

10/1/23

Computer Organisation & Architecture

Ch. 1 (i) Number Systems

Common Number Systems \rightarrow Decimal

\rightarrow Binary

\rightarrow Octal (8)

\rightarrow Hexa-Decimal (16)

Binary \rightarrow Octal (8)

\rightarrow Decimal - Con

\rightarrow (Simpler way to convert from binary to decimal)

\rightarrow

$0 = (S)B$ to $(S)D$ (Simpler way to convert from binary to decimal)

eg 1.

A student has no knowledge in computer architecture

but he is good at mathematics and has good memory

so he can convert binary to decimal by hand

Q. How to convert among bases without using calculator?

multiply by 2: if answer is 1, then add 1 to previous result

① Binary \leftrightarrow Decimal

~~divide by 2~~ $B \rightarrow D$ $= \text{sub}(S)D$

$$\text{eg 1. } 101011_2 = 1 \times 2^0 = 1$$

$$1 \times 2^1 = 2$$

$$0 \times 2^2 = 0$$

$$1 \times 2^3 = 8$$

$$0 \times 2^4 = 0$$

$$1 \times 2^5 = 32$$

eg 2.

$$\text{eg 2. } (1101101)_2 \rightarrow (?)_{10} \quad 43_{10}$$

$$\begin{aligned} & 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 0 \times 2^4 + 1 \times 2^5 + 1 \times 2^6 \\ & = 1 + 4 + 8 + 32 + 64 \end{aligned}$$

$$= (109)_{10}$$

eg. 3 - $(100110)_2 = (?)_{10}$

$$0 \times 2^0 + 1 \times 2 + 1 \times 2^2 + 0 \times 2^3 + 0 \times 2^4 + 1 \times 2^5 \\ = 2 + 4 + 32 = 38 \\ = (38)_{10}$$

D \rightarrow B

eg 1 $(125)_{10} \rightarrow (?)_2$

$2 \underline{(125)}$

$2 \underline{(62)} \quad 1$

$2 \underline{(31)} \quad 0$

$2 \underline{(15)} \quad 1$

$2 \underline{(7)} \quad 1$

$2 \underline{(3)} \quad 1$

$2 \underline{(1)} \quad 1$

$1 \quad \leftarrow \text{LSB}$

$\begin{array}{r} \text{MSB} \\ = (111101) \end{array}$

eg. 3 - 269

$2 \underline{(134)} \quad 1$

$2 \underline{(67)} \quad 0$

$2 \underline{(33)} \quad 1$

$2 \underline{(16)} \quad 1$

$2 \underline{(8)} \quad 0$

$2 \underline{(4)} \quad 0$

$2 \underline{(2)} \quad 0$

$\begin{array}{r} \text{EIGHT} \\ = (10001101)_2 \end{array}$

eg 2 $(64)_{10} \rightarrow (?)_2$

$2 \underline{(64)} \quad 0$

$2 \underline{(32)} \quad 0$

$2 \underline{(16)} \quad 0$

$2 \underline{(8)} \quad 0$

$2 \underline{(4)} \quad 0$

$2 \underline{(2)} \quad 0$

$2 \underline{(1 \rightarrow 0)} \quad \rightarrow 1$

$= (1000000)_2$

$\downarrow \text{EIGHT} \rightarrow 1$

Date _____

(2) Octal \leftrightarrow Decimal $\rightarrow (011001)_8 \rightarrow 8.02$

multiply by 8

$$\begin{array}{r} \text{eg. } (1234)_8 \rightarrow \\ \text{divide by 8} \\ 724_8 \xrightarrow{\times 8} 4 \quad 4 \times 8^0 = 4 \\ \downarrow \qquad \qquad \qquad 2 \times 8^1 = 16 \\ 7 \times 8^2 = 448 \quad (8^3) \\ \hline (448)_8 \end{array}$$

$$\text{eq. 1 } (1234)_{10} \rightarrow (?)_8$$

$$8 \overline{(1234)}$$

$$\begin{array}{r} 8 \overline{(154)} \\ 8 \overline{(19)} \\ 8 \overline{(2)} \end{array}$$

$$= (2322)_8$$

$$8 \overline{(2)(3) \quad 1111}$$

$$\begin{array}{r} \times 16 \\ 0 \rightarrow 2 \end{array}$$

(3) Hexadecimal \leftrightarrow Decimal

$$(AB1F)_{16} = (?)_{10}$$

$$0 \quad 15 \times 16^0 = 15$$

$$0 \quad B \times 16^1 = 16$$

$$0 \quad 13 \times 16^2 = 3328$$

$$0 \quad 11 \times 16^3 = 40960$$

$$16 \overline{(1234)}$$

$$16 \overline{(9)} \quad 2 \times 16 = D \quad (4D2)_{16}$$

$$0 \rightarrow 4 \quad (4D2)_{16}$$

$$= (49319)_{10}$$

$$49319$$

(101101001)

10001100

(4)

Octal \leftrightarrow Binary

$$(705)_8 = (?)_2$$

$$\begin{array}{r} 7 \ 0 \ 5 \\ \downarrow \quad \downarrow \quad \downarrow \\ 111 \ 000 \end{array} \Rightarrow (111000101)_2$$

$$\underline{(001101101)}_2 \Rightarrow (155)_8$$

(5)

Octal \leftrightarrow Hexadecimal

$$(1076)_8 = (?)_{16}$$

$$\begin{array}{cccccc} 1 & 0 & 7 & 6 \\ \underbrace{001}_{2} \quad \underbrace{000}_{3} \quad \underbrace{111}_{E} \quad \underbrace{110}_{ } & & & & & \end{array} \text{make sets of 4}$$

$$= (23E)_{16}$$

HW \rightarrow Conversions

9Ra 9Rb bRa aRc

SDS Page No.
Date

COA

MU22

Convert $(25)_{10}$ to Binary $(01+1+3)$

$$2 | 25$$

$$2 | 12 \quad 1$$

$$2 | 6 \quad 0$$

$$2 | 3 \quad 0$$

$$2 | 1 \quad 1$$

↓
↓ 2's comp.

$$1010 \quad 1001$$

$$1001$$

$$1010$$

$$1011$$

↓
↓ 2's comp.

* 1's and 2's complement

↓
↓
Invert all bits

1's comp.

$$101 = 010$$

$$1101$$

$$0111 \rightarrow 1's$$

Invert all bits and add 1

$$010$$

$$+ 1$$

$$011 \rightarrow 2's \text{ comp.}$$

1000 ← 2's comp.

e.g. Subtract 1010 from 1111 using 1's complement theory

$$1111 - 1010 \Rightarrow 111 + (-10)$$

$$1000 +$$

$$15 \rightarrow 1111$$

$$1010 \xrightarrow{1's \text{ comp.}} 0101$$

$$1111$$

$$+ 0101$$

$$(1 \rightarrow 0100)$$

$$(\text{Carry}) \quad 0001$$

$$0101 \rightarrow 5$$

& carry is present
ans → +ve

else → -ve.

