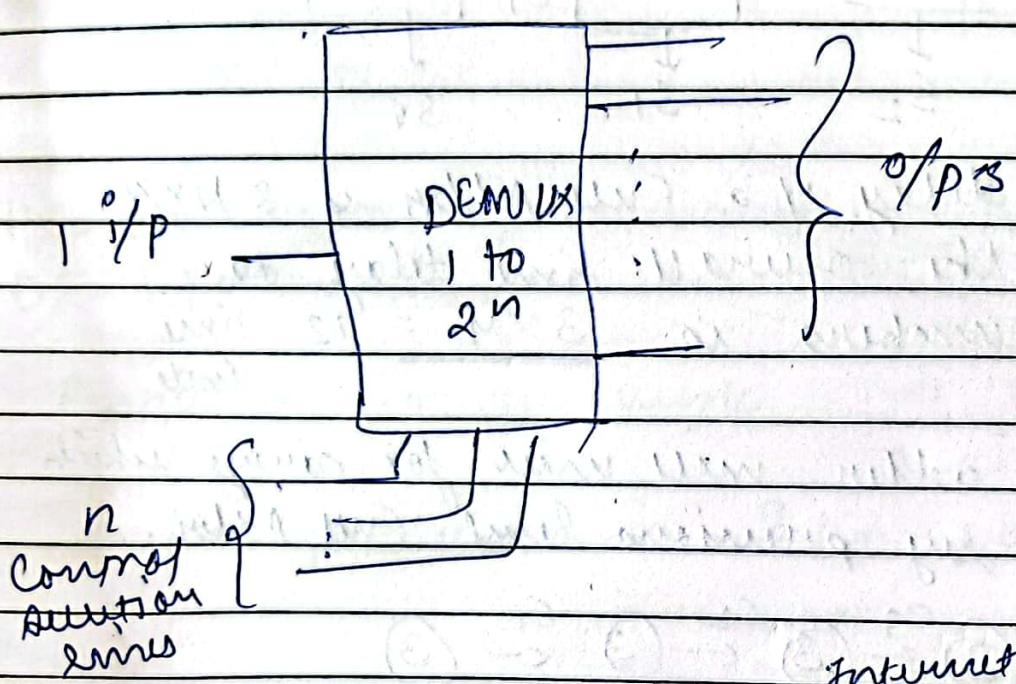
DEMULTIPLEXER : It is a combinational circuit with 1 i/p to 2^n o/ps for Boolean f^n of $(n+1)$ variables. ~~without~~ with n control selection lines.
It is a reverse operation of MUX.



At Sender end
↓
we use MUX

At Receiver end
↓
we use DEMUX.

Internet, Intranet, ~~Broadcast~~ computing use these communication

PARALLEL ADDER :
It is a combinational circuit that adds two n-bit binary numbers

Let $A = A_3A_2A_1A_0$ & $B = B_3B_2B_1B_0$ be 2 4bit binary numbers. Cy

		C_3	C_2	C_1	C_0
A	=	A_3	A_2	A_1	A_0
+ B	=	+ B_3	B_2	B_1	B_0
					S_0

A_0 & B_0 are added by Half adder

But can't take both as ICs are of only 1 type.

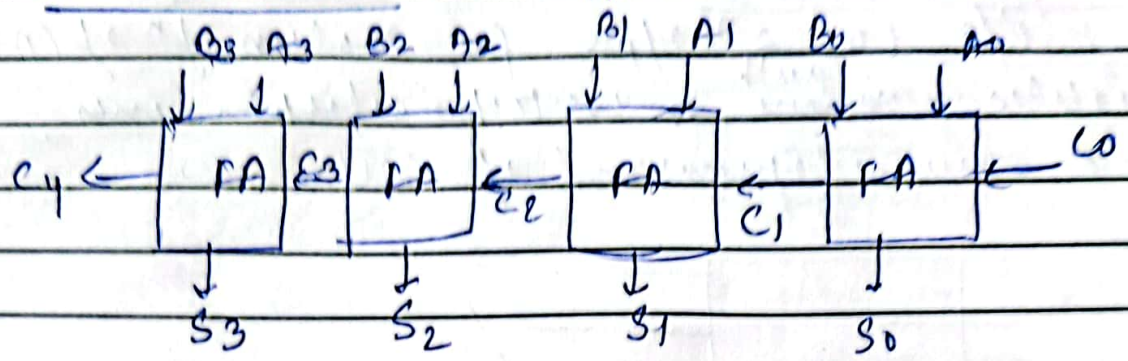
∴ We take either all Full adder or Half adder

As for each after that we need full adder

Date: / / Page no:

1) All Full adder

C_0 = initial carry
$C_0 = 0$



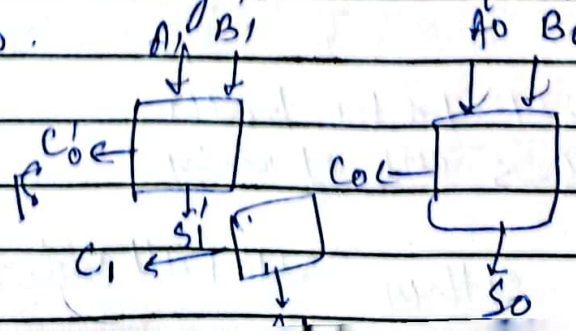
Let the time delay of a Full Adder is 3 time units. then the overall time delay to add 4 bit numbers is $3 \times 4 = 12$ time units

Every next full adder will wait for carry which is generated by previous level Full Adder.

Name is Parallel Adder

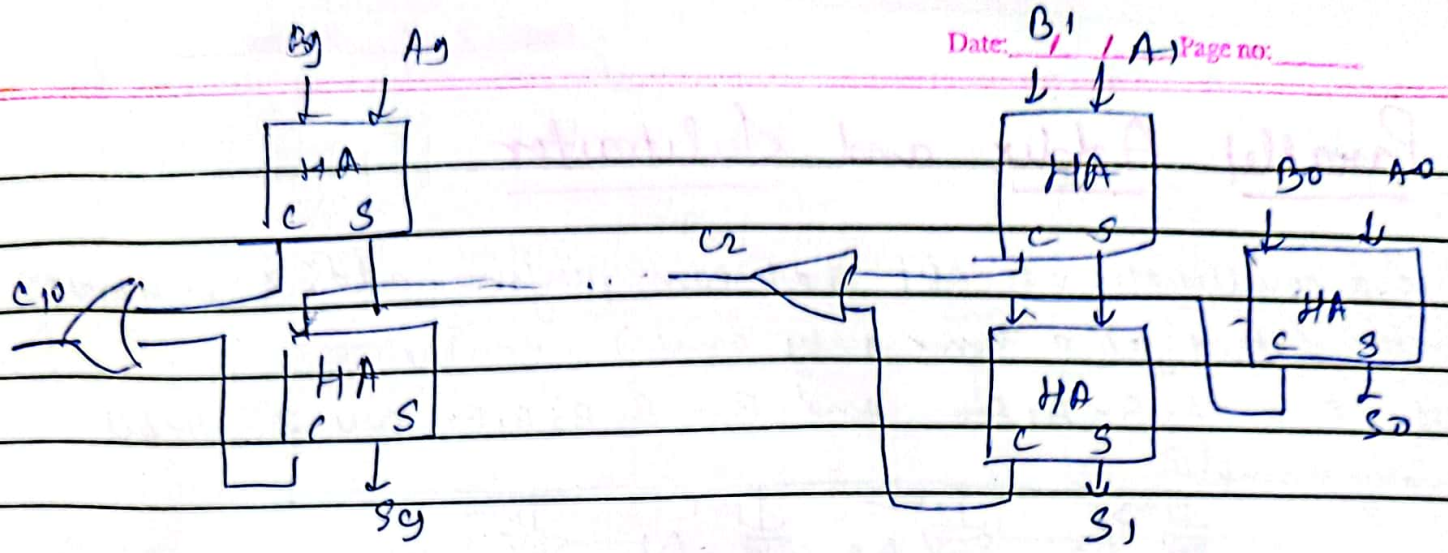
But its processing is Cascading

If parallel adder ckt is designed with the help of Half Adders and OR gates The time delay of a Half Adder and an OR gate is 2 & 1 nanosecondes respectively. Find the overall time delay to add two 10 bit binary numbers.



5
5
4s
48 s

Each ckt can be pipelined



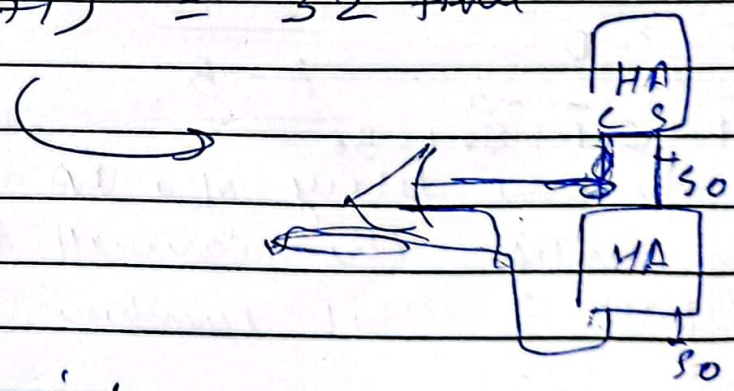
HA = 2 time units
OR = 1 time units

~~17 time units~~

Particular circuits work → when all i/ps available

All upper work in 2 time units

$2 + 10 * (2+1) = 32$ time units

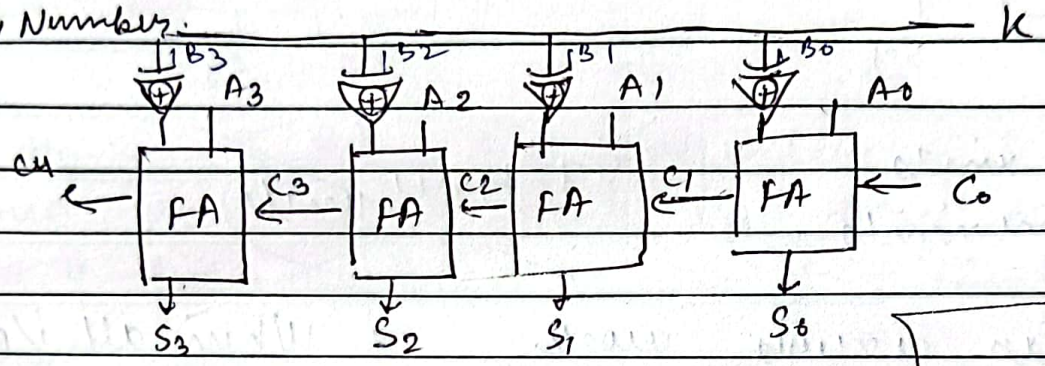


Upper ckt will save 3 time units

Parallel Adder and Subtractor

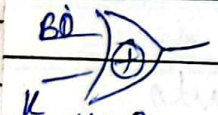
It is a combinational ckt that can perform add^n & subtract^n operation (but at a time only one)

Let $A = A_3 A_2 A_1 A_0$ and $B = B_3 B_2 B_1 B_0$ are 2 4-bit binary Numbers.



