#> DE :>

#> Syllabus :

- DE Basics
- Boolean Algebra laws & logic gates
- Boolean exp.^{n} (SOP/POS) minimiz.^{n}
- Combinational circuit
- Sequential circuit
- Number system & Number rep.^{n}

#> DE Basics:

Problem → Truth table → Boolean exp.^{n} → minimize B.E (K-map) → implement using logic gates

#> B E Laws & logic gates

•> Laws:

idempotent	associative	commutative
$a\cdot a=a$	$a\cdot(b\cdot c)=(a\cdot b)\cdot c$	$a\cdot b=b\cdot a$
$a+a=a$	$a+(b+c)=(a+b)+c$	$a+b=b+a$

Distributive	Demorgan
$a\cdot(b+c)=a\cdot b+a\cdot c$	$(a\cdot b)'=a'+b'$
$a+(b\cdot c)=(a+b)\cdot(a+c)$	$(a+b)'=a'\cdot b'$

Identity:

$a+0=a$	$a+1=1$	$\bar{0}=1$	$a+\bar{a}=1$
$a\cdot 0=0$	$a\cdot 1=a$	$\bar{1}=0$	$a\cdot\bar{a}=0$

$(a')'=a$ Involution

•> Gates:

NOT) $y=x'$

OR) $y=a+b$ — Idempotent, comm., A.M.

AND) $y=a\cdot b$ — Idempotent, comm., A.M.

NOR) $y=(a+b)'$ — comm.

NAND) $y=(a\cdot b)'$ — comm.

Ex-OR) $y=A\oplus B = a\bar{b}+\bar{a}b$ — odd 1 ⇒ 1 detect; comm. & A.M.

Ex-NOR) $y=A\odot B = \bar{a}\cdot\bar{b}+ab$ — even 1 ⇒ 1 detect; comm. & A.M.

Note: If no. of i/p's are odd, then $\odot$ follows $\oplus$

odd: $\oplus=\odot$ — $a\oplus b\oplus c$

even: $\overline{\oplus}=\odot$ — $a\oplus b$

Note:

$a\oplus b = a'\odot b = (a'\odot b')' = (a'\oplus b)'$

$a\odot b = a'\oplus b = (a\oplus b)' = a'\odot b'$

Note: $A\oplus P\oplus A\oplus P = \underbrace{A\oplus A}_{0}\oplus\underbrace{P\oplus P}_{0} = 0\oplus 0 = 0$

•> Boolean exp.:

1) SOP (considers 1) (used more often)
2) POS (" 0)

Ex:

x	y	z	o/p
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

SOP: $o/p(x,y,z) = \Sigma_m(1,4,6,7)$

$= \Pi_M(0,2,3,5)$

1) SOP: Example $= x'y + y'z' + xyz$

Product term which have all literals = minterm

As in above Ex: $o/p(x,y,z) = (\bar{x}\bar{y}z)+(x\cdot\bar{y}\cdot\bar{z})+(xy\bar{z})+(x\cdot y\cdot z)$

$= m_1 + m_4 + m_6 + m_7$

$= \Sigma m(1,4,6,7)$

Canonical form = Every product term have all the literals

Ex: $a\bar{b} + ac + abc$

$a\bar{b}(c+\bar{c}) + ac(b+\bar{b}) + abc$

$a\bar{b}\bar{c} + a\bar{b}c + abc = \Sigma_m(4,5,7)$

100

2) POS: Example $= (\bar{a}+b)\cdot(\bar{b}+\bar{c})\cdot(a+c)$

Sum terms which have all literals = maxterm

As in above Ex: $o/p(x,y,z) = (x+y+z)\cdot(x+\bar{y}+z)\cdot(x+\bar{y}+\bar{z})\cdot(\bar{x}+y+\bar{z}) = M_0\cdot M_2\cdot M_3\cdot M_5$

$= \Pi_M(0,2,3,5)$

Canonical form = Every Sum term have all the literals

Ex: $a\cdot(\bar{b}+c)\cdot(\bar{a}+b+\bar{c})$

$= (a+b\bar{b}+c\bar{c})\cdot(a\bar{a}+\bar{b}+c)\cdot(\bar{a}+b+\bar{c})$

$= (a+\bar{b}+\bar{c})(a+\bar{b}+c)(a+b+\bar{c})(a+b+c)\cdot$

$\cancel{(a+\bar{b}+c)}\cdot(\bar{a}+\bar{b}+c)\cdot$

$(\bar{a}+b+\bar{c})$

$= (011)(010)(001)(000)\cdot(110)\cdot(101)$

$m_3\cdot m_2\cdot m_1\cdot m_0\cdot m_6\cdot m_5$

•) No. of fx's possible

Variables	Comb's	Fx'ns
n	$\to 2^n$	$\to (2)^{2^n}$
z	$\to y^z$	$\to (x)^{y^z}$

y = nature of variable
x = nature of fx'n
z = no. of variable

• 2^n minterms possible

•) Duality: $0 \to 1$ & $1 \to 0$ & or → and, and → or
but nature of variable remain same
Ex: $a \cdot b$ dual is $a + b$
$a + 0$ dual is $a \cdot 1 = a$

1.) Neutral fx'n: no. of minterm = no. of maxterm
2.) Self dual: If fx'n & it's dual are same
Ex: $f = ab + bc + ca$
$f(\text{dual}) = (a+b)\cdot(b+c)\cdot(c+a)$
$= \dots$
$= ab + bc + ca$
Neutral
Self dual

From 'n' variables $(2)^{2^{n-1}}$ self dual fx'ns are possible

•) To check self-dual:
1. check if no. of minterms = no. of maxterms
2.
0 ↔ 7
1 ↔ 6
2 ↔ 5
3 ↔ 4

0 है तो 7 नहीं होना चाहिए

Pair of mutually exclusive terms $= 2^n/2$

Ex:) $f(a,b,c) = \Sigma m(0,1,2,4)$ ✓
$f(a,b,c) = \Sigma m(0,1,6,7)$ ✗

3.) Orthogonal :) If complement & dual of fx'n are same
To check orthogonal :-
1. minterms = maxterms
2.
0 — 7
1 — 6
2 — 5
3 — 4

0 है तो 7 भी होना चाहिए

4.) Functionally complete: If a fx'n can implement Not with either AND या OR
(support of 0, 1, complement not allowed)
Partially fxnlly complete:
(support of 0, 1 allowed, complement not allowed)
Note:) SOP can be implemented using AND-OR (NAND-NAND)
POS can be implemented using OR-AND (NOR-NOR)

*) BE. simplification:
1) using k-map 2) using boolean laws

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

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

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

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

c \ ab	00	01	11	10
0	0	2	6	4
1	1	3	7	5

a \ bc				
	0	1	3	2
	4	5	7	6

bc \ a		
	0	4
	1	5
	3	7
	2	6

a \ b		
	0	1
	2	3

b \ a		
	0	2
	1	3

ab\cd → cd\ab ↓ a\bc → always from a
bc\a ↓ c\ab ↓

K-map for POS

c+d \ a+b	$a+b$ 0+0	$a+\bar b$ 0+1	$\bar a+\bar b$ 1+1	$\bar a+b$ 1+0
0+0	0	4	12	8
0+1	1	5	13	9
1+1	3	7	15	11
1+0	2	6	14	10

इसमें 0 मत (?) से solve होगा
i.e, $a = 0$
$\bar a = 1$

Example

y+z \ w+x	0+0	0+1	1+1	1+0
0+0	0	0 (4)	0 (12)	8
0+1	1	X (5)	0 (13)	0 (9)
1+1	3	7	X (15)	X (11)
1+0	2	X (6)	0 (14)	10

$(\bar w + \bar z)$
$(\bar x + z)$
$\Rightarrow (\bar w + \bar z)(\bar x + z)$

literals ...

1 ⟷ 6
2 ⟷ 5
3 ⟷ 4

Pair of mutually exclusive terms $= 2^n/2$

Ex:) $f(a,b,c) = \Sigma_m(0,1,2,4)$ ✓

$f(a,b,c) = \Sigma_m(0,1,6,7)$ ✗

3.) Orthogonal :) If complement & dual of fx^n are same

To check orthogonal :—
1. minterms = maxterms
2. 0 — 7
   1 — 6
   2 — 5
   3 — 4

4.) Functionally complete: If a fx^n can implement Not with either AND या OR
(support of 0, 1, complement not allowed)

Partially fxnally complete:
(Support of 0, 1 allowed, complement not allowed)

~~Note:) SOP can be implemented using AND-OR (NAND-NAND)~~

POS can be implemented using OR-AND (NOR-NOR)

Note:)
Absorption Law
$a + ab = a$
$a \cdot (a+b) = a$

Compensation
$ab + \bar{a}c + bc = ab + \bar{a}c$
$(a+b)(\bar{a}+c)(b+c) = (a+b)(\bar{a}+c)$

$a + \bar{a}b = a + b$
$a \cdot (\bar{a}+b) = a \cdot b$

Note:)
NOT + OR = NOR
NOT + AND = NAND
] universal gates

Example: $f(abcd) = \Sigma_m\{4,5,6,7,8,9,10,11,13,14\}$

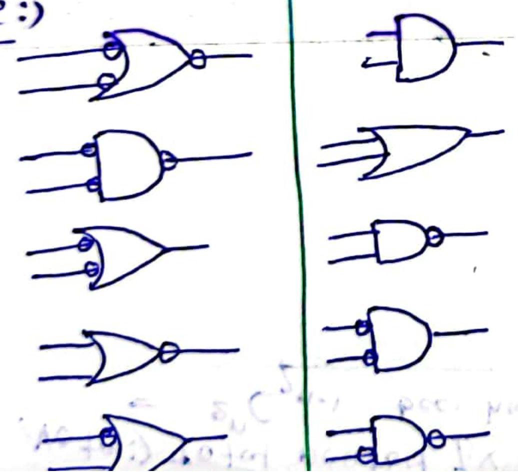
cd \ ab	00	01	11	10
00	(0)	1 (4)	(12)	1 (8)
01	(1)	1 (5)	1 (13)	1 (9)
11	(3)	1 (7)	(15)	1 (11)
10	(2)	1 (6)	1 (14)	1 (10)

EPI

$\bar{a}b + a\bar{b} + b\bar{c}d + bc\bar{d}$
$a\bar{c}d$ $ac\bar{d}$

EPI = 2
NEPI = 4,
PI = 6
Different expressions possible = 4
literals in minimal expression = 10

Note:)



Note:) To check if f is functionally complete

Ex:) $f(x,y,z) = \bar{x}yz + x\bar{y} + \bar{y}\bar{z}$

✓ ① Not preserve 0 $\Rightarrow \bar{0}00 + 0\bar{0} + \bar{0}\bar{0} = 1$

✓ ② Not preserve 1 $\Rightarrow \bar{1}11 + 1\bar{1} + \bar{1}\bar{1} = 0$

✓ ③ Not self dual. $\bar{x}yz + x\bar{y}\frac{z}{\bar{z}} + \frac{x}{\bar{x}}\bar{y}\bar{z}$

0 — 7
1 — 6
2 — 5
③ — ④

3   4, 5   0, 4

Note:) Total neutral fx'ns possible

$= {}^{2^n}C_{2^{n-1}}$ bec. half of 2^n minterms are 1

Note: #Neutral $\equiv {}^{2^n}C_{\frac{2^n}{2}}$

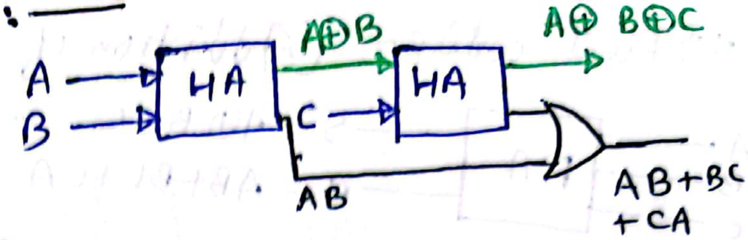
#Self Dual $\equiv 2^{\frac{2^n}{2}} \equiv$ Orthogonal

Note:)

	NOT	AND	OR	NAND	NOR	EXOR	EXNOR
	g	aun	cat	g	Play	with	mouse
NAND	1	2	3	1	4	4	5
NOR	1	3	2	4	1	5	4
	g	dub	to	play	a	sport	ball

Combinational circuits :—

#) Adders :—



*) Summary :—

1) Half adder

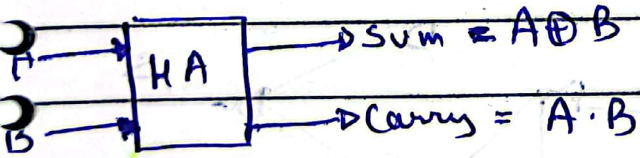
- Sum = $A \oplus B$
- Carry = $A \cdot B$
- Using NAND = 5 gates
- using NOR = 5 gates.