2) Subtract 1010 from 1000 using 1's comp.
 $8 + (-10)$

$$1010 \quad \text{1's comp} \quad 0101$$

1000

0101

$\underline{-} 1101$

0 → carry

ans → negative

→ 1's complement form

↓ convert

$$0010 = (2)_{\text{base } 2^4}$$

3)

$$45 - 60, 45 - 116$$

$$45 + (-60)$$

$$111001 + 110100$$

010

1

1

1

1

1

1

1

1

1

1

1

1

1

↓ = 101

1011

1010 → 1110

→ 1's comp → 0001

$$(111001 + 110100) + 0001$$

$$1010 \quad \text{1's comp} \quad 0101$$

111

1010 + 1

0010 ← 1

1000 ← (1100)

1010

carry 0

↓ = 0011

(-3)

ans = 0011

Using 2's complement

$$9 + (-5)$$

↓

→

00000101

↓

→ invert

00001001

+ 11111011

↓
00000100
+ 11111010

disregard (4)
gth bit

11111010

+ 1

11111011 → 2's comp.

$$87 - 9$$

↓

→ 00001001

→ invert

00000101

+ 11110111

↓
11111000

swap bits
↓
invert
00000011

11110110

+ . . . 1

11110111

↓
2's c

+ 1
00000100

neg!

00000100

→ (4)

8bit 25 - 27

00011001

00011011

11100101

→ invert

11111110

11100100

00000001

11100101

→ 2's

00000010

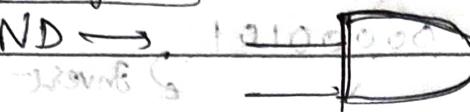
11100101

→ re (2)

Logic gates

Basic gates

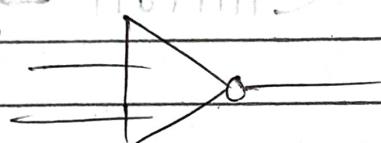
AND \rightarrow



OR \rightarrow



NOT \rightarrow



NAND, NOR, XNOR, NOR

XOR, XNOR, NOR

Exclusive

XOR, XNOR

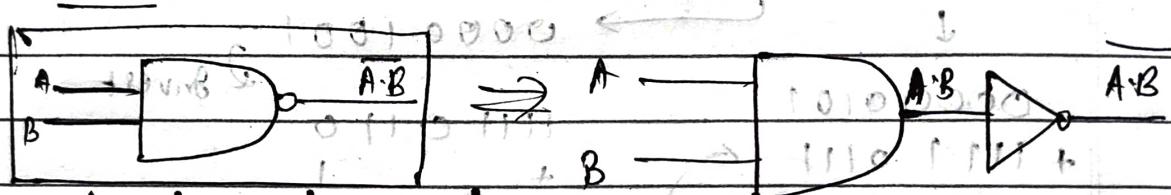
(P) bpropriety

state

19/1/23

NAND

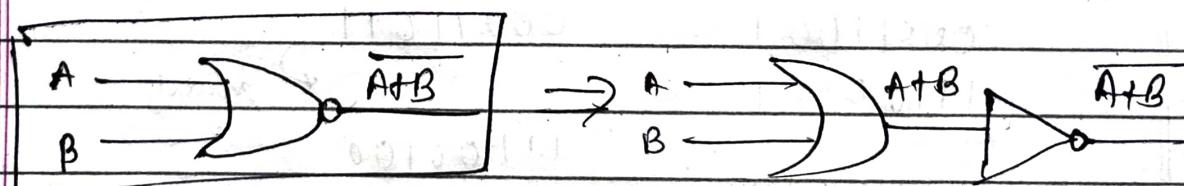
$$\overline{A+B} = \overline{\overline{A} + \overline{B}}$$



A	B	$A \cdot B$	$\overline{A \cdot B}$	$\overline{A + B}$
0	0	0	1	1
0	1	0	1	1
1	0	0	1	1
1	1	1	0	0

NOR

$$\overline{A+B} = \overline{A} \cdot \overline{B}$$



PP

A	B	\overline{A}	\overline{B}	$\overline{A+B}$	$\overline{A \cdot B}$	$\overline{A+B}$
0	0	1	1	1	1	1
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	1	0	0	1	1	1

COA continued →

Exclusive gatesXOR

$$A \oplus B$$

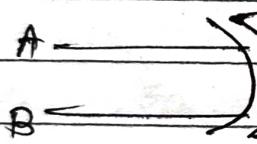
$$\overline{A}B + A\overline{B}$$

Input		O/P
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

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

XNOR

$$A \odot B$$

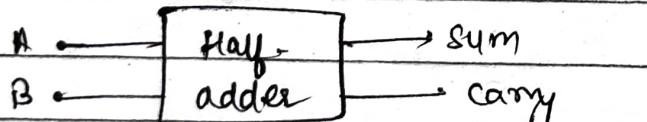


$$\overline{A} \cdot \overline{B} + AB$$

A	B	$A \odot B$
0	0	1
0	1	0
1	0	0
1	1	1

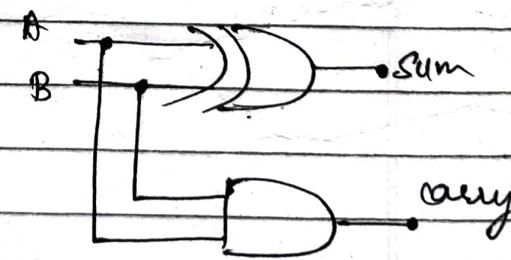
Adderi) Half-adder

K-map concept



* OR AND

A	B	sum	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Q) Full adder



A	B	C _{in}	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

K-map

Sum			
$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}\bar{C}$	0	1	0
\bar{A}^0	0	1	0
A^1	1	0	1

Carry

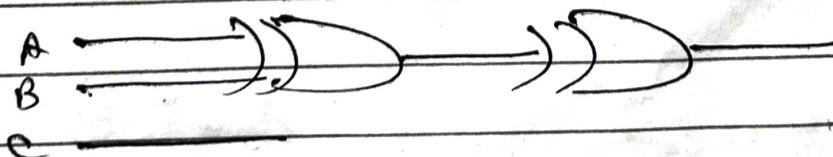
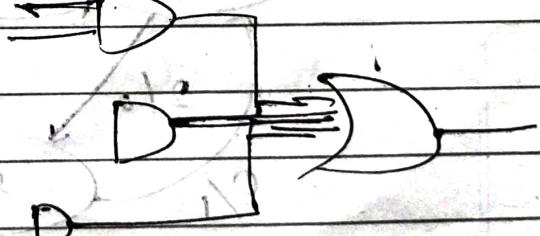
Carry			
$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}\bar{C}$	0	0	1
\bar{A}^0	0	0	1
A^1	1	0	1

$$\bar{B}C + \bar{A}C + A\bar{B}$$

$$\begin{aligned}
 & A\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC + \bar{A}\bar{B} \\
 & C(\bar{A}B\bar{C} + \bar{A}\bar{B}) + \bar{C}(AB + \bar{A}B) \\
 & = C(\bar{A}B + \bar{A}\bar{B}) + \bar{C}(AB + \bar{A}B)
 \end{aligned}$$