$K=0$

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



$K=0$

$B_0 \cdot \bar{0} + \bar{B}_0 \cdot 0$

$= B_0$

$K=1$ $B_0 \cdot 1 + \bar{B}_0 \cdot 1 = \bar{B}_0$

A
$+ \bar{B}$ } 2's complement
$+ 1$

$A - B$

K	C_0	O/P	
0	0	$A+B$	← ~~Full~~ adder
0	1	$A+B+1$	
1	0	$A+\bar{B}$	
1	1	$A+\bar{B}+1$	← ~~Full~~ Subtractor

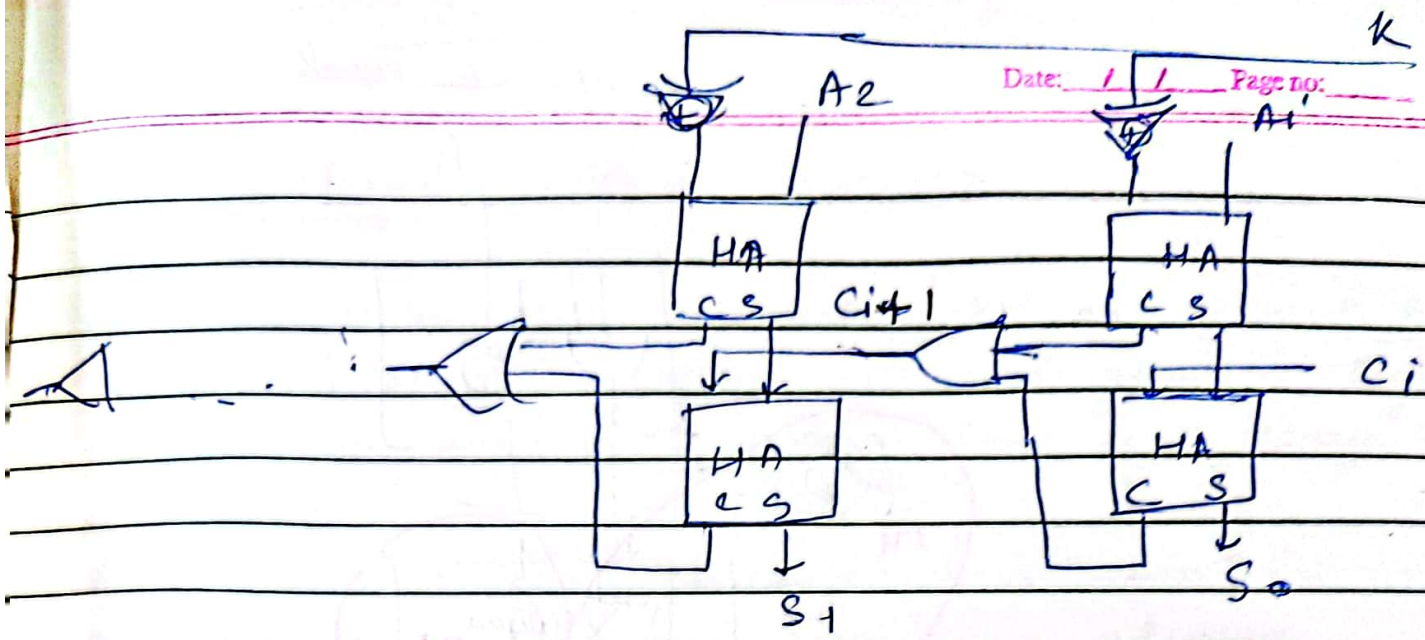
Q) Let the time delay of a FA and an XOR gate is 3 & 2 time units. The overall time delay to add or subtract 2 4bit numbers.

Sol:

~~14~~ XOR → all at once 2 (tu)

Half Adders → $4 \times 3 = 12$ (tu) $= 14$ (tu)

Q) A parallel adder & subtractor ckt is designed with the help of XOR gates, half adders & OR gates. The time delay for a HALF Adder is 2 time units, XOR gate, AND, & OR gate is 2, 2, 1 time units respectively. Find the overall time delay to add 2 Hundred bit binary No.



XOR → 2 = 2

HA (upper) → 2 = 2

OR → 100 × 3 = 300

& HA

300 time units

~~700 tu~~

304

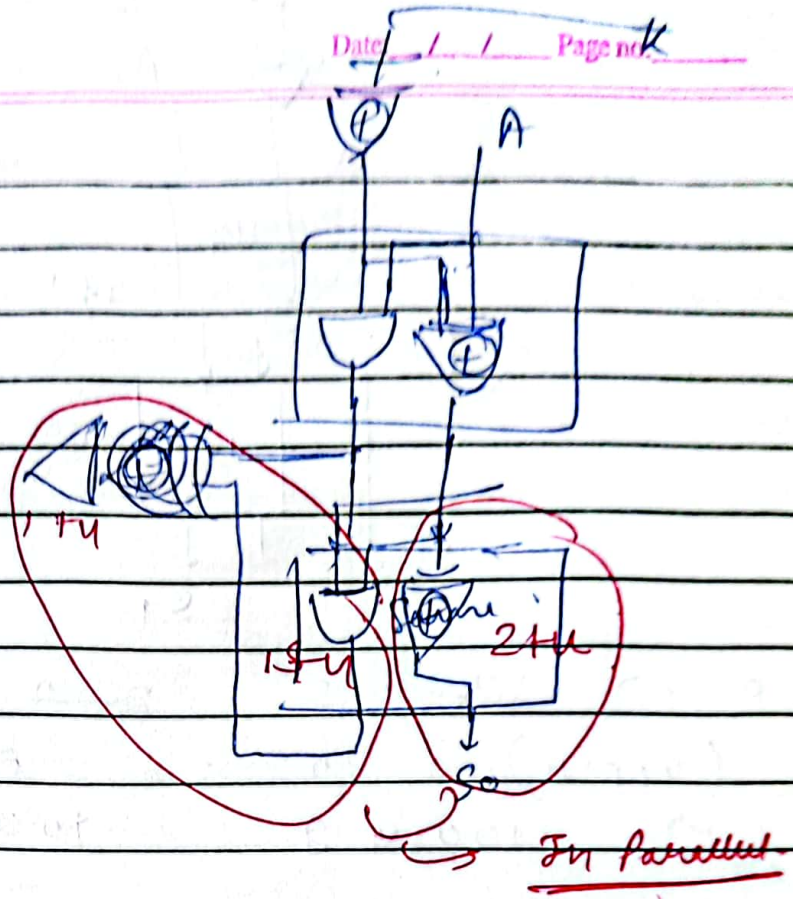
NOTE:

Any ckt will work if both the i/p's of gate are available.

All XOR works in parallel

Q) A parallel adder & subtractor ckt is designed with the help of XOR gates, AND gates & OR gates. The Time delay for an XOR gate, AND gate & OR gate is 2, 1, 1 time units respectively. Find the overall time delay to subtract two 50 bit Binary No.

Sol:



Q XOR → 2

HA upper → max (2, 1) = 2
XOR AND

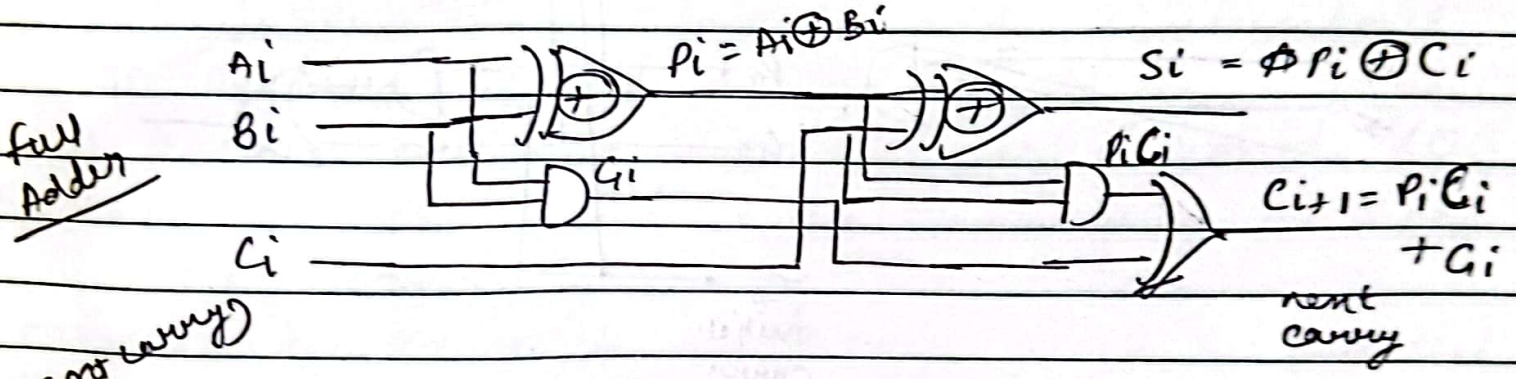
50 × 2 = ~~150~~ 100

104 time units

Lookahead Carry Generator

The parallel adder & subtractor ckt, every next full adder waits for a carry which is generated by previous full adder. It means there is a carry dependency.

A lookahead carry generator is a comb^n circuit that generates all the required carry in advance.



(next carry)

$\hookrightarrow C_{i+1} = P_i C_i + G_i$

$C_1 = P_0 C_0 + G_0$ [C_0 = initial carry]

$C_2 = P_1 C_1 + G_1$

$\Rightarrow C_2 = P_1(P_0 C_0 + G_0) + G_1$

$C_3 = P_2 C_2 + G_2$

$\Rightarrow C_3 = P_2(P_1(P_0 C_0 + G_0) + G_1) + G_2$

$= P_2(P_1 P_0 C_0 + P_1 G_0 + G_1) + G_2$

$\Rightarrow C_3 = \underset{3}{P_2 \cdot P_1 \cdot P_0 \cdot C_0} + \underset{2}{P_2 \cdot P_1 \cdot G_0} + \underset{1}{P_2 \cdot G_1} + \underset{0}{G_2}$

$$\boxed{\begin{aligned} S_i &= P_i \oplus C_i \\ P_i &= A_i \oplus B_i \\ G_i &= A_i \cdot B_i \end{aligned}}$$

