JGi JAIN DEEMED-TO-BE UNIVERSITY | Powered By Futurense

Revolutionising B.Tech

Module 1: Fundamentals and basics in COA

Course Name: Computer Architecture and organization[22CSE104]

Total Hours : 12

JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

Table of Content

JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

Table of Content

Aim

To equip students in the fundamentals and understanding the Concepts of Interconnections of Computers and make them to design the Logic Gates.

JGI JAIN DEEMED-TO-BE UNIVERSITY
Powered By Futurense

Objective

a. Discuss on the various basic concepts and Structure of Computers.
b. Understanding different types of Logic Functions and Synthesis Techniques
c. Understanding and practice of Designing Encoders and Demutiplexers

Computer Organization

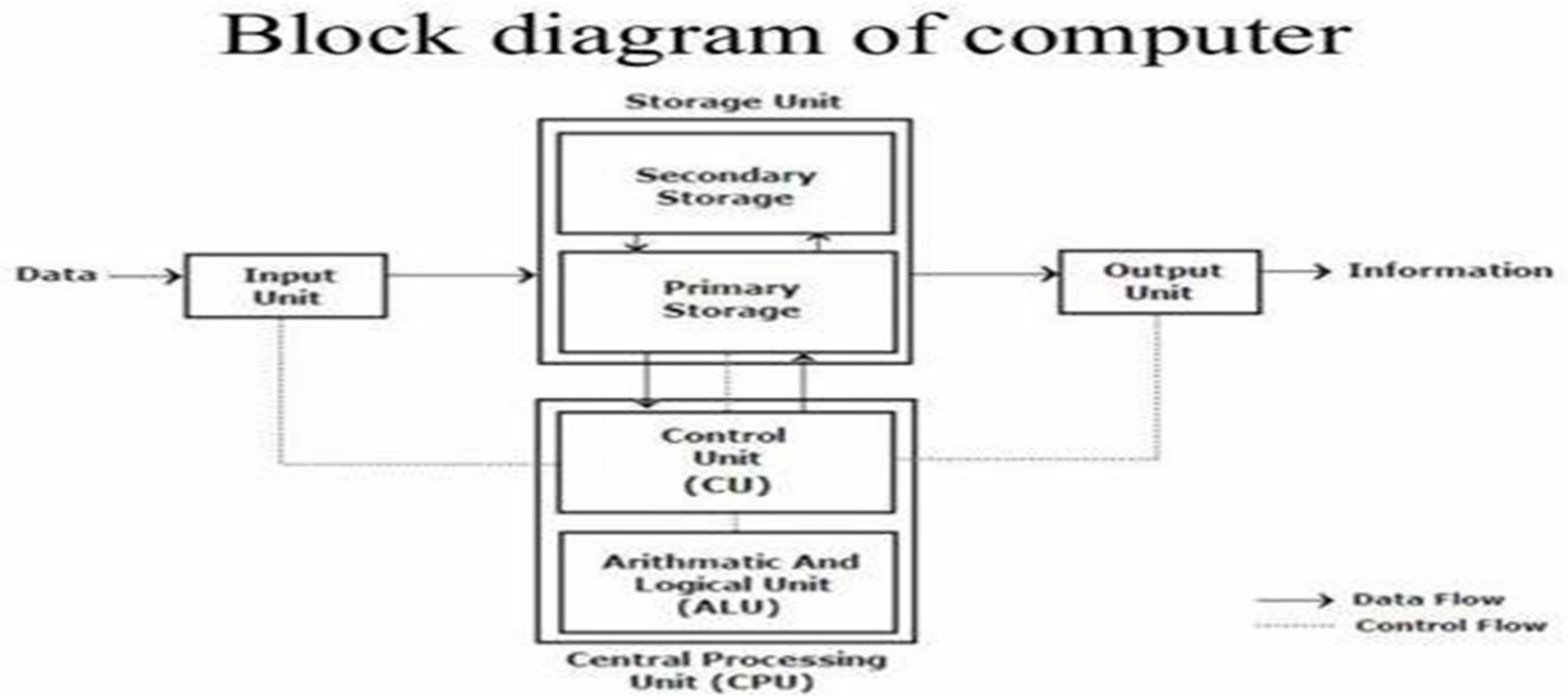
- Computer organization is concerned with the way that the hardware components of a computer system are arranged and interact with each other to carry out tasks.
- This includes the design of individual components such as the CPU, memory, and input/output devices, as well as the way that they are connected to each other to form a functioning system.
- Computer organization also deals with issues such as how data is stored and accessed, and how instructions are executed.
- Computer organization is concerned with the physical implementation like Circuit Design, Peripherals and Adders.
- Interconnection and communication between components, such as the bus structure, memory hierarchy, and input/output systems.
- It is frequently called as Micro Architecture.

JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

Computer Architecture

- Computer architecture is concerned with the overall design and structure of a computer system, including not only the hardware components, but also the software that controls them and the way that users interact with the system.
- Computer architecture is concerned with issues such as how data is processed, how programs are executed, and how the system is designed to meet the needs of different types of users and applications.
- It involves the logical functions such as Instruction sets, Data types, Registers and Addressing modes.
- Computer architecture is concerned with optimizing the performance of a computer system and ensuring that it can execute instructions quickly and efficiently.
- Computer Architecture is also called Instruction Set Architecture (ISA).

Functional Units of a Computer



Input Unit

- Computers accept coded information through input units.
- The most common input device is the keyboard.
- When we give the input the corresponding letter or digit is automatically translated into its corresponding binary code and transmitted to the processor.
- Human-computer interaction are available, including the touchpad, mouse, joystick, and trackball.
- Graphic input devices in conjunction with displays.
- Microphones can be used to capture audio input which is then sampled and converted into digital codes for storage and processing.
- Cameras can be used to capture video input.
- Digital communication facilities, such as the Internet, can also provide input to a computer from other computers and database servers.

Memory Unit

- The function of the memory unit is to store programs and data.
- There are two classes of storage, called primary and secondary.
- Primary Memory Primary memory, also called main memory, is a fast memory that operates at electronic speeds.
- The number of bits in each word is referred to as the word length of the computer, typically 16, 32, or 64 bits.
- A memory in which any location can be accessed in a short and fixed amount of time after specifying its address is called a random-access memory (RAM).
- The time required to access one word is called the memory access time. (MAT)
- This time is independent of the location of the word being accessed. It typically ranges from a few nanoseconds (ns) to about 100 ns for current RAM units

JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

Cache memory

- Smaller, faster RAM unit, called a cache, is stored between CPU and main memory.
- The cache is tightly coupled with the processor and is usually contained on the same integrated-circuit chip.
- The purpose of the cache is to facilitate high instruction execution rates at faster Execution rates.
- At the start of program execution, the cache is empty.
- As execution proceeds, instructions are fetched into the processor chip, and a copy of each is placed in the cache.
- When the execution of an instruction requires data, located in the main memory, the data are fetched and copies are also placed in the cache.

Secondary Storage

- Secondary Storage are additional, less expensive is used to store large amounts of data .
- Access times for secondary storage are longer than for primary memory.
- The devices available are including magnetic disks, optical disks (DVD and CD), and flash memory devices.

JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

Arithmetic and Logic Unit(ALU)

- Computer operations are executed in the arithmetic and logic unit (ALU) of the processor.
- Arithmetic operations Addition, Subtraction, Multiplication, Division.
- Logical Operations AND,OR,NOT
- When operands are brought into the processor, they are stored in high-speed storage elements called registers.
- Each register can store one word of data.
- Access times to registers are even shorter than access times to the cache unit on the processor chip.

JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

Control Unit

- The control unit controls and coordinate all the activities that sends control signals to other units.
- Memory, arithmetic and logic, and I/O units store and process information and perform input and output operations.
- Control units are responsible for generating the timing signals that govern the transfers.
- Data transfers between the processor and the memory are also managed by the control unit through timing signals.
- A large set of control lines carries the signals used for timing and synchronization of events in all units.

JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

Output Unit

- Output unit function is to send processed results to the outside world.
- A familiar example of such a device is a printer.
- laser printers, or ink jet streams. Such printers may generate output at speeds of 20 or more pages per minute.
- Graphic displays provide both an output function showing text and graphics and an input function, through touchscreen capability.

JGI JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

Summarisation of Operations of a Computer

- The operation of a computer can be summarized as follows:
- The computer accepts information in the form of programs and data through an input unit and stores it in the memory.
- Information stored in the memory is fetched and send to arithmetic and logic unit where it is processed.
- Processed information leaves the computer through an output unit.
- All activities in the computer are directed by the control unit.

JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

Basic operational Concepts

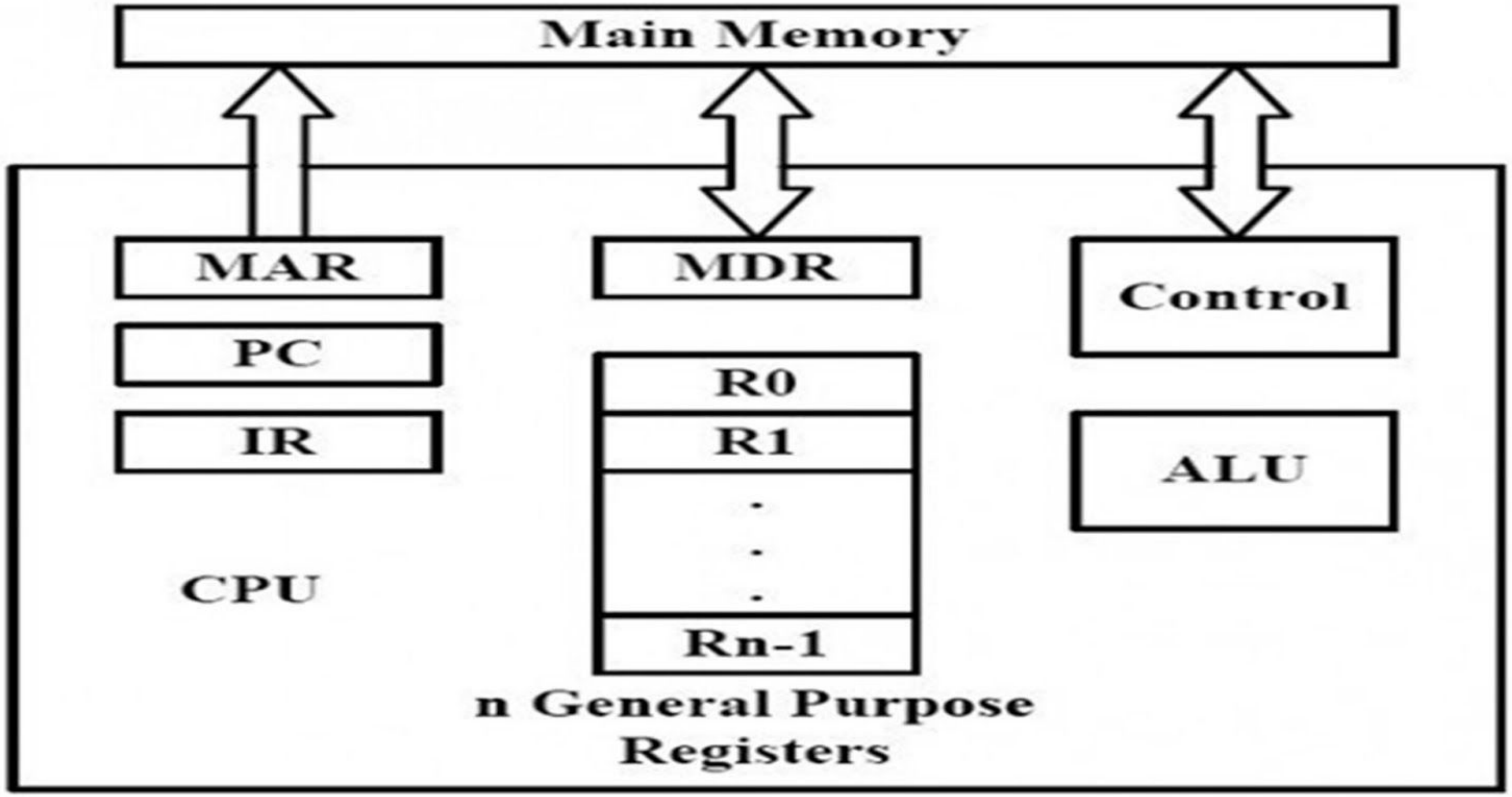


Figure: Connection between MM & Processor

Registers

- Register serves as a quick memory for accepting, storing, and sending data and instructions to the CPU .
- A register is a collection of flip-flops, Single bit digital data is stored using flip-flops.
- By combining many flip-flops, the storage capacity can be extended to accommodate a huge number of bits.

Operations performed in Registers

- **Fetch:** The Fetch Operation is used to retrieve user-provided instructions that have been stored in the main memory. Registers are used to fetch these instructions.
- **Decode:** The Decode Operation is used to interpret the Instructions, which means that the CPU will determine which Operation has to be carried out on the Instructions after the Instructions have been decoded.
- **Execute:** The CPU manages the Execute Operation. The results that the CPU generates are then stored in the memory before being presented on the user screen.

Types of Registers

- **Types of Registers:**
- Status and control registers.
- General-purpose data registers.
- Special purpose register.

Status and control register

- Status and control register report and allow the modification of the state of the processor and of the Program to be executed.
- Instruction Format:
- Address 0-11bits
- Opcode 12-14 bits
- I 14-15 bits

I	OPCODE	ADDRESS

General-Purpose Data Registers

- **General-Purpose Data Registers:**
- General purpose registers are extra registers that are present in the CPU and are utilized anytime data or a memory location is required.
- They are used to store the data temporarily.
- Operands for logical and arithmetic operations
- Operands for address calculation
- Memory pointers

Types of special purpose Register

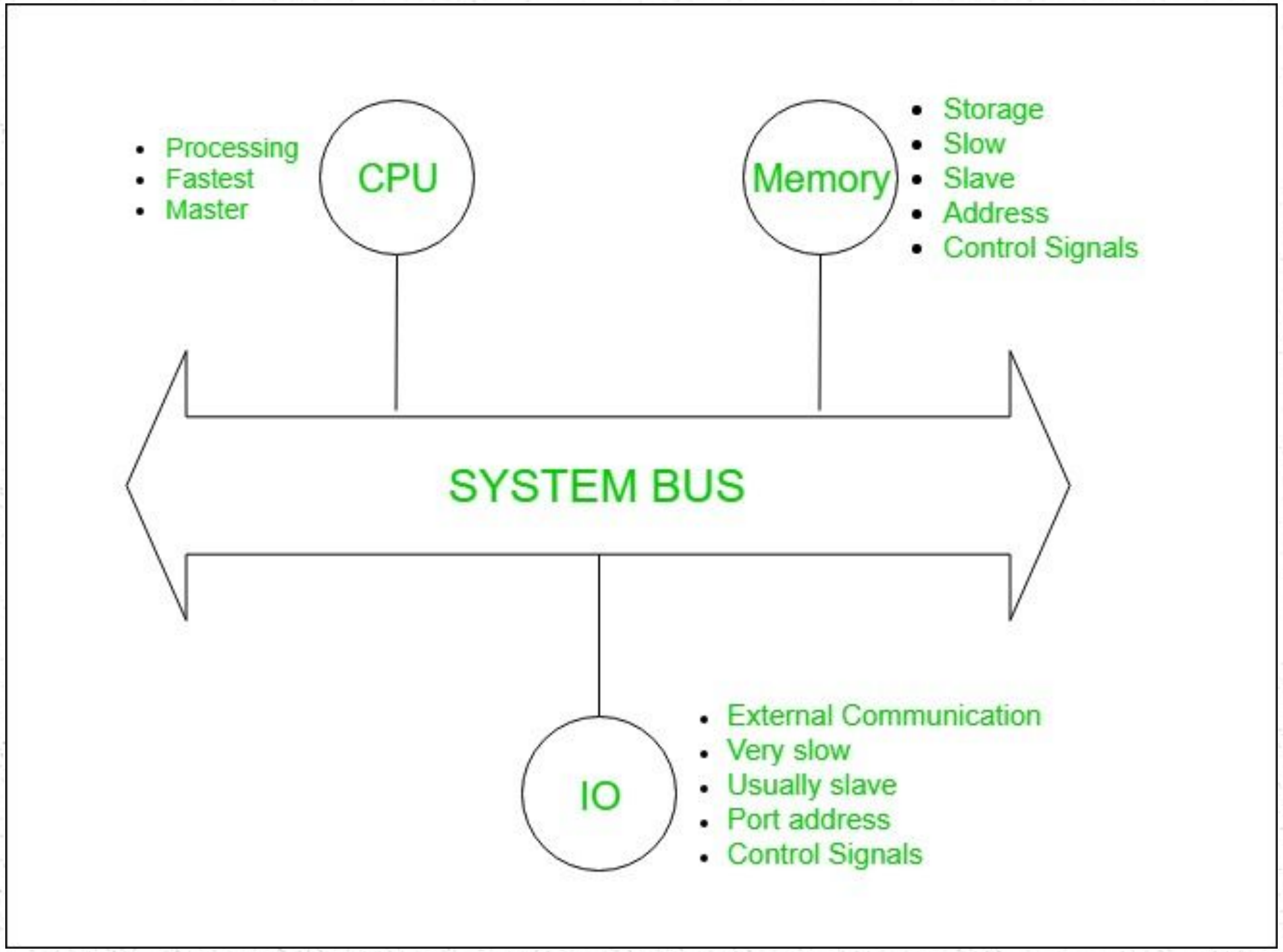
- Special Purpose Registers:
- PC (program Counter) Keep track of the program Which are being Executed.
- It contains the address of the Next Instruction to be fetched and Executed.
- IR (Instruction Register)contains of the Instructions which is currently being Executed.
- MAR(Memory Address Register)Address of the Location from the Memory.
- MDR(Memory Data Register)Data which is read from the main memory.

JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

BUS Structure

- The bus is a communication channel.
- The characteristic of the bus is shared transmission media.
- The limitation of a bus is only one transmission at a time.
- A bus used to communicate between the major components of a computer is called a **System bus**.

Block Diagram of Bus Structure



JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

Categories of Bus Structure

System bus contains 3 categories of lines used to provide the communication between the CPU, memory and IO named as:

1. Address lines (AL) **2.** Data lines (DL) **3.** Control lines (CL)

1. Address Lines:

- Used to carry the address to memory and IO.
- Unidirectional.
- Based on the width of an address bus we can determine the capacity of a main memory

JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

Data Lines

- **2. Data Lines:**
- Used to carry the binary data between the CPU, memory and IO.
- Bidirectional.
- Based on the width of a data bus we can determine the word length of a CPU.
- Based on the word length we can determine the performance of a CPU.

Control Lines

- **3. Control Lines:**

- Used to carry the control signals and timing signals
- Control signals indicate the type of operation.
- Timing Signals are used to synchronize the memory and IO operations with a CPU clock.
- Typical Control Lines may include Memory Read/Write, IO Read/Write, Bus Request/Grant, etc.

Performance

- Performance refers to the speed and efficiency at which a computer system can execute tasks and process data.

1. **Processor speed:** The speed of the processor, measured in GHz (gigahertz), determines how quickly the computer can execute instructions and process data.
2. **Memory:** The amount and speed of the memory, including RAM (random access memory) and cache memory, can impact how quickly data can be accessed and processed by the computer.
3. **Storage:** The speed and capacity of the storage devices, including hard drives and solid-state drives (SSDs), can impact the speed at which data can be stored and retrieved.
4. **I/O devices:** The speed and efficiency of input/output devices, such as keyboards, mice, and displays, can impact the overall performance of the system.
5. **Software optimization:** The efficiency of the software running on the system, including operating systems and applications, can impact how quickly tasks can be completed.

JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

Factor for improving Performance

- Reducing Time Complexity.
- Upgrading Hardware Components
- Optimizing Software
- Addressing Bottlenecks in the system.

Software: software as a collection of programs, data, and documentation that performs a set of tasks on a computer system.

Operating Systems: The book explains that an operating system is a program that manages the hardware and software resources of a computer system. It provides a user interface, manages memory and processes, and controls input/output devices. various types of operating systems, including batch, time-sharing, and real-time operating systems.

Compilers: compiler is a software program that translates source code written in a high-level programming language into machine language that can be executed by a computer. process of compiling, including lexical analysis, syntax analysis, semantic analysis, code generation, and optimization.

