic Tap(2005-10)
  - Random Access Memory (RAM) provides fixed access time independent of the location of the word.
  - Access time is known as "Memory Access Time".
- Memory and processor have to "communicate" with each other in order to read/write information.
  - In order to reduce "communication time", a small amount of RAM (known as Cache) is tightly coupled with the processor.
- Modern computers have three to four levels of RAM units with different speeds and sizes:
  - Fastest, smallest known as Cache
  - Slowest, largest known as Main memory.

Memory unit (contd..)

Manipal University Jaipur

- Primary storage of the computer consists of RAM units.
  - Fastest, smallest unit is Cache.
  - Slowest, largest unit is Main Memory.
- Primary storage is insufficient to store large amounts of data and programs.
  - Primary storage can be added, but it is expensive.
- Store large amounts of data on secondary storage devices:
  - Magnetic disks and tapes,
  - Optical disks (CD-ROMS).
  - Access to the data stored in secondary storage in slower, but take advantage of the fact that some information may be accessed infrequently.
- Cost of a memory unit depends on its access time, lesser access time implies higher cost.

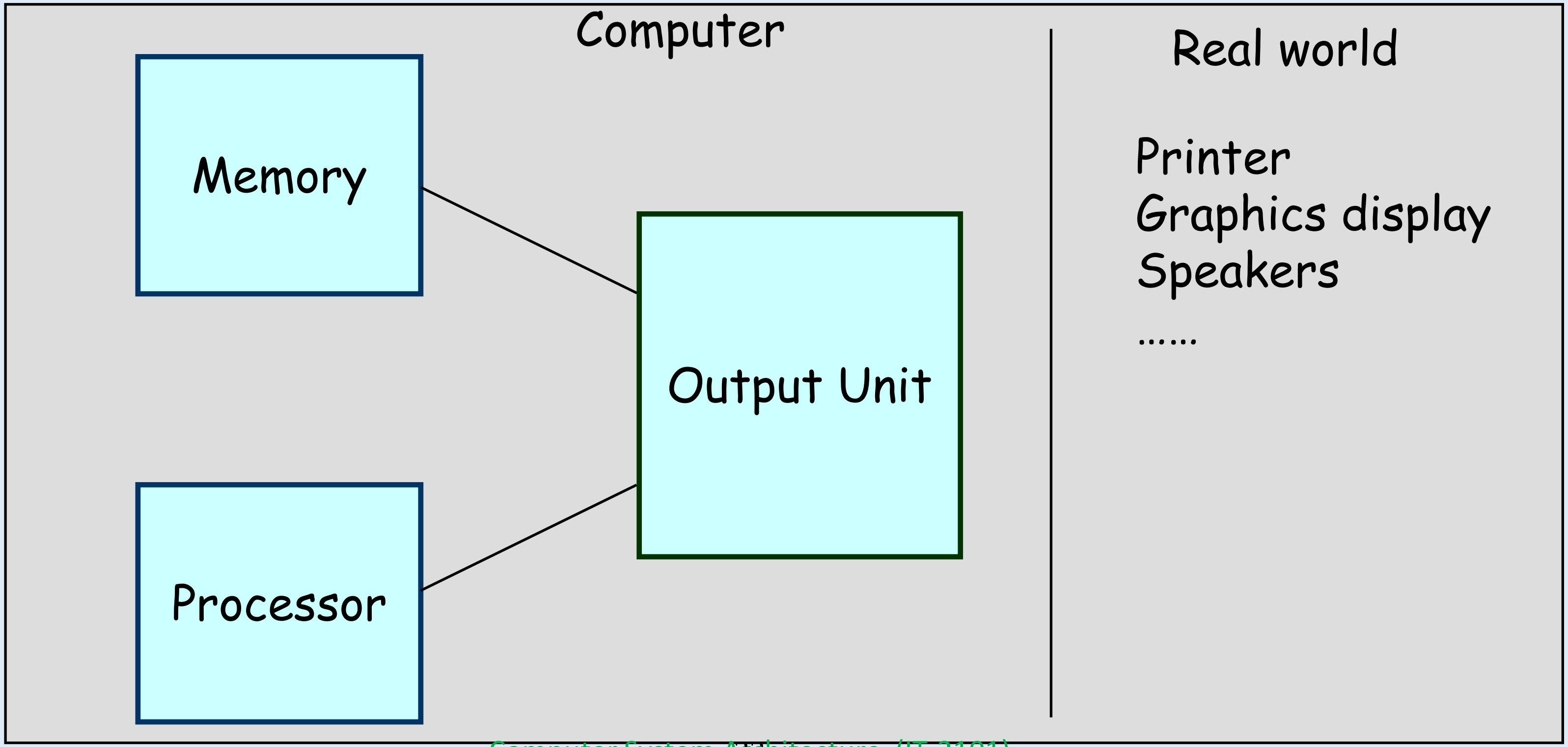
Arithmetic and logic unit (ALU)