$i = 3$

$0 + 1 + 2 + 3 = 3 \times \frac{(3+1)}{2} = 6$

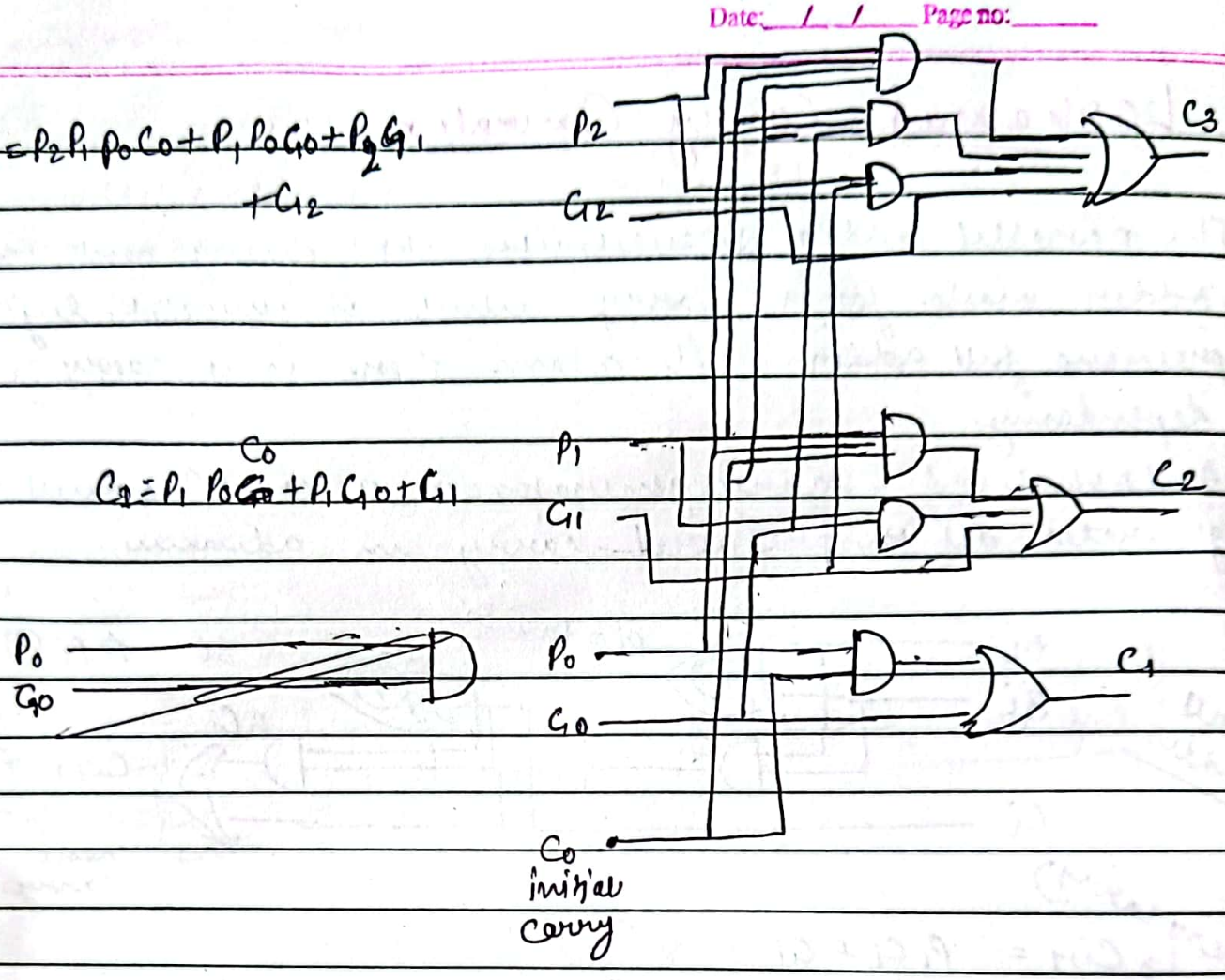
To generate any carry C_i

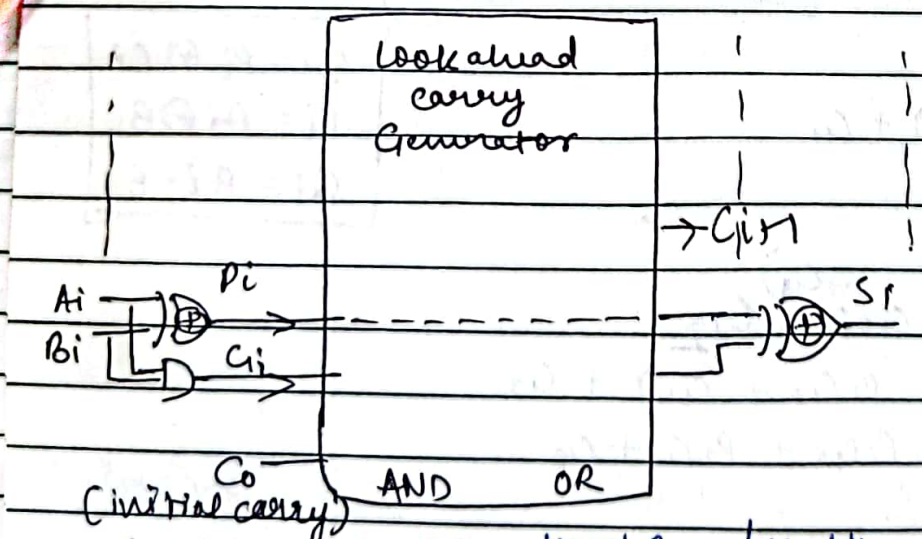
we require i ~~(i+1)/2~~ AND Gates and 1 OR gate

Date: / / Page no:

$C_3 = P_2P_1P_0C_0 + P_1P_0G_0 + P_2G_1 + G_2$

$C_2 = P_1P_0C_0 + P_1G_0 + G_1$





Let the delay of an XOR gate, OR gate & AND gate is 2, 1 & 1 time units. Find time delay.

Level-1	Level-2	Level-3	Level 4	
max (XOR, OR) = max(2,1) = 2	1 +	1 +	2 +	= 6 time units

(Parallelly work)

For 4 / 100 / 1000 bits number Then too 6 time units

07.11.202..
Monday

Date: / / Page no:

To produce C_i (any random carrys, we require i AND Gates & 1 OR Gate

Cost of lookahead carry generator is more.

Q) A parallel adder circuit is designed with the help of full adders (XOR, AND, OR) The time delay for full adders is 3 time units. Find the overall time delay to add 2 hundred bit binary no. with the circuit.
A look ahead carry ~~ahead~~ generator is designed with XOR, AND & OR gate. The time delay for XOR, AND, OR gate is 2, 1, 1 time units respectively. compare the delay with the parallel adder ckt for addition of ~~200~~ 2 hundred bits binary no.
Also compare the cost of lookahead carry generator with Parallel adder when the cost of XOR, AND, OR gate is ₹2, ₹1, ₹1 respectively.

Sol:- Time delay of parallel Adder ckt =
$3 \times 100 = 300$ time units

Time delay of lookahead carry generator ckt = 6 time units

$$\boxed{t_{lcg} << t_{pa}}$$

Date: / / Page no:

Cost Comparision

No. of Parallel Adder Ckt

1. XOR-gate = 200
2. OR-gate = 100
3. AND-gates = 200

Cost = 200 × 2 + 100 × 1 + 200 × 1

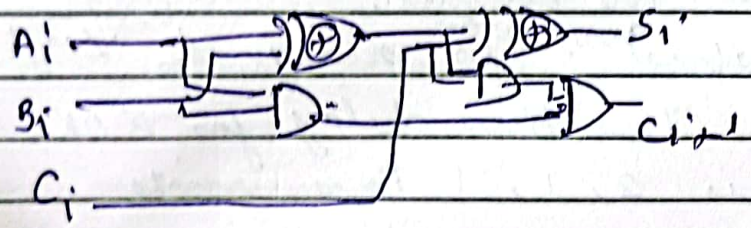
⇒ Cost = ₹ 700

Look ahead Carry Adder ckt

1. XOR-gate = 200
2. OR-gate = 100
3. AND gate = (1 + 2 + 3 + 100) + 100 = 5150

Cost = 200 × 2 + 100 + 5150 × 1

⇒ Cost = ₹ 5650



$$\boxed{t_{LCA} >> t_{PA}}$$

07.11.2022
monday.

(IMP)

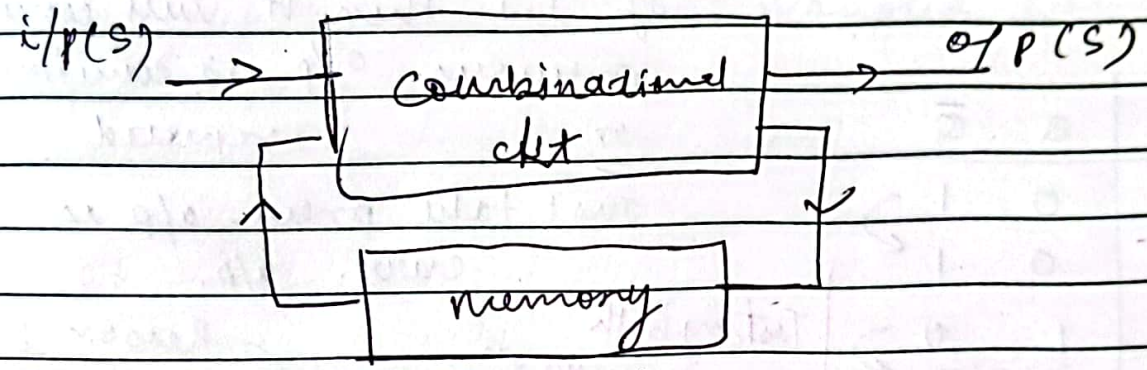
Date: / / Page:

→ pattern
→ one after another (cascade)

SEQUENTIAL CIRCUIT

"memory" is required to remember the sequence

Counting Numbers are also sequence (we get upon 4 after by ~~mem~~ memory (our brain)



Sequential ckt is a combn of combination ckt & memory element.

Memory: It is used to hold information (data)

RAM	Optical Disk { CD, DVD
ROM	Flash Drive (pendrive)
HDD	Magnetic disk / tape / floppy disk
SSD	

The basic memory element is flip flop that stores '1' bit data.

Flip flop can be designed by 2 structures:
(i) cross coupled NAND
(ii) cross coupled NOR

Q) Why only NAND and NOR
Sol: NAND and NOR → Universal Logic gates (ULG)

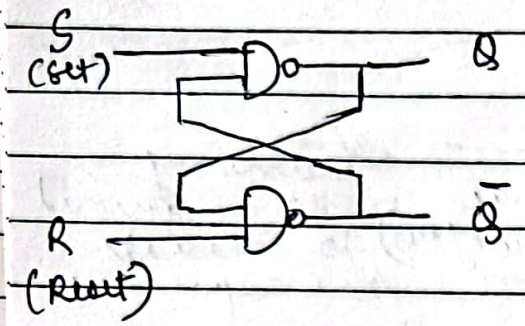
The basic structure of flip flop(s) are designed with the help of 2 structures

This is also known as SR latch

Date: / / Page no:

(i) Cross-Coupled NAND

(2 NAND gates, cross)



Key Property:

If any one of the i/p = 0
⇒ o/p = 1 (without waiting for 2nd any i/p)

S R

If 1, 1 then it will consider previous o/p ~~is considered~~ or ∴ repeated

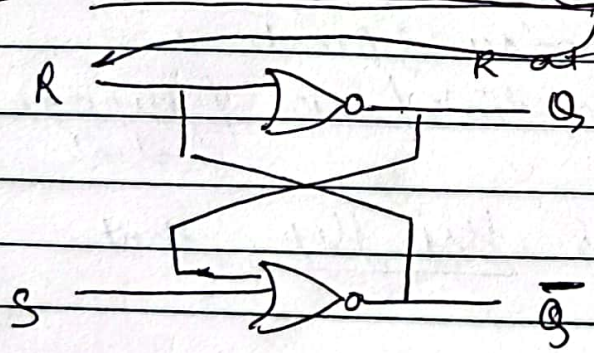
Just take prev. o/p as curr. o/p.

S	R	Q	$\bar{Q}$	
1	0	0	1	
1	1	0	1	
0	1	1	0	
1	1	1	0	
0	0	1	1	X Indeterminate state

Testing both as depends on prev.

