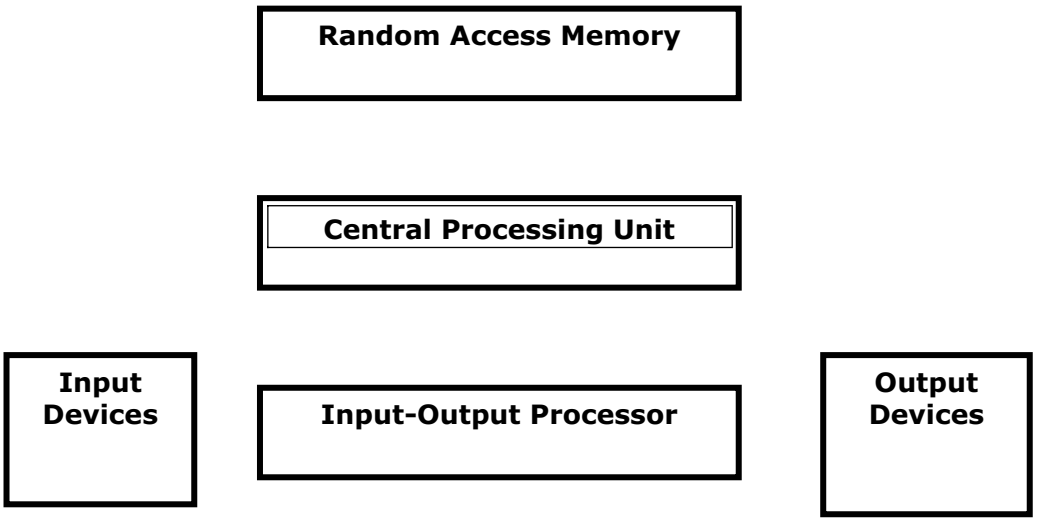
UNIT 1:
Digital Logic Circuits:

---

1. WHAT IS DIGITAL COMPUTER? OR EXPLAIN THE BLOCK DIAGRAM OF DIGITAL COMPUTERS.

- Digital computer is a digital system that performs various computational tasks.
- The word "DIGITAL" implies that the information in the computer is represented by digits.
- Digital computers use the binary number system, which has two digits, 0 and 1.
- A binary digit is called a ***bit*****.**
- Information is represented in digital computers in the groups of bit.
- By using various coding techniques, groups of bits can be made to represent not only binary numbers but also other symbols, letters of alphabets and decimal digit.
- Bits are grouped together as ***bytes*** and ***words*** to form some type of representation within the computer.
- A sequence of instructions for the computer is known as *program*.

Block diagram of a digital computer



- The hardware of the computer is usually divided into **three** major parts.
- **Central processing Unit (CPU)** :
  - The **Central processing Unit (CPU)** contains an arithmetic and logic unit for manipulating data, a number of registers for storing data and control circuits for fetching and executing instructions.
- **Random Access Memory**
  - The memory of a computer contains storage for instructions and data, it is called a **Random Access Memory (RAM)** because the CPU can access any location in memory at random and retrieve the binary information within a fixed interval of time.

- **input and output processor**
  - The **input and output processor** contains electronic circuit for communication and controlling the transfer of information between the computer and the outside world.
  - The **input and output device** connected to the computer include keyboards, printers, terminals, magnetic disk drives and other communication devices.

2. WHAT IS GATES? EXPLAIN THE LOGIC GATES IN BRIEF.

- Binary information is represented in digital computers using electrical signals.
- These signals can be represented by voltage to specify one of two possible states.
- The two states represent a binary variable that can be equal to **1 or 0.**
- The manipulation of binary information in a computer is done using logic circuits called ***gates*****.**
- Gates are blocks of hardware that produce signals of binary 1 or 0 when input logic requirements are satisfied.
- There are various types of logic gates are commonly used in digital computer.
- Each gate has a different graphic symbols and operation.
- The input-output relationship of binary variables for each gate can be represented in tabular form by Truth-Table.
- There are three types of gates:
  - **Basic / Fundamental Gates (AND, OR, NOT)**
  - **Universal Gates (NAND, NOR)**
  - **Exclusive Gates (EX-OR, EX-NOR)**

LOGICAL GATES

Basic / Fundamental Gates	**Universal Gates**	**Exclusive Gates**
(AND, OR, NOT)	**(NAND, NOR)**	**(EX-OR, EX-NOR)**

Basic Gates

AND Gate:

- In this type of gate output is high only when all its inputs are high.
- If any single input is law then the output will remain low.
- So it is said that in AND gate the output is only high when the input is also high.

SYMBOL:

F

TRUTH-TABLE:

INPUT		OUTPUT
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

FUNCTION:

F=AB OR F = A*B OR F=A.B

OR Gate:

- In this type of gate if any input signal is high then the output will be high.
- The output is only low only when all the inputs are low

SYMBOL:

F

TRUTH-TABLE:

INPUT		OUTPUT
A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

FUNCTION:

F=A + B

NOT Gate:

- This type of gate is also known as "Inverter".
- It is a gate that contains only one input and only one output.
- The output is always opposite than the input signals.

SYMBOL:

TRUTH-TABLE:

INPUT	OUTPUT
A	NOT A (A')
0	1
1	0

FUNCTION:

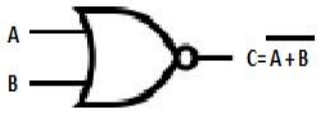
F= A' OR F= $\overline{A}$

Universal Gates

NOR Gate:

- The NOR gate is the complement of the OR gate.
- As shown in the truth table that the output of NOR gate is exactly opposite than the output of OR gate.
- This means that the output will be high when all the input is low.

SYMBOL:



F

TRUTH-TABLE:

INPUT		OUTPUT
A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

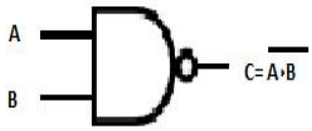
FUNCTION:

F= (A + B)' OR F= $\overline{(A+B)}$

NAND Gate:

- The NAND gate is an AND gate followed by NOT gate.
- As shown in the truth table that the output of NAND gate is exactly opposite than the output of AND gate.
- This means that the output will be high when all the input is high.

SYMBOL:



F

TRUTH-TABLE:

INPUT		OUTPUT
A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

FUNCTION:

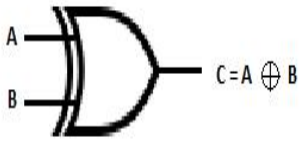
F= (A .B)' OR F= $\overline{(A .B)}$

Exclusive Gates

EX-OR Gate:

- This gate is produces high output whenever the two inputs are at opposite level.
- The EX-OR gate is the gate that produces high output for Odd number of high inputs.

SYMBOL:



F

TRUTH-TABLE:

INPUT	OUTPUT

A	B	A EX-OR B
0	0	0
0	1	1
1	0	1
1	1	0

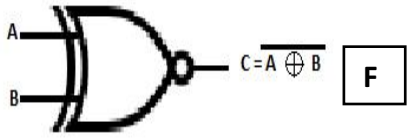
FUNCTION:

F= (A + B)' OR F= $\overline{(A+B)}$

EX-NOR Gate:

- This gate is produces high output whenever the two inputs are at same level.
- The EX-OR gate is the gate that produces high output for Even number of high inputs.
- The truth table shows that output of this gate is exactly opposite of EX-OR gate.

SYMBOL:



TRUTH-TABLE:

INPUT		OUTPUT
A	B	A EX-NOR B
0	0	1
0	1	0
1	0	0
1	1	1

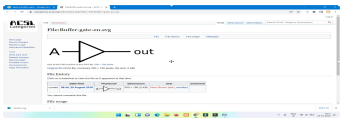
FUNCTION:

$$F=\overline{A \oplus B}$$

Buffer Gates

Buffer Gate:

- A buffer is a non-inverting amplifier that has an output drive capacity that is far greater than its input drive requirement.
- it will produce same output of its input.
- If input is 0 then output is 0 & if input is 1 then output is 1.
- **SYMBOL:**



TRUTH-TABLE:

INPUT	OUTPUT
A	F
0	0
1	1

FUNCTION:

F= A OR F= A

3. WRITE A NOTE ON BOOLEAN ALGEBRA

- In 1854 George Boole introduced a systemic treatment of logic and developed for this purpose an algebraic system called Boolean Algebra.
- Boolean Algebra is an algebra that deals with binary variables and logic operations.
- The variables are designated by letters such as A,B, X ,Y etc.
- The three basic operations are AND, OR and complement.
- A Boolean function can be expressed with binary variable, the logic operation symbols, parentheses (rounded bracket) and equal to (=) sign.

•The result of a Boolean function is either 0 or 1.

•A Boolean function can be represented by either:

a) Truth tables
b) Logic diagrams
c) Algebraic expression

•For example: F=x+y'z

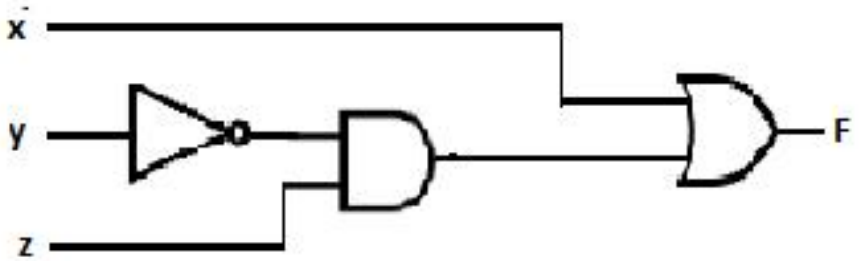
- ○F=1 only if x is 1 or if both y' and z=1.
- ○If y'(complement of y)=1 means that y=0 so we can say that F=1 only when x=1,y=0,z=1.
- ○So we can say that function F equal to 1 for those combination where x=1 or yz=01

•A Boolean function can be transformed form algebraic expression into a logic diagram composed of AND,OR and NOT gates.

•**Truth table and logic diagram For above example :**

x	y	z	y'	y'.z	x + y'z
0	0	0	1	0	0
0	0	1	1	1	1

0	1	0	0	0	0
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1	0	0	1



1	0	0	1	0	1
1	0	1	1	1	1
1	1	0	0	0	1
1	1	1	0	0	1

4. BASIC IDENTITIES OF BOOLEAN ALGEBRA .

Boolean Operations

- Boolean Algebra is an algebra, which deals with binary numbers & binary variables.
- Hence, it is also called as Binary Algebra or logical Algebra.
- A mathematician, named George Boole had developed this algebra in 1854.
- Basic Laws of Boolean Algebra are divided into three parts.
  - Commutative law
  - Associative law
  - Distributive law

There are three basic logical operations:

- AND: This operation is represented by a dot or by the absence of an operator. For example, x.y = z or xy = z is read "x AND y is equal to z." The logical operation AND is interpreted to mean that z=1 if and only if x=1 and y=1; otherwise z=0.
- OR: This operation is represented by a plus sign. For example, x+y = z is read "x OR y is equal to z", meaning that z=1 if x=1 or if y=1 or if both x=1 and y=1. If both x=0 and y=0, then z=0.
- NOT: This operation is represented by a prime (sometimes by a bar). For example, $x' = z$ (or $\bar{x} = z$) is read "x not is equal to z", meaning that z is complement of x. In other words, if x=1, then z=0, but if x=0, then z=1.

Commutative Law

- If any logical operation of two Boolean variables give the same result irrespective of the order of those two variables, then that logical operation is said to be Commutative.
- The logical OR & logical AND operations of two Boolean variables x & y are shown below
  - x + y = y + x
  - x.y = y.x

Associative Law

- If a logical operation of any two Boolean variables is performed first and then the same operation is performed with the remaining variable gives the same result, then that logical operation is said to be Associative.
- The logical OR & logical AND operations of three Boolean variables x, y & z are shown below.
  - x + (y + z) = (x + y) + z
  - x.(yz) = (x.y). z

Distributive Law

- If any logical operation can be distributed to all the terms present in the Boolean function, then that logical operation is said to be Distributive.
- The distribution of logical OR & logical AND operations of three Boolean variables x, y & z are shown below.
  - x. (y + z) = x . y + x .z
  - x + (y . z) = (x + y) . (x + z)

Theorems of Boolean Algebra

- The following two theorems are used in Boolean algebra.

1	X + 0 = X		10	(X + Y) = (Y + X)
2	X + 1 = 1		11	(X . Y) = (Y . X)
3	X + X = X		12	X + (Y + Z) = (X + Y) + Z
4	X + X' = 1		13	X . (YZ) = (XY) . Z
5	X . 0 = 0		14	X + YZ = (X + Y) (X + Z)
6	X . 1 = X		15	X .(Y + Z) = (XY + XZ)
7	X . X = X		16	X + XY =X
8	X . X' = 0		17	X(X + Y) = X
9	(X')' = X			

5. WRITE A NOTE ON DEMORGAN'S THEOREM

- It is developed greater mathematician and logician named De-Morgan.
- He developed two theorems which makes the complement of questions and product(**incomplete line).**
- It is very important in dealing with NOR and NAND gates.
- The 2 most important theorems of De-Morgan are as follows:

1) The complement of sum equal to the product of the complement.

$$\overline{x + y} = \overline{x} \cdot \overline{y}$$

2) The complement of product equal to the sum of the complement.

$$\overline{x \cdot y} = \overline{x} + \overline{y}$$

- We can prove the theorems with the help of truth table.

- **THEOREM 1:** $\overline{x + y} = \overline{x} \cdot \overline{y}$

x	y	x'	y'	(x+y)	(x+y)'	x' . y'
0	0	1	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	1	0	0	1	0	0

- Last 2 columns gives same output so LHS=RHS

THEOREM 2: $\overline{x \cdot y} = \overline{x} + \overline{y}$

x	y	x'	y'	(x.y)	(x.y)'	x' + y'
0	0	1	1	0	1	1
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	0	0

- Last 2 columns gives same output so LHS=RHS

6. WRITE A NOTE ON K- MAPS.

- The Karnaugh map is also referred as Veitch Diagrams, KV maps or K-maps.
- K-map is a method to minimizes the Boolean function.
- K-map provides a simple and straight forward method to minimizing Boolean expression.
- With the help of K-map we can simplified Boolean expression up to 4 and 6 variables.
- K-map diagram represents squares and each square represents 1 minterm.
- In K-map values of the variables are written in binary form & the logic function can be expressed in one of the following form
    - **SUM OF PRODUCTS (SOP)**
    - **PRODUCT OF SUM (POS)**

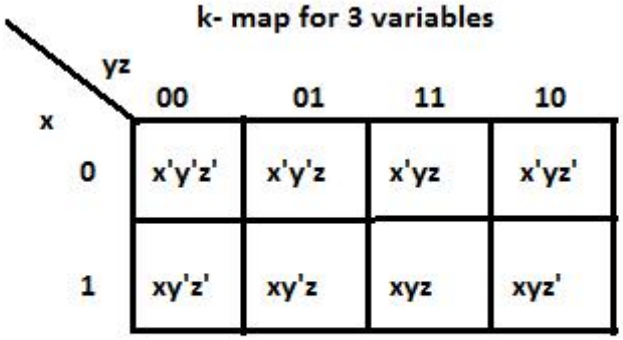
- A K-map for n variables is made up of $\mathbf{2^n}$ squares and each squares designed a product term of Boolean expression.
- For product terms which are present in expression, 1s are written in correspondence squares and 0 will be written in blank square.
- **For example: K-map for 2 variables:**

F =xy' + x'y

x \ y	0	1
0	x'y'	x'y
1	xy'	xy

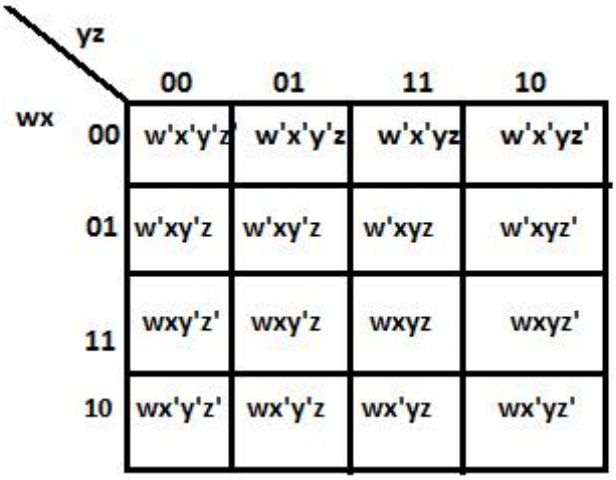
x \ y	0	1
0	0	1
1	1	0

k- map for 3 variables



x \ yz	00	01	11	10
0	x'y'z'	x'y'z	x'yz	x'yz'
1	xy'z'	xy'z	xyz	xyz'

k-map for 4 variables



wx \ yz	00	01	11	10
00	w'x'y'z'	w'x'y'z	w'x'yz	w'x'yz'
01	w'xy'z	w'xy'z	w'xyz	w'xyz'
11	wxy'z'	wxy'z	wxyz	wxyz'
10	wx'y'z'	wx'y'z	wx'yz	wx'yz'

RULES FOR K- MAP:

- Each cell with 1 must be included in at list 1 group.
- Try to form the largest possible groups.
- Try to end up with as few groups as possible.
- Groups may be in sizes that are powered of 2.
- Groups may be square or rectangular only.
- Groups may be horizontal or vertical but not diagonal.
- Groups may wrap around the table.
- Groups may overleap.
- The larger a group is, the more redundant inputs there are:
    - Group of 1 has no redundant input.
    - Group of 2 known as pair has 1 redundant input.
    - Group of 4 known as quad has 2 redundant input.
    - Group of 8 known as octet has 3 redundant input.