Manipal University Jaipur

- Operations are executed in the Arithmetic and Logic Unit (ALU).
  - Arithmetic operations such as addition, subtraction.
  - Logic operations such as comparison of numbers.
- In order to execute an instruction, operands need to be brought into the ALU from the memory.
  - Operands are stored in general purpose registers available in the ALU.
  - Access times of general purpose registers are faster than the cache.
- Results of the operations are stored back in the memory or retained in the processor for immediate use.

Output unit

•Computers represent information in a specific binary form. Output units:
- Interface with output devices.
- Accept processed results provided by the computer in specific binary form.
- Convert the information in binary form to a form understood by an output device.



Control unit

- Operation of a computer can be summarized as:
  - Accepts information from the input units (Input unit).
  - Stores the information (Memory).
  - Processes the information (ALU).
  - Provides processed results through the output units (Output unit).
- Operations of Input unit, Memory, ALU and Output unit are coordinated by Control unit.
- Instructions control "what" operations take place (e.g. data transfer, processing).
- Control unit generates timing signals which determines "when" a particular operation takes place.

Module-1 (**Basics of Digital Electronics**)

- In this Module, we have lectures on
  - :
  - Codes,
  - Logic Gates,
  - Flip-Flops,
  - Registers,
  - Counters,
  - Multiplexer,
  - De-multiplexer,
  - Encoder,
  - Decoder;

Lecture -1

Codes and Logic Gate

CODES

- In general, we understand or use digital data, but computer system understand binary.
- So, the digital data is represented and stored as group of bits called binary codes.
- Code is a symbolic representation of discrete information, which may be present in the form of numbers, letters or physical quantities.
- The symbols used are the binary digits 0 and 1 which are arranged according to the rules of codes.
- Codes are broadly classified into five groups:

  (i) Weighted Binary Codes

  (ii) Non-weighted Codes

weighted binary codes

- Weights are attached to each binary digits. Bits are multiple by the weights indicates; the sum of these weighted bits given the equivalent decimal digits.
- BCD code : represented using 4 bits and follows 8421 sequence or weights are attached t each binary digits as per 8421 sequence.

4th bit	3rd bit	2nd bit	1st bit
2^3	2^2	2^1	2^0

5----0101

Some weighted 4-bit binary codes

69
01101001
10
1010
00010
15.45
0001 0101. 0100
0101

Decimal number	8421(BCD code)	5421
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

Manipal University Jaipur

Non-weighted Codes

- Non-weighted codes are codes that are not positionally weighted.
- This means that each position within a binary number is not assigned a fixed value.
- Examples of non-weighted codes:
  - ✓Excess-3 codes
  - ✓Gray codes

Excess-3 codes:
obtain by adding 3 to a decimal no.

Decimal number	8421(BCD CODE)	Excess 3 code
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

- Excess 3 code of 643
- =➔ add 3 to every digit
- 976
- 9 7 6
- 12 10 9
- 1100 1010 1001
- 1001 0111 0110

10
+3
13
1101

1 0
3 3

4 3
0100 0011

Gray Code

Manipal University Jaipur

- The gray code belongs to a class of codes called minimum change code.
- In which only one bit in the code group changes when moving from one step to the next.
- In gray code two adjacent code number, differ by only one bit. So it is called unit-distance code.

