Digital Logic – SNote

- Boolean Algebra & logic gate :

Law/theorem	law of addition	law of multiplication
identity	$X+0=X$	$X\cdot 1=X$
complement	$X+\bar{X}=1$	$X\cdot\bar{X}=0$
Idempotent	$X+X=X$	$X\cdot X=X$
Dominant	$X+1=1$	$X\cdot 0=0$
Involution	$(X')'=X$	[illegible]
Commutative	$X+Y=Y+X$	$X\cdot Y=YX$
Associative	$X+(Y+Z)=(X+Y)+Z$	$X(YZ)=(XY)Z$
Distributive	$X+YZ=(X+Y)(X+Z)$	$A(B+C)=AB+AC$
Demorgan	$(X+Y)'=X'\cdot Y'$	$(X\cdot Y)'=X'+\bar{Y}$
Absorption	$X+XY=X$	$X(X+Y)=X$

- Duality –

$(x\cdot\bar{y}\cdot z)+(x\cdot y\cdot\bar{z})+(y\cdot z)+0$

Duality → $(x+\bar{y}+z)\cdot(x+y+\bar{z})\cdot(y+z)\cdot 1$

$1\leftrightarrow 0$
$+\leftrightarrow\cdot$

- Logic gate –

1. Basic gate → NOT, AND, OR
2. Universal gate → NAND, NOR
3. arithmetic circuit or code converter or comparator → EXOR, EXNOR.

- Expression –

~~A~~ Commutative as C ; Associative as A.

Gate Name	Diagram (I/O & O/P)	properties
NOT	A → $\bar{A}$	
AND	A, B → AB	C, A
OR	A, B → A+B	C, A
NAND	A, B → $\bar{A}\bar{B}$	C
NOR	A, B → $\overline{(A+B)}$	C, A
EXOR	A, B → $A\oplus B=\bar{A}B+A\bar{B}$	C, A
EXNOR	A, B → $\overline{A\oplus B}=A\odot B=AB+\bar{A}\bar{B}$	

- NAND & NOR gate universal gate –

Logic gate	req. no. of NAND Gate	req no. of NOR gate
NOT	1	1
AND	2	3
OR	3	2
EXOR	4	5
EXNOR	5	4

• Imp boolean fun using NAND, NOR gate

NAND	NOR
→ $Y = AB + CD$	→ $(A+B)(C+D)$
S-1: $\bar{Y} = \overline{AB+CD} = \overline{AB} \cdot \overline{CD}$	S-1: $\bar{Y} = (\overline{A+B}) + (\overline{C+D})$
S-2: $\bar{\bar{Y}} = \overline{AB} \cdot \overline{CD}$	S-2: $\bar{\bar{Y}} = \overline{(\overline{A+B}) + (\overline{C+D})}$.

** • MINIMIZATION :

• SOP (Sum of product / sum of minterm) → $\bar{A}BC + A\bar{B}\bar{C} \rightarrow \Sigma(3,4)$

• POS (Product of sum / product of maxterm) → $(A+\bar{B}+C)(A+\bar{B}+\bar{C}) = \pi(2,3)$

→ Minterm — 0 → $\bar{A}$, 1 → A.

→ Maxterm — 0 → A, 1 → $\bar{A}$

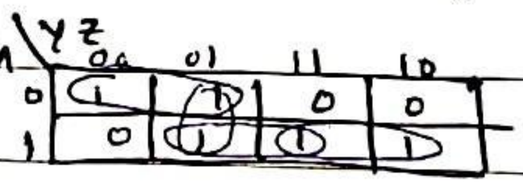
* → With N variables possible Boolean function = 2^{2^n}

→ With n variable K-map 2^n cells possible.

• Prime Implicant —

→ Implicant → (no. of 1's in K-map)

→ prime imple → no. of adjacent squares in K-map.

ex: 

Imp - (5), P.I - 4.

*** • Combinational circuit :

• ADDER —

→ Half adder :

* Sum = $A \oplus B$, Carry = AB, NAND & NOR req to imp = 5

→ Full adder : 2 HA + 1 OR gate

Sum = $A \oplus B \oplus C$, Carry = $AB + BC + AC$, NAND & NOR req to imp = 9

• Subtractors —

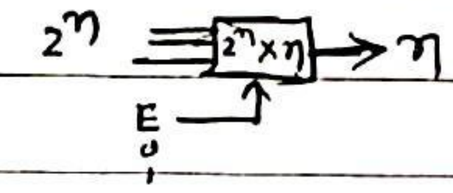
→ Half Sub :

Diff = $A \oplus B$, Borrow = $\bar{A}B$, NAND & NOR req to imp = 5

→ Full sub : 2 HS + 1 OR

Diff = $A \oplus B \oplus C$, ~~Carry~~ Borrow = $\bar{A}B + BC + \bar{A}C$, NAND & NOR = 9.

