DLD - complete - full notes

Digital Electronics (SVKM's NMIMS)

→

→ Decimal

Base - 10

Digits (10) - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

→ Binary

Base - 2

Digits (2) - 0, 1

→ Octal

Base - 8

Digits (8) - 0, 1, 2, 3, 4, 5, 6, 7

→ Hexa decimal

Base - 16

Digits (16) - 0, 1, 2, 3, 4 ...... 10, 11, 12, 13, 14, 15

0, 1, 2 ..... 8, 9 → decimal

$(10)_{10} = (A)_{16}$ $(13)_{10} = (D)_{16}$

$(11)_{10} = (B)_{16}$ $(14)_{10} = (E)_{16}$

$(12)_{10} = (C)_{16}$ $(15)_{10} = (F)_{16}$

Conversions:

1) Decimal ↔ Binary (D → B)

$(55)_{10}$

$(55)_{10} = (110111)_2$

2	55		Least significant bit (LSB)
2	27	1	
2	13	1	} divide by the base of conversion you are doing (eg here binary)
2	6	1	
2	3	0	
2	1	1	
			Most significant bit (MSB)

MSD

TH H T U → LSD

Eg. 2 1 3

Digit in decimal

Bit in binary

PAGE NO.
DATE / /

2) Binary to Decimal ($B \leftrightarrow D$)

eg. $(11001101)_2$ {* Multiply each bit by power of 2}

$(1\times2^7)+(1\times2^6)+(0\times2^5)+(0\times2^4)+(1\times2^3)+(1\times2^2)+(0\times2^1)+(1\times2^0)$

$= 128 + 64 + 0 + 0 + 8 + 4 + 0 + 1$

$= 205$

Ans $(11001101)_2 = (205)_{10}$

eg $(1101.101)_2$

$(1\times2^3)+(1\times2^2)+(0\times2^1)+(1\times2^0)+(1\times2^{-1})+(0\times2^{-2})+(1\times2^{-3})$

$= 8 + 4 + 0 + 1 + 0.5 + 0 + 0.125$

Ans $= (13.625)_{10}$

3) Decimal to Octal ($D \rightarrow O$)

eg $(70)_{10}$

8	70	
8	8	6
8	1	0
	0	1

$(70)_{10} = (106)_8$

4) Octal to Binary

eg $(968)_8 \rightarrow$ invalid

eg ~~$(436)_8$~~

Imp. $(0)_{10} \rightarrow (00)_2$

$(1)_{10} \rightarrow (01)_2$

$(2)_{10} \rightarrow (10)_2$

$(3)_{10} \rightarrow (11)_2$

$(4)_{10} \rightarrow (100)_2$

If no. of bits $\rightarrow k$
Then no. of combinations is 2^k : $k \rightarrow 2^k$

$1 \rightarrow 2$

$2 \rightarrow 4$

$(000)_2 \leftrightarrow (0)_{10}$

$(001)_2 \leftrightarrow (1)_{10}$

$(010)_2 \leftrightarrow (2)_{10}$

$(011)_2 \leftrightarrow (3)_{10}$

$(100)_2 \leftrightarrow (4)_{10}$

$(101)_2 \leftrightarrow (5)_{10}$

$(110)_2 \leftrightarrow (6)_{10}$

$(111)_2 \leftrightarrow (7)_{10}$

Every octal no. is represented using 3 bit binary

eg $(436)_8 = (100\ 011\ 110)_2$

eg $(756)_8 = (111\ 101\ 110)_2$

5. Binary to Octal

* Make group of 3 from LSB. Then add 0 to complete the set of 3 if any grp is incomplete.

eg. $(1100\ 001\ 111\ 010)_2$

$\underbrace{001}_{1}\ \underbrace{100}_{4}\ \underbrace{001}_{1}\ \underbrace{111}_{7}\ \underbrace{010}_{2}$

Ans $(14172)_8$

eg $(7AF3)_{16}$

7 – 111

A = 1010

F – 1111

3 – 0011

~~Ans~~ Ans: 111101011110011

6) Decimal to Binary

eg $(121.35)_{10}$

$(121)_{10} \longrightarrow 1111001$

only consider digit after decimal

2	121	
2	60	1
2	30	0
2	15	0
2	7	1
2	3	1
2	1	1
	0	1

$0.35 \times 2 = 0.7$ 0

$0.7 \times 2 = 1.4$ 1

$0.4 \times 2 = 0.8$ 0

$0.8 \times 2 = 1.6$ 1

$0.6 \times 2 = 1.2$ 1

$(0.35)_{10} = 01011$

$\therefore (121.35)_{10} \longrightarrow (1111001.01011)_2$

Binary Addition

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

Q

MSB
1 0 1 1 0 1 → LSB
+ 1 1 0 1 1 1
1 1 0 0 1 0 0

Q $(43)_{10} + (54)_{10}$

$(43)_{10} \rightarrow 101011$

$(54)_{10} \rightarrow 110110$

2	43	
2	21	1
2	10	1
2	5	0
2	2	1
2	1	0
	0	1

2	54	
2	27	0
2	13	1
2	6	1
2	3	0
2	1	1
	0	1

  1 0 1 0 1 1
+ 1 1 0 1 1 0
  1 1 1 → carry
1 1 0 0 0 0 1

$= (1100001)_2$

Q $(23.6)_{10} + (33.7)_{10}$

$(23.6)_{10} \rightarrow (10111.1001)_2$

$(33.7)_{10} \rightarrow (100001.1011)_2$

2	23	
2	11	1
2	5	1
2	2	1
2	1	0
	0	1

$0.6 \times 2 = 1.2$ 1
$0.2 \times 2 = 0.4$ 0
$0.4 \times 2 = 0.8$ 0
$0.8 \times 2 = 1.6$ 1

```
  1 0 1 1 1 . 1 0 0 1
1 0 0 0 0 1 . 1 0 1 1
    1 1 1 1     1 1     → carry
-----------------------
1 1 1 0 0 1 . 0 1 0 0
```

Ans 111001.0100

2	33	
2	16	1
2	8	0
2	4	0
2	2	0
2	1	0
	0	1

$0.7 \times 2 = 1.4$ 1
$0.4 \times 2 = 0.8$ 0
$0.8 \times 2 = 1.6$ 1
$0.6 \times 2 = 1.2$ 1

Binary Substraction

A	B	Diff	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0
1 - 1 - 1		1	1

9 - 6 :

```
  1 0 0 1
  0 1 1 0
      1 1  → Borrow
```

6 - 9 :

```
  1 0 0 1
  0 1 1 0
  1 1
---------
  0 0 1 1
```

0 - 1 - 1

Borrow : 1 + 0 = 1

Diff 1 - 1 = 0

PAGE No.
DATE / /

Q $(6)_{10} - (9)_{10}$

i) Convert the 2^{nd} no. into 1's complement
ii) Add 1^{st} no. and 1's complement of 2^{nd} no.
iii) If after addition, carry is not generated, the ans is -ve then take 1's complement as final ans.

* [1's complement
→ write the opp.
Eg. 10101 = 01010

iv) If carry is generated, then add carry to the final answer

6 = 0110
9 = 1001 → 0110

  0110
  0110
carry ← 11
  1100

∴ $(0011)_2 = -3$

Q 8 − 3
8 = 1000
3 = 0011 → 1100

  1000
  1100
indicates (+ve) sign of ans.
carry ← 1 0100

  0100
+   1 ← added carry
  0101
= +5

DATE: 20/07/2022

Laws Of Boolean Algebra

→ Commutative Law:

$A + B = B + A$ { $+ \rightarrow OR$

$A \cdot B = B \cdot A$ $\quad \cdot \longrightarrow AND$

→ Associative Law:

$A + (B + C) = (A+B) + C$

$A \cdot (B \cdot C) = (A \cdot B) \cdot C$

→ Distributive Law:

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

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

H.W Q Show that $AB + \bar{A}C + BC = AB + \bar{A}C$

Rules of AND

$A \cdot 1 = A$

$A \cdot 0 = 0$

$A \cdot A = A$

$A \cdot \bar{A} = 0$

Rules of OR

$A + 1 = 1$

$A + 0 = A$

$A + A = A$

$A + \bar{A} = 1$

Rules of NOT

$\bar{\bar{A}} = A$

Rules of EX-OR

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

$\overline{A \oplus B} = \text{Ex-NOR} = A \odot B$

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

L.H.S: $AB + \bar{A}C + BC$

PAGE No.
DATE / /

Q. Show that $A + AB = A$

Ans. $A + A \cdot B = A$

L.H.S $A[1+B] = E$

$= A \cdot 1 = A = R.H.S$

Q. Show that $(A+B)(A+\bar{B}) = A$

L.H.S: $(A+B)(A+\bar{B})$

$A + (B \cdot \bar{B}) = A + 0$

$= A = R.H.S$

De Morgan's Theorem

1. $\overline{A \cdot B \cdot C \ldots} = \bar{A} + \bar{B} + \bar{C} + \ldots$ (NAND → OR)

2. $\overline{A + B + C + \ldots} = \bar{A} \cdot \bar{B} \cdot \bar{C} \ldots$ (NOR → AND)

Q. Show that $\overline{\overline{A\bar{B} + ABC} + A(B + \bar{A}B)} = \cancel{}A\bar{B}$

L.H.S: $\overline{\overline{A\bar{B} + ABC} + A(B + \bar{A}B)}$

$= \overline{\overline{A\bar{B} + ABC}} \cdot \overline{A(B + \bar{A}B)}$

$= (A\bar{B} + ABC) \cdot [\bar{A} + (\overline{B + \bar{A}B})]$

$= (A\bar{B} + ABC) \cdot [\bar{A} + (\bar{B} \cdot \overline{\bar{A}B})]$

$= (A\bar{B} + ABC) \cdot [\bar{A} + (\bar{B} \cdot \cancel{\overline{\bar{A}B}} (\bar{\bar{A}} + \bar{B}))]$

$= (A\bar{B} + ABC) \cdot [\bar{A} + (\bar{B} \cdot (A + \bar{B}))]$

$= (A\bar{B} + ABC) \cdot [\bar{A} + (\bar{B} \cdot A + \bar{B} \cdot \bar{B})]$

$= (A\bar{B} + ABC) \cdot [\bar{A} + (\bar{B} \cdot A + \bar{B})]$

$= (A\bar{B} + ABC)\cdot[\bar{A} + \bar{B}(1+A)]$

$= (A\bar{B} + ABC)\cdot(\bar{A} + \bar{B})$

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

$= 0 + A\bar{B} + 0 + 0$

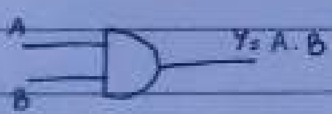
$= A\bar{B}$

H.W. 1) $\overline{\overline{\bar{A}B\bar{C}} + ACD + \overline{ACD} + \bar{B}\bar{C}\bar{D}} + BC\bar{D} = ACD$

2) $[(A+B)(A+\bar{B})] + [\overline{C \oplus BD} + C \oplus D] = A$

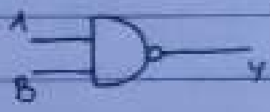
Basic Logic Gates

AND



NAND

$Y = \overline{A.B}$



OR

$Y = A + B$

NOR

$Y = \overline{A+B}$

NOT

$Y = \bar{A}$

EXCLUSIVE - OR (EX-OR)

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

$Y = A \oplus B$

Q Simplify the given boolean exp. and implement using two input logic gates.

$$Y = \overline{AB}\,(B+C) + AB\,(\overline{B+C})$$

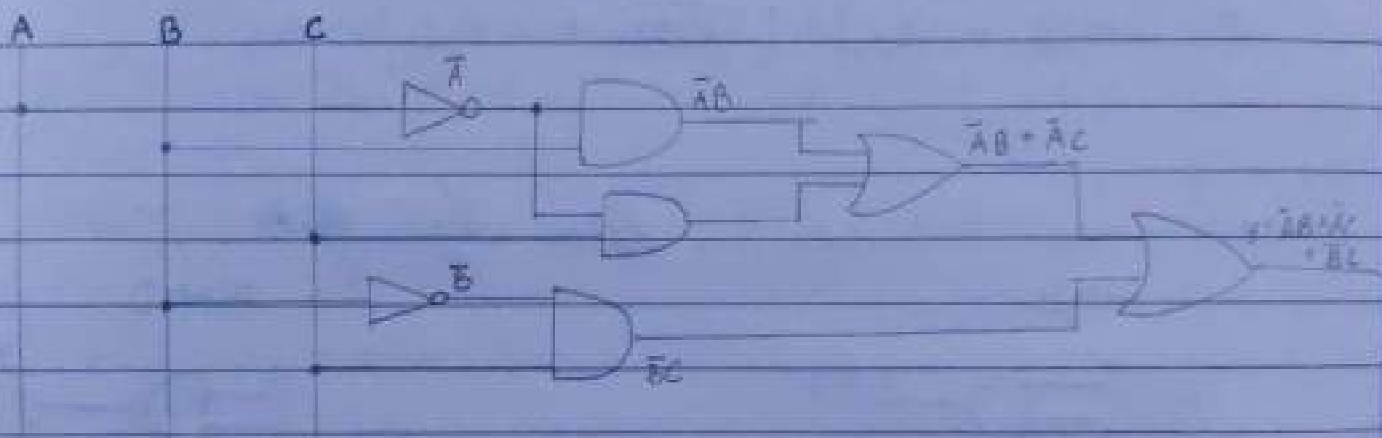
$$Y = (\bar{A} + \bar{B})(B+C) + AB\,(\bar{B}\cdot\bar{C})$$

$$Y = \bar{A}B + \bar{A}C + B\bar{B} + C\bar{B} + AB\bar{B}\bar{C}$$

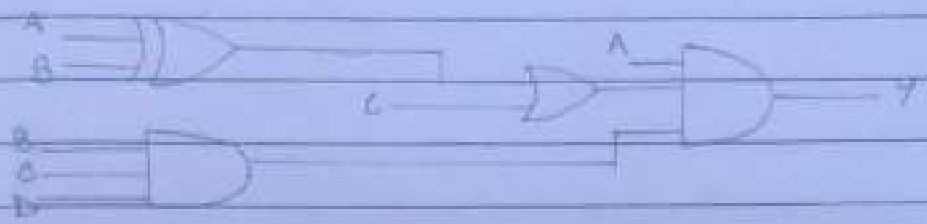
$$Y = \bar{A}B + \bar{A}C + 0 + C\bar{B} + 0$$

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

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



Q Write the boolean exp from the circuit and simplify.



