Computer organization & structure

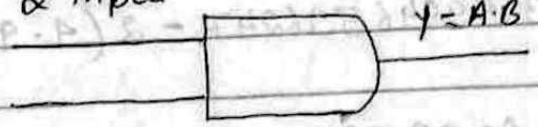
What is logic Gate?

Logic Gate are electronic circuit that operate one or more input signals to produce an output signal. The output signal of a gate is simple boolean operation of its input signal & any boolean function can be represented in the form of Gate.

AND Gate

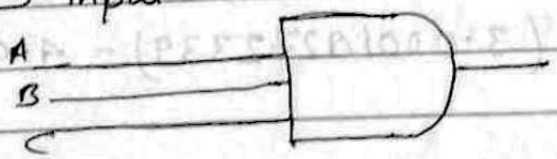
An AND Gate has two or more input signals & only one output signal. The output will be one only when both the inputs are 1.

2 input



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

3 input



$2^3 = 8/2 = 4/2 = 2/2 = 1$

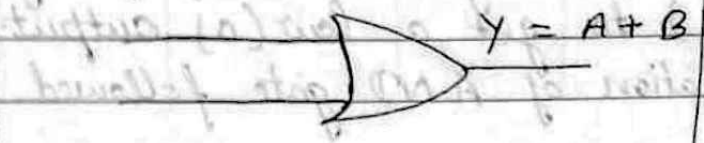
A	B	C	A B C
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Page :
Date : / /

OR Gate

The OR gate has two or more input signals but a single output signal. if any one of the input signal is high the output signal is high.

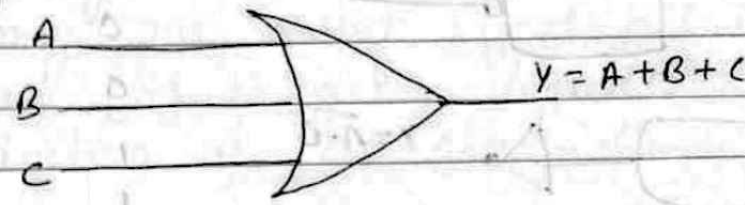
2 input



$2^2 = 4/2 = 2/2 = 1$

A	B	Y=A+B
0	0	0
0	1	1
1	0	1
1	1	1

3 input

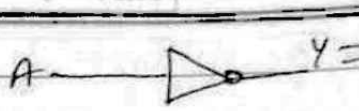


$2^3 = 8/2 = 4/2 = 2/2 = 1$

A	B	C	Y=A+B+C
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Not Gate

Not gate has only one input signal & one output signal. The output state is always the opposite of the input state. It is also called as inverters.

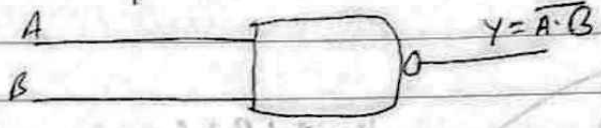


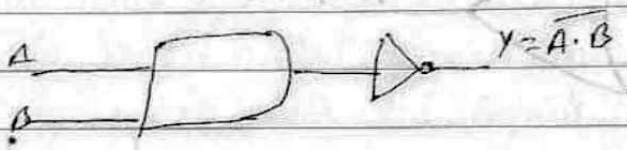
A	Y = Ā
0	1
1	0

NAND

A NAND gate has two or more input signals but only one output signal. All input signals must be high (1) to get a low (0) output. It is a Combination of AND gate followed by a NOT Gate.

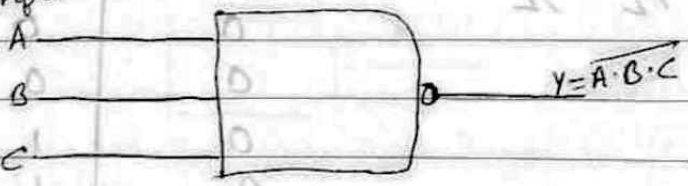
2 inputs

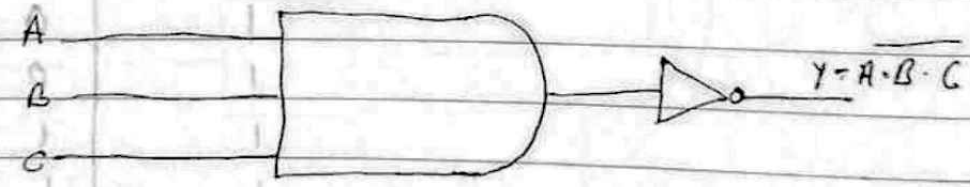




A	B	Y = A·B
0	0	1
0	1	1
1	0	1
1	1	0

3 input





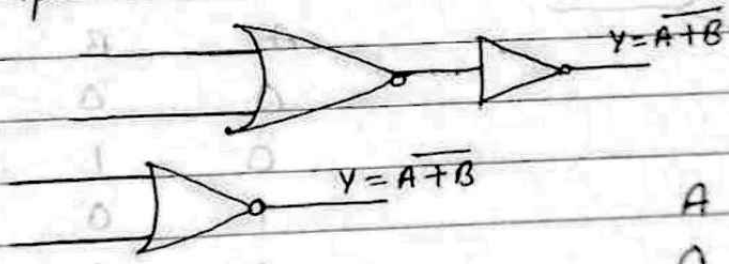
$2^3 = 8/2 = 4 \Rightarrow 4/2 \Rightarrow 2 \Rightarrow 2/2 = 1$

A	B	C	$Y = A\overline{B \cdot C}$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

NOR Gate

A NOR gate has two or more input signals but only one output signal. All input have to be low (0) to get a high (1) output. It is Combination of OR gate followed by NOT gate

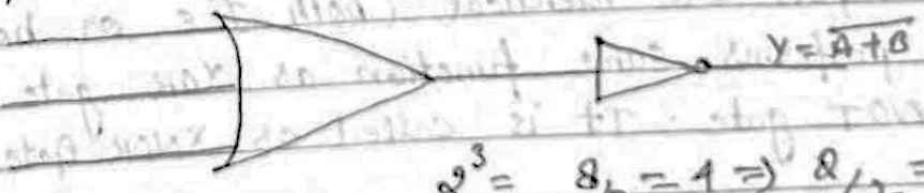
2 input



J.J

A	B	$Y=\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

3 input

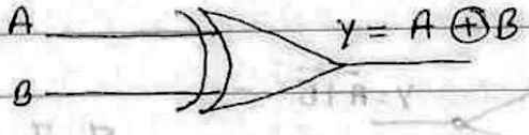


$2^3 = 8/2 = 4 \Rightarrow 8/2 = 2 \Rightarrow 1$

A	B	C	$Y = \overline{A+B+C}$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Exclusive OR Gate

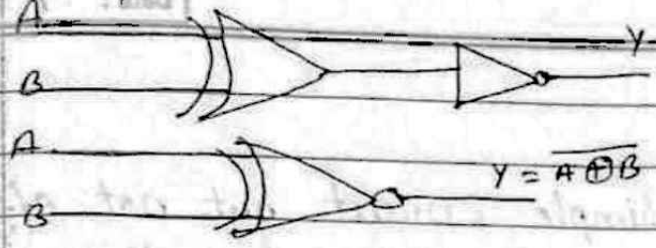
The exclusive OR gate gives high output when the inputs are not at equal logic level. The exclusive OR operation is widely used in digital circuit. It is also called as XOR



A	B	$Y = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

XNOR

The Exclusive NOR gate gives 1 at output, when the inputs are identical (both 1's or both 0's). It performs same function as XOR gate followed by a NOT gate. It is called as XNOR gate.