• Encoders —



** • Decoder —

4+2+1 = 7

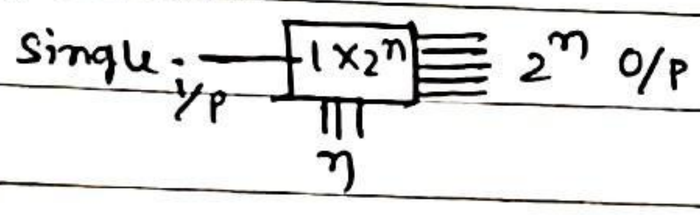
$\frac{8}{x} \cdot \frac{4}{x} = \frac{4^2}{x} = \frac{2}{x}\frac{1}{x}$

$\frac{64}{4} \times 4 \times 1$

** • MUX :

Given MUX	To be Imp	req. no.
2×1	4×1	3
2×1	8×1	7
2×1	$2^n \times 1$	$2^n - 1$
4×1	64×1	21
8×1	89×1	9
8×1	256×1	32+4+1 = 37

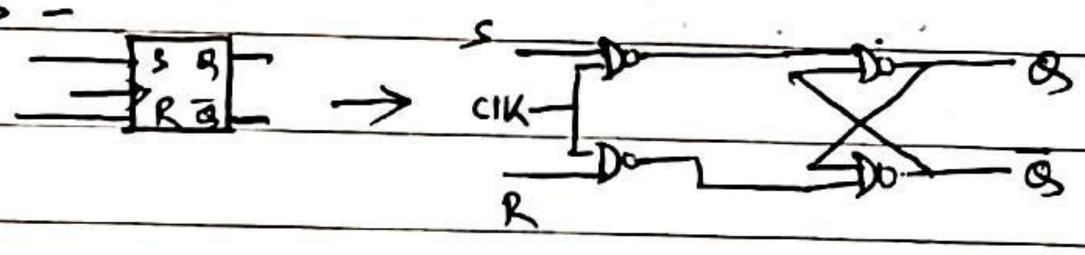
** • DE-MUX :



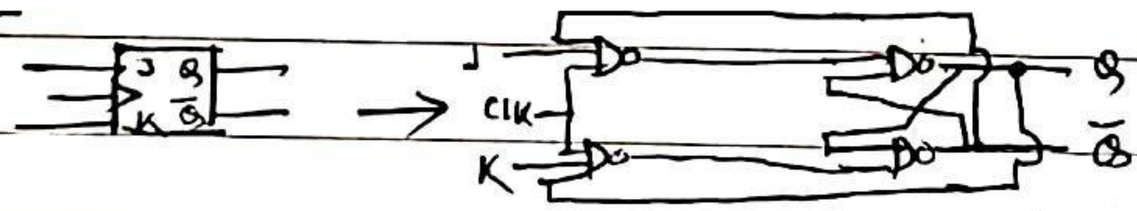
• Sequential circuit :