3

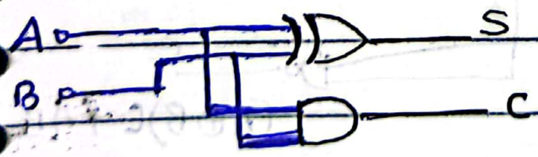
2) Full adder

- Sum = $A \oplus B \oplus C$
- carry = $AB + BC + CA$
- Using NAND = 9 gates
- using NOR = 9 gates.
- Using HA + OR = 2 HA & 1 OR

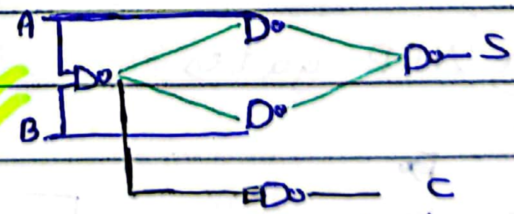
*) Half adder :— (Add.^{n} of two 1-bit numbers)



A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



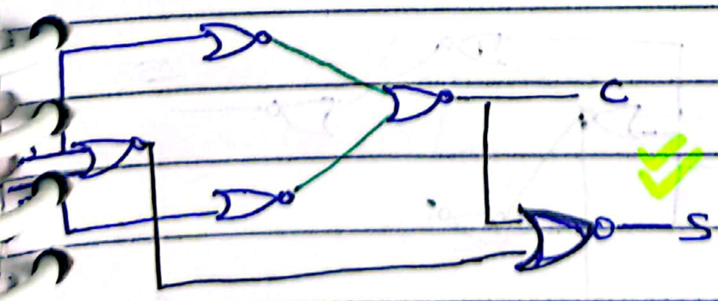
Using NAND



बांधी plate

चमचम

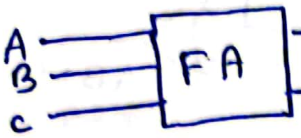
using NOR



झोपड़ी

चिमनी

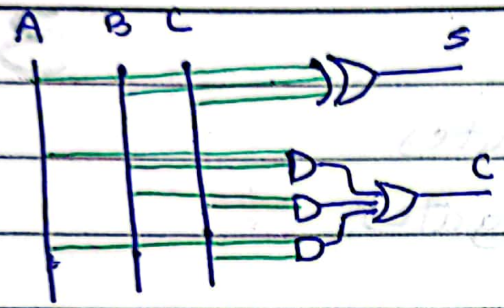
*) Full adder : (Addition of three 1-bit numbers)



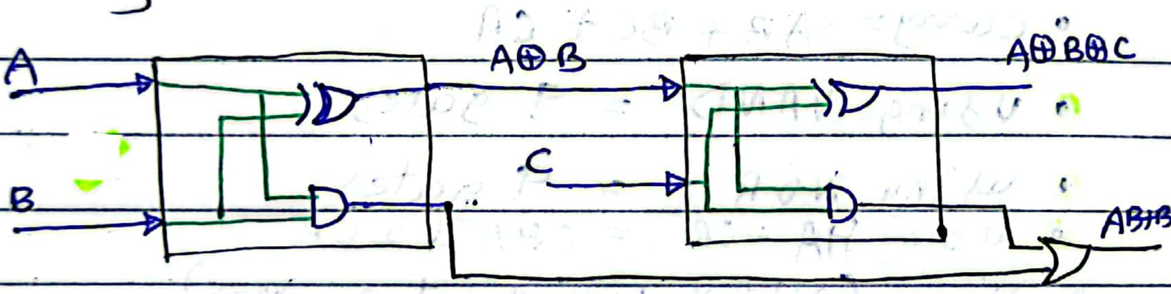
$S = A \oplus B \oplus C$

$C = AB + BC + CA$

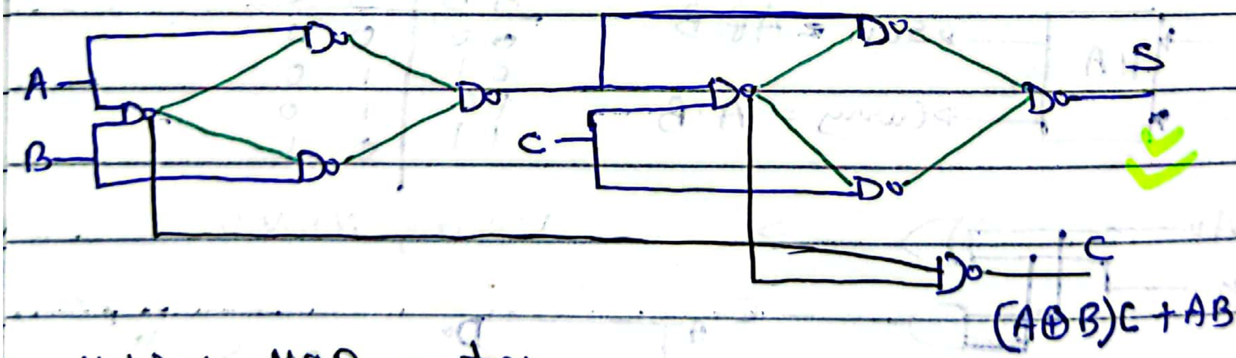
A	B	C	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



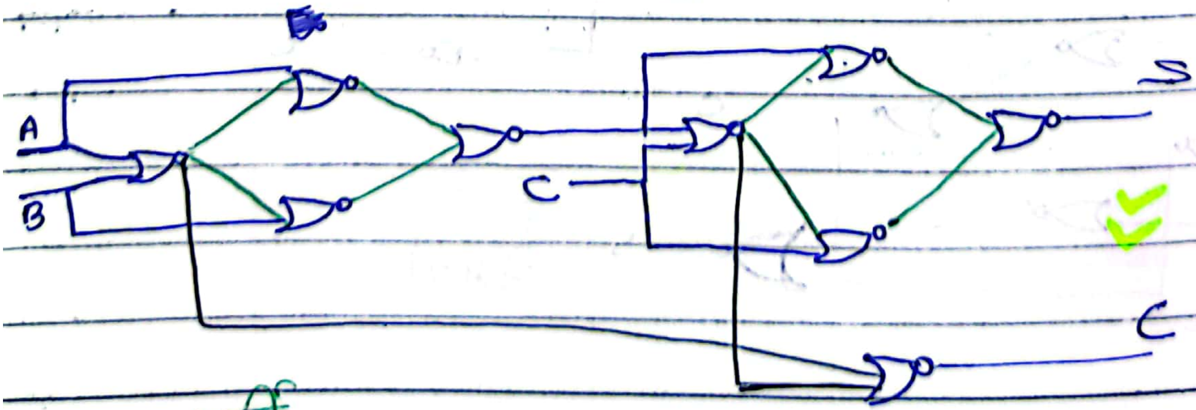
using HA's :-



using NAND gates:-



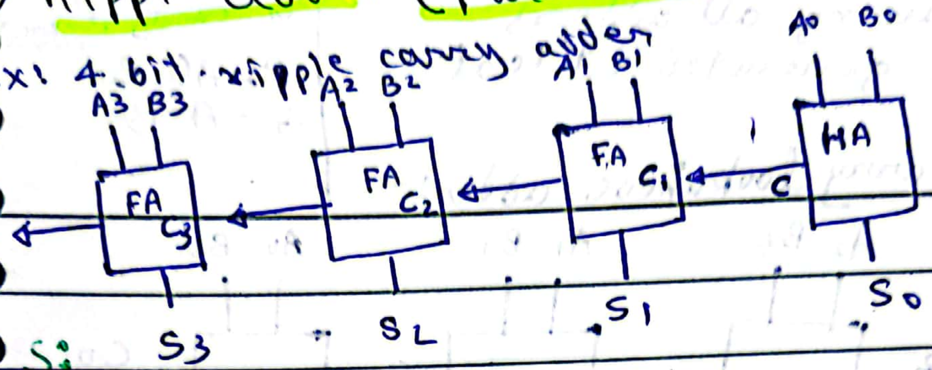
using NOR gates



2 कटोरी
2 plate
1 चम्मच

* Ripple adder (Parallel adder) (Ripple carry adder)

⇒ Ex: 4 bit ripple carry adder



4

S: S_3 S_2 S_1 S_0

- To implement 4 bit parallel adder :-
  - ⇒ 3 FA + 1 HA
  - ⇒ 3 [2 HA + 1 OR] + 1 HA
  - ⇒ 7 HA + 3 OR (Using only HA's)
  - ⇒ 7 [XOR + AND] + 3 OR
  - ⇒ 7 XOR + 7 AND + 3 OR (using only basic gates)
  - ⇒ 4 FA (using only FA's)

- To implement n bit parallel adder :-
  - ⇒ n FA's
  - ⇒ n-1 FA's + 1 HA
  - ⇒ (2n-1) HA's + (n-1) OR
  - ⇒ (2n-1) [XOR + AND] + (n-1) OR

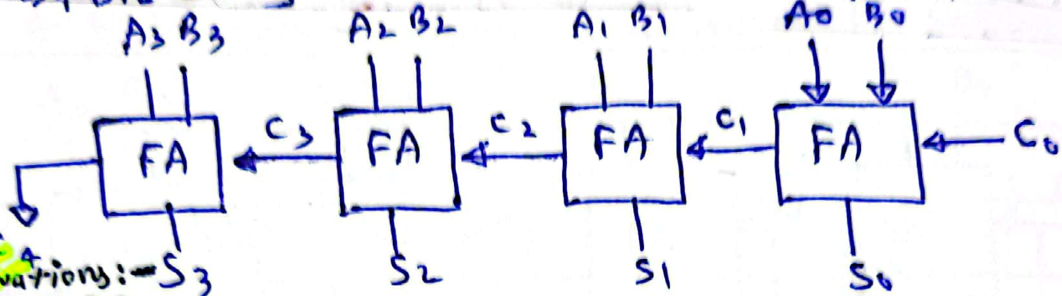
- Delay :- for n-bit parallel adder

$$\text{delay} = (n-1) \times \underbrace{t_{pd}}_{\text{of carry}} + \max\left[\underbrace{t_{pd}}_{\text{of sum}}, \underbrace{t_{pd}}_{\text{of carry}}\right]$$