Decimal number	Binary code	Gray code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100

Conversion of a binary number to gray code

5
B--0101
G—0111

B---10110
G---11101

- **A binary number can be converted to its Gray code when:**

  (i) The first bit (MSB) of the Gray code is the same as the first bit of the binary number;

  (ii) The second bit of the Gray code equals the exclusive-OR, of the first and second bits of the binary number, i.e. it will be 1 if these binary code bits are different and 0 if they are the same;

  (iii) The third Gray code bit equals the exclusive-OR of the third bits of the binary number. And so no

Conversion from Gray code to binary

Manipal University Jaipur

G--0111

B---0101

G---1011

B-- 1101

- **Conversion of a Gray code into its binary form involves the reverse of the previous procedure:**

  (i) The first binary bit (MSB) is the same as that of the first Gray code bit.

  (ii) If the second Gray bit is 0, the second binary bit is the same as that of the first binary; if the second Gray bit is 1, the second binary bit is the inverse of its first binary bit.

  (iii) Step 2 is repeated for each successive bit.

  1011----1101

Examples

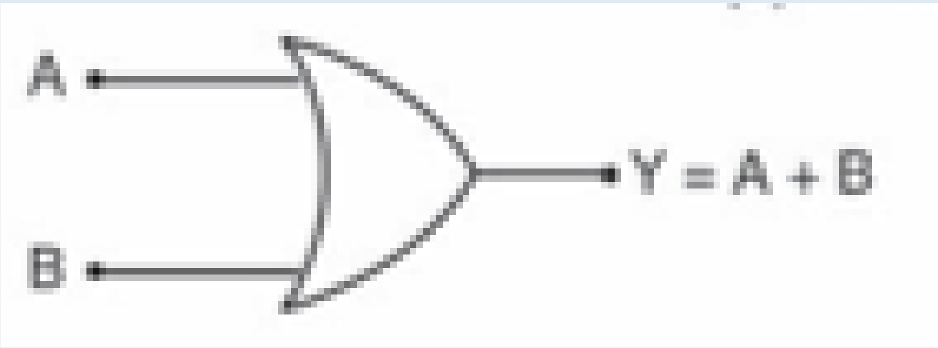
- Binary to gray
- A. 10110 B. 10101101

- Gray to binary
- 110101
- 1010111

Logic Gates-OR Gate

If A and B are the input variables of an OR gate and Y is its output, then

$$Y = A + B$$



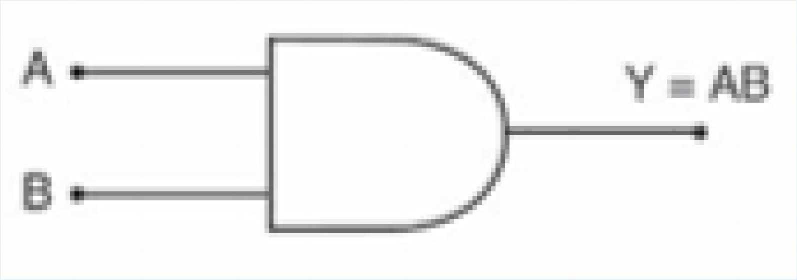
Input A	INPUT B	OUTPUT Y
0	0	0
0	1	1
1	0	1
1	1	1

Logic Gates- AND Gate

Manipal University Jaipur

If A and B are the input variables of an AND gate and Y is its output, then

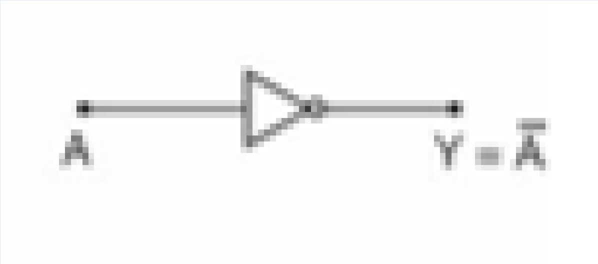
$$Y = A \cdot B$$



Input A	INPUT B	OUTPUT Y
0	0	0
0	1	0
1	0	0
1	1	1

Logic Gates- NOT Gate (Inverter)

A NOT gate using a transistor is shown in Fig. in which A represents the input and Y represents the output, i.e. = A’



Input A	OUTPUT Y
0	1
1	0

Demorgan’s Theorems

Manipal University Jaipur

- The first theorem states that the complement of a product is equal to the sum of the complements. That is, if the variables are A and B, then

$$\overline{AB} = \overline{A} + \overline{B}$$

- The second theorem states that, the complement of a sum is equal to the product of the complements. In equation form, this can be written as