↓ ↓

$A \oplus B \oplus C$



Subtractor how

23/1/23

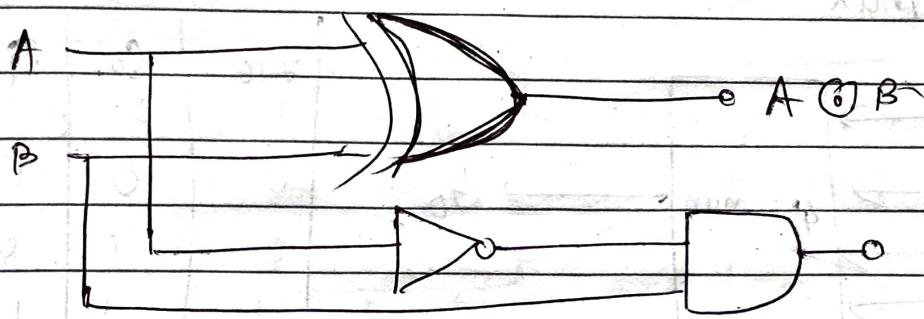
COA

~~TT~~ is not sensible
make it.

Subtractor

	diff	borrow
0 0	0	0
0 1	1	1
1 0	1	0
1 1	0	0

Bx- or



Full-Subtractor

K-map

difference

		$\bar{B}\bar{C}$	$\bar{B}C$	$B\bar{C}$	BC
		00	01	11	10
$\bar{A}0$	00	0	1	0	0
	01	1	0	1	0
$A1$	11	0	1	0	0
	10	0	0	0	1

$$\bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

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

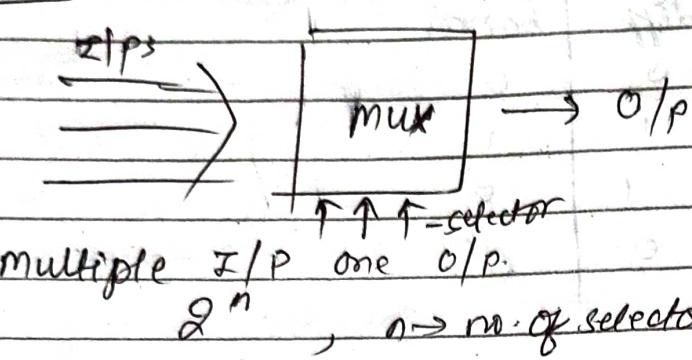
$$C(\bar{A}B + A\bar{B}) + \bar{C}(\bar{A}B + A\bar{B})$$

