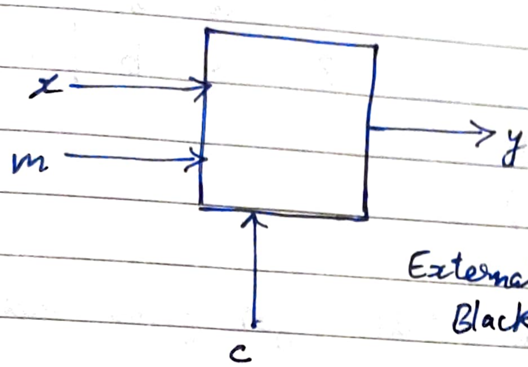
8th August, 2023

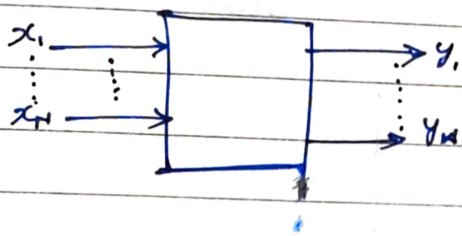
DDCO

eg: $y = mx + c$



External view
Black box

eg:

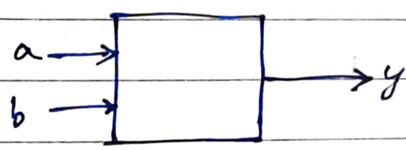


$y = f(x)$

SOP POS

Boolean Functions

$y = a \cdot b$



$y = f(a, b)$

T.T

a	b	y
0	0	0
0	1	0
1	0	0
1	1	1

a

0 1

$\bar{a}$ a literals

a	b	y
$\bar{a}$	$\bar{b}$	0
$\bar{a}$	b	0
a	$\bar{b}$	0
a	b	1

SOP → check for 1 (min terms)
POS → check for 0 (max terms)

$y = a\bar{b} + ab$

eg:

	a	b	y
0	$\bar{a}$	$\bar{b}$	0
1	$\bar{a}$	b	0
2	a	$\bar{b}$	1
3	a	b	1

$y = m_2 + m_3$

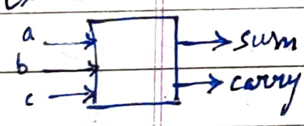
SoP $y = \Sigma m(2,3)$

PoS $y = \Pi M(0,1)$

q. Construct truth table for full adder

	2^2 A (Cin)	2^1 B (a)	2^0 C (b)	2^0 Sum	2^1 carry (C_{out})
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

External view

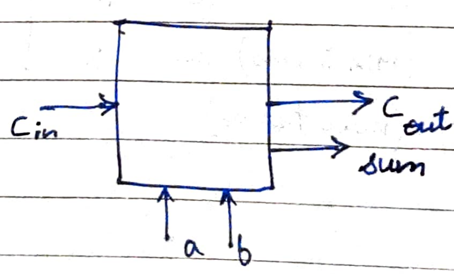


sum $= \Sigma m(1,2,4,7)$

carry $= \Sigma m(\cancel{0},3,5,6,7)$

sum $= \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c} + abc$

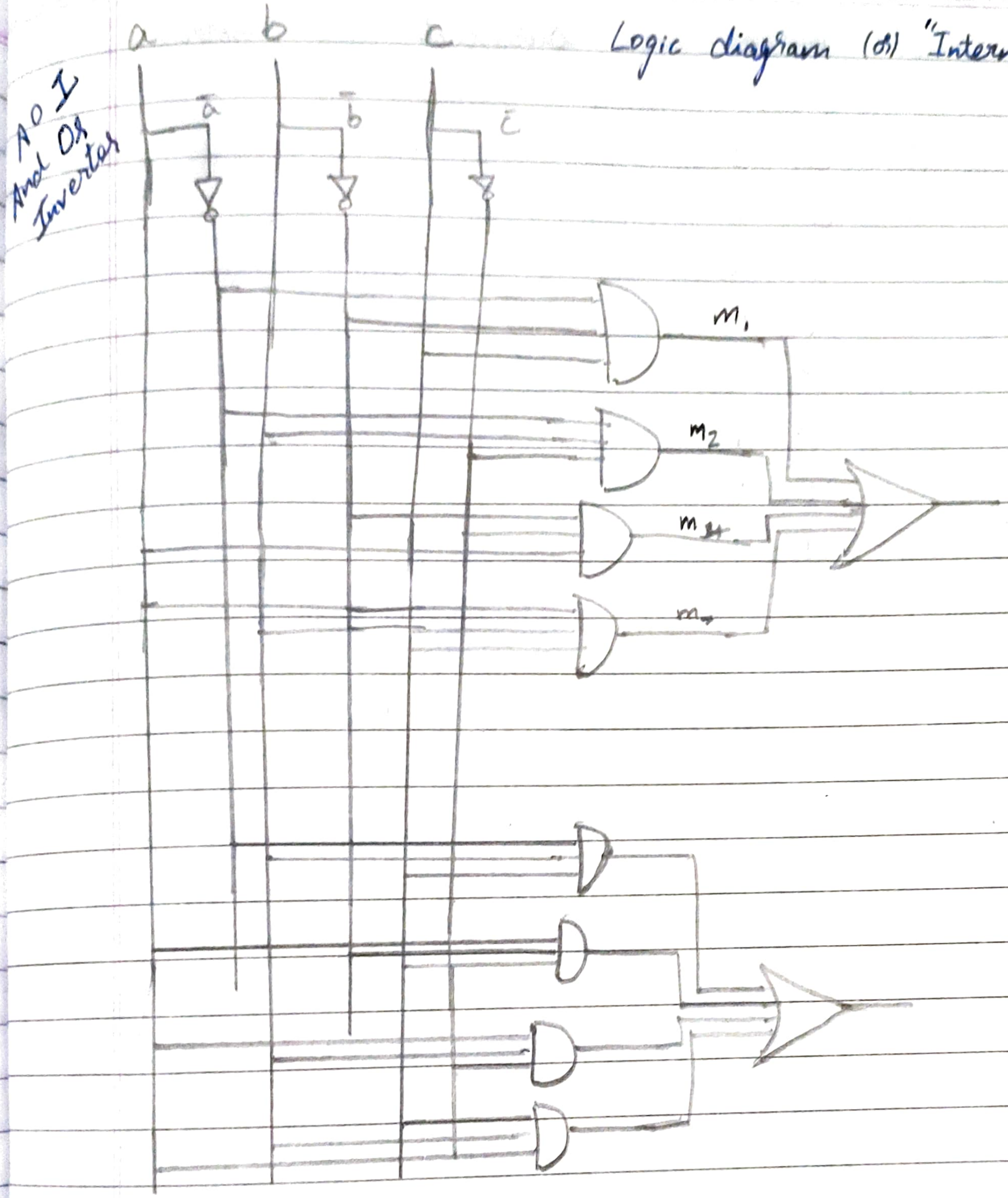
carry $= \bar{a}bc + a\bar{b}c + ab\bar{c} + abc$



$S_i = a_i \oplus b_i \oplus c_{i-1}$

PAS : Power Area Speed

Logic diagram (or) "Internal view"



q. Construct internal view for full subtracter.

10th August, 2023

UNIT-1 Number System And Logic Gates

* I/P Binary

width $\longrightarrow N$

no. of test cases $\longrightarrow 2^N$

range $\rightarrow 0$ to $2^{N}-1$

* I/P Hexadecimal : N (no. of digits)

10^N

0 to $10^N - 1$

scaling $\Rightarrow \times$ & $\div$ ~~2~~

left shift right shift

→ Binary Addition

$0111_2 + 0101_2$

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

q. Implement $y_1 = a+b$; $y_2 = a-b$

i) $a = 8$; $b = 9$
ii) $a = 5$; $b = 15$
iii) $a = 15$; $b = 5$

11th August, 2023

Signed Numbers

4 bit Binary

N	4
2^N	16
0 to $2^N - 1$	$0 \rightarrow 15$
	$0000 \rightarrow 1111$

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

2's complement

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

$$\begin{array}{r} 1111 \\ +\quad 1 \\ \hline 10000 \end{array} \qquad \begin{array}{r} 1110 \\ +\quad 1 \\ \hline 1111 \end{array} \qquad \begin{array}{r} 1101 \\ +\quad 1 \\ \hline 1110 \end{array}$$

$$\begin{array}{r} 1100 \\ +\quad 1 \\ \hline 1101 \end{array} \qquad \begin{array}{r} 0111 \\ +\quad 1 \\ \hline 1000 \end{array}$$

2's complement range

-8 to $+7$

-2^{N-1} to $2^{N-1} - 1$

Sign / Magnitude

$N = 4$

$2^N - 1$ (0 overlapping)

-7 to $+7$

$-2^{N-1} + 1$ to $2^{N-1} - 1$

$N = 5$

	width	no. of possible test cases	range (min - max)
unsigned	5	32	0 - 31
2's	5	32	-16 to +15
sign	5	31	-15 to +15

q. Compute a) $5_{10} - 3_{10}$
b) $3_{10} - 5_{10}$ using 4 bit 2's complement

Sol. $5 + (-3)$

8	4	2	1	
0	0	1	0	1
0	0	0	1	1

```
 11100
+    1
------
 11101
```

```
  0101
+ 1101
------
 10010  — 18
```

1 carry — not related for subtraction

```
  01101
+     1
-------
  01110
```

$3 - 5$

$5 \rightarrow 0101$

```
  0 0 1 1
+ 1 0 1 1
---------
  1 1 1 0   (14)
```

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

```
  0 0 0 1
+       1
---------
  0 0 1 0 — 2
```

q. $y_1 = a + b$ $\quad y_2 = a - b$

If i) $a = 35$, $b = 26$ $\quad$ ii) $a = -35$, $b = 26$

i)

	32	16	8	4	2	1
35 →	1	0	0	0	1	1
26 →	0	1	1	0	1	0

$a + b =$ ~~111001~~

111101

$a - b = 35 + (-26)$

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

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

carry

17th August, 2023

Logical minimization : K-Map

→ $y = f(a, b, c)$ [eg. of 3 variables] (8 test cases = 8 cells)

	a	b	c	min term (m)	max term (M)	Y
0	0	0	0	$\bar{a}\bar{b}\bar{c}$	$a+b+c$	0
1	0	0	1	$\bar{a}\bar{b}c$	$a+b+\bar{c}$	0
2	0	1	0	$\bar{a}b\bar{c}$	$a+\bar{b}+c$	0
3	0	1	1	$\bar{a}bc$	$a+\bar{b}+\bar{c}$	1
4	1	0	0	$a\bar{b}\bar{c}$	$\bar{a}+b+c$	1
5	1	0	1	$a\bar{b}c$	$\bar{a}+b+\bar{c}$	0
6	1	1	0	$ab\bar{c}$	$\bar{a}+\bar{b}+c$	1
7	1	1	1	abc	$\bar{a}+\bar{b}+\bar{c}$	1

$$Y = \bar{a}bc + a\bar{b}\bar{c} + ab\bar{c} + abc$$

$$Y = m_3 + m_4 + m_6 + m_7$$

$$= \sum (m_3, m_4, m_6, m_7)$$

$$= \sum_m (3, 4, 6, 7)$$

Soln. @ source

AOI :

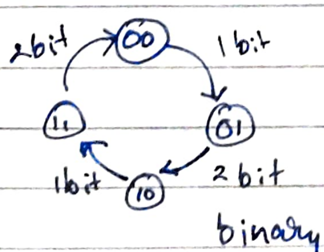
NOT : 3

AND : 4 → (3 i/p)

OR : 1 → (4 i/p)

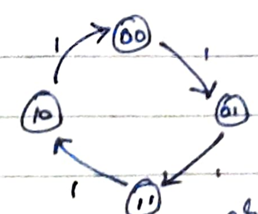
only 1 bit difference in two consecutive no. = graycode

bit flipping



binary

6



gray code

4

	Bin	gray
(00)	0	00 (0)
(01)	1	01 (1)
(10)	2	11 (3)
(11)	3	10 (2)

a \ bc	00	01	11	10
0	$\bar a \bar b \bar c$ 0	$\bar a \bar b c$ 1	$\bar a b c$ 3	$\bar a b \bar c$ 2
1	$a \bar b \bar c$ 4	$a \bar b c$ 5	abc 7	$ab\bar c$ 6

$\bar a bc + abc = bc(a+\bar a) = bc$

a \ bc				
0	0	1	1 3	2
1	1 4	5	1 7	1 6

$abc + ab\bar c = ab$

1, 6 can be grouped $a\bar b \bar c + ab\bar c = a\bar c$

grouping ① 1-cell

$\bar a bc$, $a\bar b\bar c$, abc, $ab\bar c$: Implicants (I)

② 2-cell bc, ab, $a\bar c$: prime Implicants (PI)

$y = bc + ab + a\bar c$

③ bc, $a\bar c$ Essential Prime Implicants (EPI)

$y = bc + a\bar c$ no overlaping

q:

	a	b	c	y
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

$$y = \underset{1}{\bar{a}\bar{b}c} + \underset{3}{\bar{a}bc} + \underset{4}{a\bar{b}\bar{c}} + \underset{5}{a\bar{b}c} + \underset{6}{ab\bar{c}} + \underset{7}{abc}$$

$$y = \Sigma_m(1, 3, 4, 5, 6, 7)$$

a \ bc		c	c	
$\bar{a}$	0	1 (1)	1 (3)	2
a	1 (4)	1 (5)	1 (7)	1 (6)

grouping: 1-cell : $\bar{a}\bar{b}c$, $\bar{a}bc$, $a\bar{b}\bar{c}$, $a\bar{b}c$, $ab\bar{c}$, abc

2 cell : $\bar{a}c$, $a\bar{b}$, ac, ab, $a\bar{c}$, $\bar{b}c$, bc,

4 cell : a, c

EPI 2cell : $\bar{a}c$, ac, $a\bar{c}$

EPI 4cell : a, c

$y = a + c$

1 3 5 7

$\bar{a}\bar{b}c$
$+ \bar{a}bc$
$+ a\bar{b}c$
$+ abc$

q. Construct k-map for sum and carry of full adder.

	a	b	c	sum	carry
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

sum $y = \Sigma_m(1, 2, 4, 7)$ carry $y = \Sigma_m(3, 5, 6, 7)$

0	1 1	3	1 2
1 4	5	1 7	6

Sum grouping 1-cell : $y = \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c} + abc$

Carry grouping 1-cell : $y = \bar{a}bc + a\bar{b}c + ab\bar{c} + abc$

0	1	1 3	2
4	1 5	1 7	1 6

EPI is same as PI

15 $\bar{a}bc + abc$

57 $a\bar{b}c + abc$

~~76~~ 26 $\bar{a}b\bar{c} + ab\bar{c}$

q.