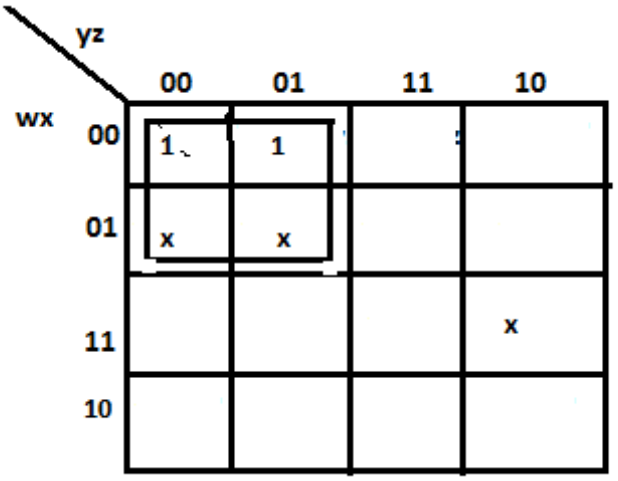
7. SHORT NOTE: DON'T CARE CONDITION.

- Don't care conditions is the condition where any single square or map will appears as x and it is not necessary to write it into Boolean expression.
- In k-map each cell represents a minterm or maxterm and the 0's and 1's in k map represents the minterm that make the function equal to either 0 or 1.

- But in some occasion, it doesn't matter whether a function produces a 0 or 1 for a given minterm. When this condition occurs, an *X* is used in the map to represent **the *don't care* condition.**
- The minterm that may produce either 0 or 1 for function are said to be Don't Care and marked as x in map.
- This don't care condition are used to further simplify the Boolean expression.
- Don't care condition is the condition where any single square or map will appear as x n it is not necessary to write into Boolean expression.

Example

F(w,x,y,z)= ∑(0,1) + d(4,5,14)



- So the answer is **W'Y'**

8. WHAT IS MINTERM?

- Minterm is referred as product term.
- minterm for each combination of the variables that produces a 1 in the function and then taking the OR of all those terms.
- In minterm variable may or may not be complemented.
- Minterm is represented by m symbol
- Minterm are product term that are multiplied with each other

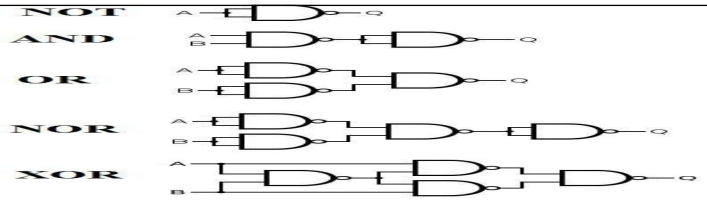
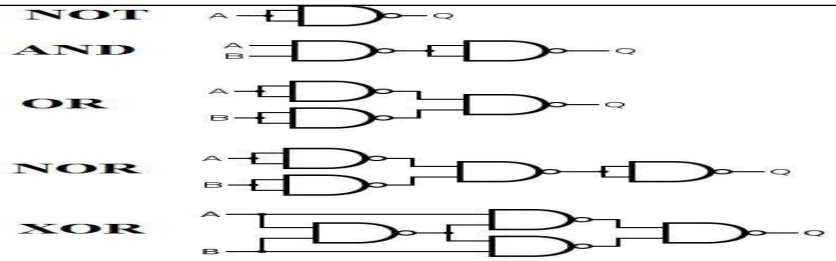
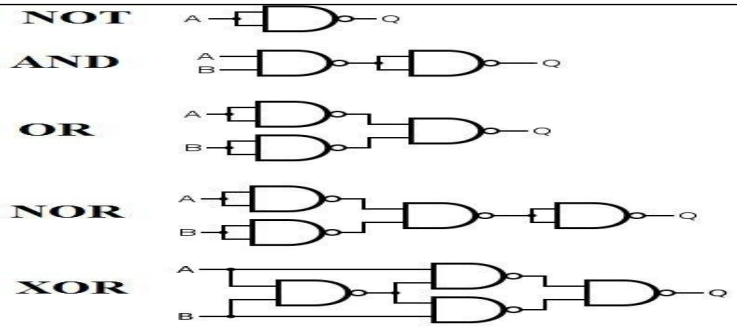
X	Y	Z	MINTERM	SYMBOL
0	0	0	X' . Y' . Z'	m0
0	0	1	X' . Y' . Z	m1
0	1	0	X' . Y . Z'	m2
0	1	1	X' . Y . Z	m3
1	0	0	X . Y' . Z'	m4
1	0	1	X . Y' . Z	m5
1	1	0	X . Y . Z'	m6
1	1	1	X . Y . Z	m7

9. WHAT IS MAXTERM?

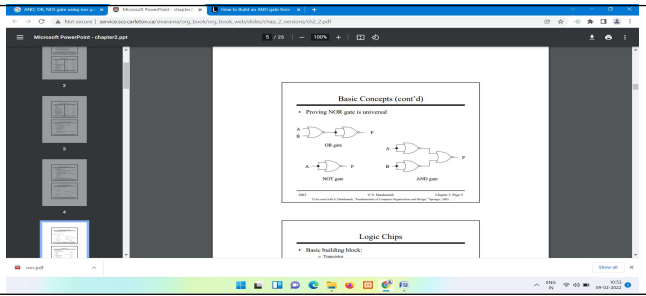
- Maxterm is referred as sum term.
- maxterm for each combination of the variables that produces a 0 in the function and then taking the AND of all those terms.
- In maxterm variable may or may not be complemented.
- Maxterm is represented by M symbol

X	Y	Z	MINTERM	SYMBOL
0	0	0	X +Y + Z	M0
0	0	1	X + Y + Z’	M1
0	1	0	X+ Y’+ Z	M2
0	1	1	X +Y’ +Z’	M3
1	0	0	X’ +Y + Z	M4
1	0	1	X’ + Y + Z’	M5
1	1	0	X’ +Y’ + Z	M6
1	1	1	X’ +Y’+ Z’	M7

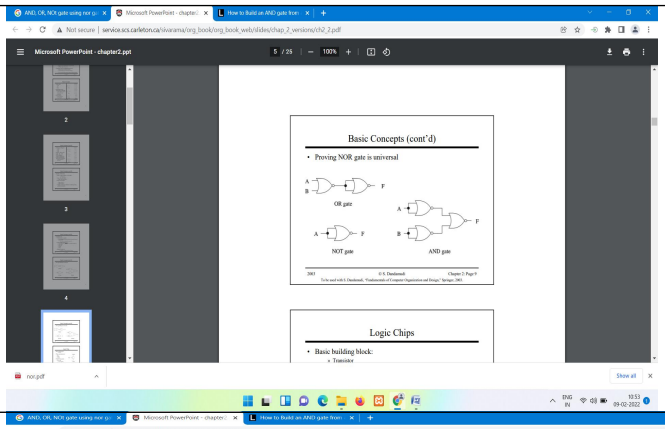
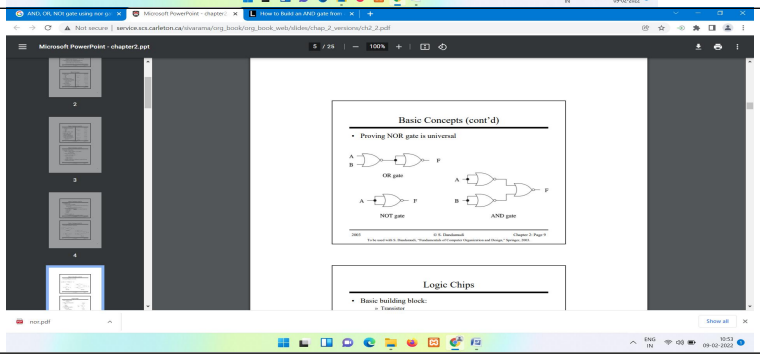
10. EXPLAIN NAND AND NOR GATE ARE AS AN UNIVERSAL GATES.

NOT gate Using NAND gate	
AND gate Using NAND gate	
OR gate Using NAND gate	

11. EXPLAIN NOR GATE ARE AS AN UNIVERSAL GATES.

NOT gate Using NOR gate	

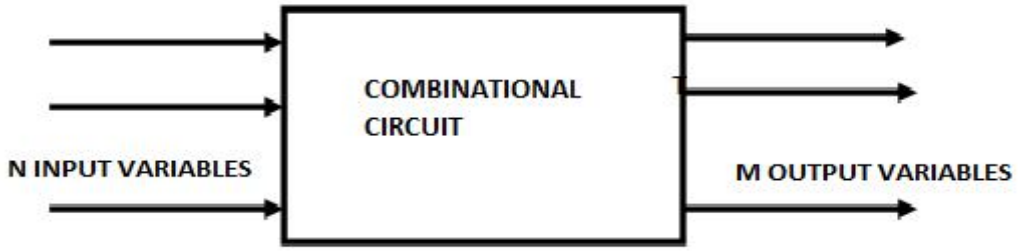
Prepared By: Prof. Meghna Bhatt

AND gate Using NOR gate	
OR gate Using NOR gate	

12. WRITE A NOTE ON COMBINATIONAL CIRCUITS

- A combinational circuit is the circuit where more than 1 circuit is designed into single component.
- It has N no of inputs and M no of outputs.
- It is basically used to design digital applications and it transforms the data into the digital manner.
- A combinational circuit is a connected arrangement of logic gates with a set of inputs and outputs.
- At any given time, the binary values of the outputs are a function of the binary values of the inputs.
- The design of a combinational circuit starts from a verbal outline of the problem and ends in a logic circuit diagram. The procedure involves the following steps:
    1. The problem is stated.
    2. The input and output variables are assigned letter symbols.
    3. The truth table that defines the relationship between inputs and outputs is derived.
    4. The simplified Boolean functions for each output are obtained.
    5. The logic diagram is drawn.

Block diagram of combinational circuit



Example of combinational circuits are:

- Arithmetic circuits
- Multiplexer
- Encoder
- Decoder
- Parity Checker
- Parity Generator

13. EXPLAIN HALF ADDER AND FULL ADDER IN DETAIL

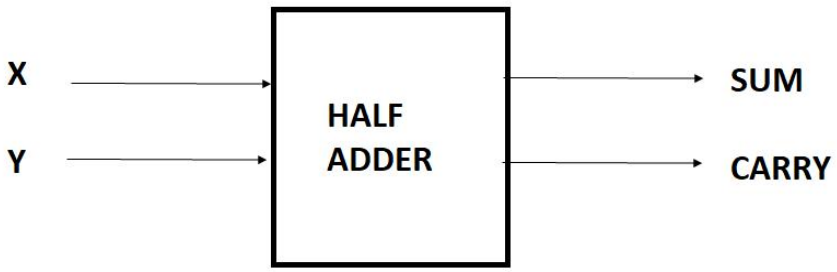
Arithmetic circuits:

- It is made of different arithmetic operators. There will be addition, subtraction, division, modules and any other arithmetic operations.

HALF ADDER

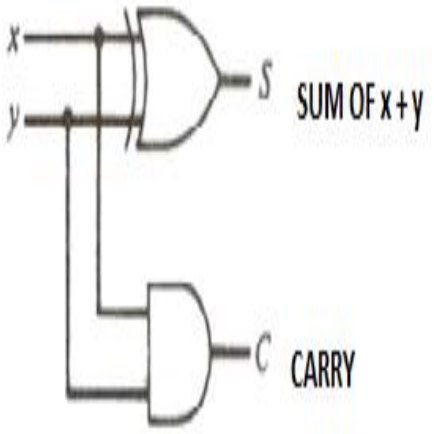
- Half-Adder is a part of combinational circuit.
- It is basically designed for arithmetic addition.
- It is most basic digital arithmetic circuit.
- Performs the addition of two binary digits.
- It contain 2 inputs and 2 outputs.
- The input variables of a half-adder are called the **augends** and the **addend.**
- The output variables of half-adder are **sum** and **carry.**
- This circuit is design with the help of EX-OR gate & AND gate.

Block diagram



Truth table

INPUT		OUTPUT	
X	Y	SUM X+Y	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



(b) Logic diagram

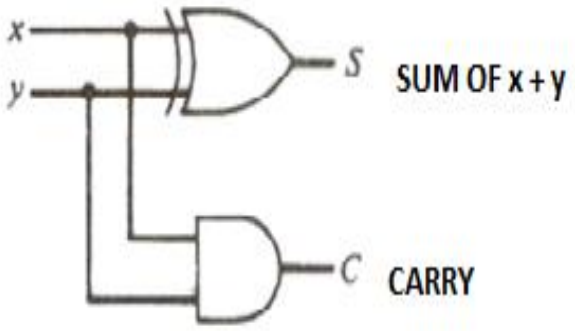
Half Adder Expression

Sum= X ⊕Y

Carry= XY

Half Adder Circuit

INPUT		OUTPUT	
X	Y	SUM X+Y	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



(b) Logic diagram

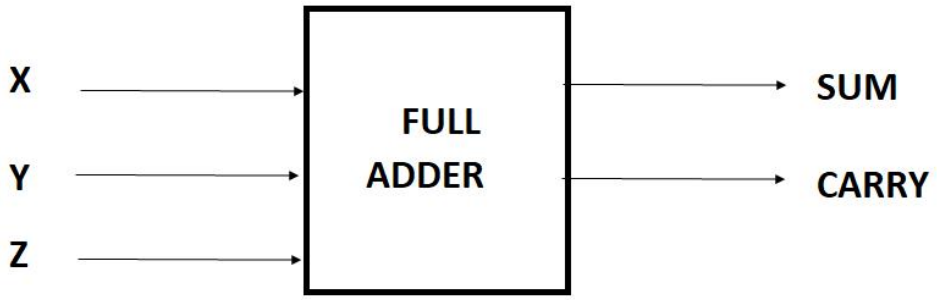
Full-Adder

- Half-Adder is a part of combinational circuit.
- It is basically designed for arithmetic addition.
- A full-adder performs the addition of three binary digits.
- Two half-adders can be combined to for a full-adder..

Prepared By: Prof. Meghna Bhatt

- Although a full adder has three inputs, it still only has two outputs since the largest number is 1+1+1 = 3, and 3 can be represented by two bits.
- It contain 3 inputs and 2 outputs.
- The input variables of a full-adder are called the **augends** and the **addend.**
- The output variables of full-adder are **sum** and **carry.**
- This circuit is design with the help of 2 EX-OR gate, 2 AND gate & 1 OR gate.

Block diagram

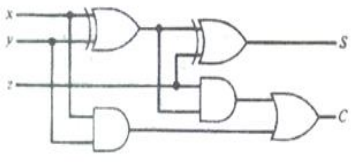


Truth table

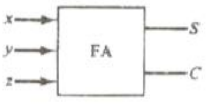
TRUTH TABLE OF FULL - ADDER

INPUT			OUTPUT	
X	Y	Z	SUM OF X+Y+Z	CARRY
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

LOGIC DIAGRAM OF FULL - ADDER



BLOCK DIAGRAM OF FULL ADDER



Full Adder Expression

Sum= X ⊕Y ⊕ Z

Carry=Z(X'Y + XY') +XY

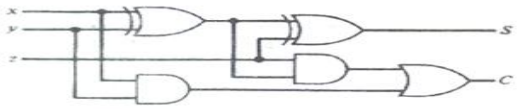
Prepared By: Prof. Meghna Bhatt

Full Adder Circuit

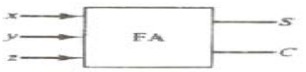
TRUTH TABLE OF FULL - ADDER

INPUT			OUTPUT	
X	Y	Z	SUM OF X+Y+Z	CARRY
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

LOGIC DIAGRAM OF FULL - ADDER



BLOCK DIAGRAM OF FULL ADDER



14. WRITE DOWN THE DIFFERENCE BETWEEN HALF ADDER AND FULL ADDER

Half adder	Full adder
• The most basic digital arithmetic circuit.	• A full-adder performs the addition of three binary digits.
• Performs the addition of two binary digits.	• It is used for multi bit additions.
• Output is sum of two signals.	• Output is sum of three signals.
• There are two input and two output terminal.	• There are three input and two output terminal.
• From full adder half adder cant not be built	• Two full adder makes one full adder
• On EX-OR gate and one AND gate are used.	• Two EX-OR, two AND and one OR gate is used.

15. WHAT IS FLIP-FLOP?

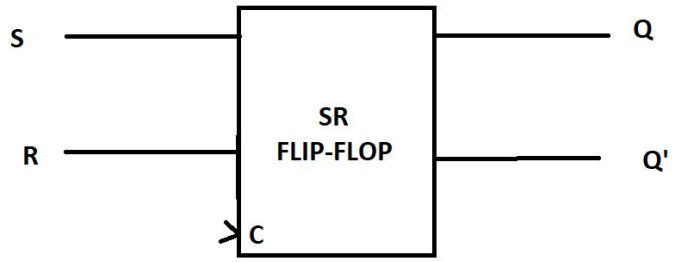
- A Flip-flop is a binary cell capable of storing one bit of information.
- It is called by stable device because it have 2 stable position either 0 or 1
- It has two outputs, one for the normal value and one for the complement value of the bit stored in it.
- Flip-flops are storage elements utilized in synchronous sequential circuits.
- Synchronous sequential circuits employ signals that effect storage elements only at discrete instances of time.
- A timing device called a clock pulse generator that produces a periodic train of clock pulses achieves synchronization.
- Values maintained in the storage elements can only change when the clock pulses.
- Hence, a flip-flop maintains a binary state until directed by a clock pulse to switch states.