Assemblers: assembler is a software program that translates assembly language code into machine language. The process of assembly, including lexical analysis, syntax analysis, and code generation.

Interpreters: interpreter is a software program that executes source code written in a high-level programming language without first translating it into machine language. process of interpreting, including lexical analysis, syntax analysis, and execution.

Software Development: the process of software development, including requirements gathering, design, coding, testing, and maintenance. It explains various software development models, including the waterfall model, the spiral model, and the agile model.

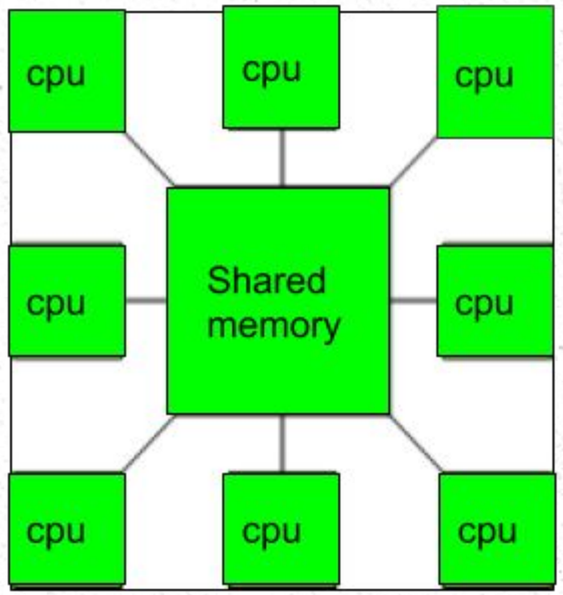
JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

Multi Processor

- **1. Multiprocessor:**
  A Multiprocessor is a computer system with two or more central processing units (CPUs) share full access to a common RAM.
- The main objective of using a multiprocessor is to boost the system's execution speed, with other objectives being fault tolerance and application matching.
- There are two types of multiprocessors, one is called shared memory multiprocessor and another is distributed memory multiprocessor.
- In shared memory multiprocessors, all the CPUs shares the common memory but in a distributed memory multiprocessor, every CPU has its own private memory.

JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

Multi Processor



JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

Applications of Multi Processor

1. As a uniprocessor, such as single instruction, single data stream (SISD).
2. As a multiprocessor, such as single instruction, multiple data stream (SIMD), which is usually used for vector processing.
3. Multiple series of instructions in a single perspective, such as multiple instruction, single data stream (MISD), which is used for describing hyper-threading or pipelined processors.
4. Inside a single system for executing multiple, individual series of instructions in multiple perspectives, such as multiple instruction, multiple data stream (MIMD).

JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

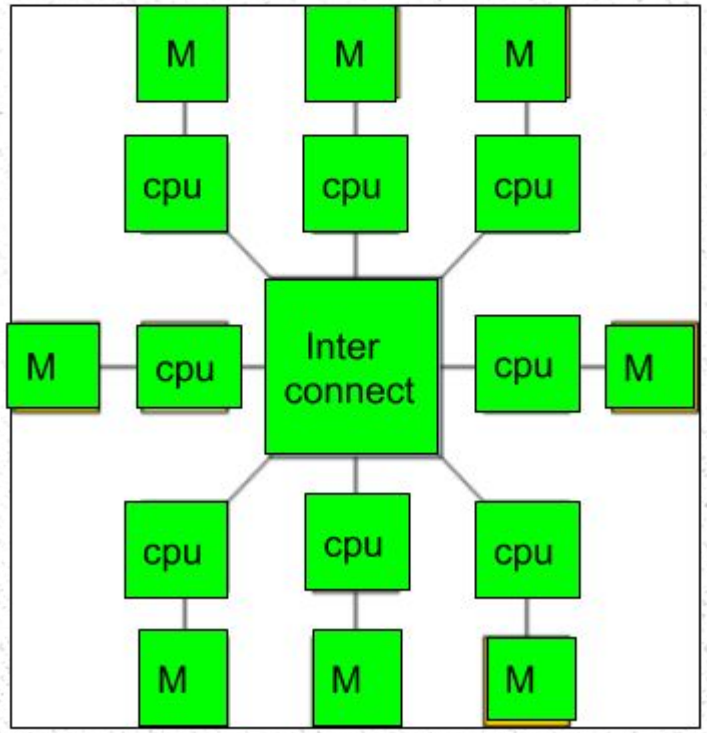
Multi Computer

- A multicomputer system is a computer system with multiple processors that are connected together to solve a problem.
- Each processor has its own memory and it is accessible by that particular processor and those processors can communicate with each other via an interconnection network.
- As the multicomputer is capable of messages passing between the processors, it is possible to divide the task between the processors to complete the task.
- Hence, a multicomputer can be used for distributed computing.
- It is cost effective and easier to build a multicomputer than a multiprocessor.

JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

Multicomputer

- Multicomputer



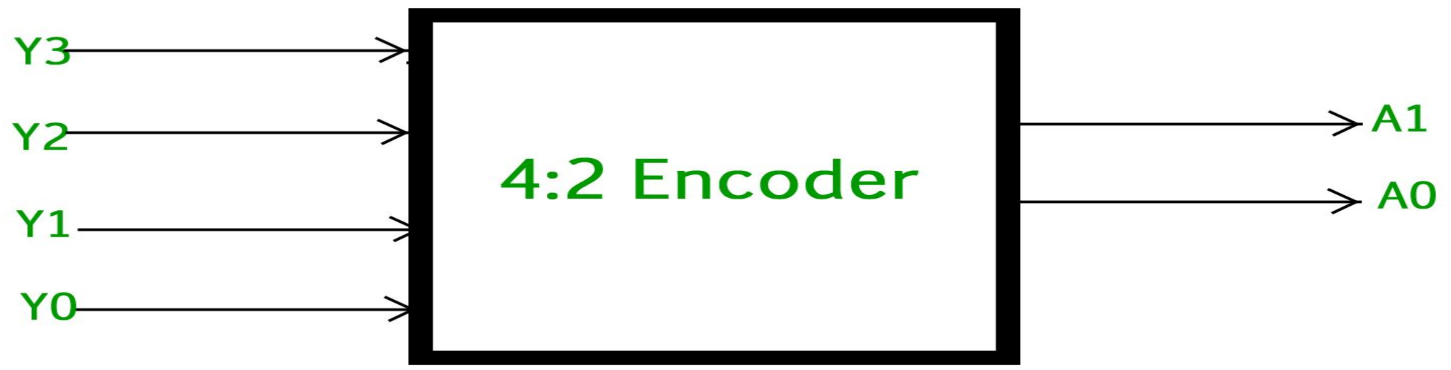
Encoder

- An encoder is a digital circuit that converts a set of binary inputs into a unique binary code.
- The binary code represents the position of the input and is used to identify the specific input that is active.
- Encoders are commonly used in digital systems to convert a parallel set of inputs into a serial code.
- An Encoder is a **combinational circuit** that performs the reverse operation of Decoder.
- It has maximum of **2^n input lines** and **'n' output lines**, hence it encodes the information from 2^n inputs into an n-bit code. It will produce a binary code equivalent to the input, which is active High.
- Therefore, the encoder encodes 2^n input lines with 'n' bits.

JGI JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

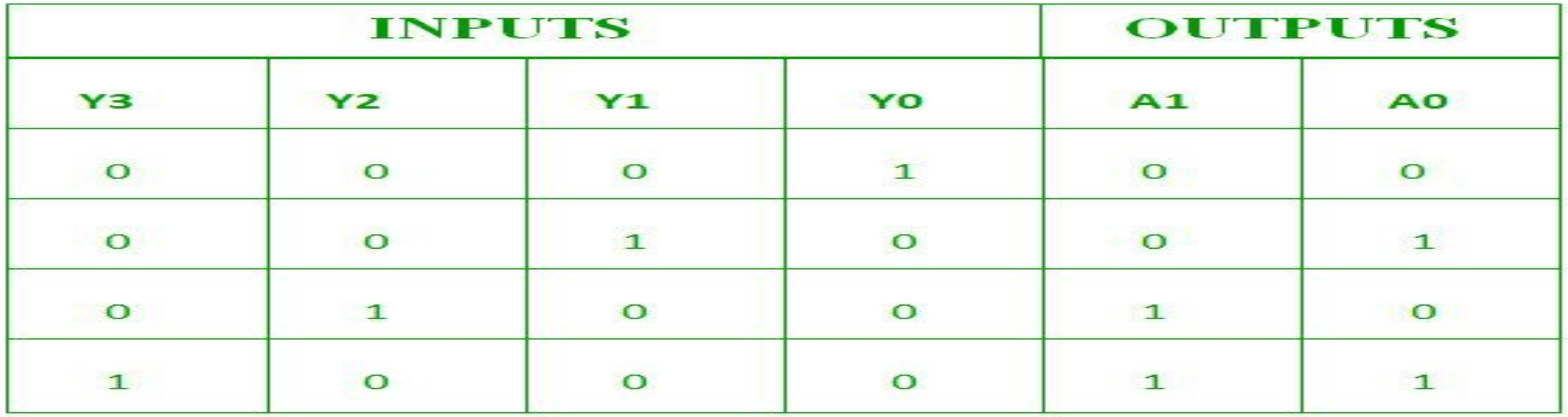
4:2 Encoder

- **4 : 2 Encoder**
- The 4 to 2 Encoder consists of **four inputs Y3, Y2, Y1 & Y0 and two outputs A1 & A0**. At any time, only one of these 4 inputs can be '1' in order to get the respective binary code at the output. The figure below shows the logic symbol of 4 to 2 encoder :