$Y = (A) \cdot (BCD) \cdot (A \oplus B + C)$

$Y = (ABCD) \cdot (\bar{A}B + A\bar{B} + C)$

$Y = ABCD\bar{A}B + ABCDA\bar{B} + ABCCD$

$Y = 0 + 0 + ABCCD$

$Y = ABCCD$

Combinational Logic Circuit

random values for example

Dec	A	B	C	Y
0	0	0	0	1
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Variable: $A = 1 \Rightarrow A$

$A = 0 \Rightarrow \bar{A}$

Truth Table
↓
Boolean exp. of Y

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

↓
Simplify & Circuit

3) A traffic light consists of red, yellow & green light. At any pt. of time only 1 light must be on. Design a comb. of logic circuit where an alarm would be sounded if the traffic light malfunctions.

Solve : → Truth Table
→ Form boolean exp.
→ Simplify exp
→ Make a circuit

R	Y	G	A
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$$A = \bar{R}\cdot\bar{Y}\cdot\bar{G} + \bar{R}\cdot Y\cdot G + R\cdot\bar{Y}\cdot G + R\cdot Y\cdot\bar{G} + R\cdot Y\cdot G$$

Boolean expression Simplification using Karnaugh Map. (K-Map)

*

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

B \ A	0	1
0	0	1
1	0	1

$Y = A$

Since A is at (1) is constantly 1 but B is [illegible] takes 0 and 1 so not count B

$Y = A\bar{B} + AB = A(\bar{B} + B) = A$

Steps / Rules -

1) Make groups of 1s
2) Vertically or Horizontally
3) No. of 1s in a group should be in powers of 2 (2^n)
4) Overlapping of groups is ~~not~~ allowed.

3- variable K- map

	A	B	C	Y
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

C \ AB	00	01	11	10
0	1	1	1	0
1	1	1	1	0

$\bar{A}$ B

$Y = \bar{A} + B$

Minterms & Sum of Products (SOP)

Minterm	A	B	C	Y
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

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

$Y = \Sigma m(0,3,4,6)$

Minterm: A term formed by the product of input variables

∴ Each row of a truth table can be a minterm

Minterm $0 \rightarrow \bar{A}\bar{B}\bar{C}$

" " $1 \rightarrow \bar{A}\bar{B}C$

" " $2 \rightarrow \bar{A}B\bar{C}$

" ⋮ " $7 \rightarrow ABC$

Q Simplify $f = \Sigma m(2,3,5,7)$ using K-map

C \ AB	00	01	11	10
0	0	1	0	0
1	0	1	1	1

$\bar{A}B$ AC

$f = \bar{A}B + AC$

Q Simplify $f = \Sigma m(0,3,4,7)$ using K-map

C \ AB	00	01	11	10
0	1	0	0	1
1	0	1	1	0

$\bar{B}\bar{C}$

BC

$f = BC + \bar{B}\bar{C}$

Q $f = \Sigma m(2,3,4,7)$

C \ AB	00	01 ($\bar{A}B$)	11	10
0	0	1	0	1 ← $A\bar{B}\bar{C}$
1	0	1	1	0

↓ BC

$f = \bar{A}B + BC + A\bar{B}\bar{C}$

Q $\Sigma m(0,1,2,3,5)$

If its a pair of 4 then only 1 term

C \ AB	00	01	11	10
0	1	1	0	0
1	1	1	0	1

↓ $\bar{A}$ ↑ $\bar{B}C$

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

HW:
1) $f = \Sigma m(0,1,6,7)$
2) $f = \Sigma m(0,2,4,6)$
3) $f = \Sigma m(1,2,3,5,7)$
4) $f = \Sigma m(0,1,4,5)$
5) $f = \Sigma m(1,3,4,5,6,7)$

DATE / /

★ 4 variable K-map

$B\bar{C}$

CD \ AB	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	1	0	0	0
10	1	0	0	0

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

simplify:

$f = \sum m(2,3,4,5,12,13)$

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

Q. simplify

$f = \sum m(4,6,7,12,14,15)$

$B\bar{D}$

CD \ AB	00	01	11	10
00	0	1	1	0
01	0	0	0	0
11	0	1	1	0
10	0	1	1	0

BC

$f = B\bar{D} + BC$

Q. simplify

$f = \sum m(0,2,5,7,8,10,13,15)$

BD

CD \ AB	00	01	11	10
00	1	0	0	1
01	0	1	1	0
11	0	1	1	0
10	1	0	0	1

$\bar{B}\bar{D}$

Wrap along col. and rows similarly

$f = BD + \bar{B}\bar{D}$
$= B \odot D$

Q $f = \Sigma m(0,1,4,5,6,8,12)$

~~ABCD~~

when vertical take AB.
when horizontal take CD

CB

CD \ AB	00	01	11	10
00	1	1	1	1
01	1	1	0	0
11	0	0	0	0
10	0	1	0	0

$\bar{C}\bar{A}$

$\bar{C}\bar{D}$

$\bar{A}B\bar{D}$

$f = \bar{A}\bar{C} + \bar{A}B\bar{D} + \bar{C}\bar{D}$

Universal Gates

NAND - $\boxed{\text{NOT, AND, OR}}$, EX-OR

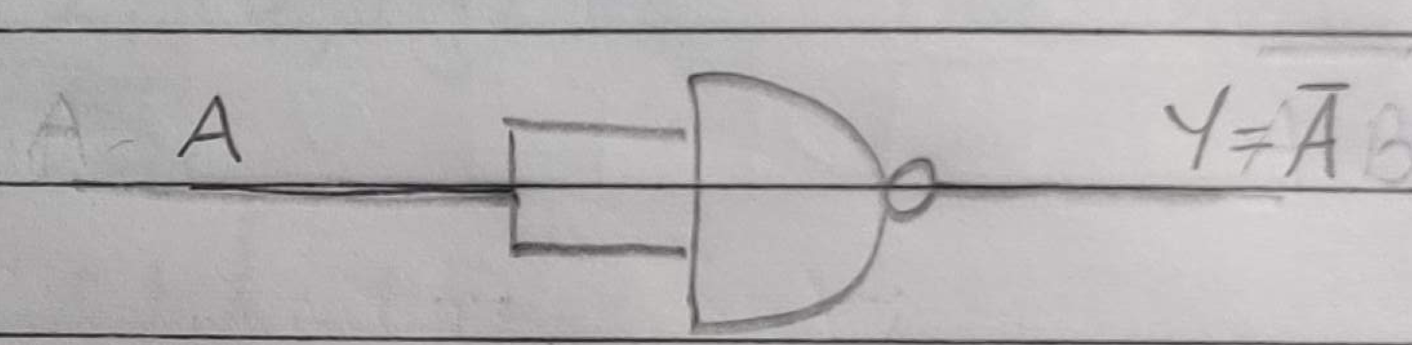
NOR - NOT, AND, OR

★ NAND : $Y = \overline{A \cdot B}$

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

1] NOT using NAND

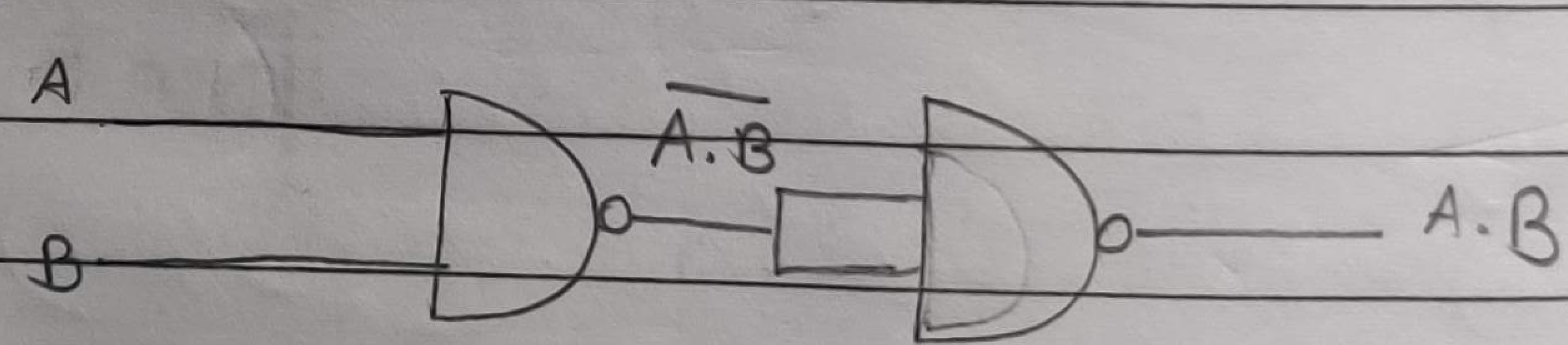
$Y = \overline{A}$



2] AND using NAND

AND ⟶ NAND + NOT

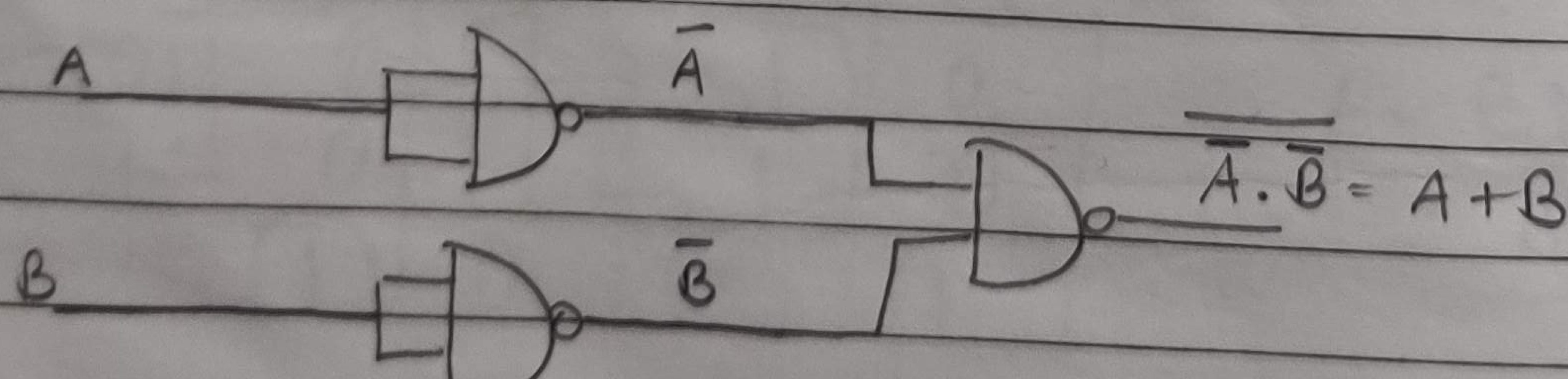
$A \cdot B = \overline{\overline{A \cdot B}}$



3] OR using NAND

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

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



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

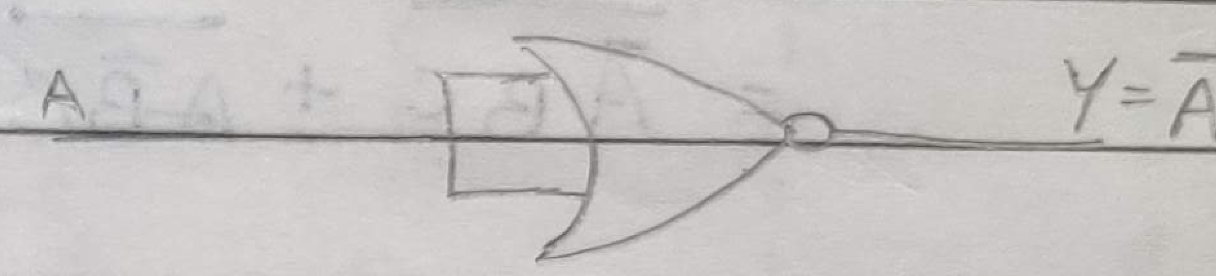
Q $\bar{A}\cdot B + \overline{A\cdot\bar{B}}$ [EX-OR] using NAND.