- The difference in the types of flip flops is in the number of inputs and the manner in which the inputs affect the binary state.
- Flip-flops can be described by a characteristic table which permutates all possible inputs (just like a truth table).
- The characteristic table of a flip-flop describes all possible outputs (called the next state) at time Q(t+1) over all possible inputs and the present state at time Q(t).
- The most common types of flip flops are:
  - SR Flip-Flop
  - D Flip-Flop
  - JK Flip-Flop
  - T Flip-Flop

16. EXPLAIN SR FLIP-FLOP IN DETAILS.

- RS OR SR Flip-flop stands for RESET-SET or SET-RESET flip-flop.
- This flip-flop can be synchronous or asynchronous.
- It contain 2 inputs & if asynchronous then have 1 extra input called clock pulse.

Block diagram



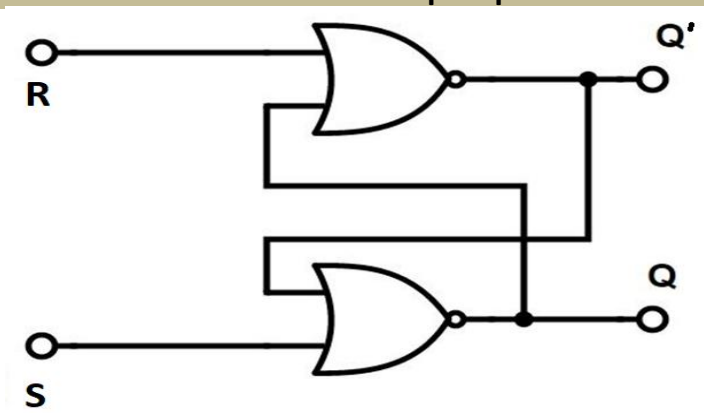
Truth Table

S	R	OUTPUT Q
0	0	No Change
0	1	Reset
1	0	Set
1	1	Invalid

- If there is no signal at the clock input C, the output of the circuit cannot change irrespective of the values at inputs S and R.
- Only when the clock signals changes from 0 to 1 can the output be affected according to the values in inputs S and R
- If S =1 and R = 0 when C changes when C changes from 0 to 1 output Q is set to 1. If S = 0 and R =1 when C changes from 0 to 1.

- If both S and R are 0 during the clock transition, output does not change.
- When both S and R are equal to 1, the output is unpredictable and may go to either 0 or 1, depending on internal timing that occur within the circuit.

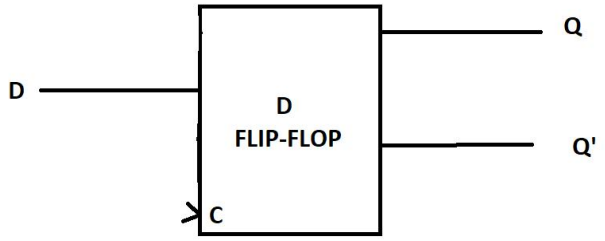
Circuit of SR flip-flop



17. EXPLAIN D FLIP-FLOP IN DETAILS.

- The D stands for Delay Flip-Flop.
- It is invented to remove the limitations of RS flip-flop.
- When R and S both received 1 Flip-flop goes to invalid stage; this limitation is removed by D flip-flop.

Block diagram



Truth Table

Clk	D	OUTPUT Q(t+1)
0	0	No Change
0	1	No Change
1	0	First Stage (0)
1	1	Next Stage(1)

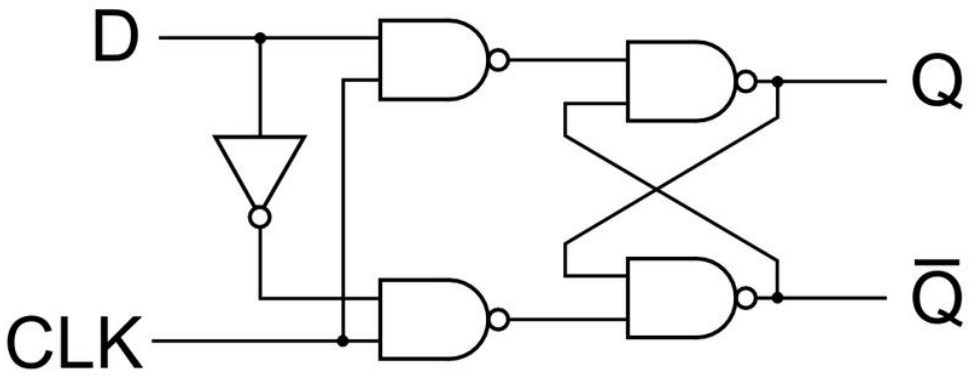
- If D=1, the output of the flip-flop goes to the 1 state, but if D=0, the output of the flip-flop goes to the 0 state.
- The next state Q(t+1) is determined from the D input. The relationship can be expressed by a characteristic equation:

$$Q(t+1) = D$$

- D Flip-Flop has the advantage of having only one input (excluding).

- The disadvantage that its characteristic table does not have a “no change” condition Q(t+1) = Q(t).

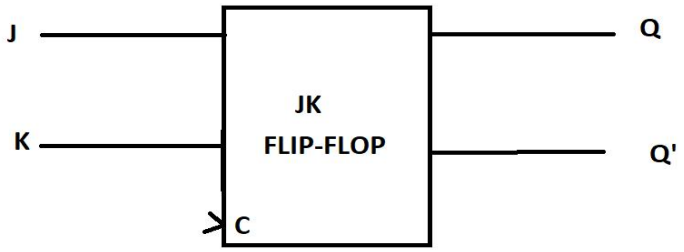
Circuit of D flip-flop



18. EXPLAIN JK FLIP-FLOP IN DETAILS.

- The JK stands for jump & kick Flip-Flop.
- It is similar to RS flip-flop but no invalid state.
- It’s a modified version of RS flip-flop where J is similar to R and K is similar to S.

Block diagram

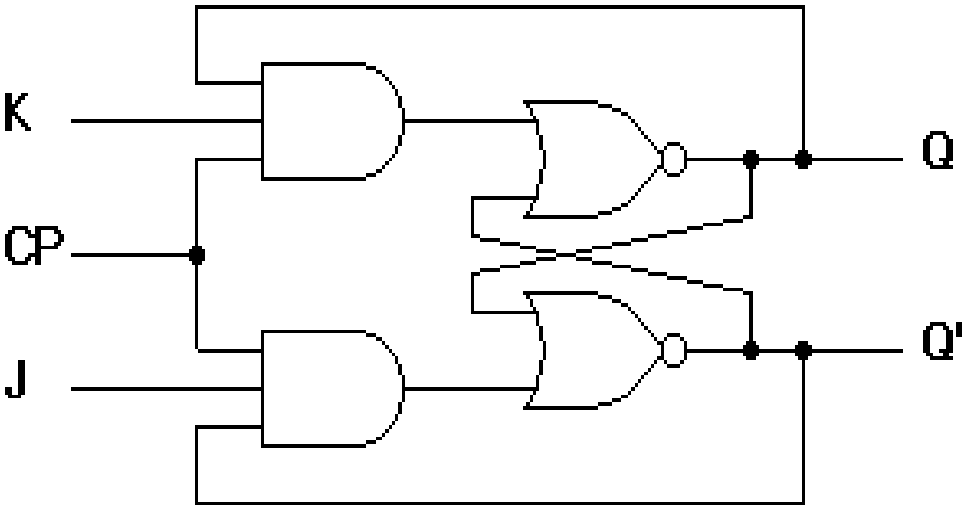


Truth Table

J	K	OUTPUT Q
0	0	No Change
0	1	Clear to 0
1	0	Set to 1
1	1	Q(t)’ – complement

- When inputs J and K are both equal to 1, a clock transition switches the outputs of the flip-flop to their complement state.
- Instead of the indeterminate condition of the SR flip-flop, the JK flip-flop has a complement condition Q(t+1) = Q’(t) when both J and K are equal to 1.

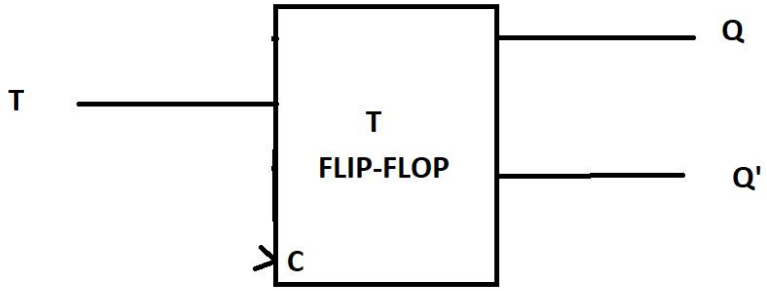
Circuit of JK flip-flop



19. EXPLAIN T FLIP-FLOP IN DETAILS.

- The T stands for toggle Flip-Flop.
- It contain only 1 input T
- It is used to store invert value of input.

Block diagram

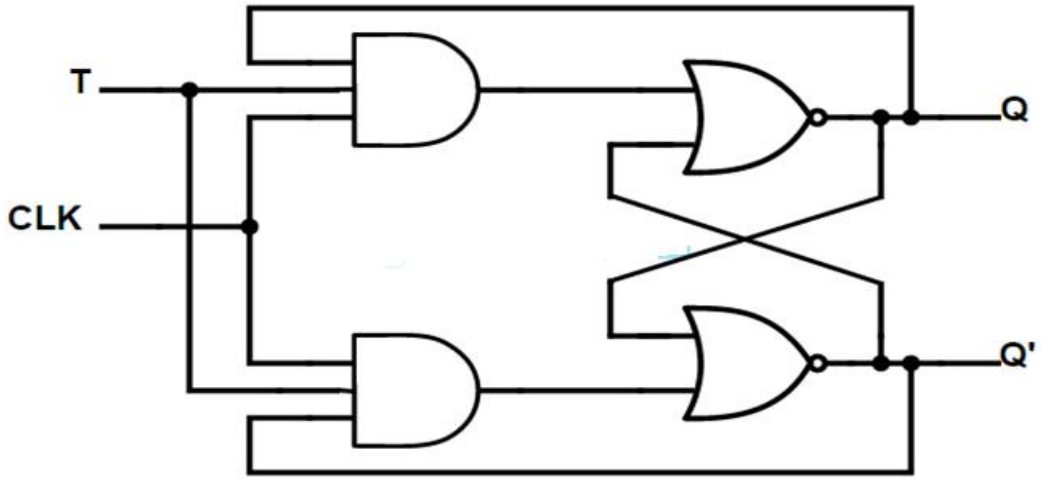


Truth Table

Clk	T	OUTPUT Q
0	0	No Change
0	1	No Change
1	0	1 (Q')
1	1	0 (Q)

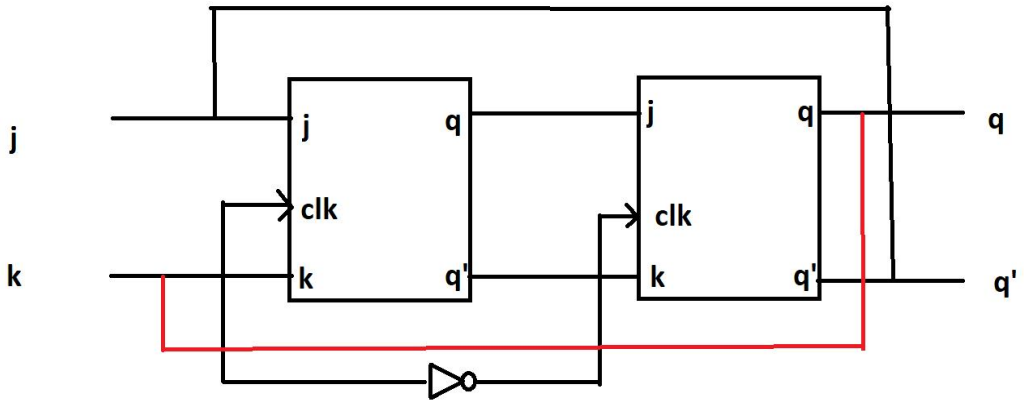
- The T flip-flop is modification of the J-K flip-flop.
- As shown in truth table that this flip flop contain single input called T.
- It is not widely available so we can construct it with JK flip-flop
- When Clock pulse 0 then this flip flop will not work.
- When clock pulse 1 & T=0(j,k=0) then it will change its state.

Circuit of T flip-flop



20. EXPLAIN MASTER-SLAVE FLIP-FLOP IN DETAILS.

- Master-slave is a combination of two flip-flops connected in series, where one acts as a master and another act as a slave.
- Each flip-flop is connected to a clock pulse complementary to each other,
- i.e., if the clock pulse is in high state, the master flip-flop is in enable state, and the slave flip-flop is in disable state, and if clock pulse is low state, the master flip-flop is in disable state, and the slave flip flop is enable state.
- Mater slave configuration is mainly used to eliminate the race around the condition and get rid of unstable oscillation in the flip flop.



- JK flip is used to produce master slave flip flop.
- Where master flip-flop will receive positive clock pulse produce output at that time slave will not perform operation.
- When negative clock pulse received master become inactive and slave become active.
- The output of master will send as an input of slave so slave will produce output.

Advantages of Master Slave Flip Flop

- Master slave can be operated on level triggered or edge triggered clock pulse; it can be used in various ways.

- A sequential circuit with an edge-controlled flip flop is straightforward to design rather than a level-triggered flip flop.
- By using the Master slave configuration, we also can eliminate the race around the condition.

21. WRITE A NOTE ON SEQUENTIAL CIRCUIT.

- It is a digital logic circuit whose output depends on the present inputs as well as previous inputs.
- It can describe by the output values as well as state values. It contains at least one memory element.
- It is difficult to design and understand.
- However, if a circuit uses both gates and flip-flops, it is called a sequential circuit.
- Hence, a sequential circuit is an interconnection of flip-flops and gates.
- If we think of a sequential circuit as some black box that, when provided with some external input, produces some external output.
- The external inputs constitute some of the inputs to the combinational circuit.
- The internal outputs of the combinational circuit are the internal inputs to the flip-flops.
- Examples of sequential circuits:
    - Registers
    - Shift Registers
    - Counters
    - Ripple Counters
    - Synchronous Counters
    - Flip-flops

Classification of Sequential Circuits

- Based on the clock signal input, the sequential circuits are classified into two types.
    - Synchronous sequential circuit
    - Asynchronous sequential circuit

Synchronous Sequential Circuits

- In Synchronous sequential circuit, the output depends on present and previous states of the inputs at the clocked instances.
- The circuits use a memory element to store the previous state.
- The memory elements in these circuits will have clocks.
- All these clock signals are driven by the same clock signal.

Synchronous Sequential Circuits characteristics

- Using clock signal, state changes will occur across all storage elements.
- These circuits are bit slower compared to asynchronous.
- These circuits can be clocked or pulsed.

- The Synchronous sequential circuits that use clock pulses in their inputs are called clocked -sequential circuits. They are very stable.
- The sequential circuits that change their state using the pulse and these are called pulsed or un-clocked sequential circuits.

Asynchronous Sequential Circuits

- The Sequential circuits which do not operate by clock signals are called "Asynchronous sequential circuits".
- These circuits will change their state immediately when there is a change in the input signal .
- The Circuit behavior is determined by signals at any instant in time and the order in which input signals change.

Asynchronous Sequential Circuits characteristics

- They do not operate in pulse mode.
- They have better performance but hard to design due to timing problems.
- Mostly we use the asynchronous circuits when we require the low power operations.
- They are faster than synchronous sequential circuits as they do not need to wait for any clock signal.

22. WRITE DOWN THE DIFFERENCE BETWEEN SEQUENTIAL & COMBINATIONAL CIRCUIT.

COMBINATIONAL CIRCUIT	SEQUENTIAL CIRCUIT
• It is a digital logic circuit whose output depends on the present inputs.	• It is a digital logic circuit whose output depends on the present inputs as well as previous inputs.
• It can describe by the output values.	• It can describe by the output values as well as state values.
• It contains no memory element.	• It contains at least one memory element.
• It is easy to design and understand.	• It is difficult to design and understand.
• It is faster in speed.	• It is slower in speed.
• It is expensive in cost.	• It is less expensive in cost.
• Examples of combinational circuit are half adder and full adder.	• Examples of sequential circuit are flip-flops like RS, Clocked RS, D and JK.
• A combinational circuit is a connected arrangement of logic gates with a set of inputs and outputs.	• However, if a circuit uses both gates and flip-flops, it is called a sequential circuit.
• At any given time, the binary values of the outputs are a function of the binary values of the inputs.	• Hence, a sequential circuit is an interconnection of flip-flops and gates.

• The design of a combinational circuit starts from a verbal outline of the problem and ends in a logic circuit diagram.	• If we think of a sequential circuit as some black box that, when provided with some external input, produces some external output
• **Examples of sequential circuits:** ○ Half adder ○ Full Adder ○ Multiplexers ○ De-multiplexers ○ Encoders ○ Decoders.	• **Examples of sequential circuits:** ○ Registers ○ Shift Registers ○ Counters ○ Ripple Counters ○ Synchronous Counters ○ Flip-flops

Unit : 2
Digital Components

---

1. WRITE A NOTE ON INTEGRATED CIRCUITS.

- The IC manufactured using silicon material and mounted in a ceramic or plastic container.
- It is Also known as Chip.
- The basic components of an IC consist of electronic circuits for the digital gates.
- The various gates are interconnected inside an IC to form the required circuit.