Truth Table

- 4:2 Encoder

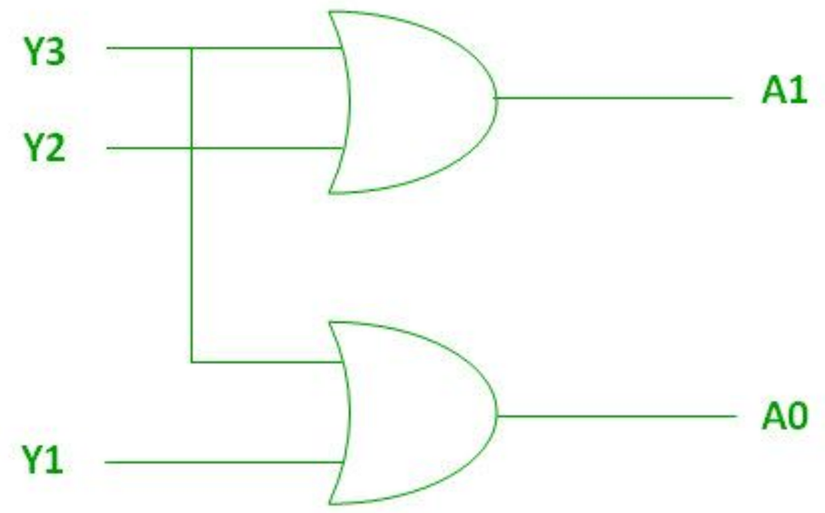


INPUTS				OUTPUTS	
Y3	Y2	Y1	Y0	A1	A0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

Implementation

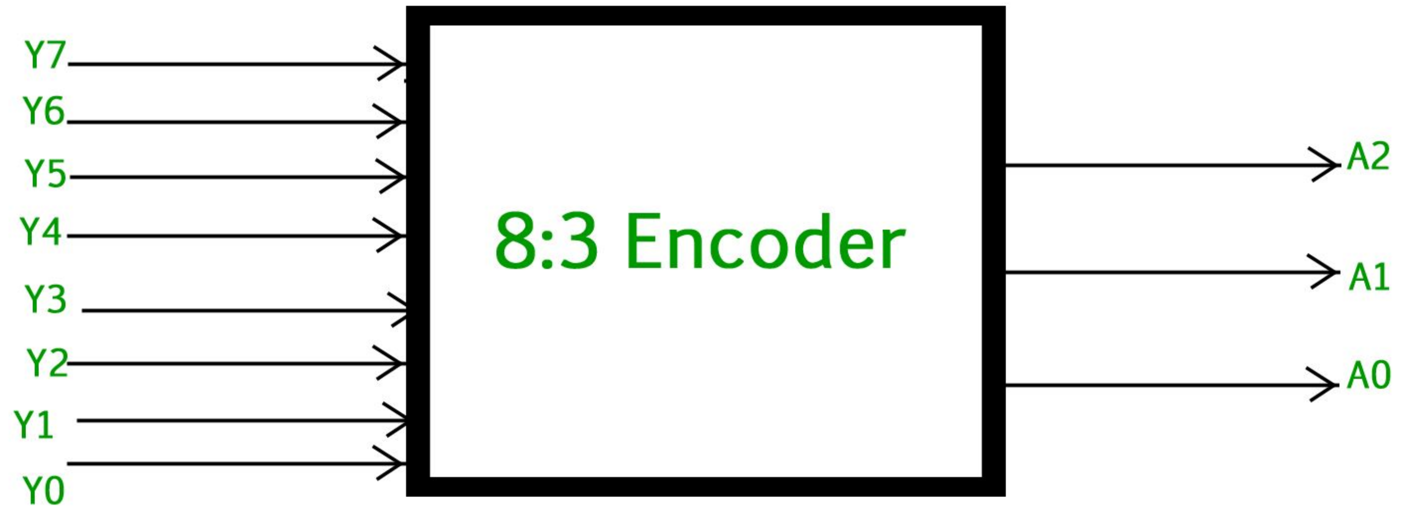
- Logical Expression for A0 and A1
- A1=Y3+Y2
- A0=Y3+Y1



JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

8 : 3 Encoder (Octal to Binary)

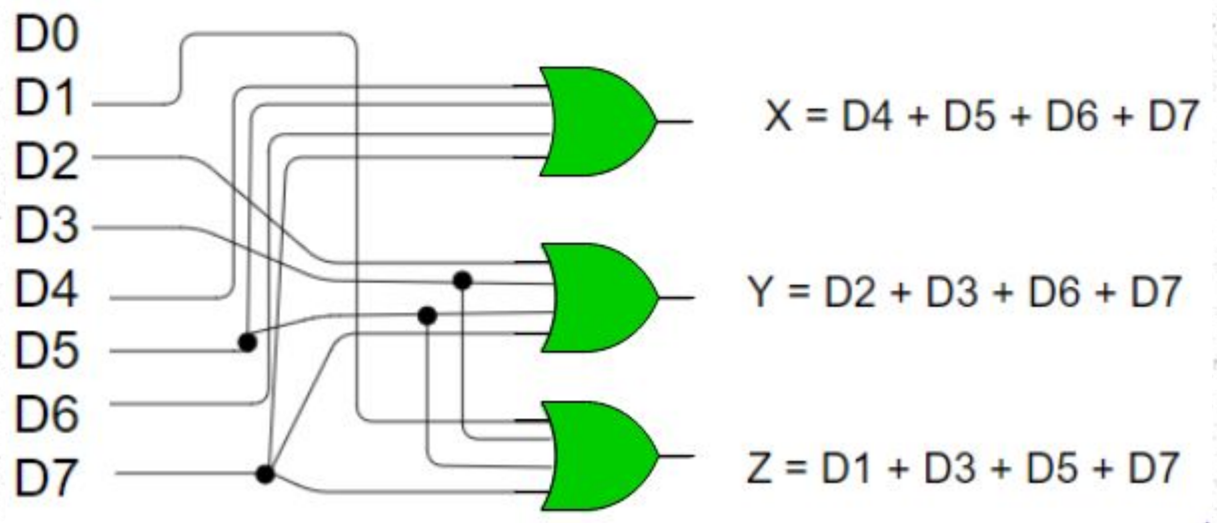
- **8 : 3 Encoder (Octal to Binary)**
- The 8 to 3 Encoder or octal to Binary encoder consists of **8 inputs** : Y7 to Y0 and **3 outputs** : A2, A1 & A0. Each input line corresponds to each octal digit and three outputs generate corresponding binary code. The figure below shows the logic symbol of octal to binary encoder:



TRUTH TABLE

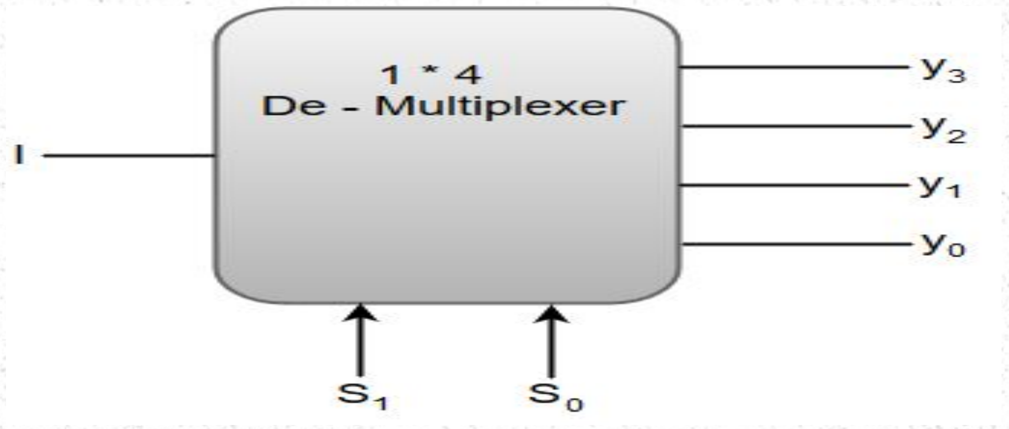
D7	D6	D5	D4	D3	D2	D1	D0	X	Y	Z
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

Implementation



De-Multiplexers

- De-Multiplexers
- A De-multiplexer (De-Mux) can be described as a combinational circuit that performs the reverse operation of a Multiplexer.
- A De-multiplexer has a single input, 'n' selection lines and a maximum of 2^n outputs.
- The following image shows the block diagram of a 1 * 4 De-multiplexer.



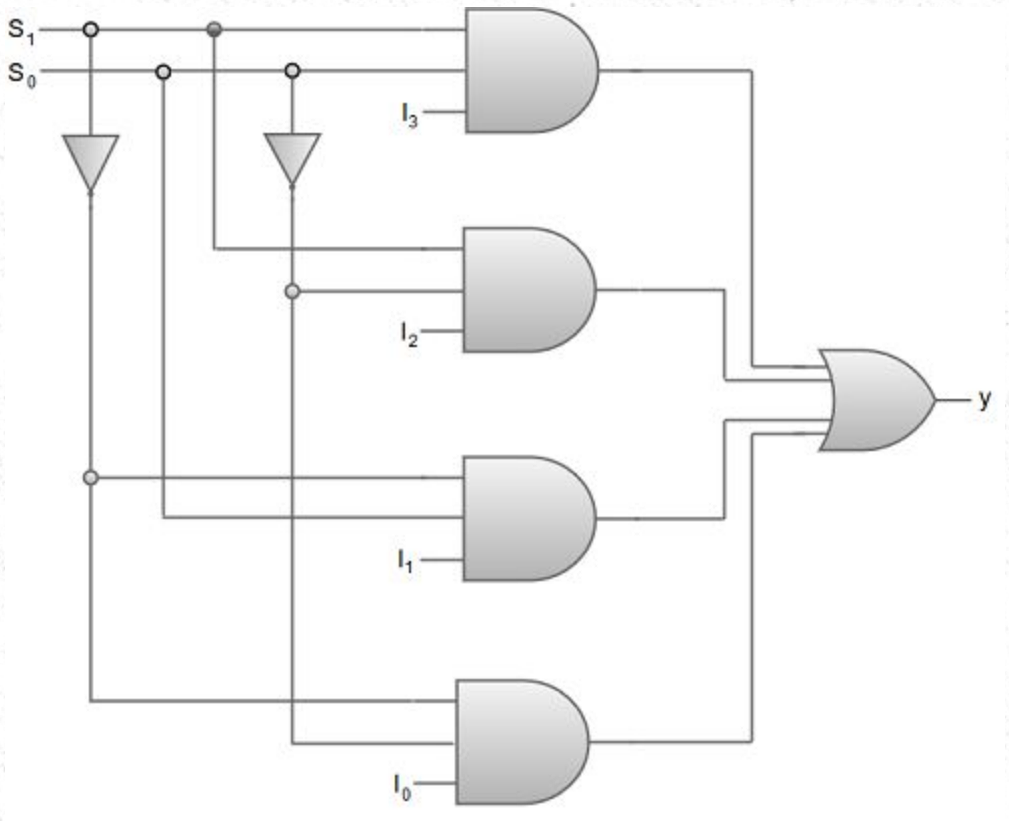
JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