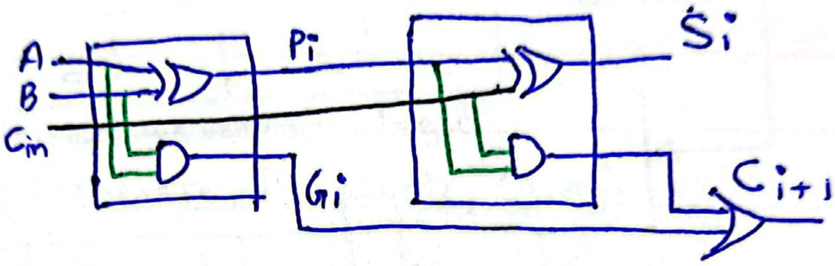
*) Carry lookahead adder:

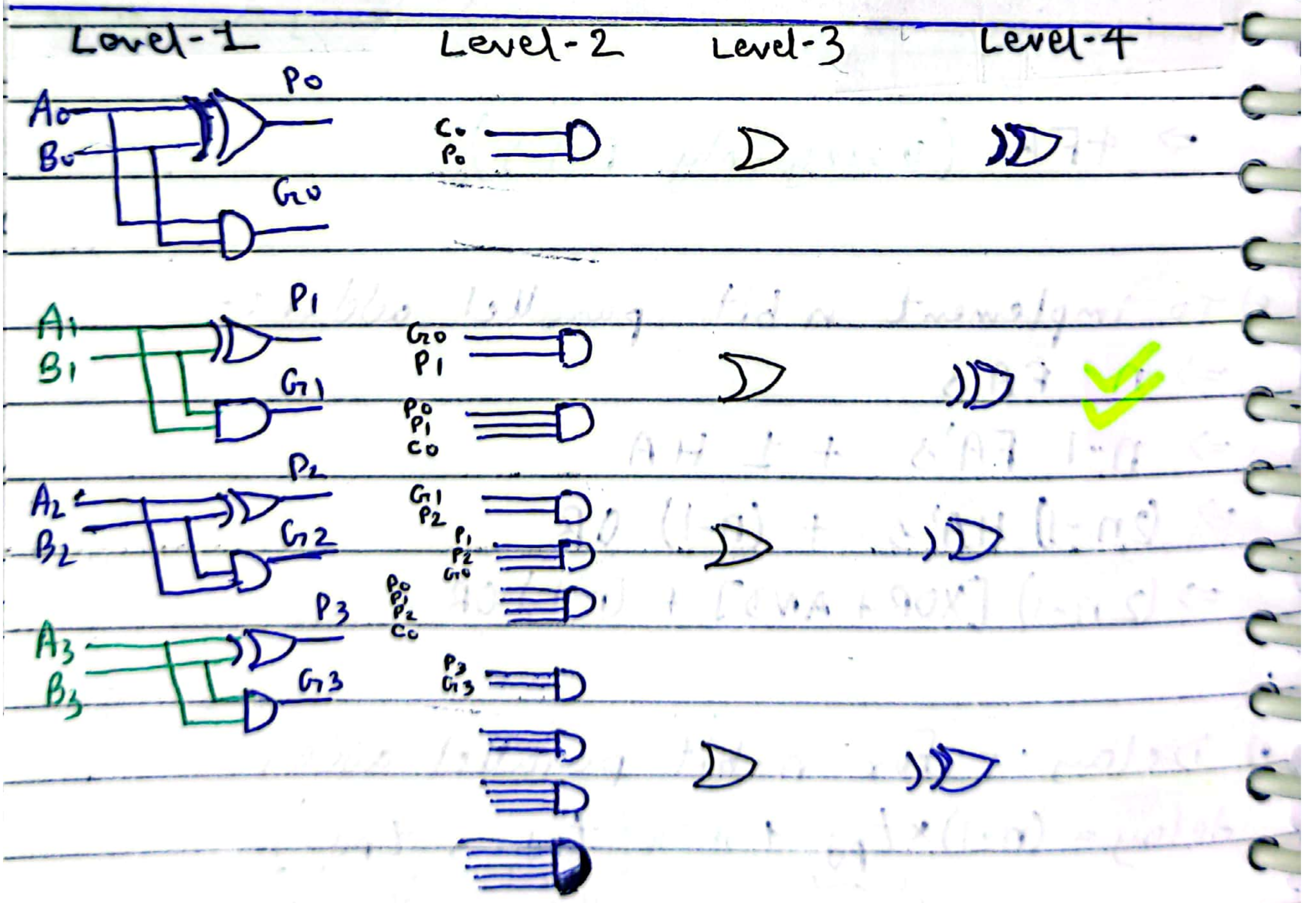
P : carry propagate
G : carry generator
$P = A \oplus B$
$G = A \cdot B$

•) Fastest among all adders
carry is generated at last
•) 4-bit carry lookahead adder :-



Equations:-
$P_i = A_i \oplus B_i$
$G_i = A_i \cdot B_i$
$S_i = P_i \oplus C_i$
$C_{i+1} = G_i + P_i \cdot C_i$





4 [AND + OR]   [1+2+3+4] AND   4 [OR]   4 [XOR]

•) for n-bit carry lookahead adder :-

n [AND + OR] + [1+2+ .. n] AND + n [OR] + n [XOR]

i) Delay :-

Level 1	Level 2	Level 3	Level 4
t_{pd} MAX(AND, XOR)	t_{pd} (AND)	t_{pd} (OR)	t_{pd} (XOR)

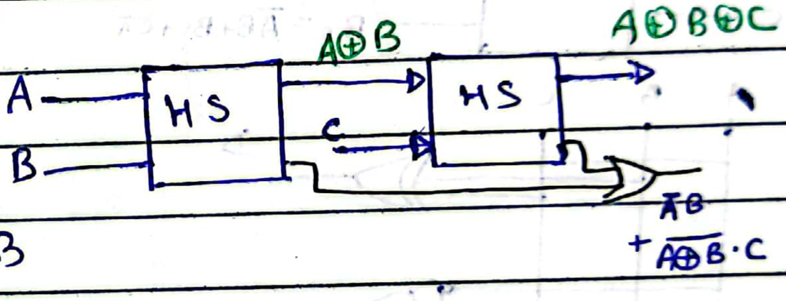
For Sum, D_{sum} = Lv 1 + Lv 2 + Lv 3 + Lv 4 5

For carry, D_{carry} = Lv 1 + Lv 2 + Lv 3

For entire addition = max [D_{sum}, D_{carry}]

= D_{sum}

4) Subtractor:

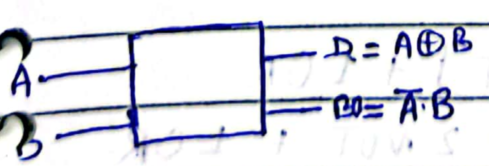


*) Summary:

1) Half subtractor
- Diff = $A \oplus B$
- Borrow = $\bar{A} B$
- Using NAND = 5 gates
- using NOR = 5 gates

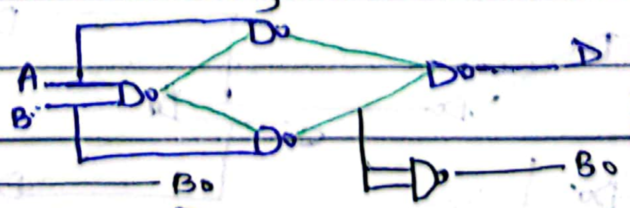
2) Full subtractor
- Diff. = $A \oplus B \oplus C$
- Borrow = $\bar{A}B + BC + C\bar{A}$
- using NAND = 9 gates
- ~~using NOR = 9 gates~~
- using HS =

~~*) Half subtractor :-~~

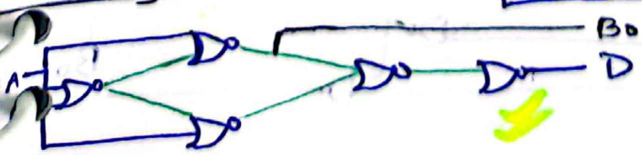


A	B	D	BO
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

using NAND

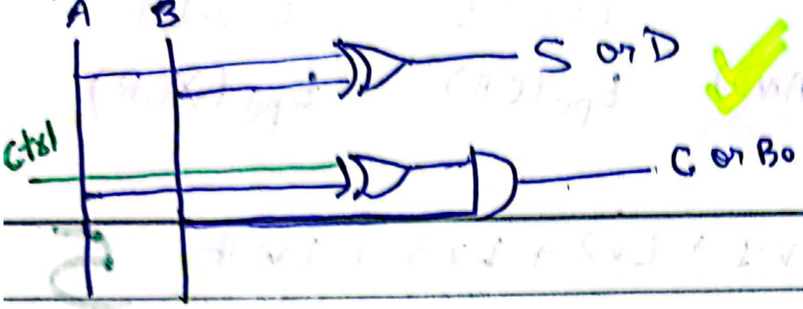


using NOR



*) Half adder/subtractor:-

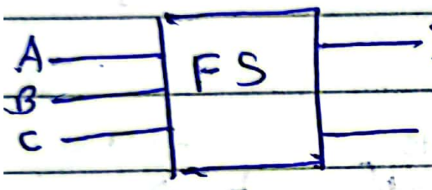
ctrl = 0 ⇒ HA
= 1 ⇒ HS



*) Full Subtractor:

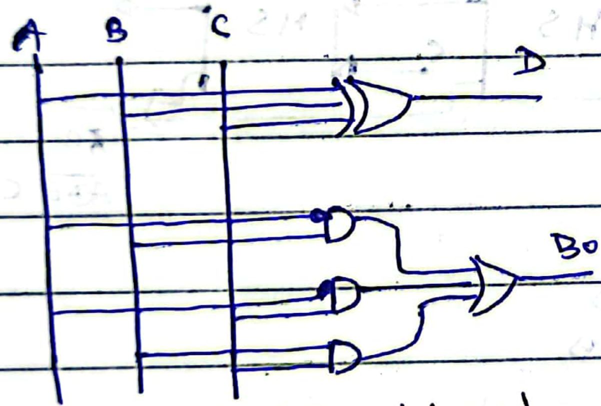
• Ex: For 3 single-bit no.s

A	B	C	D	Bo
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

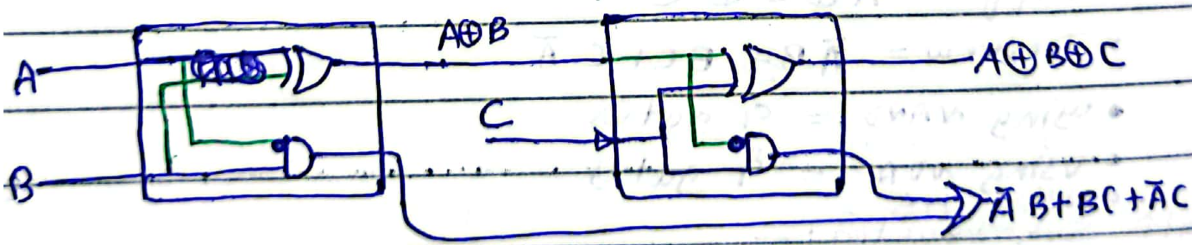


$D = A \oplus B \oplus C$

$B_0 = \bar{A}B + BC + C\bar{A}$

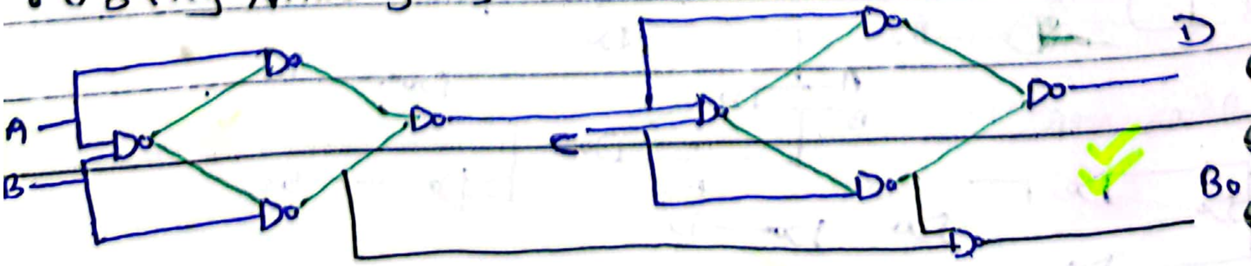


• Using half subtractors

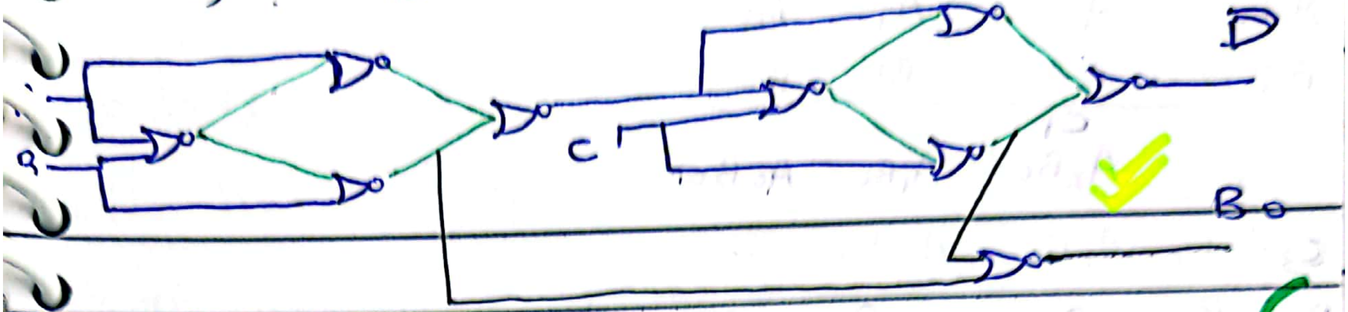


$1\ FS = 2\ HS + 1\ OR$
$= 2\ [XOR + AND + NOT] + 1\ OR$
$= 2\ XOR + 2\ AND + 2\ NOT + 1\ OR$

• Using NAND gates

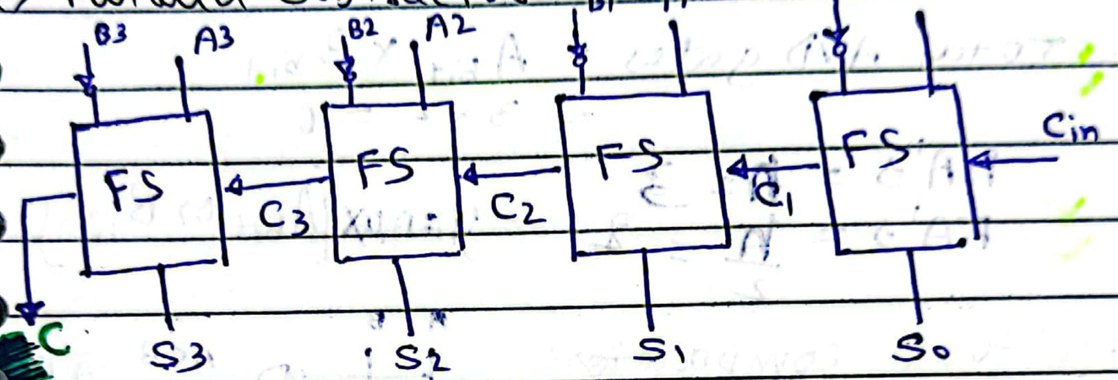


Using NOR gate

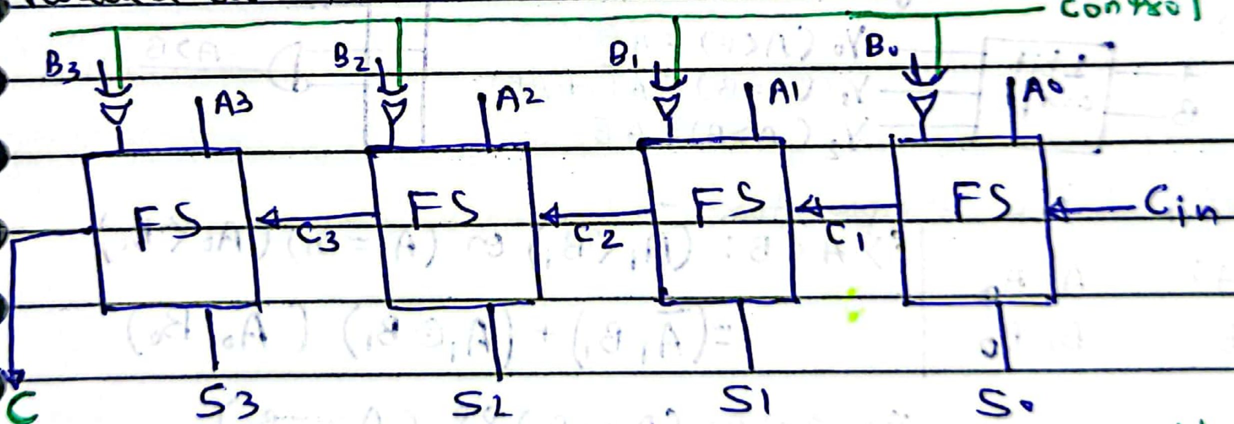


6

Parallel subtractor:



Parallel adder/subtractor:

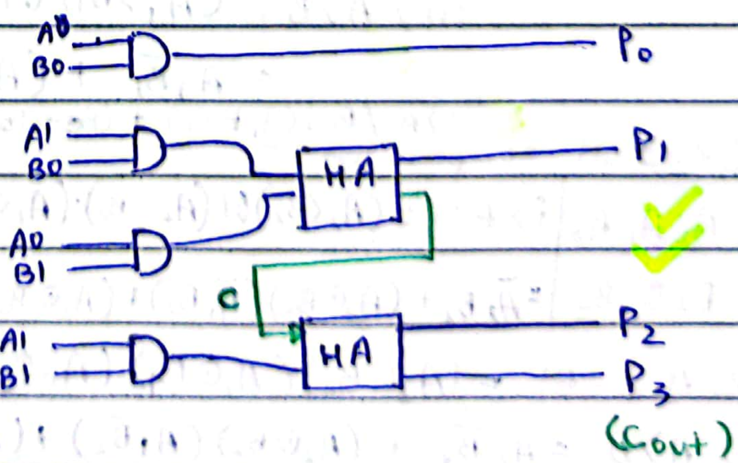


ctrl = 0 → adder
1 → Subtr.

Binary Multiplier:

• Ex:)

A:		A_1	A_0
B:		B_1	B_0
	C	$A_1 . B_0$	$A_0 . B_0$
	$A_1 B_1$	$A_0 . B_1$	0
	P_2	P_1	P_0



• Ex:

A:		A_2	A_1	A_0
B:			B_1	B_0
		C_1		
	C_2	A_2B_0	A_1B_0	A_0B_0
C_3	A_2B_1	A_1B_1	A_0B_1	0
P_4	P_3	P_2	P_1	P_0

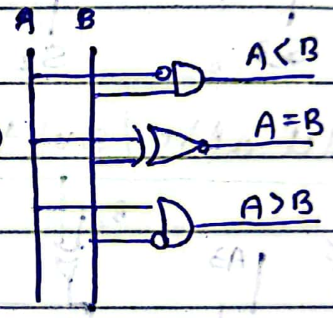
Note: Total AND gates $= A_{bit} \times B_{bit}$
$= 3 \times 2 = 6$

?

HA's $= A = 3$; $n = \max\{A_{bit} \text{ or } B_{bit}\}$

F.A's $= \frac{A}{2} = 2$

#) Magnitude comparator:



$\bar{A}B$

$A \odot B$

$A\bar{B}$

•) Comparison of two no.s

•)

A, B → 1 bit comp.

$Y_0\ (A<B) = \bar{A}B$

$Y_1\ (A=B) = \bar{A}\bar{B} + AB$

$Y_2\ (A>B) = A\bar{B}$

•) Ex: 2 bit comparator:-

A: $A_1 A_0$

B: $B_1 B_0$

i) $A<B$: $(A_1<B_1)$ or $(A_1=B_1)(A_0<B_0)$
$= (\bar{A}_1 B_1) + (A_1 \odot B_1)\cdot(\bar{A}_0 B_0)$

ii) $A==B$: $(A_1==B_1)$ && $(A_0==B_0)$
$= (A_1 \odot B_1)\cdot(A_0 \odot B_0)$

iii) $A>B$: $(A_1>B_1)$ or $(A_1==B_1)\cdot(A_0>B_0)$
$= A_1\bar{B}_1 + (A_1 \odot B_1)\cdot(A_0\bar{B}_0)$

iv) $A \neq B$: $(A_1 \oplus B_1) + (A_0 \oplus B_0)$

•) Ex: 3 bit comparator:-

A: $A_2 A_1 A_0$

B: $B_2 B_1 B_0$

i) $A<B$: $(A_2<B_2)$ or $(A_2==B_2)\cdot(A_1<B_2)$ or $(A_2==B_2)(A_1==B_1)(A_0<B_0)$
$= \bar{A}_2 B_2 + (A_2 \odot B_2)(\bar{A}_1 B_2) + (A_2 \odot B_2)(A_1 \odot B_1)(\bar{A}_0 B_2)$

ii) $A==B$: $= (A_2 \odot B_2)(A_1 \odot B_1)(A_0 \odot B_0)$

iii) $A>B$: $= A_2\bar{B}_2 + (A_2 \odot B_2)(A_1\bar{B}_2) + (A_2 \odot B_2)(A_1 \odot B_1)(A_0\bar{B}_0)$

iv) $A \neq B$: $(A_2 \oplus B_2) + (A_1 \oplus B_1) + (A_0 \oplus B_0)$

For 3 bit :-

7

A (0 to 7)   B (0 to 7)

A < B has $\boxed{\frac{2^{2n} - 2^n}{2}}$ = 28 combinations

A = = B has $\boxed{2^n}$ = 8 combinations

A > B has $\boxed{\frac{2^{2n} - 2^n}{2}}$ = 28 combinations

To construct n-bit comparator using 1-bit comp.

1 bit comparators = n

AND gates = 2n - 1

OR gates = 2

#) Counters < Synchronous / Ripple (Asynchronous)

counts from 0 to $2^n - 1$

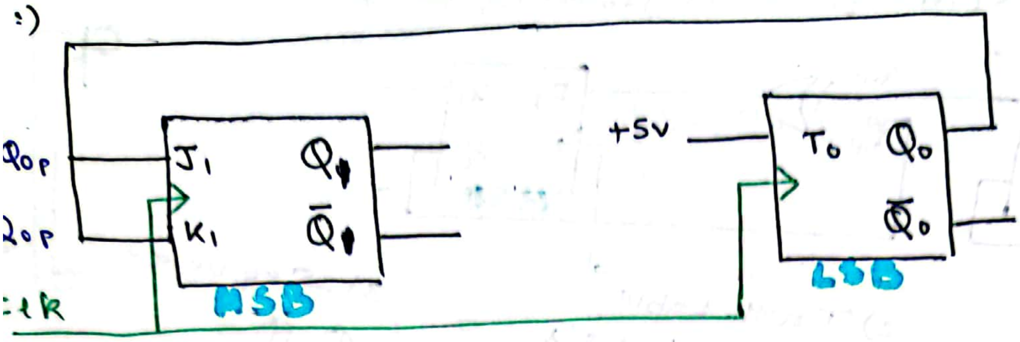
n-bit binary counter ⇒ n-flip flops ⇒ 0 to $2^n - 1$

Synchronous Counters

Q:)

Note:) • The delay formulas may be incorrect when FA's are implemented using HA's.
• In that case, draw logic diagram & calculate delay manually.