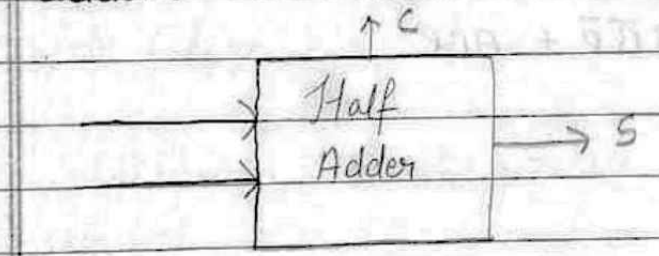
A	B	$Y = \overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

Adders

Adders are the combinational circuit which can perform addition of binary bit.

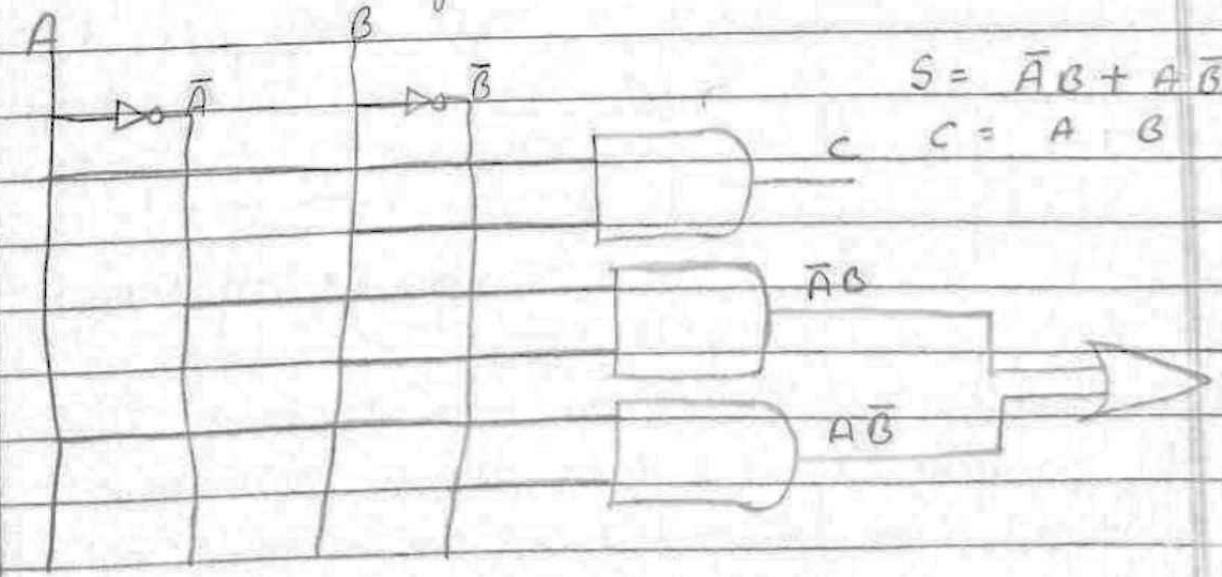
Half Adder

Half Adder adds two binary digits & produces two binary outputs called sum & carry. In half adder the inputs are the augend lets say 'A' & addend 'B' bits.



Block diagram

Inputs		Sum	Carry
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

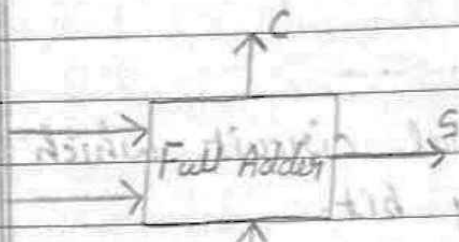


$S = \bar{A}B + A\bar{B}$

$C = A \cdot B$

Full Adder

Half - Adder is a simple circuit but not effective for adding more than two digits. When three bits need to be added, we use a full adder circuit (The third bit is the previous carry bit.)



A	B	P	S	C
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

S =

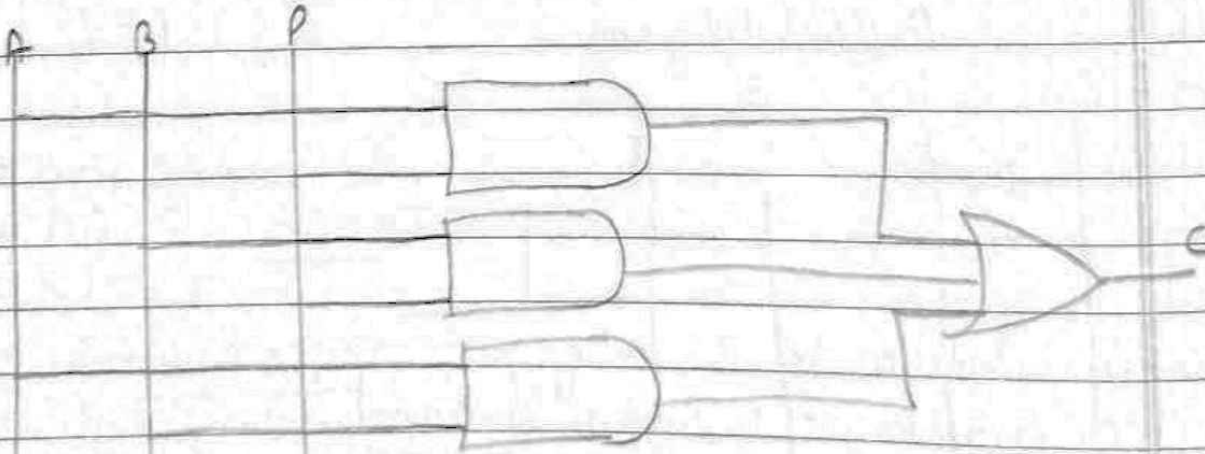
A \ BP	00	01	11	10
0		1		1
1	1		1	

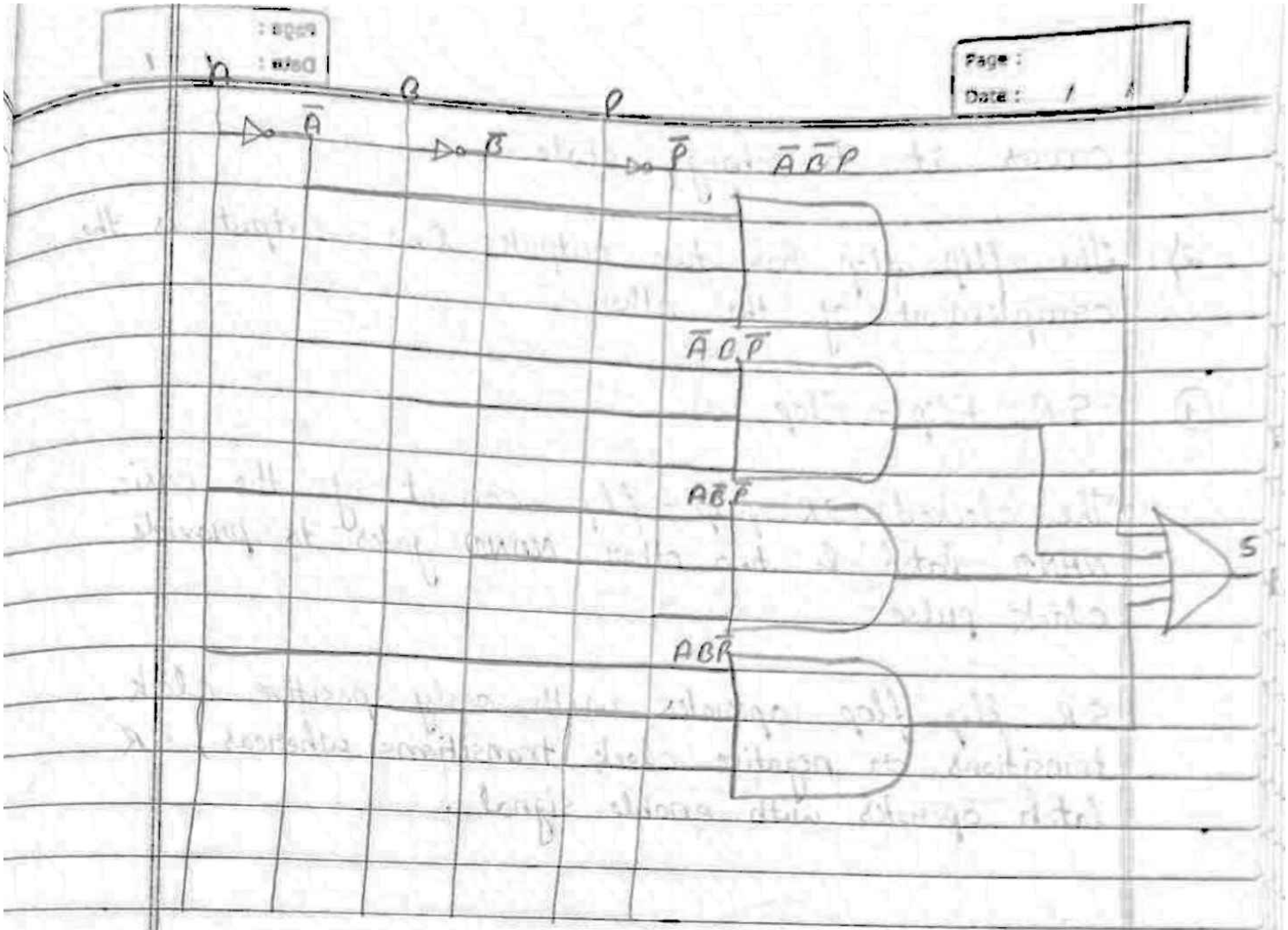
$S = \bar{A}\bar{B}P + \bar{A}B\bar{P} + A\bar{B}\bar{P} + ABP$