★ NOR: $Y = \overline{A+B}$

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

1] NOT using NOR gate

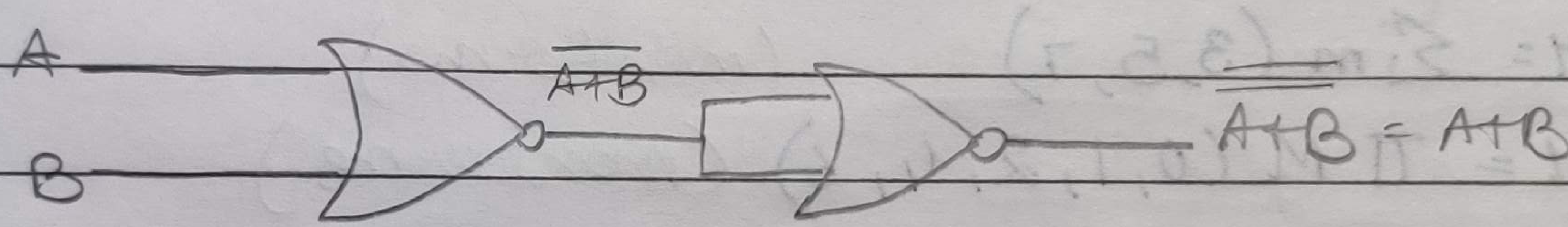
$Y = \bar{A}$



2] ~~AND~~ OR using NOR

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

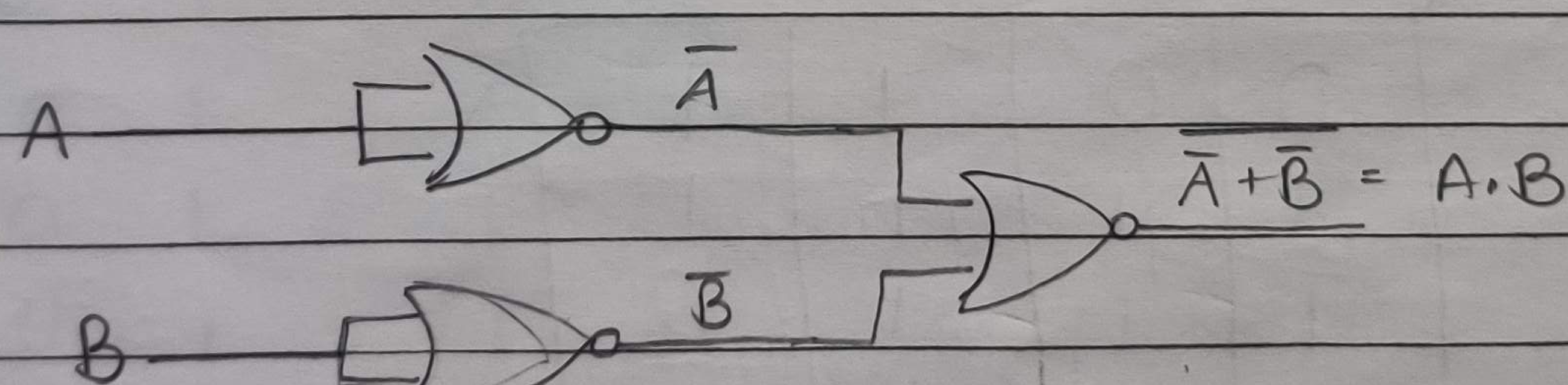
OR → NOR + NOT



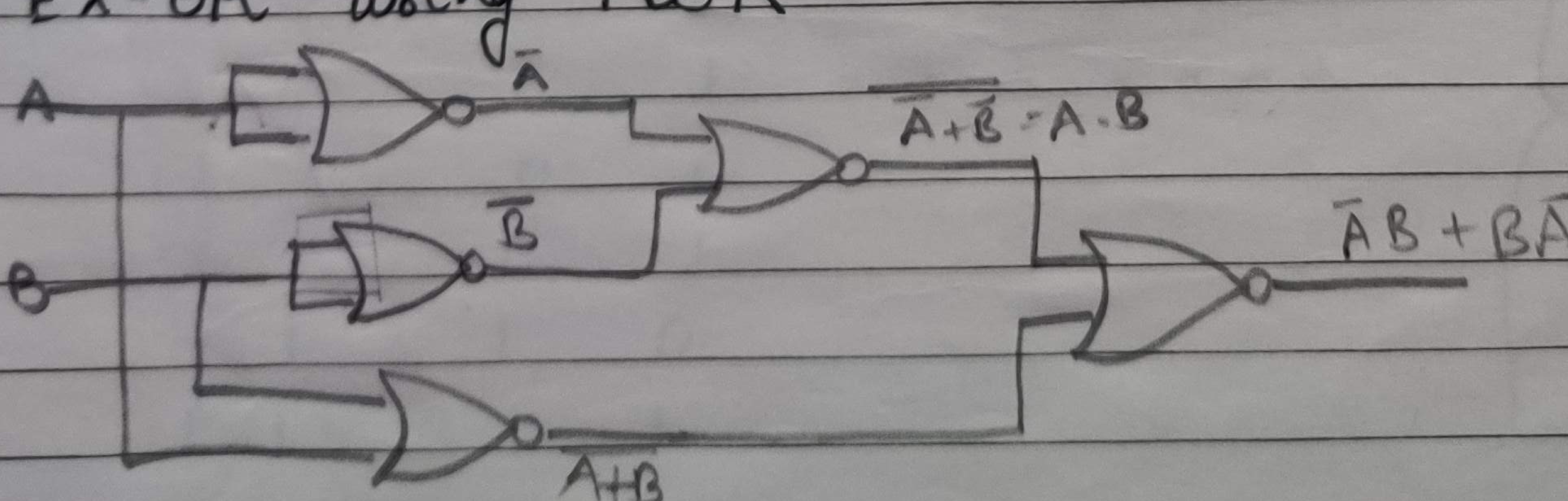
3] AND using NOR

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

$A\cdot B = \bar{A}$ NOR $\bar{B}$



4] EX-OR using NOR



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

minterm minterm minterm → product

Sum of Product (SOP)

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

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

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

maxterm maxterm maxterm

Product of Sum POS

$Y = \Sigma m(3, 5, 7)$ (minterms)

$Y = \Pi M(0, 1, 2, 4, 6)$ (maxterms)

when putting this in K-map input 0 in those places and 1 in the rest.

eg. $Y = \Pi M(0, 2, 4, 11)$

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

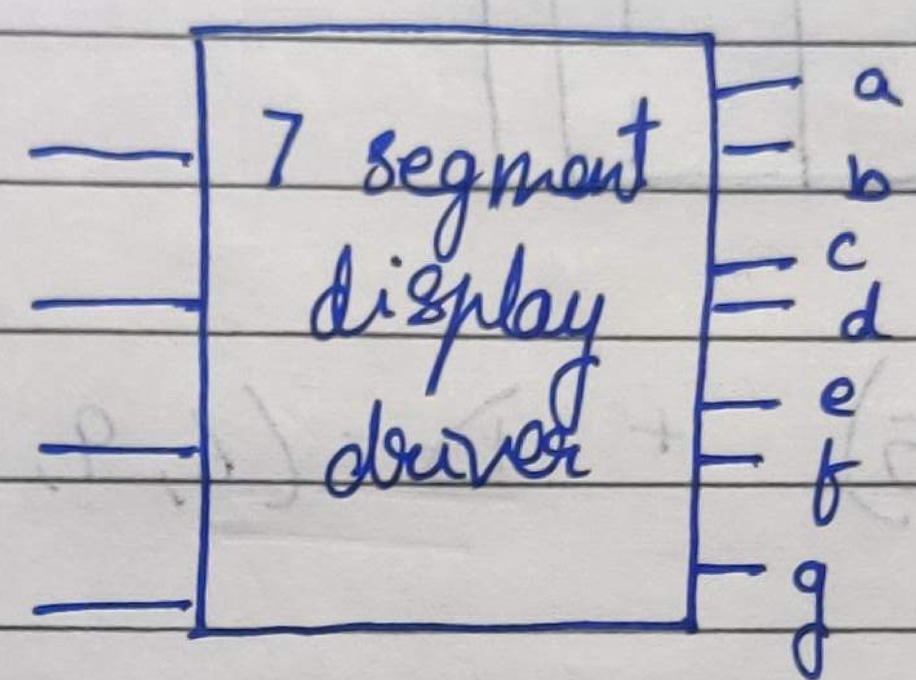
DATE 18/8/22

K-map with Don't Care / Can't happen addition

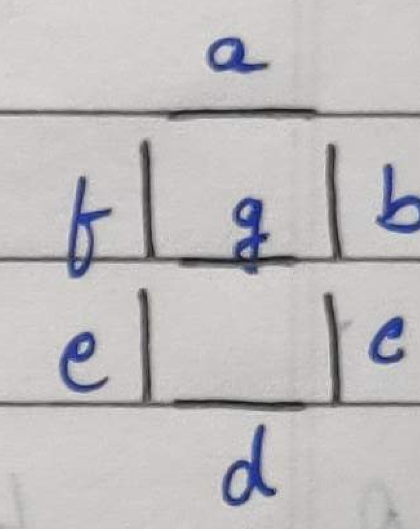
Binary to 7-segment converter

1 ⟶ b.c

⑤	①
0	0
1	0
0	0
1	1



	①	⑤
a	0	1
b	1	0
c	1	1
d	0	1
e	0	0
f	0	1
g	0	1
	1	5



Decimal	B_3	B_2	B_1	B_0	a	b	c	d	e	f	g
	0	0	0	0	1	1	1	1	1	1	0
	0	0	0	1	0	1	1	0	0	0	0
	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	0	0	0							
	1	0	0	1	x	x	x	x	x	x	x
	1	0	1	0	x	x	x	x	x	x	x
	1	0	1	1	x	x	x	x	x	x	x
	1	1	0	0							
	1	1	0	1							
	1	1	1	0							
	1	1	1	1							

don't care / Can't happen condition

Simplify:

Q $f = \sum m(8,9,10,11,12,13,14) + d(3,4,7,15)$

CD \ AB	00	01	11	10
00	0	X	1	1
01	0	0	1	1
11	X	X	X	1
10	0	0	1	1

$f = A$

Q $f = \sum m(2,5,7,15) + D = (6,9,13)$

CD \ AB	00	01	11	10
00	0	0	0	0
01	0	1	X	X
11	0	1	1	0
10	1	X	0	0

$f = BD + \bar{A}C\bar{D}$

Q $f = \sum m(2,3,5,8,11,12)$; $D = (6,15)$

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

CD \ AB	00	01	11	10
00	0	0	1	1
01	0	1	0	0
11	1	0	X	1
10	1	X	0	0

$A\bar{C}\bar{D}$

$\bar{B}CD$

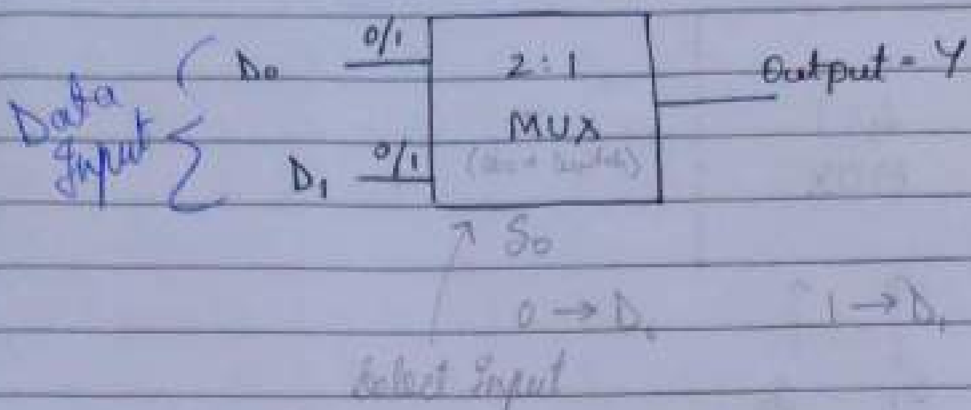
$\bar{A}C\bar{D}$

$f = \bar{A}B\bar{C}D + A\bar{C}\bar{D} + \bar{B}CD + \bar{A}C\bar{D}$

DATE: 1/9/22

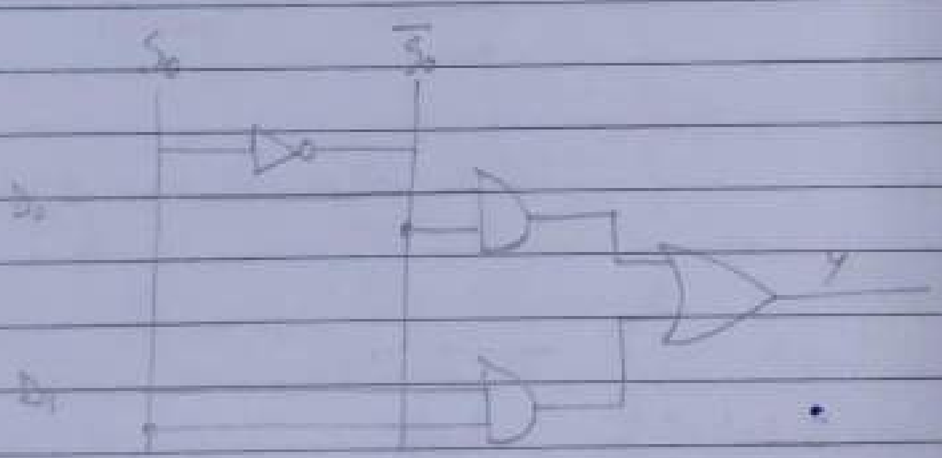
Multiplexer (MUX)

Multiple Inputs → single output

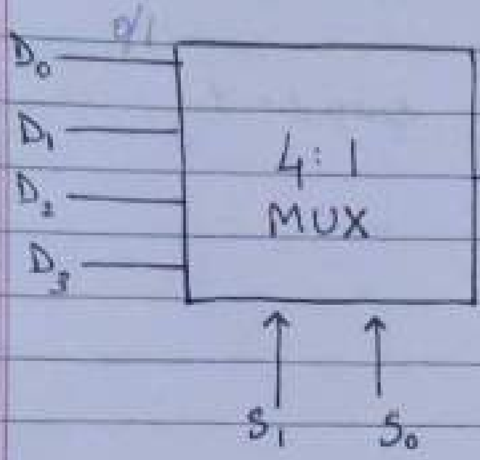


Select Input S_0	Output Y
0	D_0
1	D_1

$$Y = D_0 \bar{S}_0 + S_0 D_1$$

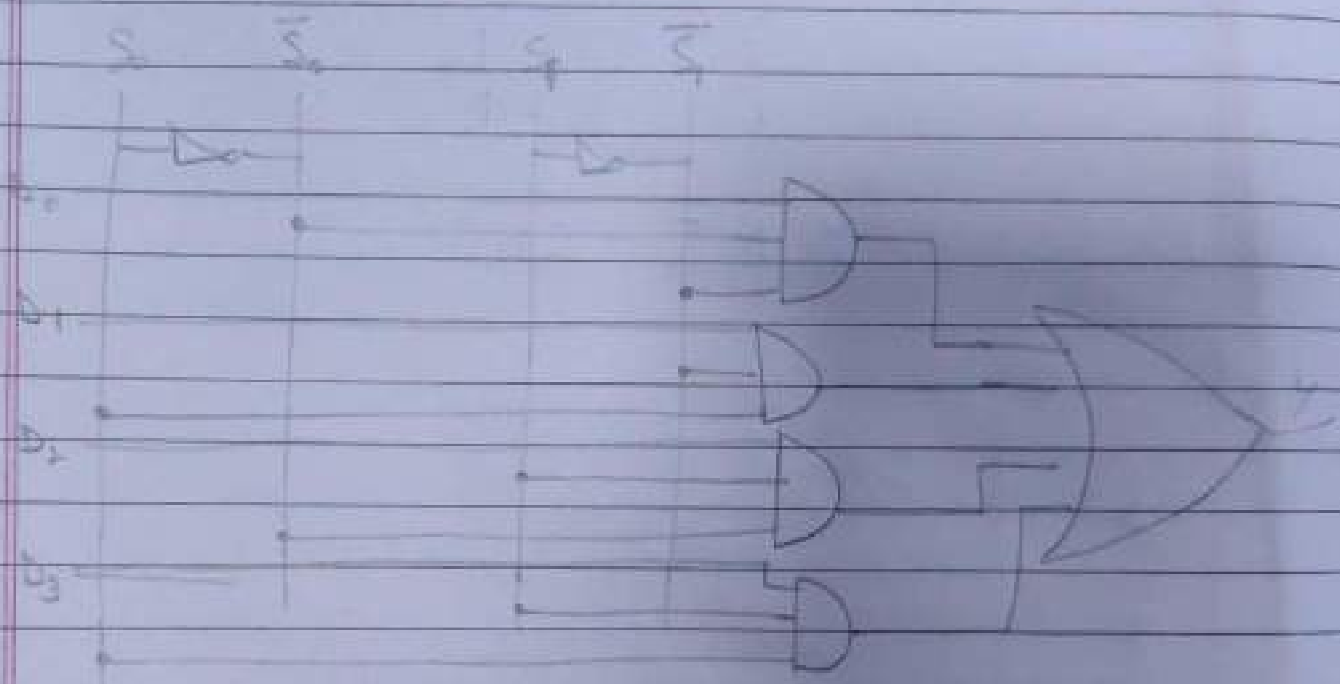


4:1 MUX



S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

$$Y = D_0 \bar{S_1} \bar{S_0} + D_1 \bar{S_1} S_0 + D_2 S_1 \bar{S_0} + D_3 S_1 S_0$$