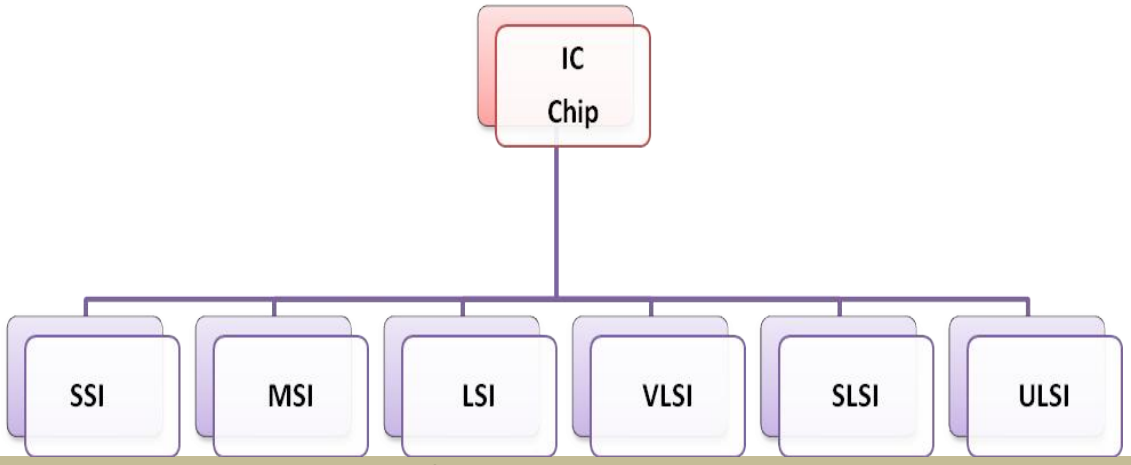
Features of IC

- It Combines The various gates are interconnected inside an IC.
- The Gates are basic component of IC.
- IC Combines other components like register, transistor etc.
- All components are fixed.
- The components that are comprises in IC Cant repair , replaced or removed.
- IC can works on low voltage.
- IC can handle limited amount of voltage.
- Very small in size as compared to other components.
- As compared to other circuit IC is Cheap in cost.
- IC gives better performance in complex operations.

Applications of IC

- They are used in smart phones, mp3 players, laptops, computers.
- IC's are also used in Television and cameras.
- IC's are the basic component used in scientific calculators and digital watches.
- They are used in control systems.

Classification Of IC



SSI (Small Scale Integration Devices)

- These type of devices contain several independent gates in a single package.
- The number of logic gates are usually less than 10 and are limited by the number of pins available in the IC.

MSI (Medium Scale Integration Devices)

- These type of devices has a complexity of approximately 10 to 200 gates in a single package.
- The basic components include decoders, adders, and registers.

LSI (Large Scale Integration Devices)

- A 200 to a 1000 gates in a single package.
- The basic components of an LSI device include digital systems, such as processors, memory chips.

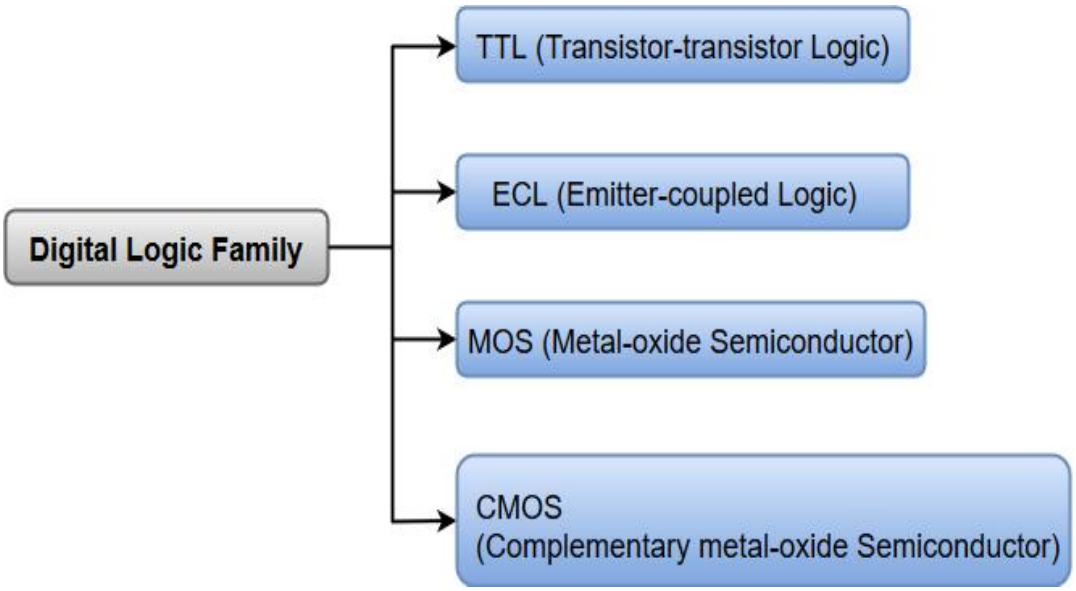
VLSI (Very Large Scale Integration Device)

- It contain 1,000-10,000 or thousands of Gates.
- The most common example of a VLSI device is a complex microcomputer chip.
- The most common component in each technology is either a NAND, a NOR, or an inverter gate.
- The most popular among the digital logic families include:
  - TTL
  - ECL
  - MOS
  - CMOS

ULSI (Ultra Large Scale Integration Device)

- In this type of fabrication more than 1 Million Transistors are integrated on a single chip.

2. WRITE A NOTE ON DIGITAL LOGIC FAMILY.



TTL- Transistor-transistor Logic:

- The DTL technology used to have diodes and transistors for the basic NAND gate
- It's an upgraded version of a DTL-Diode-Transistor Logic.
- TTL came in existence when these diodes are replaced with transistors to improve the circuit operation.
- support at most 10 gates at its output.
- used as a switching device in driving lamps and relays.
- It represents the amount of noise voltage allowed at the input, which doesn't affect the standard output.
- It is also used in printers and video display terminals.

ECL- Transistor-transistor Logic

- It provides the highest-speed digital circuits in integrated form.
- The basic gate to design ECL is OR/NOR.
- high-speed bipolar logic family.
- Used in super computer .
- power consumption is more as compared to other logic families.

MOS- Metal-oxide semiconductor

- The MOS is a unipolar transistor.
- It's a semiconductor device used for switching & amplification of the electronic signals.
- Used to control voltage & current flow.
- It has an Ability to convert low voltage to high.

CMOS- Complementary metal-oxide semiconductor

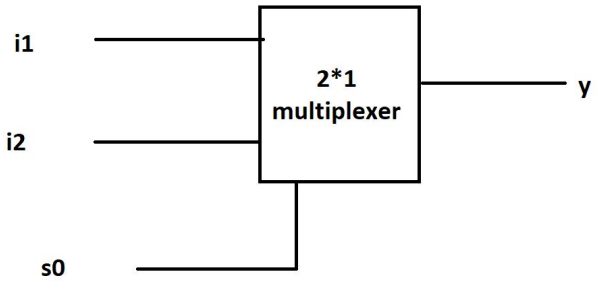
- The CMOS is a unipolar transistor.

- CMOS logic families are highly preferred in large-scale integrated circuits because of its high noise immunity and low power dissipation.
- Manage wastage of power.
- It Contain small amount of memory that store BIOS setting

3. WHAT MULTIPLEXER? EXPLAIN 2*1 MULTIPLEXER IN DETAIL.

- MUX is combinational circuit that is used to direct one out of 2^n input data Lines to A single output line.
- It is also known as data selector because it selects one of many inputs And directs it to the output.
- The selection of particular input data line is controlled by as Set of selection inputs.
- Normally there are 2n input data lines and n input selection lines.

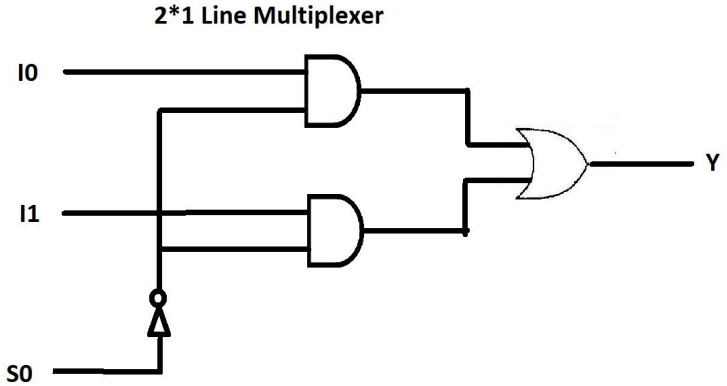
BLOCK DIAGRAM



TRUTH TABLE

Selection	Output
S0	Y
0	I0
1	I1

2*1 MULTIPLEXER CIRCUIT



- As shown circuit that 2*1 multiplexer circuit design with 2 AND gate & 1 OR gate.
- It contain 1 selection line; when selection S0=0 then input of I0 selected by multiplexer & when selection line S1=1 then input of I1 selected by multiplexer.

4. EXPLAIN 4*1 MULTIPLEXER IN DETAIL.

- MUX is combinational circuit that is used to direct one out of 2^n input data Lines to A single output line.
- To select input it required n number of selection lines.

BLOCK DIAGRAM

- 4*1 multiplexer contain 4 inputs lines **I0,I1,I2,I3** , 1 output line **Y** & 2 selection line **S0** and **S1**.

TRUTH TABLE

Selection line		Output
S1	**S0**	**Y**
0	**0**	**I0**
0	**1**	**I1**
1	**0**	**I2**
1	**1**	**I3**

4*1 multiplexer

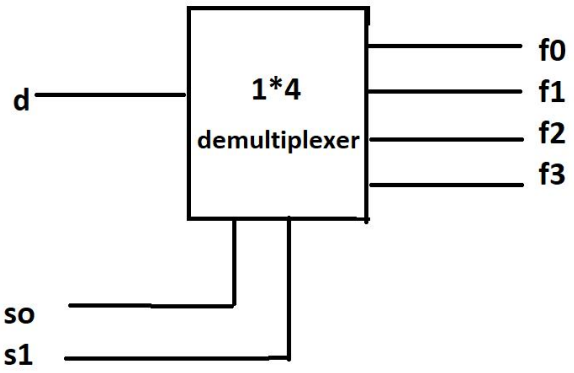
- As shown circuit that 4*1 multiplexer circuit design with 4 AND gate & 1 OR gate.
- When selection line S0,S1=0 then AND gate connected with I0 produce high output.
- When selection line S0=0,S1=1 then AND gate connected with I1 produce high output.
- When selection line S0,S1=0 then AND gate connected with I2 produce high output.
- When selection line S0,S1=0 then AND gate connected with I3 produce high output.

5. WHAT IS DE-MULTIPLEXER? EXPLAIN 1*4 LINE DEMULTIPLEXER.

- A de-multiplexer is a circuit that receives information on a single line and transmits this information on one of 2^n possible output lines.
- It is also known as DEMUX or **DATA DISTRIBUTOR.**
- Perform reverser operation then multiplexer.
- De-multiplexer is combinational circuit that is used to direct one input to 2^n output line.
- It is also known as data distributor because it send one input to more than output.
- The selection of a specific output line is controlled by the bit values of n selection lines.
- In simple word DEMUX means **ONE TO MANY**

- In 1*4 de-multiplexer **1** input Line, 2^n **= 4** output Lines , n= 2 number of selection lines.

BLOCK DIAGRAM

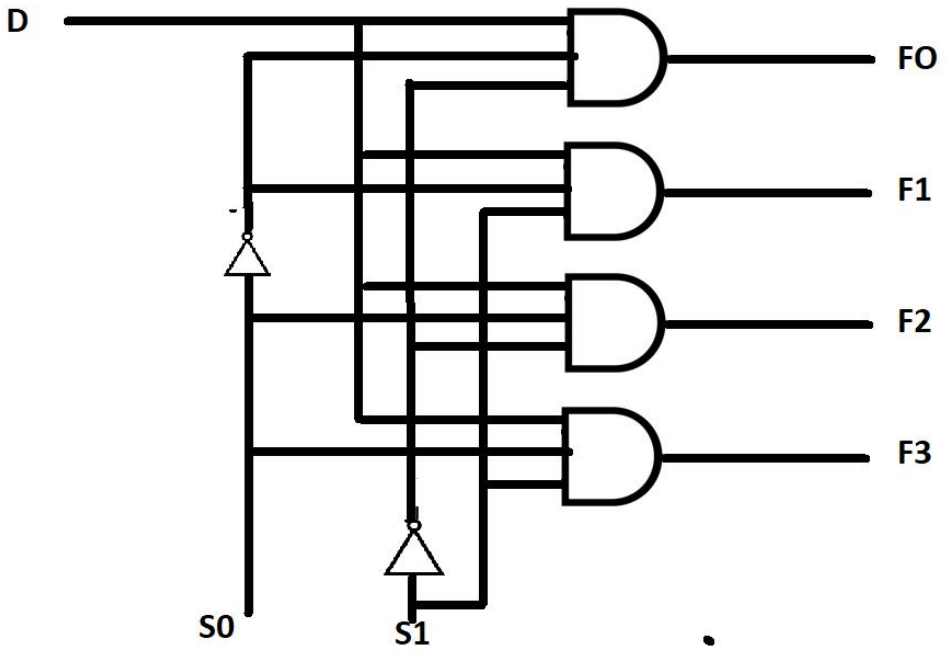


- 1*4 de-multiplexer contain 1 inputs lines **d,** 4 output line **F0,F1,F2,F3** & 2 selection line **S0** and **S1**.

TRUTH TABLE

Input	Selection Lines		Output			
D	S1	S0	F0	F1	F2	F3
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

1*4 DEMULTIPLEXER CIRCUIT



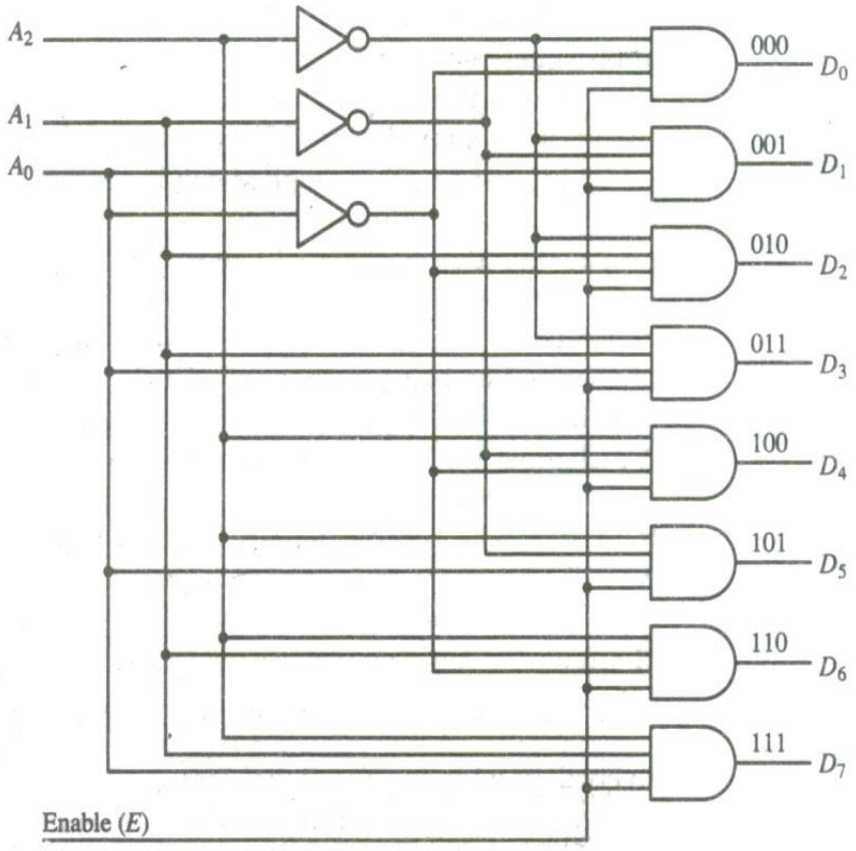
6. EXPLAIN 3*8 LINE DECODER.

- A decoder is a combinational circuit that converts binary information from the n coded inputs to a maximum of 2^n unique outputs.
- If the n-bit coded information has unused bit combinations, the decoder may have less than 2^n outputs.
- The decoders presented in this section are called n-to-m-line decoders, where m <= 2^n.
- A decoder has n inputs and m outputs and is also referred to as an n x m decoder.

TRUTH TABLE OF DECODER

ENABLE	INPUTS			OUTPUTS							
E	A0	A1	A2	D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

3*8 decoder



- The three data inputs. A0, A1, and A2, are decoded into eight outputs, each output representing one of the combinations of the three binary input variables.
- A 3-to-8-line decoder can be used for decoding any 3-bit code to provide eight outputs, one for each combination of the binary code.
- The decoder of the Figure has one enable input, E.
- The decoder is enabled when E is equal to 1 and disabled when E is equal to 0.
- The operation of the decoder can be clarified using the truth table listed in Table.
- When the enable input E is equal to 0, all the outputs are equal to 0 regardless of the values of the other three data inputs.
- When the enable input is equal to 1, the decoder operates in a normal fashion.
- For each possible input combination, there are seven outputs that are equal to 0 and only one that is equal to 1.
- The output variable whose value is equal to 1 represents the octal number equivalent of the binary number that is available in the input data lines.