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

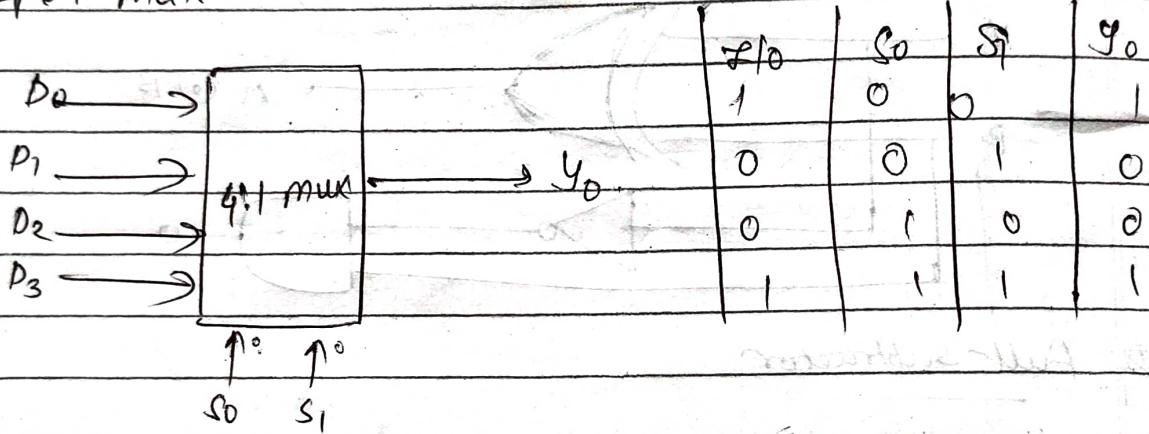
$$= A \oplus B \oplus C$$

Borrow K-map HW

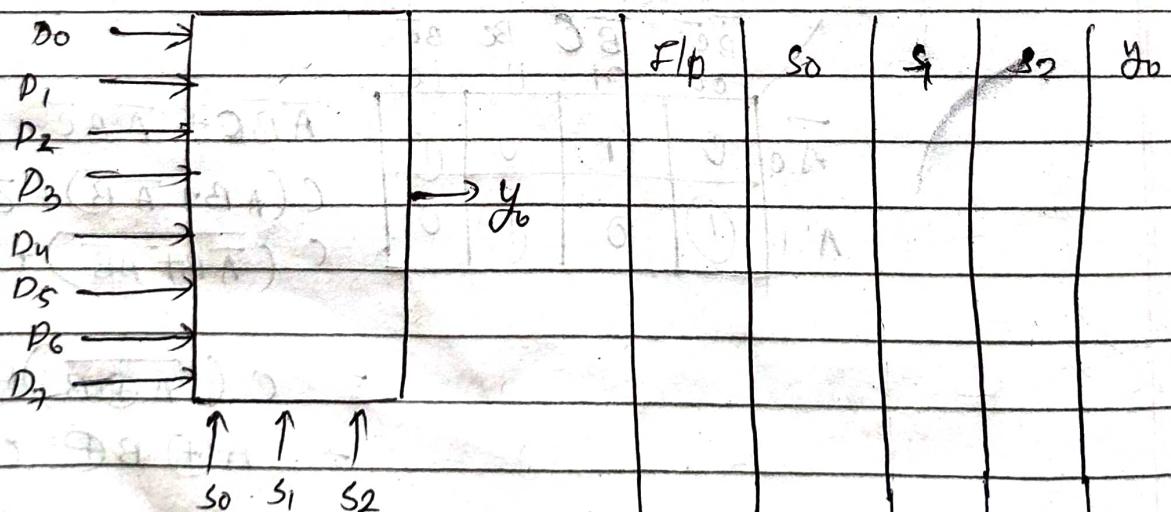
★ mux



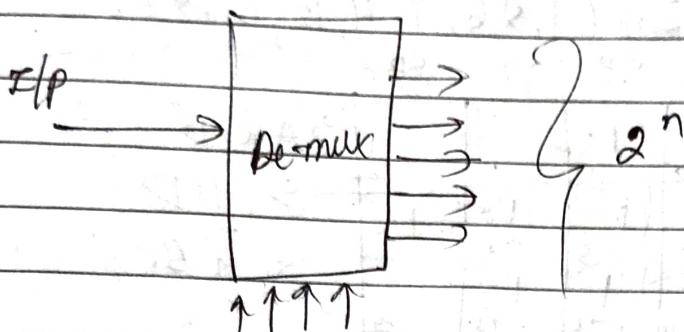
4:1 mux



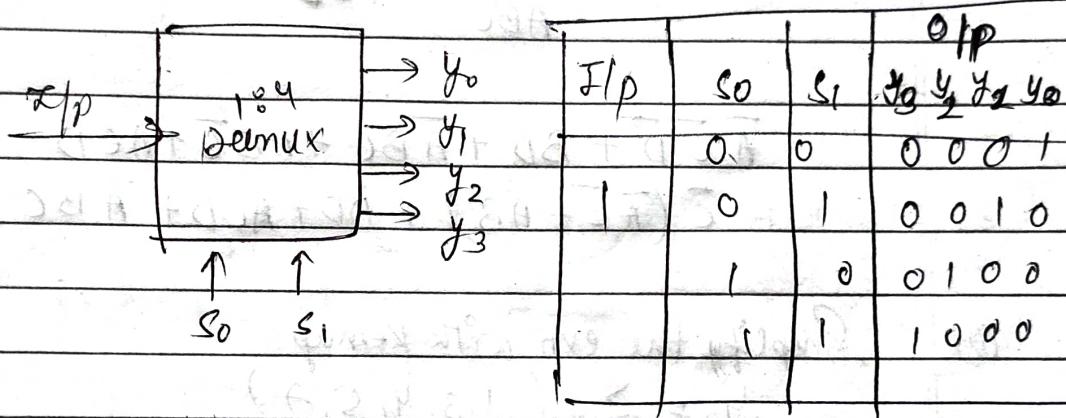
- Draw 8:1 mux & also its truth table.



De-mux

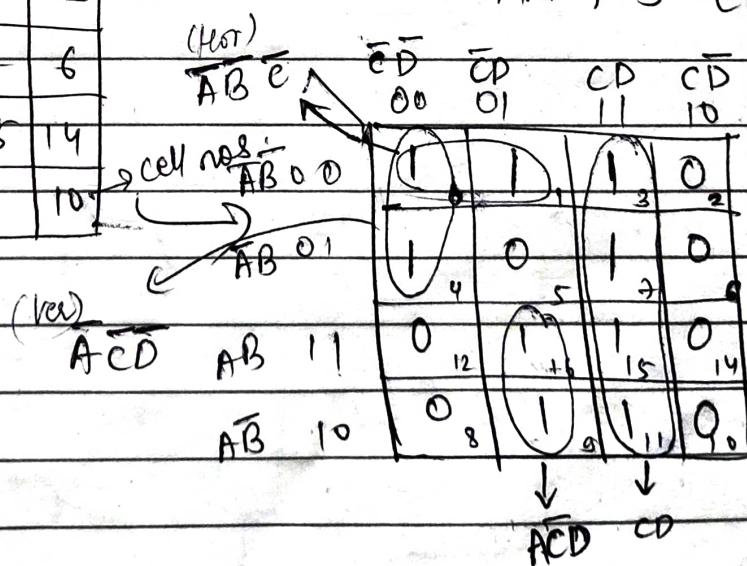


1:4 Demux



(Q) Minimise the four variable logic function using K-map.

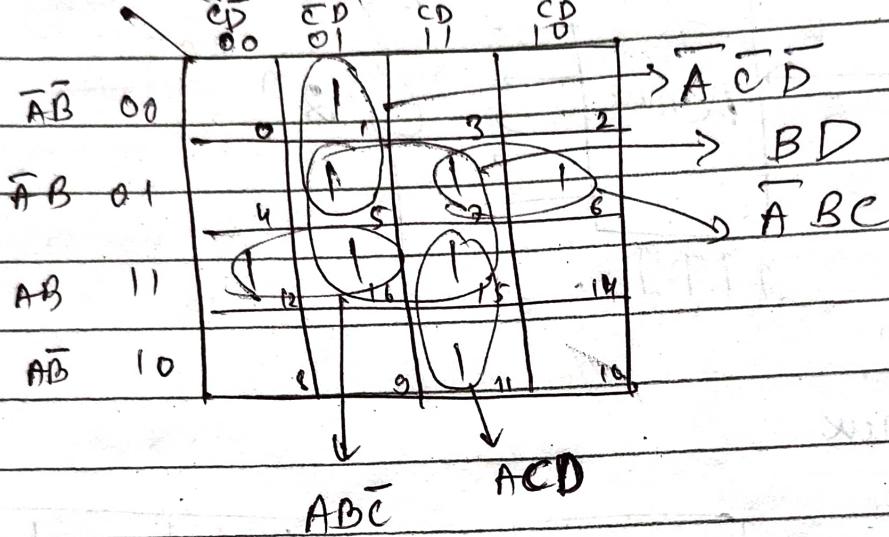
10m	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10



(Q)

Simplify & implement the foll. exp using k-map.

$$Y = \sum m(1, 5, 6, 7, 11, 12, 13, 15)$$



$$\bar{A}\bar{C}\bar{D} + BD + \bar{A}BC + AB\bar{C} + ACD.$$

$$\bar{C}(\bar{A}\bar{C} + AB) + BD + ACD + \bar{A}BC.$$

(Q)

Simplify the exp with k-map

$$Y_1 = \sum m(1, 3, 4, 5, 7)$$

$$Y_2 = \sum m(0, 1, 3, 4, 6)$$

Sum of Product (SOP)

$$(A \cdot BC) + (\overline{A} \cdot \overline{B} \cdot C) + (A \cdot \overline{B} \cdot C)$$

(1)

Grouping of 1's

Product of sums (POS)

e.g -

$$(A + B + \overline{C}) \cdot (\overline{A} + \overline{B} + D) \cdot (\overline{A} + \overline{B} + \overline{C})$$

(0)