C =

A \ BP	00	01	11	10
0			1	
1		1	1	1

$C = AP + BP + AB$





~~Flop~~ Flip flop

Flip flop is the basic memory element in a digital computer

It is used to store one bit of information with a 0 or a 1. It is also called the binary or toggle or catch.

Characteristic of flip flop

① It is a bistable device. The circuit has only two stable states 0 & 1. It responds to inputs of retains its state until some signal

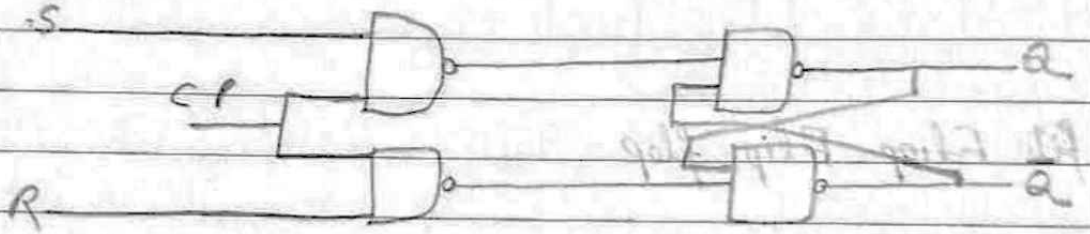
causes it to change state.

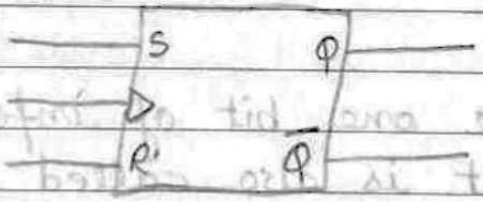
2) The flip flop has two outputs. One output is the complement of the other.

① SR Flip - Flop

The clocked SR flip - flop consist of the basic NAND latch & two other NAND gates to provide clock pulse.

SR flip flop operates with only positive clock transitions or negative clock transitions. whereas, SR latch operates with enable signal.



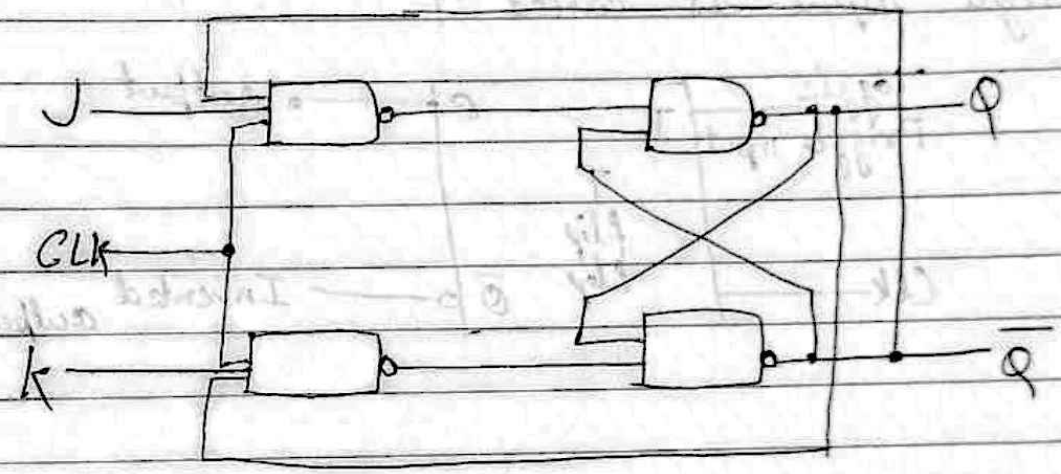


J.K Flip flop

The J.K flip flop is an improvement on the SR flip flop where S=R=1 is not a problem. The input Condition of J=K=1, gives an output

Inverting the output state.

In Simple words, if J & K data input are different (i.e high & low) then the output Q takes the value of J at the next clock edge. if J & K are both low then no change occurs. if J & K are both high at the clock edge then the output will toggle from one state to the other. JK flip flop can function as Set or Reset Flip flop.