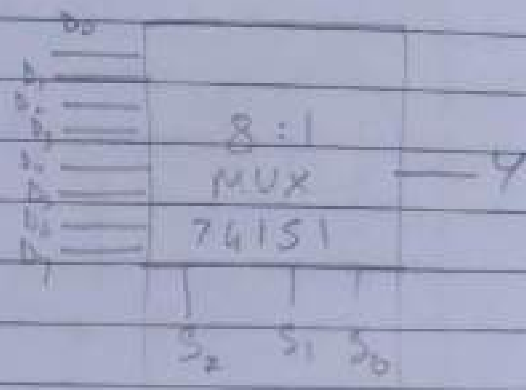
$S = 1 \rightarrow 2^1 = 2$ Data

$S = 2 \rightarrow 2^2 = 4$ Data

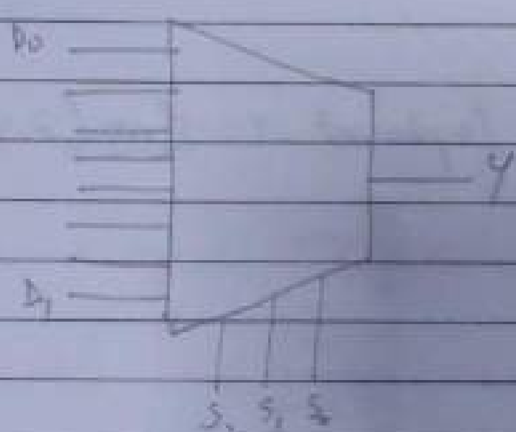
$S = 3 \rightarrow 2^3 = 8$ Data

8:1 MUX

S_2	S_1	S_0	Y
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7



S → Select Input
D → Data Inputs



Q

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

$Y = \Sigma m(2,3,5,7)$

K-map

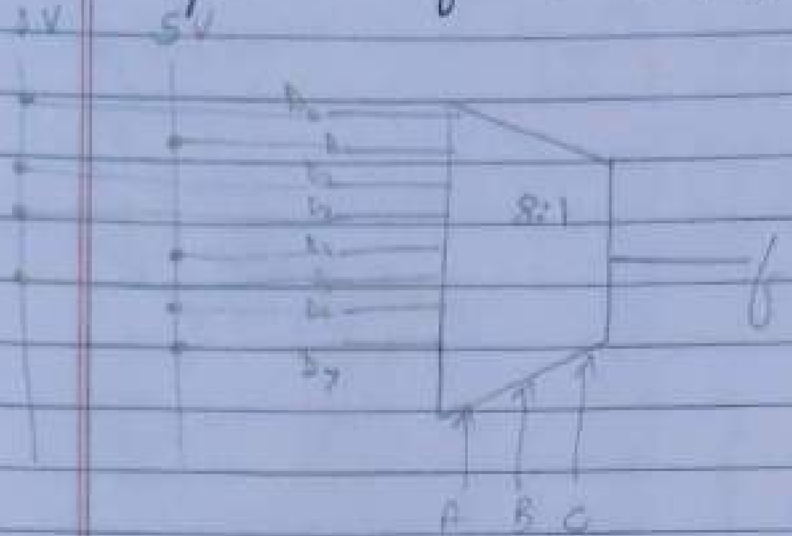
$Y = \bar{A}B + BC + \bar{A}C$

OR

$D_0 \rightarrow 0V$
$D_1 \rightarrow 0V$
$D_2 \rightarrow 5V$

$D_4 \rightarrow 0V$
$D_5 \rightarrow 5V$
$D_6 \rightarrow 0V$
$D_7 \rightarrow 5V$

Implement $f = \sum m(1,4,6,7)$ using 8:1 MUX



V.L.M
↓
Universal Logic Module

Implement $Y = \sum m(2,4,7,9,11,13)$ using MUX

DATE 7/9/22

Implement the following logic expressions using 8:1 MUX.

1) $f = \Sigma m\,(0, 2, 5, 7, 8, 10, 13, 15)$

2) $f = \Sigma m\,(0, 1, 2, 3, 4, 8, 12)$

3) $f = \Sigma m\,(1, 3, 4, 6, 9, 11, 12, 14)$

4) $f = \Sigma m\,(1, 2, 8, 9, 13, 14)$

5) $f = \Sigma m\,(0, 1, 2, 3, 8, 9, 10, 11)$

6) $f = \Sigma m\,(0, 1, 4, 8, 9, 10, 11, 15)$

7) $f = \Sigma m\,(5, 6, 7, 9, 10, 11, 13, 15)$

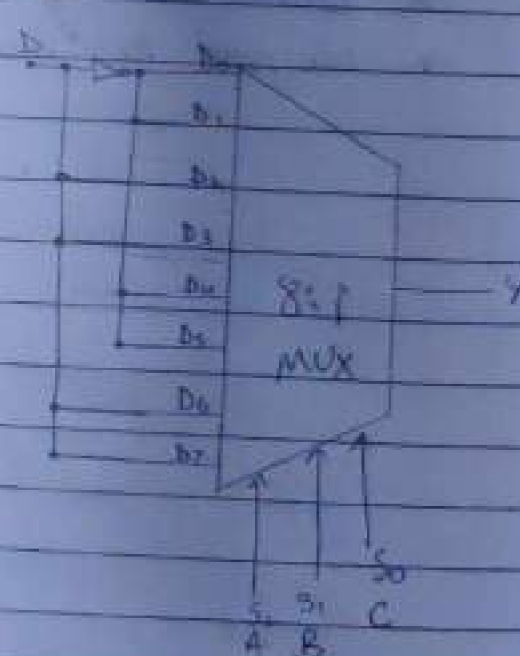
8) $f = \Sigma m\,(0, 4, 8, 9, 10, 11, 12)$

→ 16 row truth table
→ 8 row truth table
→ Connection diagram using 8:1 MUX
→ No need to mention pin numbers.

1) $f = \sum m\,(0, 2, 5, 7, 8, 10, 13, 15)$

A	B	C	D	Y	
0	0	0	0	1	$\bar{D}$
0	0	0	1	0	
0	0	1	0	1	$\bar{D}$
0	0	1	1	0	
0	1	0	0	0	D
0	1	0	1	1	
0	1	1	0	0	D
0	1	1	1	1	
1	0	0	0	1	$\bar{D}$
1	0	0	1	0	
1	0	1	0	1	$\bar{D}$
1	0	1	1	0	
1	1	0	0	0	D
1	1	0	1	1	
1	1	1	0	0	D
1	1	1	1	1	

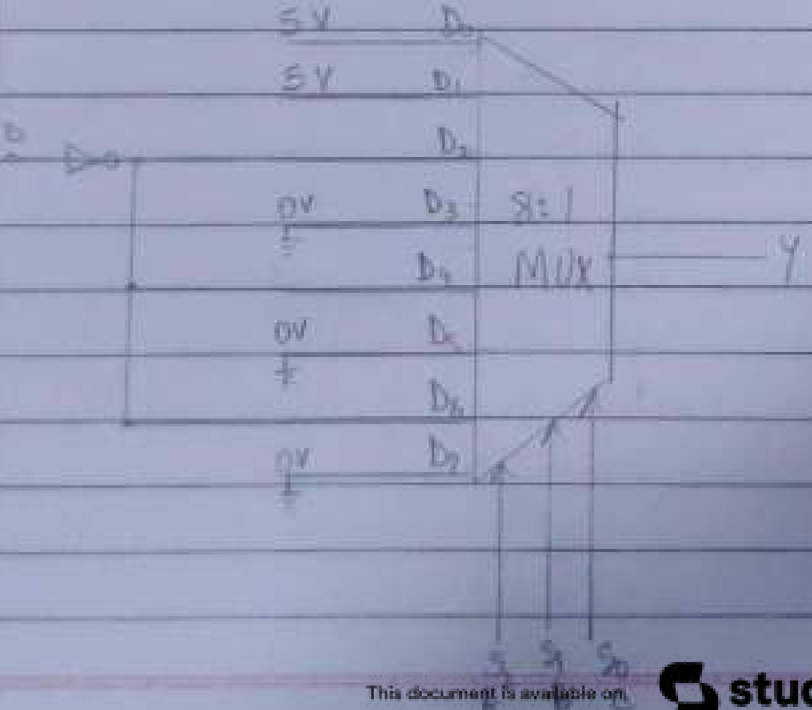
A	B	C	Y
0	0	0	$\bar{D}$
0	0	1	$\bar{D}$
0	1	0	D
0	1	1	D
1	0	0	$\bar{D}$
1	0	1	$\bar{D}$
1	1	0	D
1	1	1	D



2) $f = \Sigma m(0, 1, 2, 3, 4, 8, 12)$

A	B	C	D	Y	
0	0	0	0	1	1
0	0	0	1	1	
0	0	1	0	1	1
0	0	1	1	1	
0	1	0	0	1	$\bar{D}$
0	1	0	1	0	
0	1	1	0	0	0
0	1	1	1	0	
1	0	0	0	1	$\bar{D}$
1	0	0	1	0	
1	0	1	0	0	0
1	0	1	1	0	
1	1	0	0	1	$\bar{D}$
1	1	0	1	0	
1	1	1	0	0	0
1	1	1	1	0	

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	$\bar{D}$
0	1	1	0
1	0	0	$\bar{D}$
1	0	1	0
1	1	0	$\bar{D}$
1	1	1	0

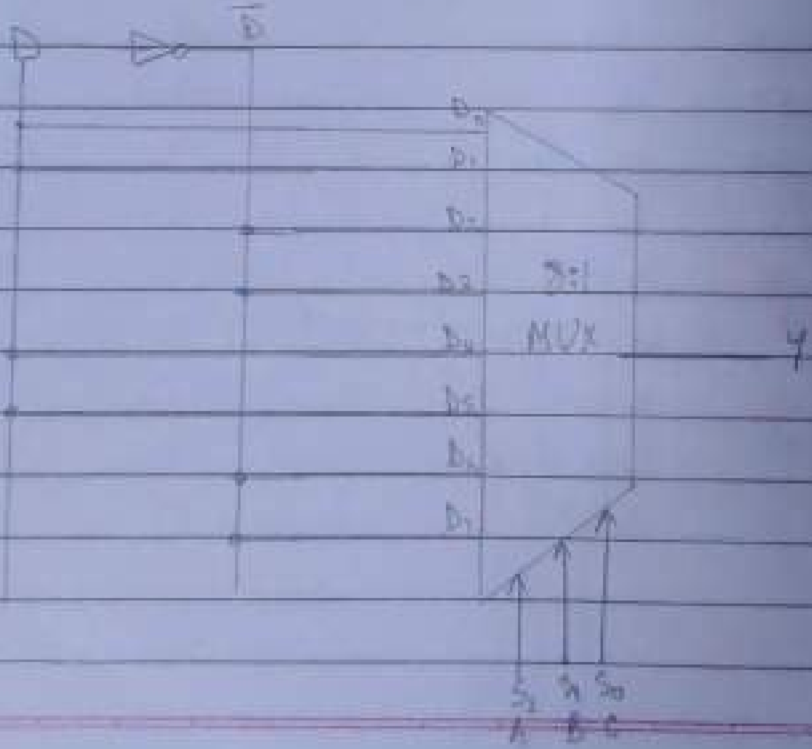


PAGE No.	
DATE	/ /

3) $y = \Sigma m(1, 3, 4, 6, 9, 11, 12, 14)$

A	B	C	D	Y	
0	0	0	0	0	D
0	0	0	1	1	
0	0	1	0	0	D
0	0	1	1	1	
0	1	0	0	1	$\bar{D}$
0	1	0	1	0	
0	1	1	0	1	$\bar{D}$
0	1	1	1	0	
1	0	0	0	0	D
1	0	0	1	1	
1	0	1	0	0	D
1	0	1	1	1	
1	1	0	0	1	$\bar{D}$
1	1	0	1	0	
1	1	1	0	1	$\bar{D}$
1	1	1	1	0	

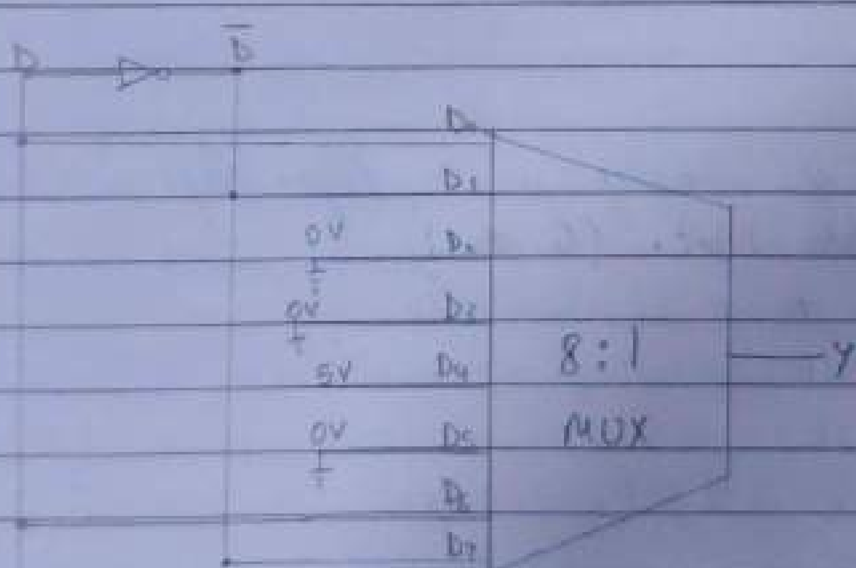
A	B	C	Y
0	0	0	D
0	0	1	D
0	1	0	$\bar{D}$
0	1	1	$\bar{D}$
1	0	0	D
1	0	1	D
1	1	0	$\bar{D}$
1	1	1	$\bar{D}$



