Date

Page

Computer Organisation

Digital computer is a digital system that performs various computational task. The word digital implies that the information in the computer is represented by variables that take a limited no. of discreate values.
In practice digital computer function more reliably if only two states are used i.e. (0 & 1) or (true or false) or (Yes/No).

Digital computer use the binary number system which has 2 digits i.e. 0 and 1.
A binary digit is called a bit. Information is represented in digital computers in group of bits by using various coding techniques groups of bit can be made to represent not only binary numbers but also other discrete symbols such as decimal digits or the letters of the alphabet.

In computer system is some times sub-divided in to functional entities.

Software and hardware. The hardware of the computer consist of all the electronic components and electrochemical devices that comprise the physical entity of the device.

Computer software consist of the instructions and data that computer manipulates to perform various data processing task.

A sequence of instructions is called a programme

The data that are manipulated by the programme constitute the database.

Computer Architecture

1. Von Neuman model :- It was designed and developed by Mathematician and Researcher Jhon Von Neuman in 1945. Most of the computers of today's world run on it.

This Architecture is based on the store programme computer concept where instructions data and programme data are stored in the same memory. It is also called stored memory architecture.

There are three main component of Von Neuman model:-

1. CPU :- The CPU is the electronic circuit responsible for executing the instructions of a computer programme. It is sometimes referred to a microprocessor or processor. The CPU contains control unit, Arithmetic unit and variety of registers.

i) CU (control unit) - Controls the operation of the computers ALU, memory & I/O devices, telling them how to respond to the programme instructions it has just read

and interpreted from the memory unit. The control unit also generates timming and control signals required by other components.

ii) AU → (Arithmetic unit) performs the arithmetic & logical operations like addition, subtraction, AND, OR, NOT.

iii) Register :- It is a high speed storage area in the CPU all data must be stored in a register before it can be processed. They are small in size but fast in response.

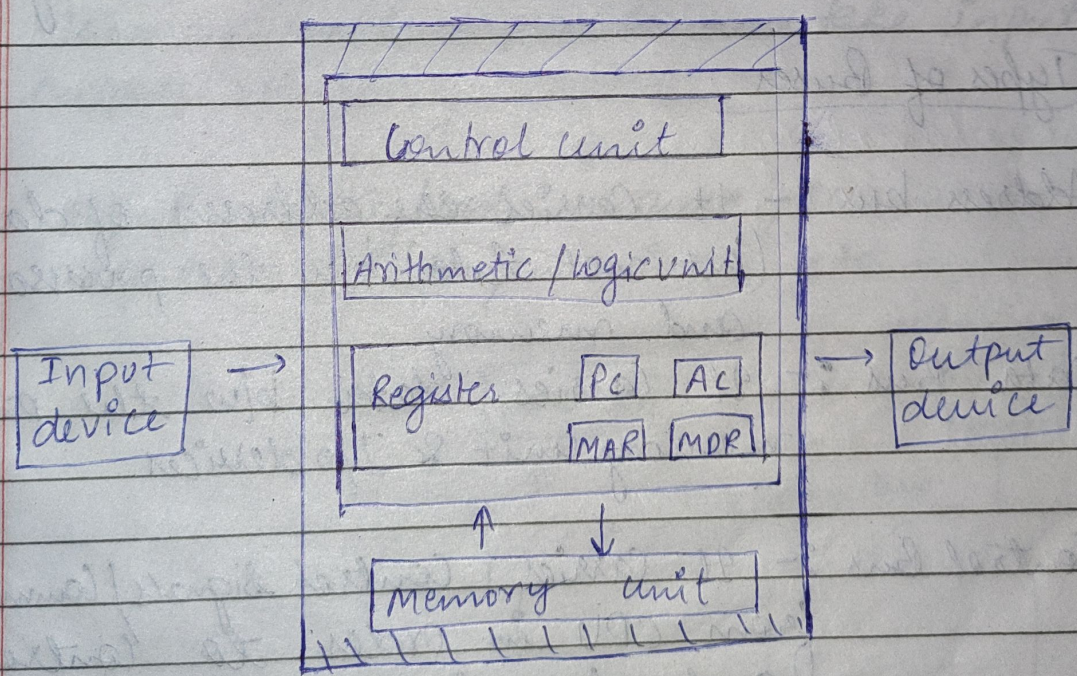


Fig :- Schematic/block diagram of Von Neuman model.

Date Page

→ MAR (memory address register) :- It holds the memory location of data that needs to be accessed.

→ MDR (Memory data Register) :- It holds data that is being transfered to or from memory to and from → read & write

→ AC (Accumulator) → It holds intermediate arithmetic and logical results

Buses are the means by which data is transmitted from one part of a computer to another, connecting all major ~~important~~ internal components ~~wit~~ to the CPU and memory.

Types of Buses

1. Address bus :- It carries the addresses of data (but not data) b/w the processor and memory.
2. Data Bus :- It carries data b/w the processor memory unit & I/O devices.
3. Control Bus :- It carries control signals/commands from CPU in order to control and coordinate all the activities within the computer.

Date

Page

2. Memory unit

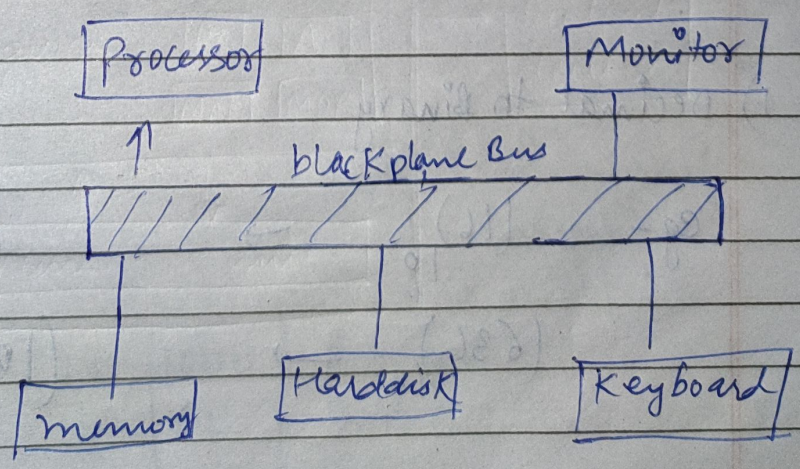
1. Memory unit Consist of RAM sometimes ~~to~~ refers to as primary or main memory unlike a harddrive (Secondary memory) this memory is fast and directly accessable by CPU

- RAM is split into partition. each partition Consists of an address and its ~~contains~~ content (both in binary form)
The address will uniquely identify every location in the memory.

3. Input/output device

These are used to provide and take input & output respectively

eg- Input → Keybord, mouse, scanner etc
Output → printer, monitor etc.



Date Page

Basic organization of a Computer System

Data Representation :-

Number System :- It is the system of base or radix (r) which have discreate symbols to represent.

1. Binary System => $r = 2$
   0 to $r-1$ = 0, 1

2. Decimal system => $r = 10$
   0 to $r-1$ = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

3. Octal system => $r = 8$
   0 to $r-1$ = 0, 1, 2, 3, 4, 5, 6, 7

4. hexadecimal system => $r = 16$
   0 to $r-1$, = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ...
   A, F

Conversion -

1) Decimal to Binary

eg- $(16)_{10} \rightarrow (10000)_2$

$(636)_{10} \rightarrow (1001111011)_2$

2) Binary to decimal

$(01100111)_2$ ~~= 16+8+0+0+1~~ $= (103)_{10}$

$(745.25)_{10} =$

2^{-1} 2^{-2}

~~745~~

$0.25 \times 2 = 0.50$ 0

$0.50 \times 2 = 1$ 1

$(745.25)_{10} \rightarrow (1011100110.01)_2$

3) Decimal to octal

$(235)_{10} = (353)_8$

4) octal to decimal

$(353)_8 =$

$8^0 \times 3 = 3$

$8^1 \times 5 = 40$

$8^2 \times 3 = 192$

$= 192 + 40 + 3 = (235)_{10}$

5) Decimal to hexadecimal

$(1000)_{10}$

16	1000	8
16	62	8
	3	E

$(1000)_{10} = (3E8)_{16}$

octal to binary

$(745)_8 = (111100101)_2$

Octal	
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

binary to octal

1	2	3	4
001	110	101	101
1	6	5	5

$= (1655)_8$

Complement : Complements are used in a digital computer for simplifying subtraction process and logical manipulation.