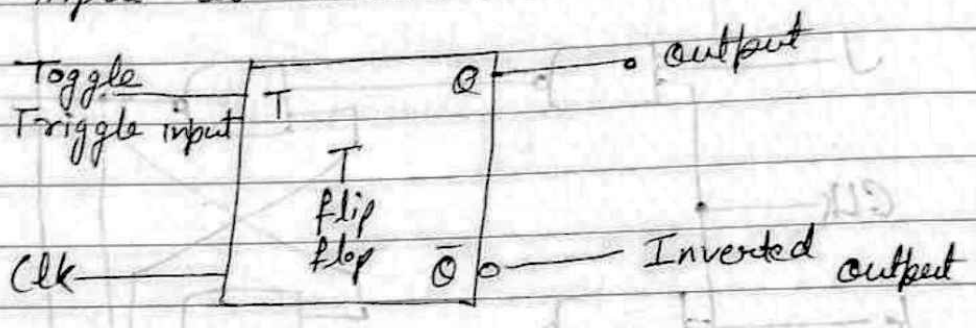


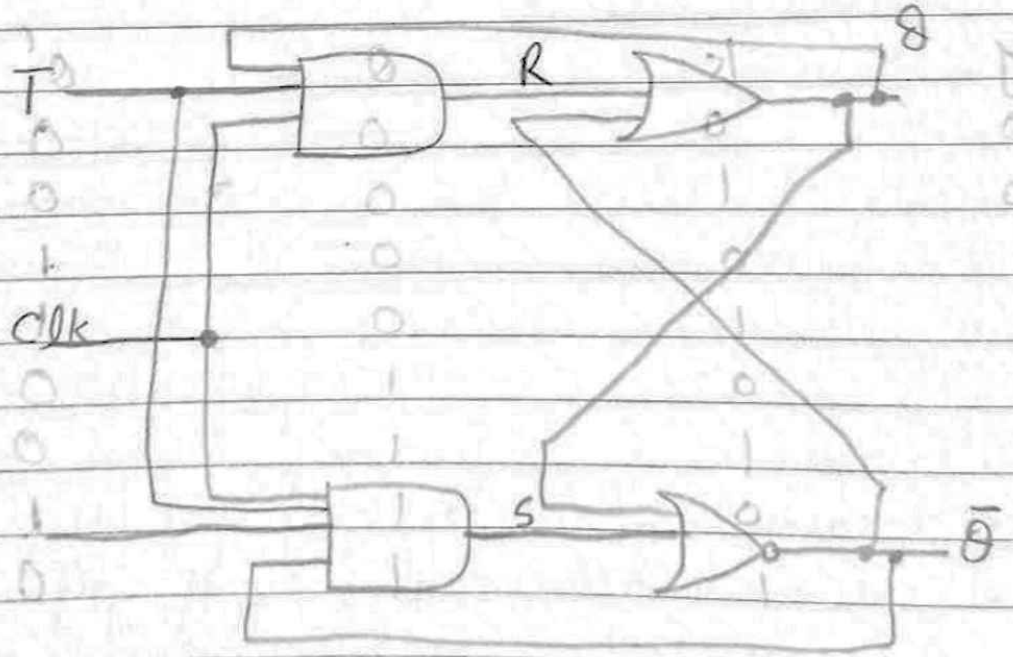
J	K	Q	Q'
0	0	0	0
0	1	0	0
1	0	0	1
1	1	0	1
0	0	1	0
0	1	1	0
1	0	1	1
1	1	1	0

T flip flop

This flip flop work as a toggle switch. The next output state is changed with the complement of the present state output. This process is known as toggling.

we can construct the "T flip flop" by making changes in the "JK flip flop. The T flip flop has only one output, which is constructed by connecting the input of JK flip flop. The Single input is called T





Page :
Date : / /

Outputs

Input	Present	Next State
T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

T	Q^+
0	Q
1	Q'

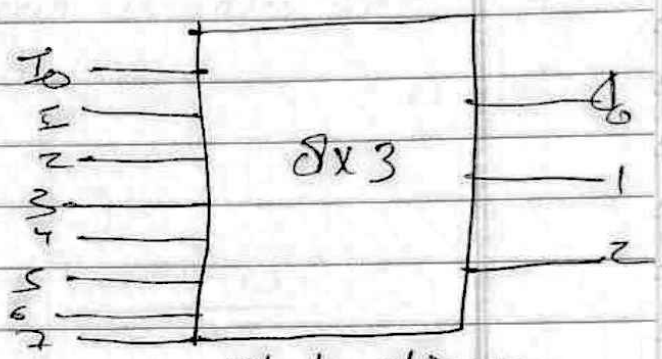
Page :
Date : / /

Encoders

Encoders is a logic circuit which performs the opposite action of the Decoder. An encoder has a number of input lines, only one of which is activated at a time. At the output it displays a value corresponding to the activated input

An encoder has 2^n input lines & n output lines.

(i) Priority
(ii) Decimal to BCD
(iii) Octal to Binary
(iv) Hex to Binary



Block diagram

4x2 Encoder

$$A = \bar{I}_0 I_1 \bar{I}_2 \bar{I}_3 + \bar{I}_0 \bar{I}_1 I_2 I_3$$

$$B = \bar{I}_0 I_1 I_2 I_3 + \bar{I}_0 \bar{I}_1 \bar{I}_2 I_3$$

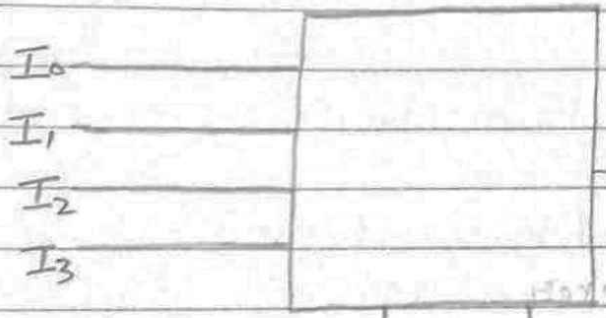
I_0	I_1	I_2	I_3	A	B
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

Page :
Date : / /

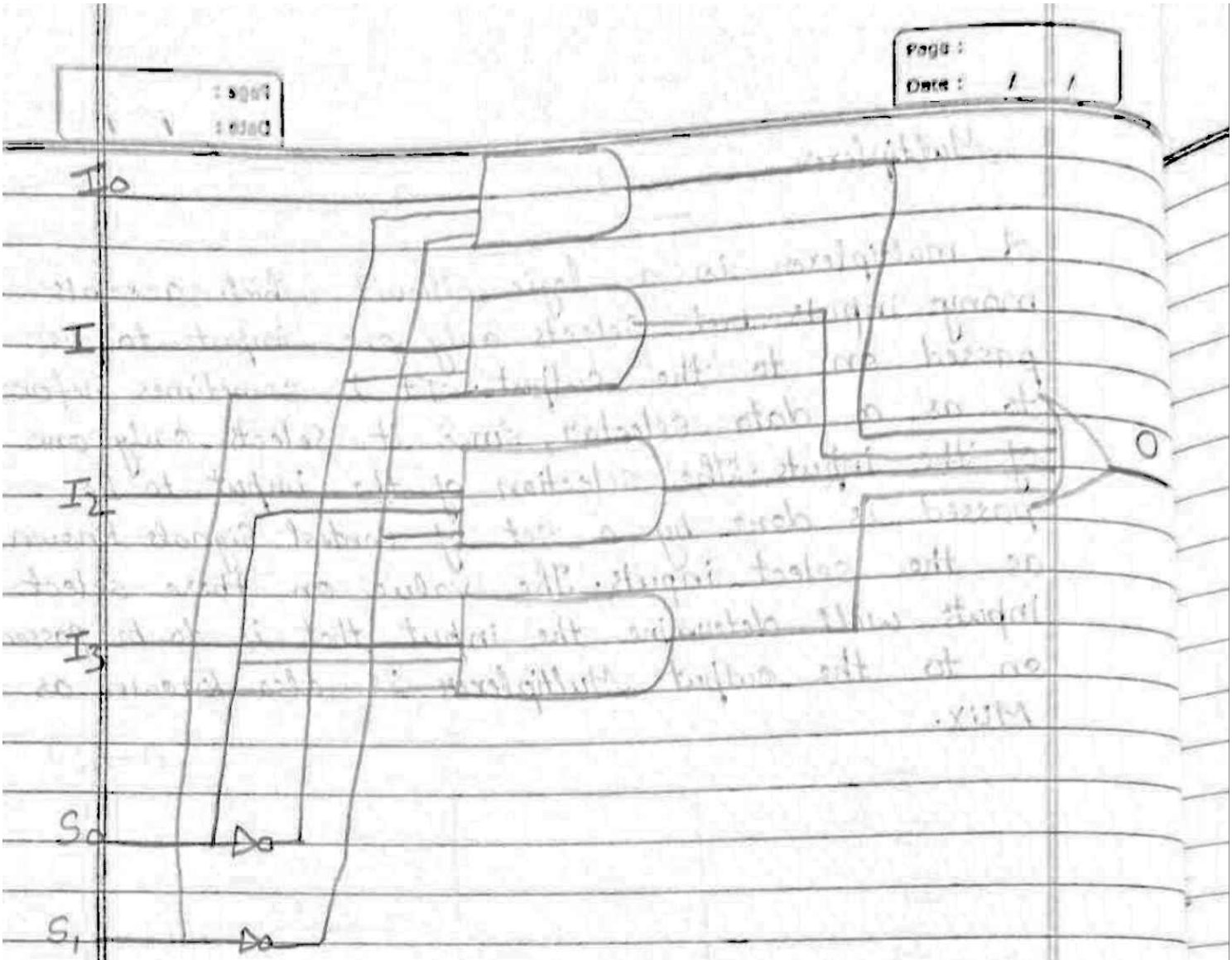
Multiplexer

A multiplexer is a logic circuit which accepts many inputs, but selects only one input to be passed on to the output. It is sometimes refereed to as a data selector, since it selects only one of the inputs. The selection of the input to be passed is done by a set of control signals known as the select inputs. The value on these select inputs will determine the input that is to be passed on to the output. Multiplexer is also known as MUX.

