— Computer System Architecture —

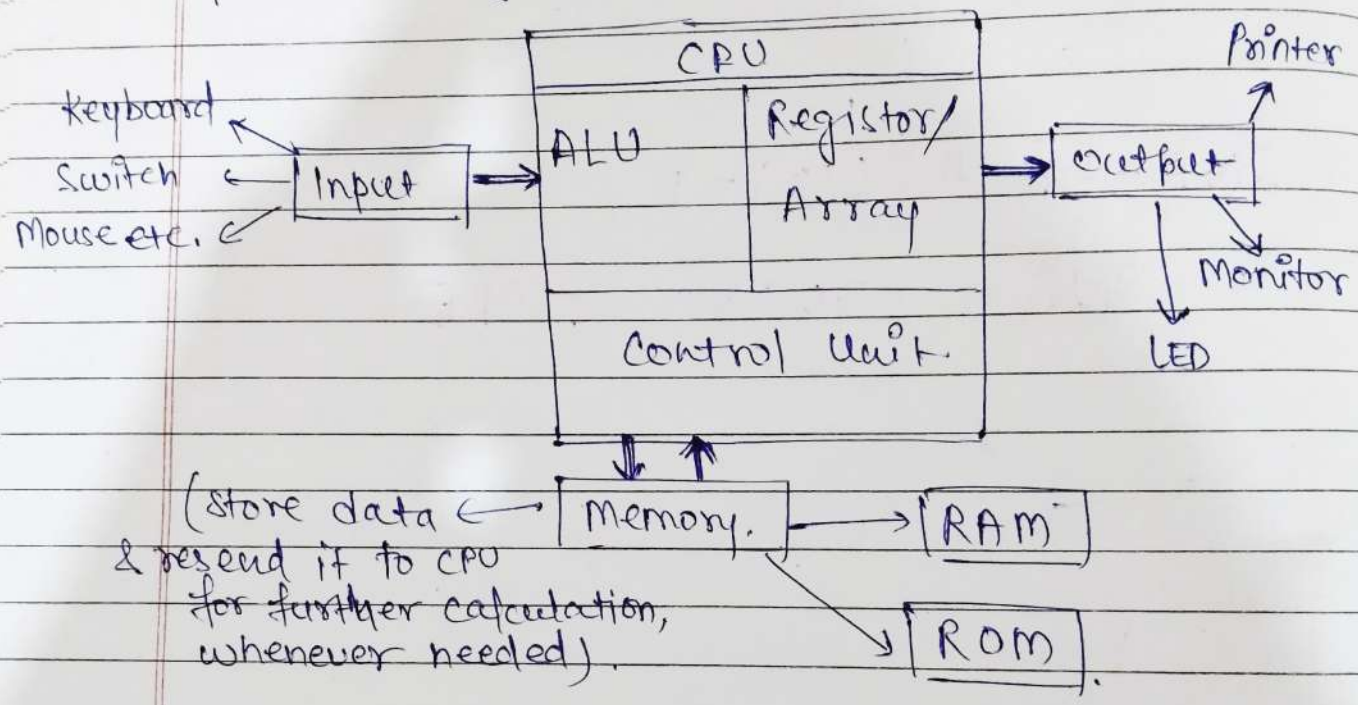
Chapter - 1st.

24-Dec.

→ Digital computers -

① Digital computers have microprocessor as its CPU [central processing unit].

② CPU combined with memory & I/O device forms a digital computer.

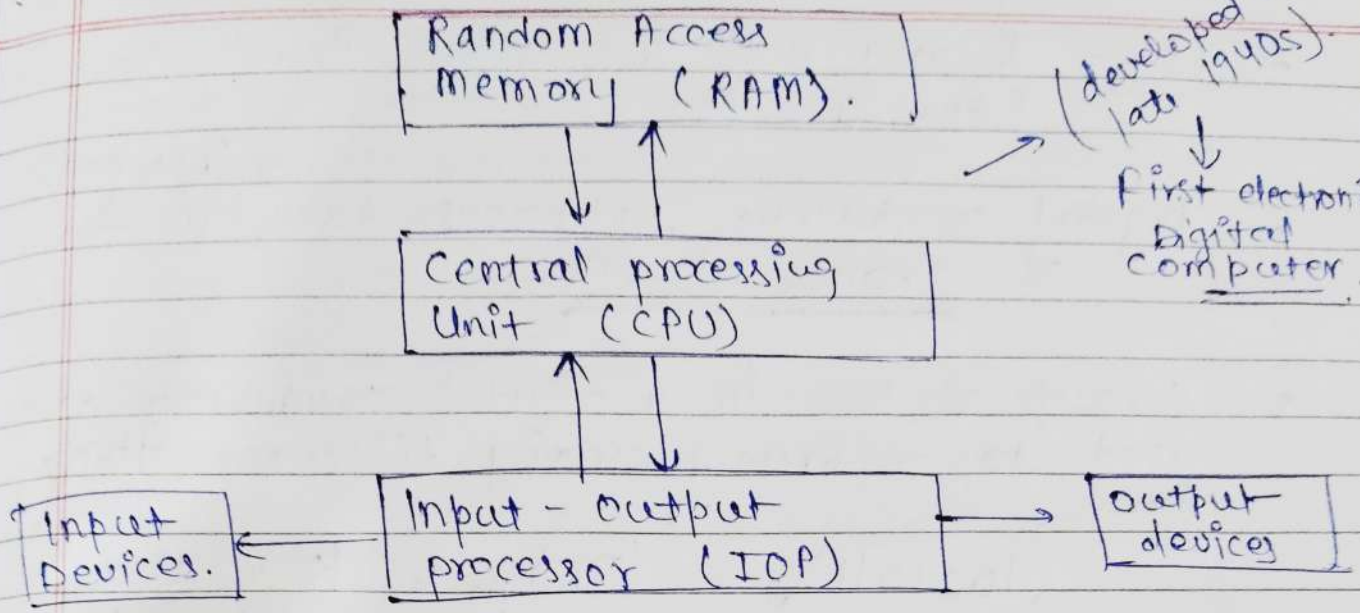


→ Simply,
This all combination of CPU (microprocessor), Input, output and memory is k/a Digital computers.

- It is a digital system that performs various computational tasks. Digital computers uses the binary number system which has 2 digits 0 & 1.

→ Block Diagram of Digital computers -

(3-parts of Hardware)



- CPU —
① ALU — for manipulating data.
② Registors — for storing data.
③ CU — fetching & execution of instructions given.

- IOP — electronic circuit for comunicating and controlling the transfer of information b/w computer & the outside world.

→ Difference between Analog and Digital Computer —

An analog computer operates on continuously varying data while digital computer performs operations on discrete data.

Note —
Discrete values — can say real nos. (like 0 to 9) or (0 & 1) or (alphabets).

"The order to succeed, your desire for success should be greater than your fear of failure." —Narendra Modi

Date __/__/__ Page No.

- **Bit** – Binary digit is called bit. (Ex – 0 & 1).
- Digital computers represent data (or information) in form of groups of bits.
- Group of bits in a digital computers are used to represent many different things.

Ex –

$(1001011)_2 = (75)_{10} = K$ = also control code. (1)

Two functional entities

Hardware	Software.
consists of all electronic components and electromechanical devices that comprise the physical entity of the device.	consists of instructions and data that the computer manipulates to perform various data-processing tasks.

Program – Sequence of instructions for the computer. (2)

Operating System – System software consists of collection of program which make more effective use of computer, one such system software package is called OS.

They are system programs (e.g. compiler) and are far different from application programs (e.g. high level language program).

Date ___/___/_____ Page No.: _____

→ Computer organisation
→ Computer design
→ Computer architecture.

LOGIC GATES

- Digital systems (circuit) are said to be constructed by using logic gates. These gates are the AND, OR, NOT, NAND, NOR, XOR and XNOR gates.

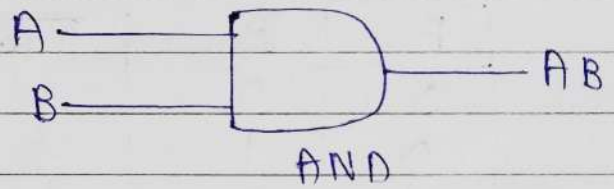
① AND Gate — (can have more than 2 inputs).

It is an electronic circuit that gives a high output (1) only if all its inputs are high. A dot (•) is used to show the AND operation i.e., A.B.

2 Input AND gate

A	B	A.B (Y)
0	0	0
0	1	0
1	0	0
1	1	1

(Truth Table).



Boolean expression —

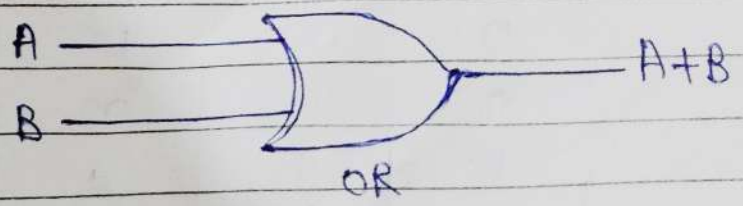
$A \cdot B\ (Y) = A \cdot B = AB$

② OR Gate —

gives a high output (1) if one or more of its inputs are high. A plus (+) is used to show the OR operation.

2 input OR gate

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1



$Y\ (A+B) = A + B.$

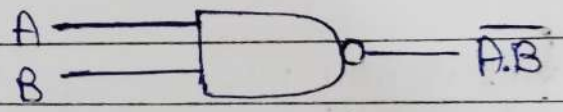
(3) NOT Gate — (1 input → 1 output).
produces an inverted version of the input at its output. It is also k/a inverter. shown to us by $\bar{A}$ or A'.

NOT Gate	
A	$\bar{A}$
0	1
1	0

$Y = A' = \bar{A}$.

(4) NAND Gate —
NOT-AND Gate equals AND gate followed by NOT Gate. Small circle in symbols represent inversion.

2 Input NAND Gate		
A	B	$\overline{A \cdot B} = A'B'$
0	0	1
0	1	1
1	0	1
1	1	0

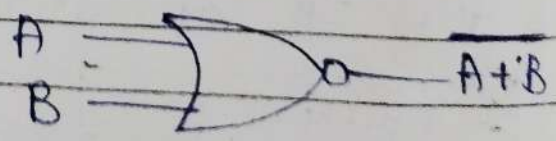


NAND.

$Y = \overline{A \cdot B} = (A \cdot B)'$

(5) NOR Gate —
OR gate followed by NOT Gate.

2 Input NOR gate		
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0



NOR.

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

"The butterfly counts not months but moments, and has time enough."

Date __/__/__

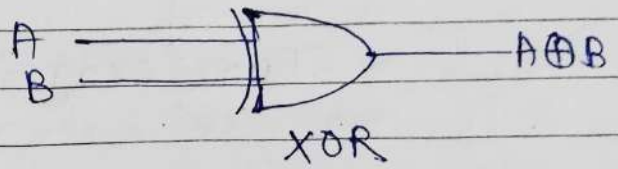
→ (commonly available with 2 inputs).

→ It is commutative as well as associative.

⑥ XOR Gate —

'Exclusive-OR' gate is a circuit which will give a high output if either, but not both, of its two inputs are high. An encircled plus sign (⊕) is used to show XOR operation.

2 INPUT XOR Gate		
A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

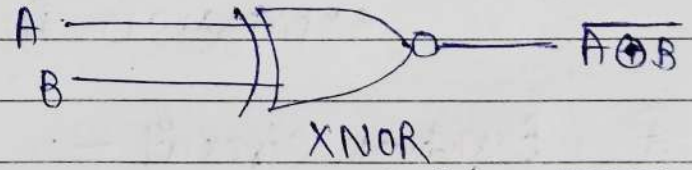


$Y = A \oplus B$.

⑦ XNOR Gate —

'Exclusive-NOR' gate does the opposite to XOR gate. It will give a low output if either, but not both, of its ~~tow~~ two inputs are high.

2 Input XNOR gate		
A	B	$\overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1



$Y = \cancel{A \oplus B}$ $\overline{A \odot B}$

Note — ⓐ NAND and NOR gates are called universal functions since with either one AND and OR functions and NOT can be generated.