There are two types of complement's in each "r" system i.e. r's complement & (r-1)'s complement

$(\ )_{10}$	$(\ )_8$	$(\ )_2$
10's complement	8's	2's
9's complement	7's	1's

Date _____ Page _____

eg - 73350

```
  99999
  72350
  -----
  27649   → 9's compliment
     +1
  -----
  27650   → 10's compliment
```

Q using 9's compliment subtract
72352 - 3250 ~~(M<N)~~ (M>N)

```
  99999
- 03250
  -----
  96749
```

```
   72352
 + 96749
 -------
1)69101(
```

indicate positive ← add +1 to obtain final value

72352 - 3250 = 69102

Q. 3250 - 72352 (M<N)

```
  99999
- 72352
  -----
  27647   9's comp

  03250
+ 27647
  -----
  30897
```

No end carry

No end carry
∴ result is negative
to obtain final result
make 9's com of 30897

```
  99999
  30897
  -----
  69102
```

Ans = -69102

Q. Subtract : 69864 - 2432 (M>N)

99999
02432
97567

69864
97567
1)67431(

Ans = 67432

Q. 2432 - 69864

99999
69864
30135

30135
02432
32567

99999
32567
67432

Ans = -67432

Date

Page

Q. Using 10's Compliment subtract 72352 − 3250

99999
03250
96749
+1
96750 − 10's Compliment

Now add 10's comp to 72352

96750
+ 72352
1)69102 → final answer

indicate positive

Q. Using 10's comp sub → 3250 − 72352

99999
72352
27647
+1
27648 − 10's comp

03250
+ 27648
30898

99999
30898
69101
+1
− 69102

Ans

Date

2's Compliment

```
  111111
- 010110
  101001   - 1's compliment
     +1
  101010   - 2's compliment
```

Q. Using 1's compliment 1010101 − 1000011 (m>N)

```
  0111100   - 1's comp
+ 1010101
  0000001 -
     +1
  0000010   - final Answer
```

Subtract 1000011 − 1010101 (m<N)

```
  0101010   - 1's comp
+ 1000011
  1101101
  0010010   - 1's comp to obtain final result
```

Ans -

Date

Page

Q. Using 2's sub 1010101 − 1000011 (m>N)

make 2's comp - 1000011.

0111100 1's comp

+1

0111101 = 2's comp

~~1010101~~

~~0111101~~

1010101

+ 0111101

10010010

Ans → 0010010

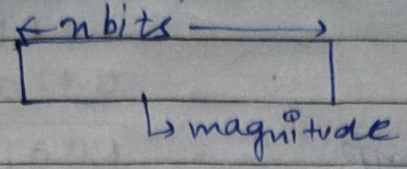
Binary number representation

For positive number only

for both +ve & -ve numbers

- unsigned representation — unambiguous
- signed representation
  - 2's complement — unambiguous
  - Sign magnitude form — Ambiguous
  - 1's complement form — Ambiguous

1. Unsigned Representation:-

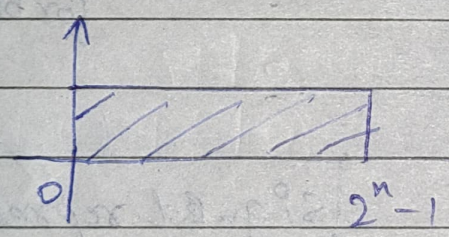


If $n = 4$

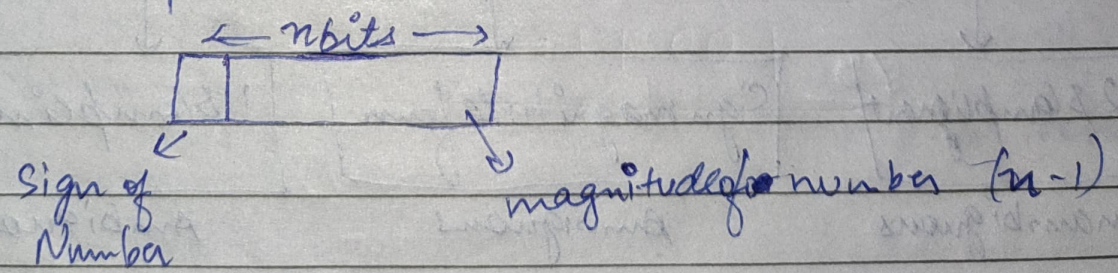
0000 = 0 — 0
0001 = 1
0010 = 2
⋮
1111 = 15 — $2^n - 1$

Range = 0 to $2^n - 1$

Range Specification graph



2. Signed representation



If 0 = +ve
1 = -ve

Range → $-(2^{n-1} - 1)$ to $+(2^{n-1} - 1)$

Date _____ Page _____

1's Compliment form :-

Decimal	Sign magnitude	1's Compliment	2's Compliment
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
+0	0000	0000	0000
-0	1111	1111	10000
-1	1001	1110	0001
-2	1010	1101	1110
-3	1011	1100	1101
-4	1100	1011	1100
-5	1101	1010	1011
-6	1110	1001	1010
-7	1111	1000	1001
			1000 = -8

2's Compliment

Range - $[-2^{n-1}$ to $+(2^{n-1}-1)]$

Date _____ Page _____

Addition/Subtraction of Signed binary number

The addition of two signed binary numbers represented in signed is complement is obtained by adding the numbers including their sign bit.

* The carry out to the signed bit is discarded in the addition.

Q. Perform (+6) + (+13) in sign 2' complement using 8 bit.

$+6 \xrightarrow{2'\text{ complement}} 00000110$

$+13 \xrightarrow{2'\text{ complement}} + \; 00001101$

$00010011 \rightarrow (+19)$

Q. (+6) + (-13)

$+6 \xrightarrow{2'\text{ complement}} 00000110$

$-13 \xrightarrow{2'\text{ complement}} 11110011$

$+$

11111001

1111001

$+1$

0000110

$+1$

0000111

Ans = 0000111

Date _____ Page _____

Q $(-36)-(+23)$ by using 8 bit

$(-36) \xrightarrow{2'\text{ Com}}$ ~~0001110 0~~ 11011100

$(+23) \longrightarrow$

Note If $C_{in} \neq C_{out}$ then overflow exist

~~$C_{n-1} \neq C_n$~~

$(+42)+(-13)$

$+42 \xrightarrow{2'\text{ Complement}}$

Q $(-42)-(-13)$

$(11010110) - (11110011)$

$(11010110) + [-(11110011)]$

$(11010110) + (00001101)$

$= 11000111$

00111100

$+1$

Ans $\longrightarrow$ 00111101

Date

Page

Q

$(6AB.CD)_{16} = (\quad)_8$

H	B
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

$\underbrace{0110}_{3}\ \underbrace{1010}_{2}\ \underbrace{1011}_{5}\ \underbrace{1011}_{3}\ .\ \underbrace{1100}_{6}\ \underbrace{1100}_{3}\ \underbrace{110}_{2}$

$(3253.632)_8$

Date

Page

Floating point number representation

Floating point representation of a number has a two parts

1) Mantissa – It is a signed fixedpoint number or a fraction
2) Exponent – It designates the position of decimal or the binary point

1. Single precision method (32 bit)
2. Double precision method (64-bit) (biased exponent)

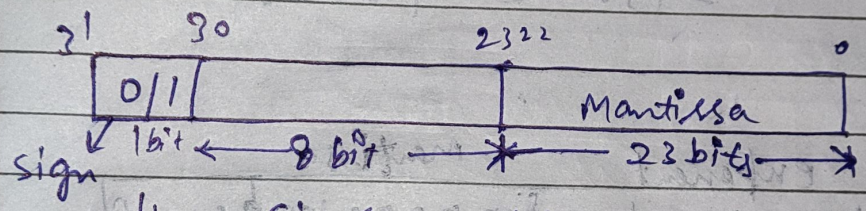
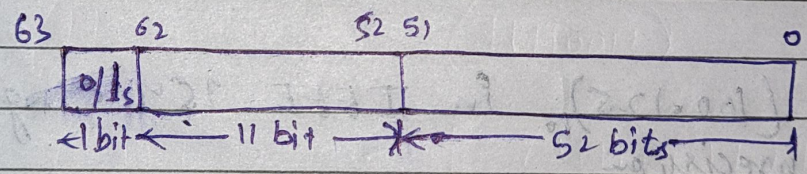


fig- Single precision



Q Represent 85.125 in IEEE 32 bit format

Step-1 Convert decimal into binary

$$85.125 = 1010101.001$$

Step-2 Normalised form.