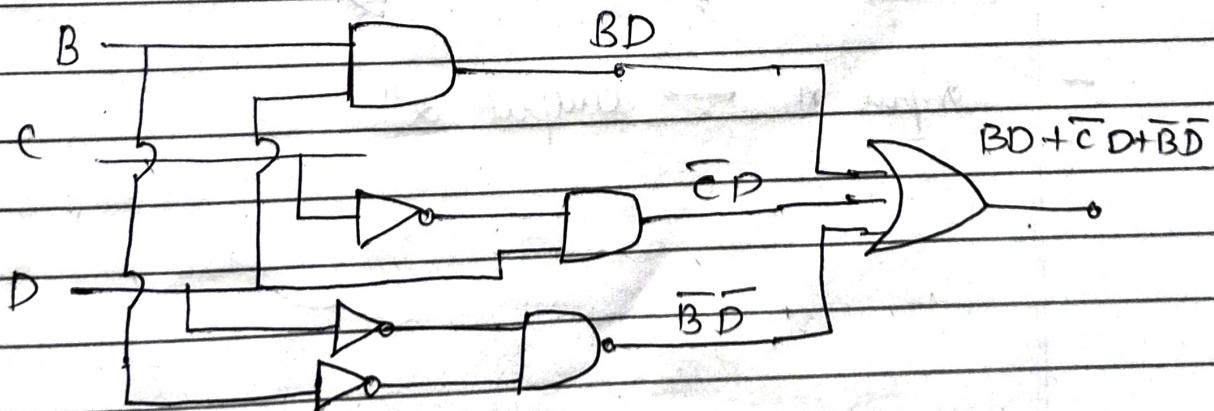
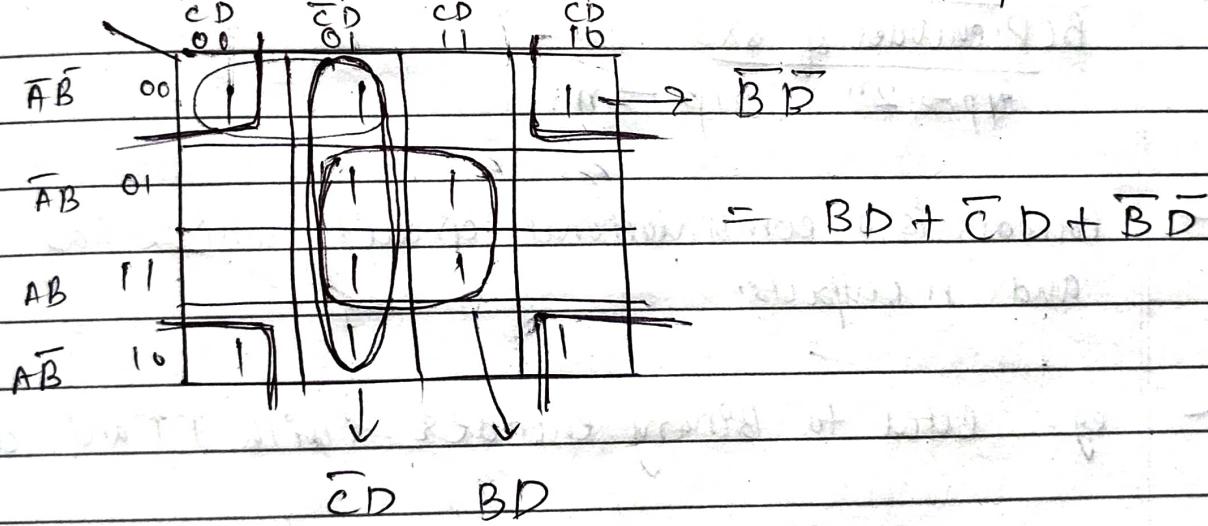
Grouping of 0's

Group the maximum available 1's & 0's

Rule for grouping

(Q) Minimise foll boolean fun^n -

$$F(A, B, C, D) = \sum_m (0, 1, 2, 5, 7, 8, 9, 10, 13, 15)$$



- Don't Care Condition

Name	AND form	OR form
Identity law	$1 \cdot A = A$	$0 + A = A$
Null law	$0 \cdot A = 0$	$A + 1 = 1$
Oidempotent law	$A \cdot A = A$	$A + A = A$
Inverse law	$A \cdot \bar{A} = 0$	$A + \bar{A} = 1$
Commutative	$A \cdot B = B \cdot A$	$A + B = B + A$
Associative	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$	$(A + B) + C = A + (B + C)$
Distributive	$A \cdot (B + C) = A \cdot B + A \cdot C$	$A + (B \cdot C) = (A + B) \cdot (A + C)$
Absorption	$A \cdot (A + B) = A$	$A + (A \cdot B) = A$
De morgan's	$\bar{A} \cdot \bar{B} = \bar{A} + \bar{B}$	$\bar{A} + \bar{B} = \bar{A} \cdot \bar{B}$

* Encoder (Inputs $\xrightarrow{\text{encode}}$ BCD)

BCD encoder of 8×3

$$8/p \rightarrow 2^n \quad 0/p \rightarrow n$$

- Encoder is a combinational circuit which has 2^n inputs and n outputs.

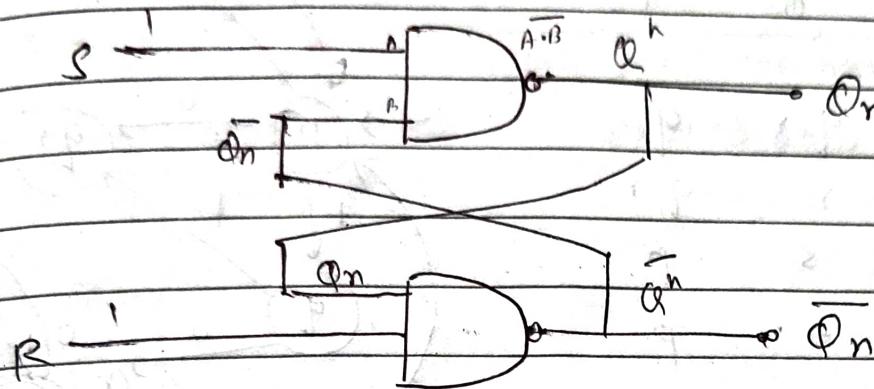
- eg - Octal to Binary encoder (with TT and circuit)

* Decoder

- Input $n \rightarrow$ Output 2^n

Sequential circuits

S-R latch



S	R	Q_n
0	0	Invalid
0	1	1
1	0	0
1	1	Hold

1,1 → me state same reha

$$\begin{aligned} 1 \cdot \bar{Q}_n &= \bar{T} + \bar{Q}_n \\ &= 0 + \bar{Q}_n \\ &= \bar{Q}_n \end{aligned}$$

$$\begin{aligned} 1 \cdot Q_n &= \bar{T} + \bar{Q}_n \\ &= 0 + \bar{Q}_n \\ &= \bar{Q}_n \end{aligned}$$

draw the above digm four times for 4 cases.