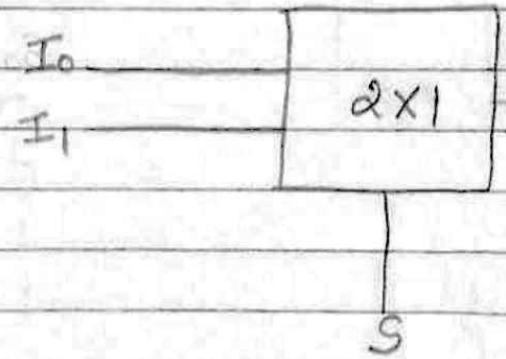
$n = 2^s$



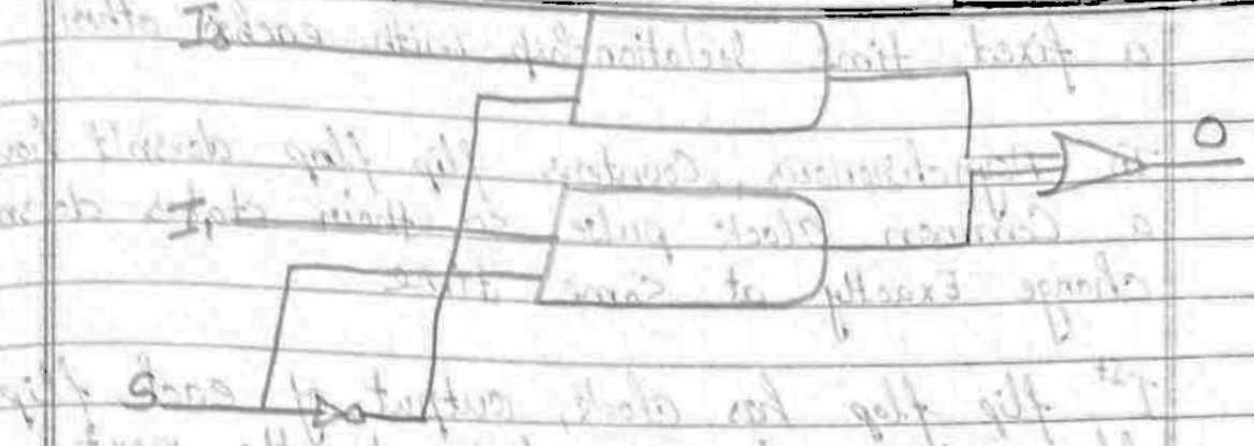
S_0	S_1	O
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3



2 X 1 Multiplexer



S	O
0	I_0
1	I_1



Counter

Counter is a digital device used to count number of pulses & it can also be used as frequency divider

Counter can count in two ways i.e

① up count (0, 1, 2 ... N)
② down count (N, N-1, ... 1, 0)

present count of the counter represents state of counter.

Counter contains set of flip flops, A n bit counter will have N flip flops & 2^n states

Each state frequency ⇒ $\frac{\text{total frequency}}{2^n}$

Asynchronous Counter

Asynchronous refers to states that doesn't have

a fixed time relationship with eachoo other.

In Asynchronous Counters flip flop doesn't have a Common clock pulse so their states doesn't change Exactly at same time.

I^st flip flop has clock, output of each flip flop will acts as clock to the next flip flop in two Counter

Synchronous Counter

In Synchronous Counter, the clock input across all the flip-flops use the same source & create the same clock at the same time. So, a Counter which is using the same clock signal from the same source at the same time is called Synchronous counter

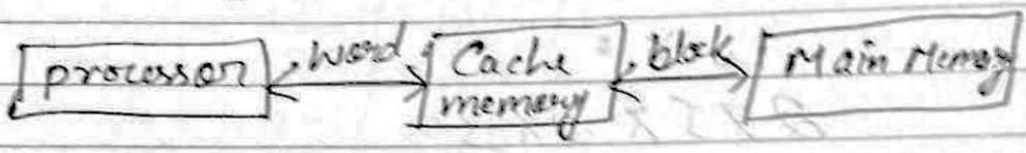
Cache Memory

Cache memory is a small size memory in b/w C.P.U & main memory

C.P.U access should have minimum delay

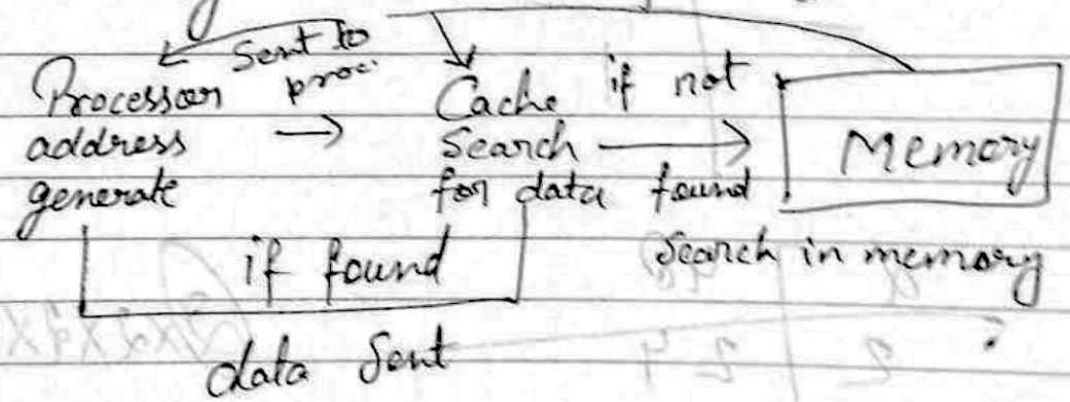
functioning is hidden from programs & users

System having cache provide faster throughput



The transfer of data b/w main memory & cache memory is block by block.

Transfer of data b/w Cache & processor is word by word.