ⓑ Universal functions or Universal Gate.

ⓒ Boolean Expression — Relationship between inputs and outputs in logic circuit.

Signals — Binary information is represented in digital computers by physical quantities called signals.

↓

Voltage

↓

Two recognizable state

↙ ↘

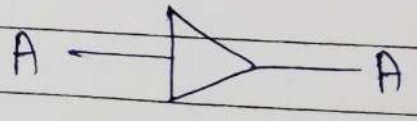
for 1 (3 volts) | for 0 (0.5 volt).

The manipulation & processing of Binary information is done by logic ~~gates~~ circuits called gates.

Truth Table — representation of boolean expression in tabular form.

(8) Buffer Circuit —

does not produce any logic function.



$y = A$.

It is used for power amplification.

The main purpose of the buffer is to drive other gates that requires a large amount of power.

Date __/__/____ Page No. _____

BOOLEAN ALGEBRA (BA).

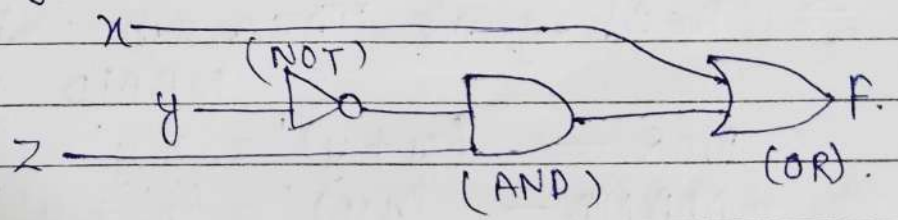
Boolean Algebra deals with binary variables & logical operations. Variables are A, B, x, y, z etc.

Example – $F = x + y'z$. (8 combinations).

Note – To represent a boolean function in the truth table we need a list of '2^n' combinations of 'n' binary variables.

→ Logic diagram – composed of AND, OR and inverter gates.
Variables are taken as input & variable symbol of the function is taken as output of the circuit.

Example – $F = x + y'z$



→ Purpose of BA –
to facilitate the analysis & design of digital circuits.

→ Boolean expression – By manupulating a Boolean expression according to boolean Algebra rules, we obtain a simpler expression that will requires fewer gates.

Table – Basic Identities –

(1) $x + 0 = x$

(2) $x \cdot 0 = 0$

(3) $x + 1 = 1$

(4) $x \cdot 1 = x$

(5) $x + x = x$.

(6) $x \cdot x = x$

(7) $x + x' = 1$

(8) $x \cdot x' = 0$

(9) $x + y = y + x$

(10) $xy = yx$

(11) $x + (y + z) = (x + y) + z$. →

(12) $x(yz) = (xy)z$.

(13) ~~x (x+z~~ $x(y + z) = xy + xz$.

(14) $x + yz = (x + y)(x + z)$.

(15) $(x + y)' = x'y'$. } (DeMorgan's theorems.)

(16) $(xy)' = x' + y'$.

(17) $(x')' = x$.

Example –

$$\underbrace{AB' + C'D}_{x} + \underbrace{AB' + C'D}_{x} = x.$$

$$\Rightarrow \quad AB' + C'D.$$

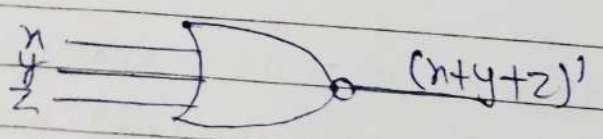
→ DeMorgan's Theorem – deals with NOR & NAND gates.

NOR – $(x + y)' = x'y'$.

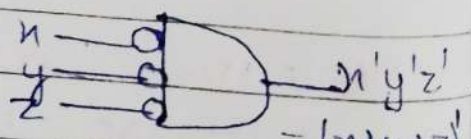
NAND – $(xy)' = (x' + y')$.

Graphic symbols –

NOR gate –



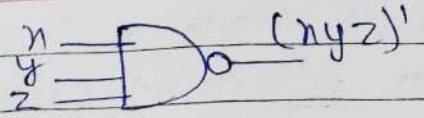
(OR – Invert)



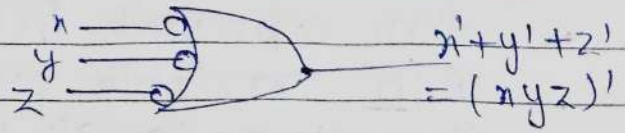
(Invert – AND)

NAND gate –.

"I sincerely believe that Indians have the ability to compete with the best in the world." – Dhirubhai Ambani



(AND-invert)



(invert-OR)

→ Complement of a Function — interchanging 1's & 0's in the values of F in the truth table.

function in algebraic form ⇒ complement derived by means of De morgan's theorem.

This is done by changing all OR operations to AND operations and all AND operations to OR operations & then complementing each individual letter variable.

Example —

$$F = AB + C'D' + B'D$$
$$F' = (A'+B')(C+D)(B+D').$$

MAP SIMPLIFICATION

Pictorial (arrangement) representation of boolean function. truth table of

- The map method provides a simple, straightforward procedure for simplifying Boolean ~~Algebra~~ expressions.

Also known as the Karnaugh or K-map.

Min term — Each combination of the variables in a truth table is called min-term.

- when expressed in a truth table a function of 'n' variable produces 2^n minterms, equivalent to 2^n binary numbers obtained from n bits.

Ex- $f(x, y, z) = \Sigma(1, 4, 5, 6, 7)$.

minterm ~~for~~ that produces 1 for the function, in decimal equivalent.

Map is a diagram made up of squares, with each square representing one minterm.
The squares corresponding to minterms that produce 1 for the function are marked by 1 and the other are marked by a 0 or are left empty.

- The number of squares in a map of n variables is 2^n.

⇒ → while pairing, avoid overlapping.
(by ma'am).

- Two types of simplication in k-map -
1) Sum-of-products form.
2) Product-of-sums simplification.

→ The squares not marked by 1 represent the minterms that produce 0 for the function. If we mark the empty squares with 0's and combine them into groups of adjacent squares, we obtain the

"I sincerely believe that Indians have the ability to compete with the best in the world." - Dhirubhai Ambani

complement of function, F'. Taking complement of F' produces an expression for F in product-of-sums form. ~~The best way~~.

Example —

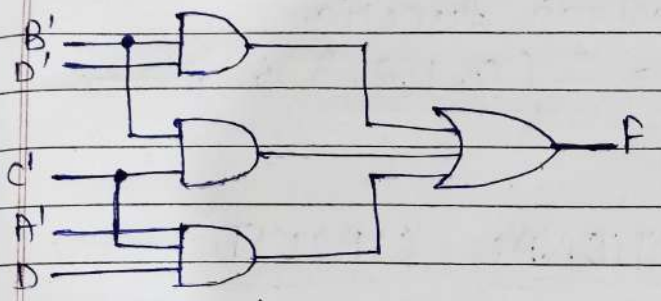
$$F' = AB + CD + BD'$$

$$F = B'D + B'C + A'C'D.$$

$$\Rightarrow F = (A'+B')(C'+D')(B'+D).$$

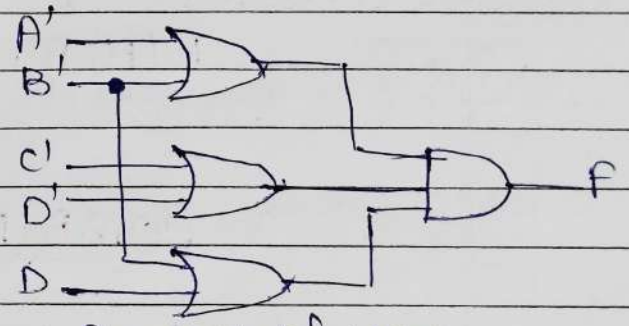
1	1	0	1
0	1	0	0
0	0	0	0
1	1	0	1

Logic Diagram —



Sum of products —

$F = B'D + B'C' + A'C'D.$



Product of sums —

$F = (A'+B')(C'+D')(B'+D).$

→ **Don't Care Condition** —

When it does not matter if the function produces 0 or 1 for a given minterm.

Minterms that may produce either 0 or 1 for the function are said to be don't-care conditions and are marked with an X in the map.

Note - In addition, an X need not be used at all if it does not contribute to the simplication of the function.

Example —

$f(A, B, C) = \Sigma(0, 2, 6)$.
$d(A, B, C) = \Sigma(1, 3, 5)$.

A	1	X	X	1
	0	X	0	1

Thus, the expression -

$$f = A' + BC'$$

represents the Boolean function

$$f(A, B, C) = \Sigma(0, 1, 2, 3, 6).$$

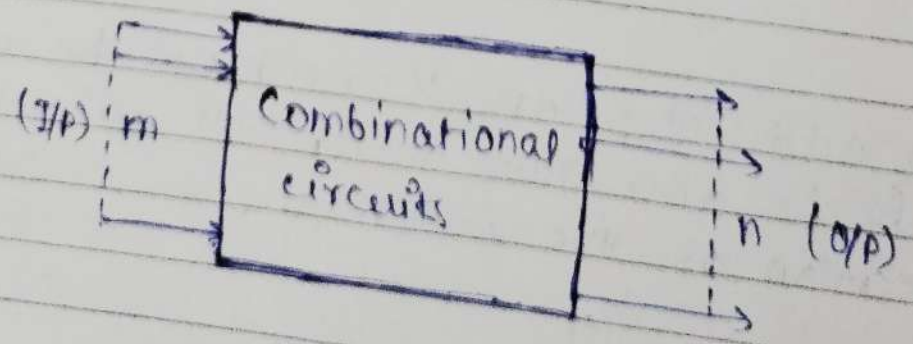
COMBINATIONAL CIRCUIT.

When logic gates are connected together to produce a specified output on certain specified combination of input variables, with no memory involved (like buffer), then it is combinational circuit.

- O/P depends only on present I/P

$\Rightarrow$ $[O/P = f(I/P)]$.

- May have m- binary I/P and n- binary O/P.



Date ___/___/_____ Page No. _____

Application—.

→ Used in ALU (Arithametic and logical unit). means computation, data transmission, code converters etc.

Combinational circuit.

- ALU
  - → Adder
  - → Subtractor
  - → Comparitors
- Data Transmission
  - → MUX → DeMUX
  - → Encoder & Decoder
  - (MUX - multiplexer).
- Code Converters.
  - → Binary
  - → BCD
  - → Gray.

→ The 'n' binary input variables come from an external source, the 'm' binary output variables go to an external destination, & in b/w their is an interconnection of logic gates.

Purpose—.

employed in digital computers for generating binary control decisions and for providing digital components required for data processing.

To demostrate the design of combinational circuits.

Two simple arithmetic circuits (basic building blocks for the construction of more complicated arithmetic circuits.)

→ Half-adder → full-adder.

→ **Half-adder** – (2 input → 2 outputs).

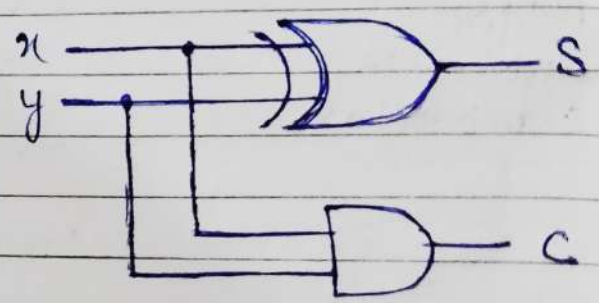
A combinational circuit that performs the arithmetic addition of two bits is called a half-adder.

- Two input variables of half-adder are called Augend and Addend bits.
- The output variables the sum and carry.

→ Truth Table –

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

→ Logic diagram –



→ Boolean Expression –

$$S = x'y + xy' = x \oplus y$$
$$C = xy.$$

→ **Full-Adder** – (3 inputs → 2 outputs)

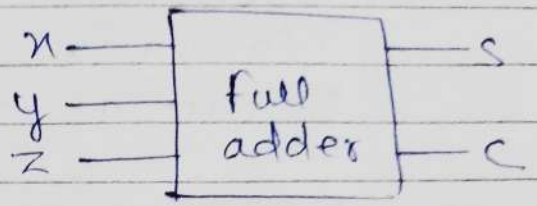
A full-adder is a combinational circuit that forms the arithmetic sum of three input bits.