$$1.010101001 \times 2^{+6}$$
$$= (1.M) \times 2^{E}$$

$$E' = E + 127$$
$$= 6 + 127 = 133$$

Make binary of 133 = 10000101

Q 8.125

1.) 1000.001

2. Nomalize

$= 1.000001 \times 2^{+3}$

$\Rightarrow (1.M) \times 2^{E}$

3. $E' = 3 + 127$

$= 130 = 10000001$

Sign	exponent	Mantissa	
0	10000001	00000100	0

Q Represent $(100.125)_{10}$ in IEEE 754 single & double precission

$(1100100.001)_2$

$1.100100001 \times 2^{+6}$

$= (1.M) \times 2^{E}$

$E' = 6 + 127$

$= 133 = 10000101$

Date _____ Page _____

Sign	exponent	mantissa
0	10000101	10010000100 . . . 0
← 1bit →	← 8bit →	← 23bit →

single precision

Sign	exponent	
0	10000101000	10010000100 . . . 0
← 1bit →	← 11bit →	← 52bit →

for double precision

$E' = 6 + 1023$

$E' = 1029$

Date

Combinational circuit

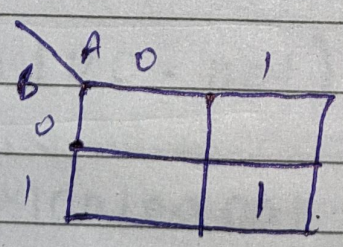
logical circuit for developing any digital System has two parts

a) Combinational circuits
b) Sequential circuits

Adder is a combinational circuit which performs the addition of numeric values
Two types of Adder. —

1) Half adder – It a combinational circuit to add two bits only and provide Sum and carry as output.

A	B	Sum	carry
0	0	0	0
0	1	1	0
1	0	0	1
1	1	0	1



$$SOP = \bar{A}.B + A\bar{B}$$
$$= A \oplus B$$

Fulladder is a combinational circuit to add three bits.

A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1
0	1	1	0	1

A \ BC	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	$B\bar{C}$ 10	BC 11
$\bar{A}$ 0		1	1	
A 1	1			1

$$\text{Sum} = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

$$= \bar{A}(\bar{B}C + B\bar{C}) + A(\bar{B}\bar{C} + BC)$$

$$= \bar{A}\cdot(B \oplus C) + A\,\overline{(B \oplus C)}$$

$$\cancel{= A \oplus B \oplus C}$$

let $B \oplus C = z$

$$= \bar{A}\cdot z + A\bar{z}$$

$$= A \oplus z$$

$$= A \oplus B \oplus C$$

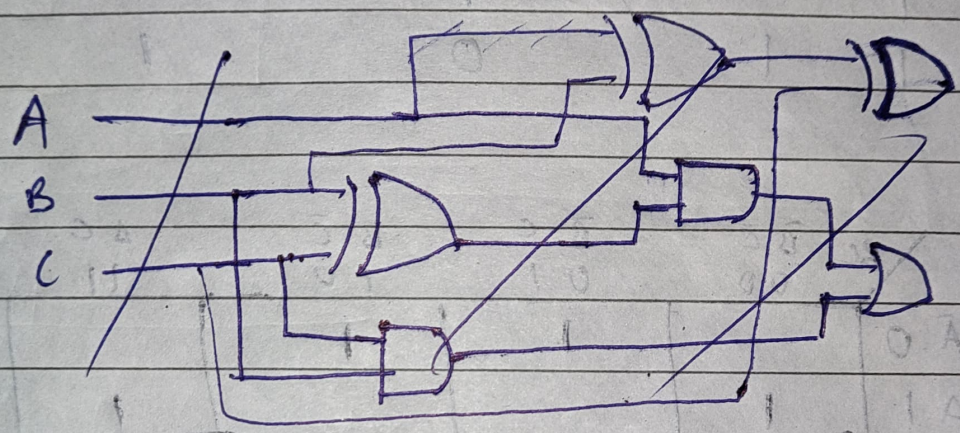
	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	$B\bar{C}$ 10	BC 11
$\bar{A}$ 0				1
A 1		1	1	1

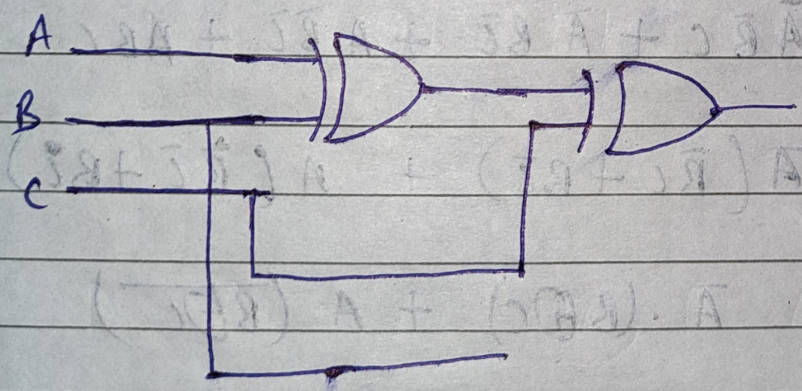
$$= \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

$$= BC(\bar{A}+A) + A(\bar{B}C + B\bar{C})$$

$$= BC + A\cdot(B \oplus C)$$

$$\cancel{= A(BC + B \oplus C)}$$

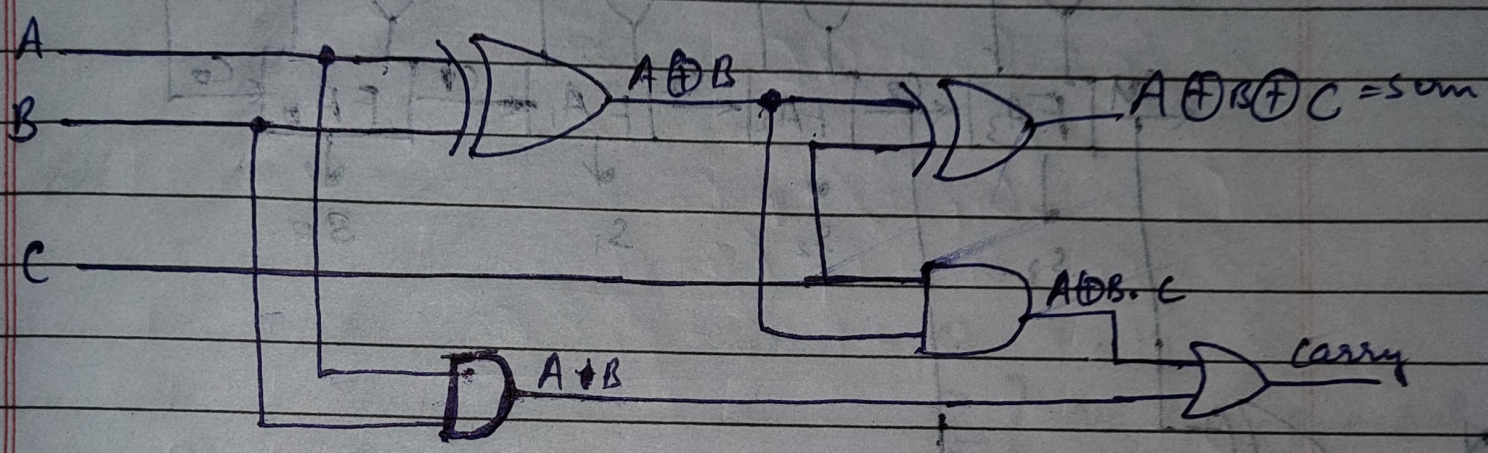


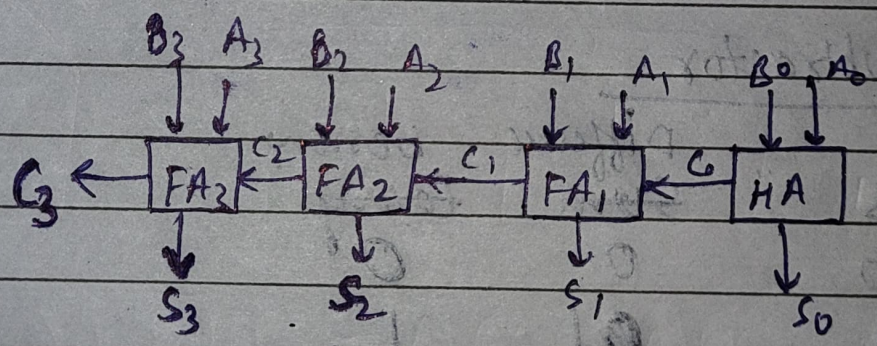


Date _____ Page _____

Q-1 Implementation of FA using two Half adder.
Q-2 Implementation of HA using NAND gate only.