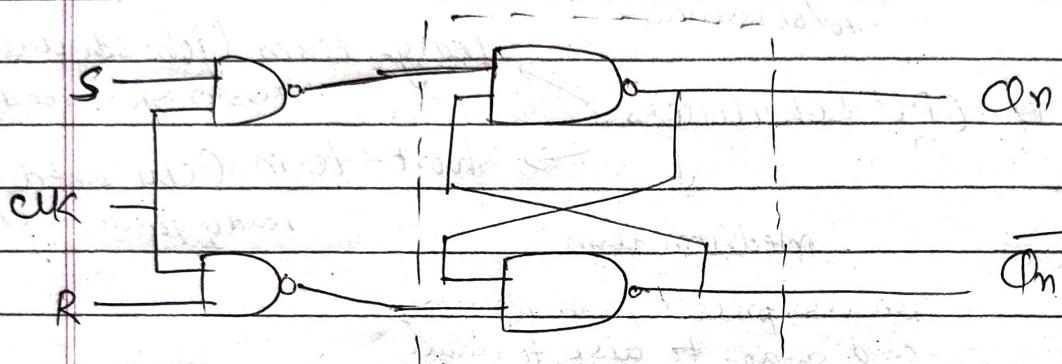
* Diff. b/w Combinational & Sequential circuits.

25/1/23

COA

==

★ C-R Flip Flop

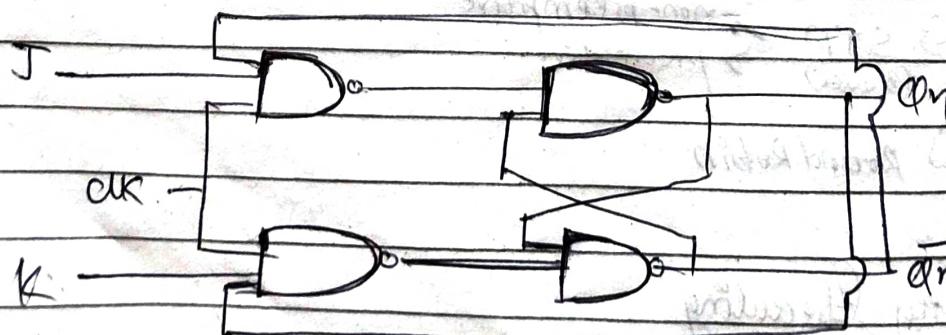


Truth table of SR flip flop.

notclock → avoid

clock	S	R	Q_n
X	0	0	Q_n
1	0	0	Hold
1	0	1	0
1	1	0	1
1	1	1	Invalid

★ J-K Flip-Plop



$$0 \rightarrow 3.5V \quad 1 \rightarrow 5V$$

$$V < 3.5 = 0 \quad V > 3.5 = 1$$

<u>clk</u>	<u>J</u>	<u>K</u>	<u>Q_n</u>
X	0	0	Q _n
1	0	0	hold
1	0	1	0
1	1	0	1
1	1	1	toggle/ Q _n / race condn

Case 1

$$\bar{Q}_n = 1, \bar{Q}_{n+1} = 0$$

$$J = 1, K = 1$$

$$clk = 1$$

$$Q_n = 0, \bar{Q}_{n+1} = 1$$

Case 2

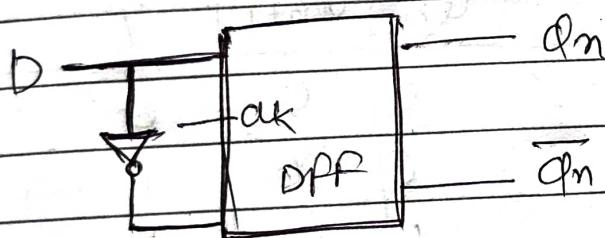
$$Q_n = 0, \bar{Q}_{n+1} = 1$$

$$J = 1, K = 1, clk = 1$$

$$Q_n = 1, \bar{Q}_{n+1} = 0$$

★ D-PiP Plop (Data/1delay) (on SR PP)

standard truth table.



<u>clk</u>	<u>D</u>	<u>Q_n</u>
X	X	Q _n
1	0	0
1	1	1

characteristic truth table

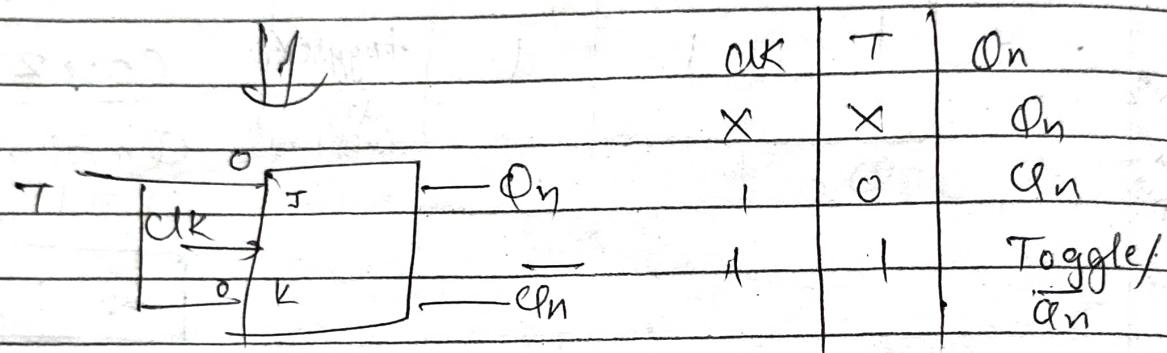
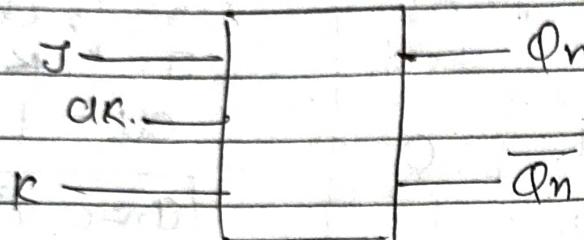
SR-PiP Plop

<u>Q_n</u>	<u>D</u>	<u>Q_{n+1}</u>
0	0	0
0	1	1
1	0	0
1	1	1

excitation table

<u>Q_n</u>	<u>Q_{n+1}</u>	<u>D</u>
0	0	0
0	1	1
1	0	0
1	1	1

* T flip flop (Based on JK FF) (Toggle)