7. WHAT IS ENCODER? EXPLAIN OCTAL TO BINARY ENCODER.

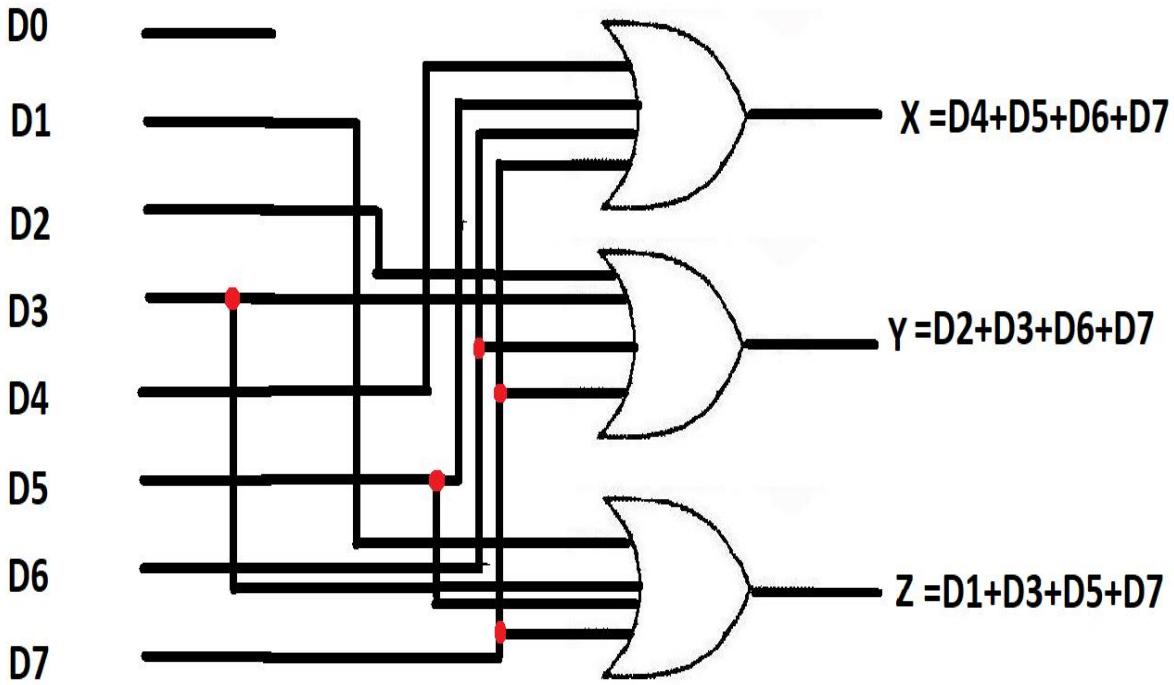
- An encoder is a digital circuit that performs the inverse operation of a decoder.
- An encoder has 2^n (or less) input lines and n output lines.
- The output lines generate the binary code corresponding to the input value.
- An example of an encoder is the octal-to-binary encoder, whose truth table is given below.

Truth Table Of Encoder

Inputs								outputs		
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

- The encoder can be implemented with three OR gates.

Octal-to-binary Encoder Circuit

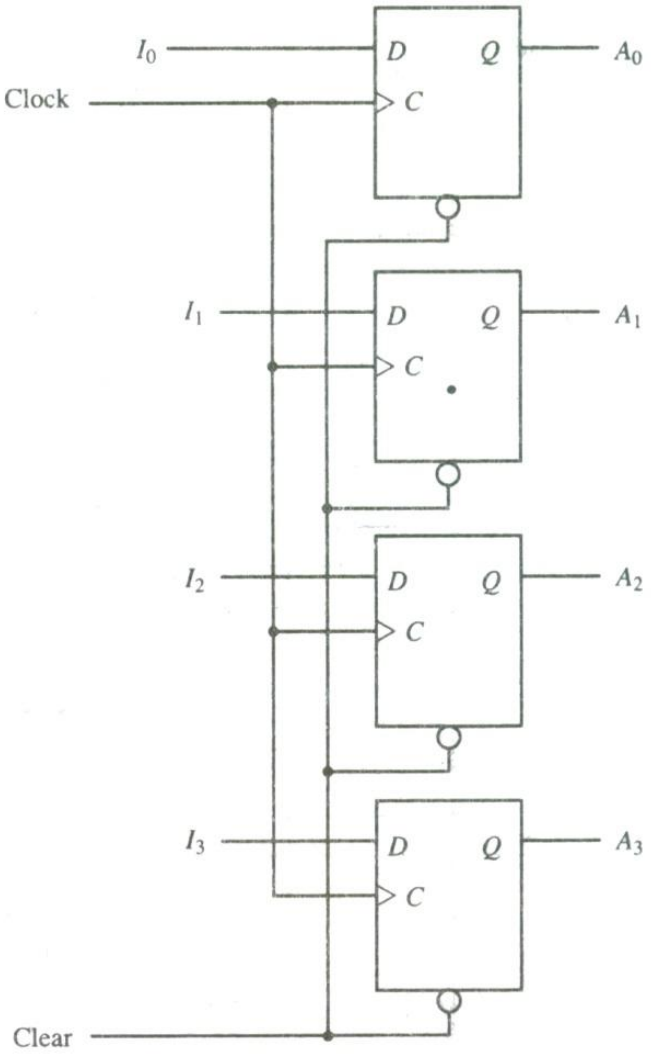


- The encoder can be implemented with OR gates whose inputs are determined directly from the truth table.
- Output A0 =1 if the input octal digit is 1 or 3 or 5 or 7. Similar conditions apply for other two outputs.
- These conditions can be expressed by the following Boolean functions :

X = D1 + D3 + D5 + D7
Y = D2 + D4 + D6 + D7
Z = D4 + D5 + D6 + D7

8. WHAT IS REGISTER?

- A register is a group of flip-flops capable of storing one bit of information.
- An n-bit register has a group of n flip-flops and is capable of storing any binary information of n bits.
- In addition to flip-flops, registers can have combinational gates that perform certain data-processing tasks.
- The gates control how and when new information is transferred into the registers.
- The transfer of new information into a register is referred to as a register load.



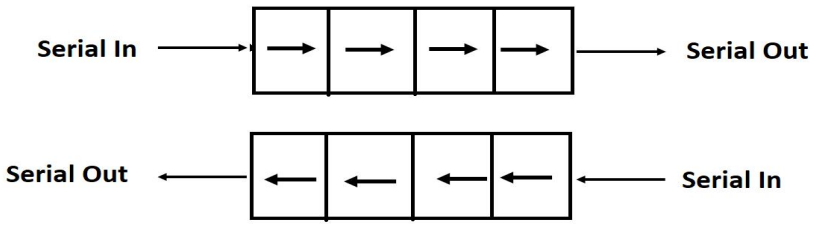
9. EXPLAIN TYPES OF SHIFT REGISTER.

Types of shift register:

- Generally register are divided into 5 types.
  - **Serial In Serial Out – (SISO)**
  - **Serial In Parallel Out –(SIPO)**
  - **Parallel In Serial Out –(PISO)**
  - **Parallel In Parallel Out—(PIPO)**
  - **Bidirectional Shift Register—(SERIAL IN SERIAL OUT SHIFT LEFT SHIFT RIGHT)**

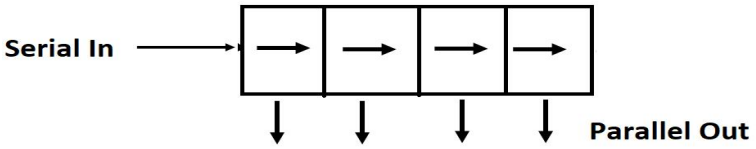
Serial In Serial Out – (SISO)

- In this type of register data inserted in serial manner.
- It means that one by one bit will be transfer inside register similarly one by one bit will be out.
- Data can be inputted from either left or right direction.



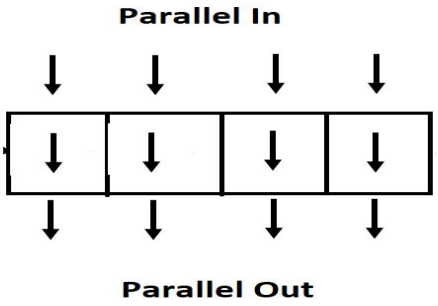
Serial In Parallel Out – (SIPO)

- In this type of register data inserted in serial manner.
- It means that one by one bit will be transfer inside register.
- When all bits are shifted into register but all bits will be out parallel.



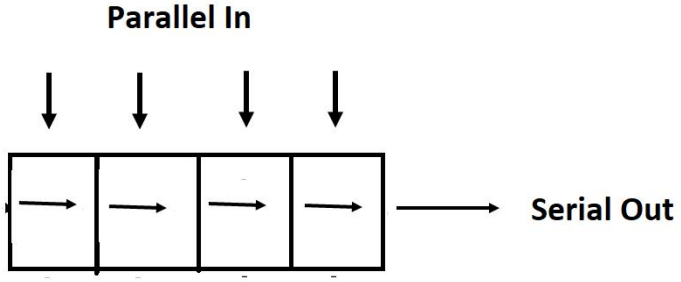
Parallel In Parallel Out – (PIPO)

- In this type of register data will in and out parallel
- It means that all bits will be transfer inside register at the same time & similarly all bits will be out parallel.



Parallel In Serial Out – (PISO)

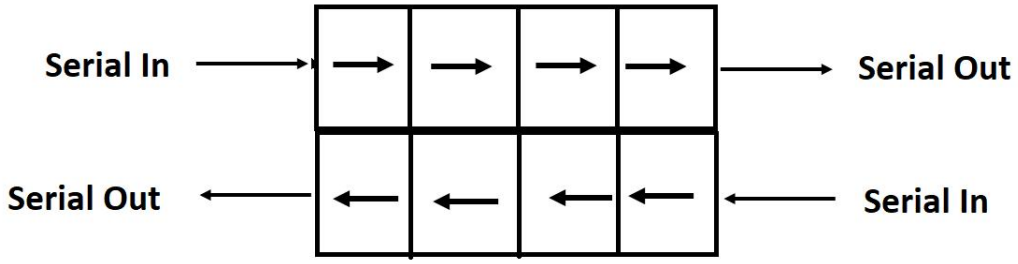
- In this type of register data will in parallel manner.
- It means that all bits will be transfer inside register at the same time but it will be out in serial manners.
- It means that more than one bit will be inputted but bit wise data will be out.



Bidirectional Shift Register

- It is also known as serial in serial out shift left shift right register.

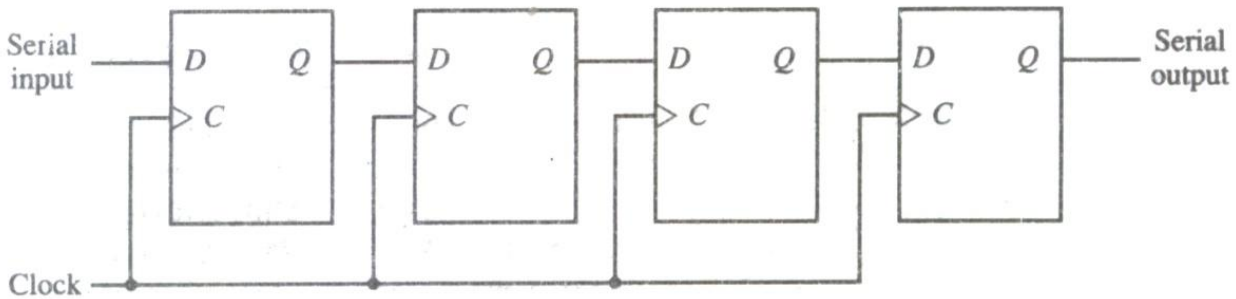
- In this type of register data will in bit wise from left and right direction at the same time; similarly data will be out from both directions.
- That's the reason that this register is known as bi-directional shift register.



10. EXPLAIN 4 BIT SHIFT REGISTER.

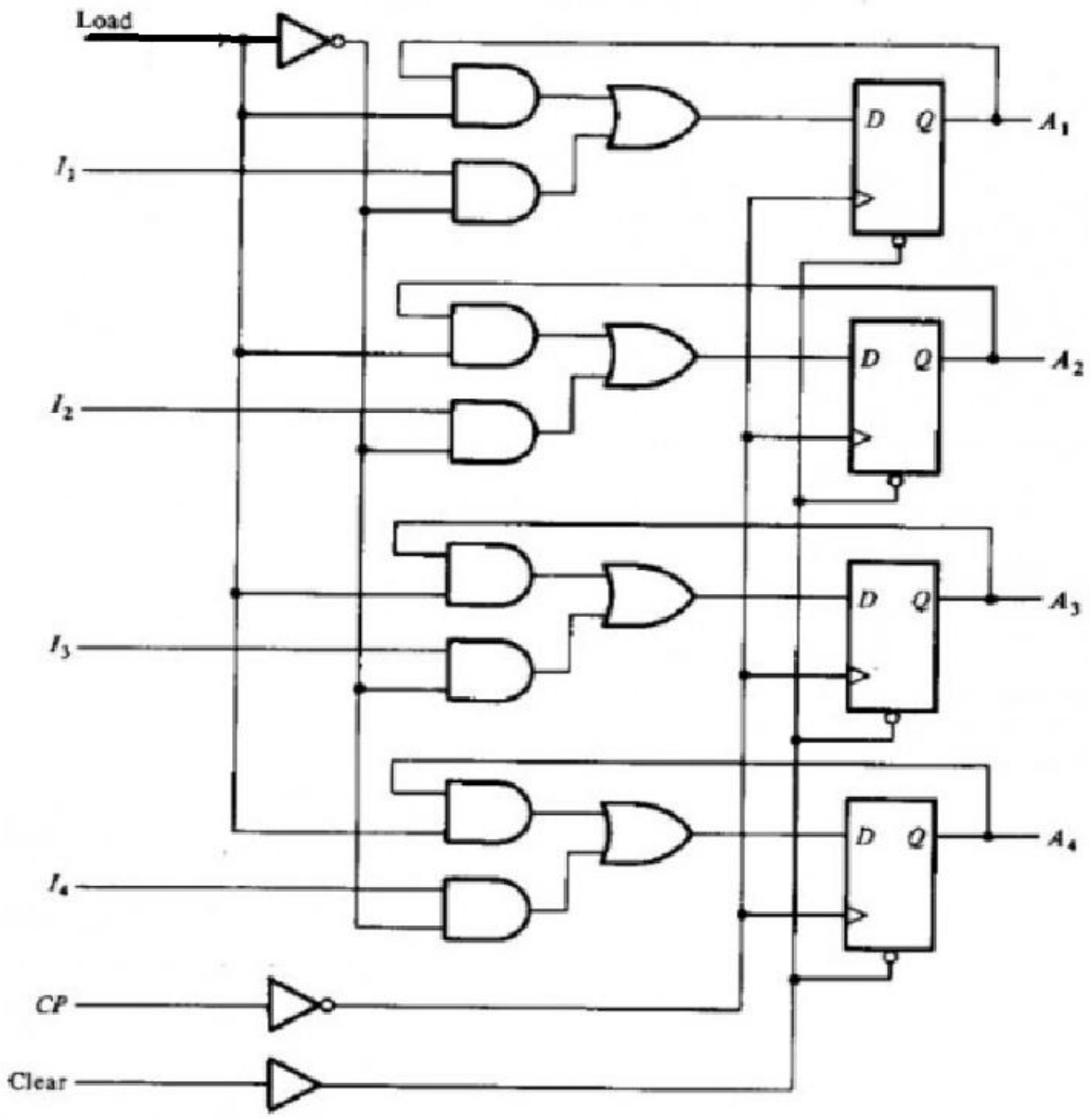
- A register capable of shifting its binary information in one or both directions is called a shift register.
- Shift registers are constructed by connecting flip-flops in cascade, where the output of one flip-flop is connected to the input of the next flip-flop.
- All flip-flops receive common clock pulses that initiate the shift from one stage to the next.
- Each remaining flip-flop uses the output of the previous flip-flop as its input, with the last flip-flop producing the external output (called the serial output).
- A register capable of shifting in one direction is called a unidirectional shift register.
- A register that can shift in both directions is called a bi-directional shift register.
- The most general shift register has the following capabilities:
    - An input for clock pulses to synchronize all operations.
    - A shift-right operation and a serial input line associated with the shift-right.
    - A shift-left operation and a serial input line associated with the shift-left.
    - A parallel load operation and n input lines associated with the parallel transfer.
    - N parallel output lines.
    - A control state that leaves the information in the register unchanged even though clock pulses are applied continuously.
    - A mode control to determine which type of register operation to perform.

4 bit shift register block diagram



11. EXPLAIN SHIFT REGISTER WITH PARALLEL LOAD.

- If the loading occurs simultaneously at a common clock pulse transition, we say that the load is done in parallel.
- The load input in a register determines the action to be taken with each clock pulse.
- When the load input is 1, the data from the input lines is transferred into the register's flip-flops.
- When the load input is 0, the data inputs are inhibited and the flip-flop maintains its present state.