0	0	1	0	1
1	0	1	1	1

grouping 1 cell : ~~$y = \bar{a}\bar{b}c + \bar{a}b\bar{c} +$~~

$y = \bar{a}\bar{b}c + \bar{a}bc + a\bar{b}c + abc + ab\bar{c}$

2 cell

$y = \bar{b}c + ac + ab + b\bar{c}$ PI

EPI $y = \bar{b}c + ac + b\bar{c}$

~~$\bar{y}\bar{z} + \bar{y}z + yz + y\bar{z}$~~

~~$\bar{y}(\bar{z} + z) + y(z + \bar{z})$~~

01	– $\bar{a}\bar{b}$
0	$\bar{a}\bar{b}\bar{c}$
1	$\bar{a}\bar{b}c$
4	$a\bar{b}\bar{c}$
5	$a\bar{b}c$

01
45
67

q.

$x \backslash yz$	00	01	11	10
	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$y\bar{z}$
0 $\bar{x}$	1	1	0	1
1 x	1	1	1	1

$\Sigma(0, 1, 2, 4, 5, 6, 7)$

grouping

1-cell

$y = \underset{0}{\bar{a}\bar{b}\bar{c}} + \underset{1}{\bar{a}\bar{b}c} + \underset{2}{\bar{a}b\bar{c}} + \underset{4}{a\bar{b}\bar{c}} + \underset{5}{a\bar{b}c} + \underset{6}{ab\bar{c}} + \underset{7}{abc}$

2-cell

$y = \bar{a}\bar{b} + a\bar{b} + ac + ab + b\bar{c}$

4 cell

$y = \bar{b}, a$

EPI 2cell $y = \bar{a}\bar{b} + a\bar{b} + ab + b\bar{c}$

18th August, 2023

→ K-Maps [4 Variables]

eg: $f(a, b, c) = \Sigma m(1, 2, 5, 6, 7)$

a \ bc	00	01	11	10
0	0_0	1_1	0_3	1_2
1	0_4	1_5	1_7	1_6

$\bar{b}c$ ac $b\bar{c}$

ab

$PI = \bar{b}c, ac, b\bar{c}, ab$

$EPI = \bar{b}c, ac, b\bar{c}$

$RPI = ab$ (shared EPI's)

(Redundant PI)

minimal exp: $y = \bar{b}c + ac + b\bar{c}$

eg: $\Sigma m(0, 1, 5, 6, 7)$

a \ bc				
0	1_0	1_1	$_3$	$_2$
1	$_4$	1_5	1_7	1_6

PI : $\underbrace{\bar{a}\bar{b}, ac, ab}_{EPI}, \underbrace{b\bar{c}}_{RPI}$

Q. $\Sigma m(0, 2, 3, 4, 6, 7)$

list out PI & EPI for 2 cells grouping & 4 cells grouping separately

2 cell grouping

a \ bc	$\bar{b}\bar{c}$ 00	$\bar{b}c$ 01	bc 11	$b\bar{c}$ 10
$\bar{a}$ 0	1 (0)	(1)	1 (3)	1 (2)
a 1	1 (4)	(5)	1 (7)	1 (6)

PI: $\bar{b}\bar{c}$, bc, $b\bar{c}$, $\bar{a}b$, ab, $\bar{a}\bar{c}$, $a\bar{c}$

EPI: $\bar{b}\bar{c}$, bc, $b\bar{c}$

RPI: remaining 4 terms

4 cell grouping

1		1	1
1		1	1

$y = b + \bar{c}$

q4. $\Sigma m(1,3,5,7)$

q5. $\Sigma m(0,4,5)$

q6. $\Sigma m(0,2,4,6,7)$

q7. $\Pi M(2,3,5,7)$

Find minimal soln.

Ans: 4.

a \ bc	00	01	11 (c)	10
0	0 (0)	1 (1)	1 (3)	0 (2)
1	0 (4)	1 (5)	1 (7)	0 (6)

$y = c$

5.

a \ bc	00	01		
0	1 (0)	(1)	(3)	(2)
1	1 (4)	1 (5)	(7)	(6)

$y = \bar{b}\bar{c} + a\bar{b}$

6.

	00	01	11	10
0	1_0	$_1$	$_3$	1_2
1	1_4	$_5$	1_7	1_6

$y = \bar{c} + ab$

7. $\Pi M(2,3,5,7)$

	00	01	11	10
0	1_0	1_1	0_3	0_2
1	1_4	0_5	0_7	1_6

$y = \bar{a}\bar{b} + \bar{b}\bar{c} + a\bar{c}$

→ $f(a,b,c,d) = \Sigma m(5,6,7,8,9,10,11,13,14,15)$

ab \ cd	00	01	11	10
00	0_0	0_1	0_3	0_2
01	0_4	1_5	1_7	1_6
11	0_{12}	1_{13}	1_{15}	1_{14}
10	1_8	1_9	1_{11}	1_{10}

bd

bc

$a\bar{b}$

grouping all 16 will give 1

$y = bd + bc + a\bar{b}$

q: $\Sigma m(3,4,5,7,9,13,14,15)$

ab \ cd		00 $\bar{c}\bar{d}$	01 $\bar{c}d$	11 cd	10 $c\bar{d}$
$\bar{a}\bar{b}$	00	$_0$	$_1$	1_3	$_2$
$\bar{a}b$	01	1_4	1_5	$_7$	$_6$
ab	11	$_{12}$	1_{13}	1_{15}	1_{14}
$a\bar{b}$	10	$_8$	1_9	$_{11}$	$_{10}$

$\bar{a}cd$

bd

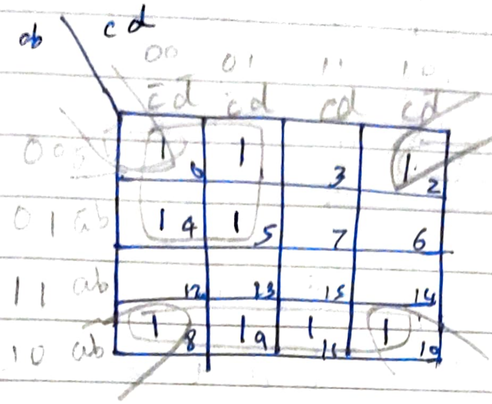
$\bar{a}b\bar{c}$

abc

$a\bar{c}d$

$y = \underbrace{bd}_{\text{redundant}} + \bar{a}cd + abc + a\bar{c}d + \bar{a}b\bar{c}$

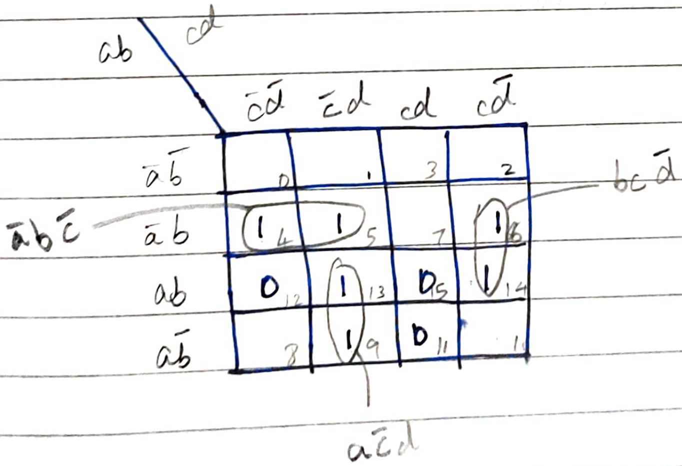
eg: $\Sigma m(0, 1, 2, 4, 5, 8, 9, 10, 11)$



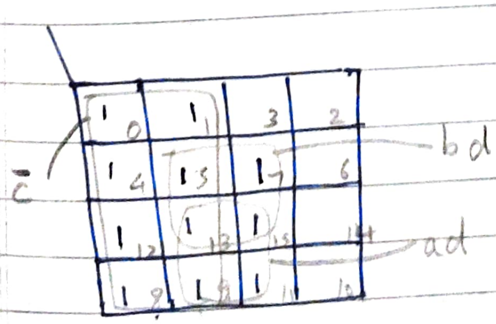
$y = \bar a \bar c + a\bar b + \bar b \bar d$

q.3. $\Sigma m(4, 5, 6, 9, 13, 14)$
q4. $\Sigma m(0, 1, 4, 5, 7, 8, 9, 11, 12, 13, 15)$
q5. $\Sigma m(1, 5, 6, 7, 8, 9, 13, 15)$

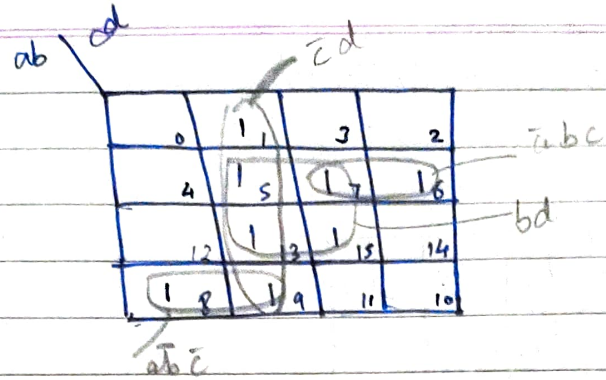
Ans 3.



4.



5.



q 6. $f(a,b,c,d) = \bar{a}\bar{b}d + bcd + a\bar{b}d + b\bar{c}\bar{d}$

$m(1,3,4,7,9,11,12,15)$

h/w q7. $f = a + \bar{c}$

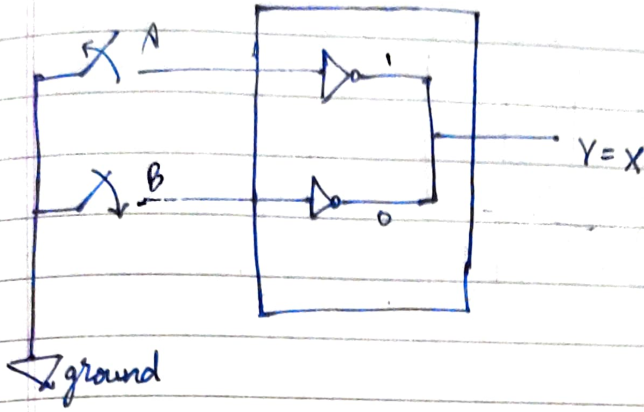
q8. $f = \bar{a} + bd + a\bar{d}$

q9. $f = \bar{a} + \bar{b}d + ac\bar{d}$

21st August, 2023

K-Maps (with don't care)

$(\underbrace{0, 1}_{\text{valid}}, \underbrace{X, Z}_{\text{invalid}})$



Z: floating value, anything b/w 0 & 1
X: 0/1.

eg: $f = \sum m(1, \cancel{2}, 3, 5, 7, 9, 11) + \underbrace{d(2, 4, 6)}_{\text{don't care terms}}$

ab \ cd	00	01	11	10
00	$_0$	1_1	1_3	X_2
01	X_4	1_5	1_7	X_6
11	$_{12}$	$_{13}$	$_{15}$	$_{14}$
10	$_8$	1_9	1_{11}	$_{10}$

$\bar{a}c$ $\bar{a}d$ $\bar{a}b$ $\bar{b}d$

$y = \bar{a}d + \bar{b}d$

q: $f = \sum m(1, 2, 3, 4, 5, 6, 7) + d(11, 12, 13, 14)$

ab \ cd	00	01	11	10
00		1	1	1
01	1	1	1	1
11	X	X		X
10			X	

$\bar{a}d$ $\bar{a}c$ $\bar{a}b$

$y = \bar{a}b + \bar{a}d + \bar{a}c$

q: $\Sigma m(9, 10, 11) + d(12, 15)$

ab \ cd	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	X 12	13	X 15	14
10	8	1 9	1 11	1 10