Cache hit :- Address) data is found in Cache

Cache miss :- Address/ data is not found

Cache performance

Hit rate = $\frac{\text{no. of hits}}{\text{total memory reference}}$

miss rate = 1 − hit rate

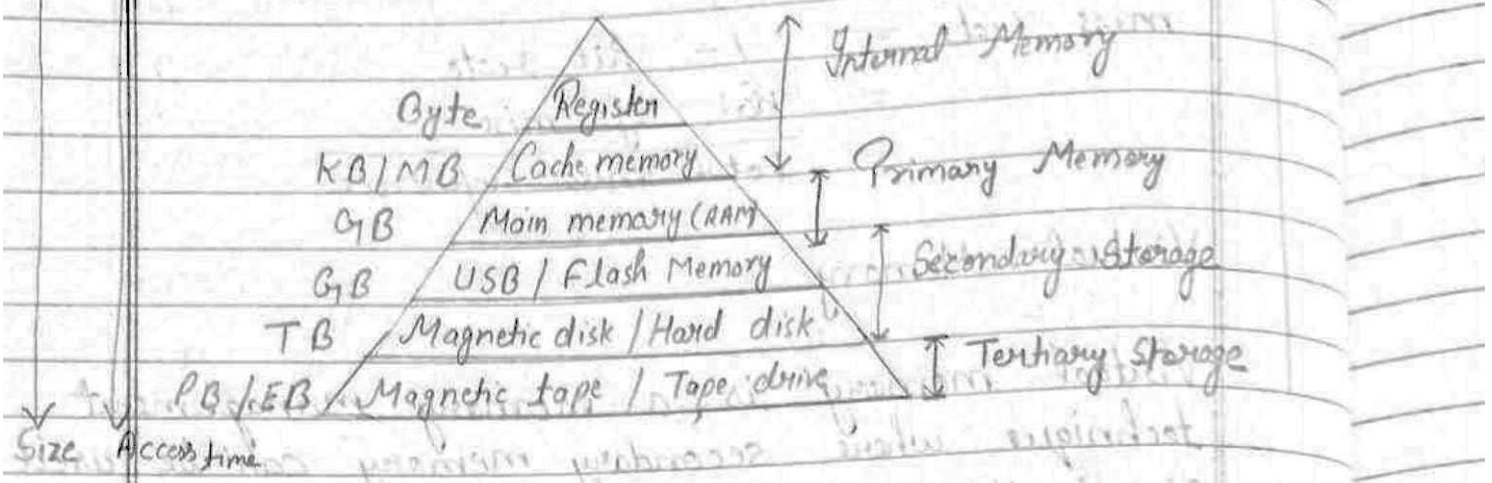
= $\frac{\text{no. of miss}}{\text{total miss reference}}$

Virtual Memory

Virtual memory is a memory management technique where secondary memory can be used as if it were a part of the main memory. Virtual memory is a very common technique used in the operating system of Computer.

Virtual memory uses hardware & Software to allow a Computer to Compensate for physical memory shortages, by temporarily transferring data from random access memory (RAM) to disk storage. In essence, virtual memory allows a computer to treat secondary memory as though it were the main memory.

Memory Hierarchy

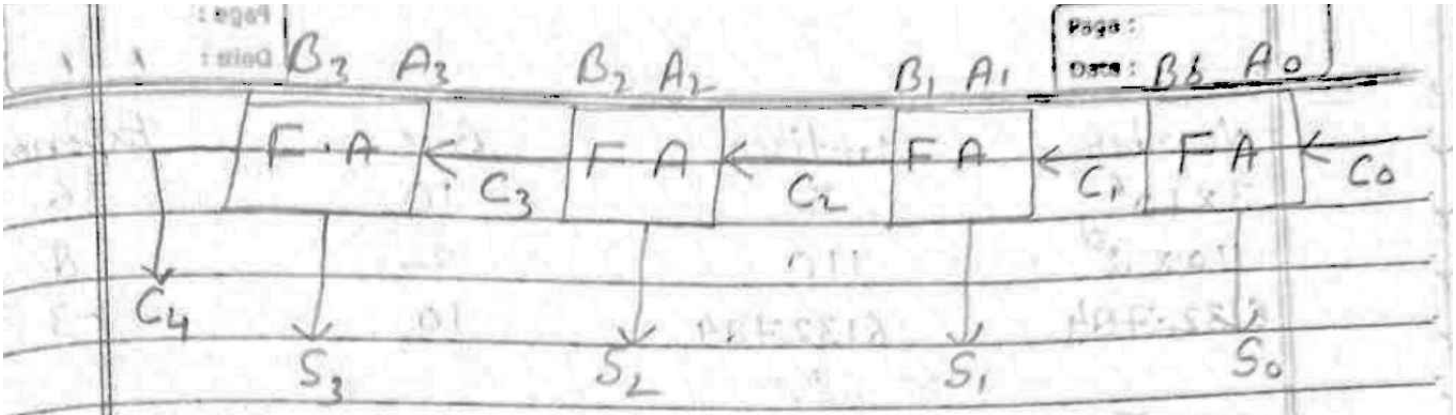


The memory hierarchy separates computer storage into a hierarchy based on response time, Since response time, complexity & capacity are related the levels may also be differciate by their performance & controlling technologies.

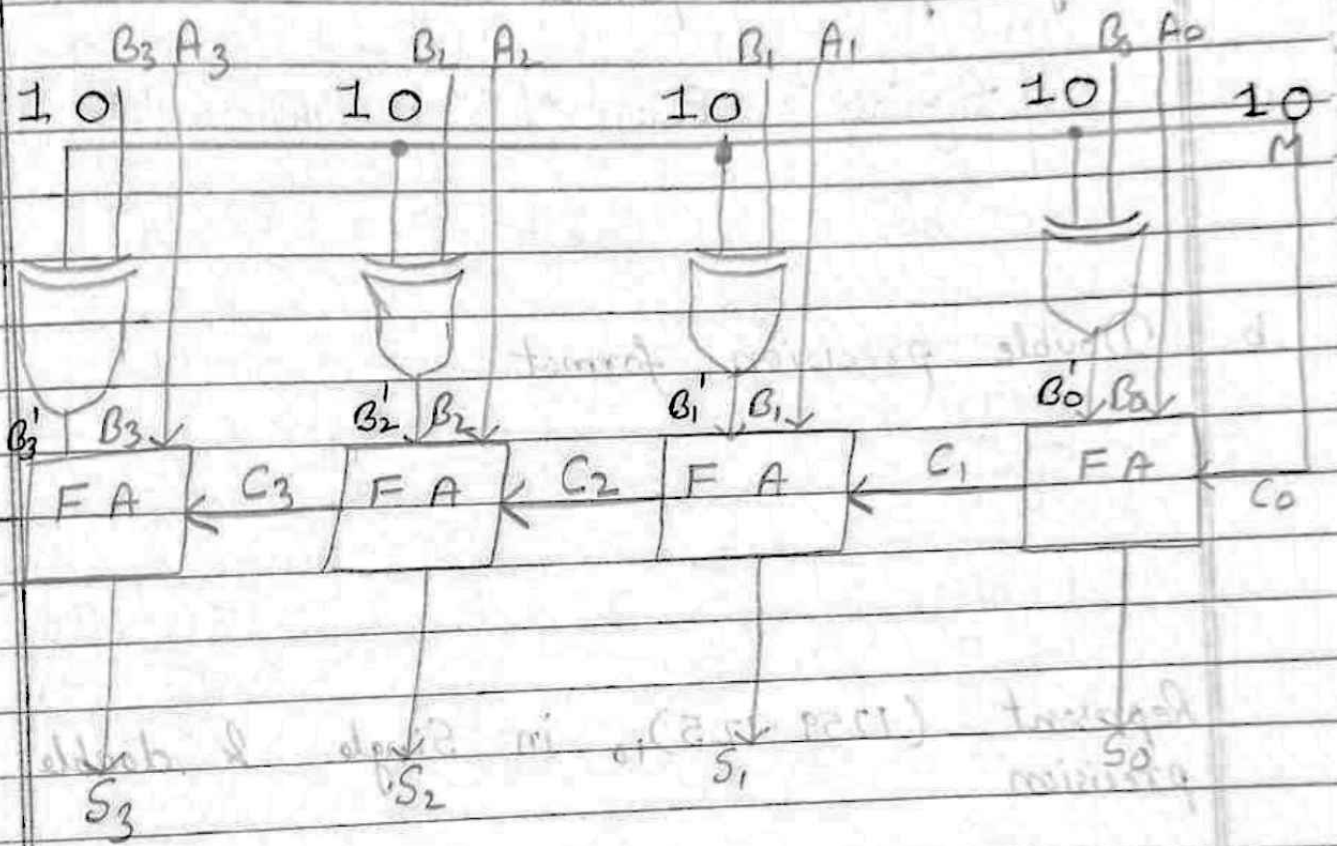
Binary Adder

The digital circuit that generates the arithmetic sum of two binary numbers of any length is called a binary adder

$$\begin{array}{r} {\scriptstyle 1\ 1\ 1\ 1\ } \\ 1\ 1\ 0\ 1 \\ +\ 0\ 1\ 1\ 1 \\ \hline 1\ 0\ 1\ 0\ 0 \end{array}$$