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

Logic Gates- NAND Gate

Manipal University Jaipur

NAND is a contraction of the NOT–AND gates. It has two or more inputs and only one output, i.e. *Y* = (*A* .*B*)’

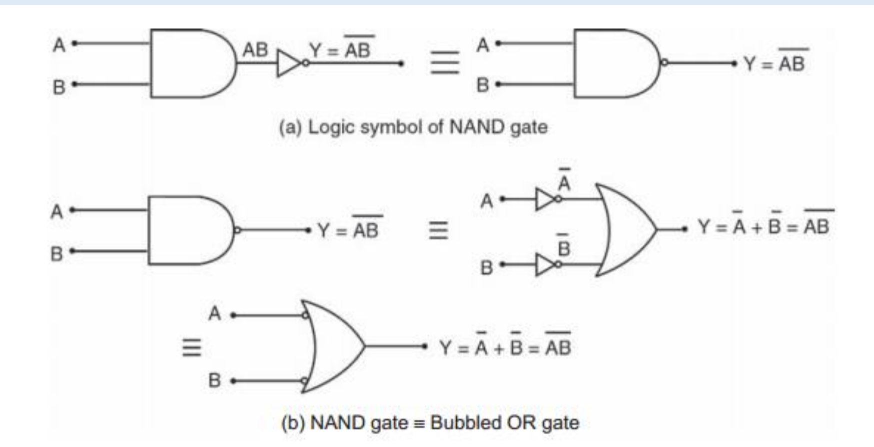


Fig. 3.5 NAND gate

Input A	INPUT B	OUTPUT Y
0	0	1
0	1	1
1	0	1
1	1	0

Logic Gates- NOR Gate

NOR is a contraction of NOT–OR gates. It has two or more inputs and only one output, i.e. $Y = (A+B)'$

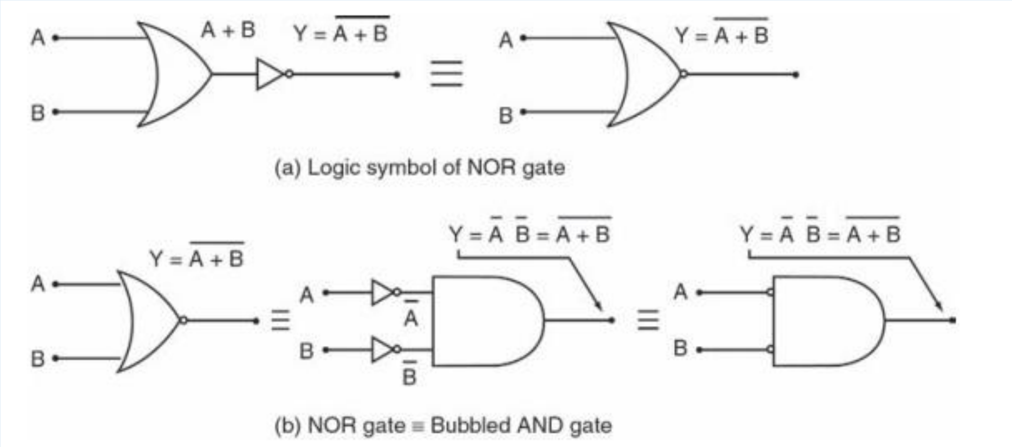


Fig. 3.6 NOR gate

Input A	INPUT B	OUTPUT Y
0	0	1
0	1	0
1	0	0
1	1	0

Universal Gates / Universal Building Blocks

NAND and NOR gates are called Universal gates or universal building blocks because both can be used to implement any gate like AND,OR and NOT gates or any combination of these basic gates.