Function Table 4*1

S1	S0	y
0	0	I0
0	1	I1
1	0	I2
1	1	13

JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

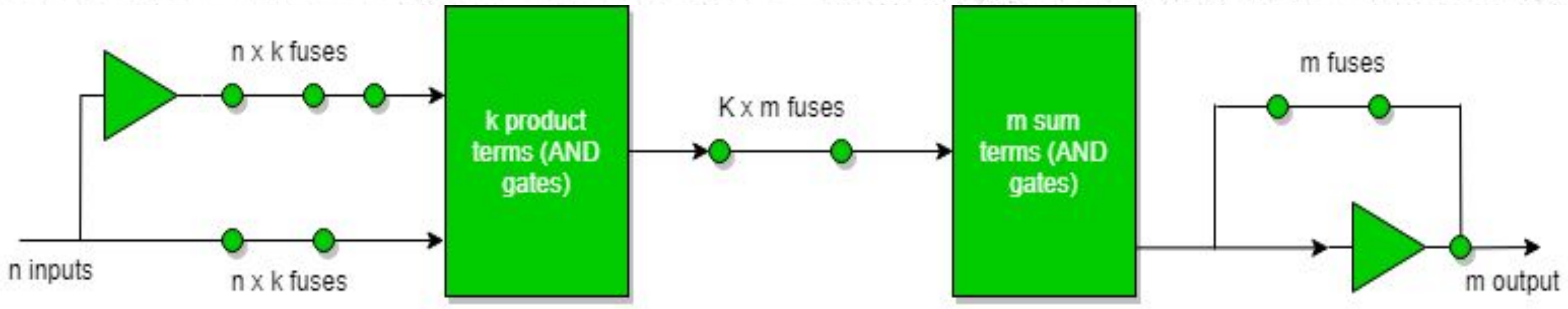
Implementation of 3AND and 1 OR



$y = S_1'S_0'I_0 + S_1'S_0I_1 + S_1S_0'I_2 + S_1S_0I_3$

Programmable Logic Array(PLA)

- Programmable Logic Array(PLA) is a fixed architecture logic device with programmable AND gates followed by programmable OR gates.
- **Basic block diagram for PLA:**



JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

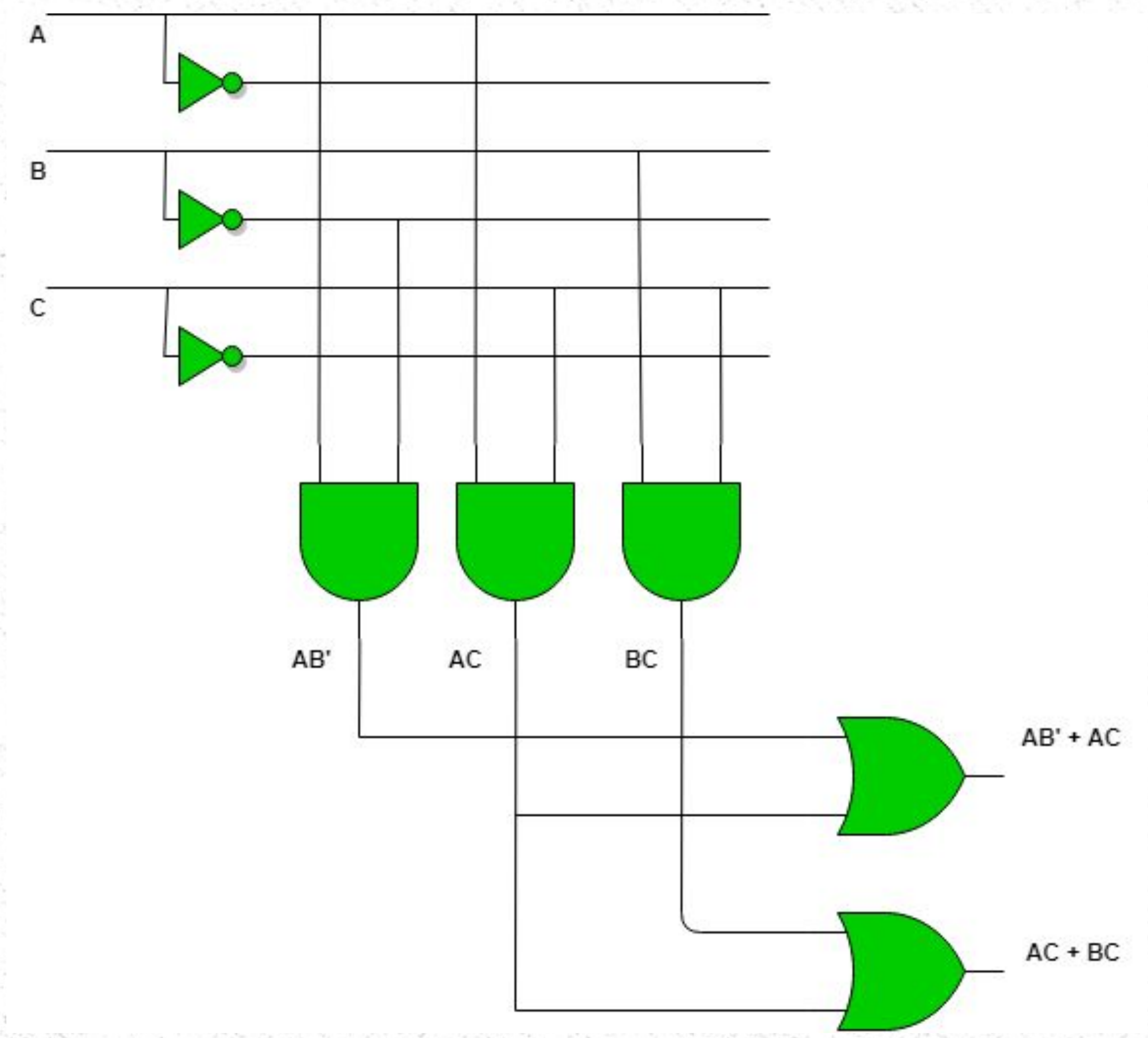
Truth Table

- F1 = AB'C' + ABC' + ABC
  on simplifying we get : F1 = AB + AC'
- F2 = A'BC + AB'C + ABC
  on simplifying we get: F2 = BC + AC

A	B	C	F1	F2
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1

JGI JAIN DEEMED-TO-BE UNIVERSITY
Powered By Futurense

Circuit Diagram



Digital Logic Circuits

- Digital logic circuits are the basis of digital systems. These logic circuits are a set of logic gates that show logical equivalence between two different groups of binary numbers.
- These digital logic circuits use 0 and 1 for on/off conditions, where 0 represents on, and 1 represents off conditions.
- Digital Logic Circuits are digital devices that use logic gates, ALU’s, microprocessors, RAM, ROM to control other circuits. It is a specific form of logic circuit that processes the numerical values 0 and 1.

Logic Circuits

- Digital circuits are also called logical circuits because they perform logical operations on digital signals. Digital circuits use logic gates like AND, OR, NOT, NAND, and NOR to perform the required digital operations.
- A digital circuit is a circuit containing digital logic. Digital circuits are the most common physical implementation of **Boolean algebra** and binary arithmetic and are the basis of all modern computers.
- It is because digital circuits are mainly used to process data that has only two values, such as true or false.
- In other words, it can be said that a digital circuit's primary function is to process the information that manages the binary system. Digital circuits are called logical circuits because they perform logical operations and produce results that can be interpreted as True or False.

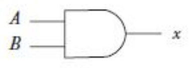
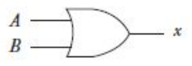
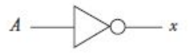
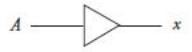
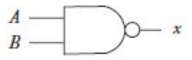
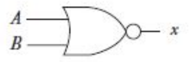
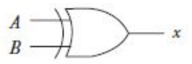
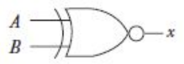
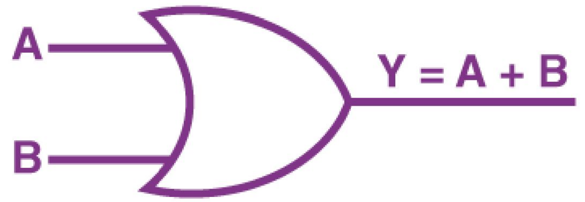
Name	*Graphic symbol*	*Algebraic function*	*Truth table*	*Input sensitivity*
AND		$x = A \cdot B$ or $x = AB$	A B | x 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1	0
OR		$x = A + B$	A B | x 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1	1
Inverter		$x = A'$	A | x 0 | 1 1 | 0	Not Applicable
Buffer		$x = A$	A | x 0 | 0 1 | 1	Not Applicable
NAND		$x = (AB)'$	A B | x 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0	0
NOR		$x = (A + B)'$	A B | x 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0	1
Exclusive-OR (XOR)		$x = A \oplus B$ or $x = A'B + AB'$	A B | x 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0	Not Applicable
Exclusive-NOR or equivalence		$x = (A \oplus B)'$ or $x = A'B + AB'$	A B | x 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1	Not Applicable

Figure 1-2 Digital logic gates with applicable input sensitivity values.

Basics of Logic Gates

- Logic gates are used to carry out logical operations on single or multiple binary inputs and give one binary output. In simple terms, logic gates are the electronic circuits in a digital system.
- **Types of Basic Logic Gates**
- There are several basic logic gates used in performing operations in digital systems. The common ones are
- OR Gate
- AND Gate
- NOT Gate
- XOR Gate
- Additionally, these gates can also be found in a combination of one or two. Therefore, we get other gates, such as NAND Gate, NOR Gate, EXOR Gate and EXNOR Gate.