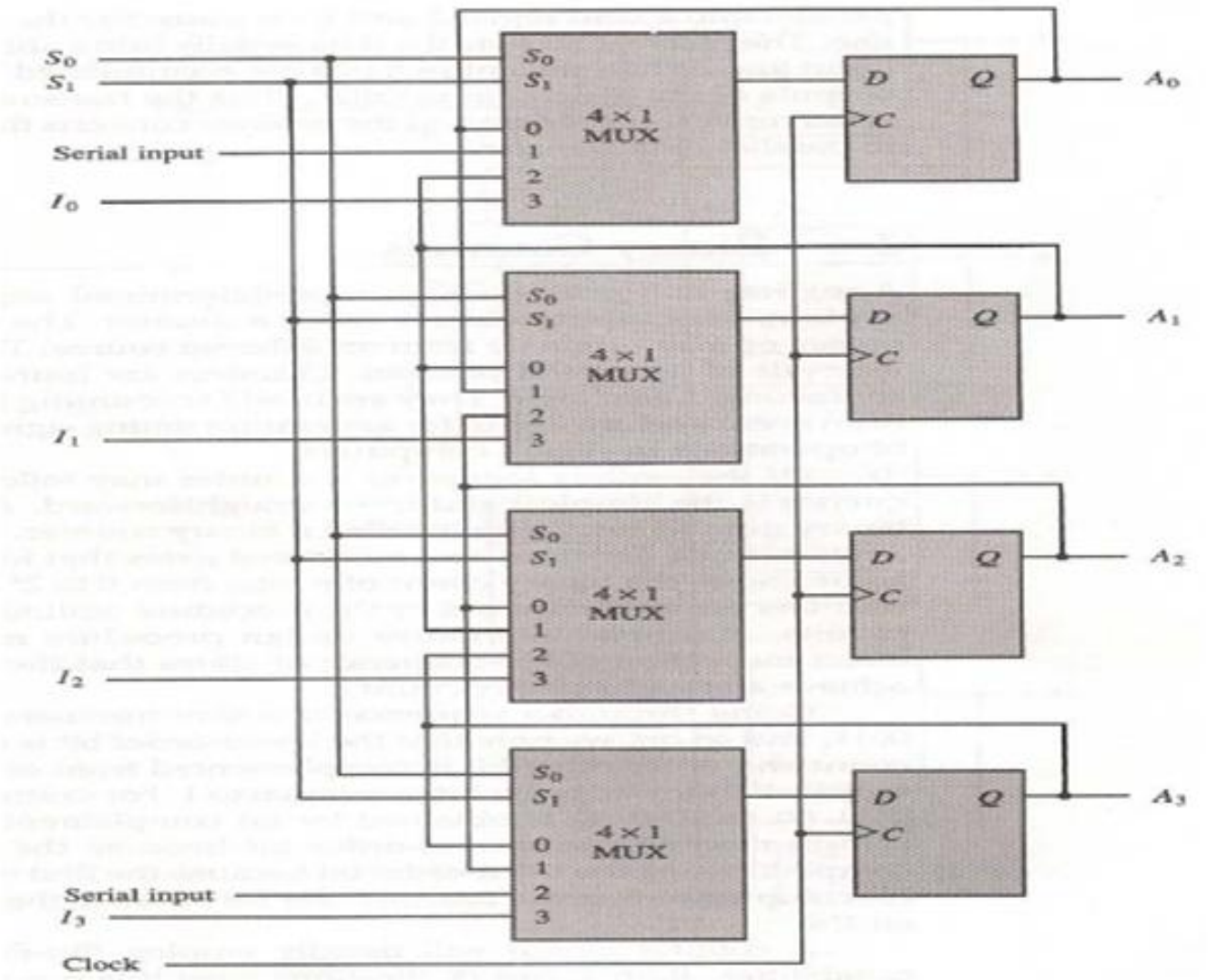


- As shown in diagram I0,I1,I2,I3 are input lines & A0,A1,A2,A3 are output lines.
- D flip flop is used because it will store the same value as we passed as input.

- All flip flops are connected with a common clock pulse. When Load input L=1 inputs will load inside register & when Load input L=0 all bits out together.
- To clear all values from flip flop we pass 0

12. EXPLAIN BI-DIRECTIONAL SHIFT REGISTER WITH PARALLEL LOAD.

- A register that can shift in both directions is called a bidirectional shift register.
- Some shift registers provide the necessary input and output terminals for parallel transfer.
- These registers is design by using flip-flop as well multiplexer.
- A 4-bit bidirectional shift register with parallel load Each stage consists of a D flip-flop and a 4 x 1 multiplexer.
- The two selection inputs s1 and s0 select one of the multiplexer data inputs for the D flip-flop.
- The selection lines control the mode of operation of the register according to following table.



- When the mode control s1s0 = 00, data input 0 of each multiplexer is selected.
- This condition forms a path from the output of each flip-flop into the input of the same flip-flop.

- When s1s0 = 01, the terminal marked 1 in each multiplexer has a path to the D input of the corresponding flip-flop.
- This causes a shift-right operation, with the serial input data transferred into flip-flop A0 and the content of each flip-flop A, _ 1 transferred into flip-flop A; for i = 1, 2, 3.
- When s1s0 = 10 a shift-left operation results, with the other serial input data going into flip-flop A, and the content of flip-flop A, + 1 transferred into flip-flop A; for i = 0, I, 2.
- When s1s0 = 11, the binary information from each input 10 through I, is transferred into the corresponding flip-flop, resulting in a parallel load operation.

13. WRITE A NOTE ON 4 BIT SYNCHRONOUS BINARY COUNTER.

- The synchronous counter can be defined as, a counter which uses a clock signal for transforming their transition.
- So, these counters mainly depend on the input of the clock to modify state values. In this counter, all flip flops (FFs) are associated with the same clock signal to activate simultaneously.
- An alternate name of this counter is a simultaneous counter .
- To design a counter we required
    - Number of possible values: 2^4=16, values 0 to 15
    - Number of flip flop➔ 4
    - Common clock pulse

- Synchronous binary counters have a regular pattern, as can be seen from the 4-bit binary counter shown in bellow
- The C inputs of all flip-flops receive the common clock.
- If the count enable is 0, all I and K inputs are maintained at 0 and the output of the counter does not change.
- The first stage A0 is complemented when the counter is enabled and the clock goes through a positive transition.
- Each of the other three flip-flops are complemented when all previous least significant flip-flops are equal to 1 and the count is enabled.
- The chain of AND gates generate the required logic for the J and K inputs. The output carry can be used to extend the counter to more stages, with each stage having an additional flip-flop and an AND gate.

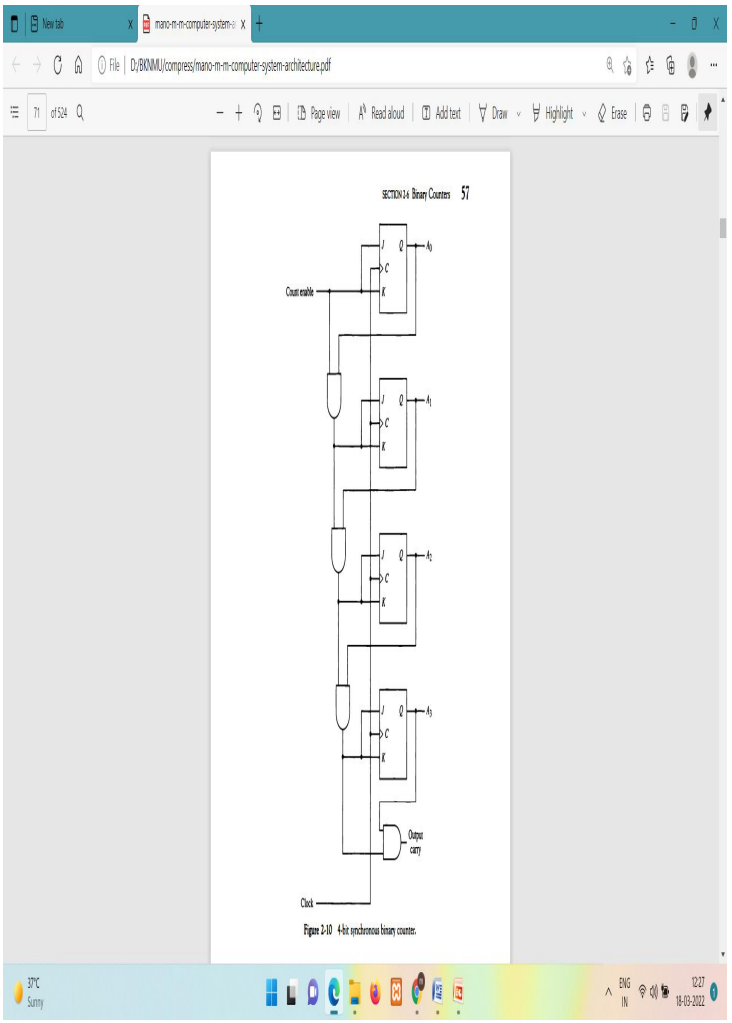
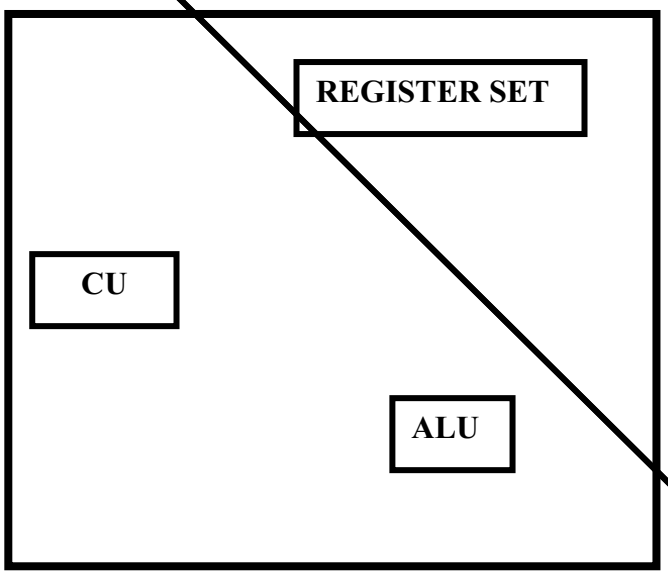


Figure 2-10 4-bit synchronous binary counter.

Unit : 4
Central Processing Unit

1. WRITE A NOTE ON CENTRAL PROCESSING UNIT.

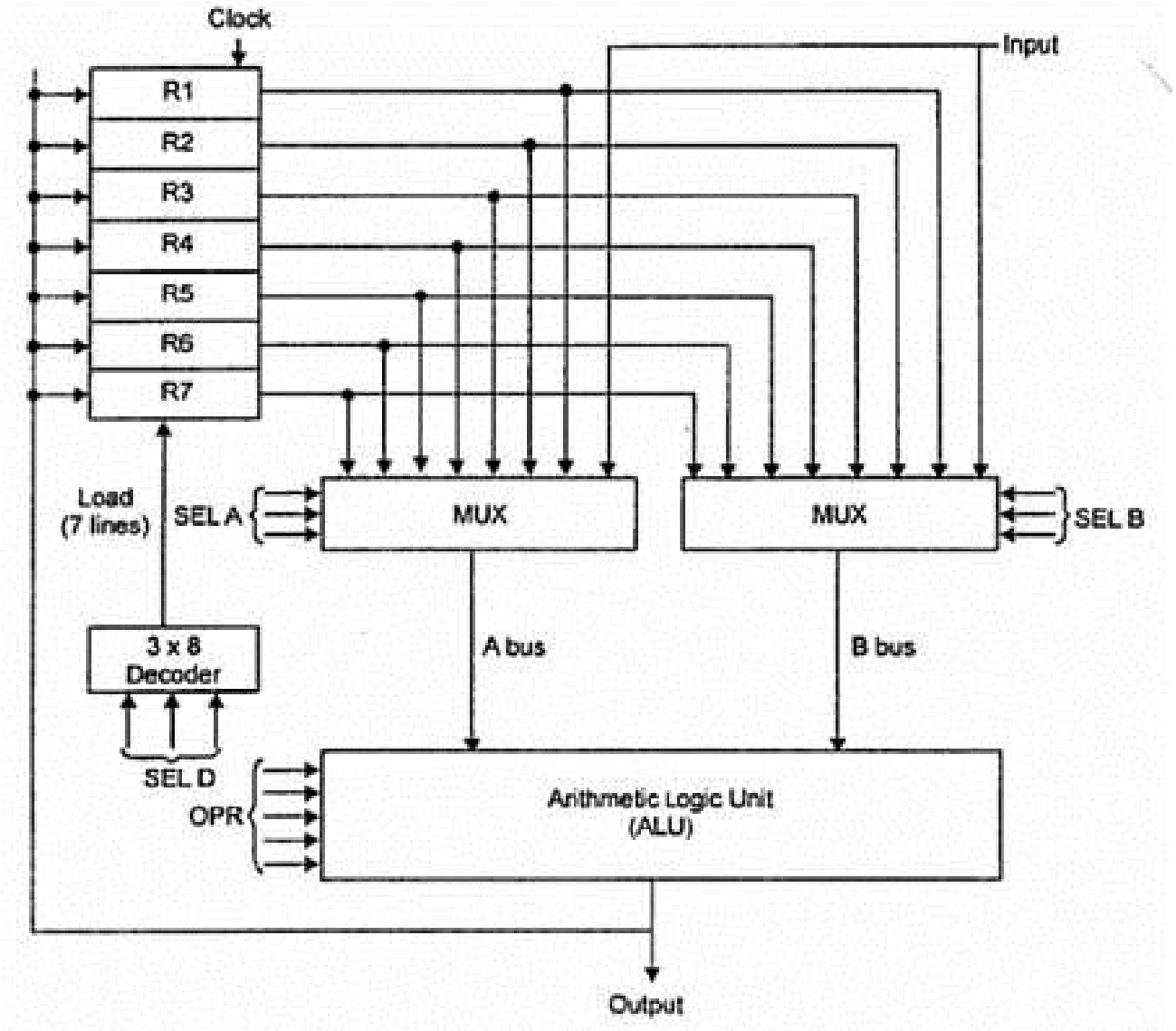
- The central processing unit is the brain of digital computer.
- It performs various types of function depending on the instruction that incorporated in the computer.
- The main function of the central processing unit is to execute the programs and generates the results, except that it also controls the internal and external parts of the computers.



- The control unit, register sets and ALU are the major three components of central processing unit.
- The main memory is also a part of central processing unit.
- The CPU of small computer contains a single microprocessor while the CPU of large computer may contain multiple microprocessors.
- The microprocessor contains two major parts that is CU – Control Unit and ALU – Arithmetic and Logical Unit.
- **Control Unit:**
  - it controls the entire operations of the system. It also controls the input and output devices.
- **Arithmetic & Logical Unit:**
  - it performs the mathematical as well as logical operations on the instruction. This unit is responsible to generate the output on inputted instructions.
- **Register sets:**
  - the register sets contain more than one registers. The registers are basically used for storing immediate results during the process of executing instruction. Various registers performs various task.

2. EXPLAIN GENERAL REGISTER ORGANIZATION.

- The set of registers in a computer are connected to the ALU using busses and Multiplexers.
- A 14-bit control word specifies two source registers (SELA & SELB), a destination register (SEL D), and an operation (OPR).
- The registers can be specified using three bits each as follows:



- A bus organization for seven CPU registers is shown in fig.
- The output of each Register Connected to two multiplexers to perform the two buses A and B.
- The selection lines in each multiplexer select one register of the input data for the particular logic unit (ALU).
- The operation selected in the ALU determines the arithmetic or logic micro operations that are to be performed.
- The result of the micro-operation is available for output data and also goes into the inputs of all the registers.
- The register that receives the information from the output bus is selected the decoder.

- The decoder activates one of the register load inputs, thus providing a transfer Path between the data in the output bus and the inputs of the selected destination register.
- The control unit that operates the CPU bus system directs the information flow through the registers and ALU selecting the various components in the system.