4) $f = \Sigma m(1, 2, 8, 9, 13, 14)$

A	B	C	D	Y	
0	0	0	0	0	D
0	0	0	1	1	
0	0	1	0	1	$\bar{D}$
0	0	1	1	0	
0	1	0	0	0	0
0	1	0	1	0	
0	1	1	0	0	0
0	1	1	1	0	
1	0	0	0	1	1
1	0	0	1	1	
1	0	1	0	0	0
1	0	1	1	0	
1	1	0	0	0	D
1	1	0	1	1	
1	1	1	0	1	$\bar{D}$
1	1	1	1	0	

A	B	C	Y
0	0	0	D
0	0	1	$\bar{D}$
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	D
1	1	1	$\bar{D}$



PAGE No.
DATE 8/9/22

Sequential Logic Design

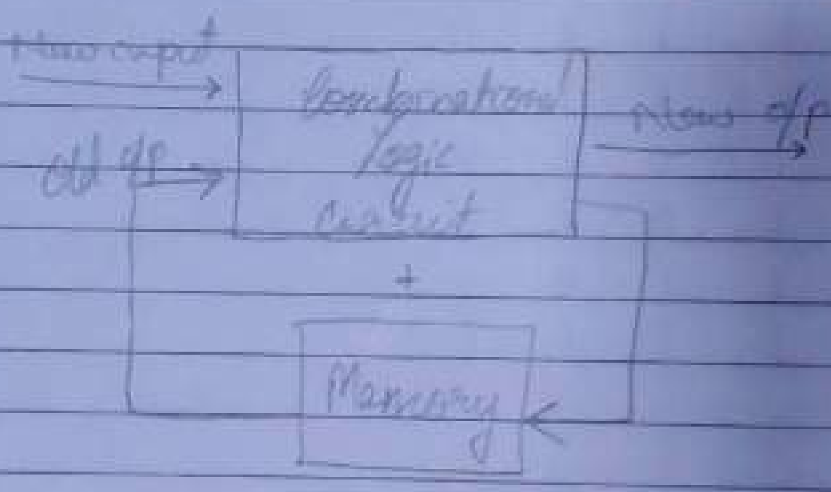
Combinational Logic Circuits:

Present input → present o/p

present Input + Past o/p → Present o/p

Get Bottle → Fill bottle → Cap the bottle → label → Carton

Sequential Circuit



* Basic unit of Memory is a Flip-Flop

→ FF can store 1 bit. (0 or 1)

1) S-R ff → Set Reset ff
2) J-K ff
3) D ff
4) T ff

1) S-R FF (Set Reset)

State Transition Table

	S (Set input) (next input)	R (Reset) (reset i/p)	Q (~~test~~ present output)	Q+ (next / future o/p)	
No change	0	0	0	0	o/p will not get updated when S=R=0
No change	0	0	1	1	
Reset	0	1	0	0	o/p will reset to 0 cuz R=1
	0	1	1	0	
Set	1	0	0	1	o/p will be set to 1 cuz S=1
	1	0	1	1	
Invalid	1	1	0	X	not used
	1	1	1	X	

K-map for Q+

Q \ SR	00	01	11	10
0	0	0	X	1
1	1	0	X	1

S

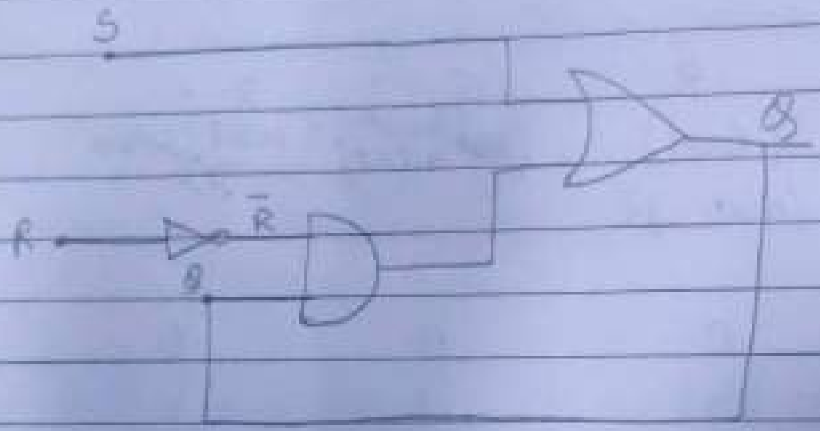
$\bar{R}Q$

$$Q+ = S + \bar{R}Q$$

PAGE No.
DATE / /

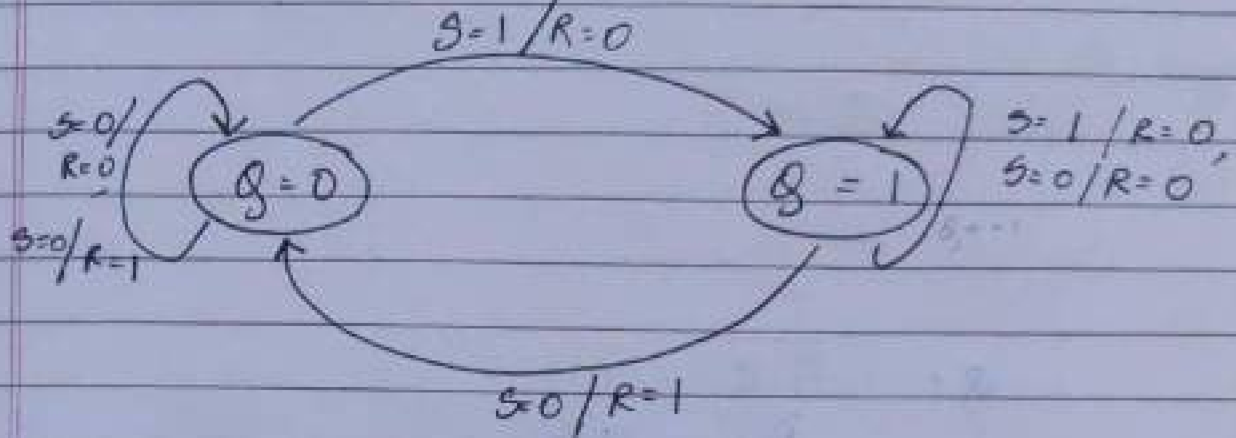
$Q^+ = S + \bar{R} Q$

No Q^+ in the diagram

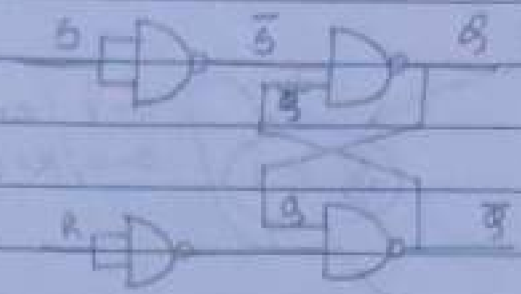


H.W) S-R FF using only NAND or only NOR

State Transition Diagram:



S-R Flip flop circuit using NAND gates



$$Q^{+} = \overline{\bar{S} \cdot \bar{Q}} \quad -(1)$$
$$\bar{Q} = \overline{R \cdot Q} \quad -(2)$$

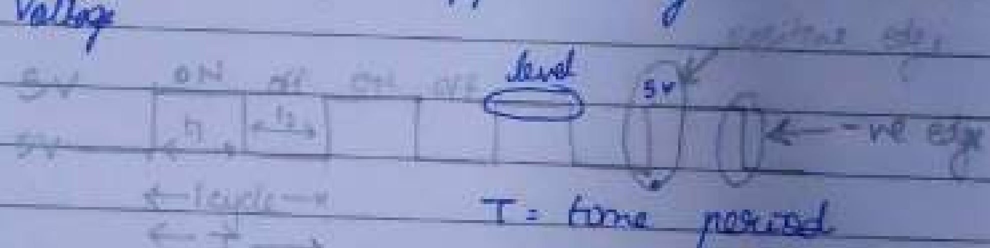
Substituting 2 in 1

$$Q^{+} = \overline{\bar{S} \cdot \overline{RQ}}$$
$$\therefore Q^{+} = \bar{\bar{S}} + \overline{\overline{RQ}}$$
$$\therefore Q^{+} = S + \bar{R}Q$$

★ Clocks:

→ Clock signal is a square / pulse voltage

→ AC Voltage



T = time period

$f = \frac{1}{T}$ $\quad t_1 + t_2 = T$

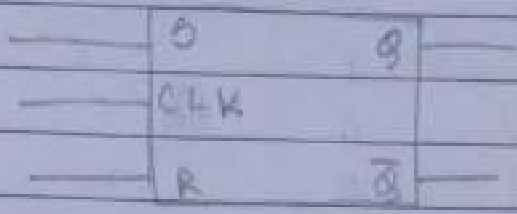
duty cycle = $\frac{t_1}{T} \times 100\%$

50% duty cycle

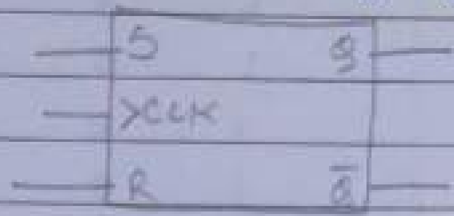
1) When clock is high (5V)
(level triggering)
2) When clock is going from (0 to 5V)
(positive edge triggering)
3) Negative edge triggering

* Symbol:

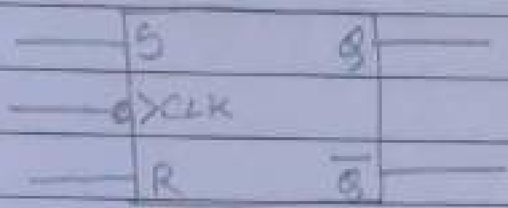
Level triggering



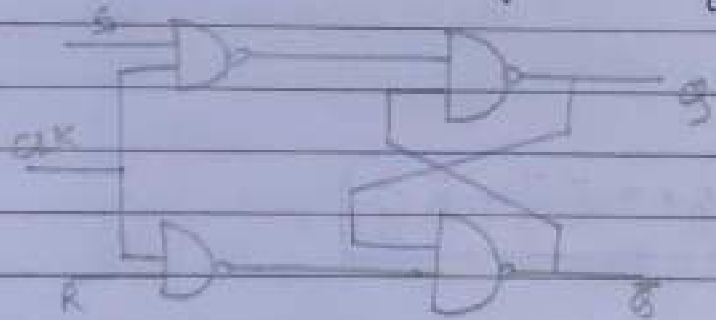
Positive edge triggering



Negative edge triggering



Clocked S-R ff using NAND gates



NAND

S	CLK	Y
0	0	1
0	1	1
1	0	1
1	1	0

Disadvantage:

Invalid output when $S=R=1$

To overcome this, J-K ff is used

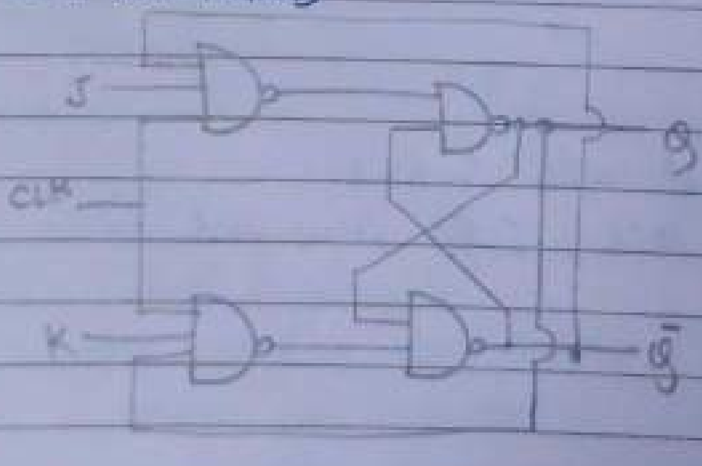
State transition table of J-K FF

CLK	J	K	Q	Q+	
0	x	x	x	Q	
1	0	0	0	0	No change
1	0	0	1	1	
1	0	1	0	0	Reset
1	0	1	1	0	
1	1	0	0	1	Set
1	1	0	1	1	
1	1	1	0	1	Toggle
1	1	1	1	0	

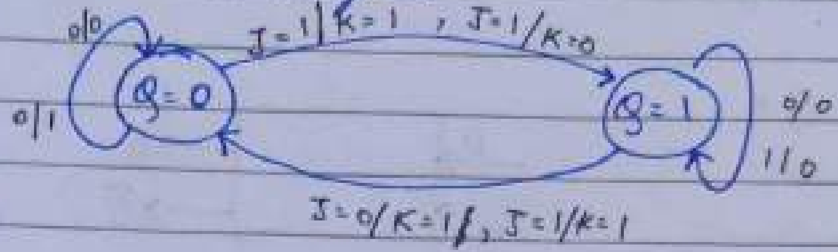
Q \ JK	00	01	11	10	
0	0	0	1	1	$J\bar{Q}$
1	1	0	0	1	$\bar{K}Q$

$$\boxed{Q^+ = J\bar{Q} + \bar{K}Q}$$

CKT with NAND



State Transition Diagram :



Excitation Table

Q	Q+	J K		J	K
0	0	0 0	0 1	0	X
0	1	1 1	0 1	1	X
1	0	0 1	1 1	X	1
1	1	0 1	0 0	X	0

3) ~~Q~~ (5-10 mks)

State Transition Table :

State excitation table

State transition diagram

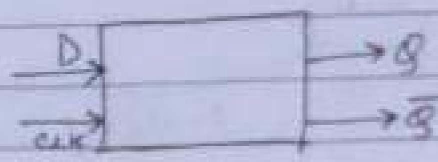
K-Map

Circuit diagram with CLK

Using NAND gates or NOR gates

3] D (Delay FF)

D	Q^+
0	0
1	1

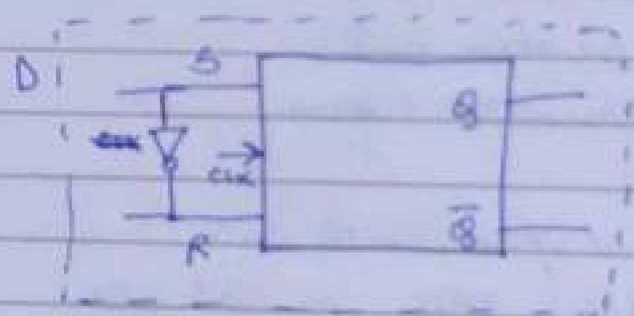


It can be made by SR ~~or~~ JK flip flop by connecting an ~~inverter~~ inverter after the inputs.