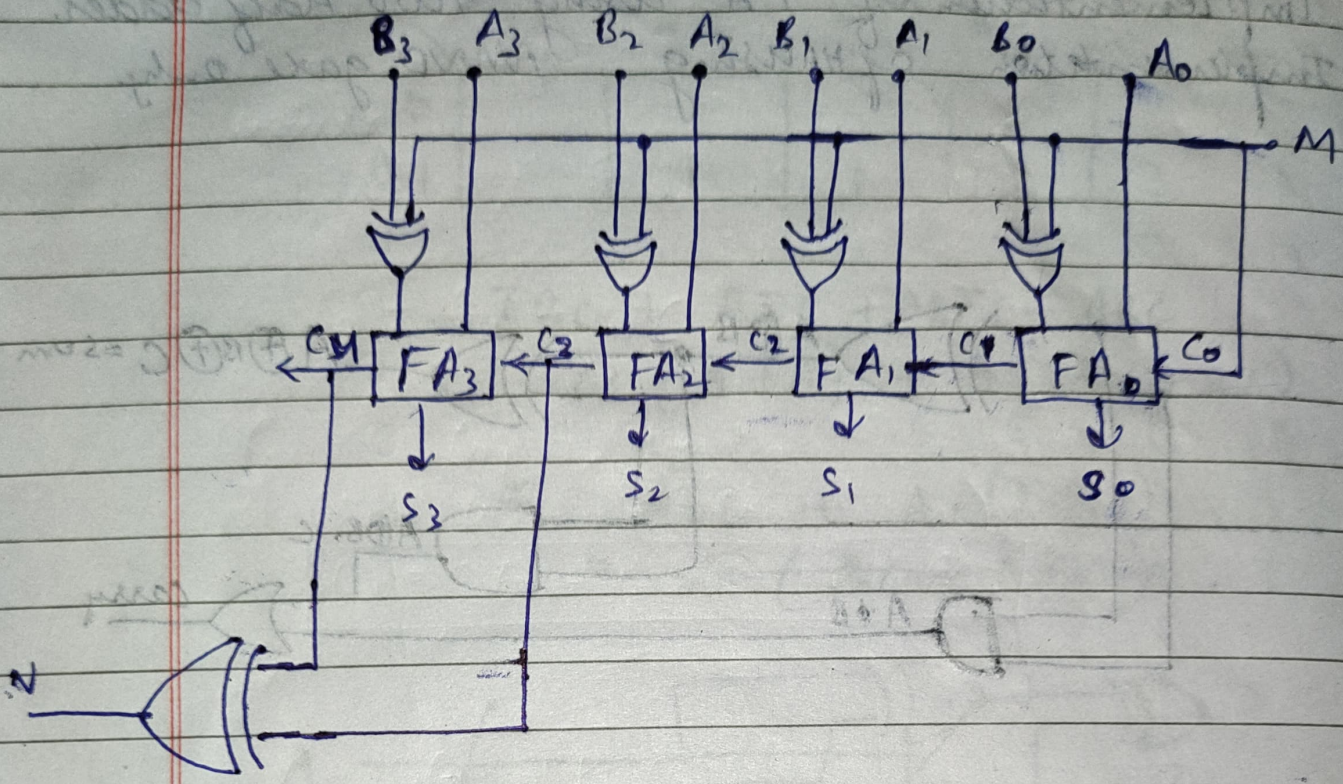
Q-1





4 bit parallel/Ripple carry

M=0 (binary adder)
M=1 (binary subtractor)



Half Subtractor

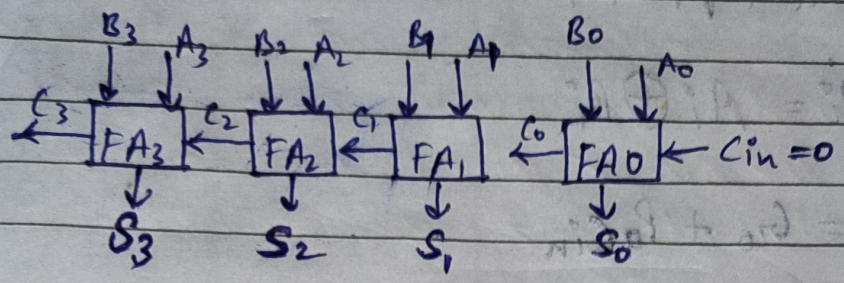
A	B	Difference ~~Sum~~	borrow ~~carry~~
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$D = \bar{A}B + A\bar{B} = A \oplus B$, $B = \bar{A}B$

Date

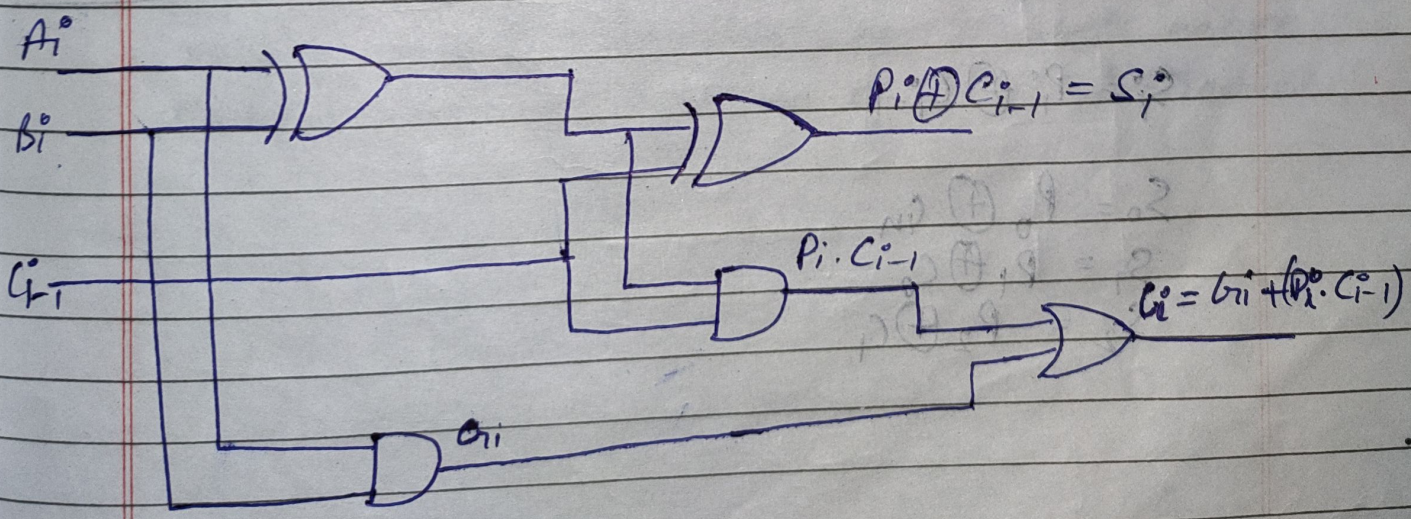
Page

Carry look Ahead (CLA) {Superior or faster adder}



~~Ad~~

A_i	B_i	C_i	S	C_o	
0	0	0	0	0	No carry generated i.e $C_{out} = 0$
0	0	1	1	0	
0	1	0	1	0	carry propagated i.e $C_{in} = C_{out}$ $P_i = A_i \oplus B_i$
0	1	1	0	1	
1	0	0	1	0	
1	0	1	0	1	
1	1	0	0	1	Carry generated $G_i = A_i \cdot B_i$
1	1	1	1	1	