8



1) characteristic eq.^{n}s:-

JK: $Q_{n+1} = J\bar{Q}_n + \bar{K}Q_n$

T: $Q_{n+1} = T \oplus Q_n$

$Q_{0N} = 1 \oplus Q_{0P} = \bar{Q}_{0P}$

$Q_{1N} = Q_{0P}\bar{Q}_{1P} + \bar{Q}_{0P}Q_{1P} = Q_{0P} \oplus Q_{1P}$

2 Flip flops
$\Rightarrow 2^2 = 4$
$\Rightarrow$ mod 4 counter

2) Truth table:-

PS's		NS's	
Q_{1P}	Q_{0P}	Q_{1N}	Q_{0N}
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

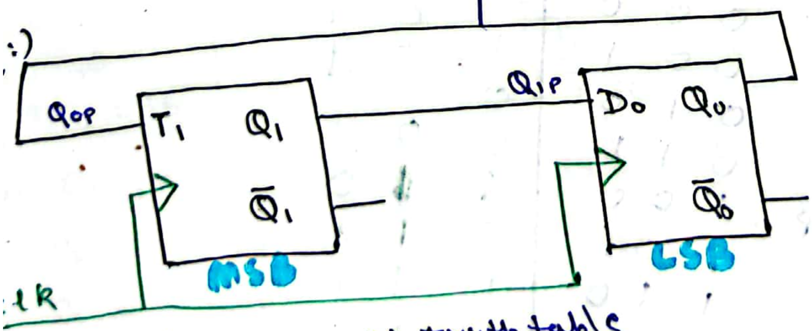
mod 4 counter

3) Sequence

$Q_1 Q_0$

$00 \rightarrow 01 \rightarrow 10 \rightarrow 11 \rightarrow 00$

$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 0$



Char. eq.^{n}s:-

$Q_{n+1} = T \oplus Q_n$

: $Q_{n+1} = D$

$Q_{0N} = Q_{1P}$

$Q_{1N} = Q_{0P} \oplus Q_{1P}$

2) truth table

PS's		NS's	
Q_{1P}	Q_{0P}	Q_{1N}	Q_{0N}
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	1

3) Sequence

$Q_1 Q_0$

$00 \rightarrow 00$

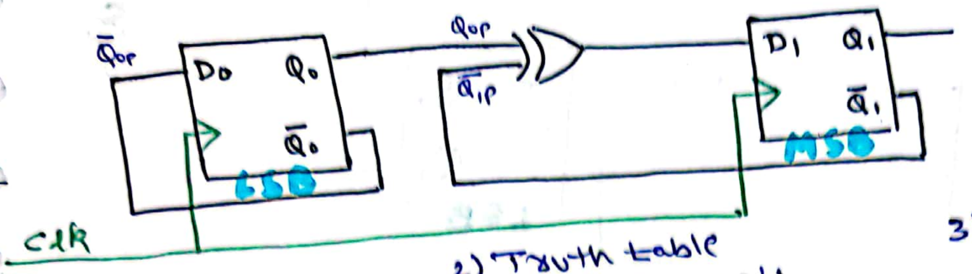
$0 \rightarrow 0$

$Q_1 Q_0$

$01 \rightarrow 10 \rightarrow 11 \rightarrow 01$

$1 \rightarrow 2 \rightarrow 3 \rightarrow 1$

Q:) Each state (in sequence) is stated as $Q_0 Q_1$.



1) char. eqⁿs:-

D: $Q_{n+1} = D$

$Q_{0N} = \overline{Q}_{0P}$

$Q_{1N} = Q_{0P} \oplus \overline{Q}_{1P}$

$= Q_{0P} \cdot Q_{1P} + \overline{Q}_{0P} \cdot \overline{Q}_{1P}$

$= Q_{0P} \odot Q_{1P}$

2) Truth table

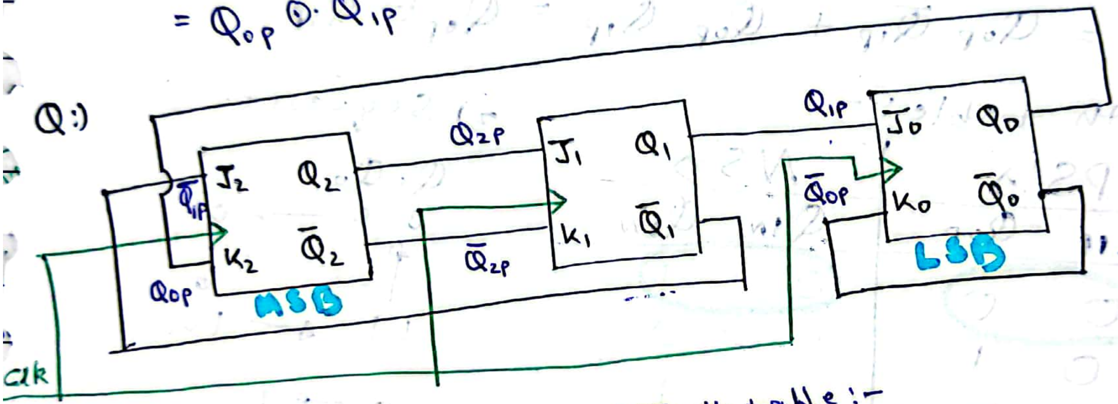
PS's		NS's	
Q_{1P}	Q_{0P}	Q_{1N}	Q_{0N}
0	0	1	1
0	1	0	0
1	0	0	1
1	1	1	0

3) Sequence

$Q_0 Q_1$

$00 \rightarrow 11 \rightarrow 01 \rightarrow 10 \rightarrow 00$

Q:)



1) char eqⁿ:

JK: $Q_{n+1} = J\overline{Q}_n + \overline{K} Q_n$

$Q_{0N} = Q_{1P}\overline{Q}_{0P} + \overline{\overline{Q}}_{0P} \cdot Q_{0P}$

$= Q_{1P} \cdot \overline{Q}_{0P} + Q_{0P}$

$= Q_{1P} + Q_{0P}$

$Q_{1N} = Q_{2P}\overline{Q}_{1P} + Q_{2P} \cdot Q_{1P}$

$= Q_{2P}$

$Q_{2N} = \underbrace{\overline{Q}_{1P}\overline{Q}_{2P} + \overline{Q}_{0P} Q_{2P}}_{1 \text{ if any sum term } 1}$

i) if $\overline{Q}_{1P}\overline{Q}_{2P} = 1$

$\Rightarrow Q_{1P} = 0,\ Q_{2P} = 0$

ii) if $\overline{Q}_{0P} \cdot Q_{2P} = 1$

$\Rightarrow Q_{0P} = 0,\ Q_{2P} = 1$

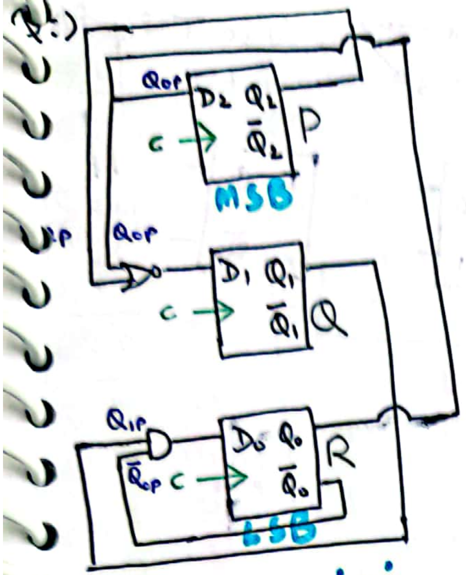
2) Truth table:-

PS's			NS's		
Q_{2P}	Q_{1P}	Q_{0P}	Q_{2N}	Q_{1N}	Q_{0N}
			1	0	0
0	0	0		0	1
0	0	1	1	0	1
0	1	0	0		
0	1	1	0	0	1
1	0	0	1	[illegible]	0
1	0	1	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1

3) sequence:-

$000 \rightarrow 100 \rightarrow 110 \rightarrow 111 \rightarrow 011 \rightarrow 001 \rightarrow 101 \rightarrow 011$

Q:)



T.T:-

PS			NS		
Q_{2P}	Q_{1P}	Q_{0P}	Q_{2N}	Q_{1N}	Q_{0N}
0	0	0	0	1	0
0	0	1	1	0	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

At an instant $\dot{P}Q\dot{R}$ have a value 0 1 0

Sequence:-

$Q_2 Q_1 Q_0$

000 → 010 → 011 → 100 → 000

D: $Q_{n+1} = D$

$Q_{0N} = \underbrace{Q_{1P} \cdot \bar{Q}_{0P}}$

1 if; $Q_{0P} = 0$, $Q_{1P} = 1$

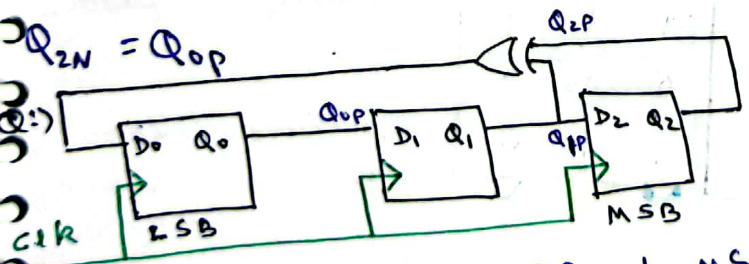
$Q_{1N} = \overline{Q_{0P} + Q_{2P}}$

$= \underbrace{\bar{Q}_{0P} \cdot \bar{Q}_{2P}}$

1 if; $Q_{0P} = 0$, $Q_{2P} = 0$

$Q_{2N} = Q_{0P}$

Q:)



initial state: $Q_2 Q_1 Q_0$
0 0 1

D: $Q_{n+1} = D$

$Q_{0N} = Q_{2P} \oplus Q_{1P}$

$Q_{1N} = Q_{0P}$

$Q_{2N} = Q_{1P}$

PS			NS		
Q_{2P}	Q_{1P}	Q_{0P}	Q_{2N}	Q_{1N}	Q_{0N}
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	1
0	1	1	1	1	1
1	0	0	0	0	1
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	1	1	0

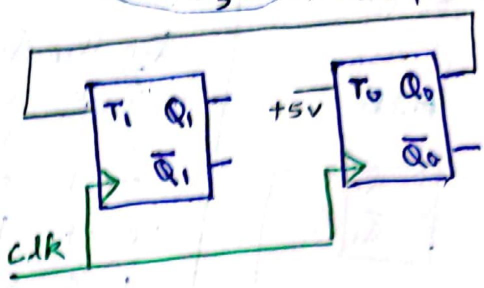
$Q_2 Q_1 Q_0$

001 → 010 → 101 → 011 → 111 → 110 → 100 → 001

*) Sequence ਦਾ counter ਕਰਨਾ

Q:) Design synchronous counter for seq: 0 → 1 → 2 → 3 → 0 ; Use T flip flo... ⇒ mod 4

PS		NS		T	
Q_{1P}	Q_{0P}	Q_{1N}	Q_{0N}	T_1	T_0
0	0	0	1	0	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	0	0	1	1