• flip-flop (SR, JK, D, T) –

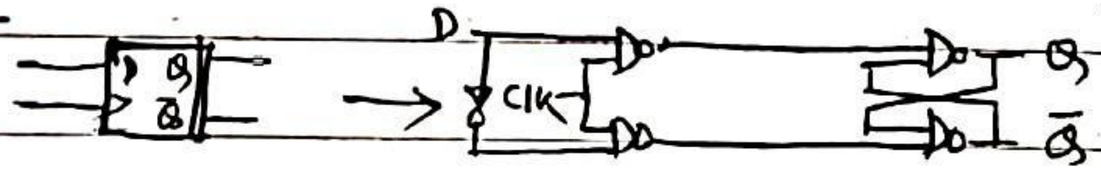
• SR flip-flop –



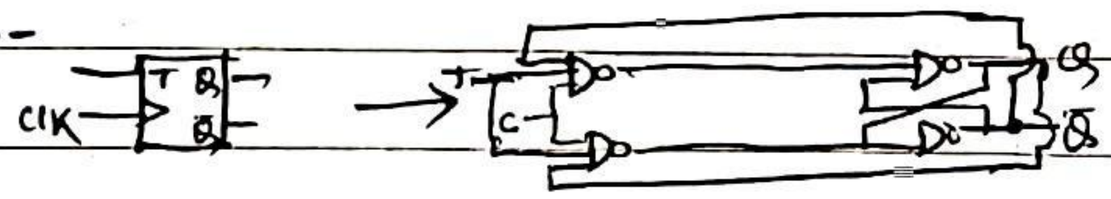
• JK-flip flop –



• D-flipflop –



• T-flip-flop –



→ Function Table –

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

JK	Q_{n+1}
0 0	Q_n
0 1	0
1 0	1
1 1	$\bar{Q}_n$

→ Toggle

D	Q_{n+1}
0	0
1	1

T	Q_{n+1}
0	Q_n
1	$\bar{Q}_n$

→ Toggle

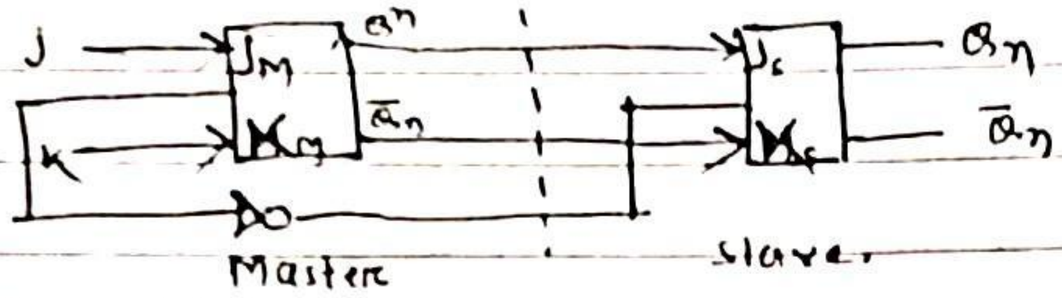
• RACE AROUND CONDITION – → propagation delay of JKff.

When, $J=K=1$ and $t_{pd} << T_{CLK}$

→ TO avoid race around condition –

$t_{pd} > T_{CLK}$

• Master-slave JK-FF -

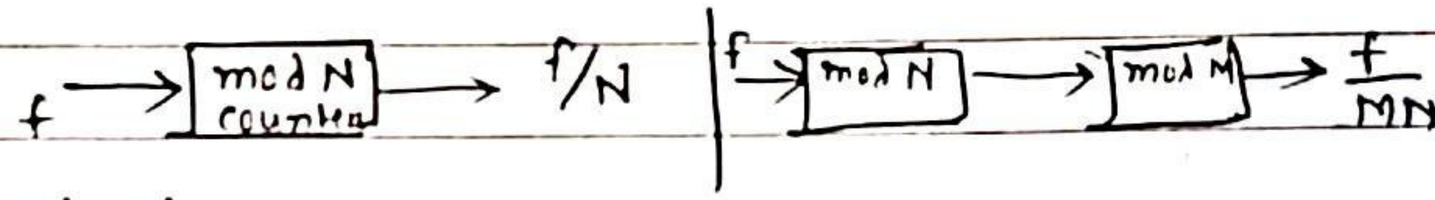


• COUNTER:

→ Seq ckt capable to count the no. of clk pulses. (set of FF)

* → with n-ff max possible states in the counter is 2^n.

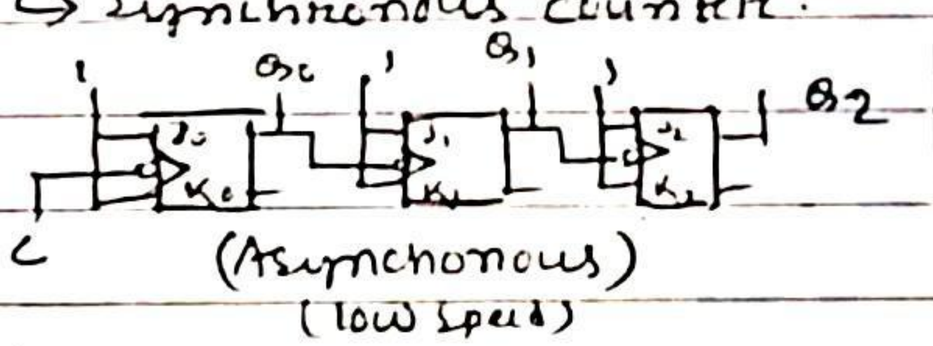
$\boxed{N \leq 2^n}$ n → no. of ff, N → no. of states.