Reason: Q = 1 & $\bar{Q}$ = 1 (Not Possible)

(ii) Cross-Coupled NOR



R at up

Why RS in NOR? Else it would give same Truth Table as NAND To get 2

S	R	Q	$\bar{Q}$	
1	0	1	0	
0	0	1	0	
0	1	0	1	
0	0	0	1	
1	1	0	0	x "Don't Care" Indeterminate

~~NAND~~ NOR	
Set = 1	Q = 1
Reset = 1	$\bar{Q}$ = 1

Just opp. in NAND

NAND → There will be never 1st State 1 1 as it waits for 2nd any i/p. ∴ first 0,1 or 1,0 & then 1,1 Sequential ckt.

Date: / / Page no:

Types of Flip Flops :-

There are 4 types of flip flops:-

(i) SR flip flop
(ii) JR flip flop
(iii) T flip flop } Generally in market, we get cross coupled NOR

(iv) D flip flop } cross coupled NAND

But can be done from both.

(i) SR Flip Flop:

Set ✓ → Reset

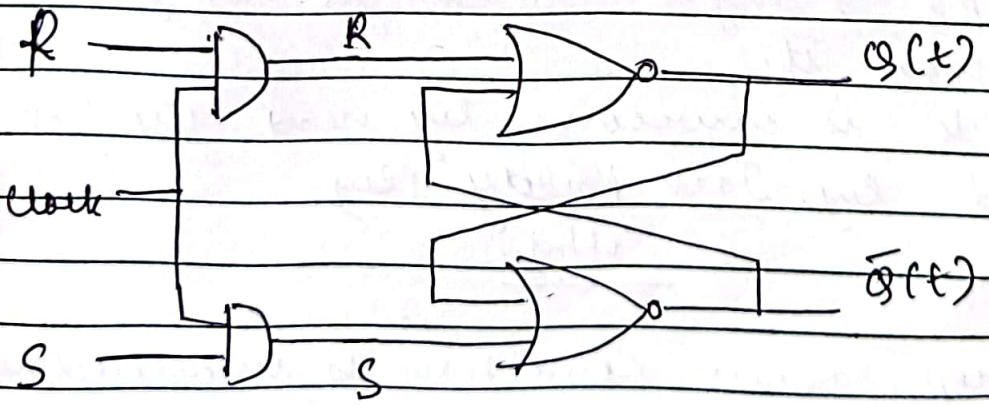
There are 2 actions performed by SR flip flop

Set — It sets the content to 1

Reset — It resets the content to 0

∴ Cross coupled NOR ideal choice

Clock → To control the signal R and S and taking the o/p control in our hands.



Characteristics Table

S	R	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	X

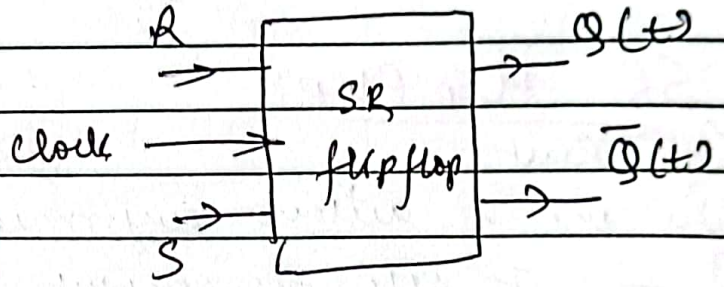
	Q(t)	S	R	Q(t+1)
(0)	0	0	0	0
(1)	0	0	1	0
(2)	0	1	0	1
(3)	0	1	1	X
(4)	1	0	0	1
(5)	1	0	1	0
(6)	1	1	0	1
(7)	1	1	1	X

Q(t) \ SR	00	01	11	10
0	0	1	X3	1 2
1	1 4	5	X7	1 6

$$Q(t+1) = S + Q(t).\overline{R}$$

Problem:

When i/p are 1, 1 then o/p is undetermined



∴ We move towards JK flip flop

(ii) JK flip flop: (Jack Kilby)

The main problem of SR flip flop is that when both the i/ps ~~of~~ high, we can ~~get~~ not find the o/p ~~go~~ from it.

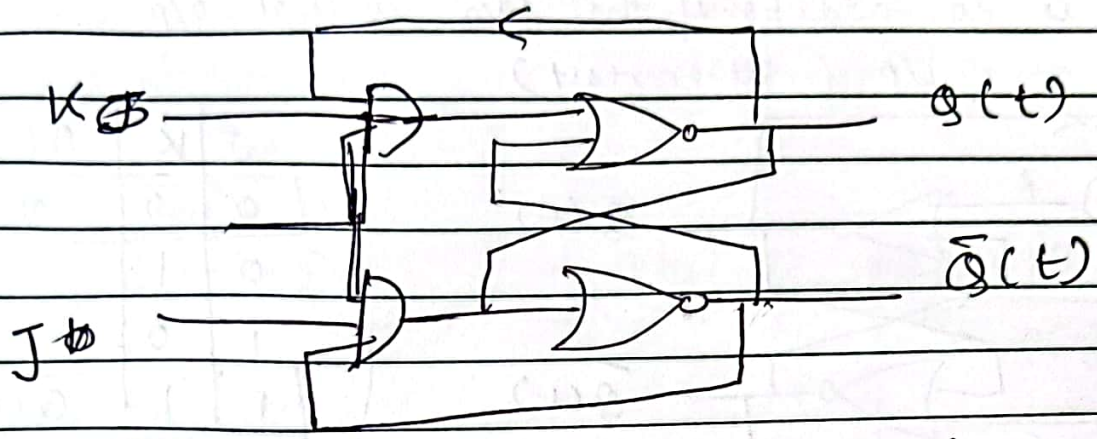
This drawback is overcome by next flip flop JK introduced by Jack ~~Kilby~~, by Kilby.

In JK flip flop, a new condition is introduced namely Toggle that returns complement of previous o/p, when both the i/ps are high & the rest of the functioning remains same.

S R
J K
R

Date: / / Page no:

Toggle condⁿ is implemented by a feedback connection from o/ps to i/ps in SR flip flops



	Q(t)	J	K	Q(t+1)
(0)	0	0	0	0
(1)	0	0	1	0
(2)	0	1	0	1
(3)	0	1	1	1
(4)	1	0	0	1
(5)	1	0	1	0
(6)	1	1	0	1
(7)	1	1	1	0

Toggle

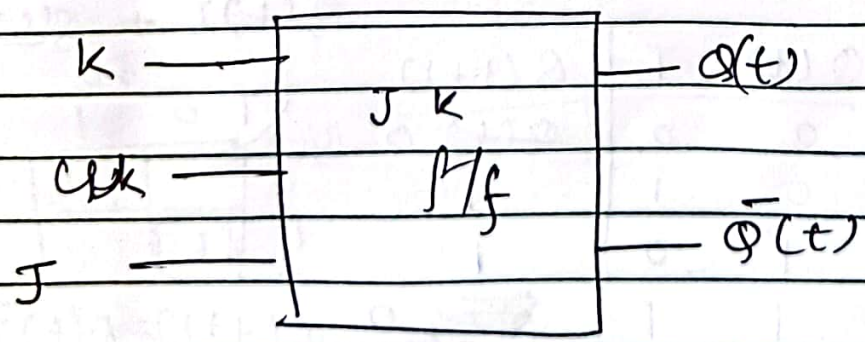
Toggle

J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	$\overline{Q(t)}$

Q(t) \ JK	00	01	11	10
0	0		1,3	1,2
1	1,4		5	7 1,6

$$Q(t+1) = \overline{Q(t)}.J + Q(t).\bar{K}$$

~~Q(t) (J + $\bar{K}$)~~



11.11.2022 Friday

→ Toggle flip flop

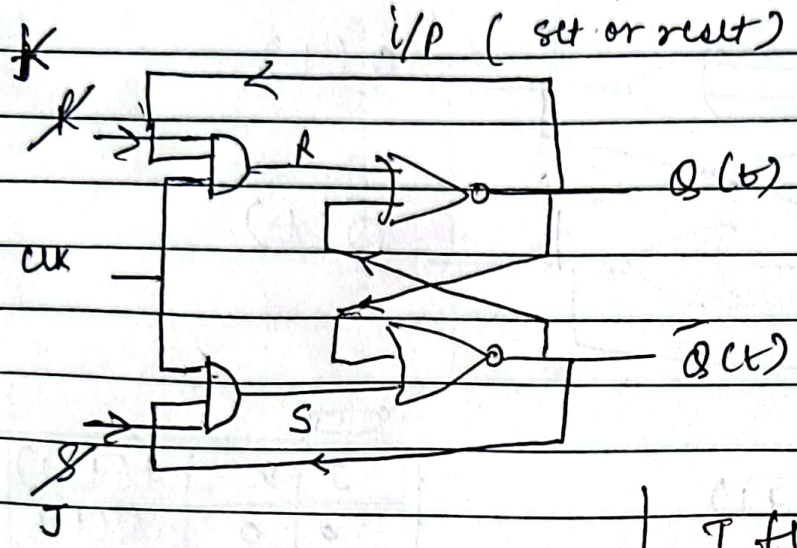
Date: / / Page no:

problem in SR and JK

(iii) T flip flop

The o/p will depends on previous o/p only.

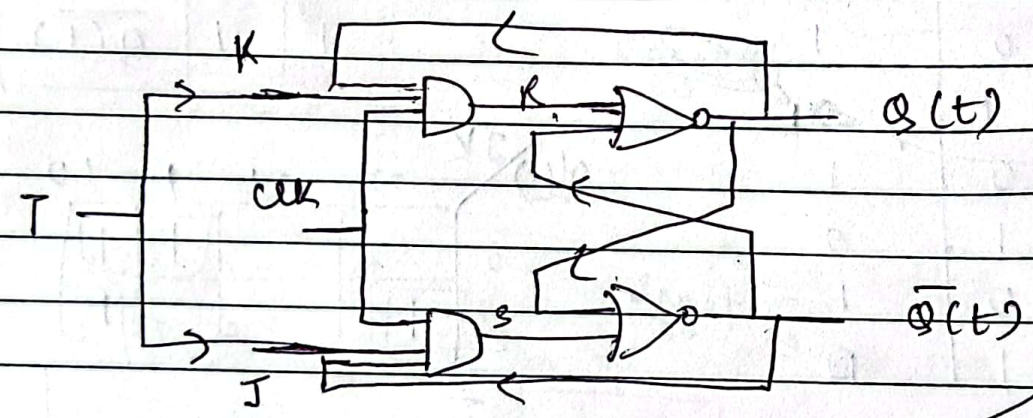
There is no additional i/p (set or reset) that can control o/p



J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	$\bar{Q}(t)$

↓ T flip flop

Look at these



In JK flip flop bind both the i/ps J & K by a single i/p T in order to construct f/f

Characteristics Table

T	Q(t+1)
0	Q(t)
1	$\bar{Q}(t)$

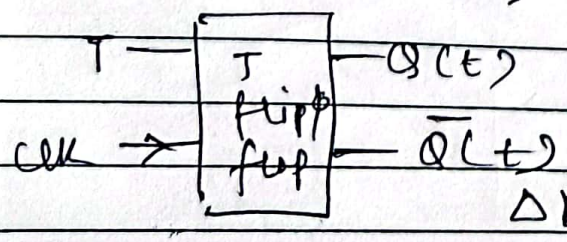
Q(t)	T	Q(t+1)
0	0	~~Q(t)~~ 0 Q(t)
0	1	1
1	0	1
1	1	~~Q(t)~~ 0

$Q(t)\bar{T} + \bar{Q}(t)T$

	T 0	T 1
Q(t)		1
$\bar{Q}(t)$	1	

$Q(t+1) = Q(t)\bar{T} + Q(t)T$

$\boxed{Q(t+1) = Q(t) \oplus T}$



△ Block diagram

Delay (Direct) Date: / / Page no:

(iv) D - Flip Flop

The o/p will be same as supplied i/p but it will be generated after a clock delay.