Characteristic TT Excitation table

C _n	T	Q _{n+1}	C _n	Q _{n+1}	T
0	0	0	0	0	0
0	1	1	0	1	1
1	0	1	1	0	1
1	1	0	1	1	0

↓

$Q_n \oplus T$

21/1/23

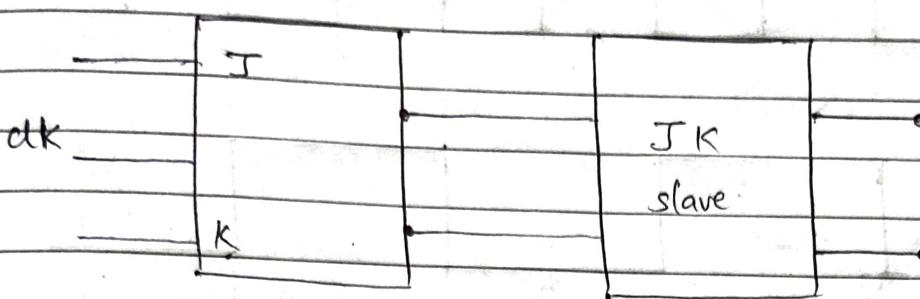
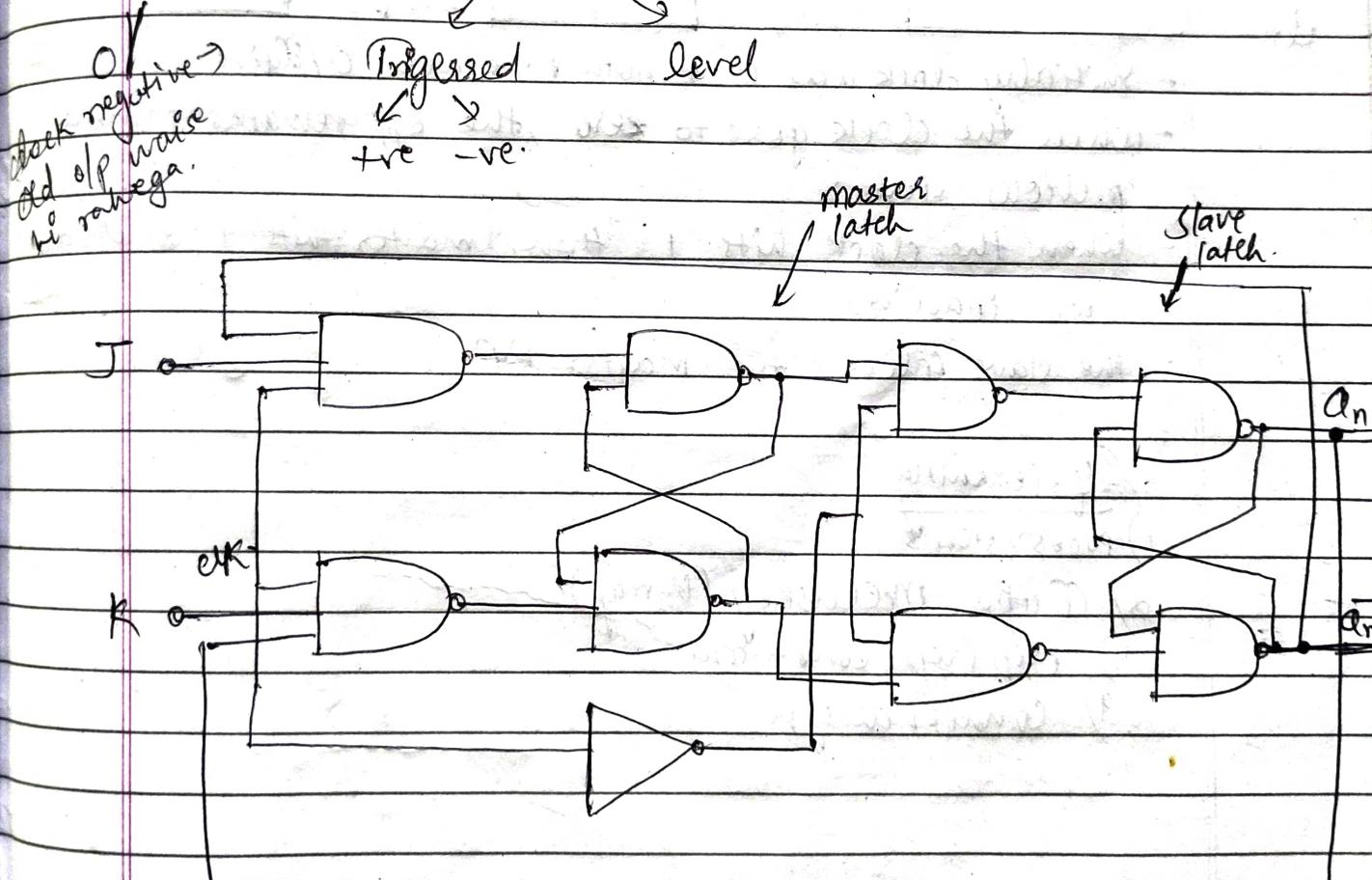
COA

SDS

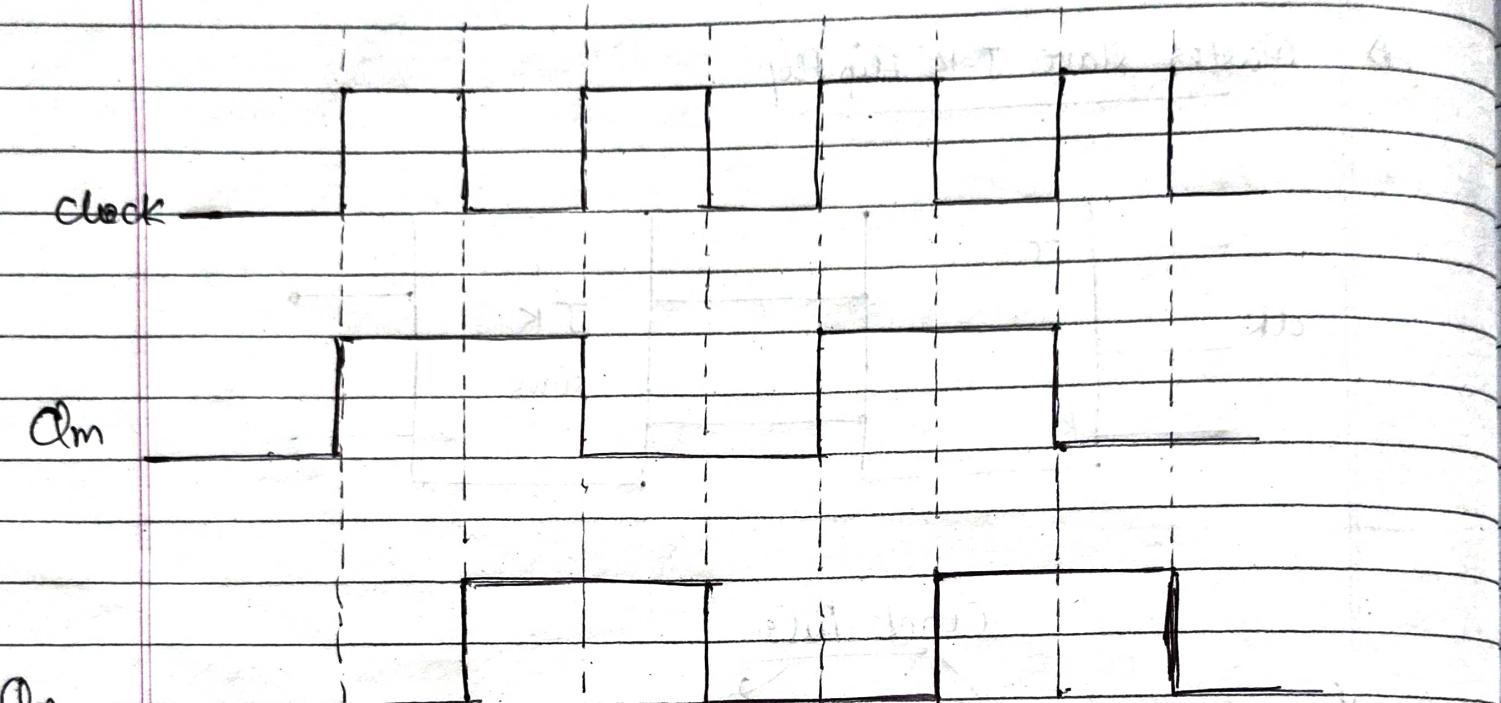
Page No.