- x & y, represent the two significant bits to be added.
- z represents the carry from the previous lower significant position. (C_{in} – carry input).

→ Truth Table —

Inputs			Outputs	
x	y	z	C (carry)	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

→ Block diagram —



→ ~~logic~~ Boolean expression —

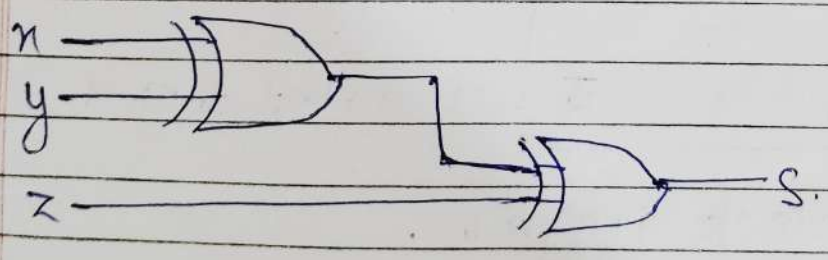
$$S = x \oplus y \oplus z.$$

$$C = xy + (x'y + xy')z.$$

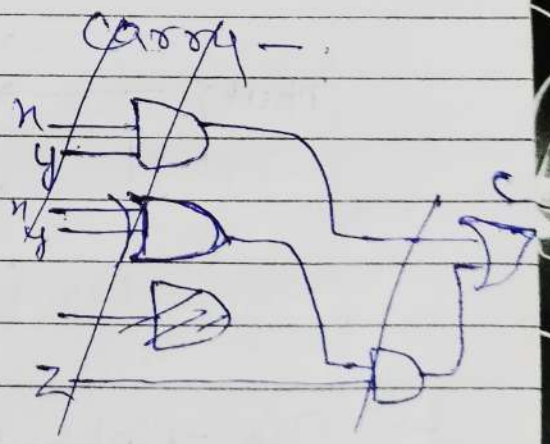
$$\Rightarrow C = xy + (x \oplus y)z.$$

→ Logic Diagram —

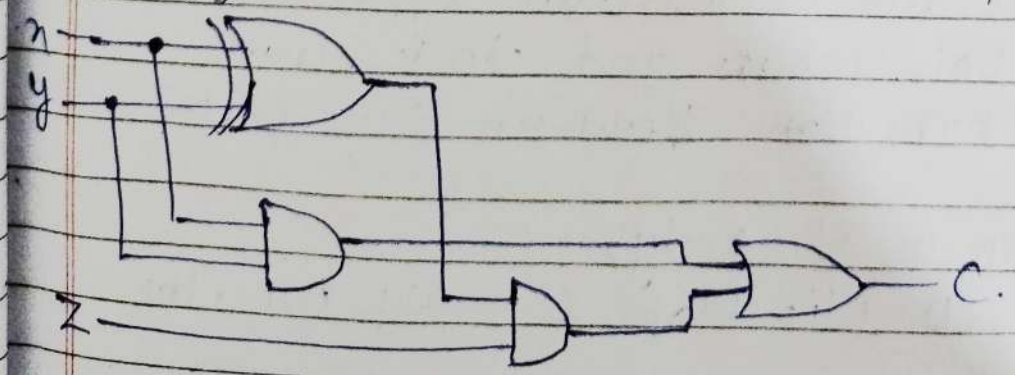
Sum —



Carry —



Carry —



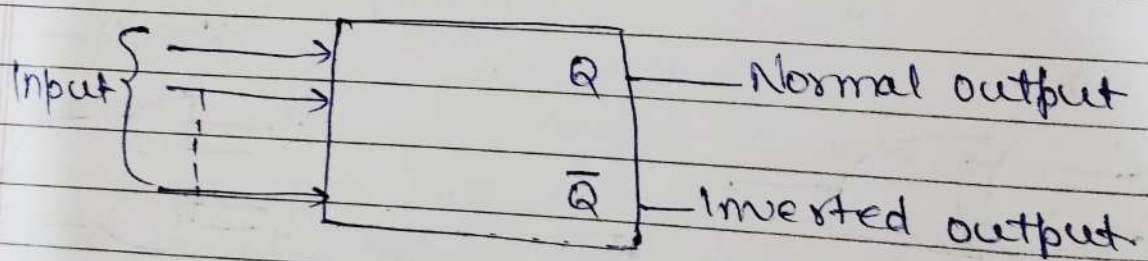
FLIP - FLOPS

→ Flip - Flop is a memory element which is capable of storing one - bit of information and it is used in clocked sequential circuits.

→ A flip - Flop has 2 outputs, one for normal value and other for the complement value of the bit stored in it.

→ A flip-flop can maintain binary state indefinitely (as long as power is supplied to the circuit) until directed by an input signal to switch states.

→ A flip-flop is also known as bistable multivibrator.



(Block diagram).

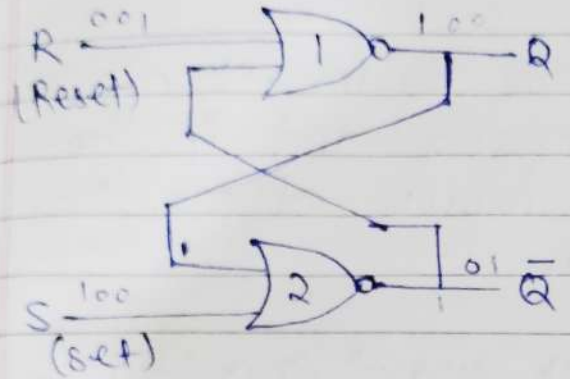
→ flip - flops are of different types depending on how their inputs and clock pulses cause transition between two states.

→ Basically, Flip - Flop is of 4 types —
SR, JK, D-Flip Flop & T ~~fill~~ flip-flop.

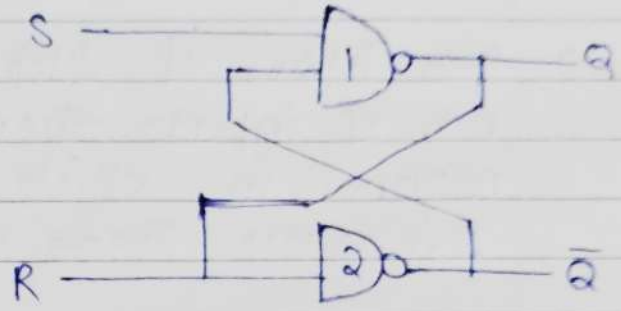
"I sincerely believe that Indians have the ability to compete with the best in th

• Basic Flip flop circuit or latch —

can be constructed from 2 NAND gates or two NOR gates.



(Using NOR gates)



(Using NAND gates)

→ Truth Table —

S	R	Q	$\bar{Q}$
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1
1	1	0	0

(invalid) since it violates the condition of FF

Truth Table —

S	R	Q	$\bar{Q}$
1	0	0	1
1	1	0	1
0	1	1	0
1	1	1	0
0	0	1	1

(invalid)

This type of flip-flop is sometimes called a direct coupled ~~as~~ f RS flip-flop or SR latch.

Date ___/___/___ Page No.:

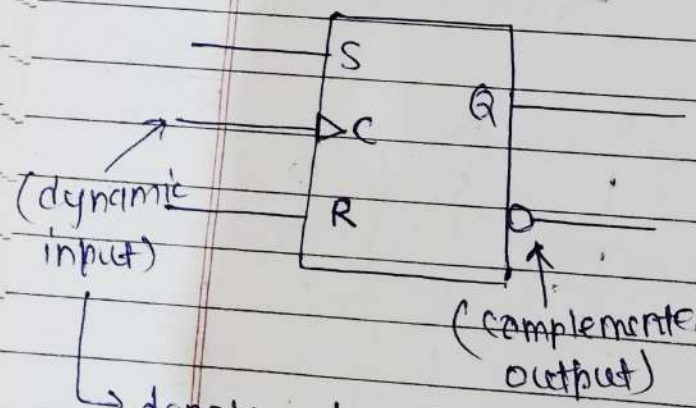
2nd Definition.— The storage elements employed in clocked sequential circuits are called flip-flop.

It is a binary cell store 1 bit info.

→ ~~No.~~ Types of flip-flops ~~is in~~ depends on no. of inputs they possess and in the manner in which the inputs affect the binary ~~state~~ state.

→ **S-R Flip Flop**— (clock signal should be 1/on)

→ Graphic symbol—



→ denotes the fact that the flip-flop responds to a positive transition (from 0 to 1). of the input clock signal.

→ Characteristic table —

S	R	Q(t+1)	
0	0	Q(t)	No change
0	1	0	Clear to 0
1	0	1	Set to 1
1	1	?	Indeterminate (Unpredictable)

(and may go to either 0 or 1, depending on internal timing delays that occurs with the circuit.)

Q(t) → (present state)
Q(t+1) → (next state).

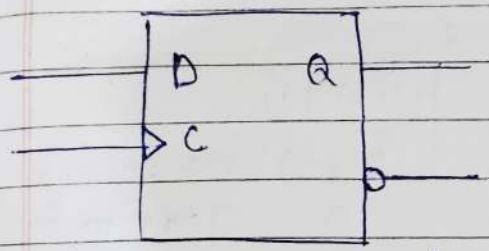
Drawback —

This indeterminate condition makes the SR flip-flop difficult to manage and therefore it is seldom used in practice.

→ **D-flip flop** — (Modification of SR flip-flop.).

(Data).

An SR is converted to D-flip-flop by inserting an inverter b/w S & R and assigning the symbol D to single input.



(graphic symbol).

D	Q(t+1)	
0	0	clear to 0
1	1	set to 1.

(characteristic table)

Relation — $\boxed{Q(t+1) = D}$.

when clock signal goes through transition from 0 to 1.

Advantage —. Have only one input (excluding C).

Disadvantage —. its characteristic table does not have a "no change" condition, $Q(t+1) = Q(t)$.

It can be accomplished either by disabling the clock signal or by feeding the output back into the input.

→ **JK Flip-Flop** —.

(refinement of SR flip-flop).

Indeterminate condition of SR flip flop is defined in JK type

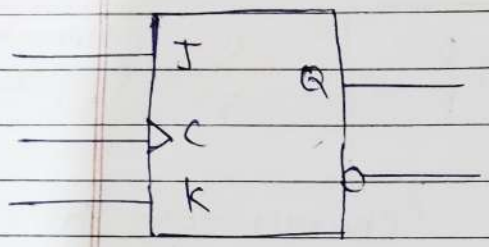
"Your desire for success should be greater than your fear of failure." —Narendra Modi

Date ___/___/___

Inputs J & K behave like inputs S & R to set & clear the flip-flop respectively.

Advantage-
when inputs J and K are both equal to 1, a clock transition switches the outputs of the flip-flop to their complement state.

→ Graphic symbol

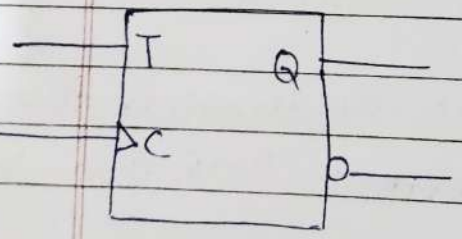


→ characteristic table.

J	K	Q(t+1)	
0	0	Q(t)	No change
0	1	0	Clear to 0
1	0	1	Set to 1
1	1	Q'(t)	complement.

→ **T (Toggle) flip-flop**-

This is obtained from JK type when inputs J and K are connected to provide a single ~~out~~ input designated by T.



(Graphic symbol)

T	Q(t+1)	
0	Q(t)	No change
1	Q'(t)	Complement.

(characteristic table)

2 conditions-.

① when T=0 (J=K=0) a clock transition does not change the state of FF.