$a\bar{b}d$ $\qquad$ $a\bar{b}c$

Karnaugh maps

0 $\qquad$ 0000 $\qquad$ 0

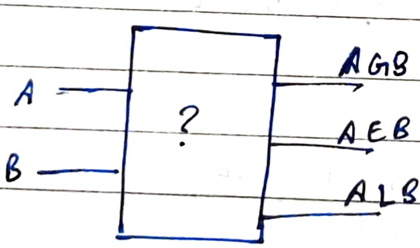
: $\qquad\qquad\qquad$: $\qquad$ Valid

9

10 $\qquad$ don't cares

15 $\qquad$ 1111 $\qquad$ 15

q: Design 1-bit comparator and generate 3 o/p values indicating A > B, A < B, A = B



A	B	A>B	A=B	A<B
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

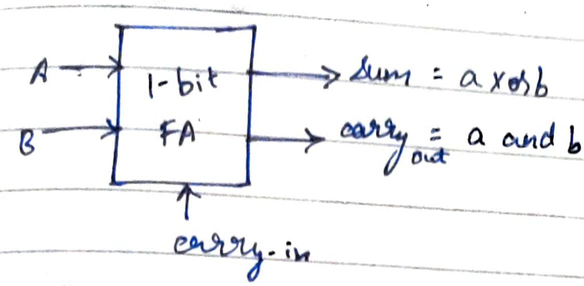
2-bit

	$A_1 A_0$	$B_1 B_0$	$A>B$	$A=B$	$A<B$
0	0 0	0 0	0	1	0
		0 1	0	0	1
		1 0	0	0	1
		1 1	1	1	1
1	0 1	0 0	1	0	0
		0 1	0	1	0
		1 0	0	0	1
		1 1	0	0	1
2	1 0	0 0	1	0	0
		0 1	1	0	0
		1 0	0	1	0
		1 1	0	0	1
3	1 1	0 0	1	0	0
		0 1	1	0	0
		1 0	1	0	0
		1 1	0	1	0

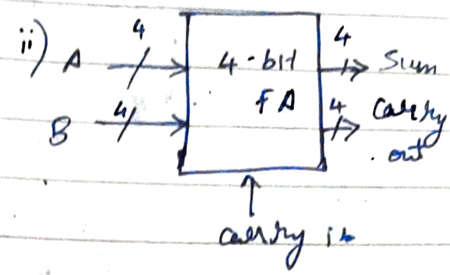
22nd August, 2023

Adder, Subtractor - 1

i)



ii)



eg: $y = A + B$

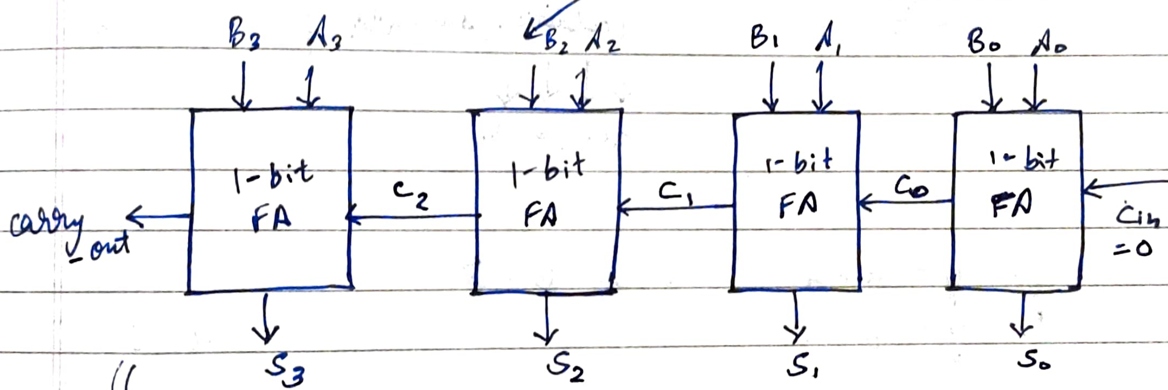
A → 4-bit, B → 4-bit

$A_3 A_2 A_1 A_0$ → $A[3:0]$

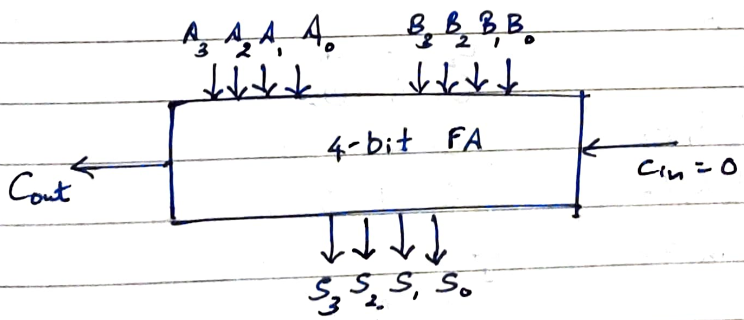
$B_3 B_2 B_1 B_0$ → $B[3:0]$

$A_3\, A_2\, A_1\, A_0$

$\underline{B_3\, B_2\, B_1\, B_0}$



⇓



structural modelling

XOR

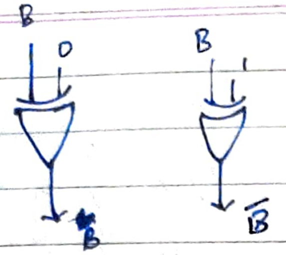
P	Q	Y
0	0	0
0	1	1
1	0	1
1	1	0

if $P = 0$; $Q = 0/1$

$y = Q$ ①

$P = 1$;

$y = \overline{Q}$ ②

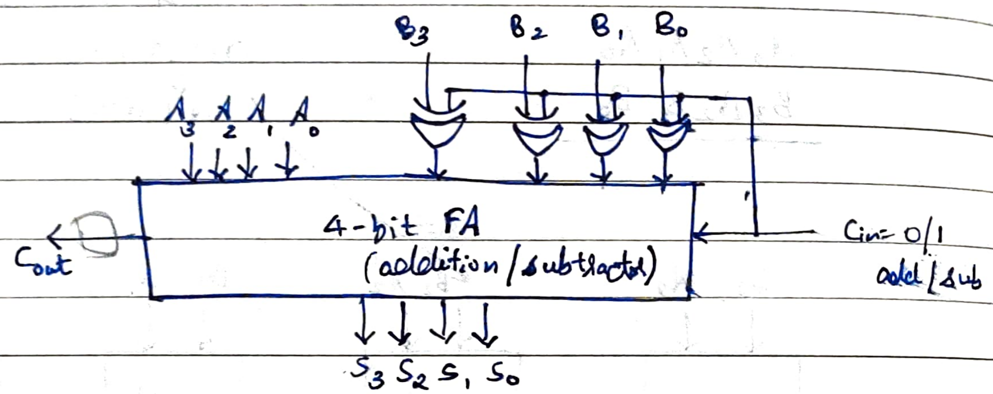


$$\begin{array}{r} A_3 \; A_2 \; A_1 \; A_0 \\ B_3 \; B_2 \; B_1 \; B_0 \\ + \qquad\qquad 0 \\ \hline \end{array}$$

$y = A + B$

$$\begin{array}{r} A_3 \; A_2 \; A_1 \; A_0 \\ \bar{B}_3 \; \bar{B}_2 \; \bar{B}_1 \; \bar{B}_0 \\ + \qquad\qquad +1 \\ \hline \end{array}$$ } 2's complement of B (-B)

$y = A - B$



Q: $y = B + A$ $\quad A = 5; \quad B = 3$

$y = B - A$

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

$y = B + A$

$C_2 \; C_1 \; C_0$

1 1 1

i/p 1 $B_3 B_2 B_1 B_0$ — 0 0 1 1

i/p 2 $A_3 A_2 A_1 A_0$ — 0 1 0 1

i/p 3 $C_{in} = 0$ — + 0

0 1 0 0 0

C_{out} $\quad S_3 \; S_2 \; S_1 \; S_0$

$y = B + (-A)$

1 1

0 0 1 1

1 0 1 0 } → (1's comp of A using XOR)

\+ 1 → 2's comp of A

~~0~~ 1 1 1 0

Since it's 0 we should neglect

(-2) by taking 2's comp. of ans.

$-A \quad -B$

$-A + (-B)$

2. $A = -5$

$B = 3$

$3 \rightarrow 0\ 0\ 1\ 1$

$5 \rightarrow 0\ 1\ 0\ 1$

$y = B + A$

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

$$\begin{array}{r} 1\,0\,1\,0 \\ +\quad\ \ 1 \\ \hline 1\,0\,1\,1 \end{array} \rightarrow -5$$

$y = B - A$

$y = B + (-A)$

$= 3 + (-(-5))$

$$\begin{array}{r} 0\,0\,1\,1 \\ +\ 0\,1\,0\,1 \\ \hline \cancel{\underbrace{0}\ 1\,0\,0\,0} \end{array}$$

$A = -5 \Rightarrow 1\,0\,1\,1$

$-A = 5 \Rightarrow$

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

$$\begin{array}{rr} 3 & 0\,0\,1\,1 \\ 5 & 0\,1\,0\,0 \rightarrow 1s \\ + & +1 \\ \hline & 0\ \underbrace{1\,0\,0\,0} \end{array}$$

sign bit

$(14)_d \quad 1110$

$$\left.\begin{array}{r} 0001 \\ +1 \end{array}\right\}$$

0010

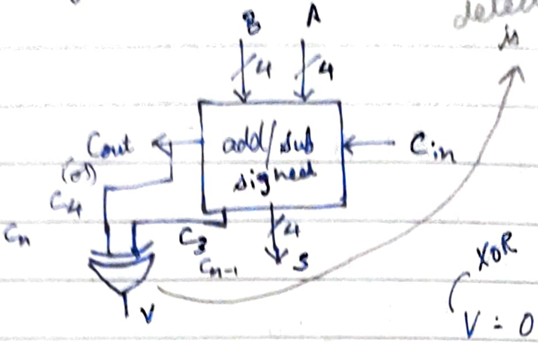
$(+2)$

24th August, 2023 — Signed 4-bit added with range -8 to +7

Adder / sub overflow - 2

additional hardware detects if O/P result is valid not

Signed (or) 2's complement (-8 to +7)



XOR

V : Overflow flag

$0 \wedge 1$

System

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

C_n	C_{n-1}	V
0	0	NO
0	1	YES
1	0	YES
1	1	NO

i) (+2) and (-4)

$$\begin{array}{rcccc} & c_4 & c_3 & c_2 & c_1 \\ & 0 & 0 & 0 & 0 \\ (-2) & 0 & 0 & 1 & 0 \\ (-4) & 1 & 1 & 0 & 0 \\ + & & & & \\ \hline & 1 & 1 & 1 & 0 \end{array}$$

$c_3 = 0$

$c_4 = 0$

$V = 0$

ii) (−3) and (−2)

1 1 0 0

(−3) 1 1 0 1

(−2) 1 1 1 0

\+

~~X~~ 1 0 1 1

$C_3 = 1$

$C_4 = 1$

$V = 0$

iii) (+7) and (+6)

0 1 1 0

(+7) 0 1 1 1

(+6) 0 1 1 0

\+

✓ 0 1 1 0 1

$C_3 = 1$

$C_4 = 0$

$V = 1$

valid in binary but we have 2's complement (system) so invalid

iv) (−7) and (−6)

1 0 0 0

(−7) 1 0 0 1

(−6) 1 0 1 0

\+

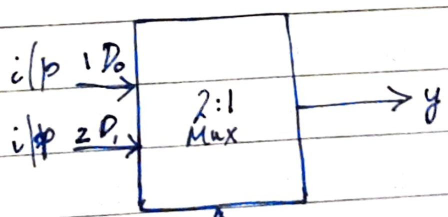
0 0 1 1

$C_3 = 0$

$C_4 = 1$

$V = 1$

Multiplexer / Mux



if Sel = 0: y = i/p 1

if Sel = 1: y = i/p 2

T.T

sel	D_0	D_1	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

if $S = 0$
$y = D_0$

if $S = 1$
$y = D_1$

sel \ $D_0 D_1$	00	01	11	10
0	0_0	0_1	1_3	1_2
1	0_4	1_5	1_7	0_6

$\overline{sel} \cdot D_0$

$sel \cdot D_1$

$$y = sel \cdot D_1 + \overline{sel}\, D_0 \quad \text{(minimal exp)}$$

reduced TT
(50%)

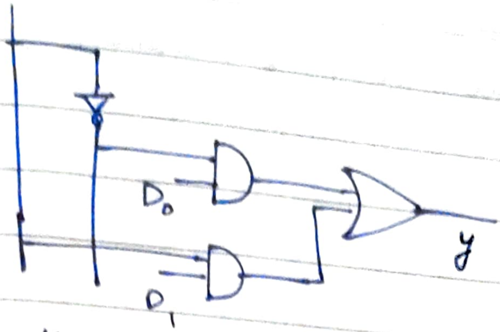
sel	D_0	D_1	y
0	0	0	0
	1	1	1
1	0	0	0
	1	1	1

doesn't matter

50% MEV
map entered Variables

sel	y
0	D_0
1	D_1

$y = \overline{sel} \cdot D_0$
$y = sel \cdot D_1$



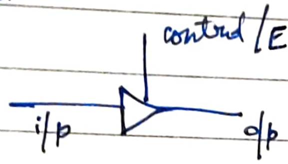
Mux circuit using 2-level implementation