$C_i = G_i + P_i C_{i-1}$

where,

$P_i = A_i \oplus B_i$

$C_0 = G_0 + P_0 C_{in}$

$C_1 = G_1 + P_1 C_0$

$C_1 = G_1 + P_1 (G_0 + P_0 C_{in})$

$\boxed{C_1 = G_1 + P_1 G_0 + P_1 P_0 C_{in}}$

$C_2 = G_2 + P_2 C_1$

$\boxed{C_2 = G_2 + P_2 G_1 + P_1 P_2 G_0 + P_2 P_1 P_0 C_{in}}$

$C_3 = G_3 + P_3 C_2$

$\boxed{C_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_{in}}$

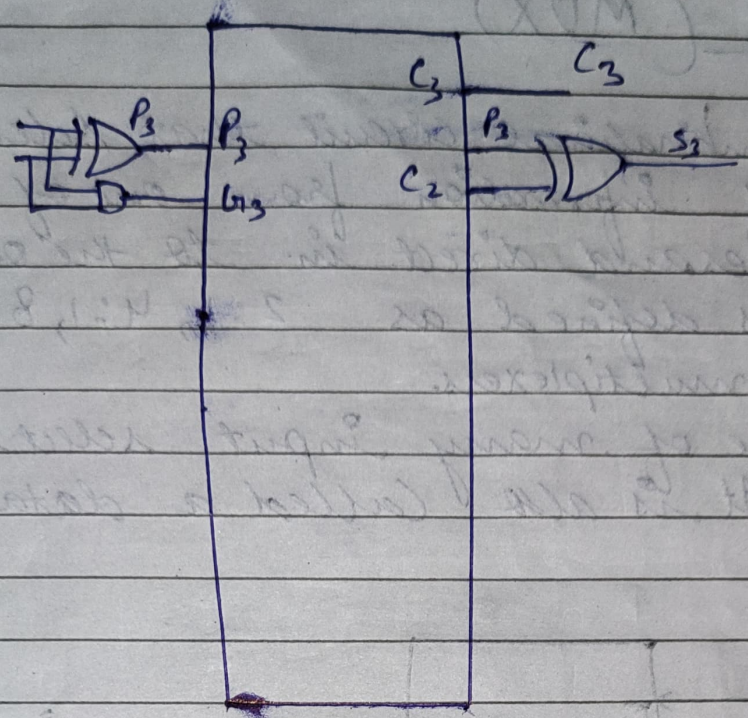
$S_i = P_i \oplus C_{i-1}$

$S_0 = P_0 \oplus C_{in}$

$S_1 = P_1 \oplus C_0$

$S_2 = P_2 \oplus C_1$

Date Page



Advantages

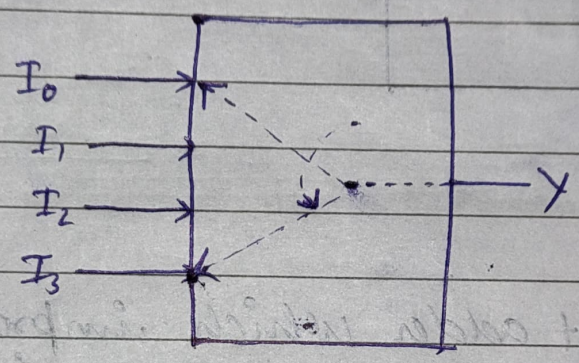
CLA is a fast adder which improve speed by reducing the response time (time required to calculate carry bit)

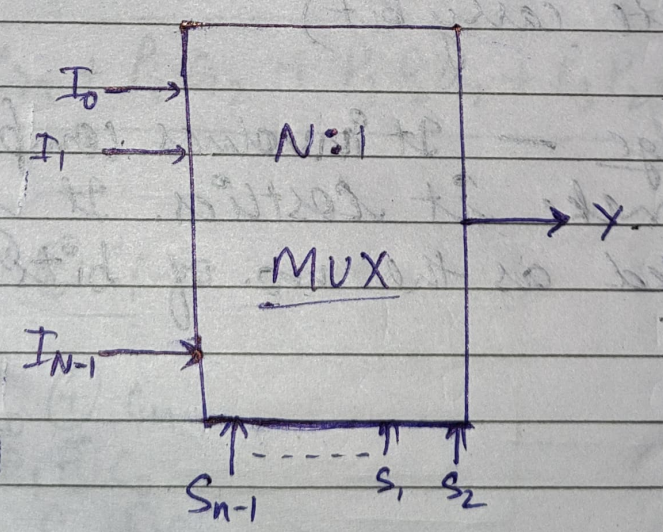
disadvantage — It involves complex hardware which make it costlier. It gets more complicated as the no. of bits increases.

Multiplexer (MUX)

It is a combinational circuit that select binary of information from one of many input lines and direct in to the output line. It is defined as 2:1, 4:1, 8:1, 16:1, 32:1 etc. multiplexers.

To select one of many input select input is used. It is also called a data selector.