3. WHAT IS CONTROL WORD?

- There are 14 – bit binary selection input in the unit and their combine value that specify a control word.
- The figure of 14 – bit combine word is as given below:

SELECTION A	SELECTION B	SELECTION D	OPR
3 bit	3 bit	3 Bit	5 bit

- In this 14 – bit combine word, first 3 columns contains 3 bits and last column contain 5 bits.
- The first field is a SELECTION A that contains 3 bits
- The second field is a SELECTION B that contains 3 bits
- The third field is a SELECTION D that contains 3 bits
- The last field is a OPR that contains 5 bits
- The 3 bits of SELECTION – A, SELECTION – B and SELECTION – D that selects a source register for the input of the ALU
- The 5 bits of OPR that selects a destination register for output of the ALU

4. WHAT IS STACK? EXPLAIN REGISTER STACK IN DETAIL.

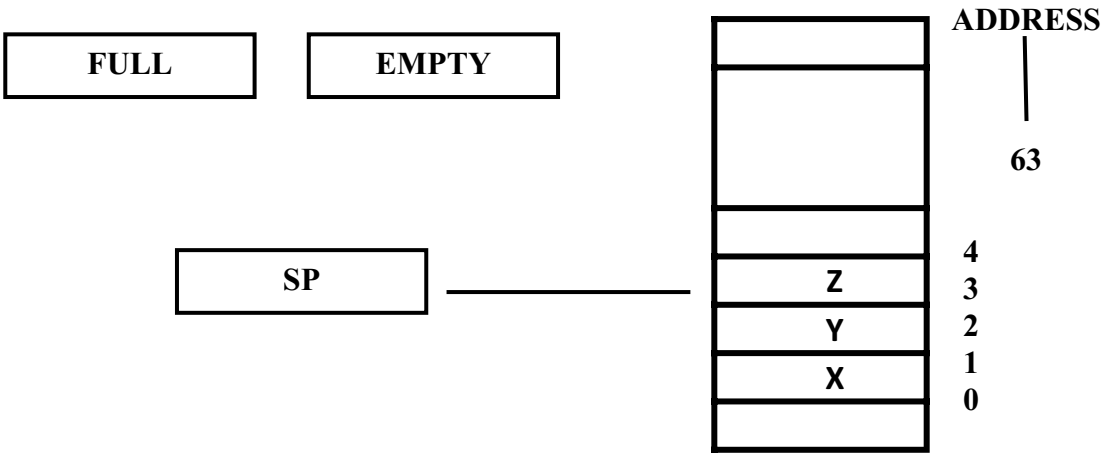
- A stack is an ordered set of elements only one of which can be used at a time
- The point of access is called top of the stack
- The number of elements in the stack or length of the stack is a variable
- The items may be added to or deleted from the top of the stack as Push Down list or Last In First Out (LIFO) List
- A useful feature that is included in the CPU of most computer is stack or Last In First Out (LIFO) List
- A stack is a storage device that stores information in such a manner that the item stored last is the first item retrieved.
- The register that holds the address for the stack is called a Stack Pointer (SP) because its value always points at the top item in the stack
- The two operation of stack are PUSH and POP
- The operation of insertion is called PUSH
- The operation of deletion is called POP

TYPES OF STACK OPERATION

- There are two types of stack operation like REGISTER STACK & MEMORY STACK

REGISTER STACK

- A stack can be placed in a portion of a large memory or it can be organized as a collection of a finite number of memory words or registers
- Register stacks organized the operation from each and every word and its operation into the memory in register stack operation
- The stack pointer (SP) register contains 4 binary numbers whose value is equal to the address of the word that is currently on top of the stack



- The fig shows the organization of 64 words register stack.
- Three items are place in stack that is x, y and z item is top of the stack.
- So that SP is now 3.
    - SP means Stack pointer.
    - DR means Data Register
- There are two types of operations are used in this type of Register Stack Operation like PUSH and POP

PUSH

- If the stack is not full then a new item is inserted with a push operation
- If the stack is full and user tries to insert a new value in the stack then it is called Stack Overflow error.

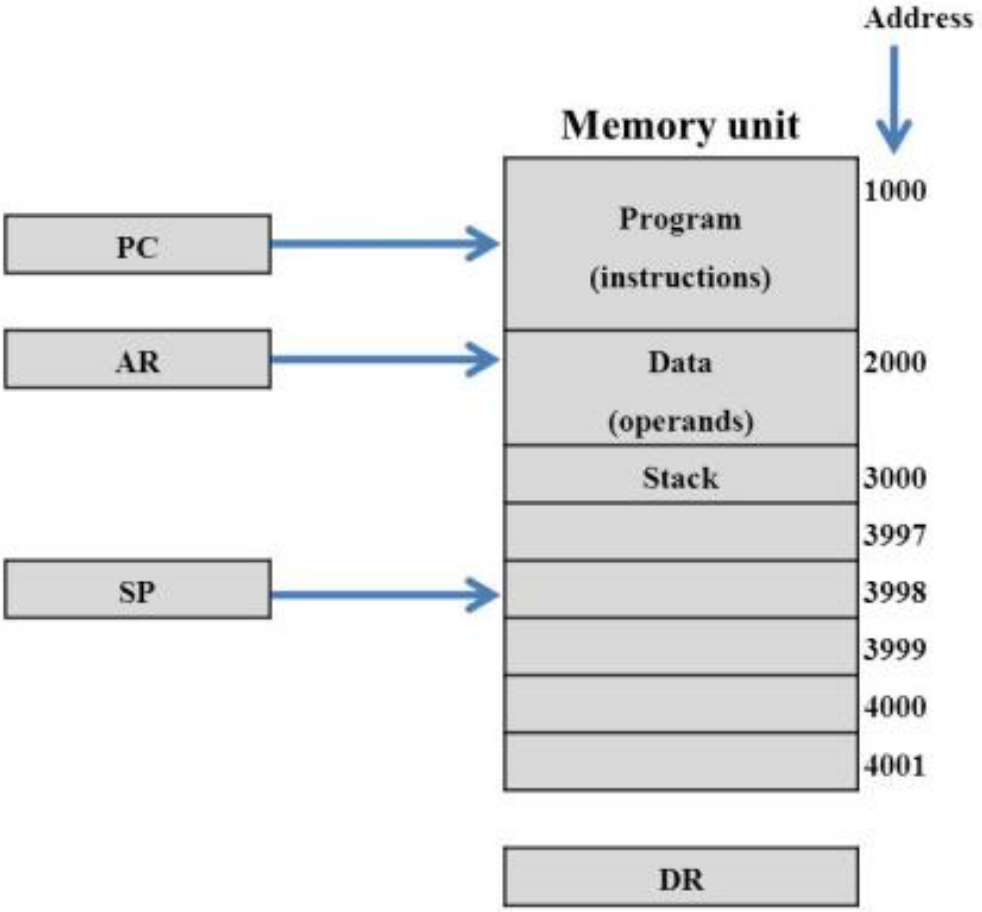
POP

- A new item is deleted from the stack if the stack is not empty.
- If the stack is empty and user tries to delete an item from the stack then it is called Stack Underflow error.

5. WRITE A NOTE ON MEMORY STACK

- A stack exist as a standalone unit or it can be implemented in RAM that attached to CPU.
- The advantage of a memory stack is that the CPU can refer to it without having to specify an address, since the address is always available and automatically updated in the stack pointer

- There are three types of segments are used in the memory stack like **Program, Data and Stack.**



- The Program Counter (PC) points at the address of the next instruction in the program
- The address Register (AR) points at an array of data
- The Stack Pointer (SP) points at the top of the stack
- This type of three segments are connected with the common address bus
- There are two types of operations are used in this type of Memory Stack Operation like PUSH and POP

PUSH

- If the memory is not full then a new item is inserted with a push operation
- If the memory is full and user tries to insert a new item in the memory then it is called Memory Overflow error or Full Memory error.

POP

- A new item is deleted from the memory if the memory is not empty.
- If the memory is empty and user tries to delete an item from the memory then it is called Memory Underflow error or Empty Memory error.

6. SHORT NOTE: POLISH AND REVERSE POLISH NOTATION.

- Evaluating ordinary arithmetic expressions using a computer is difficult, particularly when an Expression consists of parenthesis and brackets.
- In this case an expression has to be scanned from Left as well from the right.
- These problems arise because in an ordinary arithmetic expression operator is placed in between the two operands.
- Such types of ordinary expressions are called Infix expression.
- To overcome this problem the polish mathematician Lukasi showed that a Arithmetic Expression can be represented in prefix or postfix Notation.
- In prefix notation (also known as Polish notation) the operator is placed before the operands while in postfix notation (also known as Reverse polish notation, RPN) the operator is placed after the operands.
- In short arithmetic expression can be represented in three different way as follows.

A+B : INFIX NOTATION
+AB : PREFIX OR POLISH NOTATION
AB+ : POSTFIX OR REVERSE POLISH NOTATION (RPN)

- RPN has number of advantages over infix notation for expressing algebraic formulas.
- Any formula can be expressed without parenthesis.
- It is convenient for evaluating formulas on computer with stack.
- The infix operator has precedence which is undesirable.
- This disadvantage is eliminated by reverse polish notation.
- The main advantage of polish notation is that any formula can be expressed without Parenthesis.
- This is possible by combining reverse polish notation with a stack registers.
- Hence stack is very useful for handling long and complex problems involving arithmetic Expressions.
- This procedure is employed in some electronic calculators and computers.
- In this method The arithmetic expressions first converted reverse polish notation and then operands are pushed Into the stack in the order in which they appear.
- Let us discuss one example for more clarification of this procedure. Consider arithmetic
- Expression.

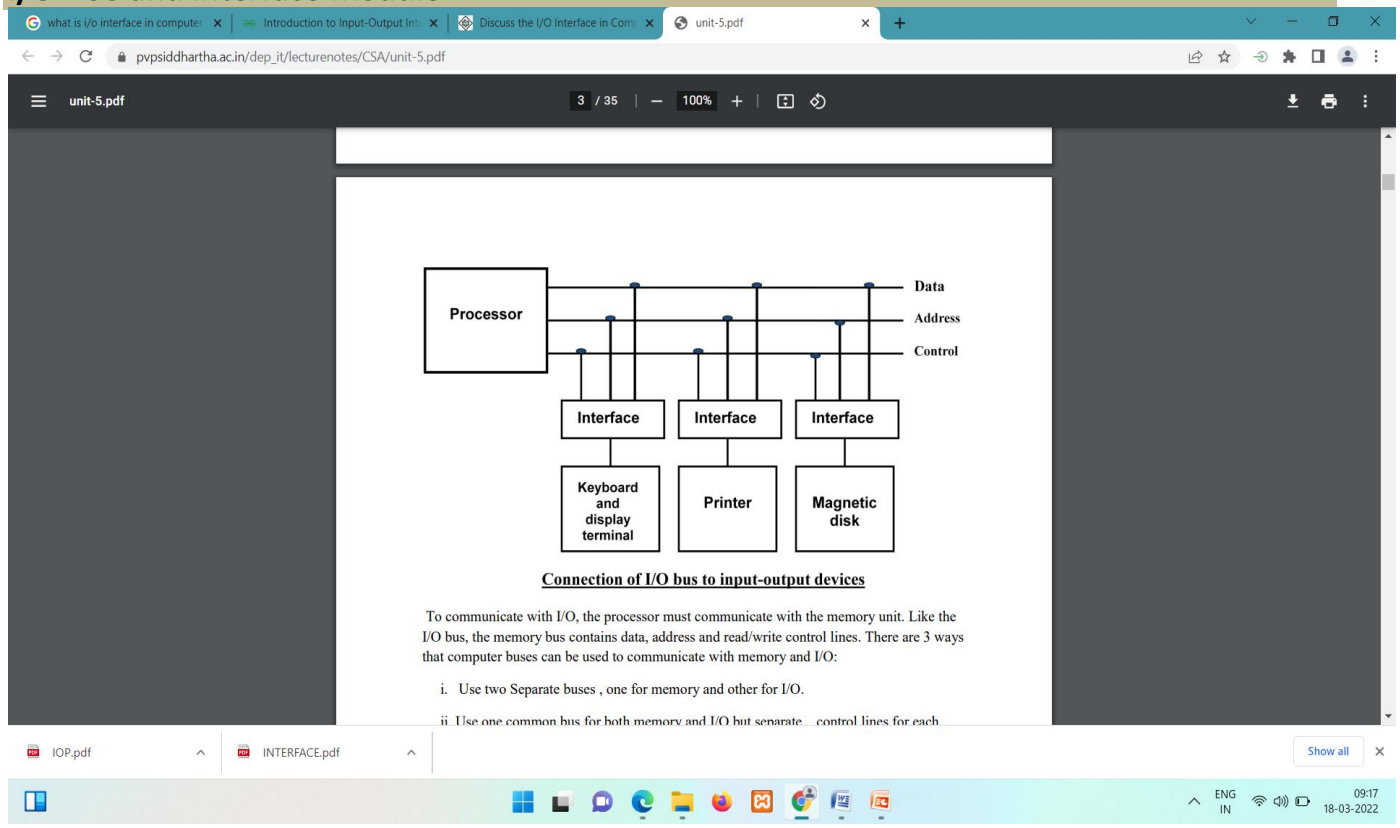
(6*3) + (5*9)
In reverse polish notation it is expressed as
36*95*+

Unit : 5
Input-Output Organization

1. WRITE A NOTE ON IO INTERFACE.

- Input Output Interface provides a method for transferring information between internal storage and external I/O devices.
- Peripherals connected to a computer need special communication links for interfacing them with the central processing unit.
- The purpose of communication link is to resolve the differences that exist between the central computer and each peripheral.

I/O BUS and Interface Module



- I/O BUS and Interface Module It defines the typical link between the processor and several peripherals.
- The I/O Bus consists of data lines, address lines and control lines.
- The I/O bus from the processor is attached to all peripherals interface.
- To communicate with a particular device, the processor places a device address on address lines.
- Each Interface decodes the address and control received from the I/O bus, interprets them for peripherals and provides signals for the peripheral controller.
- It is also synchronizes the data flow and supervises the transfer between peripheral and processor.
- Each peripheral has its own controller.
- An interface receives any of the following four commands –

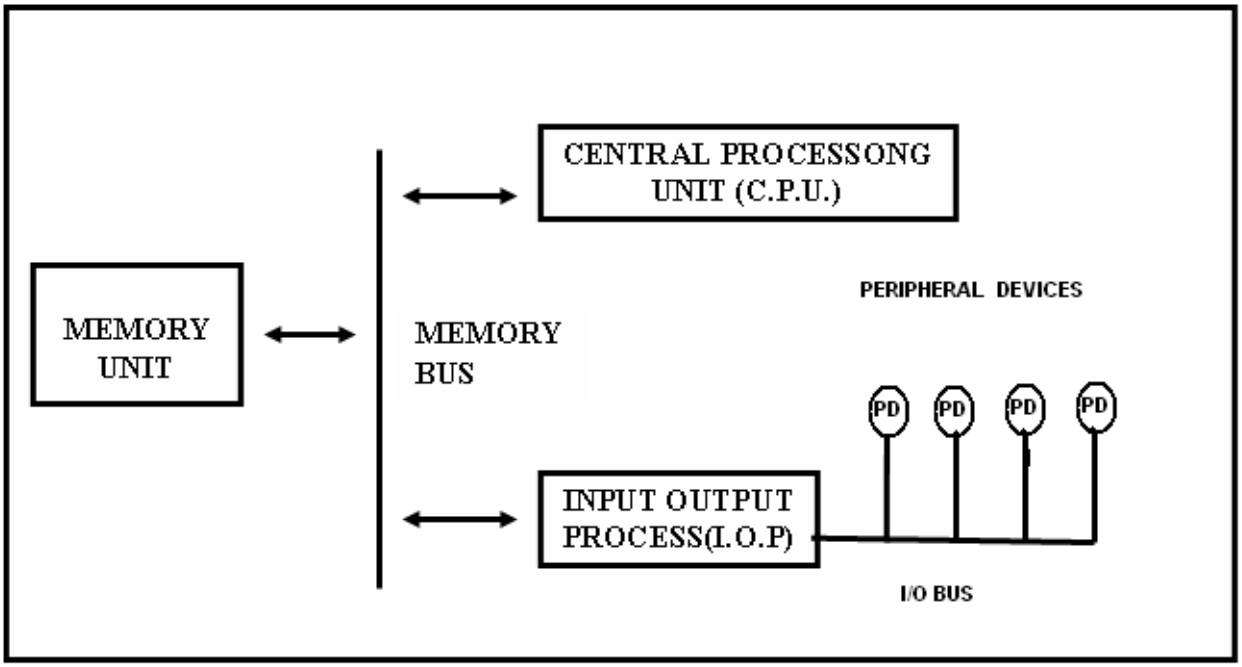
Control	• A command control is given to activate the peripheral and to inform its next task.

	• This control command depends on the peripheral, and each peripheral receives its sequence of control commands, depending on its mode of operation.
Status	• A status command can test multiple test conditions in the interface and the peripheral.
Data output	• A data output command creates the interface counter to the command by sending data from the bus to one of its registers.
Data input	• The data input command is opposite to the data output command. • In data input, the interface gets an element of data from the peripheral and places it in its buffer register.

- To communicate with I/O, the processor must communicate with the memory unit. Like the I/O bus, the memory bus contains data, address and read/write control lines.
- There are 3 ways that computer buses can be used to communicate with memory and I/O:
  - Use two Separate buses , one for memory and other for I/O.
  - Use one common bus for both memory and I/O but separate control lines for each.
  - Use one common bus for memory and I/O with common control lines.

2. WRITE A NOTE ON IOP

- An Input – Output Processor (IOP) is a software program that is used to control input – output operations
- An Input – Output Processor (IOP) controls sending and receiving data between Input – Output Process
- An Input – Output Processor (IOP) finds and corrects sending and receiving errors between Input – Output Process
- An Input – Output Processor (IOP) assembles and disassembles messages between Input – Output Process
- An Input – Output operation is performed by the CPU



- In figure, we can show that, Memory bus contains CPU and IOP for communication
- Input / Output bus (I / O) Bus is connected with the Input – Output Processor (IOP) with the help of Peripheral devices (PD).
- An Input – Output Processor (IOP) communicates with the peripheral device through the I/O bus and with the memory through the memory bus.
- When IOP needs to transfer data to or from memory, it starts a memory cycle from the CPU and then it transfers the data to or from memory.
- In computer, CPU is the master processor and IOP is slave processor.
- CPU sends information by using IOP path.
- IOP contains status word into that memory location.
- CPU checks the status word if the status word is correct then CPU inform to the IOP about sending information.
- After that, IOP reads and executes commands from the memory that located at the specified location.
- When the IOP completes the Input Output data transfer then it informs to the CPU for data transfer is completed.

3. WRITE A NOTE: ON VARIOUS TYPES OF BUSES.

- A computer system consist of number of internal & external components .
- This components are physically interconnected & communicate with each other via wires across computer system.
- These wires are known as buses.
- Theses buses are essential to run a computer system.
- A bus is a group of lines/wires which carry computer signals.
- A bus is the means of shared transmission.
- Data can be transferred from one computer system location to another (between different I / O modules, memory, and CPU).