(Earlier flip flops waits for previous o/p)
↳ Disadvantage

Characteristics table

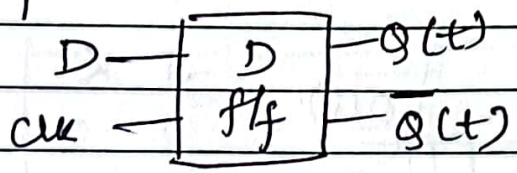
D	Q(t+1)
0	0
1	1

→

Q(t)	D	Q(t+1)
0	0	0
0	1	1
1	0	0
1	1	1

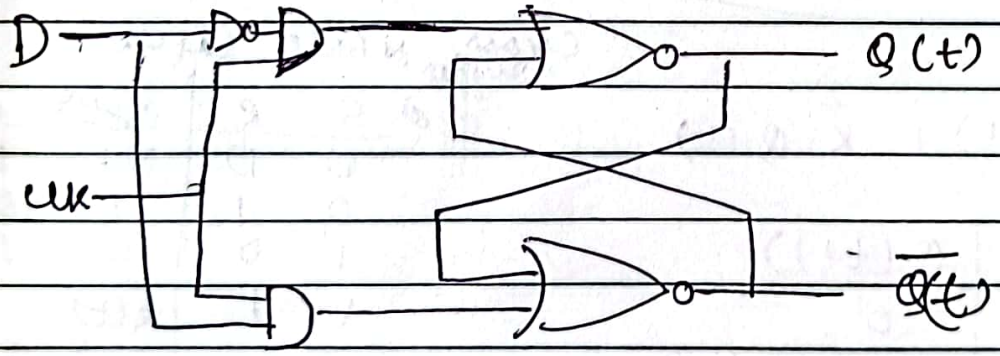
Q(t) \ D	0	1
0		1
1		1

$Q(t+1) = D$



Block diagram

How to draw with cross coupled NOR / NAND



Always first write what we know to draw this

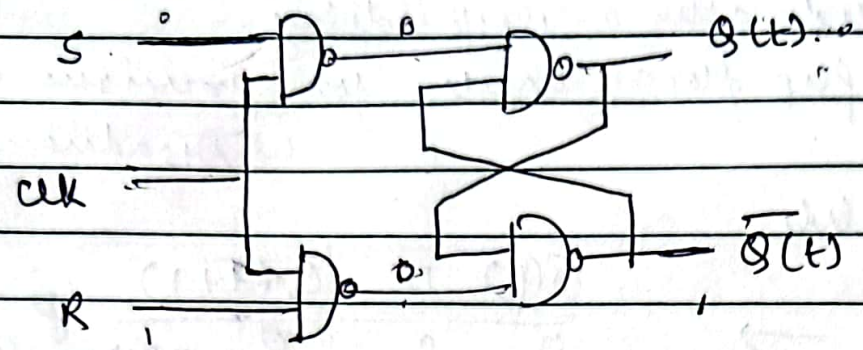
any i/p = 0
o/p = 1

Date: / / Page no:

clk = 1 (Triggered)

Cross Coupled NAND

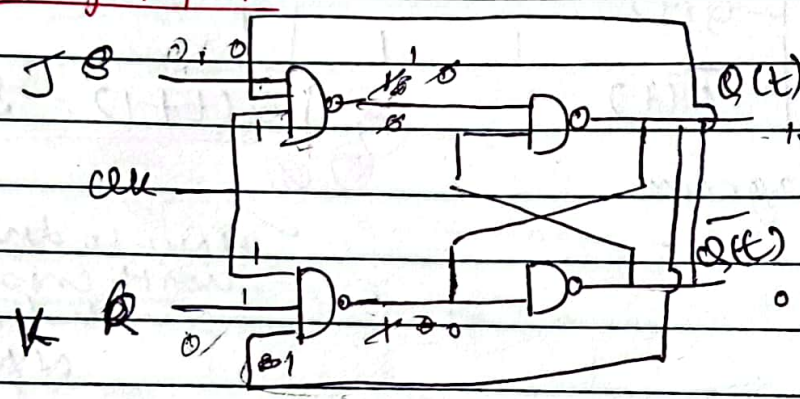
① SR flip flops



S	R	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	x

$Q(t+1) = S + Q(t)\bar{R}$

② JK flip flops



J	K	Q(t+1)
0	0	Q(t) = holds
0	1	0
1	0	1
1	1	$\bar{Q}(t)$ = Toggle

$Q(t+1) = J\bar{Q}(t) + \bar{K}Q(t)$

Q(t)	J	K	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Cross Coupled NAND Latch

S	R	Q	$\bar{Q}$
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q(t)	

Q(t) \ JK	00	01	11	10
0	0	1	1 (3)	1 (2)
1	1 (4)	(5)	(7)	1 (6)

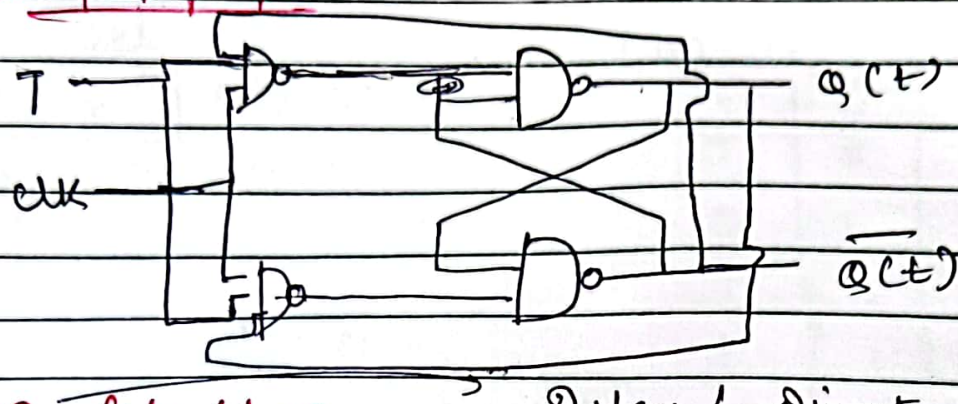
$Q(t+1) = J\bar{Q}(t) + \bar{K}Q(t)$

→ Toggle

Date: / / Page no:

(3) ~~D~~ T flip flops

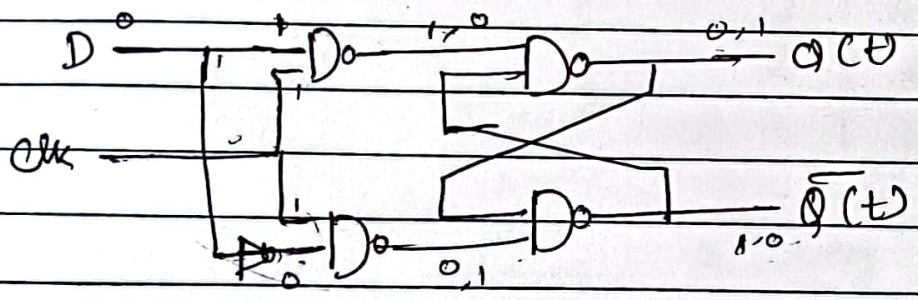
$J = K = T$



~~Q(t)~~ T	Q(t+1)
0	Q(t)
1	$\overline{Q(t)}$

(4) D flip flops

Delay / Direct



D	Q(t+1)
0	0
1	1

Characteristics Table

(i) SR ff

S	R	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	~~Q(t)~~ X

(ii) JK ff

J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	$\overline{Q(t)}$

(iii) T ff

T	Q(t+1)
0	Q(t)
1	$\overline{Q(t)}$

(iv) D ff

D	Q(t+1)
0	0
1	1

Excitation Table

Directly XOR ⊕

Q(t)	Q(t+1)	S	R	J	K	D	T
0	0	0	~~X~~ 0	0	~~X~~ 0	0	0
0	1	1	0	1	~~X~~ 1	1	1
1	0	0	1	~~X~~ 1	1	0	1
1	1	~~X~~ 0		~~X~~ 0	0	1	0

Remember

We will be given o/p & prev. o/p then we need to find i/p

14.11.2022
Monday

Mealy M/C ⟺ Moore M/C
(Transition on edges) (Transition on states)

Date: / / Page no:

TYPES OF SEQUENTIAL CIRCUIT :

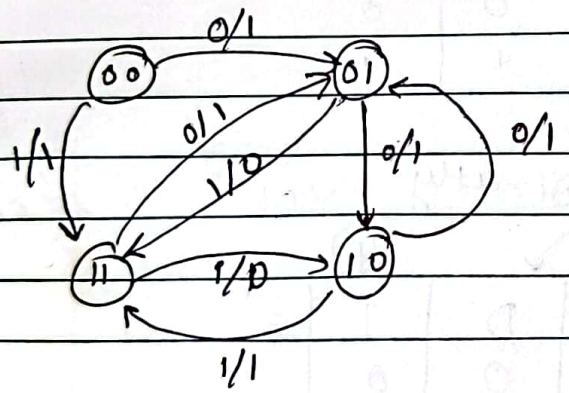
Theory of computation (4th sem)

(i) Finite State machine (Mealy Machine) → part of Discrete maths
(ii) Counters
(iii) Registers

Another is moore machine

(i) Finite State Machine (Mealy M/C)

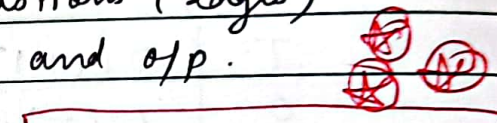
(1) set of states
(2) states are connected via transitions (edges)
(3) Every transition represents i/p and o/p.