→ 

y = a Buffer

* Tri state Buffer

E : enable

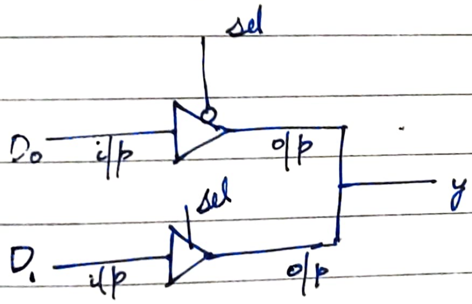


E	o/p
1	I/p
0	Z

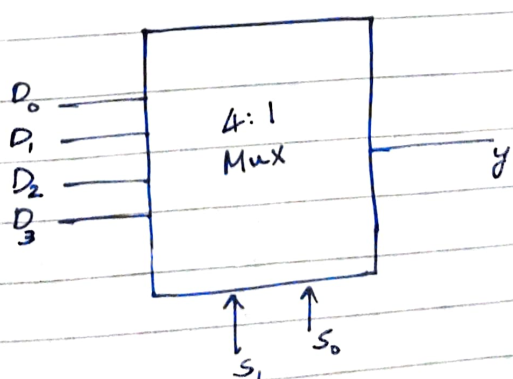
Z : high impedence / floating value

E	I/p	o/p
1	0	0
1	1	1
0	0	Z
0	1	Z

Mux circuit using Tri state buffer



→ 4 : 1



25th August, 2023

Logic implementation using Mux

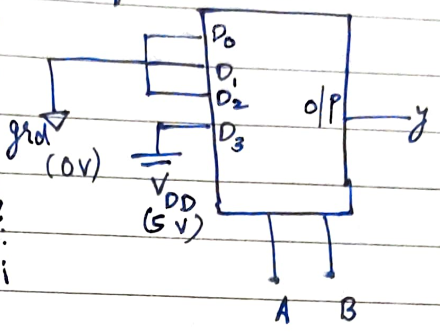
eg1. $y = A \cdot B$

	A	B	Y
1	0	0	0
	0	1	0
2	1	0	0
	1	1	1

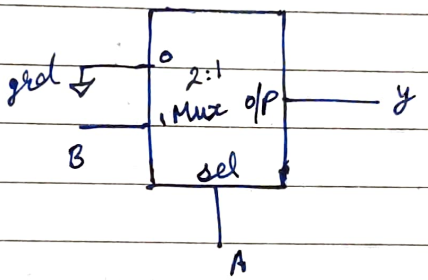
1: if $A=0$, $Y=0$

2: if $A=1$, $Y=B$

i) using 4:1 Mux



ii)



Q2. implement a 2 i/p xor gate using

i) 4:1 Mux

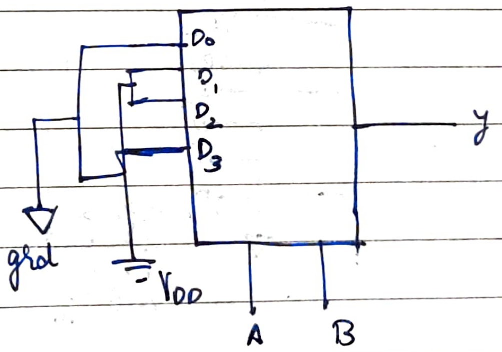
ii) 2:1 Mux

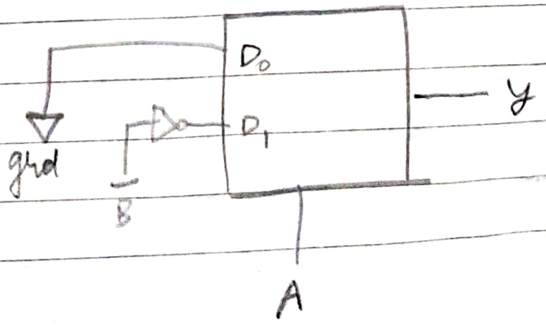
$y = A \oplus B$

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

$A=0$, $Y=B$

$A=1$, $Y=\bar{B}$

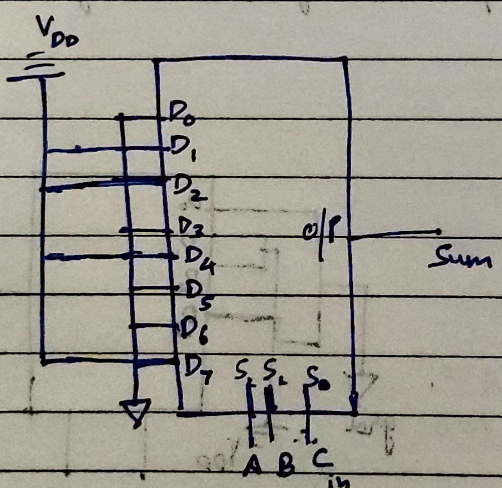




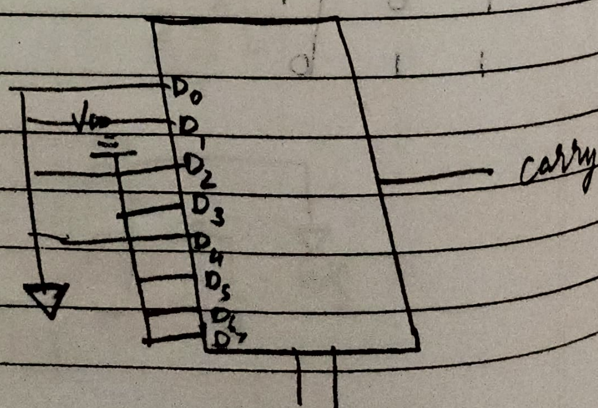
eg 3. Implement 1-bit full-adder sum and carry expression using 8:1 and 4:1 Mux.

	A	B	C_{in}	sum	carry	
0	0	0	0	0	0	B
1	0	0	1	1	0	
2	0	1	0	1	0	C_{in}
3	0	1	1	0	1	
4	1	0	0	1	0	C_{in}
5	1	0	1	0	1	
6	1	1	0	0	1	B
7	1	1	1	1	1	

8:1 sum $f = \Sigma m(1, 2, 4, 7)$

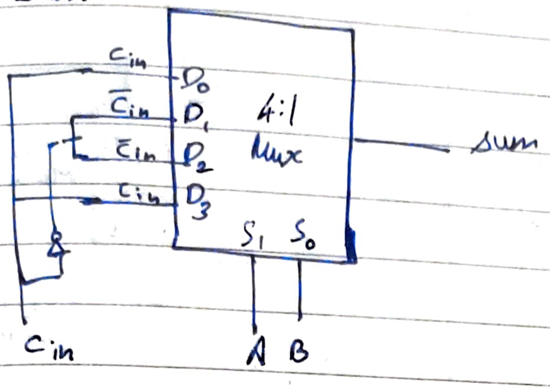


carry

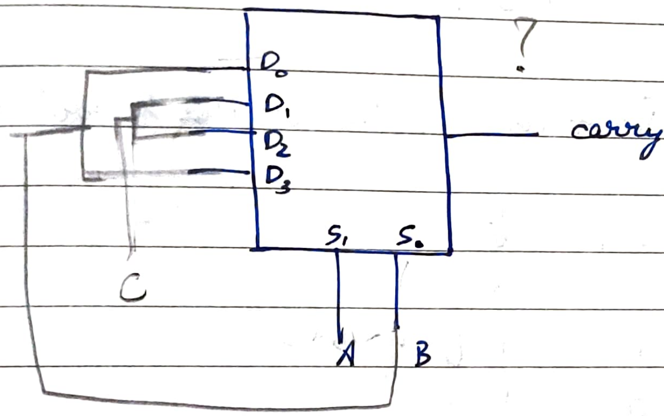


4:1

Sum:

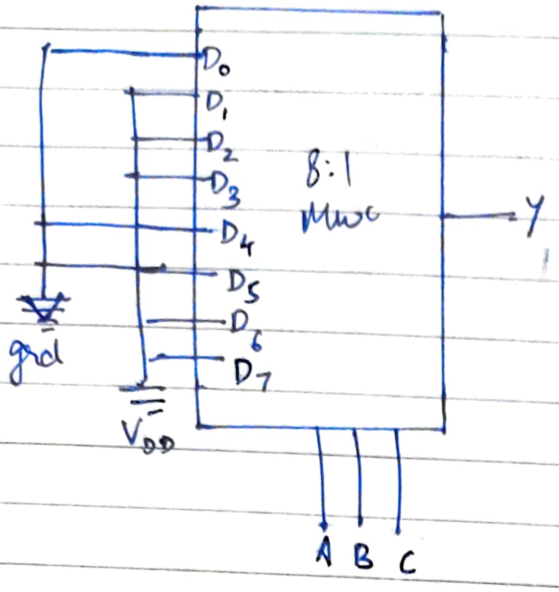


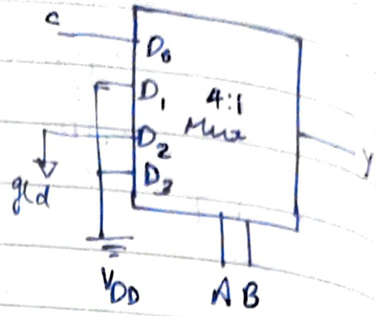
carry:



q4. Implement the given SOP $y = \bar{a}b + \bar{a}\bar{b}c + ab$ using 8:1 & 4:1 Mux

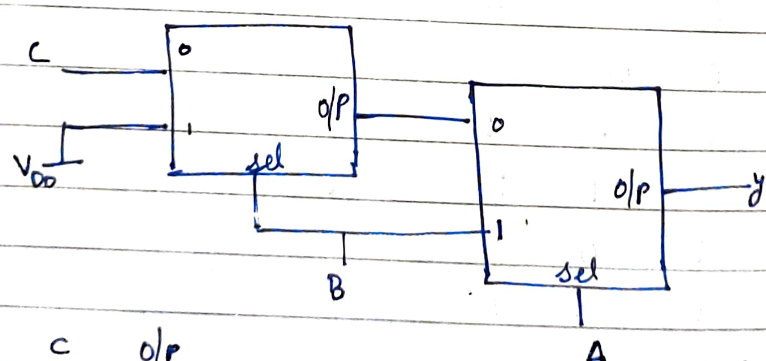
a	b	c	y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1





Q5. $y = \bar{a}b + \bar{a}\bar{b}c + ab$ (using 2:1 Mux & additional hardware)

a	b	c_{in}	y	
0	0	0	0	
0	0	1	1	$y = B + C$
0	1	0	1	
0	1	1	1	
1	0	0	0	
1	0	1	0	$y = B$
1	1	0	1	
1	1	1	1	

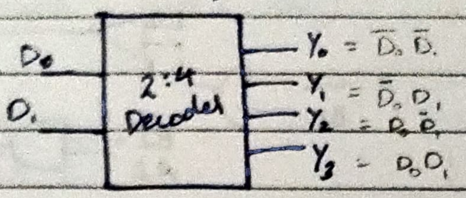


B	C	o/p
0	0	0
0	1	1
1	0	1
1	1	1

29th August, 2023

Decoders

maping i/p to decimal equivalent



decoders

2:4 → 3:8 → … → $N : 2^N$ (i/p : o/p)

Mux

2:1 → 4:1 → … → $2^n : 1$ (sel n; i/p : o/p)

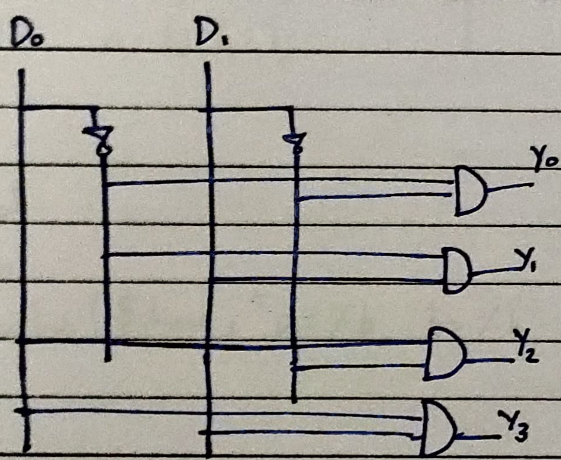
	D_0	D_1	Y_0	Y_1	Y_2	Y_3
0	0	0	1	0	0	0
1	0	1	0	1	0	0
2	1	0	0	0	1	0
3	1	1	0	0	0	1