Select lines = m

No. of input = N

$$\boxed{N = 2^m} \qquad m = \log_2 N$$

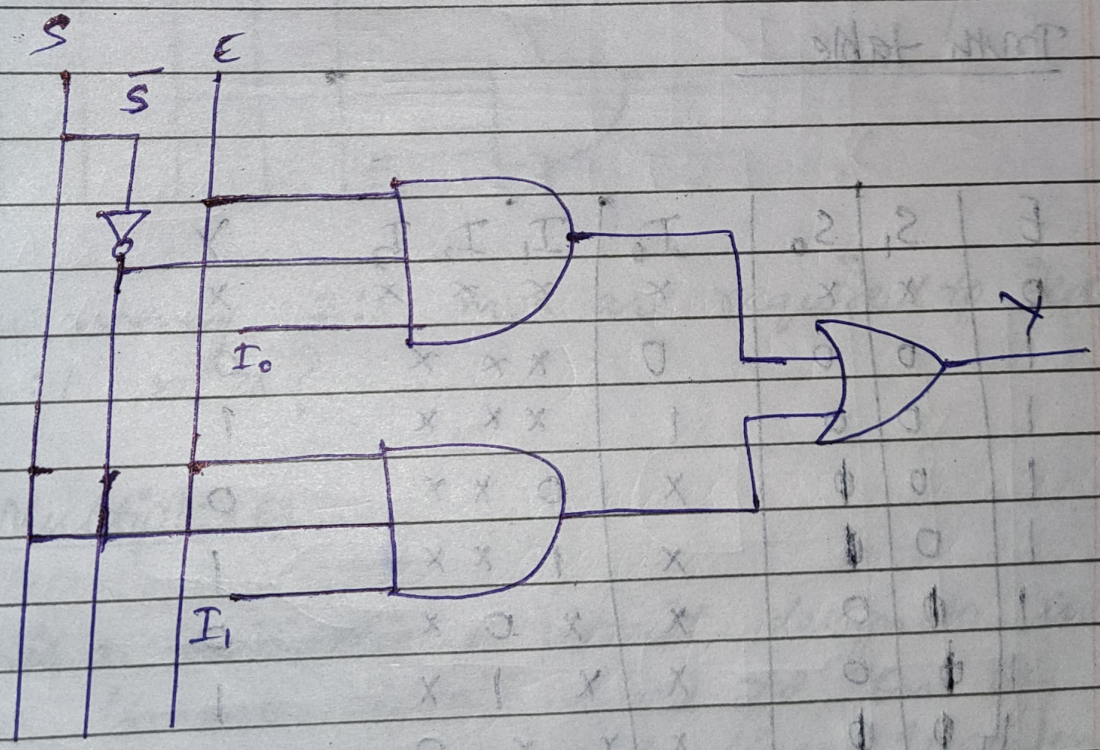
Date

Page

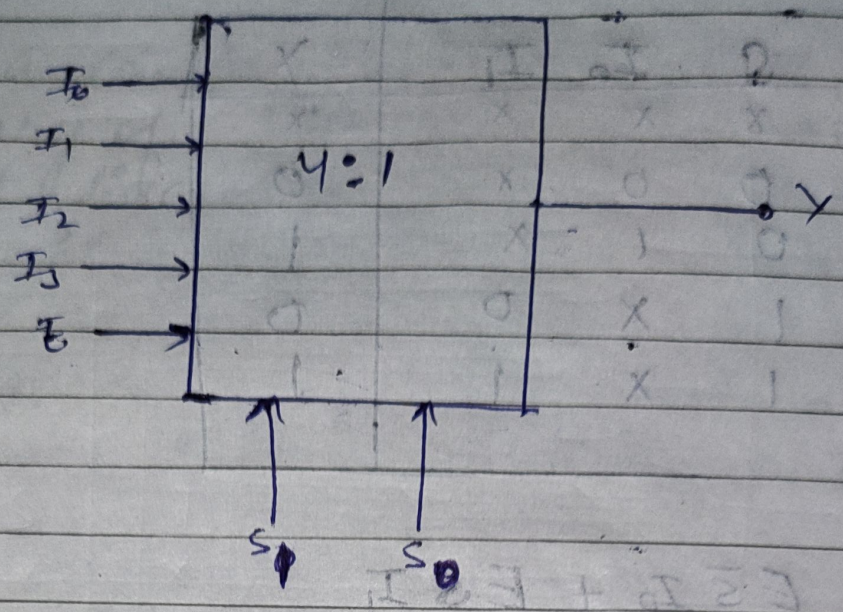
Truth table

E	S	I_0	I_1	Y
0	x	x	x	x
1	0	0	x	0
1	0	1	x	1
1	1	x	0	0
1	1	x	1	1

$$Y = E\bar{S}I_0 + ESI_1$$



~~Truth table~~



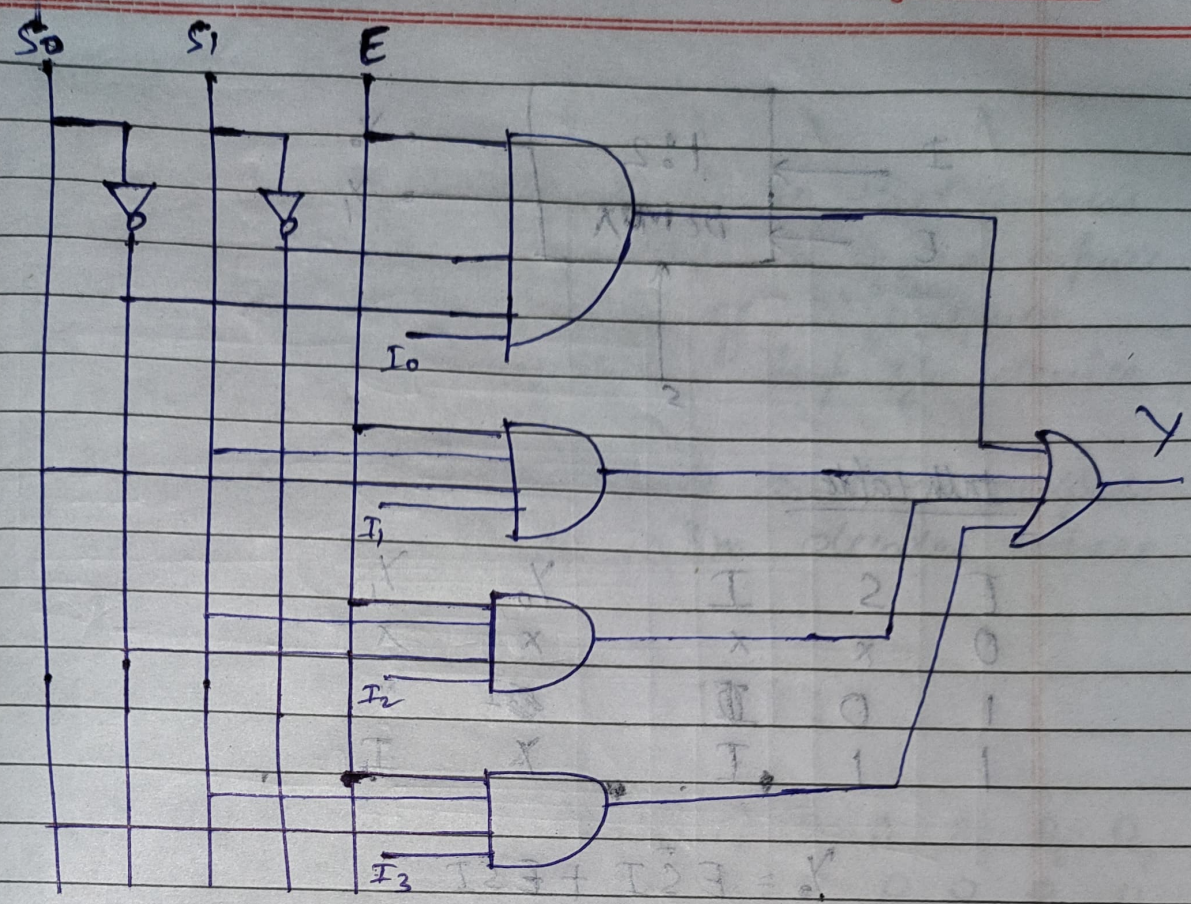
Truth table

E	S_1	S_0	I_0	I_1	I_2	I_3	Y
0	x	x	x	x	x	x	x
1	0	0	0	x	x	x	0
1	0	0	1	x	x	x	1
1	0	ϕ	x	0	x	x	0
1	0	ϕ	x	1	x	x	1
1	ϕ	0	x	x	0	x	0
1	ϕ	0	x	x	1	x	1
1	ϕ	ϕ	x	x	x	0	0
1	ϕ	ϕ	x	x	x	1	1

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

Date

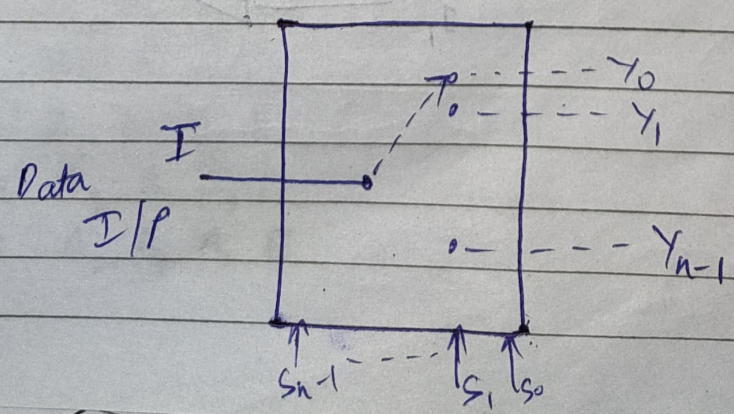
Page

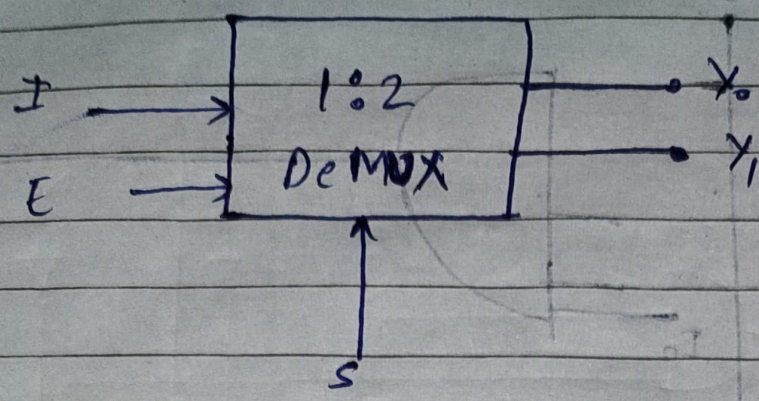


Q How many 2:1 mux are required to design 16:1 mux?

DeMultiplexer

It is a combinational circuit design to switch one common input line to one of several seperate output lines. It performs reverse operation of MUX. It is also called data distributor or 1:many.

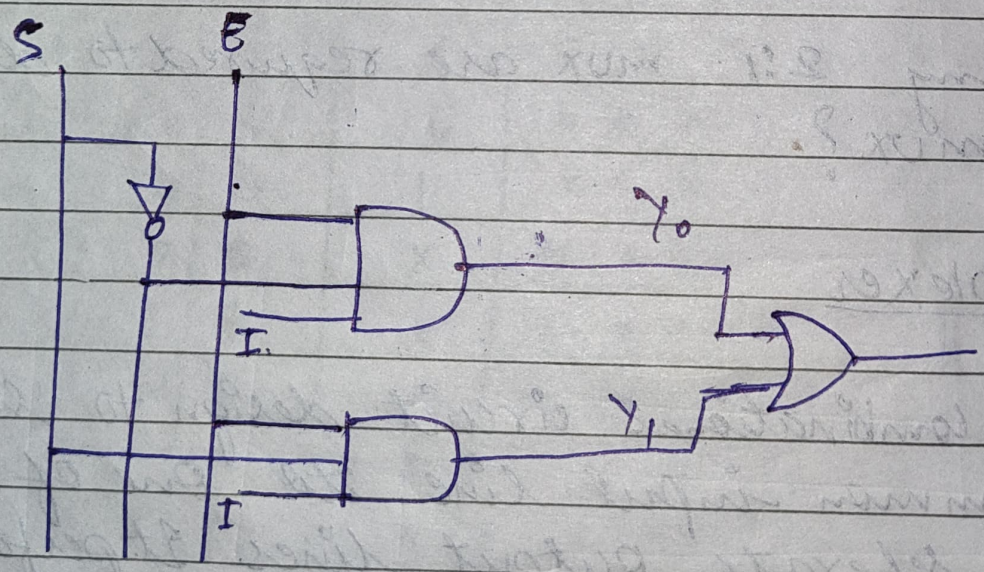


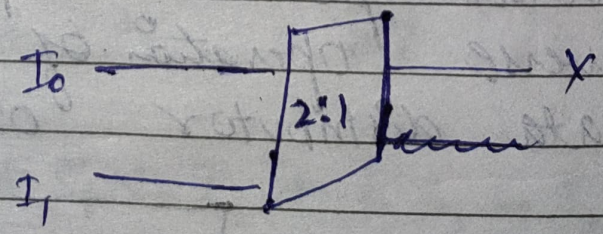


truth table

E	S	I	Y_0	Y_1
0	x	x	x	x
1	0	I	I	x
1	1	I	x	I

$$Y_0 = E\bar{S}I + ESI$$





Central processing unit

1. Introduction
2. General Register organisation
3. Stack organization
4. Instruction format
5. Addressing modes
6. Program control.