Date

Master Slave J-K Flip Flop

Clock Pulse

Timing diagrams $f_1 = 1, f_2 = 1 \text{ case}, //$



- Initially clock was 0 & both P.Rs were 0/active
- when the clock goes to 1, the Q/P remains same as previous state.
- when the clock hits 1 then master hits 1 and slave was inactive.
- The slave follows the master P.R

Self-learning

- 1) No. systems
- 2) Quine - McCluskey (K-map)
- 3) Flip Flop conversions
- 4) Counter design

7/2/23

COA

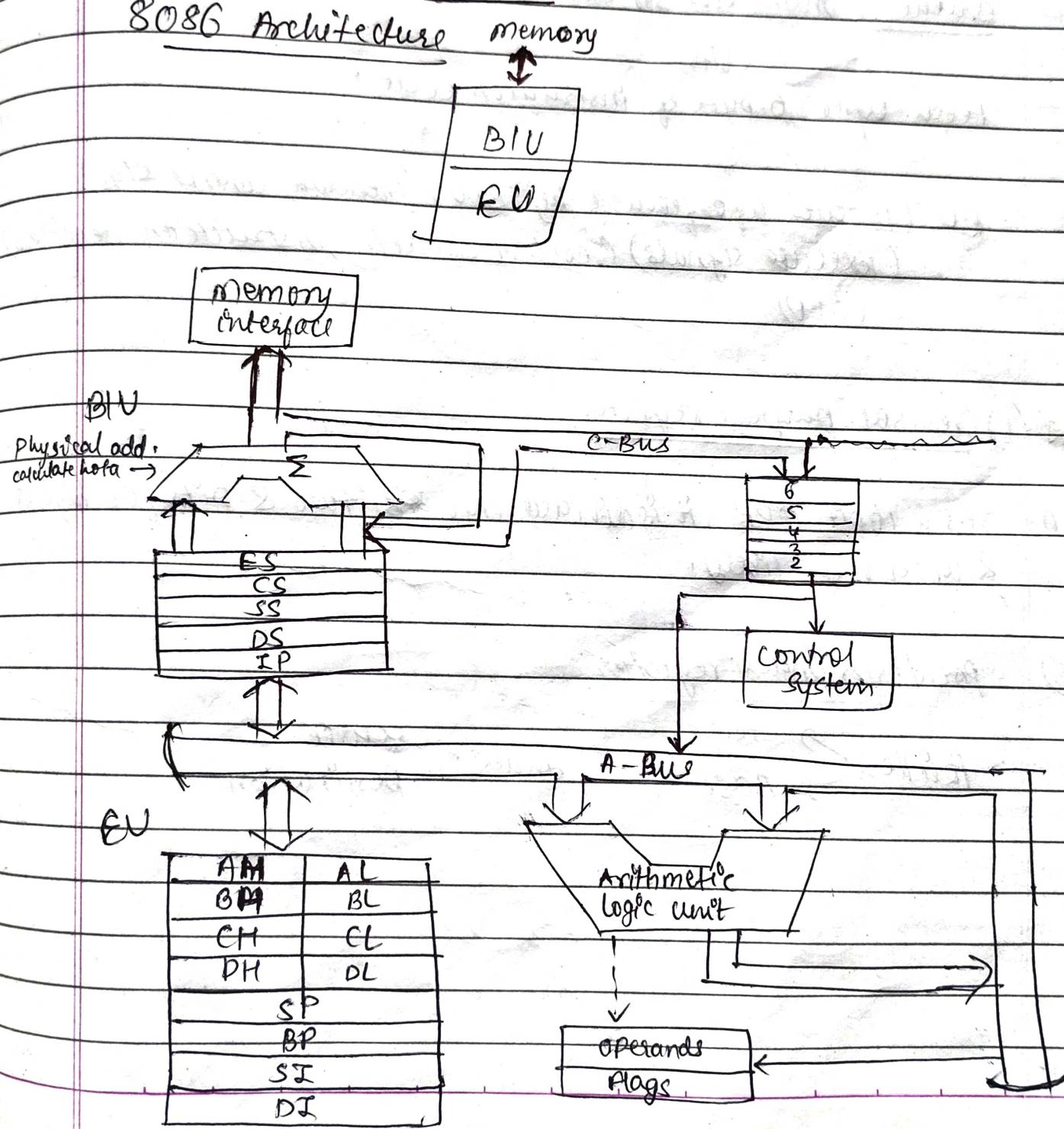
SDS	Page No.
Date	/ /

Chp 2:

- QB 1) Draw 8086 architecture and explain.
- 2) Diff. b/w computer architecture & organisation.
- x 3) Von Neumann model

COA

8086 Architecture



★ Instructional pointer (IP)

- ★ It's a 16-bit register it's used to store the address of next instruction which is to be executed.

★ Segment Registers (Total 1Mb memory) (Address of content)

- ★ It is divided into 4 segments code segment, Data segment, Extra segment, Stack segment 64kb each.

★ Queue (Instruction stream queue)

(48 bits)

- Fetch upto 6 byte of instruction code.

★ EU (Decodes info fetched by BIU) (Generate control signals (execute signals) (control circuitry, instruction decoder) (ALU))

★ (1) General purpose register

- It's a 16bit GPR, It is divided into two 8 bits for higher & lower respectively.

(2) Pointers & Index register

Pointers → stack
Pointers → base

Index → source
Index → destination

Q) (B) ALU - 16 bit
Arithmetic calcⁿ.
eg ADD AX, BX
MOV AX, a.

4) Flag register - 16 bit.
- 9 sections imp has.

5) Control system: ~~Control~~, Control the flow of instructions.

10/2/23

Direct addressing with bracket []
↓
address

With good brackets	Register
Date	/ /

Q) Instruction set of 8086.

- * Modes of 8086
 - minimum mode (single) (operated when MN/MX pin to logic 1 proc.)
 - maximum mode (multiple) (global/local resources)

Adv. & Appl.

- * (Q) Distinguish b/w Min and Max mode.

HW
= Why is
Parting chip? (2pin vs 3pin)

Diff. Address bus & Data bus.

- Q) Addressing modes of 8086
 - Types (explain all groups)
 - Register
 - Immediate
 - memory