OR GATE

- OR Gate
- In an OR gate, the output of an OR gate attains state 1 if one or more inputs attain state 1.
- 

The Boolean expression of the OR gate is Y = A + B, read as Y equals A ‘OR’ B.
The truth table of a two-input OR basic gate is given as

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

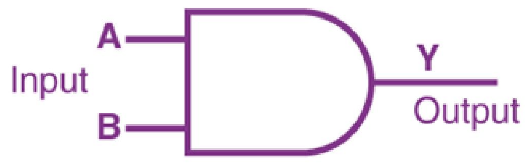
AND GATE

- AND Gate

- In the AND gate, the output of an AND gate attains state 1 if and only if all the inputs are in state 1. The Boolean expression of AND gate is Y = A.B

- The truth table of a two-input AND basic gate is given as

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

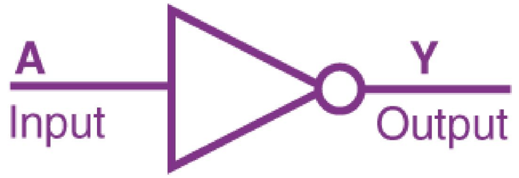


NOT GATE

- NOT Gate
- In a NOT gate, the output of a NOT gate attains state 1 if and only if the input does not attain state 1.
- The Boolean expression is

Y=A

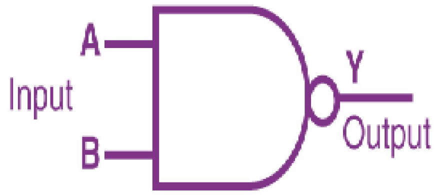
- It is read as Y equals NOT A.
- The truth table of NOT gate is as follows



A	Y
0	1
1	0

NAND GATE

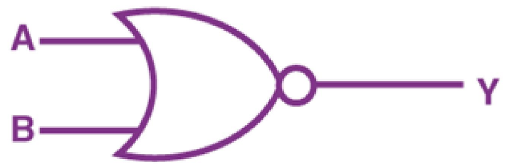
- NAND Gate
- This basic logic gate is the combination of AND and NOT gates.
- The Boolean expression of the NAND gate is
- Y=A.B
- The truth table of a NAND gate is given as



A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

NOR GATE

- NOR Gate
- This gate is the combination of OR and NOT gates.
- The Boolean expression of the NOR gate is
- Y=A+B
- The truth table of a NOR gate is as follows

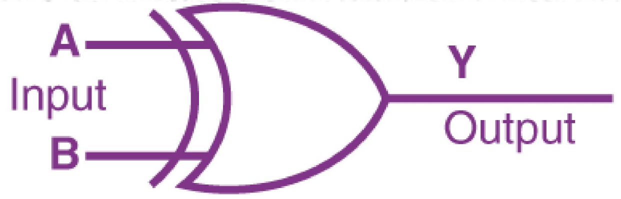


A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

JGi JAIN DEEMED-TO-BE UNIVERSITY
Powered By Futurense

Exclusive-OR gate (XOR Gate)

- Exclusive-OR gate (XOR Gate)
- In an XOR gate, the output of a two-input XOR gate attains state 1 if one adds only input and attains state 1.
- The Boolean Expression for XOR gate is



- A.B+A.B or Y=A + B

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Synthesis of digital circuits

- *Synthesis* is the process of generating a circuit that realizes a functional behavior of a logic system from a given description (stated in form of verbal statements, truth table, K-map, state diagram, etc.)
- Example: Synthesize a logic function that realizes the following truth table. Use AND, OR, and NOT gates

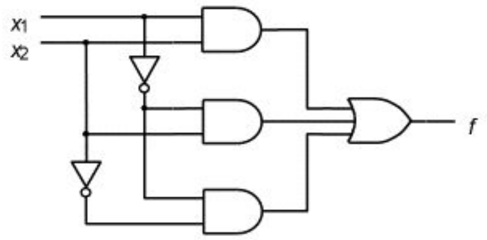
x1	x2	F(x1,x2)
0	0	1
0	1	1
1	0	0
1	1	1

JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

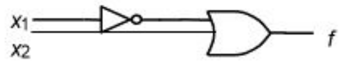
Synthesis of Digital Circuits

- Synthesis of digital circuits

  - Two implementations of a function



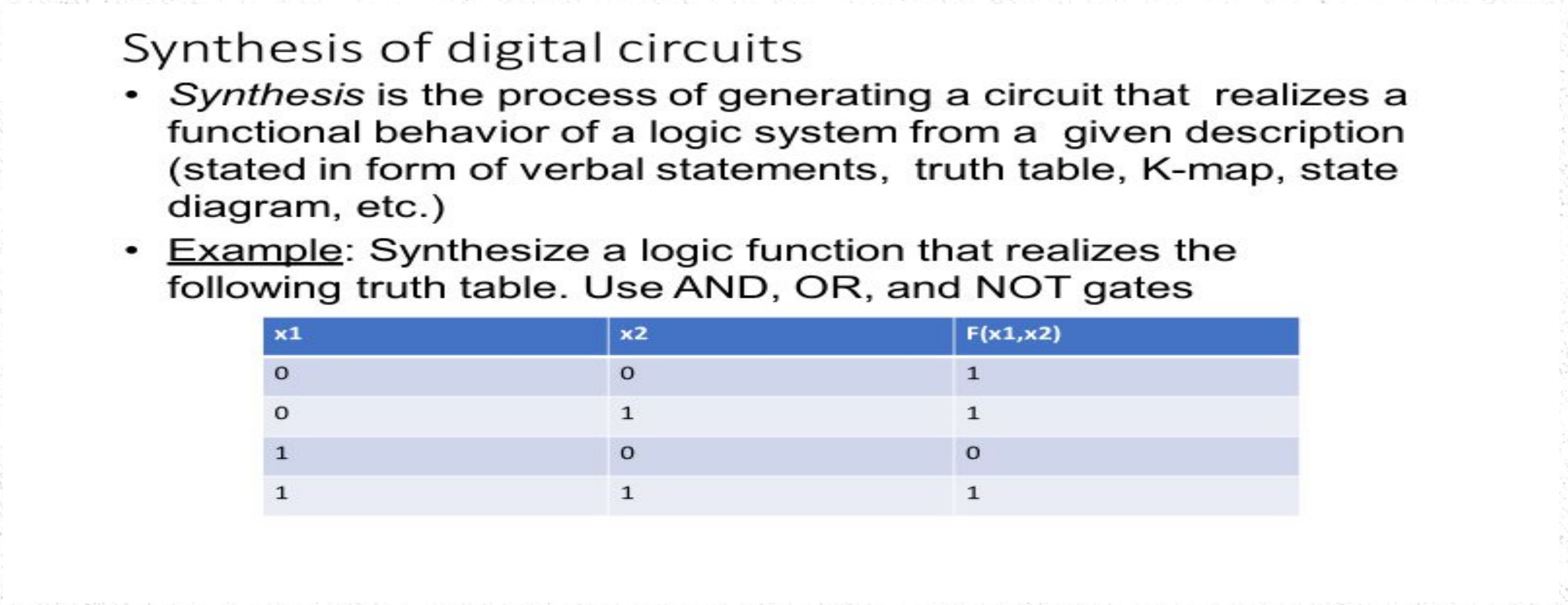
(a) Canonical sum-of-products



  - Minimal Cost Realization

-

Synthesis of Digital Circuits



Synthesis of digital circuits

- *Synthesis* is the process of generating a circuit that realizes a functional behavior of a logic system from a given description (stated in form of verbal statements, truth table, K-map, state diagram, etc.)
- Example: Synthesize a logic function that realizes the following truth table. Use AND, OR, and NOT gates

x1	x2	F(x1,x2)
0	0	1
0	1	1
1	0	0
1	1	1

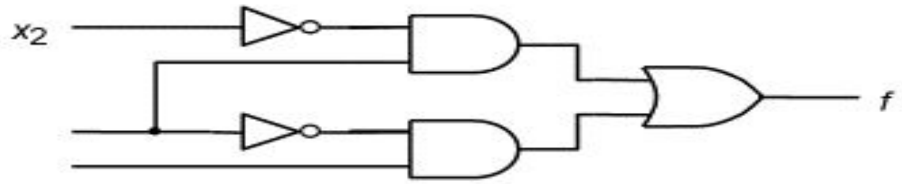
Synthesis of Digital Circuits

Row Number	X1 x2 x3	F(x1,x2,x3)
0	0 0 0	0
1	0 0 1	1
2	0 1 0	0
3	0 1 1	0
4	1 0 0	1
5	1 0 1	1
6	1 1 0	1
7	1 1 1	0