Date ___/___/_____ Page No.: _____

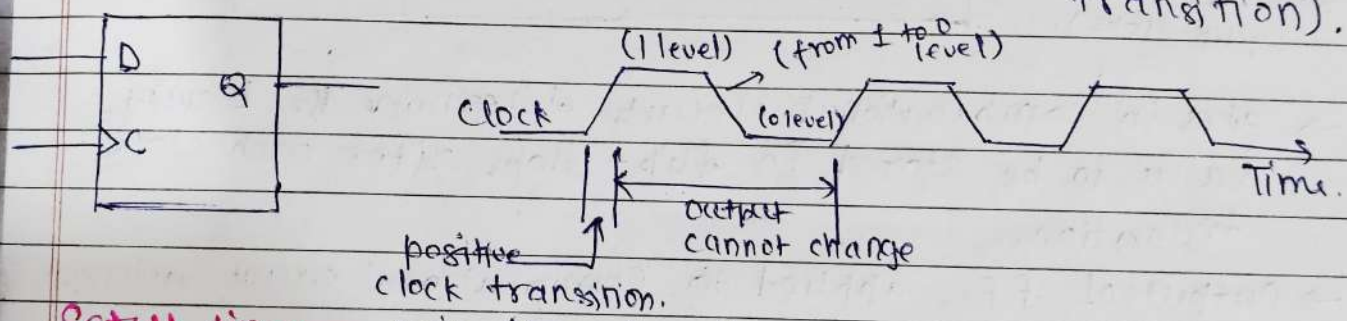
(2) when T=1 (J=K=1) a clock transition complements the state of FF.

→ **Edge-Triggered Flip-Flops-** (synchronize the state change during a clock pulse transition).

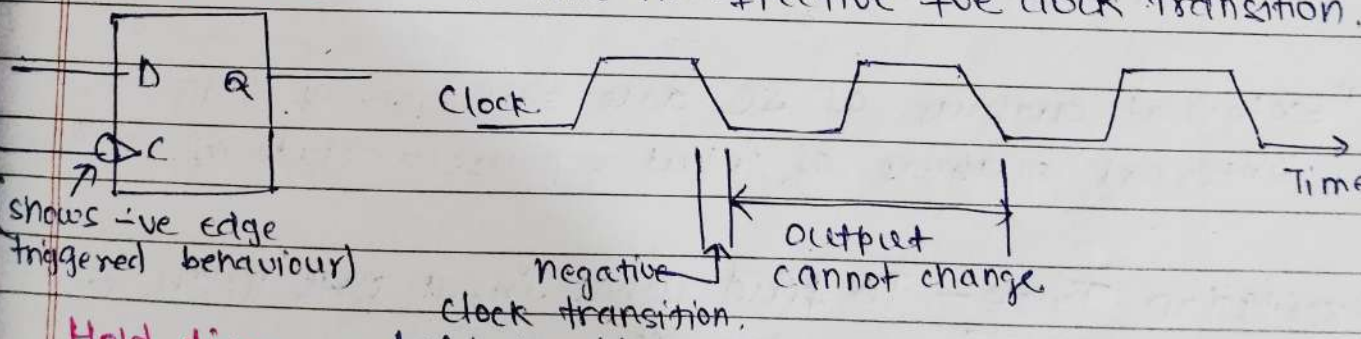
(In this, output transitions occurs at a specific level of clock pulse).

Some edge-triggered flip-flops cause a transition on rising edge of clock signal (positive-edge transition).

& other cause on falling edge (negative-edge transition).



Setup time- min. time in effective +ve clock transition.



Hold time- definite time in which D input must not change after positive transition.

→ **Master Slave Flip-Flop-** (consists of 2 FF(s))

Master- responds to +ve level of clock.

Slave - responds to -ve level of clock.

Result is that output changes during 1 to 0 transition of clock signal.

"In order to succeed, your desire for success should be greater than your fear of failure." —Narendra Modi

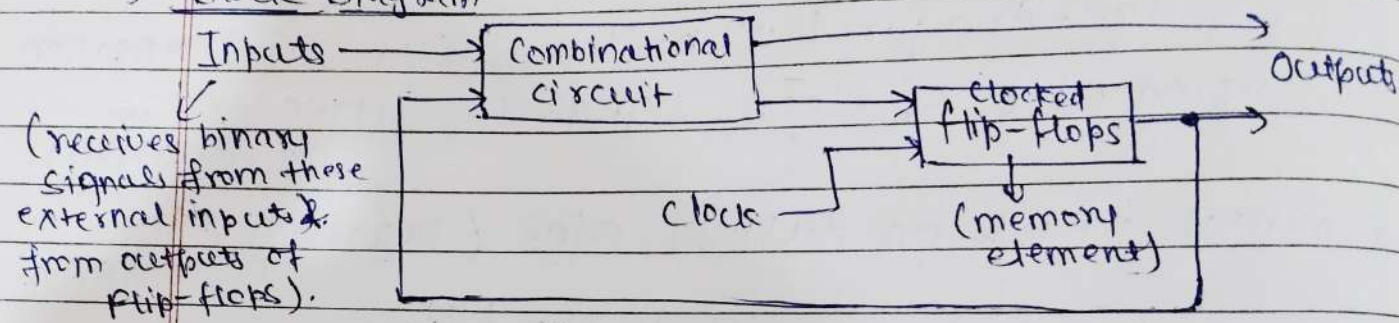
Date ___/___/___

- SEQUENTIAL CIRCUIT -

(interconnection of flip-flops and gates).

"The gates by themselves constitute a combinational circuit, but when included with the flip-flops, the overall circuit is classified as sequential circuit."

→ Block Diagram-



→ gates in combinational circuits determine the binary value to be stored in flip-flops after each clock transition.

→ Output of FF, applied to combinational circuit determine the circuit's behaviour.

"External outputs of SC are functions of both external ~~outputs of~~ input & present state of FF."

Excitation Table - To find what input was given to the flip-flop.

Ch-1
"Digital Logic Circuits"

BY ANJALI YADAV

03-Jan.

– Computer System Architecture –

Chapter - 2nd

→ Integrated Circuit

- Birth of electronic computing – from roughly 1940's through mid-1960s every computer was built from individual parts called "discrete components", which were all wired together.

e.g. ENIAC – consisted of more than 17,000 vacuum tubes, 70,000 resistors, 10,000 capacitors & 7,000 – diodes, all of which require 5 million hand-soldered connections.

By mid 1950s, transistors were becoming commercially available & being incorporated. This were much smaller, faster & more reliable than vacuum tubes, but each transistor was still one discrete component.

In 1959, IBM upgraded their vacuum-tube-based "709" computers to transistors by replacing all discrete vacuum tubes with discrete transistors.

The New Machine, the IBM 7090, was six times faster & half the cost.

These transistorized computers marked the second generation of electronic computing.

However, although faster & smaller, discrete transistors didn't solve the tyranny of

Date __/__/____ Page No.. _____

Numbers.

By the 1960s, this was reaching a breaking point. The insides of computers were often just huge tangles of wires.

The Breakthrough came in 1958, when Jack Kilby, working at Texas Instruments, demonstrated such an electronic ~~circuit are~~ part, "wherein all the components of the electronic circuit are completely integrated.
This is called "IC".

In 1959, ICs were made practical. (Fairchild & Robert Noyce) semiconductor

- Kilby - made IC of germanium - unstable material.
  Fairchild. - made it of Silicon. - more stable
  - more reliable.

- Therefore, Robert Noyce, father of Modern ICs

In early days, IC might contain simple circuit with just a few transistors.

ICs - ICs are sort of like lege for computer engineers "building blocks" that can be arranged into an infinite array of possible designs.

Printed circuit Boards (PCBs) - by using PCBs and ICs together, one can get exactly same functional circuit as made by discrete components

Date __/__/____ Page No.

Basic definition – Circuits in which circuit components like transistors, diodes, registors etc. combine together to perform various functions is called
– Integreted circuit (IC).

Electronic Circuit
(interconnection of electronic components).

① Discrete circuit	② Integrated circuit.
• components are seperable • They occupy large scale. • It is less reliable. • joined by wires or on (PCBs).	• It is a semiconductor wafer, also called chip or microchip • chip of silicon. • It is a complete electronic circuit consisting of both active and passive components on single chip of silicon.

- can function as an amplifier, oscillator, gate, flip-flop, timer, counter, computer memory, microprocessor etc.
- Depending on type of signal process, IC can be categorised as

a) Linear (Analog) ICs.

b) Digital ICs.

Date ___/___/_____ Page No. _____

Definition by book – An IC is a small silicon semiconductor crystal, called a chip, containing the electronic components for digital gates.

Chip is mounted on ceramic or plastic container, & connection are welded by thin gold wires.

Chip contains no. of pins ranging from 14 to 100 or more.

→ Depending on no. of gates and logic operations chip contains, it is divided into four parts –

1. SSI
2. MSI
3. LSI
4. VLSI.

- SSI – (small scale integration) devices contain less than 10 gates & limited by no. of pins.
- MSI – (Medium scale integration) contains 10 to 200 gates in single package. & performs elementary function such as decoders, adders, and registors.
- LSI – (Large scale integration) contains b/w 200 to few thousand gates in single package. They contain digital systems such as processors, memory chips and programmable modules.

Date __/__/____ Page No.: __

- VLSI – (Very large scale integration).
  contain thousands of gates.
  Ex – large memory arrays & complex microcomputer chips.
  It has revolutionized the computer system design technology since it is small in size and of low cost.

→ Integrated circuit belongs to specific circuit technology, referred to as "digital logic family".

Each logic family has its basic electronic circuit upon which complex circuits & fn are developed.
This basic circuit is either NAND, NOR or invertor Gate.

→ Some logic family of integrated circuits are –

① TTL (Transistor – transitor logic) – considered as standard. Evolution of previous technology that uses diodes & Transistors for basic NAND gate k/a DTL (Diode – transistor logic). Later, it ~~wer~~ was redesigned as TTL to improve circuit operation.

It has several variations –
a) Standard TTL
b) High – speed TTL
c) Low – power TTL

Date __/__/____ Page No.. _____

d) Schottky TTL
e) Low-power schottky TTL
f) Advanced Schottky TTL.

- Power supply voltage for TTL is (5 Volts) & two logic levels are approximately (0 & 3.5 volts)

(2) ECL (Emitter-coupled Logic) — provides high-speed digital circuits in integrated form. Used in computers such as ~~Digita~~ super computers and signal processors. It operates in non-saturated state.

(3) MOS (Metal-oxide semiconductor) — unipolar transistor that depends on the flow of only one type of carrier (either electrons or holes) i.e., n-channel or p-channel.

→ A p-channel MOS is reffered to as PMOS.
→ An n-channel MOS is reffered as NMOS.

(4) CMOS (complementary MOS) — uses PMOS and NMOS transistors connected in complementary fashion in all circuits.

Advantages —

- High packing density
- Simpler processing technique during fabrication.
- economical operation.

Note → TTL & ECL uses bipolar transistor.

Date __/__/____ Page No.: ____

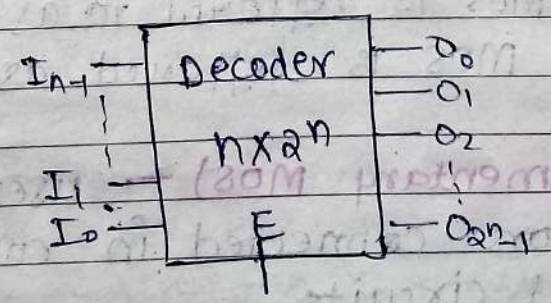
DECODER

- A decoder is a combinational circuit that decodes binary information from n input lines to a maximum of 2^n unique output lines.

- The decoders are called n-to-m line decoders, where $m \leq 2^n$.

- Their purpose is to generate the 2^n (or fewer) minterms of n input variables. Each combination of inputs will assert a unique output.

- If the n-bit coded information has unused combination, the decoder may have fewer than 2^n outputs.