Microoperations, Register transfer, Register Transfer language

operations on the content of Registers are termed as Microoperation.
eg - Clear, Shift, Count, Sum, MUL

There are four types of microoperations —
1. data transfer microoperation
2. Arithmetic microoperation
3. Logical microoperation
4. ~~Da~~ Shift microoperation

Data transfer type —
i) Register microoperation
ii) Memory microoperation

Two types of memory transfer microoperation

1. Read: $DR \leftarrow M[AR]$

2. Write: $M[AR] \leftarrow R$

•) Arithmetic Microopc.

eg $R_1 \leftarrow R_1 + R_2$

$R_2 \leftarrow R_1 + \bar{R}_2 + 1$ or $R_2 \leftarrow R_1 - R_2$

•) logical microoper. includes logical operations

eg - $F \leftarrow A \wedge B$ $\wedge$ is for and

$F \leftarrow \overline{A \wedge B}$ nand

$F \leftarrow A \cup B$ OR

$F \leftarrow \overline{A \cup B}$ NOR

$F \leftarrow A \oplus B$ EXOR

$F \leftarrow \overline{A \oplus B}$ EX-NOR.

•) Shift

1. logical shift microoperation
2. circular shift "
3. Arithmetic " "

1. eg $R_1 \leftarrow Sh\,l\,R_1$

$R_1 \leftarrow Sh\,R_1$

2. eg $R_1 \leftarrow Cir\,l\,R_1$

1	0	0	1	0	1	R

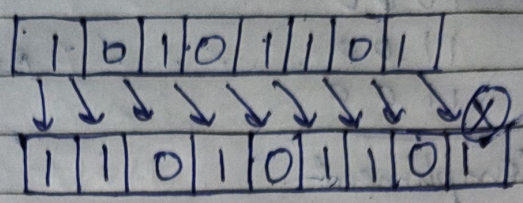
0	0	1	0	1	1

eg $R_1 \leftarrow Cir\,R_1$

Date _____ Page _____

3. Used for sign number

eg - ~~ash r~~ $R_1 \leftarrow$ ashr R_1



Common Bus system for registers using MUX (Bus & memory transfer)

Q A digital computer has common Bus system for 16 registers of 32 bit each how many mux are needed and what is the size of mux. How many selection lines are there.

Ans 32 mux of 16:1 and 4

Date _____ Page _____

Application of logical microoperation

1. Selective Set

$$\begin{array}{r} A = \\ B = \\ \hline \end{array} \begin{array}{c} 1010 \\ 1100 \\ \hline 1100 \end{array}$$

$A \leftarrow A + B$

OR operation

$A \leftarrow A \vee B$

2. Selective Compliment

selective bit compliment, the selective bit has corresponding bit 1 rest all zero and perform EX-OR

$$\begin{array}{r} A = \\ B = \\ \hline \end{array} \begin{array}{c} 1010 \\ 0110 \\ \hline 1100 \end{array}$$

$A \leftarrow A \oplus B$

3. Selective Clear microoperation

to clear the selective bit, put ~~zero~~ one corresponding to the selective bit and rest and perform AND and operation

$$\begin{array}{r} A = \\ B = \\ \hline \end{array} \begin{array}{c} 1010 \\ 0011 \\ \hline 0010 \end{array} \qquad \begin{array}{r} A = \\ B \\ \hline \end{array} \begin{array}{c} 1010 \\ 1100 \\ \hline 0010 \end{array}$$

$A \leftarrow A \wedge B$ ✗

$A \leftarrow A \cap B'$ ✓

4. Mask operation

Corresponding bits are zero. and perform AND

$A \leftarrow A \cap B$

$$\begin{array}{r} A \\ \\ \hline \end{array} \begin{array}{c} 1010 \\ 1100 \\ \hline 1000 \end{array}$$

Date _____ Page _____

5. insert operation

to insert a given bits at a particular bits

A = 01101010
B = 00001111

00001010 $A \leftarrow A \cap B$

10010000

10011010 $A \leftarrow A \cup B$

Q Register A holds the 8 bit binary
A → 11011001
Determine the B operation & logical microopr. to be perform in order to change the value in A to : 01101101, 11111101

~~A 11011001~~
~~B 01~~
~~01101101~~

A 11011001
B 00000000 ~~A∩B~~
A∩B → 00000000
01101101 ← A∪B
01101101

Date _____ Page _____

Logical microoperation types

Boolean Function	microoperation	Name
$F_0 = 0$	$F \leftarrow 0$	Clear
$F_1 = xy$	$F \leftarrow A \cap B$	AND
$F_2 = xy'$	$F \leftarrow A \cap B'$	
$F_3 = x$	$F \leftarrow A$	transfer A
$F_4 = x'y$	$F \leftarrow A' \cap B$	
$F_5 = y$	$F \leftarrow B$	transfer B
$F_6 = x \oplus y$	$F \leftarrow A \oplus B$	Exclusive OR
$F_7 = x + y$	$F \leftarrow A \cup B$	OR
$F_8 = (x+y)'$	$F \leftarrow (A \cup B)'$	NOR
$F_9 = (x \oplus y)'$	$F \leftarrow (A \oplus B)'$	Ex-NOR
$F_{10} = y'$	$F \leftarrow B'$	Complement
$F_{11} = x + y'$	$F \leftarrow A + \bar{B}$	
$F_{12} = x'$	$F \leftarrow \bar{A}$	Complement A
$F_{13} = x' + y$	$F \leftarrow \bar{A} + B$	
$F_{14} = (xy)'$	$F \leftarrow \overline{(A \cdot B)}$	NAND
$F_{15} = 1$	$F \leftarrow$ All 1's	Set to All 1

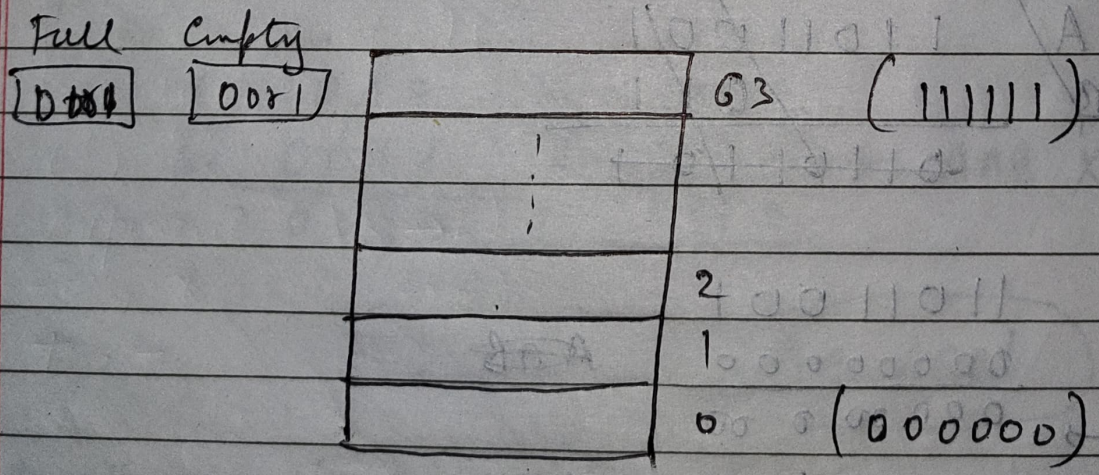


fig- word register stack

Date _____ Page _____

A Stack is defined as a collection of items in a computer architecture system just like a storage device which stores information in such a way that the last item stored. last item stored, first item retrieve (LIFO)

- A stack pointer is needed to point the stack always. A stack pointer holds the address of the top item of stack.
- Only push and pop instructions are required to access the stack.
- Stacks are also used to evaluate the arithmetic expressions.

There are two types of stack.