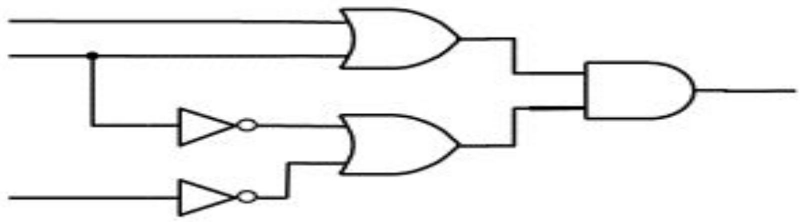
Synthesis of Digital Circuits

Synthesis of digital circuits

- Minimal Sum of Product Realisation



- Minimal product of sum Realisation



JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

Minimization of Logic EXPRESSIONS

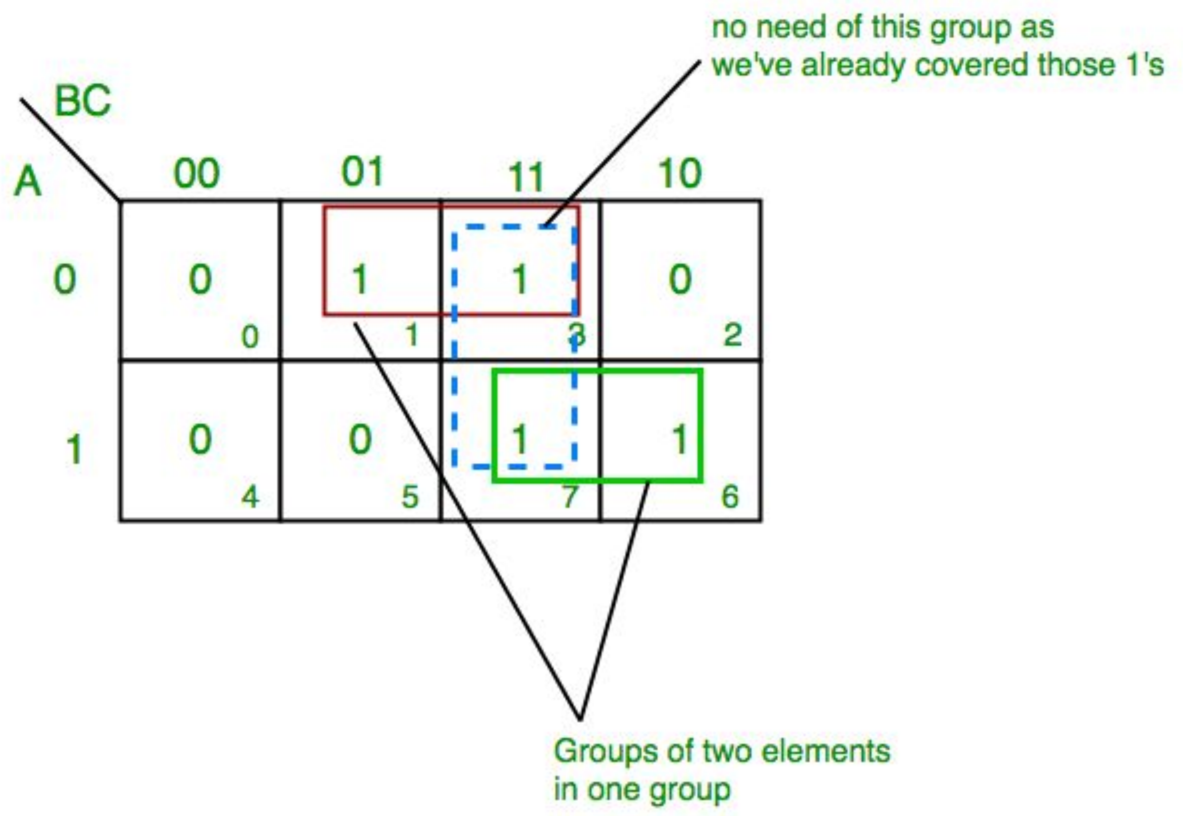
- K-map can take two forms Sum of Product (SOP) and Product of Sum (POS) according to the need of problem. K-map is table like representation but it gives more information than TRUTH TABLE. We fill grid of K-map with 0’s and 1’s then solve it by making groups.
- **Steps to solve expression using K-map-**

1. Select K-map according to the number of variables.
2. Identify minterms or maxterms as given in problem.
3. For SOP put 1’s in blocks of K-map respective to the minterms (0’s elsewhere).
4. For POS put 0’s in blocks of K-map respective to the maxterms(1’s elsewhere).
5. Make rectangular groups containing total terms in power of two like 2,4,8 ..(except 1) and try to cover as many elements as you can in one group.
6. From the groups made in step 5 find the product terms and sum them up for SOP form.

JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

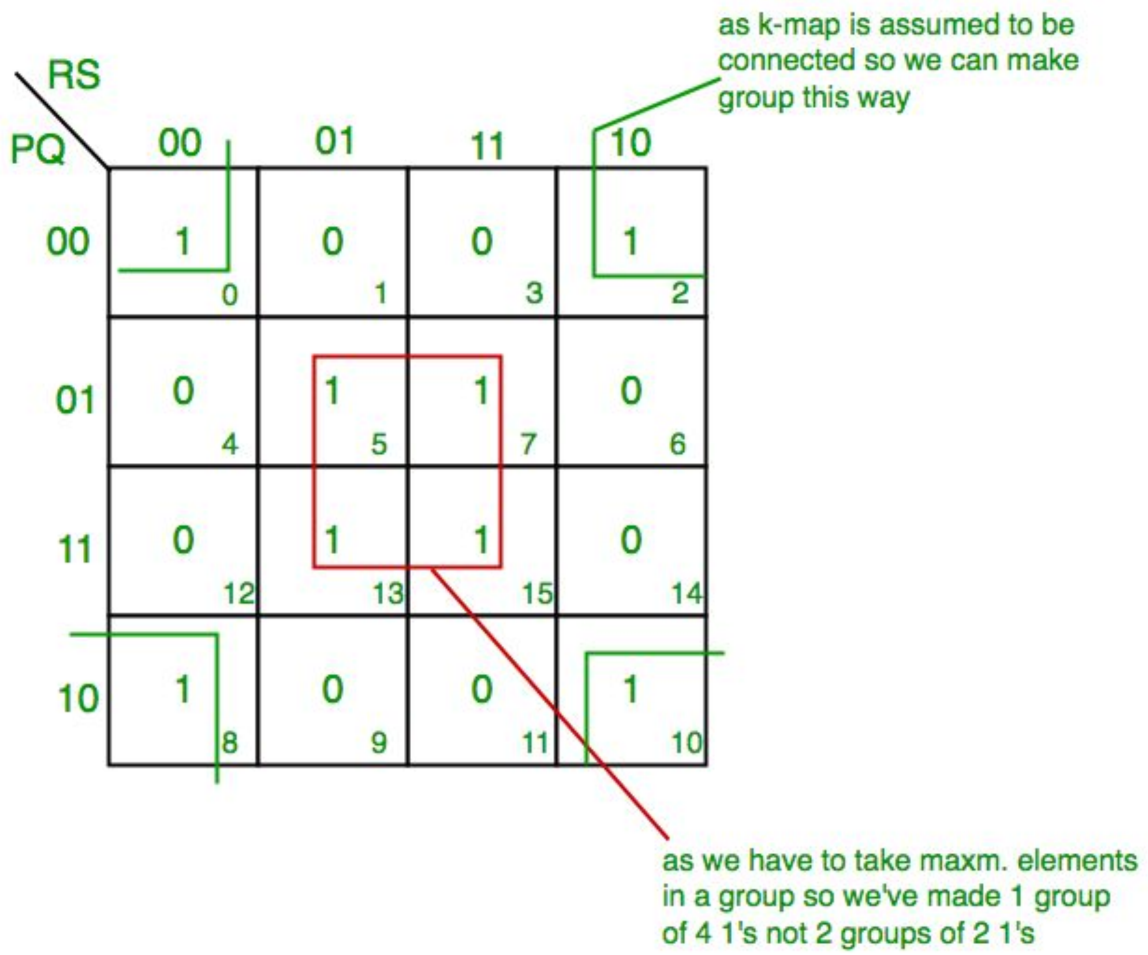
SOP Form K-Map for 3 variable

Z= ∑A,B,C(1,3,6,7)


no need of this group as we've already covered those 1's
BC
A
00
01
11
10
0
1
0 0
1 1
1 3
0 2
0 4
0 5
1 7
1 6
Groups of two elements in one group

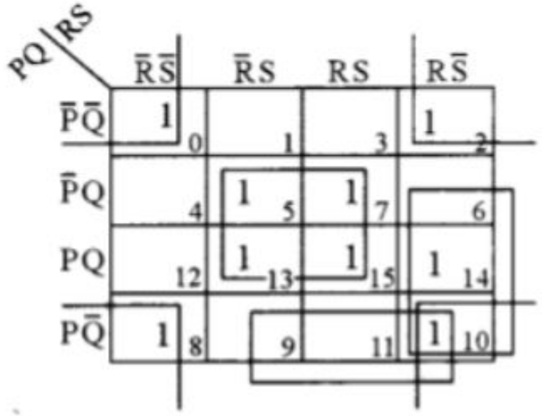
SOP Form K-Map for 4 Variable

F(P,Q,R,S)=∑ (0,2,5,7,8,10,13,15)



RS
PQ
00
01
11
10
as k-map is assumed to be connected so we can make group this way
as we have to take maxm. elements in a group so we've made 1 group of 4 1's not 2 groups of 2 1's

	00	01	11	10
00	1 (0)	0 (1)	0 (3)	1 (2)
01	0 (4)	1 (5)	1 (7)	0 (6)
11	0 (12)	1 (13)	1 (15)	0 (14)
10	1 (8)	0 (9)	0 (11)	1 (10)



k - Map

1st quad $\quad m_0 + m_2 + m_8 + m_{10}$

$= \bar{P}\,\bar{Q}\,\bar{R}\,\bar{S} + \bar{P}\,\bar{Q}\,R\bar{S} + P\,\bar{Q}\,\bar{R}\,\bar{S} + P\,\bar{Q}\,R\bar{S} = \bar{Q}\,\bar{S}$

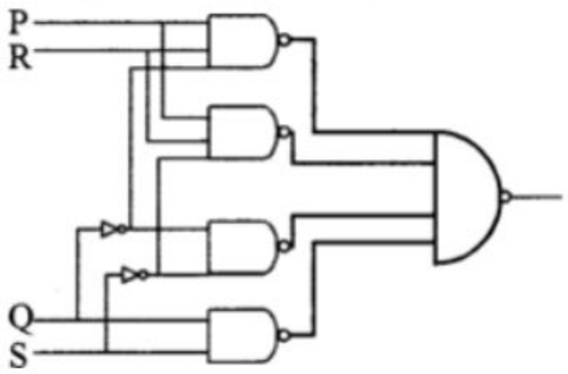
2nd quad $\quad m_5 + m_7 + m_{13} + m_{15}$

$= \bar{P}Q\,\bar{R}\,S + \bar{P}QRS + PQ\,\bar{R}\,S + PQRS$

1st pair $\quad = P\bar{Q}\,RS + P\,\bar{Q}\,R\,\bar{S} = P\bar{Q}\,R$

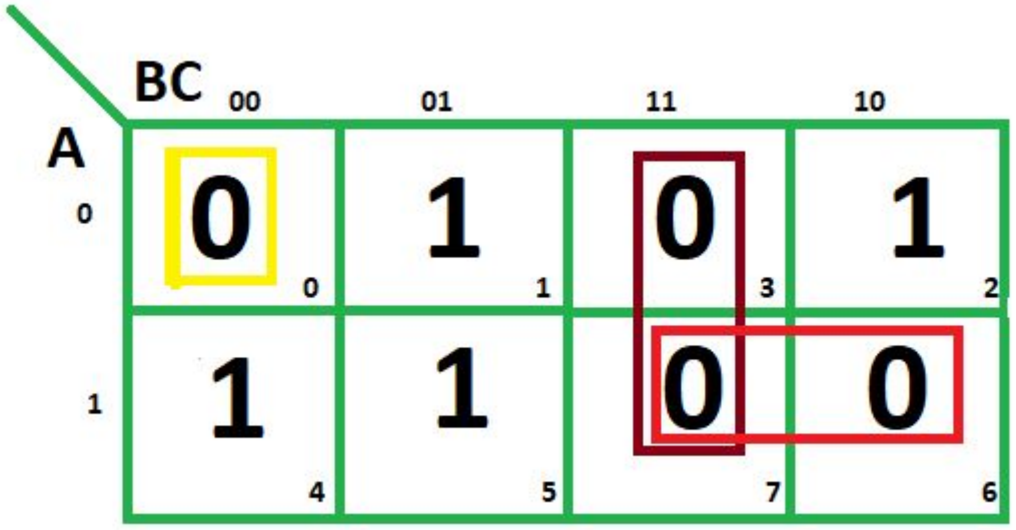
2nd pair $\quad = PQR\,\bar{S} + P\,\bar{Q}\,R\,\bar{S} = PR\,\bar{S}$