4 states
00
01
10
11

Transitions = = edges

from state ab ⟺ if we select input i we reach state cd producing output 'o'



A flow graph

Input : 1 bit
output : 1 bit

States are stored in flip flop

$$\text{No. of f/f needed} = \lceil \log_2 (\text{No. of states}) \rceil$$

ceil

4 bits → 2 f/f

00
01
10
11

Transition Table

Date / / Page no:

	Present State A(t)	B(t)	i/p x	Next State A(t+1)	B(t+1)	o/p y	J_A	K_A	J_B	K_B
(0)	0	0	0	0	1	1	0	X	1	X
(1)	0	0	1	1	1	1	1	X	1	X
(2)	0	1	0	1	0	1	1	X	X	1
(3)	0	1	1	1	1	0	1	X	X	0
(4)	1	0	0	0	1	1	X	1	1	X
(5)	1	0	1	1	1	1	X	0	1	X
(6)	1	1	0	0	1	1	X	1	X	0
(7)	1	1	1	1	0	0	X	0	X	1

Ryu A(t) & A(t+1)

5 K maps y, J_A, K_A, J_B, K_B

A(t) \ B(t)x	00	01	11	10
0	0	1 1	1 3	1 2
1	X 4	X 5	X 7	X 6

A(t) \ B(t)x	00	01	11	10
0	X	X	X	X
1	1			1

$$\boxed{J_A = \overline{A(t)} + B(t) + x}$$

$$\boxed{K_A = \bar{x}}$$

A(t) \ B(t)x	00	01	11	10
0	X	1	X	X
1	1	1	X	X

	00	01	11	10
0	X	X	·	1
1	X	X	1	·

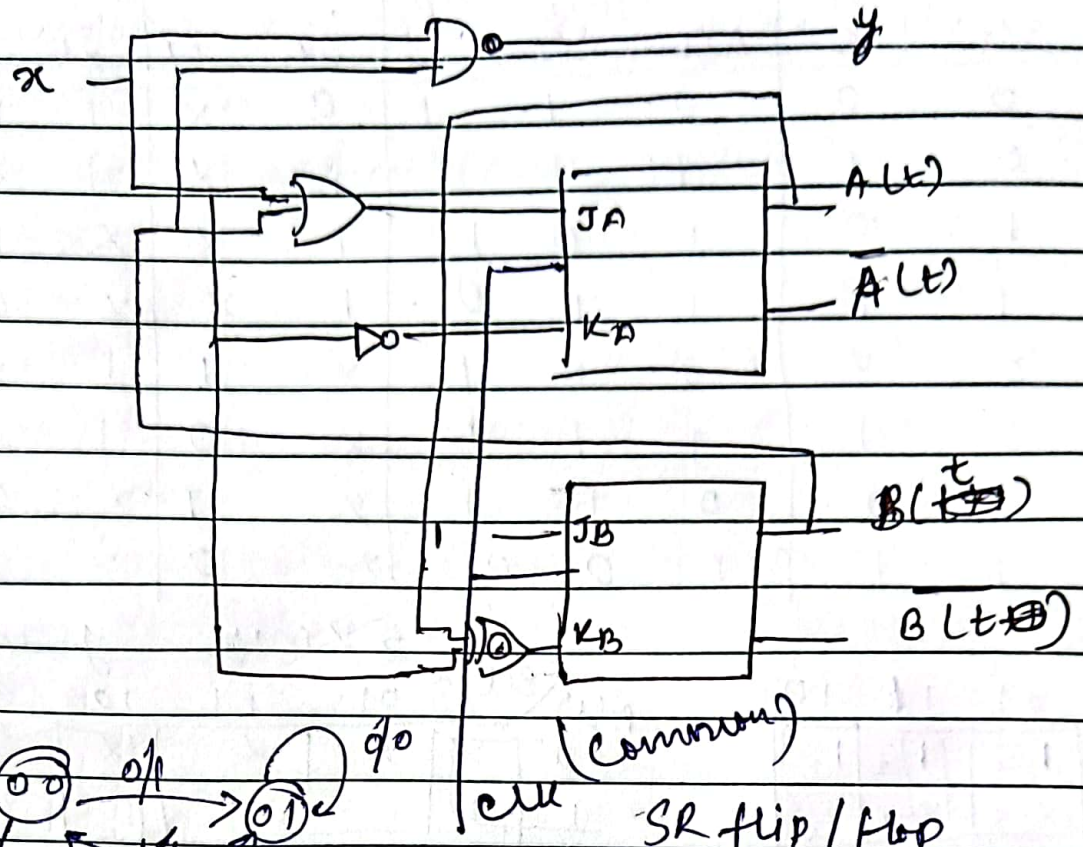
$$A(t)\,x + \overline{A(t)}\,\bar{x}$$
$$= A(t) \oplus x$$

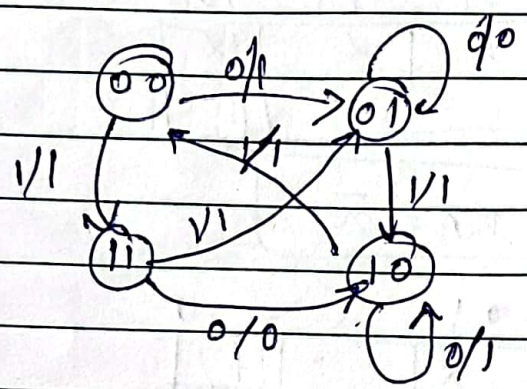
$$\boxed{J_B = 1}$$

A(t) \ B(t)x	00	01	11	10
0	1	1		1
1	1	1		1

$$y = \overline{B(t)} + \bar{x}$$
$$= (B(t) \cdot x)'$$

Why this? → To use directly NAND



Q) 

Sol:

4 States:
$\log_2 4 = 2$ f/f

Previous State A(t)	B(t)	i/p x	N ext state A(t+1)	B(t+1)	o/p y	J_A	R_A	J_B	R_B
0	0	0	0	1	1	0	X	1	0
0	0	1	1	1	1	1	0	1	0
0	1	0	0	1	0	0	X	X	0
0	1	1	1	0	1	1	0	0	1
1	0	0	1	0	1	X	0	0	X
1	0	1	0	0	1	0	1	0	X
1	1	0	1	0	0	X	0	0	1
1	1	1	0	1	1	0	1	X	0

$A = A(t)$

$B = B(t)$

Date: / / Page no:

A \ Bx	00	01	11	10
0	0	1	1	2
1	X 4	5	7	X 6

	00	01	11	10
0	X			X
1		1	1	

(1) $S_A = \bar{A}x$

$R_A = Ax$

A \ Bx	00	01	11	10
0	1	1	0	X
1	0	0	X	0

	00	01	11	10
0	0	0	1	0
1	X	X	0	1

$S_B = \bar{A}.\bar{B} = (\overline{A(t)+B(t)})$

$R_B = \bar{A}Bx + A\bar{x}$

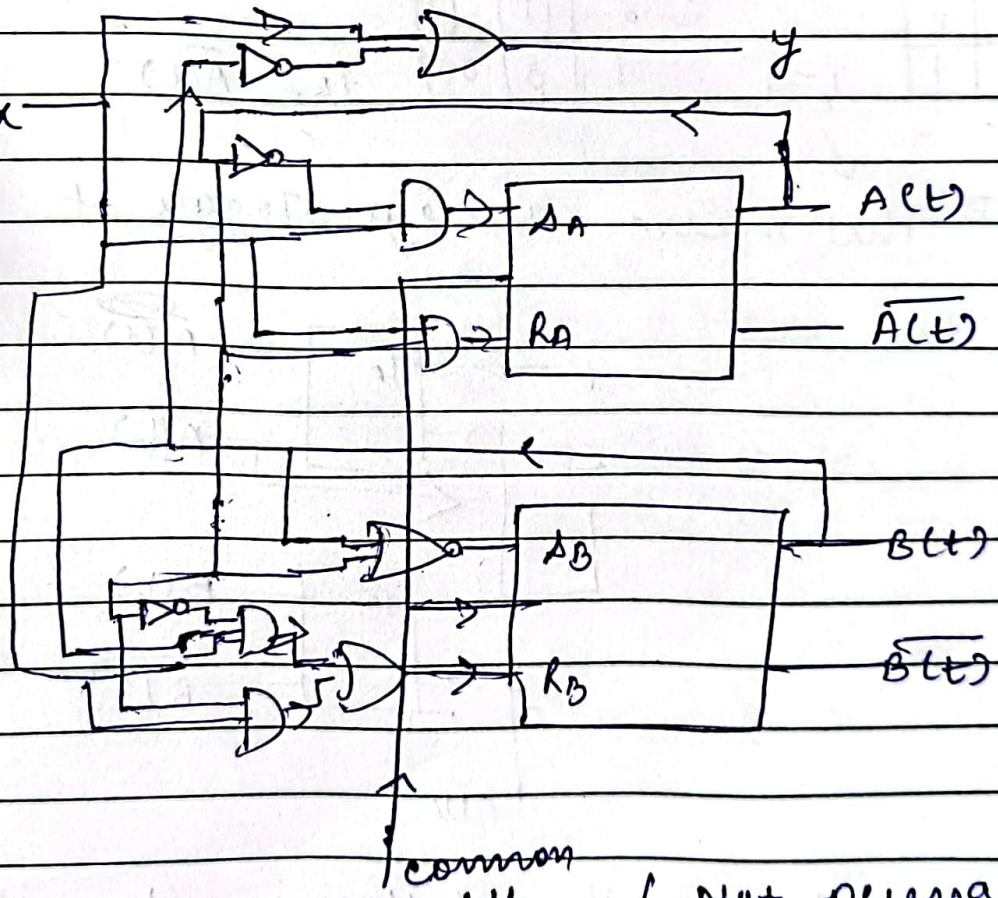
To use only one gate

	00	01	11	10
0	1	1	1	0
1	1	1	1	0

$y = \overline{B(t)} + \bar{x}$

(A) (B)

(A) diagrams of ckt are important



(Not necessary always common clk)

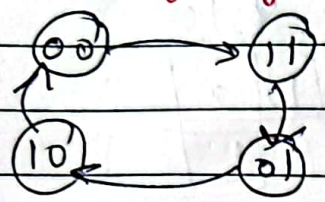
Sequential Synchronous ckts → synchronous

"Asynchronous" → diff clk, o/p of one as clk of next

random sequence ~~but it~~ there

(ii) **Counter** : It is a sequential circuits that repeat finite sequence.

Designing of counters:



We can start from any state & it repeat.

4 States

from diagram

Present State A(t)	Present State B(t)	Next State A(t+1)	Next State B(t+1)	T_A	T_B
0	0	1	1	1	1
0	1	1	0	1	1
1	0	0	0	1	0
1	1	0	1	1	0

Present state HOW TO

4 States

T_A

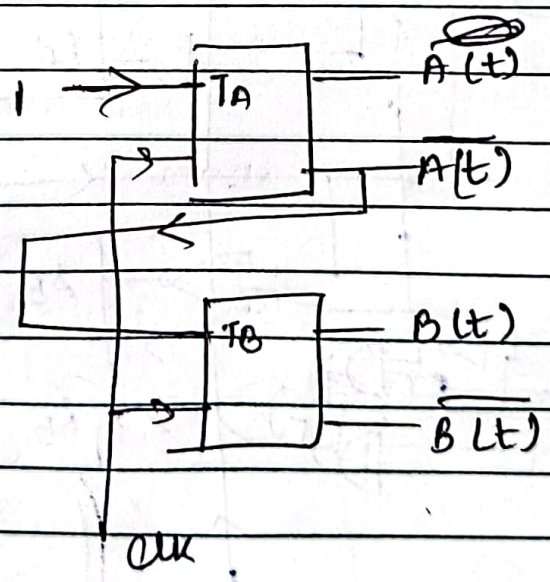
A(t) \ B(t)	0	1
0	1	1
1	1	1

$T_A = 1$

A(t) \ B(t)	0	1
0	1	1
1	0	0

$T_B = \overline{A(t)}$

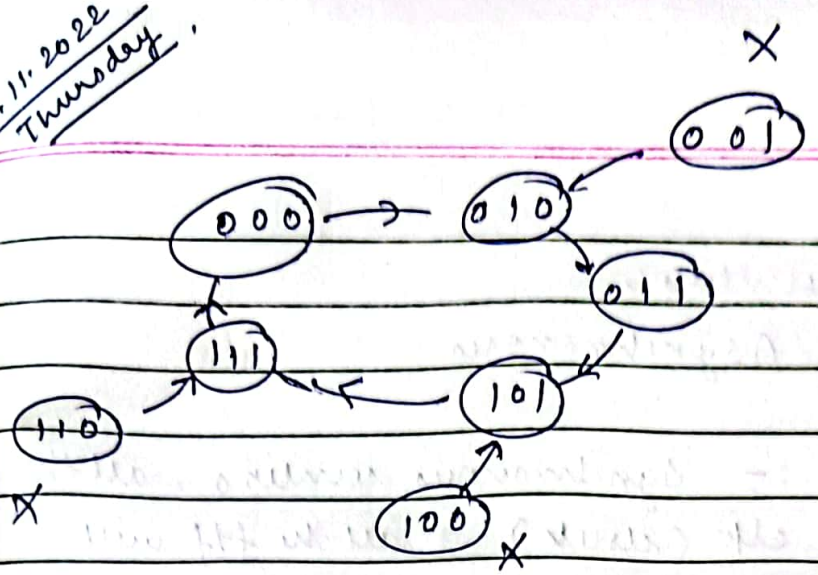
that means always Toggle at T_A



Q) Design counter for the following state diagram use D flip

17.11.2022 Thursday.

Date: / / Page no:



Sol: counting sequence = 5 (Central circle)

X → Useless state

Not repetitive (violates counter defn)

No. of f/f = $\lceil \log_2(N) \rceil$

$= \lceil \log_2 5 \rceil = 3$

	Present State			Next State					
	A(t)	B(t)	C(t)	A(t+1)	B(t+1)	C(t+1)	D_A	D_B	D_C
(0)	0	0	0	0	1	0	0	1	0
(2)	0	1	0	0	1	1	0	1	1
(3)	0	1	1	1	0	1	1	0	1
(5)	1	0	1	1	1	1	1	1	1
(7)	1	1	1	0	0	0	0	0	0

Don't cares = 1, 4, 6

A \ BC	00	01	11	10
0	0_0	X_1	1_3	0_2
1	X_4	1_5	0_7	X_6

A \ BC	00	01	11	10
0	1	X		1
1	X	1		X

A \ BC	00	01	11	10
0		X	1	1
1	X	1		X

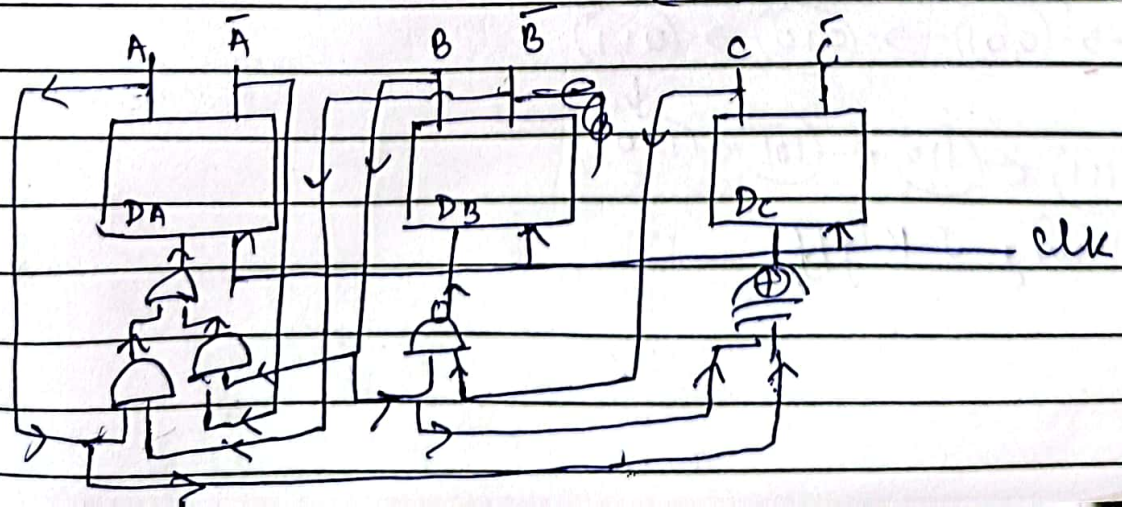
$D_A = A\bar{B} + \bar{A}C$

$D_B = \bar{B} + \bar{C}$

$= \overline{(B \cdot C)}$

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

$= A \oplus B$



Types of counters

There are 2 types of counters;
Synchronous and Asynchronous
(Previous done)

① **Synchronous Counters** :- Synchronous counters, all the f/f will have same clk (clock) ⇒ all the f/f will work parallely.

② **Asynchronous Counters** :- Asynchronous counters, o/p of f/fs may work as a clock for other f/fs.
No common clock
It means all the f/fs will not work at the same time and working of f/fs will depend on clock nature of o/p of f/fs

Synchronous Counters:

There are three types of Synchronous counters.

① **Binary Counter** It counts sequences in order. There are 2 types of Binary counters

(a) **Binary up counter**; It counts forward sequences in order

For ex. let the counting sequences for 3 bit Binary up counter are

000 → 001 → 010 → 011 → 100 → 101 → 110 → 111 → 000

Design using J K f/f

Date: / / Page no:

	Present State			Next State								
	$Q_2(t)$	$Q_1(t)$	$Q_0(t)$	$Q_2(t+1)$	$Q_1(t+1)$	$Q_0(t+1)$	J_1	K_1	J_2	K_2	J_3	K_3
(0)	0	0	0	0	0	1	0	x	0	x	1	x
(1)	0	0	1	0	1	0	0	x	1	x	x	1
(2)	0	1	0	0	1	1	0	x	x	0	1	x
(3)	0	1	1	1	0	0	1	x	x	1	x	1
(4)	1	0	0	1	0	1	x	0	0	x	1	x
(5)	1	0	1	1	1	0	x	0	1	x	x	1
(6)	1	1	0	1	1	1	x	0	x	0	1	x
(7)	1	1	1	0	0	0	x	1	x	1	x	1

$Q_2 \backslash Q_1Q_0$	00	01	11	10
0	0	1	3	2
1	x 4	x 5	x 7	x 6

$J_1 = Q_1Q_0$

$Q_2 \backslash Q_1Q_0$	00	01	11	10
0	x	x	x	x
1			1	

$K_1 = Q_1Q_0$

$J_0 = 1$

$K_0 = 1$

(all filled)

[No map required]

$Q_2 \backslash Q_1Q_2$	00	01	11	10
0		1	x	x
1		1	x	x

$J_2 = Q_0$

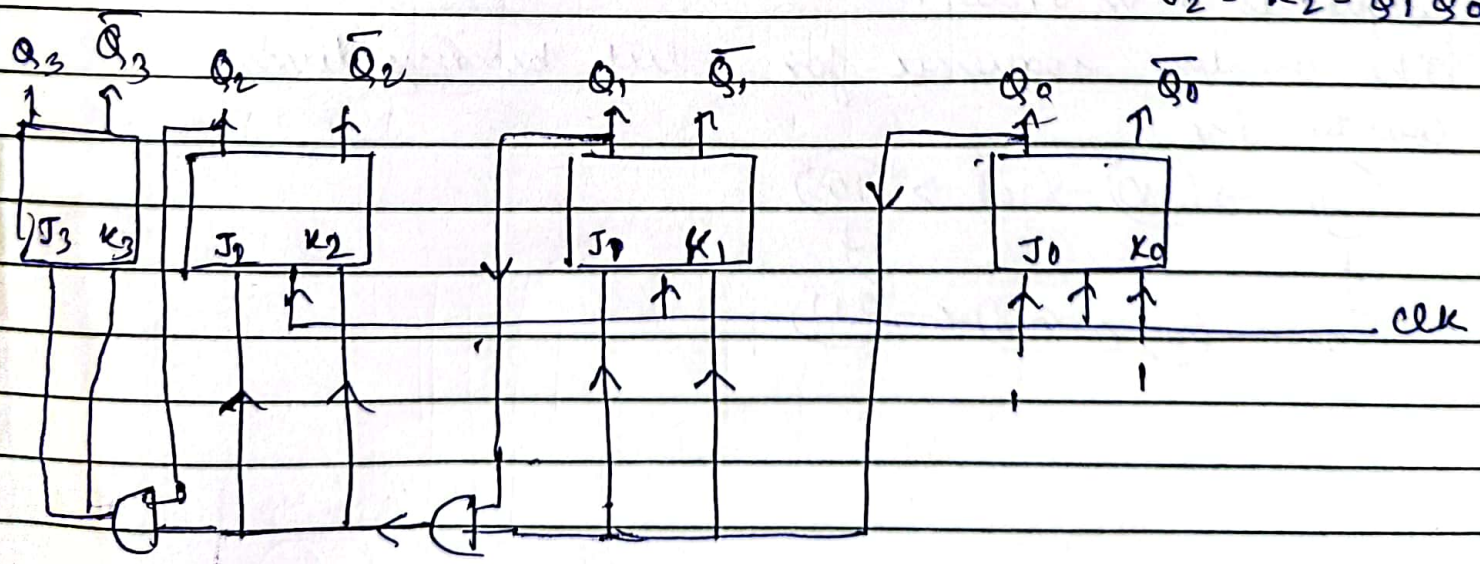
$Q_2 \backslash Q_1Q_2$	00	01	11	10
0	x	x	1	
1	x	x	1	

$K_2 = Q_0$

$J_0 = K_0 = 1$

$J_1 = K_1 = Q_0$

$J_2 = K_2 = Q_1Q_0$

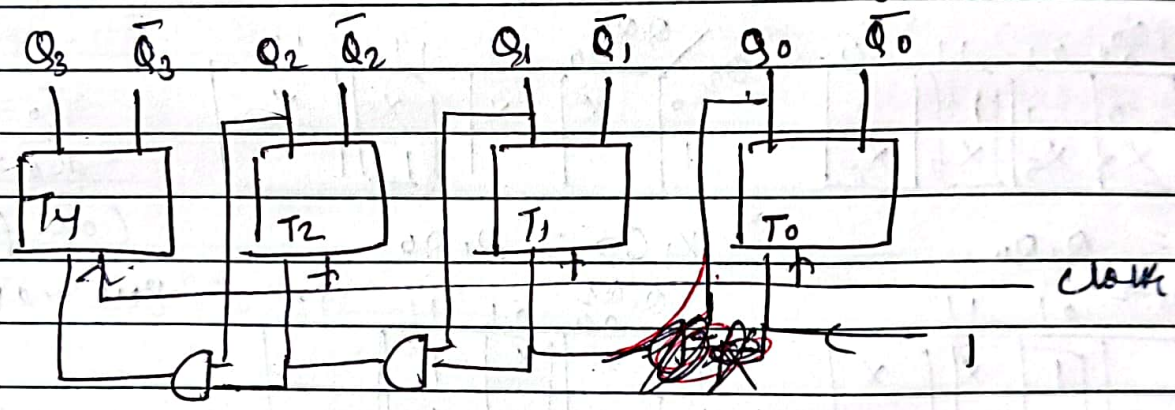


Same for up & down

Generalised

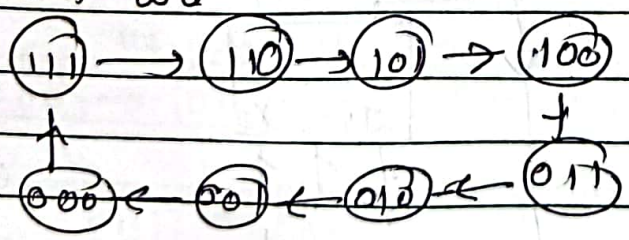
$T_0 =$	$J_0 = K_0 = 1$
$T_1 =$	$J_1 = K_1 = Q_0$
$T_2 =$	$J_2 = K_2 = Q_1 Q_0$
$T_3 =$	$J_3 = K_3 = Q_2 Q_1 Q_0$
⋮	⋮
$T_{n-1} =$	$J_{n-1} = K_{n-1} = Q_{(n-2)} Q_{(n-3)} \ldots Q_1 Q_0$

T ff JK (J = = K) ∴ only in this we observe



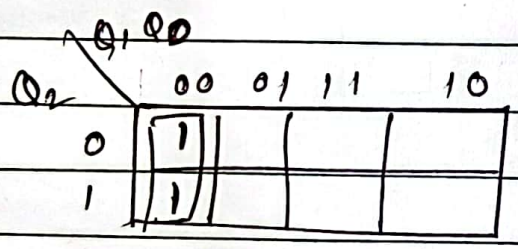
(b) Binary Down Counter : It counts backward sequences in order.