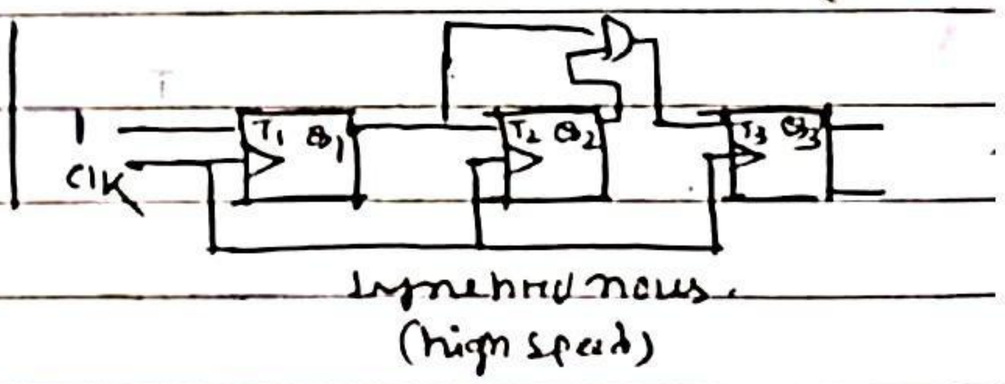


→ counter type

→ Asynchronous ckt (ripple counter)

→ Synchronous counter.





• Asynchronous (ripple)

→ 2 bit counter, req. no. of ff = 2, 2^n Mod = 2^2 = 4 MOD counter.

Note:

(I) -ve edge trigger FF → Q as clk → up counter.

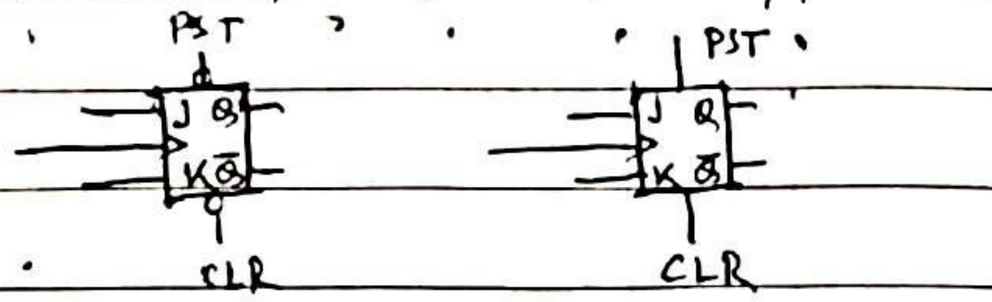
(II) -ve edge trigger FF → $\bar{Q}$ as clk → Down counter.

(III) +ve edge trigger FF → Q as clk → Down counter.

(IV) +ve edge trigger FF → $\bar{Q}$ as clk → Up counter.

→ FF should have toggle mode (JK, T). $\boxed{J=K=1 \text{ or } T=1}$

→ JK-FF with preset & clear i/p -



→ if $t_{ps} > t_{CLK}$ → counter may skip a state.

→ To avoid $t_{CLK} \geq n.t_{ps}$.

• Synchronous Counters –

→ Req steps to design synchronous counters –

→ No. of FF req $n=2,\ N \leq 2^n = 4$

→ state diagram $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$ (with $3 \rightarrow 0$)

→ choice of FF & excitation table of FF

→ ~~minimal exp for the excitations.~~

→ Draw logic diagram.

**

Excitation table –

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

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

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

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

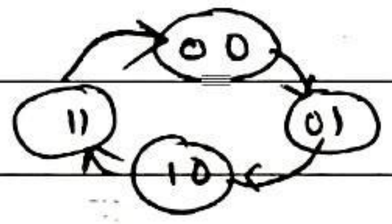
→ 2-bit up counter,

$n = 2$ (no. of FF) $\quad N \leq 2^n$

$N \leq 4$

(i) no. of ff = 2

(ii)



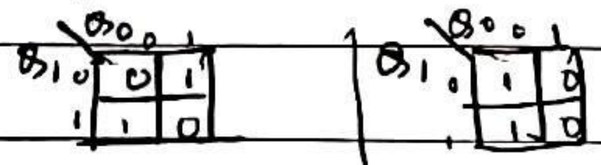
(iii) D-FF (Excitation)

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

(iv)

CLK	Q_1	Q_0	Q_1	Q_0	D_1	D_0
1	0	0	0	1	0	1
2	0	1	1	0	1	0
3	1	0	1	1	1	1
4	1	1	0	0	0	0

(v)



$D_1 = \overline{Q}_1 \oplus Q_0$; $D_2 = \overline{Q}_0$

(vi) Diagram.