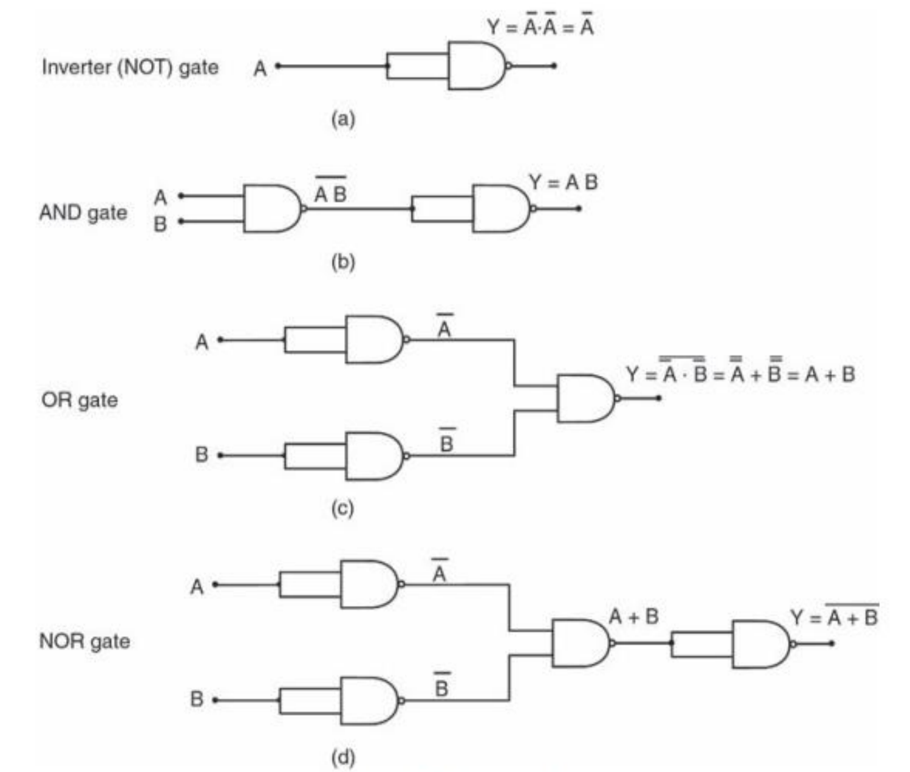


Fig. 3.7 Realisation of
(a) NOT, (b) AND, (c) OR and (d) NOR gates using NAND gates

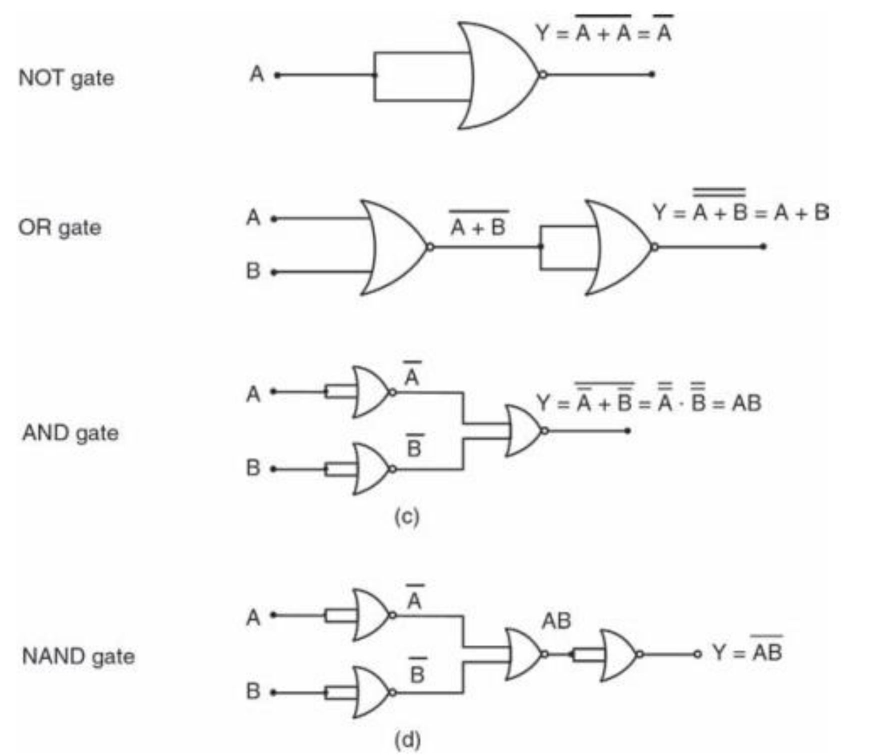
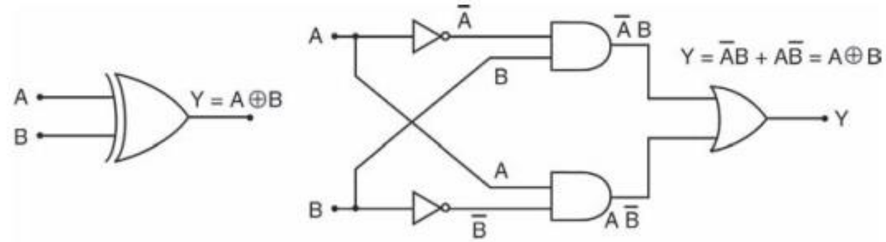