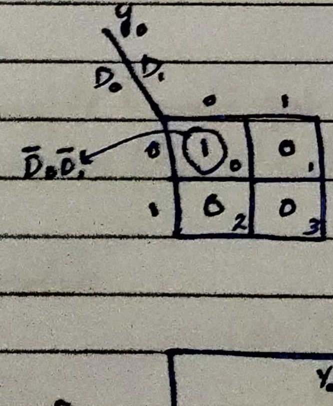
0 0 0 0

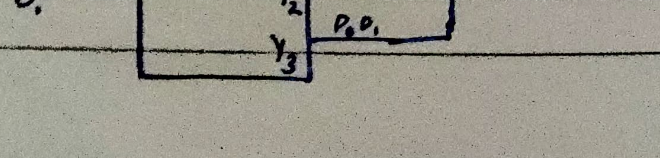
⋮

1 1 1 1

2-level imp







$Y_3 \Rightarrow$ 2 i/p AND

$Y_1 + Y_2 \Rightarrow$ 2 i/p XOR

$\bar{D_0} D_1 + D_0 \bar{D_1}$

$Y_A = D_0 \oplus D_1$

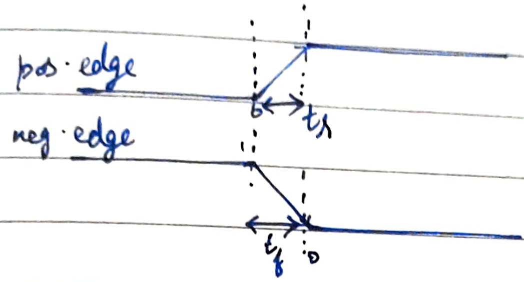
$\bar{Y_3} \Rightarrow$ 2 i/p NAND

$Y_0 + Y_3 =$ 2 i/p XNOR

$Y_B = \bar{D_0}\bar{D_1} + D_0 D_1$

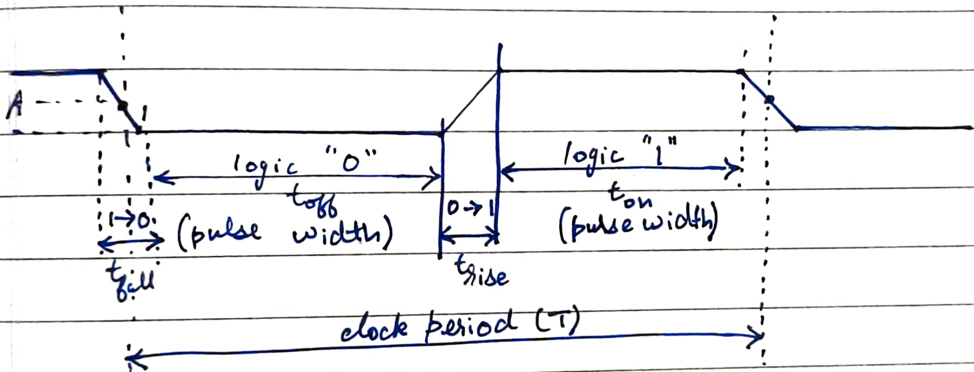
$= \overline{D_0 \oplus D_1}$

* changing value from 0 to 1 : Positive edge (rise time) (t_r)



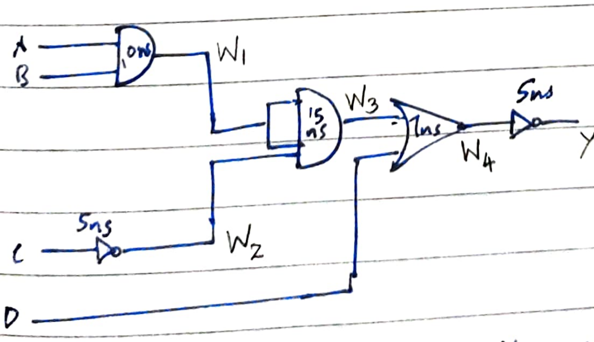
0 } static value
1

0 → 1 } Dynamic values
1 → 0



start point / time $\quad f = 1/T \quad$ stop time

q:

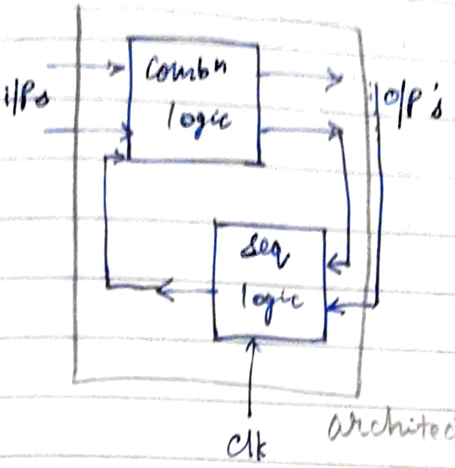


37ns
(10 + 15 + 7 + 5)

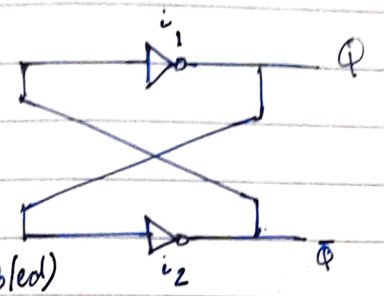
For the given circuit find out propagation delay.

31st August, 2023

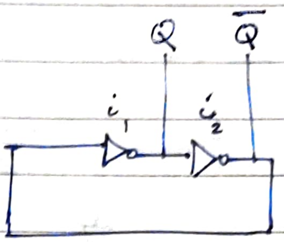
Sequential logic / Block



architecture



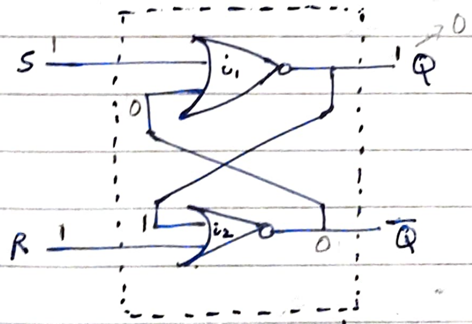
(cross coupled)



(connected back to back)

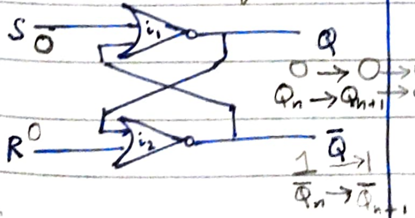
Bistable element
(or) 1-bit storage element

→ Latch

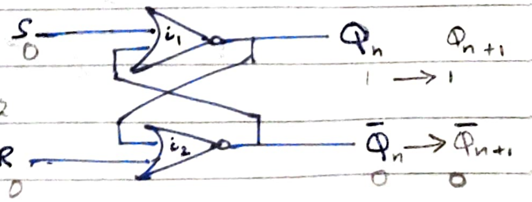


1. SR - latch
(using nor)

assuming $Q = 0$



assuming $Q = 1$



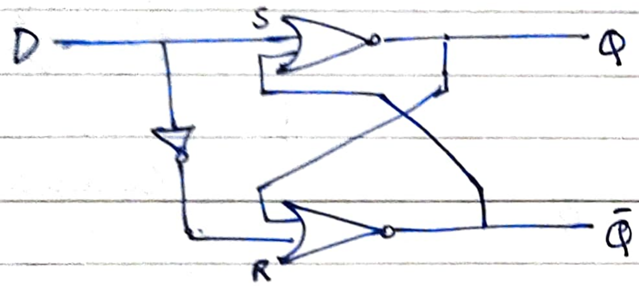
we need to tap two state to understand behaviour
Q_n : current state
Q_{n+1} : next state

ip1	ip2	nor
0	0	1
0	1	0
1	0	0
1	1	0

i/ps

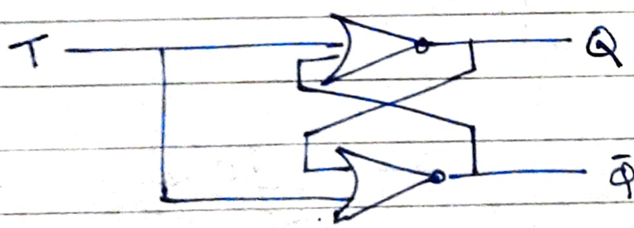
	S	R	Q_n	Q_{n+1}	
①	0	0	0	0	No change
			1	1	
②	0	1	0	1	Set
			1	1	
③	1	0	0	0	Reset
			1	0	
④	1	1	0	X	Invalid
			1	X	

2. D - Latch

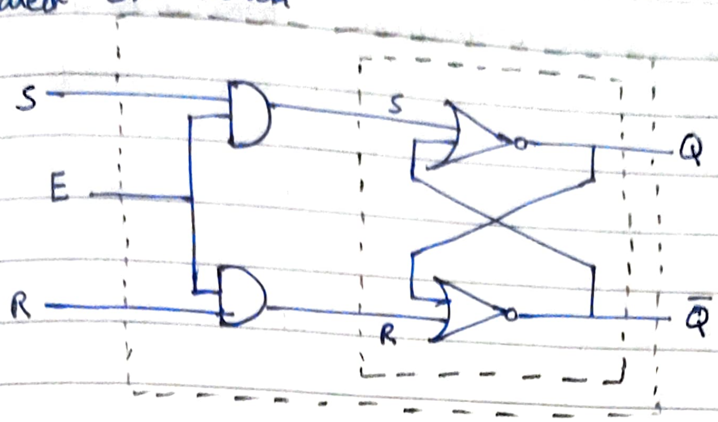


D	Q_n	Q_{n+1}	
0	0	1	set
	1	1	
1	0	0	reset
	1	0	

3. T - Latch



4. Gated SR Latch



E	S	R	Q_n	Q_{n+1}
0				No change
1				

* Latches are level triggering
* Flip Flops are edge triggering

E	S	R	Q_n	Q_{n+1}
0	X	X	X	N.C

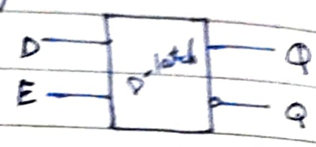
1st September, 2023

dataflow and structural programming of 2:1 Mux

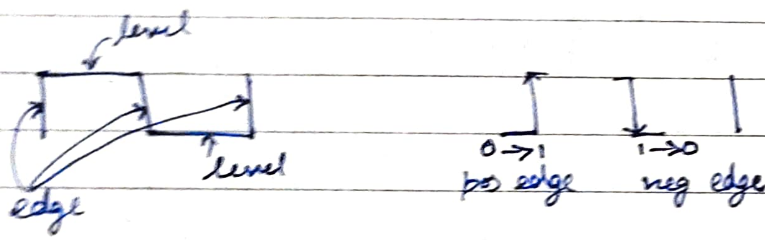
10 recys fixed:
6

4th September, 2023
(Refer Morris Mano txtbk)

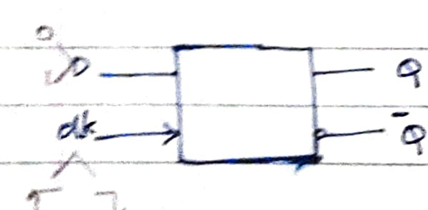
Gated D-latch (level triggered)
(levels of clock)



E/Clk	D	Q_n	Q_{n+1}		
0	X	0/1	0/1	NC	Transparent
1	0	0/1	1	set	Not Transparent
1	1	0/1	0	reset	



→ D - F/F
(+ve edge)



Q_{n+1}

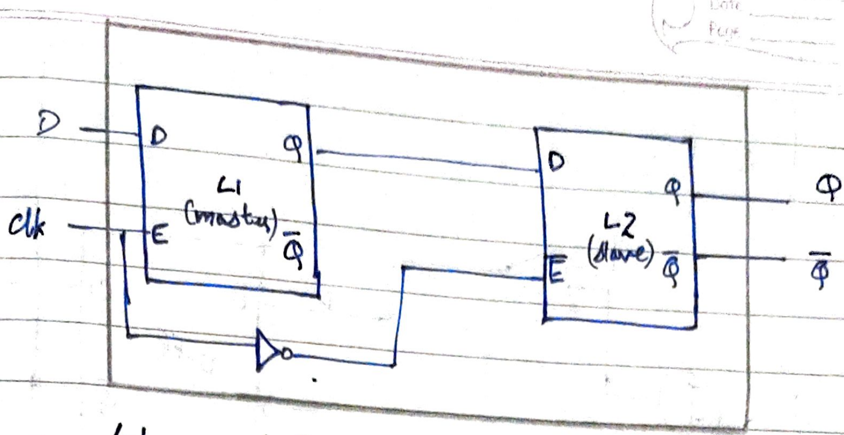
E \ DQ_n	00	01	11	10
0	Q_n (0)	Q_n (1)	Q_n (3)	Q_n (2)
1	1 (4)	1 (5)	0 (7)	0 (6)

$\bar{E}Q_n$

$E\bar{D}$

$$\boxed{Q_{n+1} = E\bar{D} + \bar{E}Q_n}$$
characteristic of D-latch

* No change = transparent ⇒ i/p = o/p



E	L1	L2
0	Transparent	Not Transparent
1	Not-Transparent	Transparent
0	T	NT

Clk	D	Q_{slave}
↑	0/1	0/1
↓	0/1	S/R

order : (Verilog implementation)

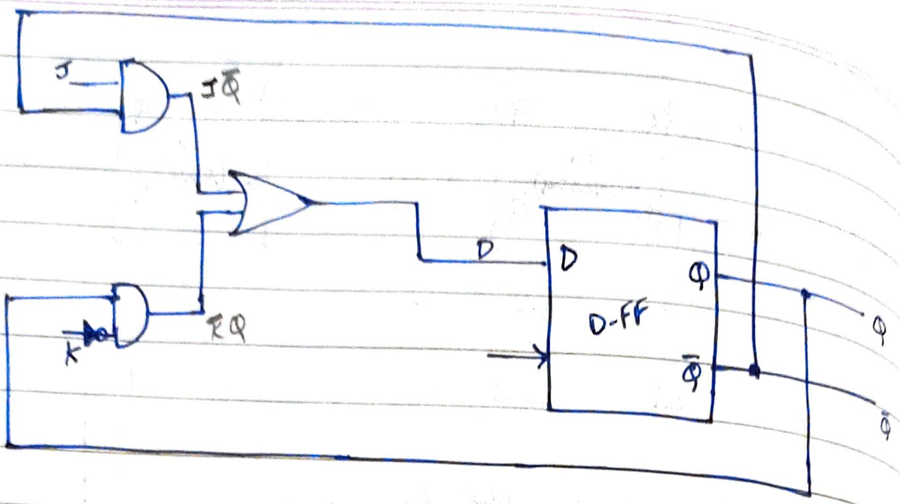
→ Bistable → SR (nor) → D latch → D-FF → JK-FF → T-FF

→ J-K FF

J	K	Q_n	Q_{n+1}
0	0	0/1	0/1
0	1	0/1	1
1	0	0/1	0
1	1	0/1	1/0

→ pos edge

→ neg edge



$$D = J\bar{Q} + \bar{K}Q$$

① If $J = 0$; $K = 0$

$D = 0 + Q = Q$

② If $J = 0$; $K = 1$

$D = 0 + 0 = 0$ (set) at Q

③ If $J = 1$; $K = 0$

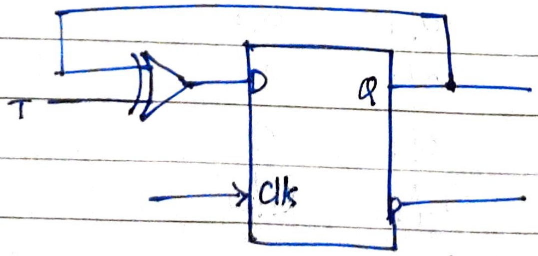
$D = \bar{Q} + Q = 1$ (reset) at Q

④ If $J = 1$; $K = 1$

$D = \bar{Q} + 0 = \bar{Q}$ (Toggle)

* D-FF if clk is +ve it will hold memory for 1 clk period.