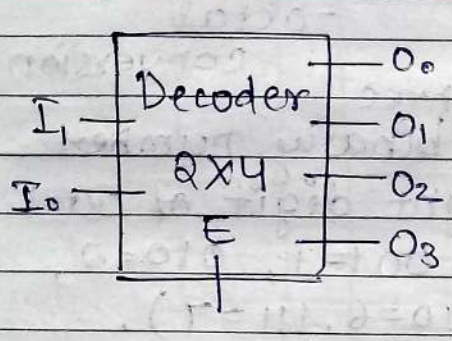


Value of E should be 1 to make it function properly.

- Address से memory location तक का रास्ता decoder ही bताता hai.
- It can be used to express boolean fn.
- to generate different circuits.

- Decoders are also combinational circuits logically we can say a DeMux can be converted into a decoder by setting input line as enable line and selection line as input lines.

- **2-to-4 Decoder**—



Truth Table—

I/p		O/p			
I_1	I_0	O_3	O_2	O_1	O_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

- **from book**—

We know, binary code of n bits represents upto 2^n distinct elements of coded information.

Definition— refer to previous page.

- **Purpose of Decoders** — To generate 2^n (or fewer) binary combinations of n input variables.

Decoder has n inputs & m outputs and is also referred to as n×m decoder.

- **3-to-8 line Decoder**—

Data inputs — A_0, A_1, A_2.

8 outputs — each representing one of the combinations of three binary input variables.

Date ___/___/_____ Page No.: ___

The 3 inverters provide complement of inputs.

8 ~~and~~ AND gates generate one of the binary combination.

- <u>Application of this decoder</u> - Binary-to-octal conversion.

Here, input represents three binary number & outputs represent eight digit of octal no. system (i.e., 000=0, 001=1, 010=2, 011=3, 100=4, 101=5, 110=6, 111=7).

3-~~to~~-8 line decoder can be used for decoding any 3-bit code to provide 8-outputs.

- <u>Enable Input(E)</u> - Used to control the operation of circuit.

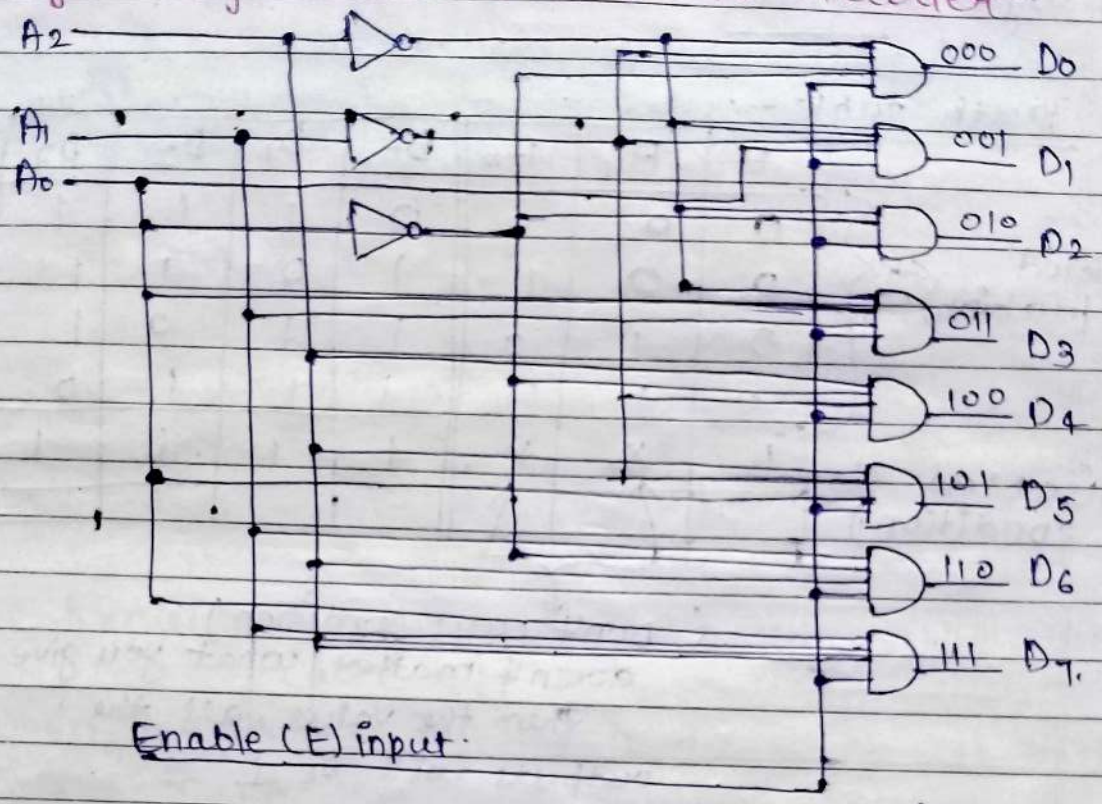
⇒ The decoder is enabled (means it give some output) when E is equal to 1 and disabled (not work) when E is equal to 0.

When ~~E0~~ E=0, all outputs will be 0 regardless all 3 inputs, whatever you gave.

When E=1, decoders operates in Normal Fashion.

Date __/__/____ Page No.. _____

- Logic diagram for 3-to-8 line decoder-



- Truth Table- Refer to book. (Refer to book).

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 no. equivalent of the binary number that is available in the input data lines.

- NAND gate decoder-

Instead of AND gate, we use NAND gate.

It'll complement all the outputs, which we are getting in AND gate 2-to-4 decoder.

Here,

The decoder is Enabled when E = 0, & only one output is 0 at any time and all 3 are 1.

→ Logic diagram – refer to book pdf.

→ Truth table –

	E	A_1	A_0	D_0	D_1	D_2	D_3
(enabled situation)	0	0	0	0	1	1	1
	0	0	1	1	0	1	1
	0	1	0	1	1	0	1
	0	1	1	1	1	1	0
(disabled condition)	1	X	X	1	1	1	1

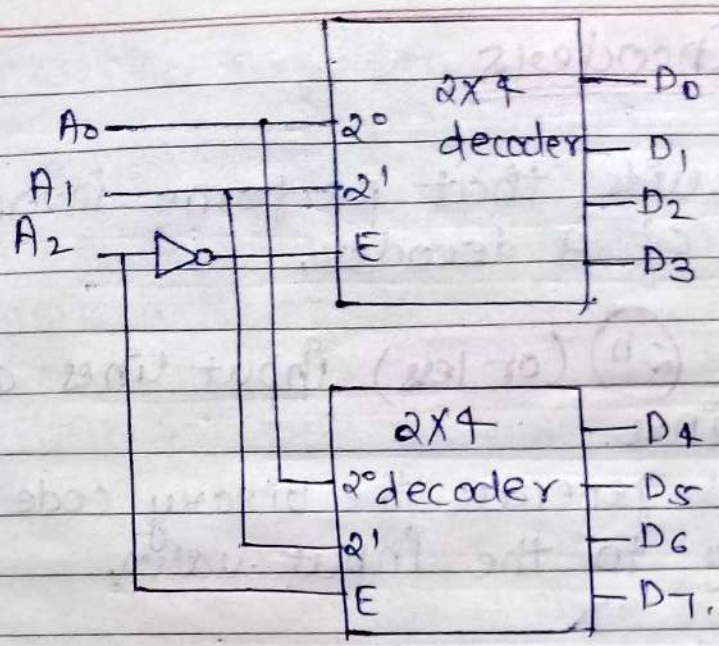
(Don't care condition) – doesn't matter, what you give them the value, all the outputs will be 1.

Note – Some decoders may have 2 or more enable inputs that must satisfy a given logic condition in order to enable the circuit.

→ Decoder Expansion –

When larger size decoder is needed and we have only smaller size, then "we combine two or more decoders with enable inputs to form a larger decoder."

→ 3 X 8 decoder constructed with two 2X4 decoders –



We know that, each decoder is enabled when $E = 1$.

When, $E = 0$, the decoder is disable & all outputs are in 0 level.

So Here —

- When $A_2 = 0$, upper decoder is enabled & lower is disabled.

  lower one become inactive with all outputs at 0.

  ~~op~~ Output of upper decoder is D_0 to D_3 depending on value of A_0 & A_1.

- When $A_2 = 1$, lower decoder is enabled & upper in disabled.

lower decoder output generates the binary equivalent D_4 through D_7.

Therefore, Enable inputs are convenient feature for interconnecting 2 or more circuits for expanding digital components.

Date ___/___/_____ Page No.: ___

Encoders

Digital circuits that performs inverse operation of a decoder.

Encoder has 2^n (or less) input lines and n output lines.
Output lines generate the binary code corresponding to the input value.

Example – Octal to binary encoder. –

Truth Table –

Inputs								Outputs		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_0
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

It has 8 inputs, one for each octal digits, 3 outputs that generates the corresponding binary number.

Note – Only 1 input has a value of 1 at any given time, otherwise circuit has no meaning.

The encoder can be implemented with three OR gates.

When output $A_0 = 1$ (in truth table), input octal digit is 1 or 8 or 5 or 7. Similarly with others.

Boolean functions –

$$A_0 = D_1 + D_3 + D_5 + D_7$$
$$A_1 = D_2 + B_3 + D_6 + D_7$$
$$A_2 = D_4 + D_5 + D_6 + D_7.$$

MULTIPLEXERS

(special and most widely used combinational circuit).

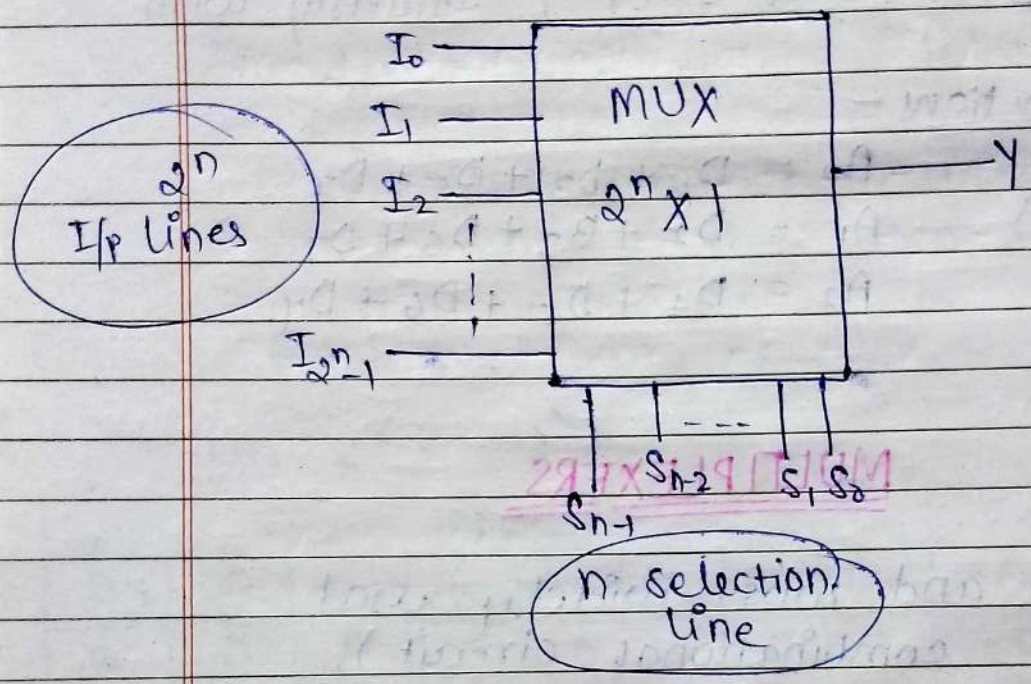
- Main requirement is out of many I/p we have to select one. e.g. telephone, train lowing the station.
- Here we select I/p from one of many I/p lines (maximum 2^n) and direct it to single output line.
- The selection of particular I/p line is controlled by a set of selection line.

$$\text{no. of I/P lines} \leq 2^n$$

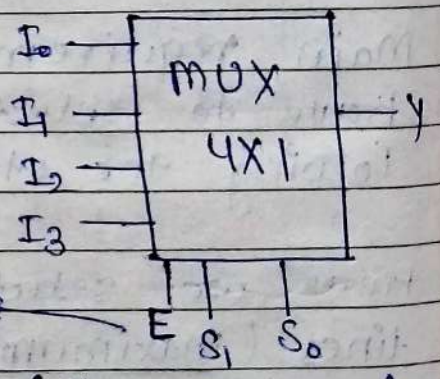
- can never select more than one I/p line at a time.