Fig. 3.8 Realisation of
(a) NOT, (b) OR, (c) AND and (d) NAND gates using NOR gates

Exclusive-OR (Ex-OR) Gate

An Exclusive-OR gate is a gate with two or more inputs and one output. The output of a two-input Ex-OR gate assumes a HIGH state if one and only one input assumes a HIGH state. This is equivalent to saying that the output is HIGH if either input A or input B is HIGH exclusively, and low when both are 1 or 0 simultaneously.



(a) Logic symbol (b) Using AND-OR-NOT gates

Fig. 3.11 Ex-OR gate

Input A	INPUT B	OUTPUT Y
0	0	0
0	1	1
1	0	1
1	1	0

EX OR GATE

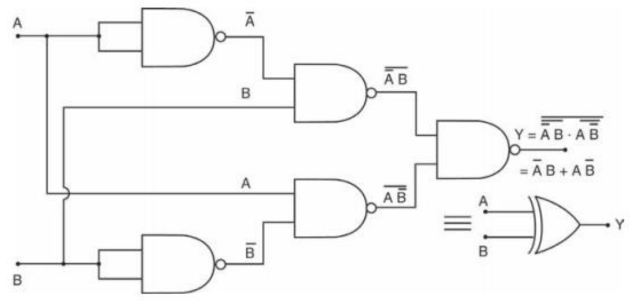
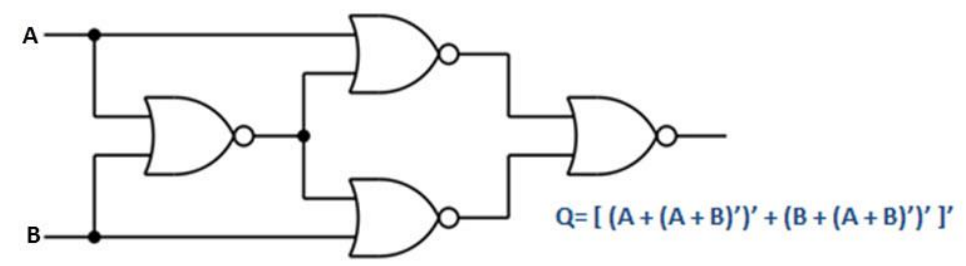


Fig. 3.12 Ex-OR gates using NAND gates



EX-OR GATE USING NOR GATE

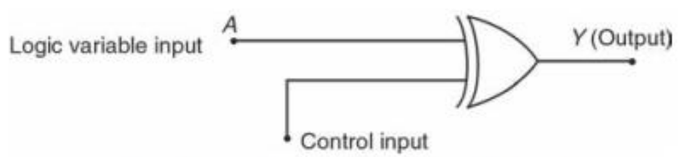


Fig. 3.14 Ex-OR gate as a controlled inverter

Exclusive-NOR (Ex-NOR) Gate

Manipal University Jaipur

The exclusive-NOR gate, is an Ex-OR gate, followed by an inverter. An exclusive-NOR gate has two or more inputs and one output. The output of a two-input Ex-NOR gate assumes a HIGH state if both the inputs assume the same logic state or have an even number of 1s, and its output is LOW when the inputs assume different logic states or have an odd number of 1s

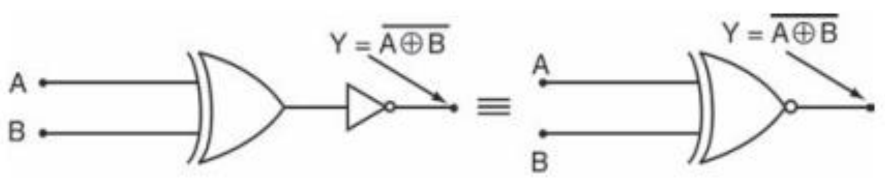


Fig. 3.15 Logic symbol of 2-input Ex-NOR gate

Input A	INPUT B	OUTPUT Y
0	0	1
0	1	0
1	0	0
1	1	1

Example 3.5 Realise (a) $Y = A + BC\overline{D}$ using NAND gates and

(b) $Y = (A + C)(A + \overline{D})(A + B + \overline{C})$ using NOR gates.