→ T-FF



$D = T \oplus Q$

$= TQ' + T'Q$

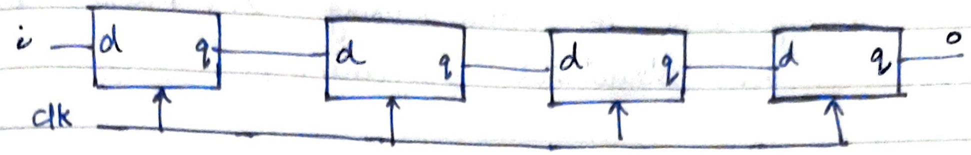
(from D.FF)

(can also be built from JK F by shorting J & K)

shifting data towards right ⇒ ÷
shifting data towards left ⇒ ×

SISO
SIPO

4-bit registers in sequence



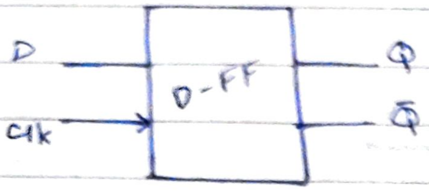
clk	i	Q_3	Q_2	Q_1	Q_0	
		0	0	0	0	
① ↑	1	1	0	0	0	$(8)_d$
② ↑	0	0	1	0	0	$(4)_d$
③ ↑	0	0	0	1	0	$(2)_d$
④ ↑	0	0	0	0	1	$(1)_d$

		8	4	2	1	
↑	1	1	0	0	0	(8)
↑	1	1	1			(12)
↑	1	1	1	1		(14)
↑	1	1	1	1	1	(15)

5th September, 2023

Flip-Flop with "Enable"

case study : D - FF



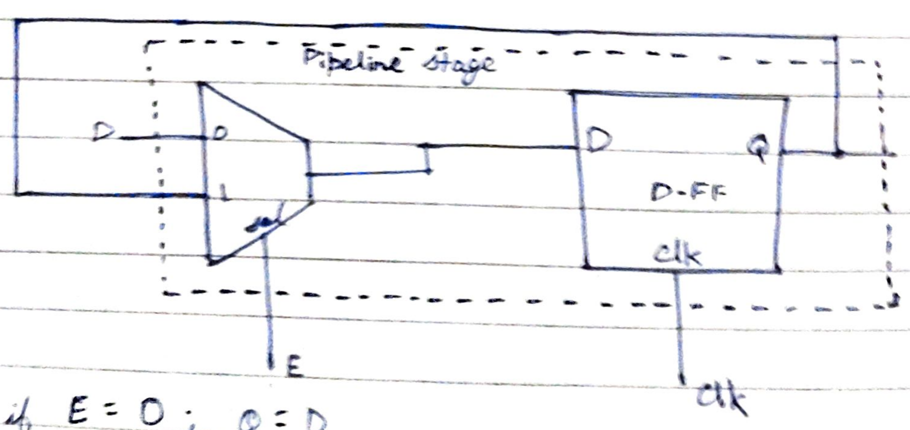
T.T	clk	D	Q
(o/p follow)	↑	0	0
	↑	1	1

T.T.	E	clk	D	Q
	0	↑	0	0
	0	↑	1	1
	1	↑	0	Q_{prev}
	1	↑	1	Q_{prev}

* Universal shift register

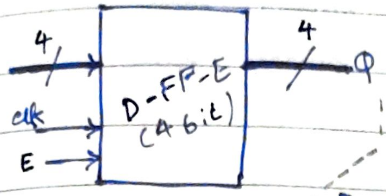
1. Right (÷) → logical : cascading MSB with 0s
   → Arithmetic : 1s (sign)
2. Left (×)
3. Rotate → left
   → right
4. Nop

→ D - FF with Enable

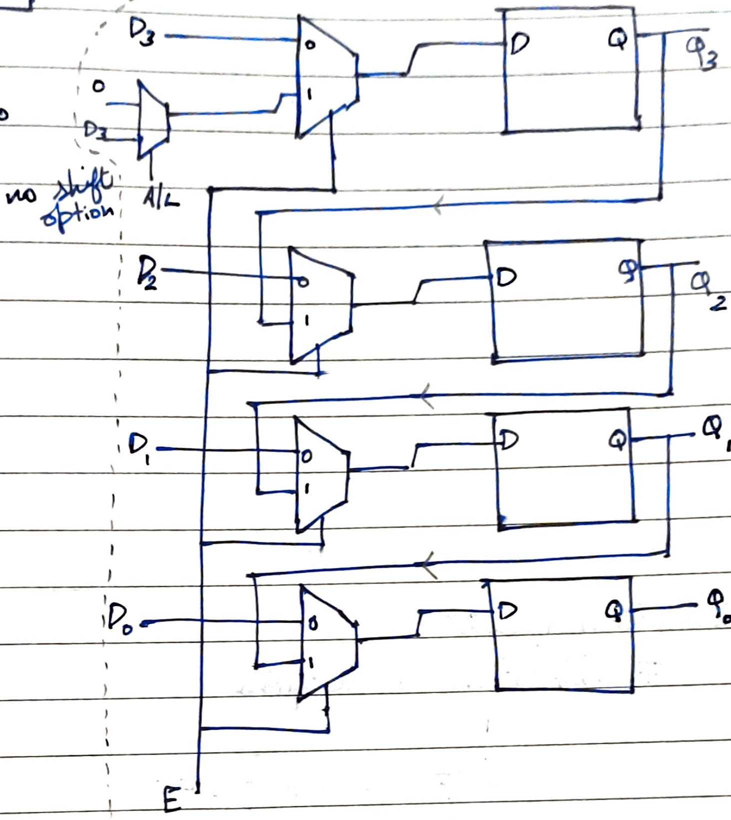


if E = 0 ; Q = D
E = 1 ; Q = prev. value

→ 4-bit D-FF with E



4-bit shifting reg
(shift right, NOP)



$D_3\ D_2\ D_1\ D_0$

$Q_3\ Q_2\ Q_1\ Q_0$

$\downarrow\ \downarrow\ \downarrow\ \downarrow$

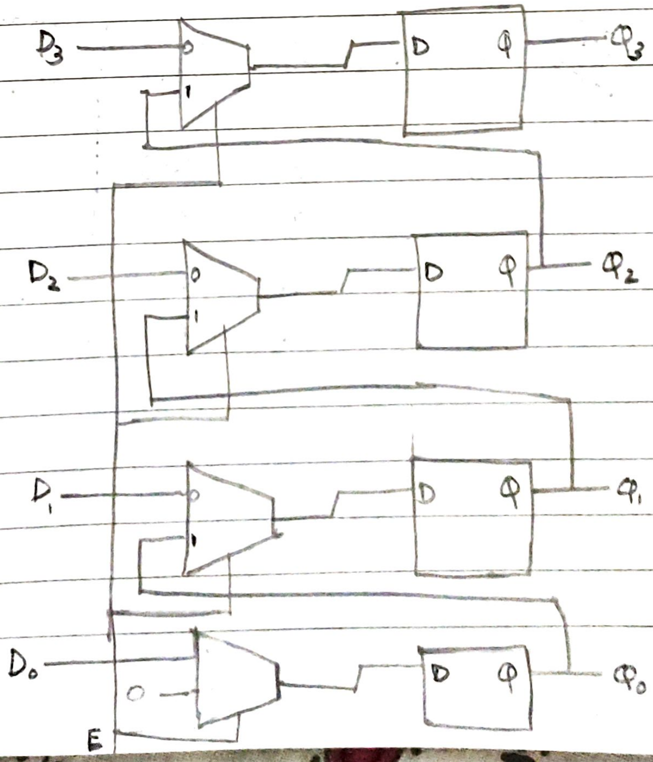
$D_3\ D_2\ D_1\ D_0$

$Q_3\ Q_2\ Q_1\ Q_0$

$0 \mid D_3\ D_3\ D_2\ D_1$

LSR | ASR

→ shift left, NoP



$Q_3\ Q_2\ Q_1\ Q_0$

$\downarrow\ \downarrow\ \downarrow\ \downarrow$

$D_3\ D_2\ D_1\ D_0$

$Q_3\ Q_2\ Q_1\ Q_0$

$D_2\ D_1\ D_0\ 0$

no arithmetic

q. Implement $y = ax^3$ using less no. of hardware resources.

a : 4 bit binary value

x : 2 bit binary value

7th September, 2023

Flip-Flop conversions

SR - FF ⟶ D - FF

(available) (Target)

T.T SR-FF

S	R	P.S Q_n	N.S Q_{n+1}
0	0	X	N.C
0	1	X	0
1	0	X	1
1	1	X	?

excitation Table of S-R-FF

Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

mandatory i/p

Step 1. T.T of target device (D-FF)

D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

Step 2. Excitation Table of source (SR-FF)

Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Step 3. Conversion Table X

D	Q_n	Q_{n+1}	S	R
0	0	0	0	X
0	1	0	0	1
1	0	1	1	0
1	1	1	X	0

Step 4.

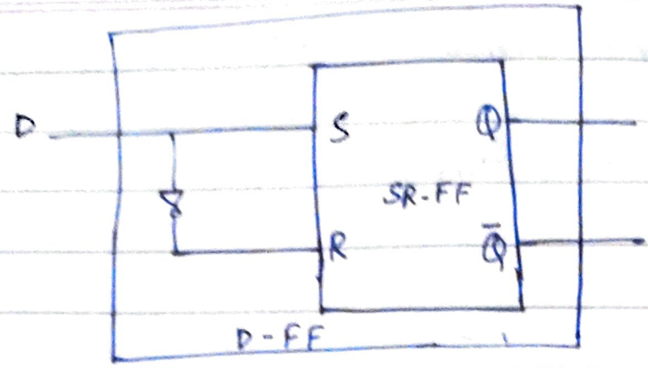
S: D \ Q_n	0	1
0	0	0
1	1	X

$S = D$

R: D \ Q_n	0	1
0	X	1
1	0	0

$R = \bar{D}$

Step 5



Q2 Convert SR-FF to T-FF

Step 1 TT of T-FF

T	Q_n	Q_{n+1}	
0	0	0	} N.C
0	1	1	
1	0	1	} $Q_{n+1} = \bar{Q}_n$
1	1	0	

Step 2 excitation table of source (SR-FF)

Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Step 3 conversion table

T	Q_n	Q_{n+1}	S	R
0	0	0	0	X
0	1	1	X	0
1	0	1	1	0
1	1	0	0	1

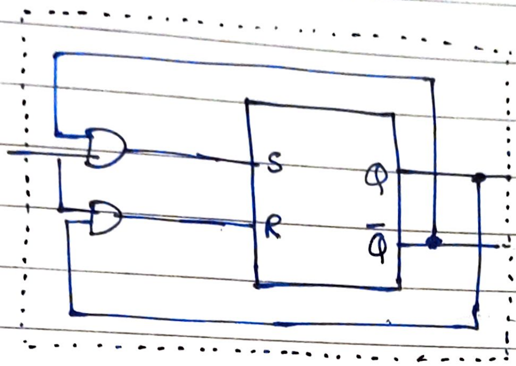
Step 4.