• REGISTERS:

→ n bit reg consists n no. of ff.

→ D-FF are used in reg.

• Shift reg type –

↳ SISO, SIPO, PIPO, PISO.

→ If n-bit shift reg SISO → $(2n-1)$ clk pluses req. to

• Sift register counter –

→ ring counter & Johnson counter (twisted ring counter)

→ n bit ring counter count n-clock pulses.

→ n " Johnson counter count 2n -CP

* Frequency division

n-bit ripple counter	$\rightarrow f/2n$
" ring "	$\rightarrow f/n$
" Jonson's "	$\rightarrow f/2n$

- Numbers Systems —
- any num → Decimal num , r → base.

$$(N)_r = a_n r^n + a_{n-1} \cdot r^{n-1} + a_{n-2} \cdot r^{n-2} + \dots + a_1 r^1 + a_0 r^0 + a_{-1} r^{-1} + a_{-2} r^{-2}$$

$(123)_4 = 1\times4^2 + 2\times4^1 + 3\times4^0 \Rightarrow 16 + 8 \Rightarrow (24)_{10}$

- Decimal num → any num -

$(42)_{10} = (\ )_2 = (\ )_3 = (\ )_8 = (\ )_{16}$

42 | 2|42 | 3|42 | 8|42 | 16|42

2|42
2|21 — 0
2|10 — 1
2|5 — 0
2|2 — 1
1 — 0
$(101010)_2$

$(42.63)_{10} = (\ )_2 = (\ )_3 = (\ )_8 = (\ )_{16}$

2|42
2|21 — 0
2|10 — 1
2|5 — 0
2|2 — 1
1 — 0
$(101010)_2$

$0.63 \times 2 = 1.26 \rightarrow 1$
$0.26 \times 2 = 0.52 \rightarrow 0$
$0.52 \times 2 = 1.04 \rightarrow 1$
⋮

8|42
8|5 — 2

$0.63 \times 8 = 5.04 \rightarrow 5$
$0.04 \times 8 = 0.32 \rightarrow 0$
$0.32 \times 8 =$ ⋮

$(101010.101)_2$

- Complement:
  - → (r-1)'s complement.
  - → r's complement.

r → bare number.

r = 2

(r-1)'s → 1's complement → $\begin{array}{r} 111 \\ -\ 110 \\ \hline 001 \end{array}$

r's → 2's complement → 1's comp + 1.

r = 8

(r-1)'s → 7's comp → $\begin{array}{r} 777 \\ -\ 125 \\ \hline (652) \end{array}$

r's → 8's comp → $\begin{array}{r} 778 \\ -\ 125 \\ \hline (653) \end{array}$

- Addition & subtraction —

$\begin{array}{r} 101 \\ +\ 001 \\ \hline 110 \end{array}$	$\begin{array}{r} 10010 \\ +\ 00111 \\ \hline 11001 \end{array}$	$\begin{array}{r} (1011.1)_2 \\ -\ (0011.0)_2 \\ \hline (1000.1)_2 \end{array}$

- Signed binary num -

→ Unsigned (0 - N) → Signed (-N to +N)

+ve → '0'	-ve → '1'

- Range of Binary num - [n → no. of bit]

→ Unsigned num (0 to $2^n - 1$)

→ Signed num - $(2^{n-1} - 1$ to $+ 2^{n-1} - 1$

→ Signed 1's complement - "

→ Signed 2's comp - -2^{n-1} to $+2^{n-1} - 1$

- Codes:

Decimal	BCD or 8421	Excess-3
0	0 0 0 0	→ 0000 + 0011 (3)
1	0 0 0 1	→ 0001 + 0011 (3)
2	0 0 1 0	→ 0010 + 0011 (3)
3	0 0 1 1	→ 0011 + 0011 (3)

Binary → 1 1 0 1 0 1 1 1 0
to
Gray → 1 0 1 1 1 1 0 0 1

Gray → 1 0 1 1 1 1 0 0 1
to
Binary → 1 1 0 1 0 1 1 1 0