Date __/__/____ Page No.: ____

→ **Application** – Parallel to series combination, communication system, computer memory etc.



- **4 to 1 multiplexer** –

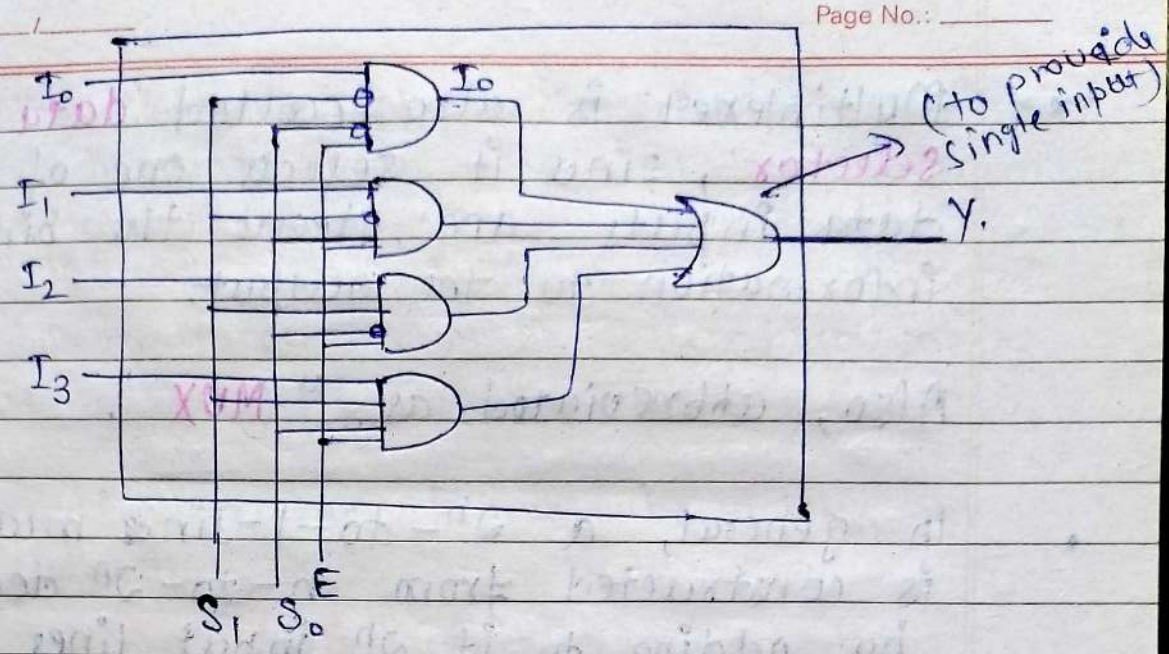


(Block Diagram)

- ~~Truth~~ Function Table –

S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

- Logic Diagram – Implementation –



• Characteristic Equation:-

$$Y = \bar{S}_1\bar{S}_0 I_0 + \bar{S}_1 S_0 I_1 + S_1\bar{S}_0 I_2 + S_1 S_0 I_3$$

(SOP expression)

when $E = 0$,

$$Y = E\left[\bar{S}_1\bar{S}_0 I_0 + \bar{S}_1 S_0 I_1 + S_1\bar{S}_0 I_2 + S_1 S_0 I_3\right]$$

→ From book:-

A 2^n-to-1 multiplexer has 2^n input data lines and n input selection lines whose bit combinations determine which input data are selected for the output.

Also, 4-to 1-line multiplexer has 6 inputs & one output.

So, truth table describing the circuit needs 64 rows since six input variables can have 2^6 binary combinations.

- Multiplexer is also called data selector, since it selects one of many data inputs and steers the binary information to the output.

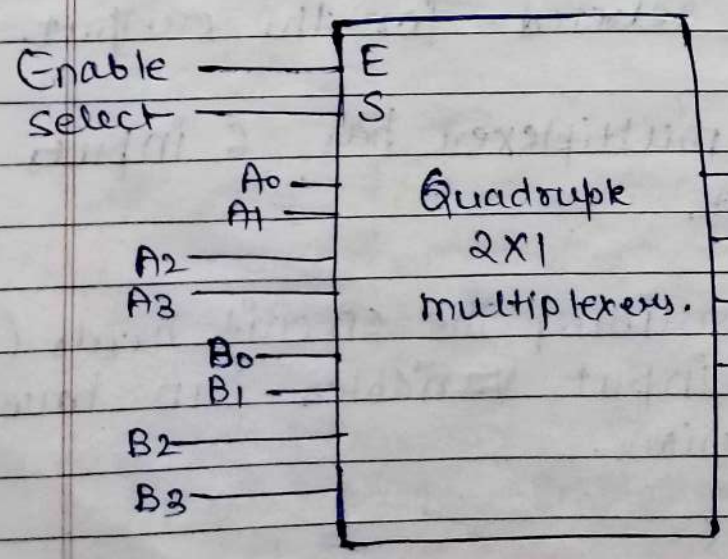
Also, abbreviated as "MUX".

- In general, a 2^n-to-1-line multiplexer is constructed from n-to-2^n decoder by adding to it 2^n input lines, one from each data input.

- Enable input in multiplexer is useful for expanding 2 or more multiplexers to a multiplexer with a larger no. of inputs.

- Two or more multiplexers are enclosed within a single integrated circuit package.

Block diagram of quadruple 2-to-1 line multiplexers.



E	S	Y
0	X	All 0's
1	0	A
1	1	B.

Date __/__/____ Page No.: _____

Selection & enable inputs in multiple-unit ~~integrated~~ construction are usually common to all multiplexers.

Here, Output Y_x ~~is~~ can be selected to come either input A_x or B_x. $(x = 0, 1, 2, 3.)$.

For normal operation, E must be active.

REGISTERS.

Register is a form of memory that uses a series of Flip-Flop to store the binary digits.

Need- 1 Flip-Flop can store only 1 bit of data either 0 or 1. To store a group of bits, we required a group of flip-flop that is called register.

Notes- n bit of register can store n bit of binary data & has n F.F.

Example-

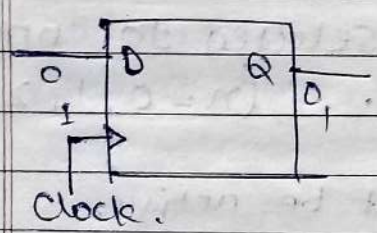
4-bit register	8-bit register
↓	↓
4-bit data	8-bit data
↓	↓
4 no. of F.F.	8 no. of F.F.

Flip-Flop- We use D-Flip-Flop in registers because it require 1 bit data to store. If we give 1 as input, we simply get 1 as

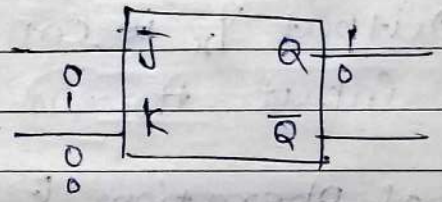
Date ___/___/_____ Page No.: ___

output whenever we require.

- <u>Conversion of JK flip-flop into D-flip-flop</u>

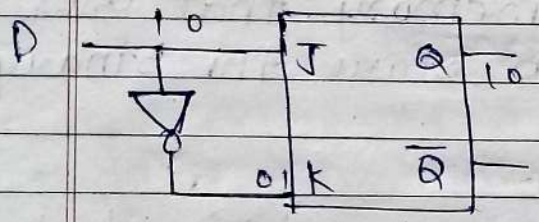


(D-flip flop)



(JK-Flip-flop)

(JK into D-FF) —



Imp' Note → In addition to the flip-flops, a register may have combinational gates that perform certain data-processing tasks.

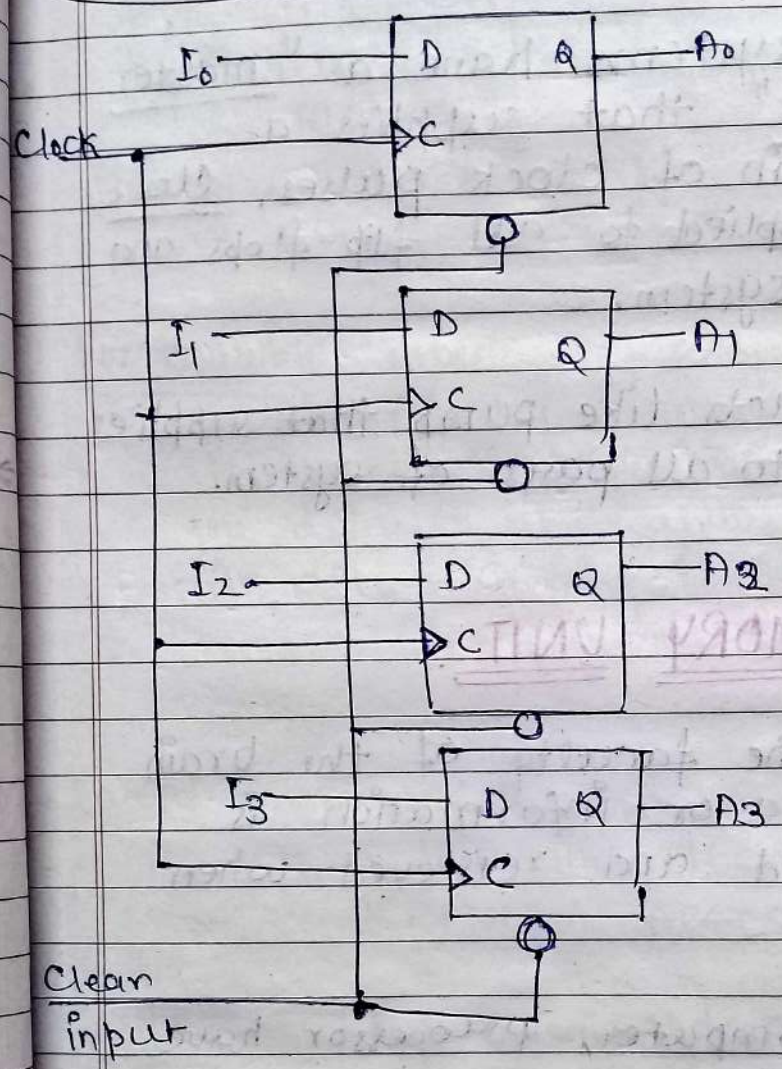
from book → <u>Register</u> consists of group of flip-flops and gates that effect their transition.

The <u>flip-flops</u> hold the binary information and the <u>gates</u> control when and how new information is transferred into the register.

simplest registers contain only flip-flop and not gates.

Date __/__/__ Page No.: _____

4-bit register:-



The clear input is useful for clearing the register to all 0's prior to its clocked operation.

When clear input goes to 0, all flip-flops are reset asynchronously.

The clear input must be maintained at logic 1 during Normal clocked operation.

→ **Register load** – The transfer of new information into a register is referred to as loading of register.

If all the bits of the register are loaded simultaneously with a common clock pulse transition, we say that the loading is done in parallel.

Date ___/___/_____ Page No.:

→ ~~More dis~~ Register with Parallel load →

Most digital systems have a "master clock generator" that supplies a continuous train of clock pulses. Clock pulses are applied to all flip-flops and registers in system.

- master clock acts like pump that supplies constant beat to all parts of system.

→ MEMORY UNIT

(write-in read-out)

Memory is the faculty of the brain by which data or information is encoded, stored and retrieved when needed.

→ In phone or computer, processor have frequency in GHz, means—

$$\text{frequency} = 2\,GHz$$

$$\text{time} = \frac{1}{\text{frequency}}$$

$$= \frac{1}{2 \times 10^9}\ \text{sec.}$$

$$= \frac{1}{2} \times 10^{-9}\ \text{sec.}$$

$$= \frac{1}{2}\ \text{nsec} \quad (\text{nano sec.})$$

1 kilo <unit> $= 10^3$ <unit>
1 mega <unit> $= 10^6$ <unit>

It means our processor can perform

"The butterfly counts not months but moments, and has time enough." - Rabindranath Tagore

a single task in $\frac{1}{2}$ nsec.