$T \backslash Q_n$	0	1
0	0	X
1	1	0

$S = T\overline{Q}_n$

$T \backslash Q_n$	0	1
0	X	0
1	0	1

$R = TQ_n$



Q3. Convert D-FF to JK-FF

Step 1. TT of JK-FF

J	K	Q_n	Q_{n+1}
0	0	0/1	0/1
0	1	0/1	1
1	0	0/1	0
1	1	0/1	1/0

Step 2. excitation table of Source (D-FF)

Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

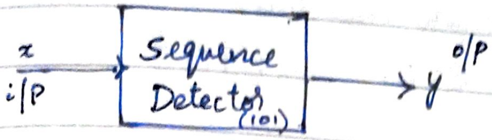
Step 3 conversion table

D	Q_n	Q_{n+1}	J	K

11th September, 2023

FSM: Finite State Machine

- FSM (overlap)



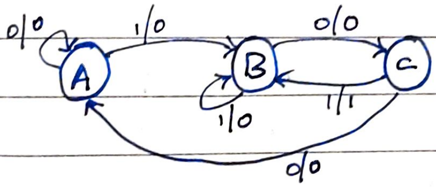
x   1 0 1 0 1 1 1 1 0 1

y (overlapped)   1 1   1

y (non - overlapped)   1   1

① State Diagrams

ip / op (or) x/y



A is initial state

@B < 10, 11

@C 10 < 100 C to A, 101 C to B

② State Table

PS	ip (x)	NS	op (y)
A	0	A	0
	1	B	0
B	0	C	0
	1	B	0
C	0	A	0
	1	B	1

$+ \Rightarrow$ next value

③ Conversion Table

(with Binary Encoding)

A: 00, B: 01, C: 10

PS	ip	NS	op	F-F ip eqn.	
$Q_1 Q_0$	x	$Q_1^+ Q_0^+$	y	D_1	D_0
00	0	00	0	0	0
00	1	01	0	0	1
01	0	10	0	1	0
01	1	01	0	0	1
10	0	00	0	0	0
10	1	01	1	0	1

TT of D-FF

clk	D	$Q_n (Q)$	$Q_{n+1} (Q^+)$
↑	0	0	0
		1	0
↑	1	0	1
		1	1

excitation table of D-FF

Q_0	Q_0^+	D_0
0	0	0
0	1	1
1	0	0
1	1	1

$D = Q^+$

④ K-map

ip variable : Q_1, Q_0, x

op variable : y, D_1, D_0

D_1

$Q_1 \backslash Q_0x$	00	01	11	10
0	0_0	0_1	0_3	1_2
1	0_4	0_5	X_7	X_6

$D_1 = Q_0\bar{x}$

D_0

$Q_1 \backslash Q_0x$	00	01	11	10
0	0	1	1	0
1	0	1	X	X

$D_0 = x$

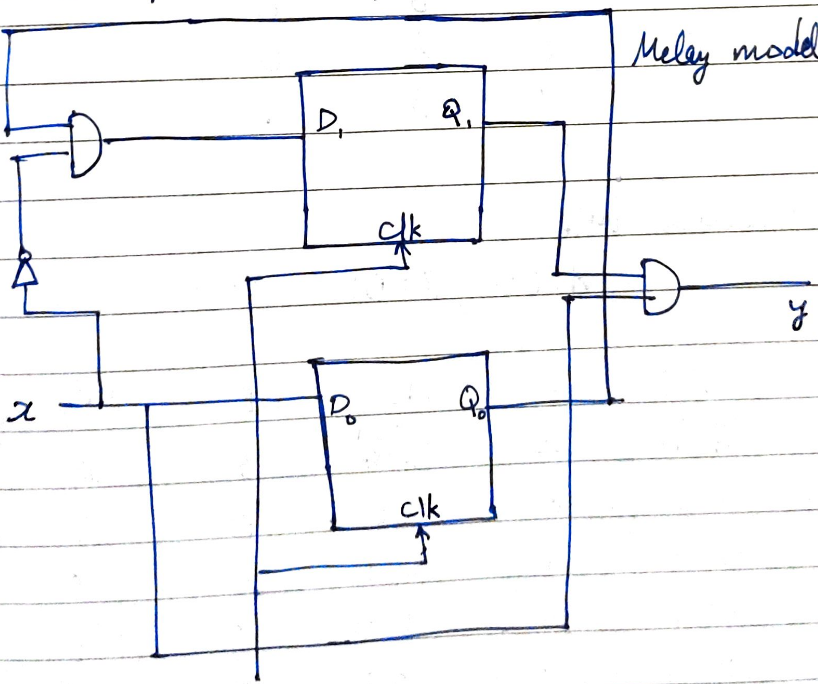
y

$Q_1 \backslash Q_0x$	00	01	11	10
0	0	0	0	0
1	0	1	X	X

$y = Q_1x$

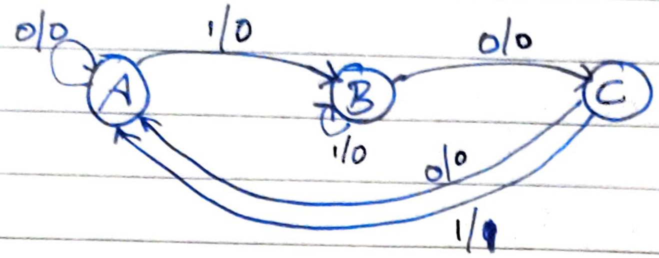
⑤ Hardware implementation / External view

Meley model FSM



— FSM (non-overlap)

1. State Diagram



2. State Table

PS	IP	NS	OP
A	0	A	0
	1	B	0
B	0	C	0
	1	B	0
C	0	A	0
	1	A	1

3. Conversion Table

A : 00 , B : 01 , C : 10

PS		IP	NS		OP	FF IP eqn.	
Q_1	Q_0	x	Q_1^+	Q_0^+	y	D_1	D_0
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	1	0	0	1
1	0	0	0	0	0	0	0
1	0	1	0	0	1	0	0

D_1

$Q_1 \backslash Q_0x$	00	01	11	10
0				1
1			X	X

$D_1 = Q_0\bar{x}$

D_0

$Q_1 \backslash Q_0x$	00	01	11	10
0		1	1	
1			X	X

$D_0 = \overline{Q_1}\,x$

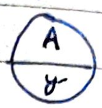
$y = Q_1 x$ (same as prev.)

9 10111

12th September, 2023

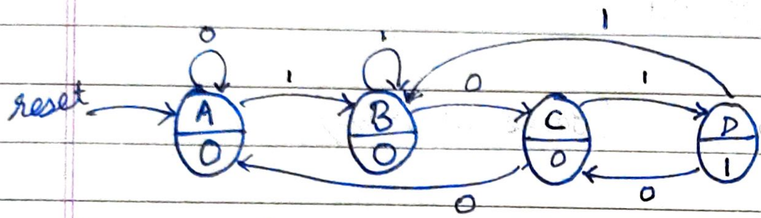
Moore Model FSM

Mealy	Moore
N states	N+1 states
ip / op	op associated with labels of states



* reset = 1 ⇒ trigger for FSM
  marks the initial state

eg: 101 (overlap)



A is initial state

B 1 → 10
    → 11

C 10 → 100
       101

D 101 → 1010
        → 1011

if no overlap (only last 2 states will change)

TT: $Q = D$

ET: $D = Q^+$

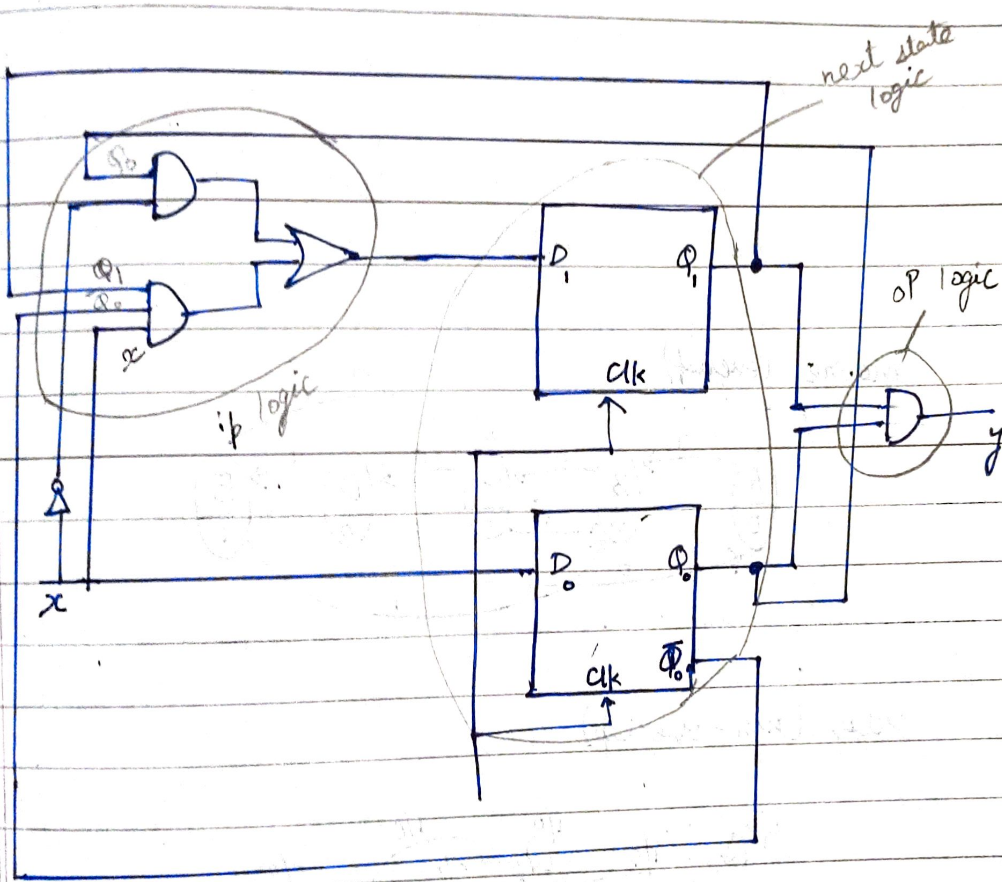
P.S $Q_1 Q_0$		x	N.S $Q_1^+ Q_0^+$		Y	D_1	D_0
00	A	0	A	00	0	0	0
		1	B	01		0	1
01	B	1	B	01	0	0	1
		0	C	10		1	0
10	C	1	D	11	0	1	1
		0	A	00		0	0
11	D	0	C	10	1	1	0
		1	B	01		0	1

$Q_1 \backslash Q_0 x$	00	01	11	10
0	0 (0)	0 (1)	0 (3)	1 (2)
1	0 (4)	1 (5)	0 (7)	1 (6)

$$D_1 = Q_0 \bar{x} + Q_1 \bar{Q}_0 x$$

$$D_0 = x \quad \text{(from Table)}$$

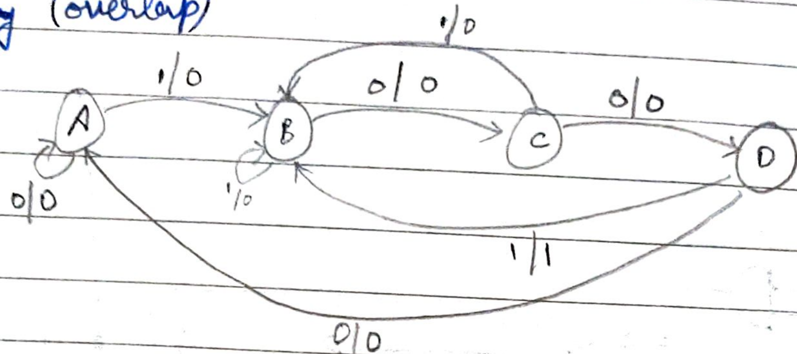
$y = Q_1 Q_0$ (depends only on Present state)
(independent of x)



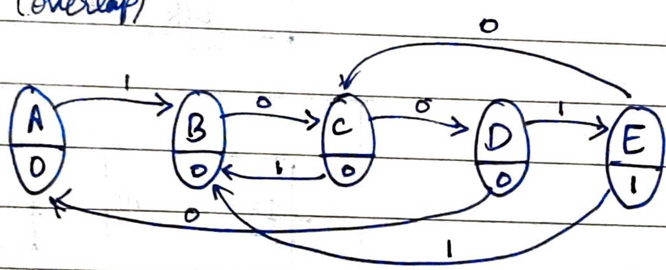
1 0 0 1

melay (overlapped) | moore (overlapped) | melay (non-overlapped) | moore (non-overlapped)

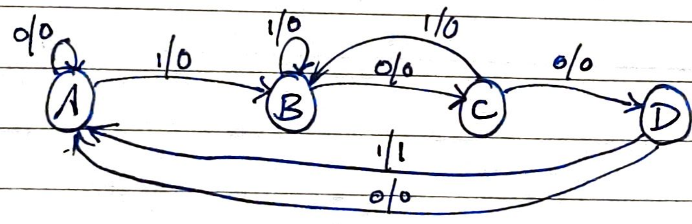
~~moore~~ melay (overlap)



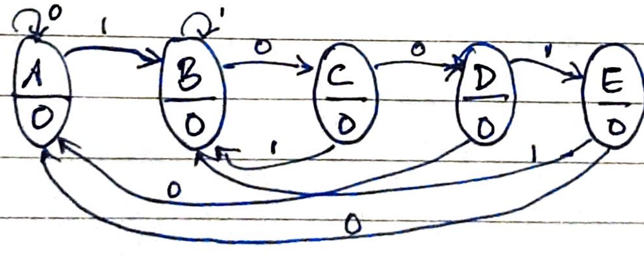
moore (overlap)



melay (non-overlap)

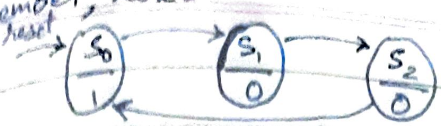


moore (non-overlap)



11th September, 2023

Q.



one hot coding

P.S			N.S					
Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+	D_2	D_1	D_0
0	0	1						
0	1	0						
1	0	0						

This increases complexity (3-FFs) — but still its better

hardware — Decoder

State Table :

P.S	N.S	y
S_0	S_1	1
S_1	S_2	0
S_2	S_0	0

Binary encoding (2 FFs) :