Binary Adder – Subtractor



$M = A + B$: 0 (Addition)

$M = A + \bar{B} + 1$ (Subtraction)

Floating point representation

It has three parts

a Mantissa

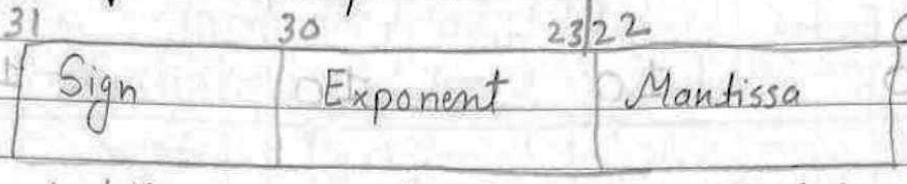
b Base

c Exponent

Number	Mantissa	Base	Exponent
3×10^6	3	10	6
110×2^8	110	2	8
6132.784	6132784	10	-3

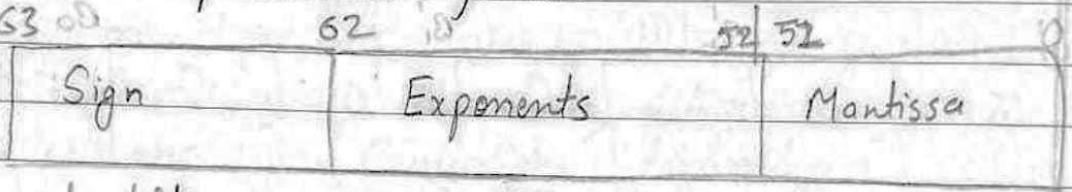
IEEE 754 floating point number representation

a Single precision format



1 bits 8 bits 23 bits

b Double precision format



1 bits 11 bits 52 bits

Represent $(1259.125)_{10}$ in Single & double precision

Step 1:- Convert decimal number to binary number

$(1259)_{10} = (10011101011)_2$

$(0.125)_{10} = 0\ 0\ 1$ → multiply with 2

$(1259.125) = (10011101011.001)_2$

Step 2:- Normalize the number

Date : / /

$(1.N)_2^{E-127}$ → Single precision

$(1.N)_2^{E-1023}$ → double precision

$1|00111010111.001$ →

$(1.00111010111001 \times 2^{10})$ → Normalize no.

Step 3 :- Single precision format

$$(1.N)_2^{E-127}$$

$$1.00111010111001 \times 2^{10}$$

$$E - 127 = 10$$

$$E = 127 + 10$$

$E = 137$ → Exponent

↳ Convert in binary

$$E = (137)_{10} = (10001001)_2$$

31 S	30 E 23	22 M 0
0	10001001	00111010111001...
1 bit	8 bit	23 bit

Step 4 :- Double precision format

$$(1.N)_2^{E-1023}$$

$$1.00111010111001 \times 2^{10}$$

$$E - 1023 = 10$$

$$E = 1023 + 10$$

$$E = 1033$$

$$E = (1033)_{10} = (10000001001)_2$$

63	62	52 51	0
0	10000001001	00111010111001....	
1 bits	11 bits	52 bits	

Instruction Code

An instruction code is a group of bits that instruct the computer to perform a specific operation.

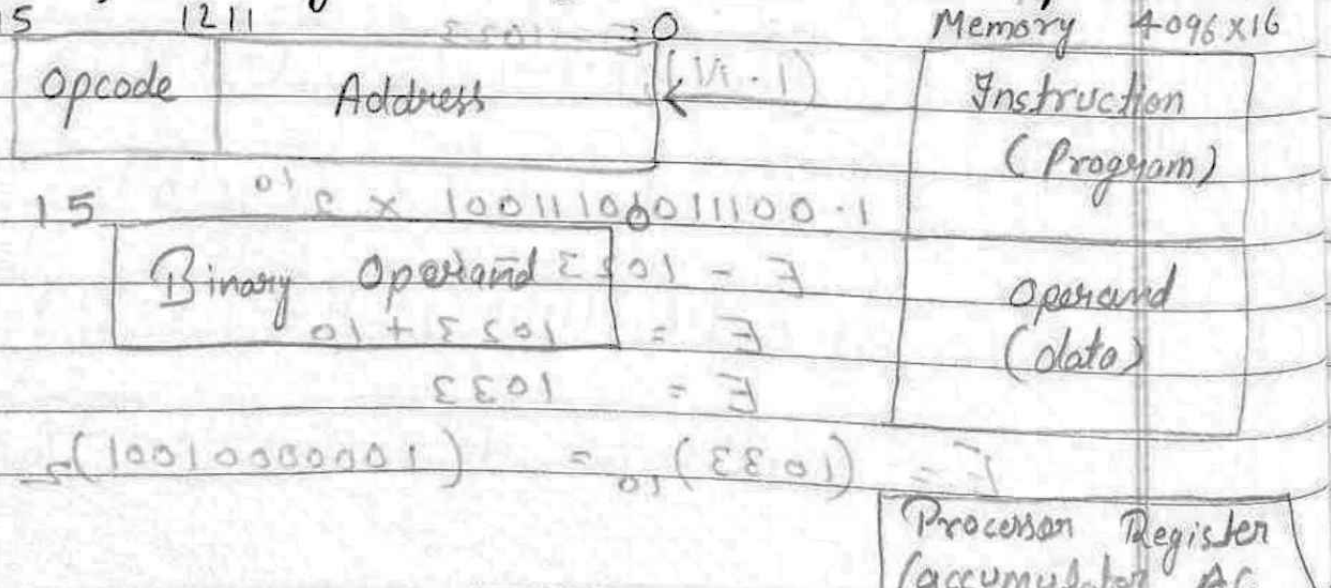
ADD 457.

Operation Code

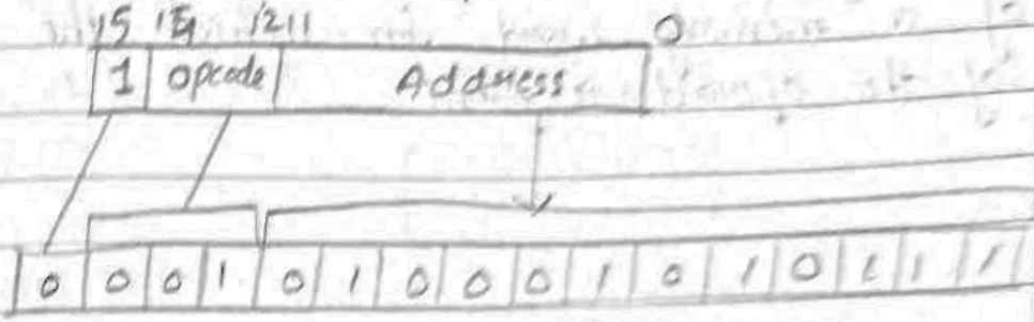
The operation Code of an instruction is a group of bits that define such operation as add, subtract, multiply, shift & complement.

The number of bits required for the operation code of an instruction depends on the total number of operations available in the computer.

The operation code must Consist of at least n bits for a given 2^n (or less) distinct operations.