- **Primary Memory** (to perform immediate tasks)
- **Secondary Memory** (memory used for permanent storage).

For example- when we play an audio, OS goes to primary memory, and performs instructions such as mouse click, opening up the default application and playing the file after bringing it from secondary storage into the primary memory.

- Data from primary memory can be accessed in any order ~~since~~ therefore, named as "RANDOM ACCESS MEMORY". (R.A.M).
- DRAM- "Dynamic Random Access Memory". It contains a transistor (to retain binary bits) associated with Capacitor (have charge). Periodically charging is needed to transistor for value retaintion.

But, It's slow for modern day processors.

- SRAM- (cache) - "Static Random Access Memory". fast than DRAM. Cache is made up of static-RAM which does have transistors or capacitors and is costly.

Cache is fastest memory among all.

Date ___/___/_____ Page No.:

They all are volatile, either cache or main memory.
means they can only retain the data untill power offs.

Therefore, to store the data permanently, we opt for other type of memory called -

"SECONDARY MEMORY".

- Slower than Primary Memory.
- Retains Data permanently.
- Bigger in size.
- Cost - effective.
- Example - Hard Disk - Drive (HDD).
- Non-volatile Storage.

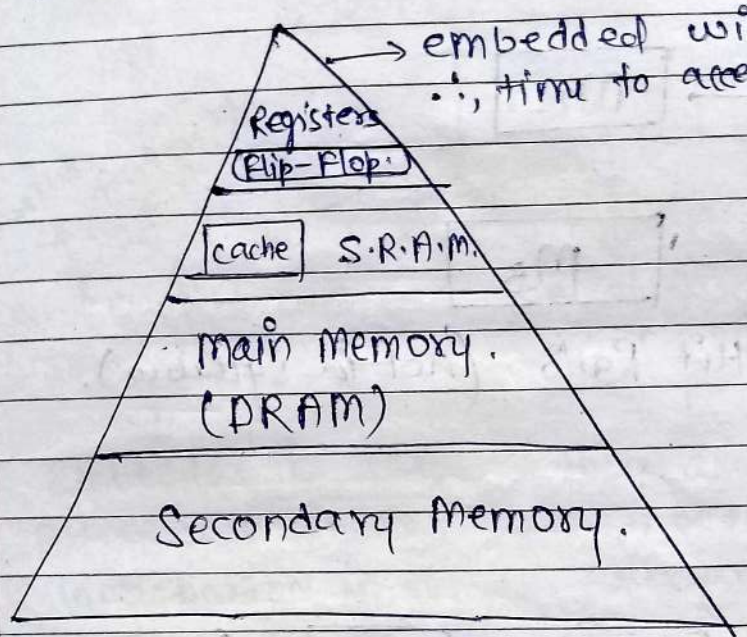
Note - Processor didn't know the existence of secondary memory since it doesn't have any direct link with it, Main memory communicate with secondary memory uses pages.

- Memory Hierarchy & interfacing -

Hierarchy means ranking.

It requires parameter to rank types of memories.

If we talk of "Access Time" and "size" they are —



In case of "Cost" and "Usage Frequency" this hierarchy is reversed.

- Memory interfacing -

→ Part of computer organization.

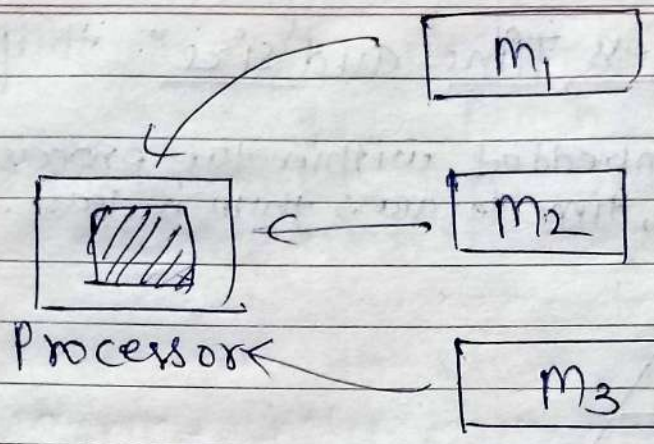
→ Deals with the way of connecting various levels of memory units to Processor and I/O peripherals.

Unit of measuring speed of processor is : MIPS (million instruction Per Second).

$$\boxed{n^{th} \text{ level} \subseteq (n+1)^{th} \text{ level}}$$

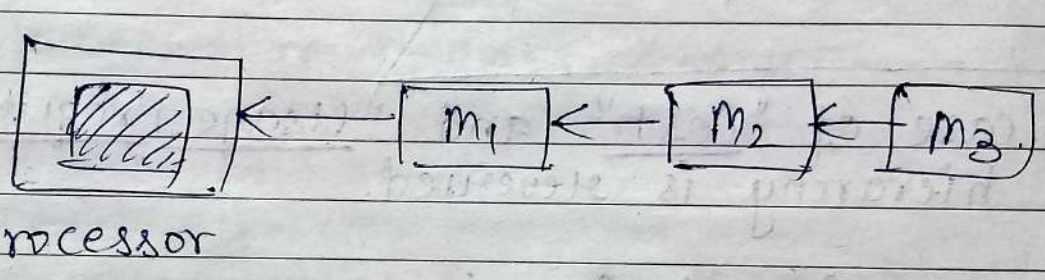
① way ① —

Processor is connected to all memory unit at a time and while accessing it uses all memory units side by side.



Note - Miss Ratio & Hit Ratio (not in syllabus).

(2) Way - (2) -



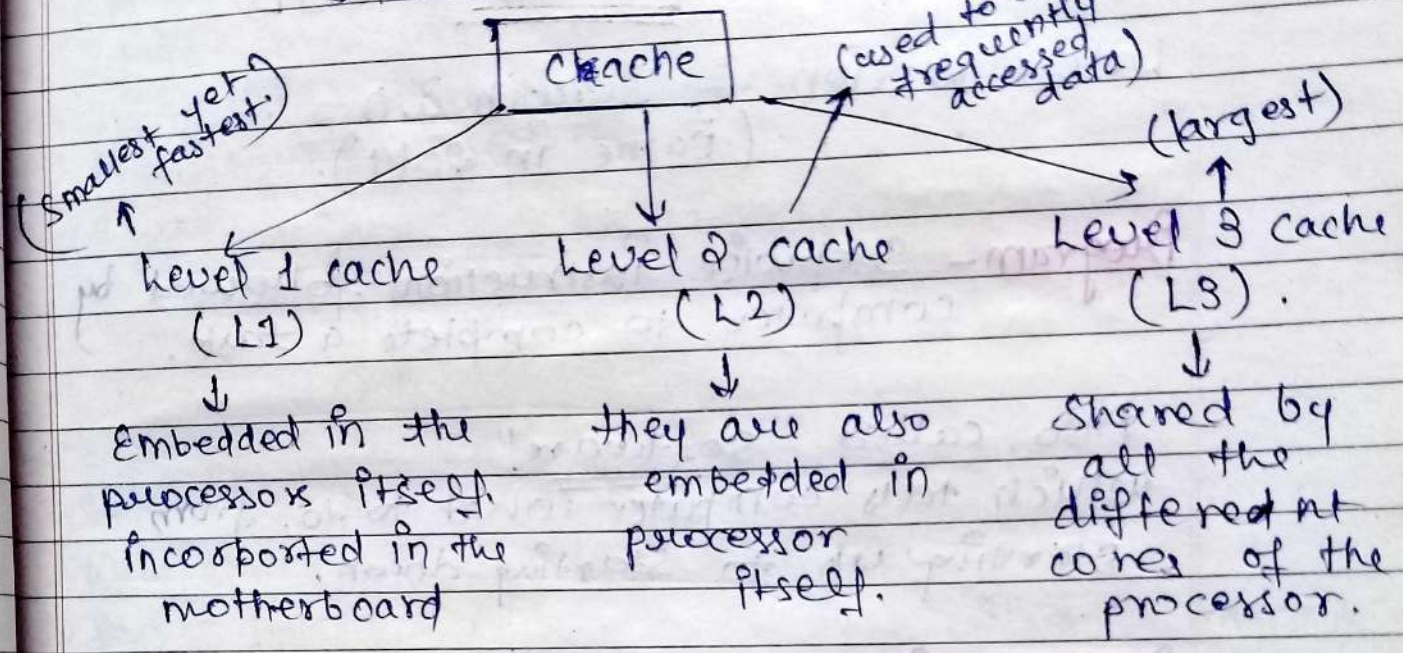
- Introduction to Cache Memory -

Some computer games requires 100+ GB memory storage, so, RAM (or main memory) suggested ~~for~~ or recommended for them to play is either 8 or 16 GB as required.
Our processors are executing the codes of these games but not 100+ GB code at once. that's why having a very smaller main memory, we can play our game.

- Modern Day systems have level of cache memory -

• The information who requirment frequency is high is kept in "cache".

Generally, it is of 3 levels means mostly used is 3 levels.



Cores- DUAL CORE, QUAD CORE, OCTA CORE } MIMD family.

• Cache related terms-

1. Cache Hit — When processor get the required data in cache, it's called cache hit.
2. Hit latency — time required for cache hit.
3. Tag Directory (Data structure).
4. Cache Miss.
5. Miss latency.

• Some other terms -

1. Page Fault - If data is absent in main memory and cache as well.
2. Page Hit.

28 Jan — Computer System Architecture —

Date / / Page No.:

Chapter-5

"BASIC COMPUTER ORGANISATION AND DESIGN"

(1) Instruction Code

Group of bits that tell the computer what to do basically.

Format —

Opcode	Address

→ tells the address of the operand on which the operation is going to perform.

Opcode - type of operation we want to perform (e.g. +, -, *)

operand - on which we perform operations.

Size of word in PC — 1024 bits = 2^{10}

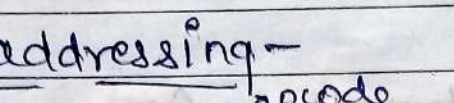
Size of format (16 bits) → 12 bits (address), 4 bits (opcode) } → Division of memory for instruction code.

- Three type of operand is used (or methods) to fetch the operand for operation —

(1) Immediate operand
(2) Direct addressing (0)
(3) Indirect addressing. (1)

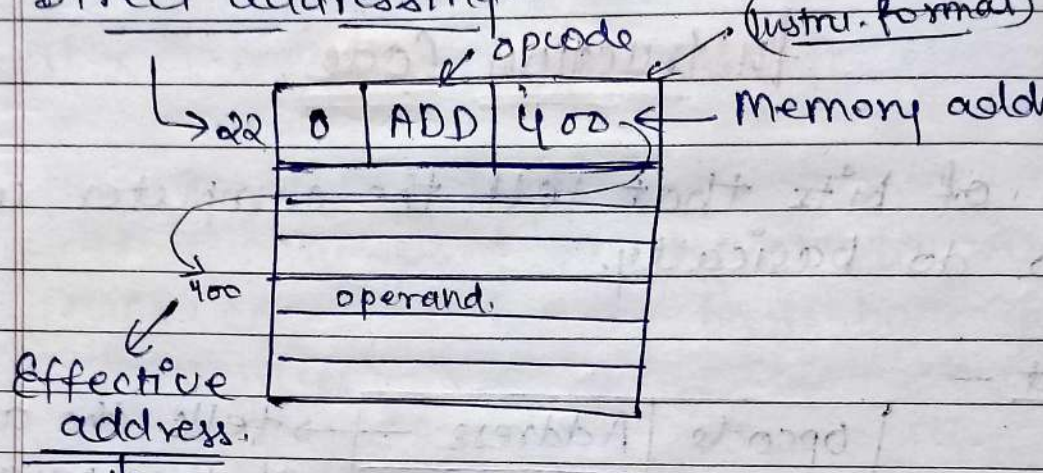
Date __/__/____ Page No.