Converting SR to D FF

S	R	Q^+
0	Q	Q
0	1	0
1	0	1
1	1	Invalid

$\left.\begin{matrix}0\\1\end{matrix}\right\} Q^+ = S$

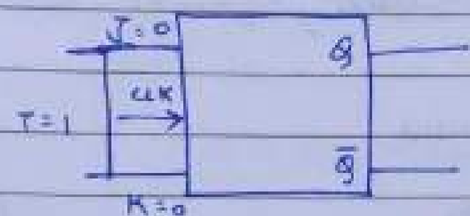


Here value of R is inversion of S

4] T (Toggle FF)

T	Q	Q+	
0	0	0	No change
0	1	1	
1	0	1	Toggle
1	1	0	

T flip flop can be constructed from J-K FF by connecting the two inputs together.



J	K	Q+
0	0	Q
0	1	0
1	0	1
1	1	Toggle

Q Can we use SR FF to make T FF
No. Since when both inputs on SR is 1, the output is invalid but for T-FF, when both inputs is 1, the output is toggle.

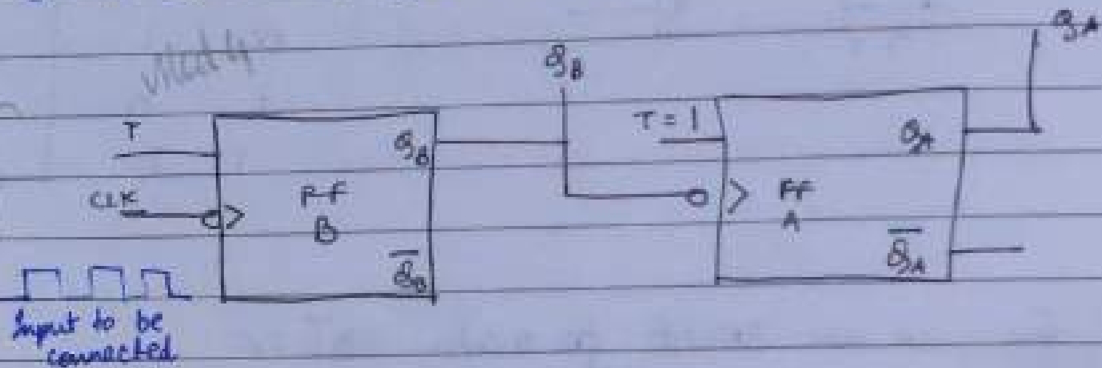
Applications of FF

1) ~~Mean~~ Memory element → Store 1 bit
2) Counter → Synchronous / Asynchronous
3) Shift register

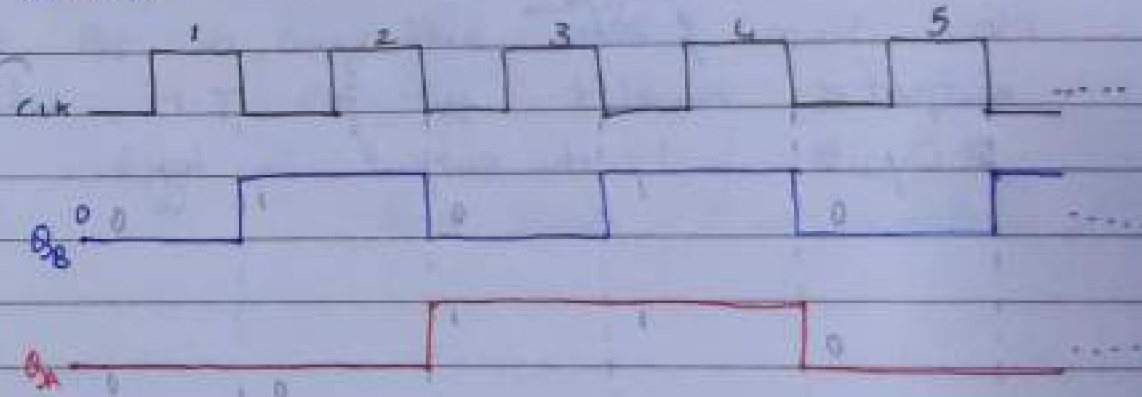
Asynchronous Counter

→ Used to count clock pulses
→ Every FF gets a different clock, hence the name 'asynchronous'
→ J-K or T-FF can be used

Circuit Diagram



Timing Diagram



CLK (Pulse)	Q_A	Q_B	Count
0	0	0	0
1	0	1	1
2	1	0	2
3	1	1	3
4	0	0	0

State Transition Table

States of the counter →

00
01
10
11

4 states / Modulus

∴ This counter is called Modulo - 4 asyncounter

★ Modulo is decided by no. of FFs

2 FFs → $2^2 = 4$ ~~Mod~~ Mod - 4

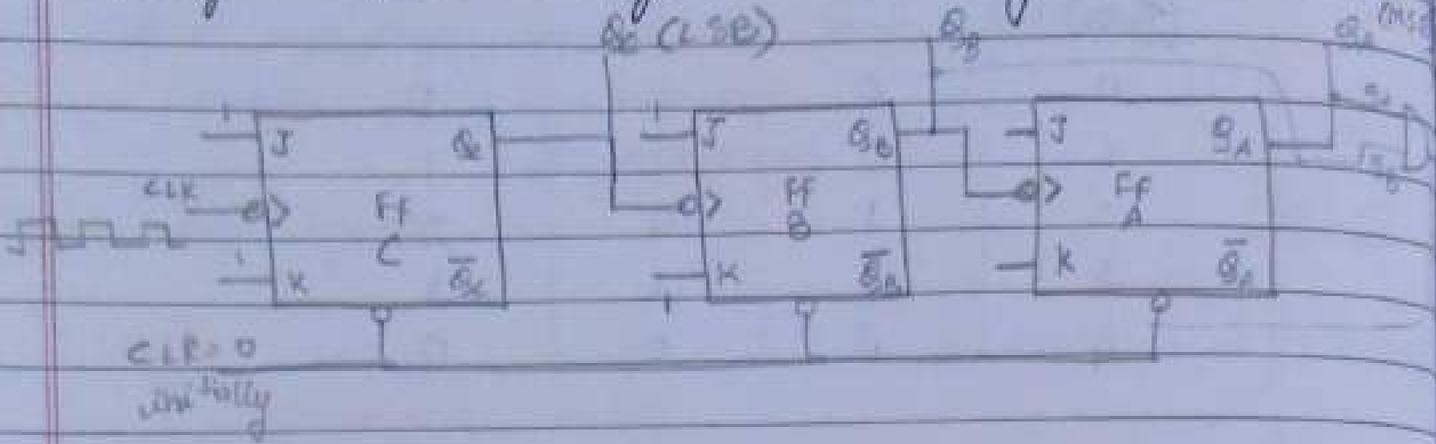
3 FFs → $2^3 =$ ~~8~~ Mod - 8

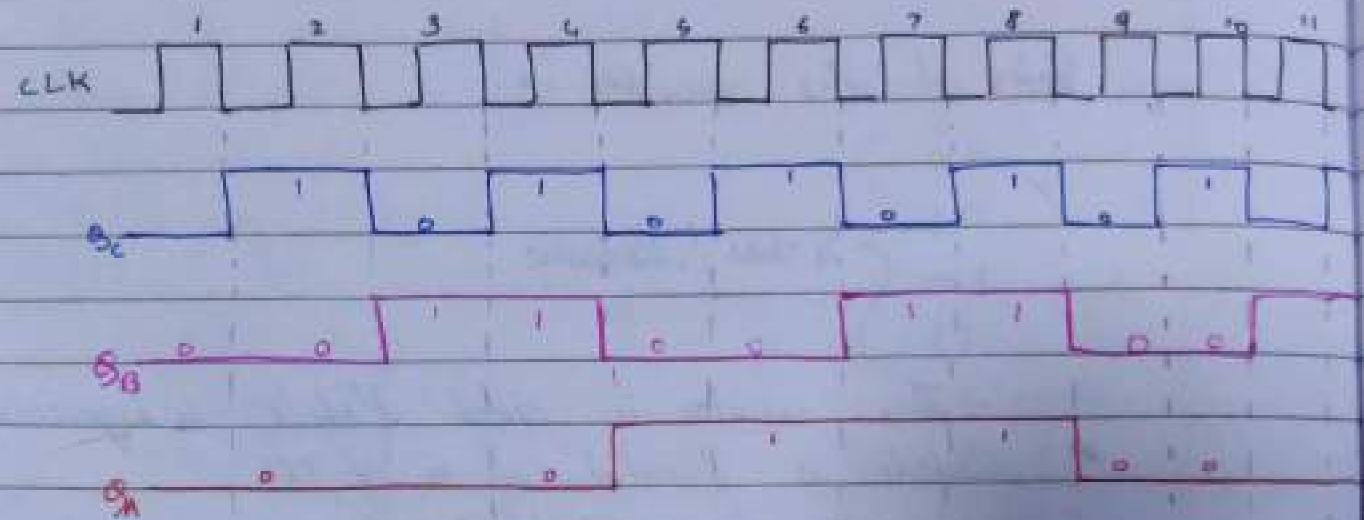
4 FFs → $2^4 =$ Mod - 16

Max count = No. of states - 1

PAGE No.
DATE / /

Design a Mod-8 asyn counter using J-K FFs





CLK	QA	QB	Qc	Count
0	0	0	0	0
1	0	0	1	1
2	0	1	0	2
3	0	1	1	3
4	1	0	0	4
5	1	0	1	5
6	1	1	0	6
7	1	1	1	7
8	0	0	0	0

PAGE No.	
DATE	13/9/22

Experiment

Verification of Truth Table of J-K ff

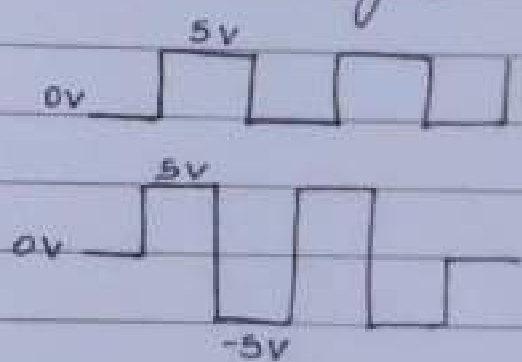
★ J-K ff ~~f.T~~

Clock	J	K	Q+
0	x	x	No change (Q)
1	0	0	No change (Q)
1	0	1	0
1	1	0	1
1	1	1	Toggle

Toggle means $1 \longrightarrow 0$ &
$0 \longrightarrow 1$

IC used :- IC7476

Clock signal :-

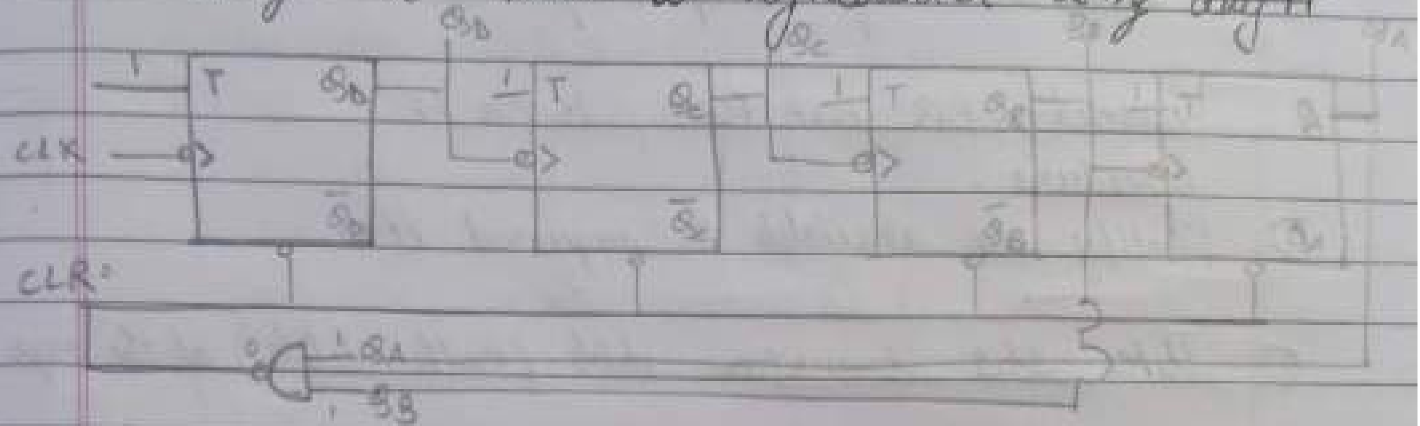


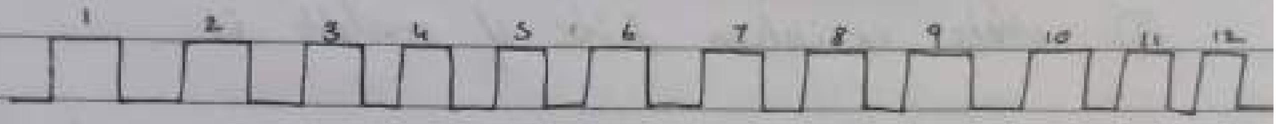
MOD-12

⇒ 12 counts

⇒ 0 - 11

Design a mod-12 async counter (ripple) counter. ~~using~~ any FF





CLK	Q_A	Q_B	Q_C	Q_D	Count
0	0	0	0	0	
1	0	0	0	1	
2	0	0	1	0	
3	0	0	1	1	
4	0	1	0	0	
5	0	1	0	1	
6	0	1	1	0	
7	0	1	1	1	
8	1	0	0	0	
9	1	0	0	1	
10	1	0	1	0	
11	1	0	1	1	
12	1 0	1 0	0 0	0 0	

DATE 21/9/22

☆ Conversion of Flip Flops

Convert SR flip flop to JK FF →

Procedure:

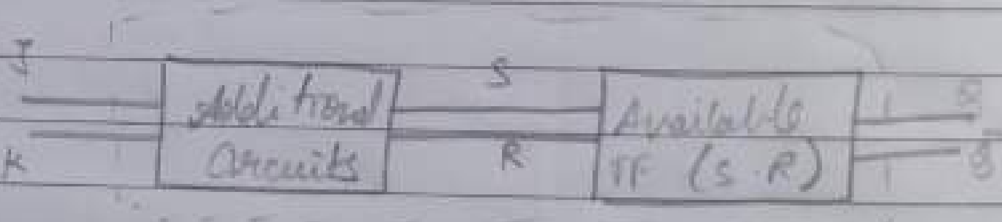
① Identify the available & required FFs
   (available ↓ SR, required ↓ JK)

② Make state transition table (truth table) of the req. FF

③ Make excitation table of available FF

④ Simplify and write boolean expression of req. FF in terms of req. FF

⑤ Make the circuit diagram:



1) SR → D ✓ inverter b/w inputs
   SR → JK ~~[illegible]~~
   SR → T × not possible
2) ~~J [illegible]~~
   JK → T ✓ connect input together
   JK → D
3) T → SR

① Available : SR FF , required : JK - FF

Truth Table (State table) of JK FF

②

J	K	Q	Q+	S	R
x	x	x			
0	0	0	0	0	x
0	0	1	1	x	0
0	1	0	0	0	x
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	x	0
1	1	0	1	1	0
1	1	1	0	0	1

Check Q & Q+

③ Excitation Table of SR FF

Q	Q+	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

④ Find Boolean exp. for S & R in terms of J, K & Q

K-map for S:

Q \ JK	00	01	11	10
0	0	0	1	1
1	x	0	0	x

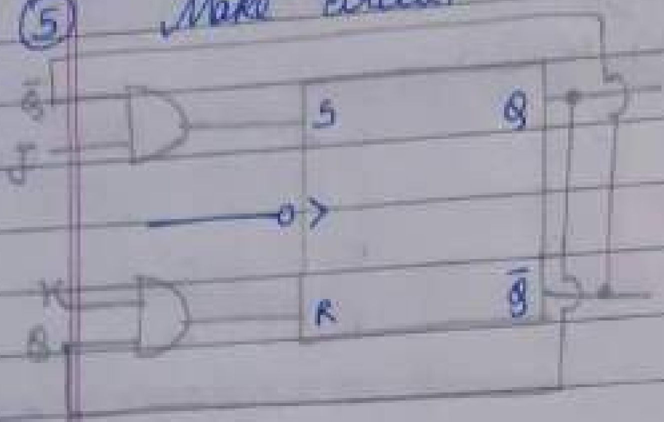
$S = J\bar{Q}$

K-Map for R

Q \ JK	00	01	11	10
0	x	x	0	0
1	0	1	1	0

$R = KQ$

⑤ Make circuit



Q Convert JK FF to D FF

① Available : JK FF , Required : D-FF

② Transition table of D-FF

value of D = Value of Q^+

Table 1

D	Q	Q^+	J	K
0	0	0	0	x
0	1	0	x	1
1	0	1	1	x
1	1	1	x	0

③ Excitation table of JK-FF

Table 2

Q	Q^+	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

Table 3

④ Find boolean exp of J & K in terms of D & Q

Q \ D	0	1
0	0	1
1	x	x

K-Map of J

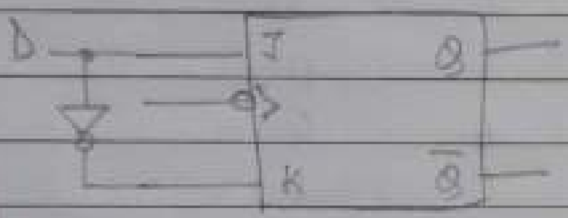
$J = D$

K-Map of K

Q \ D	0	1
0	x	x
1	1	0

$K = \bar{D}$

⑤ Circuit



Q] Convert T-ff to SR ff

① Available : T ff   Required = SR ff

② Transition ~~of~~ table of SR-ff

S	R	Q	Q+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	Invalid
1	1	1	Invalid

③ ~~Q~~ Q

③ Excitation table

Q	Q+	T
0	0	0
0	1	1
1	0	1
1	1	0

S	R	Q	Q+	T
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	0
1	~~0~~1	0	Invalid	x
1	1	1	Invalid	x

④ Boolean exp.

K-Map for T

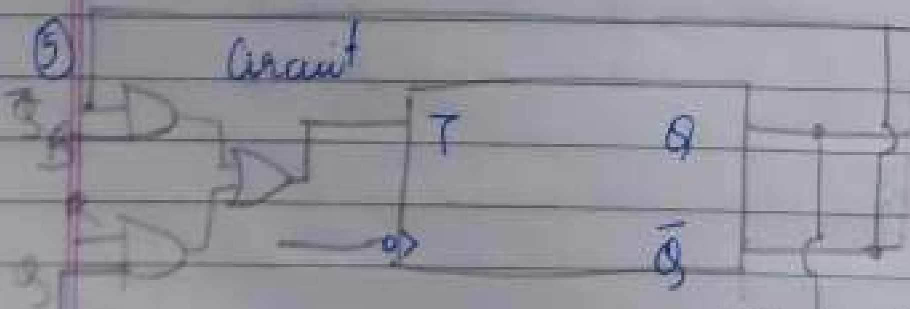
Q\SR	00	01	11	10
0	0	0	x	1
1	0	1	x	0

$\rightarrow S\bar{Q}$

$\rightarrow RQ$

$T = S\bar{Q} + RQ$

⑤ Circuit



DATE 22/9/22

How to represent various clocks in truth table of flip flops.

Level trigg clock

Clk	J	K	Q	Q+
0	x	x	x	Q
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Negative edge trig clock

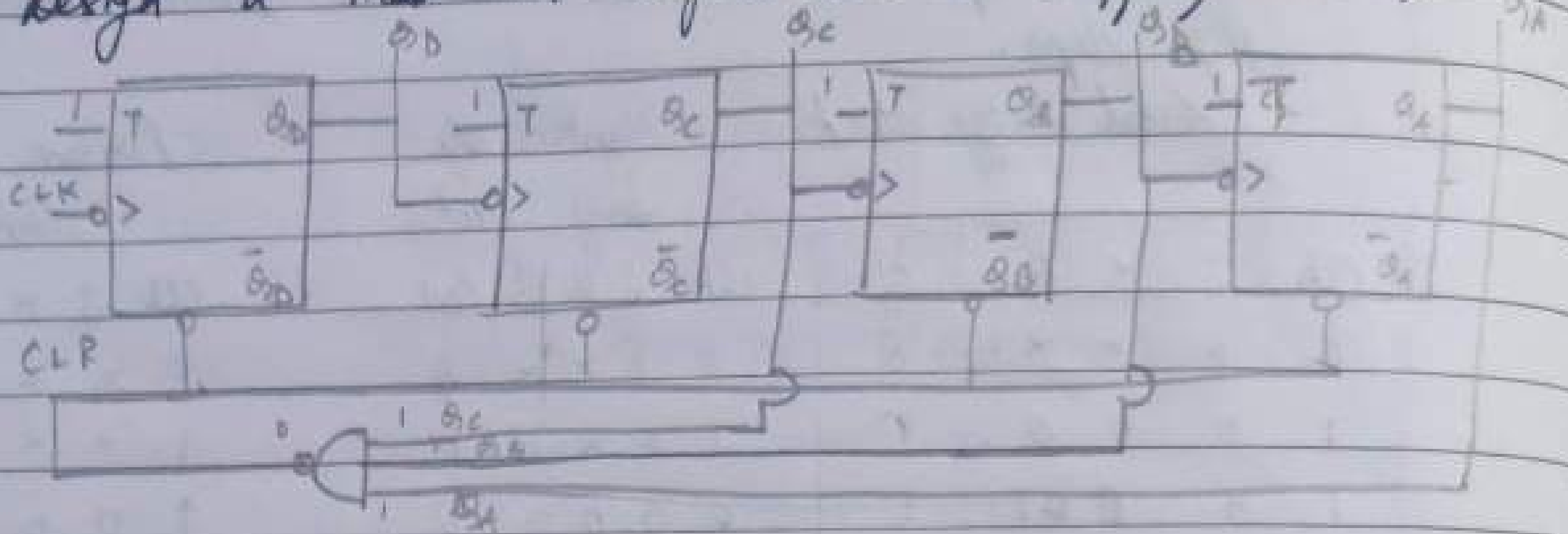
Clk	J	K	Q	Q+
0	x	x	x	Q
1	x	x	x	Q
↓	0	0	0	0
↓	0	0	1	1
↓	0	1	0	0
↓	0	1	1	0
↓	1	0	0	1
↓	1	0	1	1
↓	1	1	0	1
↓	1	1	1	0

Positive edge trig clock

Clk	J	K	Q	Q+
0	x	x	x	Q
1	x	x	x	Q
↑	0	0	0	0
↑	0	0	1	1
↑	0	1	0	0
↑	0	1	1	0
↑	1	0	0	1
↑	1	0	1	1
↑	1	1	0	1
↑	1	1	1	0

PAGE No.
DATE 6/10/22

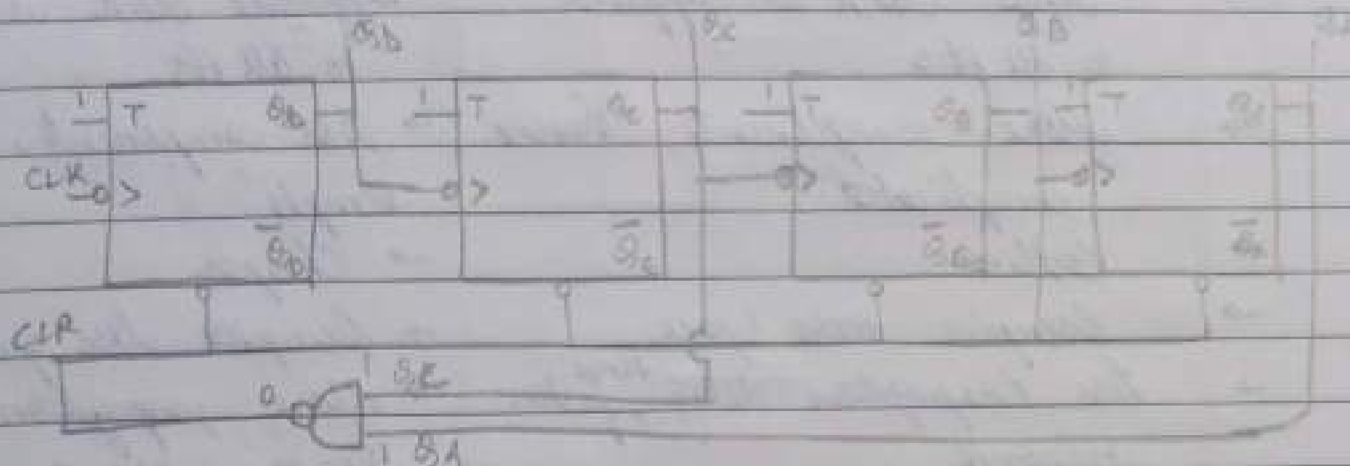
Design a Mod-14 asynchronous (ripple) counter.



Decade Counter:

Mod = 10

Counts – 0 to 9

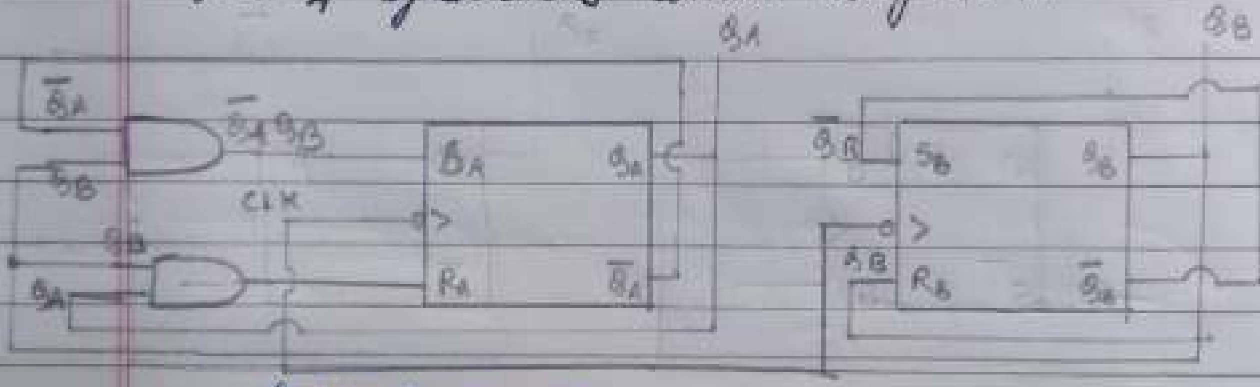


PAGE No.
DATE 12/10/22

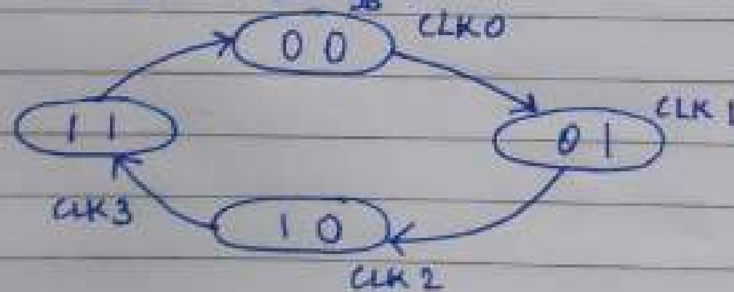
★ Synchronous Counter:

Synchronous Counter	Asynchronous Counter
→ Same clock is applied to all ffs	→ Different clock is applied to all ffs
→ More components, hence complicated.	→ Less components, hence simple
→ Costlier	→ Cheaper
→ Occupies more space	→ Occupies less space, compact
→ Less Propagation delay, hence faster response	→ More propagation delay, slower response.
→ [Draw a sample circuit]	→ [Draw a sample circuit]
→ Any ff can be used	→ Only JK or T-ff can be used.