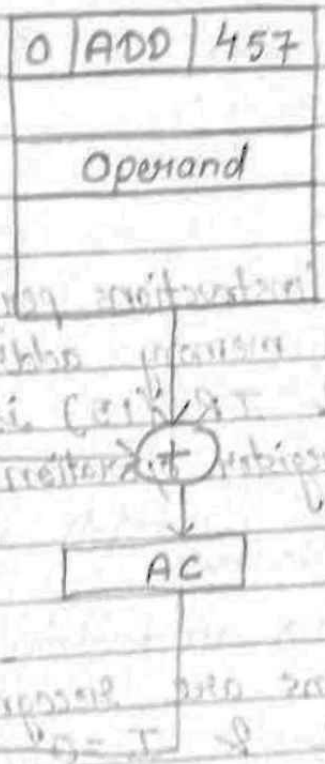
Instruction format



Add instruction - ADD 457

Direct & Indirect Addressing of Memory

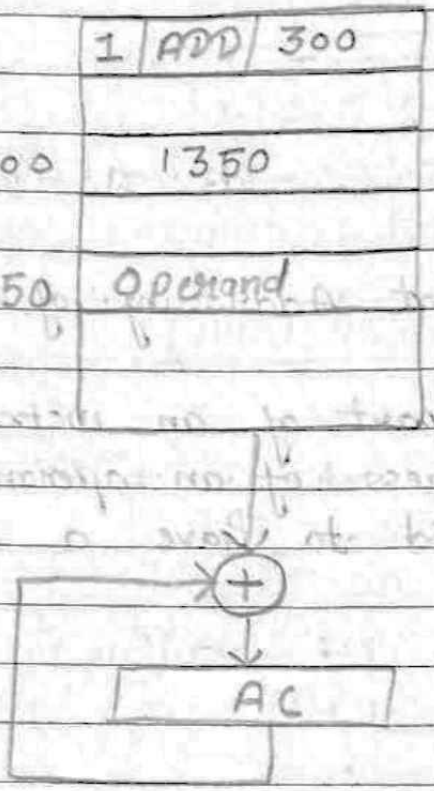
if the second part of an instruction format specifies the address of an operand, the instruction is said to have a direct address



Indirect Addressing

In indirect address, the bits in the second

part of the instruction designate an address of a memory word in which the address of the operand is found.



Register Reference :- These instructions perform operations on registers rather than memory addresses. The IR (14-12) is 111 & IR (15) is 0. The rest 12 bits specify register operation

Register reference instructions are recognized by the control when $D_7 = 1$ & $I = 0$

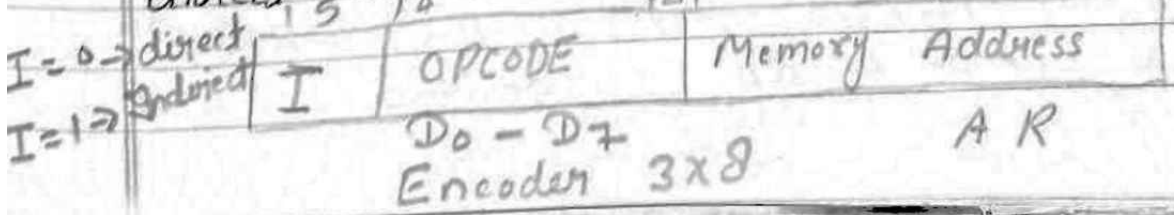
These instruction use bits 0 through 11 of the instruction code is specify operations to be performed.

Register Reference Instruction (Standard Register)

CLA	Clear AC	AC ← 0
CLE	Clear E	E ← 0
CMA	Complement AC	AC ← AC'
CME	Complement E	E ← E'
CIR	Circulate right AC & E	AC ← CIR (AC)
CIL	Circulate left AC & E	AC ← CIL(AC)
INC	Increment AC	AC ← AC+1
SPA	Skip next Instruction if AC positive	if (AC(15)=0) PC ← PC+1
SNA	Skip next Instruction if AC negative	if (AC(15)=1) PC ← PC+1
SZE	Skip next instruction if E = 0	if (E=0) PC ← PC+1
HLT	Halt Computer	

Memory Reference

These instructions refer to memory address as an operand. The other operand is always accumulator. Specifies 12-bits address, 3 bit opcode & 1 bit addressing mode for direct & indirect Addressing.



000 to 110:
$D_0 - D_6$ = Memory reference
D_7 = Register reference / I/p / o/p

D_0 000
AND
AND to AC
$D_0 t_4$: $DR \leftarrow M(AR)$
$D_0 t_5$: $AC \leftarrow AC \wedge DR$, $SC \leftarrow 0$

D_1 ADD
$D_1 t_4$: $DR \leftarrow M(AR)$
$D_1 t_5$: $AC \leftarrow AC + DR$ $SC \leftarrow 0$
$E \leftarrow$ count
↙ Extended AC

D_2 Loading operation (LDA)

$D_2 t_4$: $DR \leftarrow M(AR)$
$D_2 t_5$: $AC \leftarrow DR$ $SC \leftarrow 0$

D_3 StA (Store)
$D_3 t_4$: $M(AR) \leftarrow AC$, $SC \leftarrow 0$

D_4 Branch unconditionally
$D_4 t_4$: $PC \leftarrow AR$ $SC \leftarrow 0$

D_5 BSA (Branch & save return add.)
$D_5 t_4$: $M(AR) \leftarrow PC$, $AR \leftarrow AR+1$
$D_5 t_5$: $PC \leftarrow AR$, $SC \leftarrow 0$

D_6 ISZ { Increment & Skip if Zero }

$D_6 T_4$: DR ← M[AR]

$D_6 T_4$: DR ← DR+1 , M[AR] ← DR
if (DR=0) PC ← PC+1
SC ← 0

What is one address two address & three address instruction.

three field instruction

Opcode	Address	operand	Mode

Opcode
Operation code

Example :-
add (addition), sub (subtraction), mul (Multiplication), mov etc

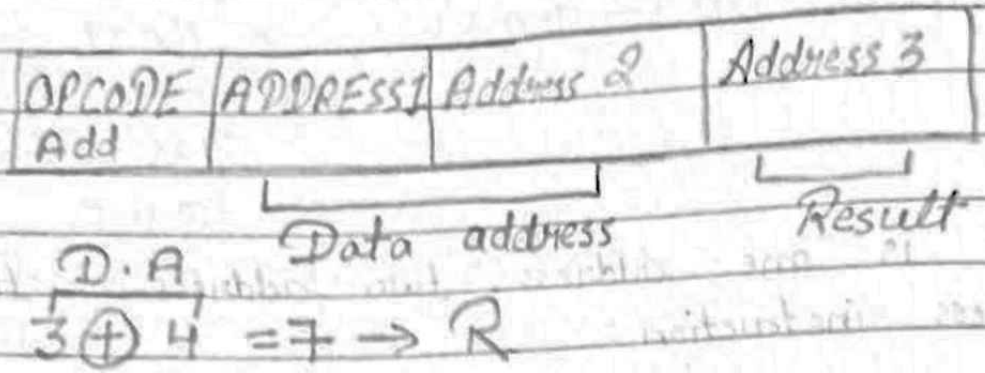
Address of operand (data)

data stored in main memory & its address stored in address field of instruction.

Mode
define way of Calculating address of operand whether it is memory address or register address

Types of Instruction format

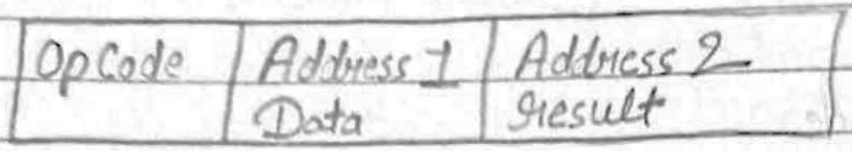
Three address instruction



D.A

3 ⊕ 4 = 7 → R

Two address Instruction



One address Instruction

Result will store in Accumulator Register

What is Interrupt

when a process is executed by the C.P.U and when a user request for another process then this will create disturbance for the running process. This is called as the Interrupt

(i) Internal Interrupt (divide by zero, Register overflow)
(ii) External interrupt (I/O devices)