P.S	N.S	y
00	01	1
01	10	0
10	00	0

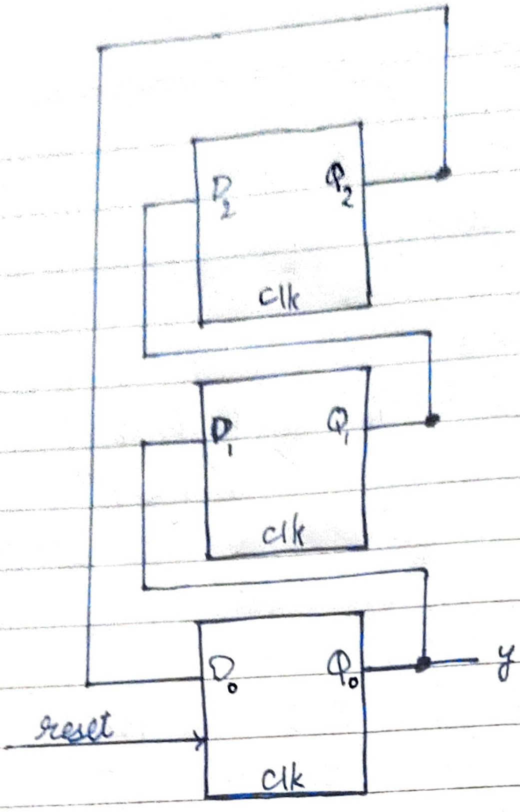
One Hot coding :

	P.S			N.S			y	$D_2\ D_1\ D_0$ ($D = Q^+$)
	Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+		
S_0 :	0	0	1	0	1	0	1	0 1 0
S_1 :	0	1	0	1	0	0	0	1 0 0
S_2 :	1	0	0	0	0	1	0	0 0 1

we use $Q_1 Q_0 x$ & $y D_1 D_0$ k-map but here we don't need k-maps, D can be represented in terms of Q

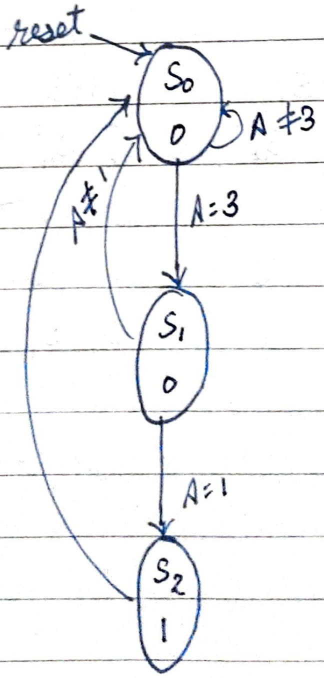
$$\left.\begin{aligned} D_2 &= Q_1 \\ D_1 &= Q_0 \\ D_0 &= Q_2 \end{aligned}\right\} \text{i/P eqn. of D-FF}$$

$y = Q_0$



clk	reset	
↑	1	
↑	0	
↑	0	
↑	0	S_0

9.



Password

A is 2bit
[1:0]
0, 1, 2, 3

State Table:

P.S	I/P	N.S	y
S_0	0	S_0	0
	1	S_0	
	2	S_0	
	3	S_1	
S_1	0	S_0	0
	1	S_2	
	2	S_0	
	3	S_0	
S_2	0	S_0	1
	1	S_0	
	2	S_0	
	3	S_0	

Conversion Table

PS	I/P	N.S	y	$(D = Q^+)$	
$Q_1 Q_0$	$A_1 A_0$	$Q_1^+ Q_0^+$		D_1 D_0	
S_0	00	00		0 0	0
00	01	00	0	0 0	1
	10	00		00	2
	11	01		01	3
S_1	00	00		0 0	4
01	01	10	0	10	5
	10	00		00	6
	11	00		00	7
S_2	00	00		00	8
10	01	00	1	00	9
	10	00		00	10
	11	00		00	11

12, 13, 14, 15 dont cares

K-map
4-variables
$(4 \times 4 = 16)$

D_1

I/P : $Q_1 Q_0$ $A_1 A_0$

$Q_1Q_0 \backslash A_1A_0$	00	01	11	10
00				
01		1		
11	X	X	X	X
10				

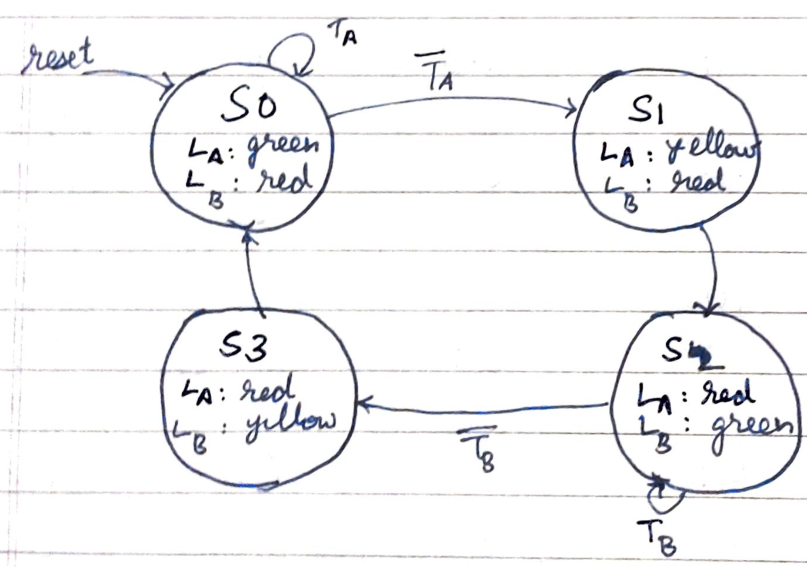
$\bar{Q}_1 Q_0 \bar{A}_1 A_0$

$Q_0 \bar{A}_1 A_0$

$D_1 = \bar{Q}_1 Q_0 \bar{A}_1 A_0$ (without don't cares) ✓
$D_1 = Q_0 \bar{A}_1 A_0$ (with don't cares) ✗
we cannot give don't care as i/p in password

$D_0 = \bar{Q}_1 \bar{Q}_0 A_1 A_0$ (we don't need k-map as there is only one 1)

q. Traffic Light Control



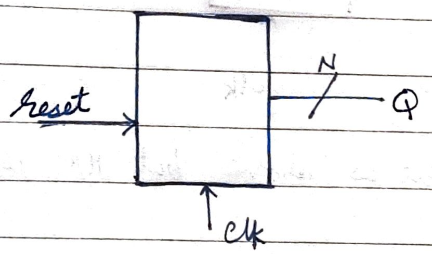
PS	I/P	NS
S_0	$\overline{T_A}$	S_1
	T_A	S_0
S_1	—	S_2
S_2	$\overline{T_B}$	S_3
	T_B	S_2
S_3	—	S_0

15th September, 2023

Counters

reset = 1 ⇒ start from 00

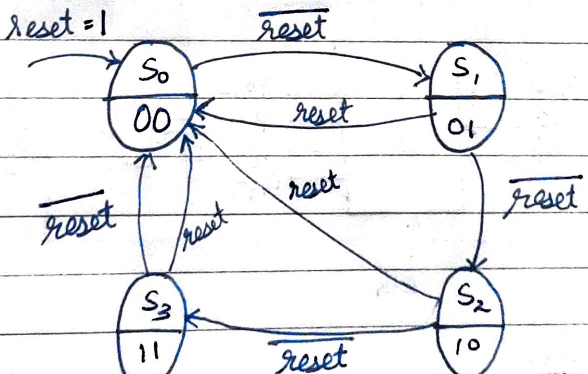
reset = 0 ⇒ take prev. value then +1.

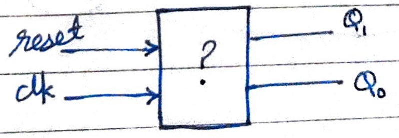


$$Q_{n+1} = Q_n + 1$$

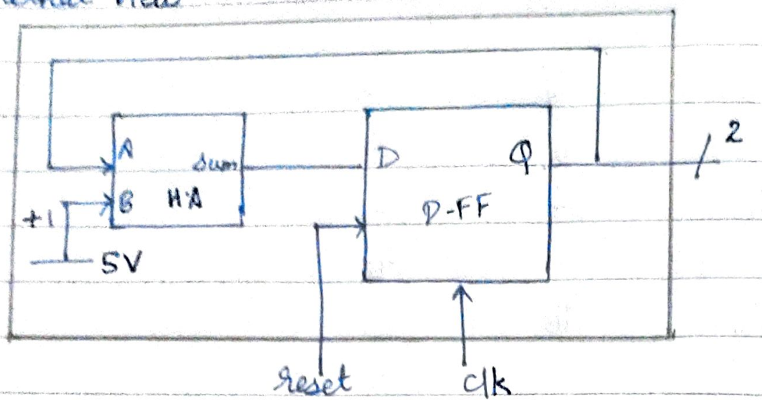
eg: N = 2 bit

$0 \to 1 \to 2 \to 3$



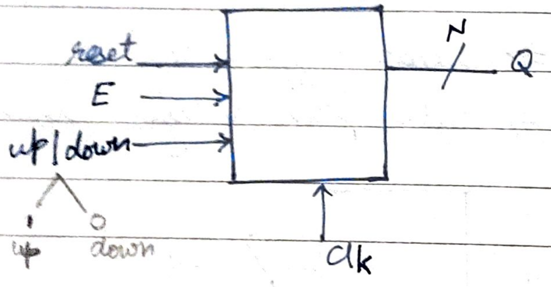


→ internal view



(it has no user i/p)

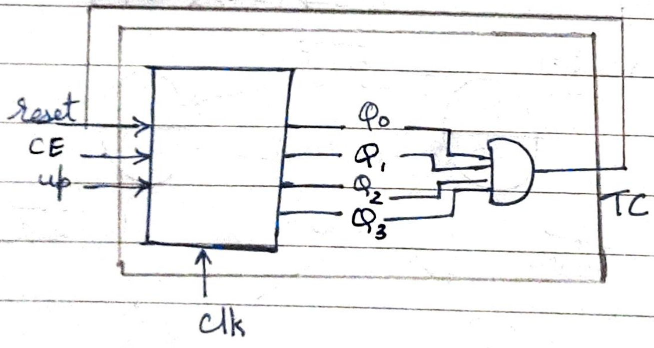
① → With Enable, Reset

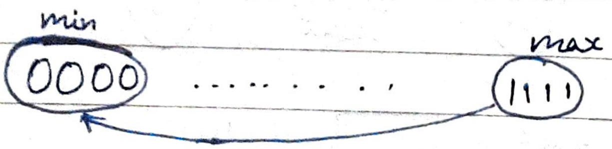


(internal view as above but HA with add/sub)

② Terminal counter

N = 4 no. of OP pins





(3) modulus (or) mod - N

$N = 10$

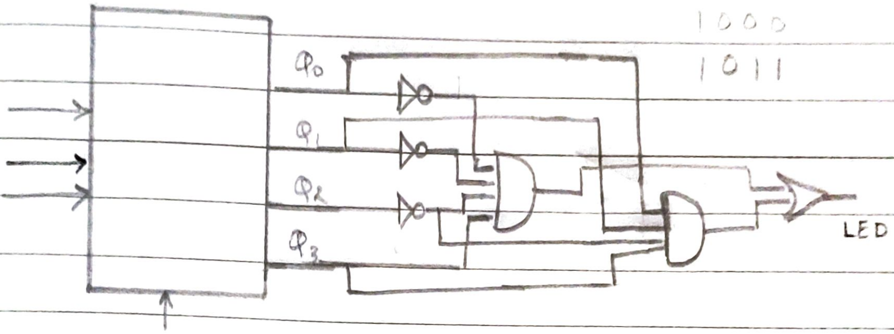
0 — 9

q. Design a 4-bit counter, activate LED as final OP once counter reaches 9th clock cycle & 12th clock cycle.

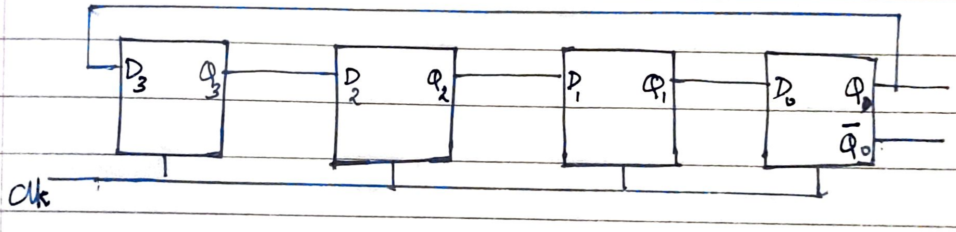
$Q_3 Q_2 Q_1 Q_0$

1000

1011



q. Design a 4-bit ring counter, considering initial state as 1000



Ring Counter

clk	Q_3	Q_2	Q_1	Q_0
↑	1	0	0	0
↑	0	1	0	0
↑	0	0	1	0
↑	0	0	0	1

1-iteration of 4-bit ring counter

$\boxed{Q_3 = Q_0}$ ring counter

$\boxed{Q_3 = \bar{Q}_0}$ Johnson counter

arithmetic right shift	1	0	0	0
	1	1	0	0
	1	1	1	0
	1	1	1	1
logical right shift	0	1	1	1
	0	0	1	1
	0	0	0	1
	0	0	0	0

1-iteration of 4-bit Johnson counter