The counting sequence for 3 bit binary down counter are

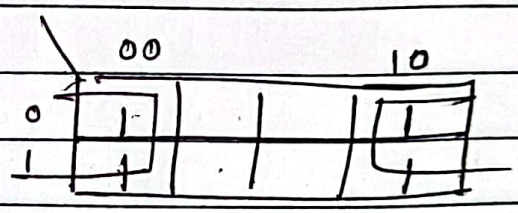


PS			NS			XOR		
$Q_2(t)$	$Q_1(t)$	$Q_0(t)$	$Q_2(t+1)$	$Q_1(t+1)$	$Q_0(t+1)$	T_2	T_1	T_0
(7) 1	1	1	1	1	0	0	0	1
(6) 1	1	0	1	0	1	0	1	1
(5) 1	0	1	1	0	0	0	0	1
(4) 1	0	0	0	1	1	1	1	1
(3) 0	1	1	0	1	0	0	0	1
(2) 0	1	0	0	0	1	0	1	1
(1) 0	0	1	0	0	0	0	0	1
(0) 0	0	0	1	1	1	1	1	1

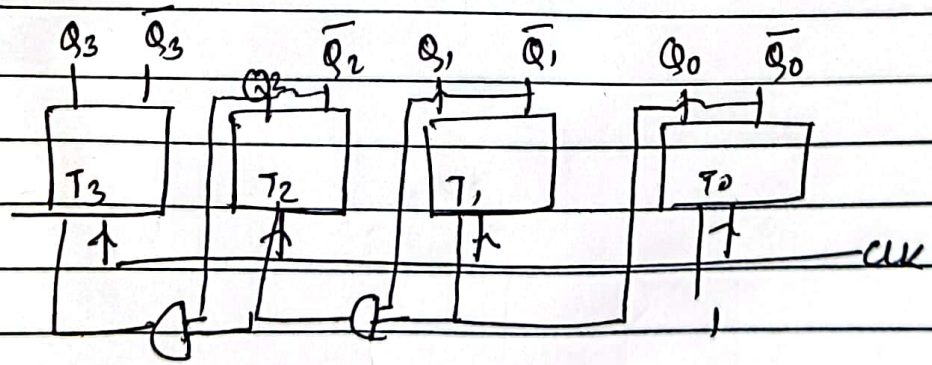
$T_0 = 1$ (write directly in exam)



$T_2 = \bar{Q}_1 \bar{Q}_0$



$T_1 = \bar{Q}_0$



$T_0 = 1$

$T_1 = \bar{Q}_0$

$T_2 = \bar{Q}_1 \bar{Q}_0$

$T_3 = \bar{Q}_2 \bar{Q}_1 \bar{Q}_0$

$\vdots$

$T_{n-1} = Q_{n-2} \cdots \bar{Q}_1 \cdot \bar{Q}_0$

18.11.2022
Friday

Date: / / Page no:

Design both binary up counter & binary down counter ~~by~~ in a single circuit

Always in sequential ckt, knowledge about initial state is must.

Date: / / Page no:

(2) Shift Counter : It is also known as switch tail ring counter or Johnson's counter

A K bit shift counter counts 2K different sequences

Let we have 3 bit shift counter then possible counting sequences are

We start from initial state 000. When all the f/fs have same contents then different bit will be inserted at MSB, content of all the f/fs will be shifted 1 bit right and the content of LSB f/f will be destroyed this process is repeated until the content of all the f/fs are same & same & same sequences repeated.

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

	Present States			Next States					
	$Q_2(t)$	$Q_1(t)$	$Q_0(t)$	$Q_2(t+1)$	$Q_1(t+1)$	$Q_0(t+1)$	D_2	D_1	D_0
(0)	0	0	0	1	0	0	1	0	0
(4)	1	0	0	1	1	0	1	1	0
(6)	1	1	0	1	1	1	1	1	1
(7)	1	1	1	0	1	1	0	1	1
(3)	0	1	1	0	0	1	0	0	1
(1)	0	0	1	0	0	0	0	0	0

↳ Don't Cares = 2, 5

Always number in MSB first in diagram

Date: / / Page no:

D_2 \ Q_1Q_0	00	01	11	10
Q_2 0	1 0	1 1	3	X 2
1	1 4	X 5	7	1 6

$D_2 = \bar{Q}_0$

Q_2 \ Q_1Q_0	00	01	11	10
0				
1	1	X	1	1

$D_1 = Q_2$

Q_2 \ Q_1Q_0	00	01	11	10
0			1	X
1			1	1

$D_0 = Q_1$

$\therefore D_2 = \bar{Q}_0$

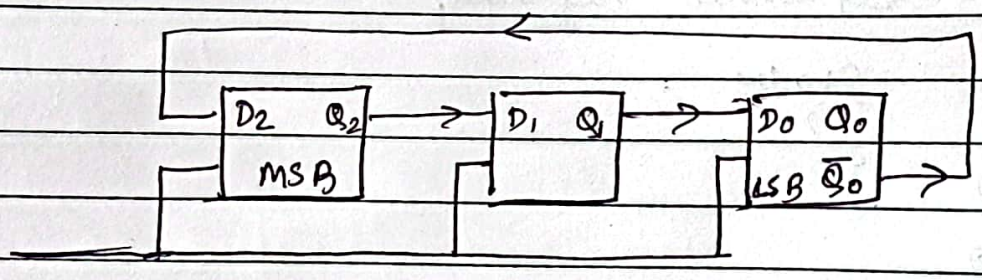
$D_1 = Q_2$

$D_0 = Q_1$

No. of ff/fs $= \lceil \log(\text{No of states}) \rceil$

$= \lceil \log 6 \rceil = 3$

Ring



Clk

Why switch tail ring?
LSB = Tail and then is complemented form Ring → There is cyclic ring

Also design it using T flf

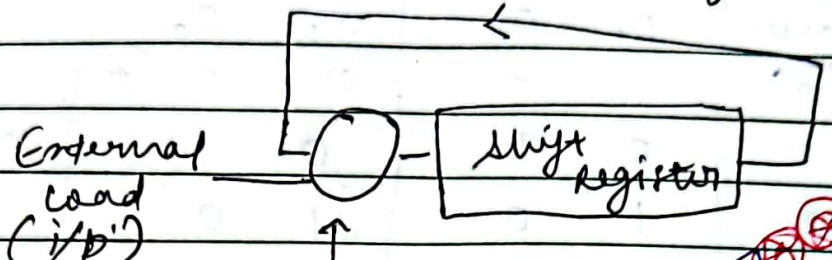
T flf is not that easy as it seems.

$\boxed{\text{D} \oplus = \text{D} = \text{XOR}}$ ≠ XNOR

Date: / / Page no:

(3) Ring Counter :

It is synchronous counter that uses internally shift register and external loads. Shift Register shifts the contents either in left dir^n or in right dir^n and ~~load~~ with the help of external load we can insert new content for counting.



Characteristics table is required (knowing i/ps)

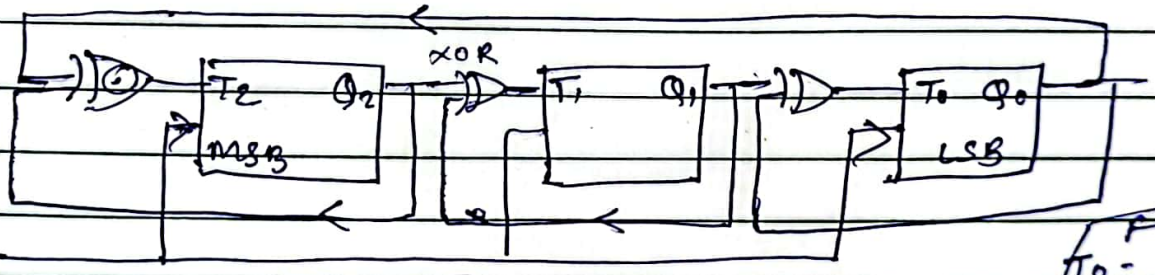
All will work parallelly only from previous i/p.

We will study at the end of counter after Register

Processing of Synchronous Counters

Counter ई | counting sequence जानना है | does not depends on Nature of clock.

Q) consider the following counter with initial state 000



From diagram
$T_0 = Q_1 \oplus Q_0$
$T_1 = Q_2 \oplus Q_1$
$T_2 = Q_2 \odot Q_0$

Sol.

clk	T_2	T_1	T_0	Q_2	Q_1	Q_0
0	—	—	—	0	0	0
1	1	0	0	1	0	0
2	0	1	0	1	1	0
3	0	0	1	1	1	1
4	1	0	0	0	1	1
5	0	1	0	0	0	1
6	0	0	1	0	0	0

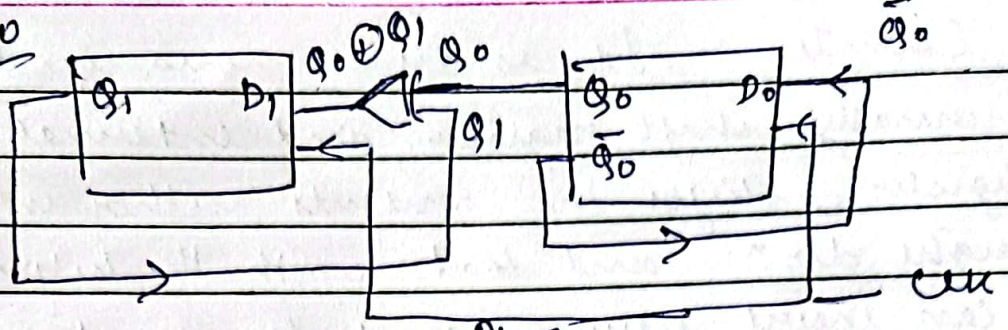
Same sequence stop.

Fill T table using this & previous

Fill next Q table using characteristic table of T flf & Q table previous

Date: / / Page no:

Q) Initial state 00



Q_1	Q_0	clk	$D_1 = Q_0 \oplus Q_1$	$D_0 = \bar{Q}_0$	Q_1	Q_0
0	0	0	–	–	0	0
		1	0	1	0	1
		2	1	0	1	0
		3	1	1	1	1
		4	0	0	0	0

Same sequence stop