Mod-4 ⇒ 2 ff's needed

T_1:

Q_{1P} \ Q_{0P}	0	1
0		1
1	0	1

$T_1 = Q_{0P}$

T_0 : $T_0 = 1$

Q:) Seq: 0 → 2 → 4 → 5 → 7 → 6 → 0 ; 1 → 3 → 1 ; use D Flipflops

sequence ⇒ mod 8 ⇒ 3 FF's

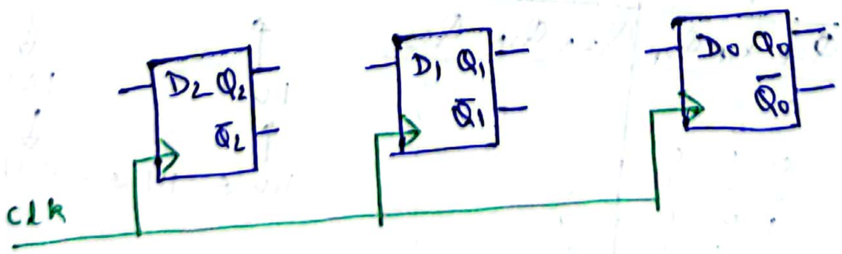
P S			NS			D		
Q_{2P}	Q_{1P}	Q_{0P}	Q_{2N}	Q_{1N}	Q_{0N}	D_2	D_1	D_0
0	0	0	0	1	0	0	1	0
0	0	1	0	1	1	0	1	1
0	1	0	1	0	0	1	0	0
0	1	1	0	0	1	0	0	1
1	0	0	1	0	1	1	0	1
1	0	1	1	1	1	1	1	1
1	1	0	0	0	0	0	0	0
1	1	1	1	1	0	1	1	0

D_2:-

Q_{2P} \ $Q_{1P}Q_{0P}$	00	01	11	10
0				1
1	1	1	1	

$D_2 = \overline{Q}_{2P} Q_{1P} \overline{Q}_{0P} + Q_{2P} Q_{0P} + \overline{Q}_{1P} Q_{2P}$

Proceed like previ...



Q:) 2 bit saturating up-counter.
Synchronous.
Use T FF's

⇒ Seq: 0 → 1 → 2 → 3 ↺ ; 2 FF's there

PS		NS		T	
Q_{1P}	Q_{0P}	Q_{1N}	Q_{0N}	T_1	T_0
0	0	0	1	0	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	0	0

$T_1 = \overline{Q}_{1P} Q_{0P}$

$T_0 = \overline{Q}_{1P} + \overline{Q}_{0P}$

Q_{0P} \ Q_{1P}	0	1
0	1	1
1	1	0

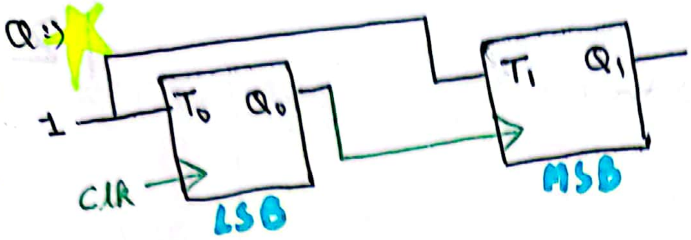
*) Synchronous counter

- All FF's triggered with same clock.
- Faster
- Delay = $T_{FF} + T_{CC}$; T_{FF} = flip flop delay, T_{CC} = comb. circuit delay
- Parallel counter
- Operate in any desired count sequence
- Ex: Ring counter
  Johnson "

Asynchronous (Ripple) counter

- Different FF's triggered with different clock
- Slower
- Delay = $n \times T_{FF} + T_{CC}$
- Serial counter
- Operate only in fixed count sequence (up / down)
- Ex: Ripple up counter
  Ripple down "

*) Asynchronous counters:-



+ve triggering

i) Char. eq^n s:-

T: $Q_{n+1} = T \oplus Q_n$

$Q_{0N} = 1 \oplus Q_{0P} = \overline{Q}_{0P}$

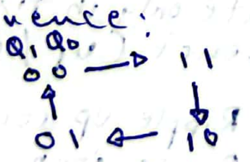
ii) For Q_{1N} observe FF-0:-

Toggle mode (Q_0: 0 → 1)	Latch mode (Memory)
$Q_{n+1} = T \oplus Q_n$ $Q_{1N} = 1 \oplus Q_{1P}$ $= \overline{Q}_{1P}$	$Q_{1N} = Q_{1P}$

iii)

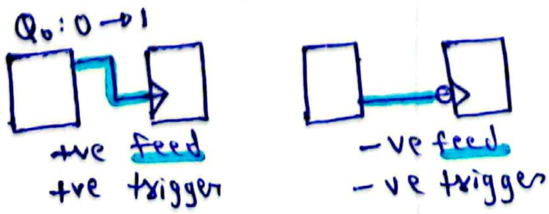
PS		NS	
Q_{1P}	Q_{0P}	Q_{1N}	Q_{0N}
0	0	1	1
0	1	0	0
1	0	0	1
1	1	1	0

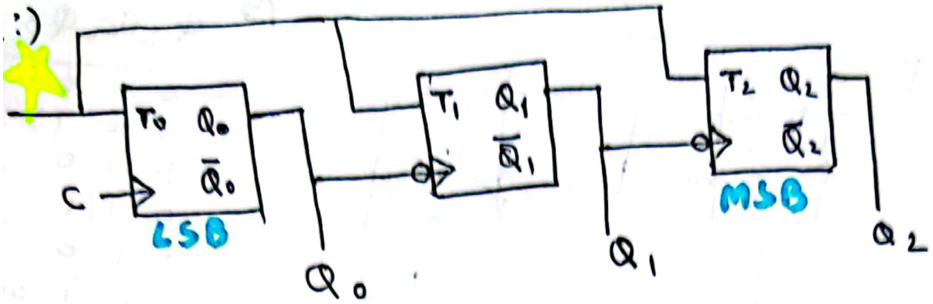
iv) Sequence:



Note:

	+ve Feedback	-ve Feedback
+ve Edge trigger	Down counter	Up counter
-ve Edge trigger	Up counter	Down counter





-ve trigger 14
+ve feed
⇒ Up counter

$T: \; Q_{n+1} = T \oplus Q_n$

$Q_{0N} = 1 \oplus Q_{0P}$
$= \bar{Q}_{0P}$

Now, for ff - 1

TM (Q_0: 1 → 0)	LM (else)
$Q_{1N} = 1 \oplus Q_{1P}$ $= \bar{Q}_{1P}$	$Q_{1N} = Q_{1P}$

for ff - 2

TM (Q_1: 1 → 0)	LM (else)
$Q_{2N} = 1 \oplus Q_{2P}$ $= \bar{Q}_{2P}$	$Q_{2N} = Q_{2P}$

Now,

PS			NS		
Q_{2P}	Q_{1P}	Q_{0P}	Q_{2N}	Q_{1N}	Q_{0N}
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

Now,
Sequence:-

0 → 1 → 2 → 3
↑ ↓
7 ← 6 ← 5 ← 4

Note: • Self starting counter ⇒ Sequence = Connected graph (irrespective of initial state, counter provides counting sequence)

• Free running counter ⇒ Sequence = Perfect cycle (counter is maintaining all possible states in counting sequence)

• Ex:

0 → 3
↑ ↓
1 ← 2

SS✓ FR✓

0 → 1 → 3
↓
2 → 1

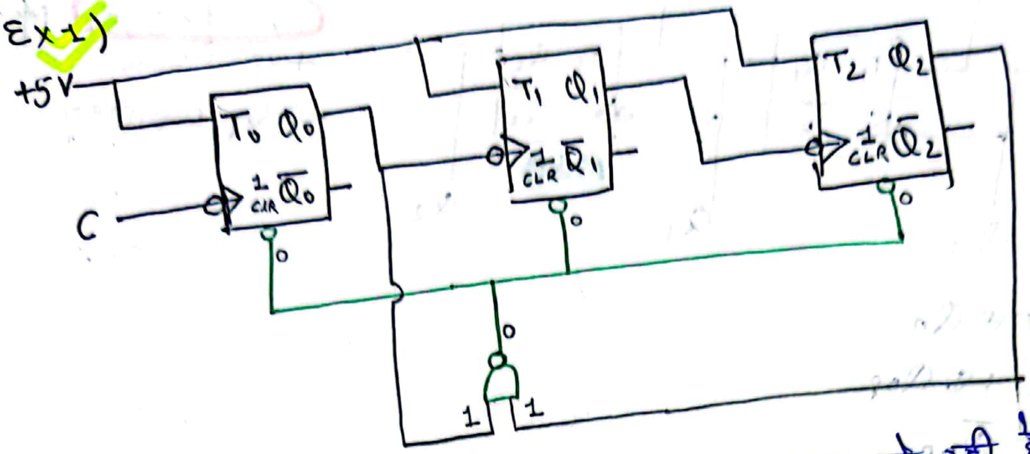
SS✓ FR✗

• SS FR

*) Restricted mod counter:

15

Ex-1)



Q_2	Q_1	Q_0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

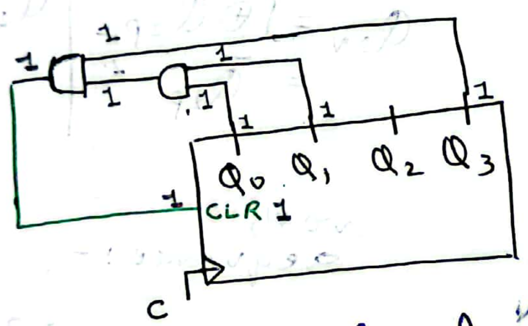
101 → clr will reset here

⇒ 0→1→2→3→4 (4←3)

⇒ mod 5 counter

ये देखो के सारी CLR pins 1 कब हो रही है
Bec. तब counter reset हो जाएगा।

Ex-2)



Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
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
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

1011 → reset

clr = 1; when $Q_0 = 1$ & $Q_1 = 1$ & $Q_3 = 1$

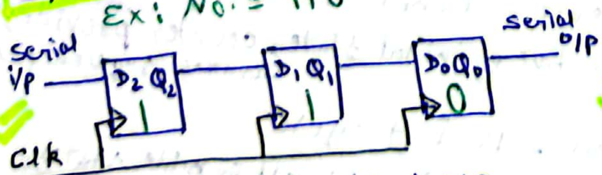
0 → 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 10 → 0

mod-11 upcounter

*) Registers: • storage device designed using FF's (usually D)
• Shift register types/modes: SISO, SIPO, PISO, PIPO

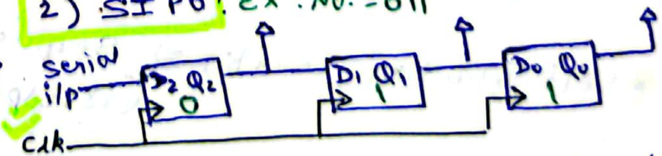
1) SISO:

Ex: No. = 110



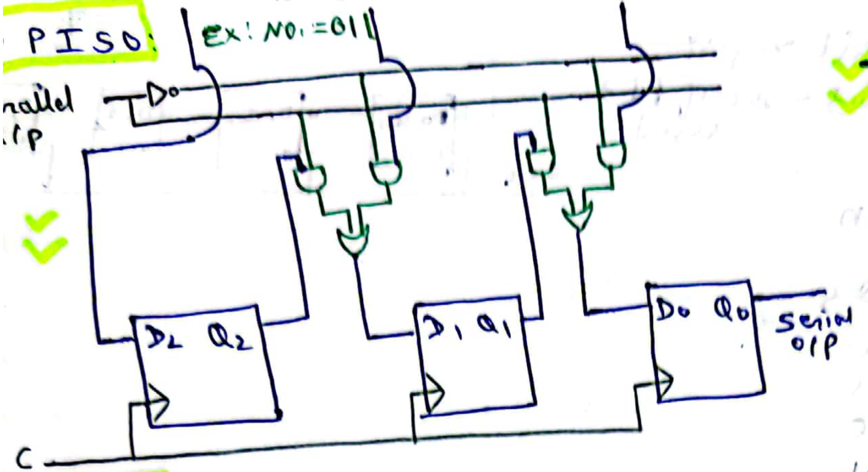
clk	i/p	Q_2	Q_1	Q_0	o/p
1	0	Ø	Ø	Ø	Ø
2	1	0	Ø	Ø	Ø
3	1	1	0	Ø	Ø
4	Ø	1	1	0	0
5	Ø	Ø	1	1	1
6	Ø	Ø	Ø	1	1

2) SIPO: Ex: No. = 011



clk	i/p	Q_2	Q_1	Q_0	O/P$_1$	O/P$_2$	O/P$_3$
		Ø	Ø	Ø	Ø	Ø	Ø
1	1	1	Ø	Ø	1	Ø	Ø
2	1	1	1	Ø	1	1	Ø
3	0	0	1	1	0	1	1

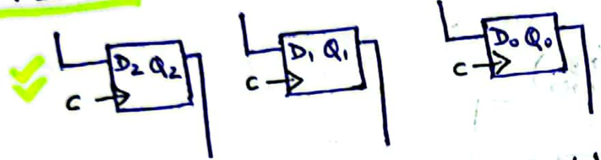
PISO: Ex: No. = 011



c	i/p	Q_2 Q_1 Q_0	o/p
1	1	1 Q Q	= Q
2	1	1 1 Q	= Q
3	0	0 1 1	= 1
4	Q	Q 0 1	= 1
5	Q	Q Q 0	= 0

16

PIPO

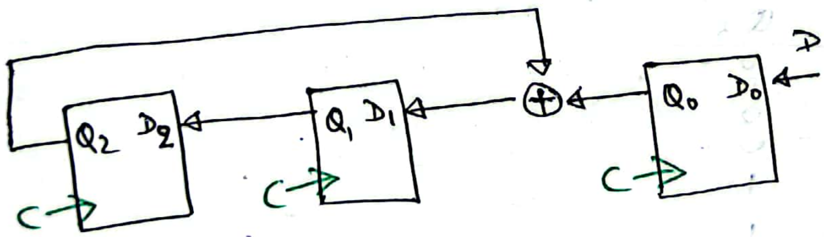


clk	i/p1	i/p2	i/p3	$Q_2 Q_1 Q_0$	o/p1 o/p2 o/p3
1	1	1	0	1 1 0	1 1 0

Note:

	No. of clock (write)	No. of clock (read)	total clock
SISO	n	n−1	2n−1
SIPO	n	0	n
PISO	1	n−1	n
PIPO	1	0	1

:)



Data : 100110000 supplied in 9 clocks find $Q_2Q_1Q_0$ after 9 clocks.

C	Q_{2p}	Q_{1p}	Q_{0p}	D	Q_{2N}	Q_{1N}	Q_{0N}
1	0	0	0	1	0	0	1
2	0	0	1	0	0	1	0
3	0	1	0	0	1	0	0
4	1	0	1	1	0	1	1
5	0	1	1	1	1	1	1
6	1	1	1	0	1	0	0
7	1	0	0	0	0	1	0
8	0	1	0	0	1	0	0
9	1	0	0	0	0	1	0

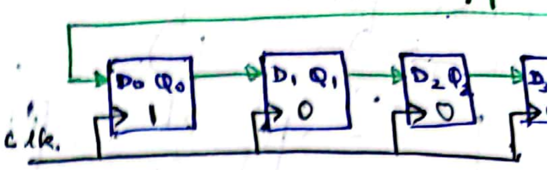
Q_{0N} = Data

$Q_{1N} = Q_{2p} \oplus Q_{0p}$

$Q_{2N} = Q_{1p}$

*) Ring counter:

- It's a circular shift register
- No. of states it has = No. of FF's used

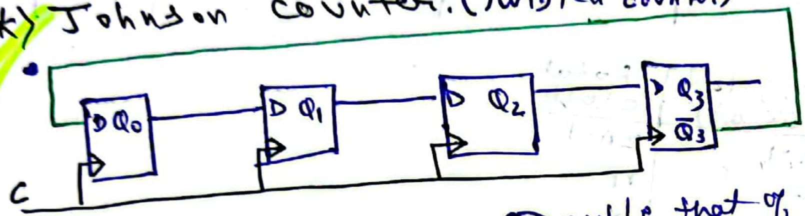


- Used states = n
  Unused states = $2^n - n$

-

clk	Q_0	Q_1	Q_2	Q_3
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
0	1	0	0	0

*) Johnson counter: (Twisted ring counter)

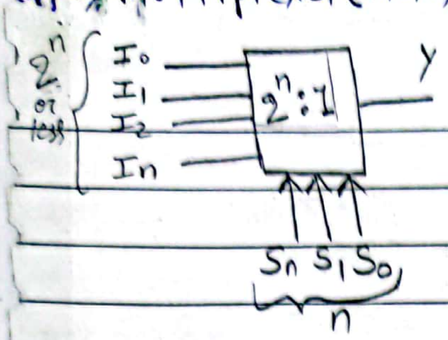
- 

- used states = 2n (Double that of ring counter)
- unused states = $2^n - 2n$

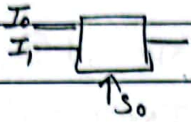
-

C	Q_0	Q_1	Q_2	Q_3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
0	0	0	0	0

#) Multiplexer (Mux)

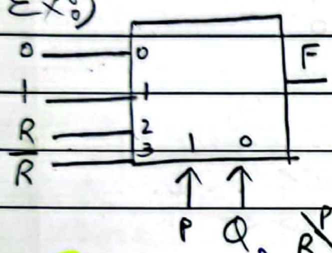


2:1) $y = \bar S_0 I_0 + S_0 I_1$



4:1) $y = E[(\bar S_1 \bar S_0 I_0 + \bar S_1 S_0 I_1 + S_1 \bar S_0 I_2 + S_1 S_0 I_3]$

Ex:)



$y = [\bar P \bar Q 0 + \bar P Q 1 + P \bar Q R + P Q \bar R]$

$= \bar P Q + P \bar Q R + P Q \bar R$

R \ PQ	00	01	11	10
0		1	1	
1		1		1

i.e. Minimize करके बताया करो Ans

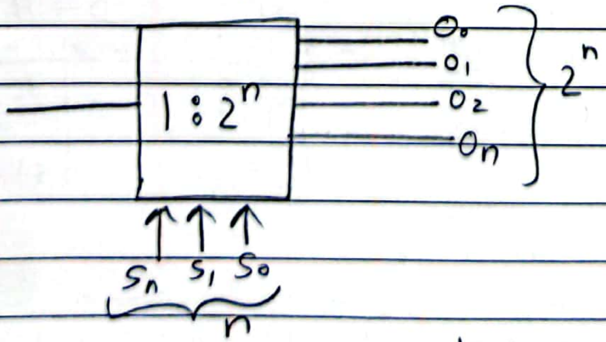
$= Q\bar R + \bar P Q + P \bar Q R$

Ex:) 128×1 from 4×1

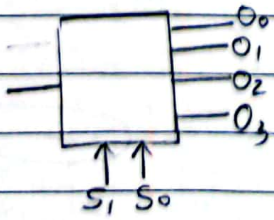
$\frac{\cancel{2}}{\cancel{4}}\ 1 \quad \frac{\cancel{8}}{\cancel{4}}\ 2 \quad \frac{\cancel{32}}{\cancel{4}}\ 8 \quad \frac{\cancel{128}}{\cancel{4}}\ 32$

$1 + 2 + 8 + 32 = 43$ (4×1 mux)

#) De-Multiplexer (DeMux)



1:4)

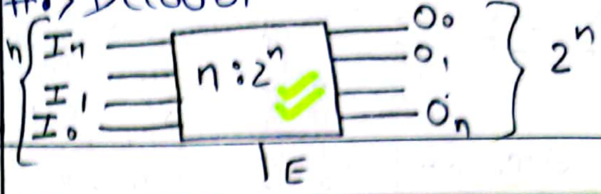


S_1	S_0	O_3	O_2	O_1	O_0
0	0	0	0	0	I
0	1	0	0	I	0
1	0	0	I	0	0
1	1	I	0	0	0

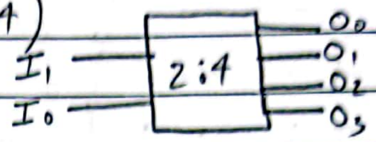
$O_0 = \bar S_1 \bar S_0 I$

$O_1 = \bar S_1 S_0 I$

#) Decoder



2:4)



I_1	I_0	O_3	O_2	O_1	O_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Active high decoder = AND-OR

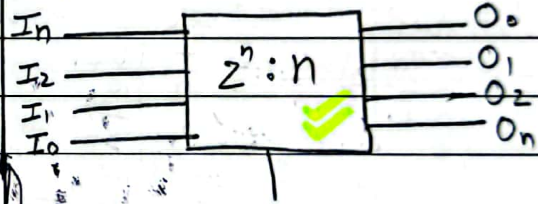
" Low " = NAND-NAND

Ex: 7:128 from 3:8

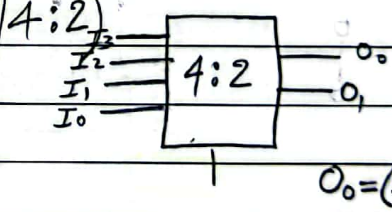
$\frac{\cancel{2}}{\cancel{8}}\ 1 \quad \frac{\cancel{16}}{\cancel{8}}\ 2 \quad \frac{\cancel{128}}{\cancel{8}}\ 16$

$1 + 2 + 16 = 19$

#) Encoder



4:2)



I_3	I_2	I_1	I_0	O_1	O_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

$O_0 = (I_3 \oplus I_1)\cdot \bar I_2 \cdot \bar I_0$

$O_1 = (I_3 \oplus I_2)\cdot \bar I_1 \cdot \bar I_0$

Priority encoder) $I_3 > I_2 > I_1 > I_0$

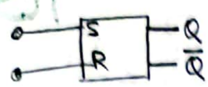
$O_0 = I_3 + \bar I_2 I_1$

$O_1 = I_3 + I_2$

I_3	I_2	I_1	I_0	O_1	O_0
0	0	0	1	0	0
0	0	1	d	0	1
0	1	d	d	1	0
1	d	d	d	1	1

#) Sequential circuits (combinational circuit + memory)

1.) SR:



Eq.^{n}:

$\boxed{Q_{n+1} = S + \bar{R} \cdot Q_n}$

Truth table:

S	R	Q_n	Q_{n+1}
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	x
1	1	1	x

Fx^n table:

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

Excit^n table:

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

2) JK:

Eq.^{n}:

$Q_{n+1} = J\bar{Q}_n + \bar{K} Q_n$

Truth table:

J	K	Q_n	Q_{n+1}
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

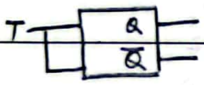
Fx^n table:

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

Excit^n table:

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

3) T:



eq.^{n}:

$Q_{n+1} = T \oplus Q_n$

Truth:

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

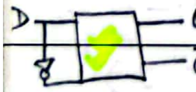
Fx.^{n}:

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

excitat^n:

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

4) D:



Eq.^{n}:

$Q_{n+1} = D$

Truth:

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

Fx.^{n}:

D	Q_{n+1}
0	0
1	1

excit.^{n}:

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

Example) D to T conversion

tt T

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

ex D

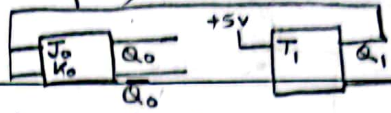
Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

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

$D = T \oplus Q_n$

#> Counters:

Example) Derive transition sequence



1) Equations:

JK $\quad Q_{n+1} = J\bar{Q}_n + \bar{K} Q_n$

T $\quad Q_{n+1} = T \oplus Q_n$

$Q_{0N} = Q_{1P} \cdot \bar{Q}_{0P} + \bar{Q}_{1P} \cdot Q_{0P} = Q_{1P} \oplus Q_{0P}$

$Q_{1N} = 1 \oplus Q_{1P} = \bar{Q}_{1P}$

2) Truth table

Present		Next	
Q_{1P}	Q_{0P}	Q_{1N}	Q_{0N}
0	0	1	0
0	1	1	1
1	0	0	1
1	1	0	0

3) Sequence:

$Q_1 \; Q_0$

00 → 10 → 01 → 11 → 00

0 → 2 → 1 → 3 → 0

Example) Mod-18

$2^4 = 16, \quad 2^5 = 32$

Ans = 5 ff needed

~~i.e. 2 FFlops ⇒ 2^2 = modulo 4~~

Self starting counter = Connected graph

Free running " = Perfect cycle

~~#> Registers~~

#> Number System:

Decimal, Binary, Octal, Hex

D → B

$(117)_{10}$ = 2|117 1, 2|58 0, 2|29 1, 2|14 0, 2|7 1, 2|3 1, 1

$= (1110101)_2$

D → O

= 8|117 5, 8|14 6, 1

$= (165)_8$

D → H

= 16|117 5, 7

$= (75)_{16}$

B → D

$(10101)_2$

$= 1\times2^4 + 1\times2^2 + 1\times2^0$

$= (21)_{10}$

B → O

010 101

2 5

$= (25)_8$

B → H

0001 0101

1 5

$= (15)_{16} = (F)_{16}$

O → D

$(345)_8$

$3\times8^2 + 4\times8^1 + 5\times8^0$

$= (229)_{10}$

O → B

3 4 5

011 100 101

$(11100101)_2$

O → H

3 4 5

011 100 101

E 5

$(E5)_{16}$

O → B → H

H → D, H → B, H → O

H → D	H → B	H → O
$(AE5)_{16}$	A E 5	A E 5
$16^2\times10+16^1\times14+16^0\times5$	1010 1110 0101	1010 1110 0101
$=(2789)_{10}$	$(101011100101)_2$	5 3 4 5
		$(5345)_8$

→ 2's, 2's →

# → 2's		2's → #	
−7	+17	1011	01101
+7 = 0111	= 010001	= −8+2+1	= +13
1's = 1000		= −5	
2's = 1001			

→ 1's, 1's →

# → 1's		1's → #	
−12	+20	10111	01001
+12 = 01100	= 010100	01000	= +9
1's = 10011		= −8	

1's & 2's comp. of +ve / −ve

No. Extension: 0111 → 0000 0111 ; 1011 → 1111 1011

Sign magnitude Example: −7 = 1111

Floating pt. rep^n:

Example: −21.75 (s=1, K=7, m=8)

sign	exponent	mantissa
1	1000101	10101110
s=1	K=7	m=8

$(C5AE)_{16}$

I) $(21.75)_{10} = (10101.11)_2$

$= 0.1010111 \times 2^{5}$ (5 → e, mantissa m)

II) Bias $= 2^{K-1} = 2^{7-1} = 64$

BE $= 64+5 = (69)_{10} = (1000101)_2$

Note: 0.M → Explicit (Default in normal)

1.M → Implicit (Default in 754 and BCD)

IEEE 754:

	S	K	M
Half Precision	1	5	10
Single "	1	8	23
Double "	1	11	52

Note:

	S	E	M
+0	0	0	0
−0	1	0	0
+∞	0	255	0
−∞	1	255	0
NAN	0/1	255	≠0
Fraction	0/1	0	≠0

Example: IEEE 754)

→ 40.1

1	10000100	01000 00011 0011 0011 00110
s=1	K=8	m=23

C2206666

$(40.1)_{10} = (101000.000110011 00)_2$

$= 1.01000\,000110011 00 \times 2^5$

Bias $= 2^{K-1}-1 = 2^{8-1}-1 = 127$

B.E $= 127+5 = (132)_{10} = (10000100)_2$

Note:) $(\leftarrow n_1 \text{ bits} \rightarrow)_{x_1} = (\leftarrow n_2 \text{ bits} \rightarrow)_{x_2}$

$x_1^{n_1} = x_2^{n_2}$

Note:) Ranges:-

1's compl.:) $-2^{n-1}-1$ to $2^{n-1}-1 = 2^n-1$

2's compl.:) -2^{n-1} to $2^{n-1}-1 = 2^n$

Sign magnitude:) $-2^{n-1}-1$ to $2^{n-1}-1 = 2^n-1$

Note:) If P = 101 then $8P = 2^3 \times P = 101000$

Note:) Sign magnitude, example –

−7 = 1111

Number extension in sign magnitude

8bit	8bit
1101	011101
10000101	00011101

Note:) If base is 16, we write things like

$(-1)^s (0.M) \times 16^{E-\text{bias}}$

Note:) Condition for overflow during addition of two no.'s in 2's complement form :-

(+A) + (+B) = −C (or) i.e. for MSB $C_{in} \oplus C_{out} = 1$

(−A) + (−B) = +C

Note:) when two 4-bit no.'s multiplied

$$
\begin{array}{cccccccc}
 & & & & a_3 & a_2 & a_1 & a_0 \\
 & & & \times & b_3 & b_2 & b_1 & b_0 \\
\hline
 & & & & a_3b_0 & a_2b_0 & a_1b_0 & a_0b_0 \\
 & & & a_3b_1 & a_2b_1 & a_1b_1 & a_0b_1 & X \\
 & & a_3b_2 & a_2b_2 & a_1b_2 & a_0b_2 & X & X \\
 & a_3b_3 & a_2b_3 & a_1b_3 & a_0b_3 & X & X & X \\
\hline
C_7 & C_6 & C_5 & C_4 & C_3 & C_2 & C_1 & C_0
\end{array}
$$

XOR

$C_1 = a_0b_1 \oplus a_1b_0$

Note:) 8085 MP uses 2's complement rep.^{n} for no.'s, where first bit is sign & rest is 2's comp. no.

Note:) Prime implicant => Biggest possible grouping
Implicant => Any possible grouping

#) Binary codes:

① Grey code :-

Decimal → Gray

$(123)_{10} = (1111011)_2$

1 ⊕ 1 1 1 0 1 1

↓

1 0 0 0 1 1 0

~~Gray → Decimal~~

~~11100~~

1 1 1 0 0

↓ ⊕

1 0 1 1 1

= 23

② BCD code :-

Decimal → BCD

30 → 3 0

0011 0000

$(00110000)_{BCD}$

BCD → Decimal

0101 1001 → $(59)_{10}$

5 9

Note:) $(1101)_{BCD}$
=> 13
=> invalid BCD
∵ 1 to 9 you can have

Note:) BCD adder adds 6 to correct invalid BCD i.e. 1101

0110

1 0011

Carry

i.e. (0001 0011)

③ Excess-3 :-

Decimal → Excess-3

683 → 6 8 3

+3 +3 +3

9 11 6

$(1001\ 1011\ 0110)_{XS3}$

Excess-3 → Decimal

0101 1001

5 9

-3 -3

2 6

~~$(26)_{10}$~~ ×

Note:) Static hazard free

Note: Q:) 9's complement of the given number $(122100)_3$ when it is expressed with the base of 9? 2]

$(122100)_3 \to (?)_9$

$r_1 = 3,\quad r_2 = r_1^m = 3^2$

=> grouping of 2

$\boxed{1\ 2}\ \boxed{2\ 1}\ \boxed{0\ 0}$

$(5\ 7\ 0)_9$

Now, Directly you can't find 9's complement

i) 8's complement

888
570
318

ii) 9's complement

318
+ 1
320

∵ Base 9 = {0 to 8}

so 8+1 = 9 = 0 & 1 carry

=> $\boxed{320}$

Q:) no. of unused states = ?

Seq: 0 → 4 → 7 → 8 → 6 → 2

A:) Max.^{m} no. = 8

=> 4 FF's needed [∵ 0 to 15]

unused states = 16 − 6 = 10

Q:) 5 bit Down counter. Present state is 01010, how many clocks to reach 10100?

$(01010)_2 \rightsquigarrow (10100)_2$

$(10)_{10} \rightsquigarrow (20)_{10}$

0 ⇐ 0 0 0 0 0

31 ⇐ 1 1 1 1 1

$10 \rightsquigarrow 1 \to 0 \to 31 \rightsquigarrow 20$

9, 1, 1, 11

=> 22 clocks

Note: $(\bar{a} \oplus b)' = (a \oplus b)$

$(a \oplus b) = (a \odot \bar{b})$

$(\bar{a} \oplus b \oplus c)' = a \odot b \odot c$

$(\bar{a} \oplus \bar{b} \oplus c)' = a \oplus b \oplus c$

Note:) Min.^{m} no. of 2 i/p NOR gates required

$= (3n - 3)$ − # of variables which are with complement

n → no. of literals in the product

Ex:) $f(A,B,C,D) = \bar{A}\,B\,\bar{C}\,D$

=> $(3 \times 4 - 3) - 2$

=> 7 NOR gates

#> Important points:

*> Adder:

② → carry lookahead adder delay when fan in is 2 ⇒ $O(\lg n)$

① → Ripple carry adder delay when fan in is 2 ⇒ $O(n)$

③ → Multiplier delay when n-bit no. is fed ⇒ $O(n)$

*> Binary Codes:

→ Hamming distance = No of mismatches occuring at the same corresponding positions in two strings

like: 0 0 0 0 0
0 1 0 1 1 ⇒ HDs = 3

→ If more than 2 strings there then HDs = min (All hamming distances possible)

→ HDs = 2d+1 ; d = max.^{m} no. of erroneous bits that can be corrected by code.

*> Number rep.^{n}:

→ In binary adder, while we are Adding unsigned binary no.'s in 2's complement representation

$$\begin{array}{rccc} & a_{n-1} & \cdots & a_0 \\ & b_{n-1} & \cdots & b_0 \\ \hline C_{out} & C_{n-1} & \cdots & C_0 \\ \hline \end{array}$$

overflow cond.^{n} ⇒ $a_{n-1} b_{n-1} \overline{C_{n-1}} + \overline{a_{n-1}}\, \overline{b_{n-1}}\, C_{n-1} = 1$

→ In ripple carry adder, while adding binary no.s in 2's complement rep.^{n}

$$\begin{array}{rccc} & A_7 & \cdots & A_0 \\ & B_7 & \cdots & B_0 \\ \hline \text{carrys} \to & C_7 & \cdots & C_0 \\ \text{sums} \to & S_7 & \cdots & S_0 \\ \hline \end{array}$$

overflow cond.^{n} ⇒ $A_7 \cdot B_7 \cdot \overline{S_7} + \overline{A_7} \cdot \overline{B_7} \cdot S_7 = 1$