Reduced Expression $= \bar{Q}\;\bar{S} + QS + P\,\bar{Q}\,R + PR\,\bar{S}$



POS FORM K-Map for 3 variable

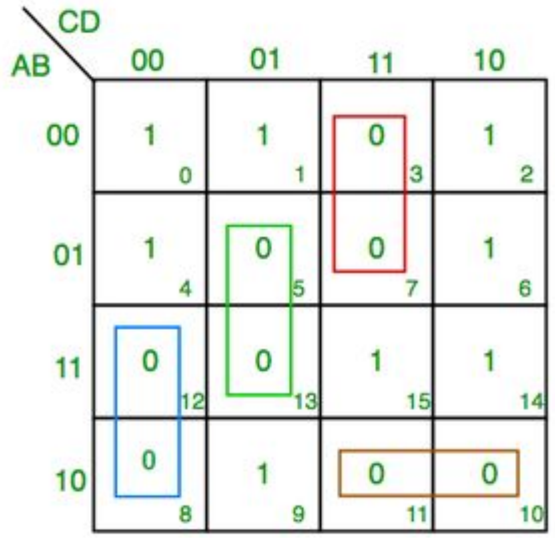
F(A,B,C)=π(0,3,6,7)



From red group we find terms
A B
Taking complement of these two
A' B'
Now sum up them
(A' + B')
From brown group we find terms
B C
Taking complement of these two terms
B’ C’
Now sum up them
(B’+C’)
From yellow group we find terms
A' B' C’
Taking complement of these two
A B C
Now sum up them
(A + B + C)
We will take product of these three terms :
Final expression – (A' + B’) (B’ + C’) (A + B + C)

POS FORM K-Map for 4 variable

F(A,B,C,D)=π (3,5,7,8,10,11,12,13)



AB \ CD	00	01	11	10
00	1_{0}	1_{1}	0_{3}	1_{2}
01	1_{4}	0_{5}	0_{7}	1_{6}
11	0_{12}	0_{13}	1_{15}	1_{14}
10	0_{8}	1_{9}	0_{11}	0_{10}

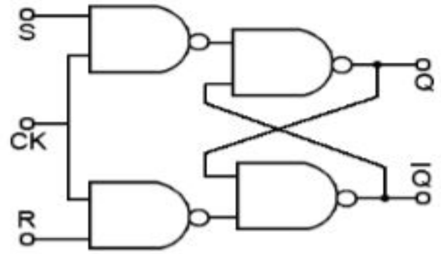
JGi JAIN DEEMED-TO-BE UNIVERSITY
Powered By Futurense

Flip Flop

- Flip-flop is a circuit that maintains a state until directed by input to change the state. A basic flip-flop can be constructed using four-NAND or four-NOR gates. **Types of flip-flops:**

1. SR Flip Flop
2. JK Flip Flop
3. D Flip Flop
4. T Flip Flop

- Logic diagrams and truth tables of the different types of flip-flops
- are as follows:
- **S-R Flip Flop :**
- Characteristics Equation for SR Flip Flop: $Q_{N+1} = Q_N R' + SR'$
-

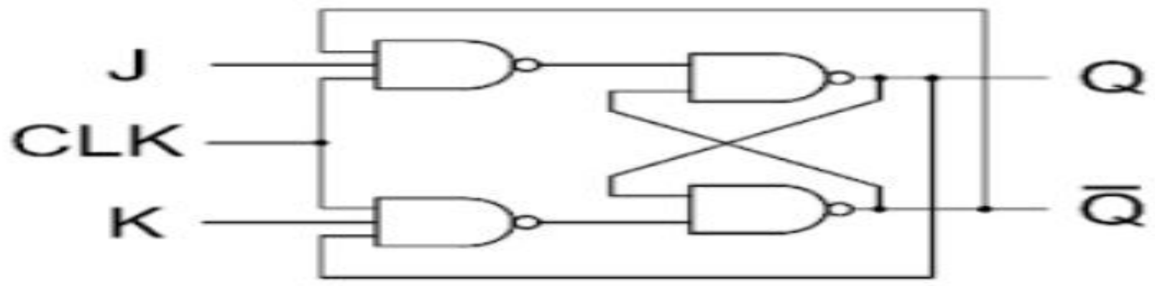


TRUTH TABLE

S	R	Q_N	Q_{N+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	-
1	1	1	-

JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

JK FLIP FLOP

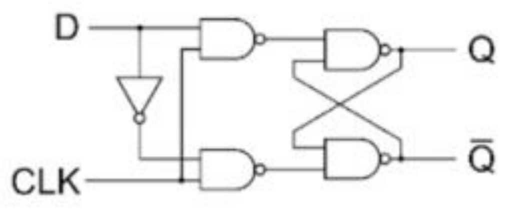


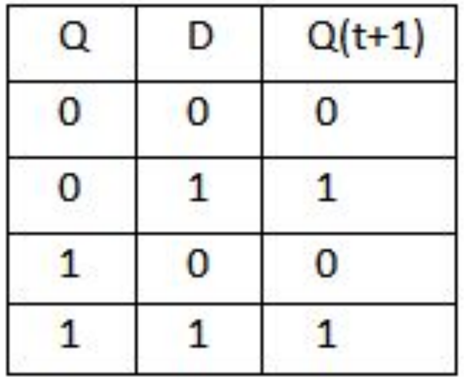
TRUTH TABLE

J	K	Q_N	Q_{N+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

JGi JAIN DEEMED-TO-BE UNIVERSITY
Powered By Futurense

D-Flip Flop

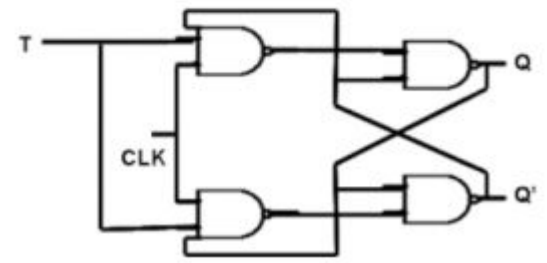


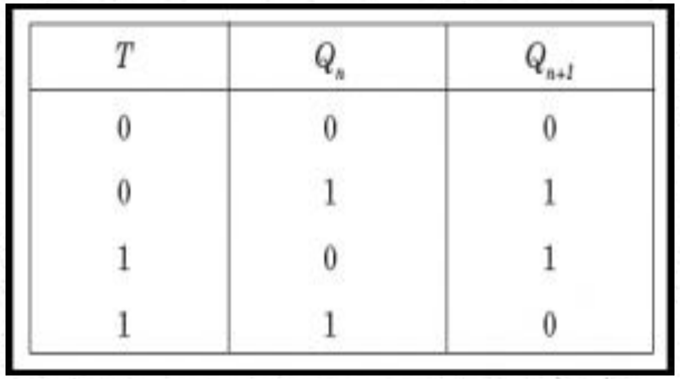


Q	D	Q(t+1)
0	0	0
0	1	1
1	0	0
1	1	1

Characteristics Equation for D Flip Flop: $Q_{N+1} = D$

T FLIP FLOP

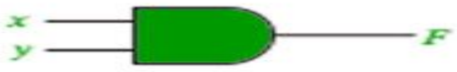
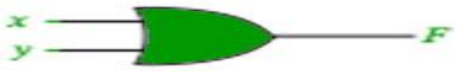
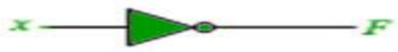
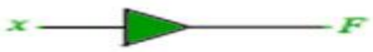
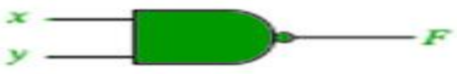
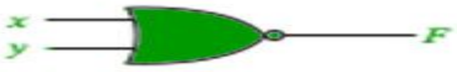
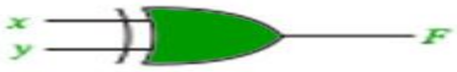
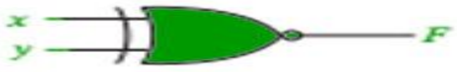




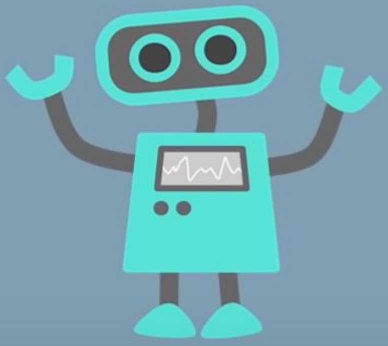
T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

JGi JAIN DEEMED-TO-BE UNIVERSITY Powered By Futurense

LOGIC GATES

Name	Graphic symbol	Algebraic function	Truth table
AND		$F = x \cdot y$	x y | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		$F = x + y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
Inverter		$F = x'$	x | F 0 | 1 1 | 0
Buffer		$F = x$	x | F 0 | 0 1 | 1
NAND		$F = (xy)'$	x y | F 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR		$F = (x + y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
Exclusive-OR (XOR)		$F = xy' + x'y$ $= x \oplus y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0
Exclusive-NOR or equivalence		$F = xy + x'y'$ $= (x \oplus y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1

Did You Know?



Registers are fast Computer Memory

An encoder converts a set of binary inputs into a unique binary code.

Summary

Outcomes:

a. Discuss the theory functionality and basic architecture of CPU
b. Discuss the Design Issues on the basis of speed, Technology, cost and performance.
c. Illustrate the different Logic Gates and Minimization of Logic gates.

Terminal Questions

1) What are the different types of Logic Gates?
2) What is Encoders?

Reference Links

- https://www.geekforgeeks.org/computer organization

Reference Material:

- William Stallings "Computer Organization and Architecture”,6Th Edition,Pearson/PHIISBN:10:0-13-609704-9

Thank you

JAIN DEEMED-TO-BE UNIVERSITY

Powered By Futurense