Mod-4 synchronous counter using SR ff



State Diagram



Excitation of Table of SR

Q	Q^+	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	X	0

CLK	Present state		Next state					
	Q_A	Q_B	Q_A^+	Q_B^+	S_A	R_A	S_B	R_B
0	0	0	0	1	0	x	1	0
1	0	1	1	0	1	0	0	1
2	1	0	1	1	x	0	1	0
3	1	1	0	0	0	1	0	1

K-Map of S_A

$Q_B \backslash Q_A$	0	1
0	0	X
1	1	0

$S_A = \overline{Q}_A Q_B$

K-Map of R_A

$Q_B \backslash Q_A$	0	1
0	X	0
1	0	1

$R_A = Q_A \cdot Q_B$

K-Map of S_B

$Q_B \backslash Q_A$	0	1
0	1	1
1	0	0

$S_B = \overline{Q}_B$

K-Map of R_B

$Q_B \backslash Q_A$	0	1
0	0	0
1	1	1

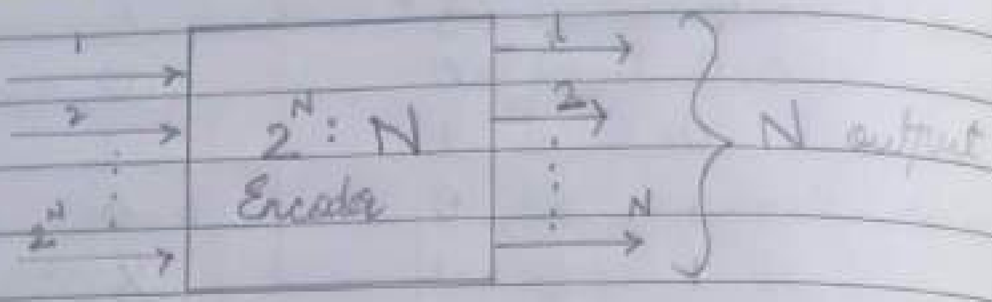
$R_B = Q_B$

NAISHA SHETTY

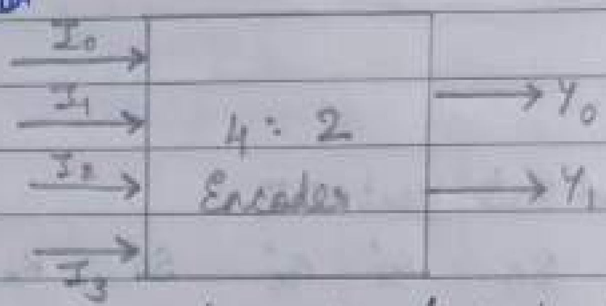
PAGE No.
DATE 19/10/22

Binary Encoder

⇒ $2^N : N$ Encoder

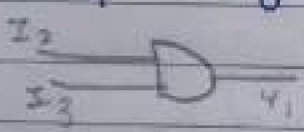


⇒ 4:2 Encoder



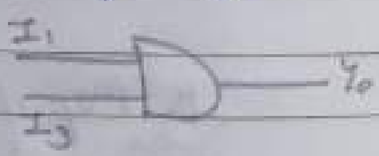
Only one input can be high at a time

I_3	I_2	I_1	I_0	Y_1	Y_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1



$Y_0 \rightarrow I_1 + I_3$

$Y_1 \rightarrow I_2 + I_3$

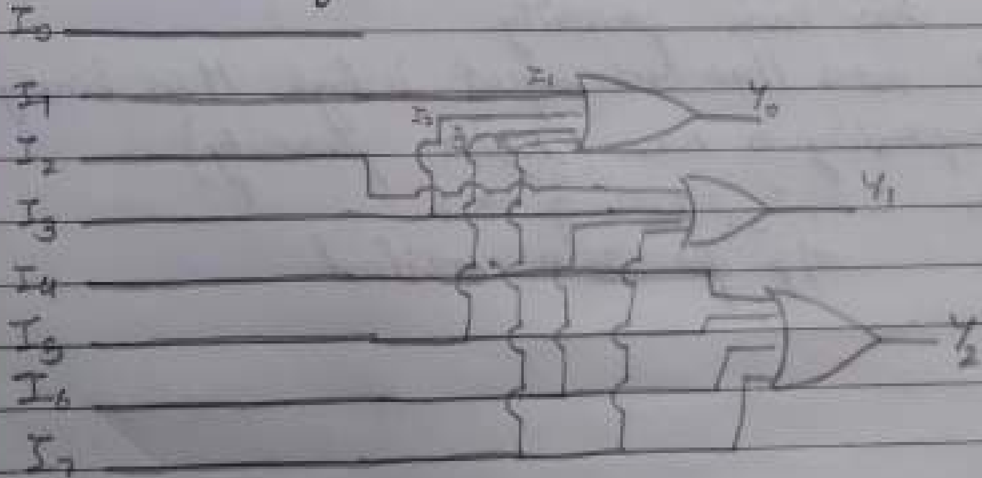


→ 8 : 3 Encoder

I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	Y_2	Y_1	Y_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

$$Y_0 = I_1 + I_3 + I_5 + I_7$$
$$Y_1 = I_2 + I_3 + I_6 + I_7$$
$$Y_2 = I_4 + I_5 + I_6 + I_7$$

Circuit Diagram



PAGE No.
DATE / /

Why do we take only one input high?

→ If we take $I_3, I_5 = 1$

$\therefore Y_0 = 0+1+1+0 = 1$

$Y_1 = 0+1+0+0 = 1$

$Y_2 = 0+1+0+0 = 1$

$\therefore$ Output → 1, 1, 1

$\therefore$ But for o/p 1, 1, 1 → i/p is $I_7 = 1$
which is not true

$\therefore$ Only one input high

→ If we take all inputs zero, we won't get any ~~o/p~~ output.

Priority Encoder

→ Every i/p is given priority
→ I_7 has max priority
→ I_0 has min. priority
→ If more than one input is high, then binary o/p. corresponding to max (higher priority)

Eg, I_1, I_3, I_5 i/p → 1
o/p for I_5 is considered.

4:2 Priority Encoder

Table:

I_3	I_2	I_1	I_0	Y_1	Y_0	Valid
0	0	0	0	x	x	0
0	0	0	1	0	0	1
0	0	1	x	0	1	1
0	1	x	x	1	0	1
1	x	x	x	1	1	1

[V: 0, Y_0, Y_1, off]

$I_1, I_0 \Rightarrow I_1$ has higher priority

$\therefore I_0 \rightarrow 0, 1 \rightarrow x$

$I_1 \rightarrow 1$

Table:

I_3	I_2	I_1	I_0	Y_1	Y_0	Valid
0	0	0	0			
0	0	0	1			
0	0	1	x			
0	1	x	x			
1	x	x	x			

PAGE No. ___ DATE / /

Truth table:

I_3	I_2	I_1	I_0	Y_1	Y_0	Valid
0	0	0	0	x	x	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	0	1	1	0	1	1
0	1	0	0	1	0	1
0	1	0	1	1	0	1
0	1	1	0	1	0	1
0	1	1	1	1	0	1
1	0	0	0	1	1	1
1	0	0	1	1	1	1
1	0	1	0	1	1	1
1	0	1	1	1	1	1
1	1	0	0	1	1	1
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1

K-Map for Valid

I_1I_0 \ I_3I_2	00	01	11	10
00	0	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$Y_1 = I_0 + I_1 + I_2 + I_3$

K-Map for Y_1

$I_1I_0 \backslash I_3I_2$	00	01	11	10
00	X	1	1	1
01	0	1	1	1
11	0	1	1	1
10	0	1	1	1

$Y_1 = I_2 + I_3$

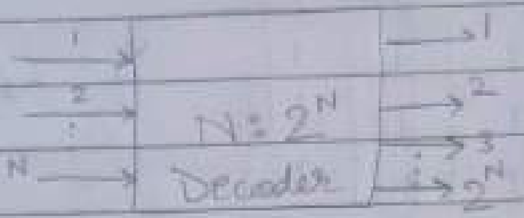
K-Map for Y_0

$I_1I_0 \backslash I_3I_2$	00	01	11	10
00	X	0	1	1
01	0	0	1	1
11	1	0	1	1
10	1	0	1	1

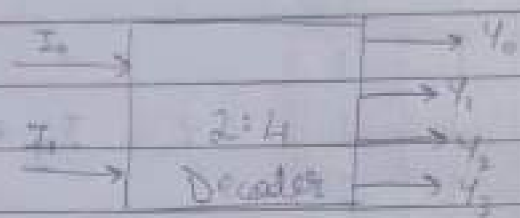
$Y_0 = I_3 + I_1\bar{I}_2$

Decoder :

→ N : 2^N Decoder.



→ 2 : 4 Decoder.



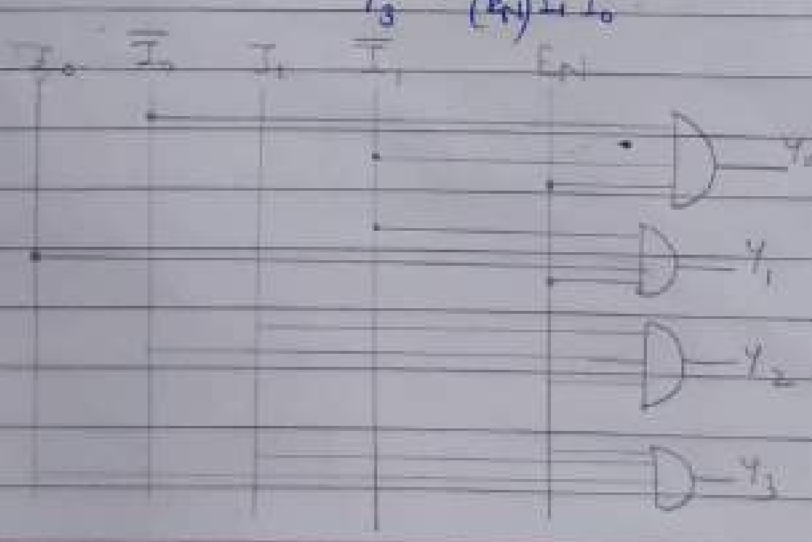
E_N	I_1	I_0	Y_3	Y_2	Y_1	Y_0
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

$Y_0 \rightarrow (E_N)\,\overline{I_0}\,\overline{I_1}$

$Y_1 \rightarrow (E_N)\,\overline{I_1}\,I_0$

$Y_2 \rightarrow (E_N)\,I_1\,\overline{I_0}$

$Y_3 \rightarrow (E_N)\,I_1\,I_0$



⇛ 3 : 8 Decoder.

E_N	I_2	I_1	I_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	x	x	x	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

$Y_7 \rightarrow (E_N)\, I_2 I_1 I_0$

$Y_6 \rightarrow (E_N)\, I_2 I_1 \overline{I_0}$

$Y_5 \rightarrow (E_N)\, I_2 \overline{I_1} I_0$

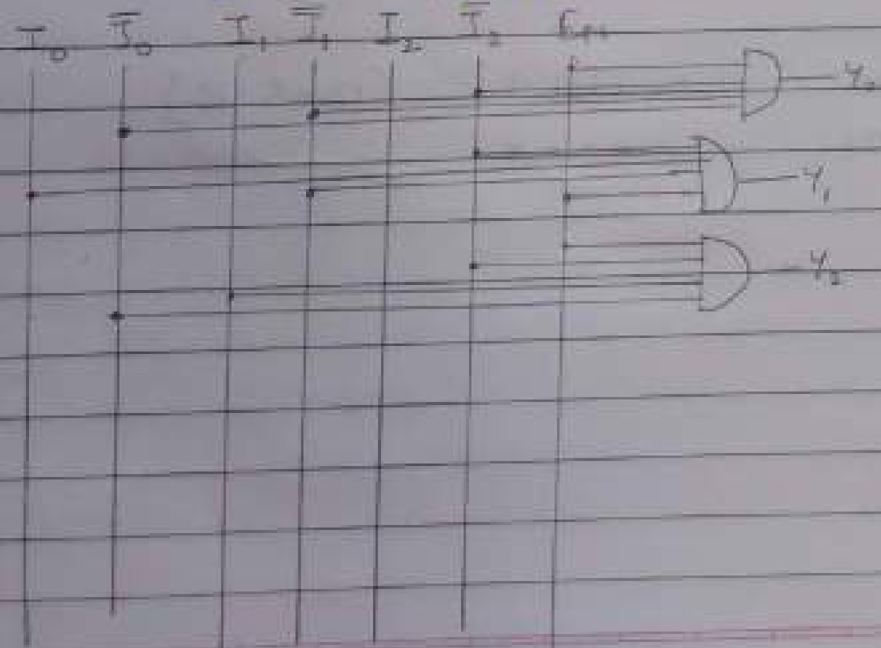
$Y_4 \rightarrow (E_N)\, I_2 \overline{I_1}\, \overline{I_0}$

$Y_3 \rightarrow (E_N)\, \overline{I_2} I_1 I_0$

$Y_2 \rightarrow (E_N)\, \overline{I_2} I_1 \overline{I_0}$

$Y_1 \rightarrow (E_N)\, \overline{I_2}\, \overline{I_1} I_0$

$Y_0 \rightarrow (E_N)\, \overline{I_2}\, \overline{I_1}\, \overline{I_0}$



Binary Coded Decimal (BCD)

Decimal	Binary	BCD
0	0000	0000
1	0001	0001
⋮	⋮	⋮
9	1001	1001
10	1010	0001 0000
11	1011	0001 0001
12	1100	$\underbrace{0001}_{1}\ \underbrace{0010}_{2}$
⋮		
245		$\underbrace{0010}_{2}\ \underbrace{0100}_{4}\ \underbrace{0101}_{5}$

BCD → 4 bit representation

BCD → 0 to 9 Unique number

Binary to Excess 3 Code (XS-3)

Decimal	Binary	XS-3	
0	0000	0011	(0 + 3 = 3)
1	0001	0100	(1 + 3 = 4)
2	0010	0101	(2 + 3 = 5)
3	0011	0110	(3 + 3 = 6)
⋮	⋮	⋮	
12	1100	1111	(12 + 3 = 15)
13	1101	xxxx	

BCD to XS-3

	BCD	XS-3
0	0000	0011
5	0101	1000
6	0110	1001
7	0111	1010
9	1001	1100
15	$\underbrace{0001}_{1}\ \underbrace{0101}_{5}$	$\underbrace{0100}_{(1+3=)4}\ \underbrace{1000}_{8\ (5+3)}$

Q Find BCD to XS-3 code of 837

	B CD	XS-3
837	1000 0011 0111	1011 0110 1010
	8 3 7	11 6 10