- The bus is not only cable but also hardware (bus architecture), protocol, program, and bus controller.

FUNCTIONS OF BUSES:

- Data sharing
- Addressing
- Generate control signals
- Share system time
- Provide power supply

Types of buses:

Internal Bus

- Internal bus is placed inside the processor.
- Internal bus is used to connect internal components of computer system such as processor, RAM , hard disk etc.

External Bus

- External bus is placed outside the processor.
- It is used to connect the external part connected with computer like keyboard, mouse, monitor, printer etc
- External bus is used to send information between one parts of computer into another part of computer

System Bus

- System bus is used to connect the processor with main memory or graphic card this bus is used.
- It contain 3 buses: Address bus, Data bus and control bus

Data Bus

- Data bus is used to send data from one parts of computer into another part of computer
- Data bus is used to send data to the processor, memory and other parts of computers
- Data bus is used to connect the processor, memory and other parts of computers for communicate with each other for transferring the data

Address Bus

- Address bus is used to send address from one parts of computer into another part of computer when read or right operation required to perform by processor & receive control signal.

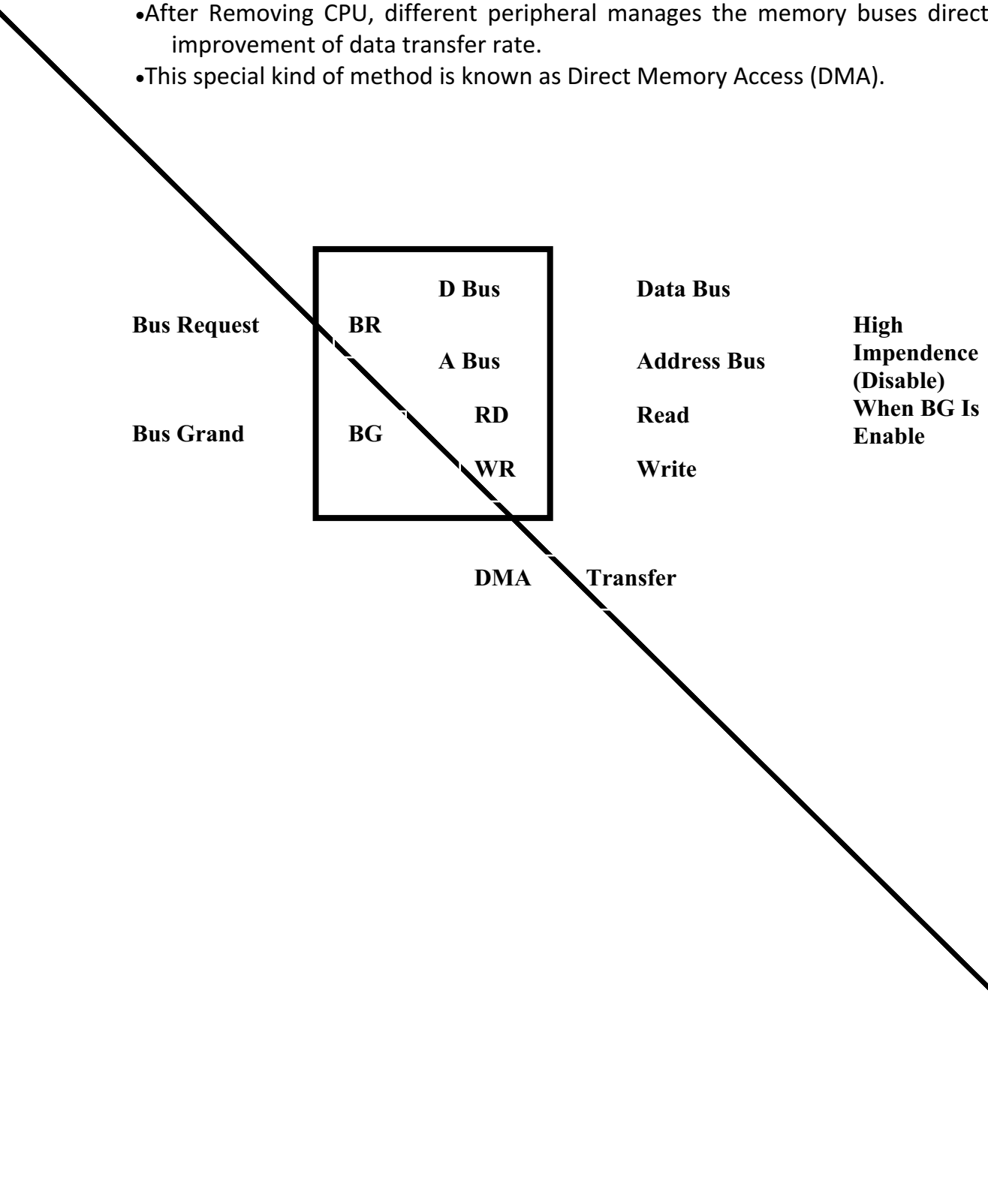
Control Bus

- Control bus is used to control the data and information of the data bus and address bus.
- It is hold by processor itself and used by processor itself.

4. WRITE A NOTE ON DMA.

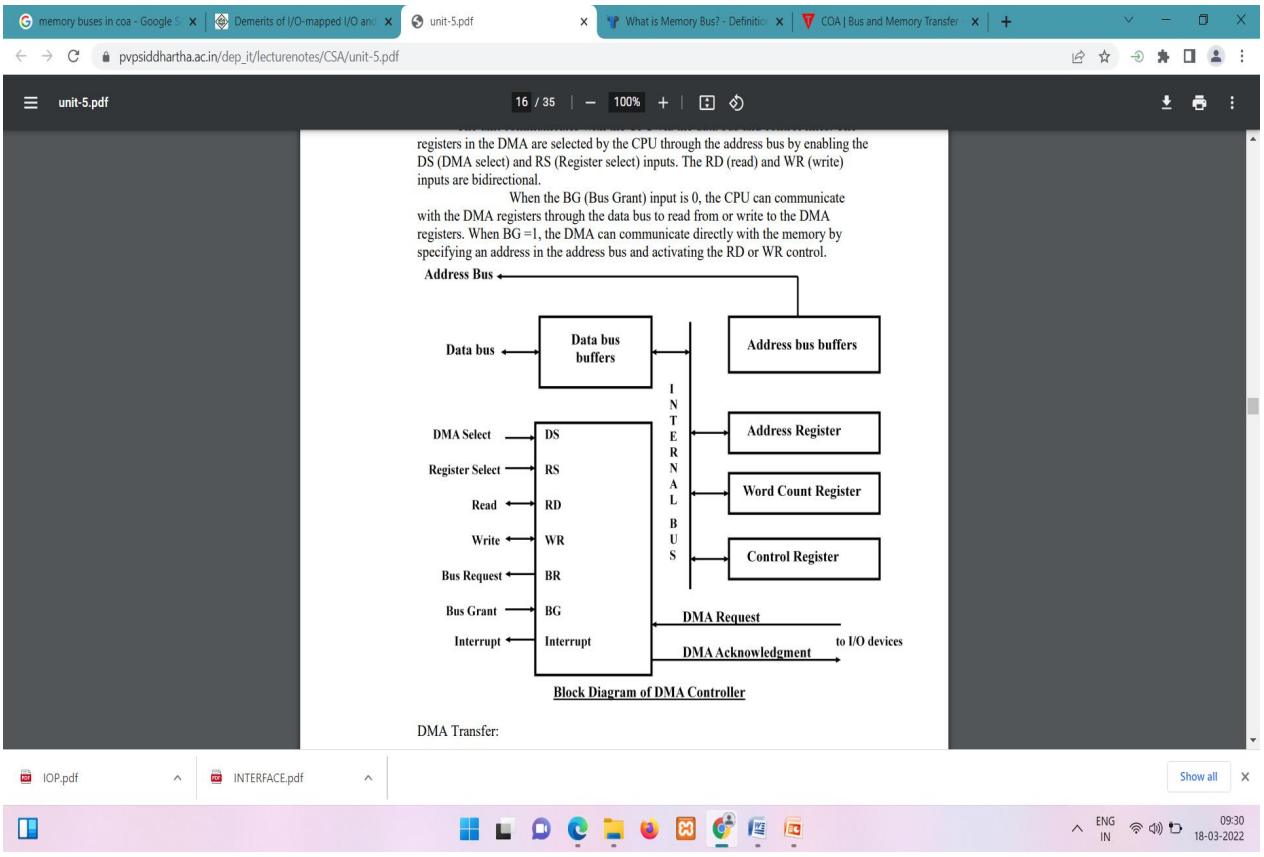
- DMA means Direct Memory Access.
- DMA is a process of communication or data transfer that controlled by an external peripheral.
- When the DMA is used in the memory, the microprocessor controlled the data transfer.
- For 8085 microprocessor, HOLD and HLDAC signal use for this type of data transfer.
- Direct -memory access is an input output technique for fast data transfers.

- Direct memory access (DMA) allows hardware system for reading or writing process of the central processing unit.
- Many hardware systems use DMA including Disk drive controllers, graphics cards, network cards and sound cards.
- The data transfer speed of Central processing unit is limited between secondary memory and Main memory.
- This problem can be resolved by removing CPU from the path.
- After Removing CPU, different peripheral manages the memory buses directly for the improvement of data transfer rate.
- This special kind of method is known as Direct Memory Access (DMA).



5. SHORT NOTE: DMA CONTROLLER.

- DMA means Direct Memory Access.
- DMA is a process of communication or data transfer that controlled by an external peripheral.



- In this type of DMA process, When Input device needs to transfer data to or from memory, it request to the DMA controller by setting DMA Bus Request Input to 1 and DMA Bus Grant Input to 1.
- Here 1 means Enable and 0 means Disable
- Bus Request (BR) is used to pass the request to the CPU.
- After that, CPU accept the request from the Bus Request (BS) signal when the Bus Grant (BG) is Enable.
- After that, CPU can read and write the request by using Read (RD) and Write (WR) signal.
- After that CPU gives address into that request by using Address Bus (A BUS) and also give data into that request by using Data Bus (D BUS).
- After that CPU can transfer to that request from one place into another by using DMA controller.
- When data transfer is completed then Bus Request and Bus Grant signal becomes disable.
- DMA method allows the Input / Output devices to directly communicate with the main memory of a computer.
- DMA method directly reads and writes data from the memory by using memory bus
- DMA is a technique for moving data directly between main memory and peripheral of computer without need of CPU

Prepared By: Prof. Meghna Bhatt

- DMA Controller contains 3 types of register like Address Register, Word Count Register and Control Register.

ADDRESS REGISTER

- Address register contains the address of memory where the data from the input device needs to be stored or from where the data for the output device needs to be fetched
- CPU gives this address to the data bus.
- The address is transferred to the Address Register from the Data Bus through the Internal Bus.
- Address in this register is incremented after transferring each word to or from memory

WORD COUNT REGISTER

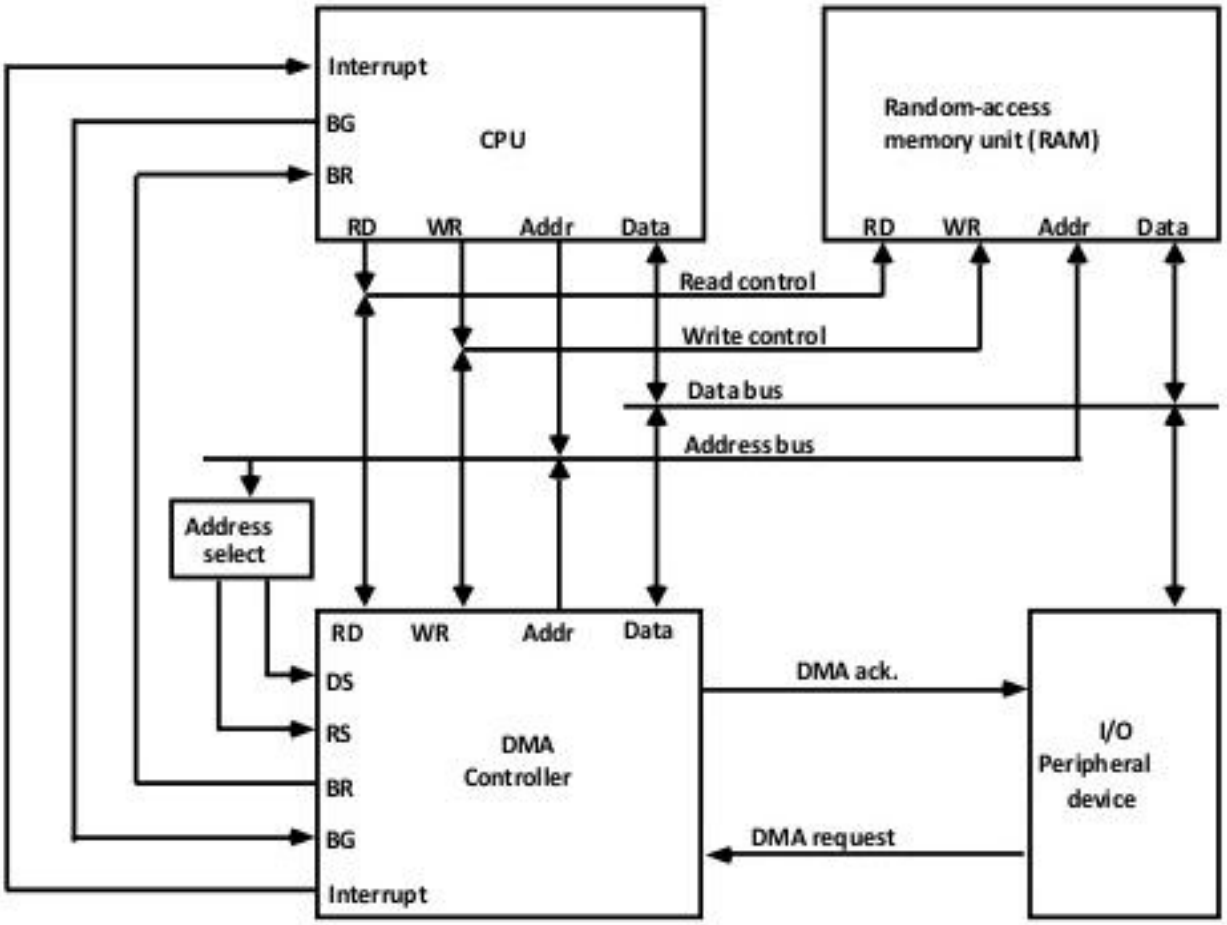
- Word Count Register stores the number of words the need to be read from or written into memory
- It is decremented after each word transfer.
- When the value in this register becomes zero, it is assumed that data transfer has completed

CONTROL REGISTER

- The control register specifies the mode of data transfer like it is a read operation or it is a write operation
- CPU reads the data through the data bus after setting to Read Input to 1
- CPU writes the data through the data bus after setting to Write Input to 1

6. WHAT IS DMA? HOW DMA TRANSFER DATA?

- DMA is a process of communication or data transfer that controlled by an external peripheral.
- DMA is a technique for moving data directly between main memory and peripheral of computer without need of CPU



- When Input device needs to transfer data to or from memory, it request to the DMA controller by setting DMA Bus Request Input to 1 and DMA Bus Grant Input to 1.
- Here 1 means Enable and 0 means Disable
- Bus Request (BR) is used to pass the request to the CPU.
- After that, CPU accept the request from the Bus Request (BS) signal when the Bus Grant (BG) is Enable.
- After that, CPU can read and write the request by using Read (RD) and Write (WR) signal.
- After that CPU gives address into that request by using Address Bus (A BUS) and also give data into that request by using Data Bus (D BUS).
- After that CPU can transfer to that request from one place into another by using DMA controller.
- When data transfer is completed then Bus Request and Bus Grant signal becomes disable.
- DMA method allows the Input / Output devices to directly communicate with the main memory of a computer.

- DMA method directly reads and writes data from the memory by using memory bus.

7. WHAT IS BURST TRANSFER AND CYCLIC STEALING MODE. OR EXPLAIN MODE OF DATA TRANSFER USED BY DMA.

- DMA use burst transfer or cycle stealing mode to transmit data from peripheral to main memory that describe bellow.

Burst Transfer Mode

- Burst transfer is the fastest way to transfer memory.
- It is the DMA data transfer technique in which number of data words are transferred continuously until whole data is not transferred.
- Data transfer Continues until whole data is not transferred.
- So that it is very fast data transfer technique and is used to transfer data for fast speed devices.
- The DMA controller assumes that source and destination memory address/IO port can transfer and accept the data as quickly as the DMA controller can produce them.
- So after the controller is set up and the CPU has released control of the address and data buses, the entire block of memory is copied to the destination as a single contiguous block.
- After transmit whole words the control of buses will transfer to CPU.

Cycle Stealing Mode

- Cycle Stealing mode is similar to Burst Transfer mode, but instead of the data being transferred all at once, it is transferred one byte at a time.
- It is the data transfer technique in which one data word is transferred and then control is returned to CPU.
- The DMA controller, after transferring one byte of data, releases control of the system buses by sending a bus grant signal through the control bus, lets the CPU process an instruction and then requests access to the bus by sending the bus request signal through the control bus and then transfers another byte of data.
- This keeps going on until all the data has been transferred.
- The transfer rate is slower but it prevents the CPU from staying idle for a long period of time.