- Immediate operand –



(address acts as a operand)

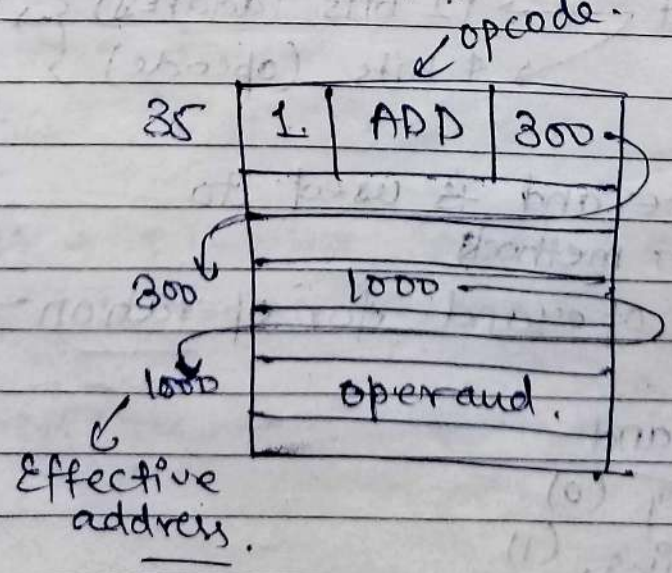
- Direct addressing –



By using which you can retrieve ur desired operand.

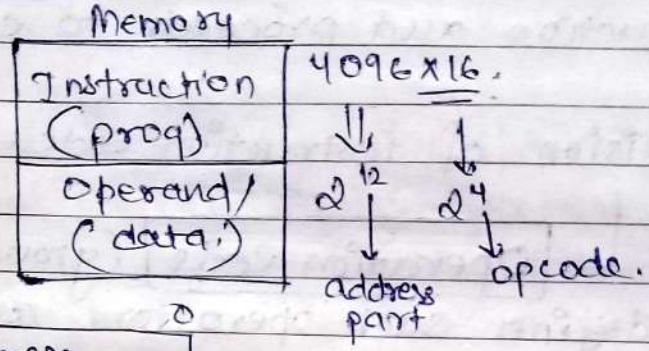
In Direct addressing we can directly fetch the address of operands using memory address.

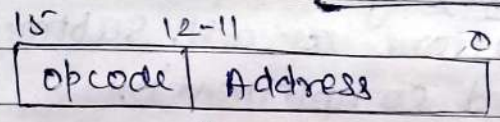
- Indirect addressing –



Date ___/___/_____ Page No. or unique.

→ Every computer have their different instruction formats./sets.





⇓

Instruction format.

It is of three types —

1. Memory Reference Instruction.
2. Register Reference Instruction.
3. I/O Reference Instruction.

→ Organisation of Basic computers is specified with —
- Internal Registers
- Timing & control structure
- Set of instructions that it uses.

→ Interal organisation is defined by sequence of micro operations it perform on data stored in its registers.
General purpose computers is capable of executing various micro operations.
User can control it using programs and can alter too.

→ Instruction code (binary code) along with data is stored in memory. Computer reads it

Date __/__/__ Page No.

from memory and places it in control register. It interprets the binary code of instruction and proceeds to execute it.

→ Division of instruction codes—

↳ Operation Code (group of bits that defines such operations as add, subtract, multiply, shift and complement).

- No. of bits required by O.C. depends on no. of operations available in PC.

Note— Operation code must consists of n bits for a given 2^n (or less) distict operations.

Operation Code — Binary code that tells the computer to perform specific operations.
(op code)

→ Control unit ~~rec~~ receives the instructions from the memory and interprets the operation code bits.

Note— For every operation code, the control issues a set of micro-operations needed for the hardware implementation of specified operations, sometimes called "~~AA~~ macrooperation" (set of microoperations)

Date __/__/____ Page No.: _____

→ Instruction codes must be specified with registers or memory word where operands are found and result to be stored.

memory word is specified by their "address" in instruction & processor registers by another binary code of k bits that specifies on of 2^k registers.

→ Basic organisation of computer—

① Processor Registers.

② Instruction Code formats.

(operations to be performed) (address) → tells where to find operand in memory.

→ For memory unit with 4096 words, 12 bits is required to specify address.
since $2^{12} = 4096$.

If each instruction is stored in one 16-bit memory word, we have 4-bits for operation code (opcode) (since, 16 possible operations are there) & 12 bits to specify address for operand.

Control reads 16 bit instruction from program part of memory. & Execute operation.

Date __/__/__ Page No.

→ Accumulator / AC — single processor register in computer

The operation is performed with the memory operand & content of AC.

Note:

If operation in an instruction code doesnot need operand from memory e.g. clear AC, complement AC, & increment AC, it operates on data stored in AC.

∴, 2nd part of ~~operat~~ instruction code (bits 0 through 11) is not need & used to specify other operations for PC.

→ for Direct & Indirect Addressing —

15	14 — 12	11 — 0
I	Opcode	Address

(Instruction Format)

(mode bit) tells that it is direct / indirect.

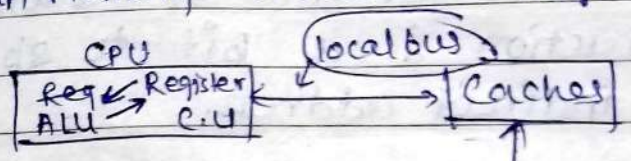
Note — The memory word that holds address of operand in indirect address instruction is used as pointer to an array of data.

The pointer could be placed in a processor register instead of memory

(2) Computer Registers

Temporary storage device present in the processor.

Main Memory → Cache → Register → operations



- Different Registers in memory —

(1) PC (16 bits) → Program counter (holds address of instruction to be executed next).

(2) AR (16 bits) → Address Register. (address from memory).

(3) IR (16 bits) → Instruction Register (holds instruction codes).

(4) TR (16 bits) → Temporary Register (holds temporary Data)

(5) OUTR (8 bits) INPR (8 bits) → Input / Output Register. (holds Inp./out. characters)

(6) DR (16 bits) → Data Register (holds the operands)

(7) AC (16 bits) Accumulator (used to store results) also k/a "Processor Register" (also used for manupulating data).

- Why is register needed ???

Ans – Control Unit needs register for storing the instruction code after it is read from the memory.

Date __/__/____ Page No.:

Note — The memory unit has a capacity of 4096 words & each word contains 16 bits. 12 bits of instruction word are needed to specify address of an operand. This leaves 3 bits for the operation part of the instruction & a bit to specify a direct or indirect address.

→ Program Counter (PC) — has 12 bits. It reread the sequential instructions previously stored in the memory.

PC is incremented by one, so it holds the address of next instruction in sequence. Since branch instruction is encountered in between sequential instructions & transfered to PC to become address of next instruction.

→ Common Bus System —

It is path for transfering information from one register to another and between memory and registers.

- It is constructed using "multiplexer" or "three-state buffer gates".

- Load (LD) — It should be enabled to receive data from the bus during next clock transition.

Since, AR & PC have 12 bits, when contents of AR and PC are applied to the 16-bit common bus, the 4 most significant bits are set to 0's. when AR and PC receive information from the bus, only 12 least significant bits are transferred into the register.

- INPR is connected to provide info to bus but OUTR can only receive info from bus.

INPR receives character from input device & transfer it to AC. & OUTR receives character from AC and delievers it to output device.

- Bus lines (16) are connected to inputs of 6 registers & memory. 5 registers have 3 control inputs (LD-Load, INR-Increment, CLR-clear).

→ 16 inputs of AC come from adder & logic circuit. This circuit has three sets of inputs —.

(1) 1 come from outputs of AC, used to implement register microoperations (e.g. complement AC & shift AC).

(2) 1 come from data register, DR, used for arithemetic and logic micro-operations (e.g. add DR to AC or And DR to AC) & result is transferred to AC.

(3) 1 come from input register, INPR.

Date __/__/____ Page No.

(3) Computer Instructions

- Basic computer Instruction format is-

(1) Memory Reference Instruction-

16 bit registor-

15	14	12-11	0
I	OPCODE	ADDRESS	

Addressing mode
I = 0, direct address
I = 1, Indirect address.

Range of OPCODE - 000 - 110

(2) Register Reference Instruction-

15	14	11	0
0	111	Register operation	

(what to perform)

Opcode = 111 } (always)
I = 0

(3) Input - Output Instruction-

15	14	12-11	0
1	111	I/O Operation/test	

Opcode = 111
I = ~~0~~ 1.

Note - A register-reference instruction specifies an operation on or a test of (AC) register.

Page No.

Date __/__/____

An operand from memory is not needed, ∴, 12 bits are used to specify the operation or test to be executed.

Similary, I/O instruction doesn't need a reference to memory & 12 bits are used to specify the type of input-output operation or test performed.

- The type of instruction is recognized by the computer control from 4 bits in positions 12 through 15 of the instructions.

⇒ Total number of instructions choosen for the Basic computer is equal to 25.

7 - MRI , 12 - RRI , 6 - I/O Instr.

It may seem that PC is restricted to max. 8 distinct operations, but RRI & IORI use their remaining 12 bits as part of operation code, so they exceed 8.

• Hexadecimal code

Well explained in book pg. no. 134 paragraph - 1 & 2.

"Faith is the bird t

Date __/__/____ Page No.

→ Instruction Set completeness -

Type of instructions computer must include while constructing so that the set of instructions are said to be complete —.

1. Arithmetic, logical & shift instructions.
2. Instructions for moving information to & from memory & processor registers.
3. Program control instructions together with instructions that check status conditions.
4. Input & output instructions. (needed for communication b/w computer and user).

Although, set of instructions for basic ~~But~~, computer is complete (i.e. 25) but it is not efficient, coz instructions such as subtract, multiply, OR & exclusive-OR are not performed rapidly.

These operations must be programmed, but it would made the design too complex.

ex. of above — 1. ADD, CMA, INC, CIR, CIL, CLA, AND. (for shifts)

2. LDA, STA.
3. BUN, BSA, ISZ
4. INP, OUT.

"I sincerely believe that Indians have the ability to compete with the best in the world."–Dhirubhani Ambani

Date __/__/____ Page No. ____

(4) Timing And control

- components of control unit in basic computer are —:

(1) 2 Decoders
(2) 1 sequence counter (SC)
(3) 1 control logic gates.

Instruction ~~IR~~ register (IR) reads the instruction from memory.

0-11 bits of IR is applied to (control logic gates).
↓
holds address of operand.

12-14 bit (operation code). — It is decoded by [3 X 8 decoder].
→ 8 o/p of decoder, $D_0 - D_7$ and connected to control logic gate.

→ 4 bit sequence counter → o/p timing signals ($T_0 - T_{15}$).

Timing ~~in~~ for all registers in basic computer is controlled by "Master Clock Generator". Clock pulse is applied to all flip-flop & registers.

- Two type of control organization —:

(1) Hardwired control
(2) Microprogrammed control.

(I = flip flop)

Date __/__/____ Page No. _____

(5) Instruction Cycle

A program in memory consists of a sequence of instructions. The program is executed by going through cycle for each instruction.

· Phases for each instruction cycle consists –

1. Fetch an instruction from memory.
2. Decode the instruction.
3. Read effective address from memory if the instruction has an indirect address.
4. Execute the instruction.

After completion, control goes back to step (1) & process continues indefinitely unless (HALT) instruction is encountered.

→ **Fetch And Decode –**

Initially, PC is loaded with address of 1st instruction & Sequence counter (SC) is cleared to 0, providing a decoded timing signal T_0. After each clock pulse, SC is incremented by one, so that the timing signals go through a sequence T_0, T_1, & T_2 and so on.

T_0: $AR \leftarrow PC$

T_1: $IR \leftarrow M[AR]$, $PC \leftarrow PC+1$

T_2: $D_0, \ldots, D_7 \leftarrow$ Decode IR (12-14),
$AR \leftarrow IR\ (0-11)$, $I \leftarrow IR\ (15)$

(Flip-Flop)