Solution Using NAND gates

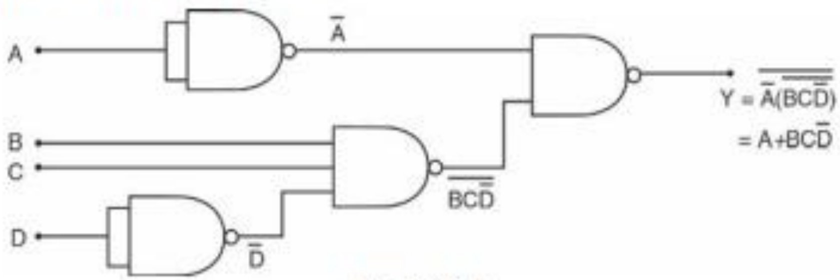


Fig. E3.5(a)

(b) Using NOR gates

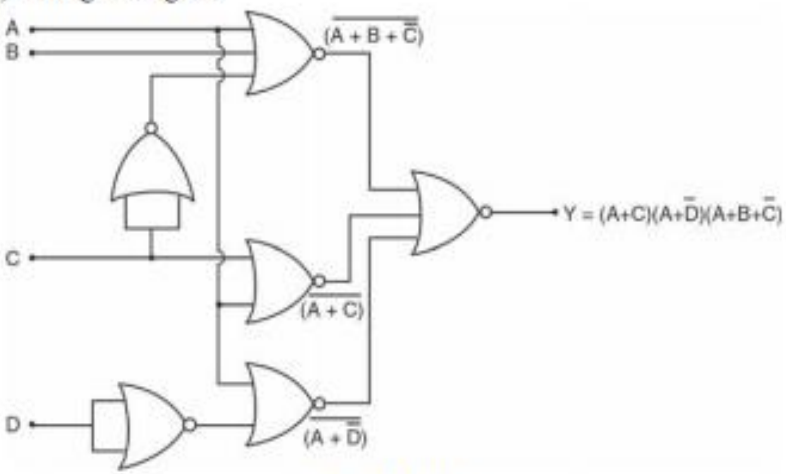


Fig. E3.5(b)

Example 3.2 Realise the logic expression $Y = (A + B)(\overline{A} + C)(B + D)$ using basic gates.

Solution In the given expression, there are 3 sum terms which can be implemented using three 2-input OR gates and their outputs are AND operated together by a 3-input AND gate. A NOT gate can be used to obtain the inverse of *A*. Now, the realised circuit is shown in Fig. E3.2.

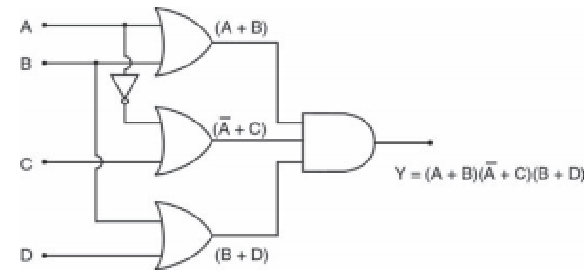


Fig. E3.2

Example 3.3 Implement $Y = \overline{AB} + A + (\overline{B + C})$ using NAND gates only.

Solution The implementation of the given function is shown in Fig. E3.3.

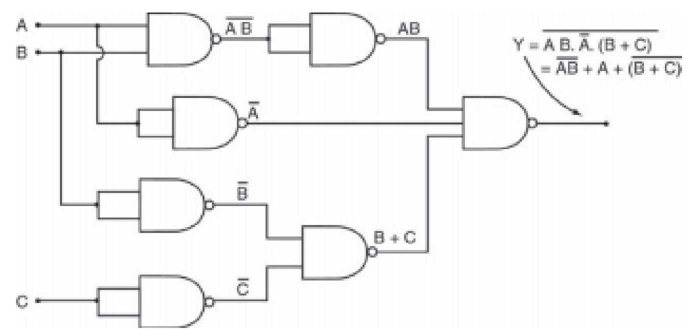


Fig. E3.3