1. Register stack - build using registers
2. Memory stack - logical memory allocation as stack.

Implementation of Register Stack

push - /* initially sp=0; full = 0; Empty = 1 */

SP ← sp+1
M[SP] ← DR
If (SP=0) then (Full ← 1)
EMPTY ← 0

pop

DR ← M[SP]
SP ← SP - 1
If (SP = 0) then empty ← 1
Full ← 0

A + B

operand operator

A+B : Infix notation
+AB : Prefix notation or polish
AB+ : postfix " or Reverse polish notation

eg - (3×4) + (5*6)

priority
1. ( )
2. exponent (R → L)
3. ÷, * (L → R)
4. +, - (L → R)

Sol^n eg - 34* + 56* RPN Notation
→ (34*56*+)

Date _____ Page _____

Q Find $4*6 + (5*3 \div 3) - 2$ (RPN)

$4*6 + (5*33\div) - 2$	$4*6 + (53*\div 3) - 2$
$4*6 + 5*33\div - 2$	$4*6 + 53*3\div - 2$
$46*5*33\div + - 2$	$46* + 53*3\div - 2$ ✓
$46*5*33\div 2 + -$	$46*53*3\div + - 2$
	$46*53*3\div + 2 -$

~~2. Stack n~~

RPN in stack

3	4	*	5	6	*	+
				6		
	4		5	5	30	
3	3	12	12	12	12	42

$= 42$

Q

$8\times3 + (4+8) + (4*7+2)$

$8*3 + 48+ + 47*2+$

$83* + 48+ + 47*2+$

$83*48++47*2++$

8	3	*	4	8	+	+
				8		
	3		4	4	12	
8	8	24	24	24	24	36

4	7	*	2	+	+
	7		2		
4	4	28	28	30	
36	36	36	36	36	66

$= 66$

Date

Booth's algorithms

Flow ~~of~~ Chart Of Booth's algorithm

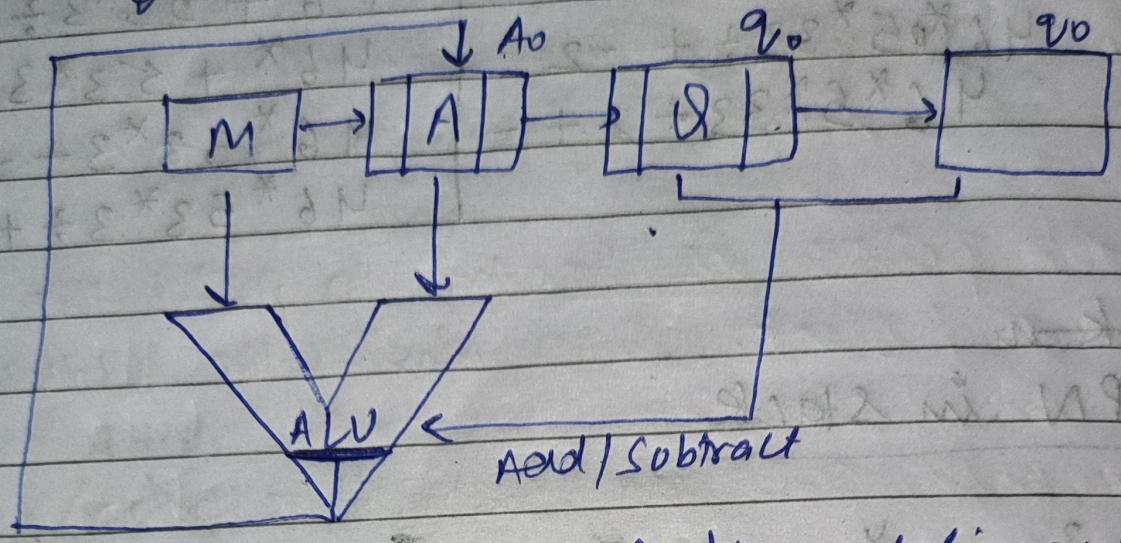
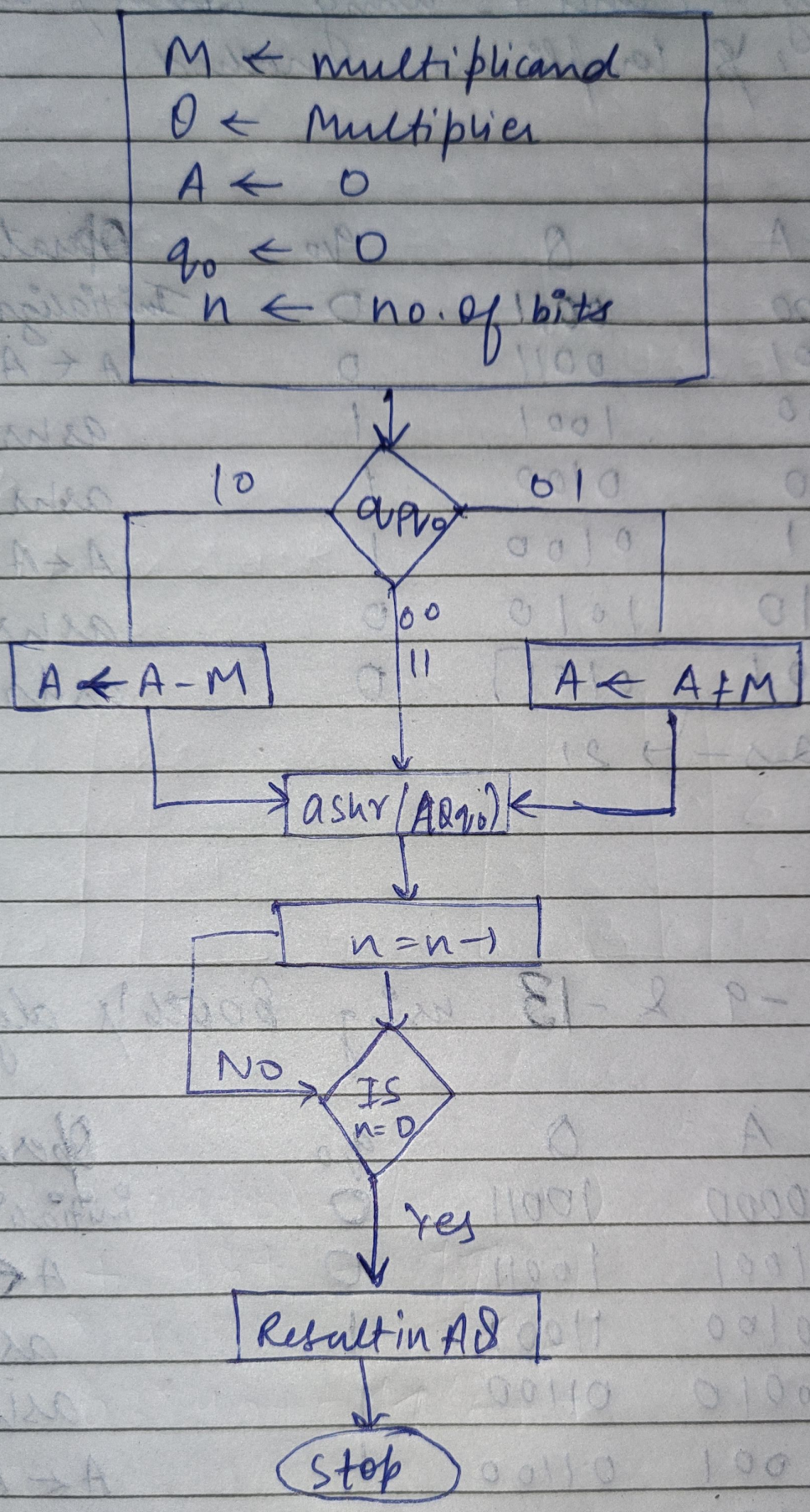


fig- Hardware Implementation of booth algorithm.

M ← multiplicand
Q ← Multiplier
A ← 0
q_0 ← 0
n ← no. of bits

↓


M ← multiplicand
Q ← Multiplier
A ← 0
q_0 ← 0
n ← no. of bits
10
01
q_0q_{-1}
00
11
A ← A − M
A ← A + M
ashr (AQq_{-1})
n = n − 1
NO
IS n = 0
yes
Result in AQ
Stop

Date

multiply +7 and +3 using Booth's algorithm (use 2's compliment method

n	A	Q	q_0	Operation
4	0000	0011	0	Initialization
	1001	0011	0	A ← A-M
	1100	1001	1	ashr
3	1110	0100	1	ashr
2	0101	0100	1	A ← A+M
	0010	1010	0	ashr
1	0001	0101	0	ashr

Ans → 21

Multiply -9 & -13 using Booth's algorithm

n	A	Q	q_0	Operation
5	00000	10011	0	initialization
	01001	10011	0	A ← A - M
	00100	11001	1	ashr
4	00010	01100	1	ashr
3	11001	01100	1	A ← A + M
	11100	10110	0	ashr
2	11110	01011	0	ashr
1	00111	01011	0	A ← A - M
	00011	10101	1	ashr