①

Lec (08)
Counters

• هي registers بنصممها بحيث تعد ارقام معينة في شكل loop معين

ex 1 → 2 → 3 → 4 → 1 , 1 → 3 → 5 → 7 → 1 , etc..

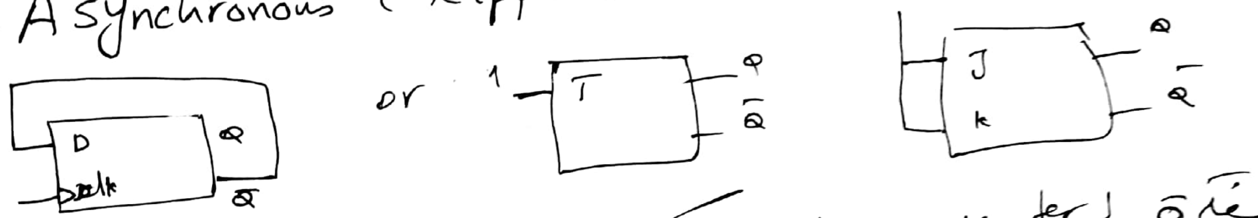
• تنقسم ال counters لنوعين هما

Synchronous counter) — Asynchronous counter (Ripple)

1- (Synchronous) :- كل ال F-F داخلها clk واحدة

2- (Asynchronous) = Ripple : خرج كل FF داخل ك clk لل FF اللي بعده

[1] Asynchronous (Ripple counter) .. using (toggle / complement theory)

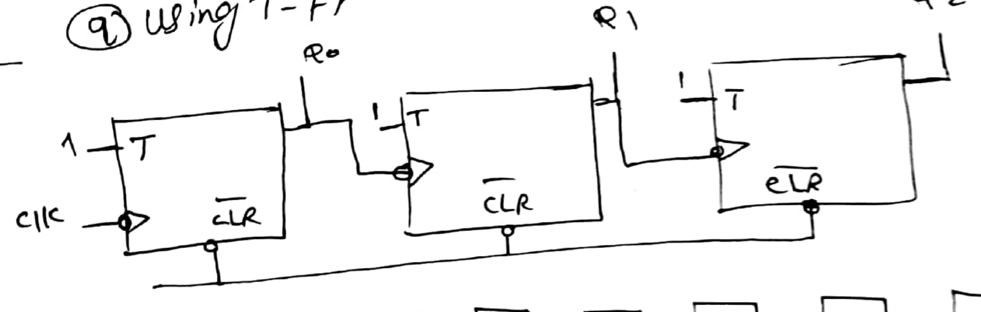


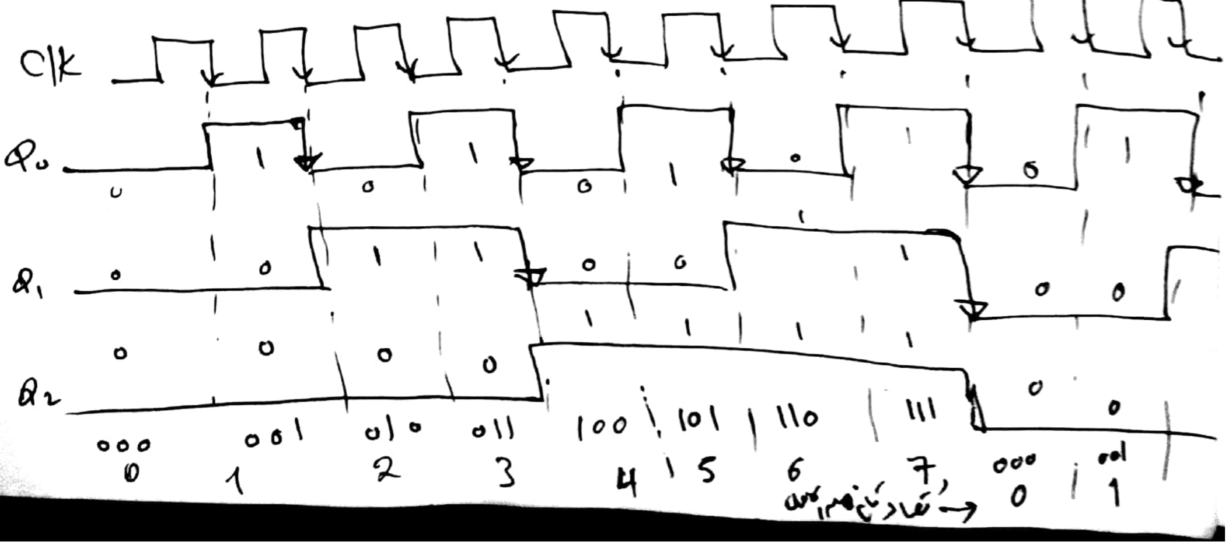
فكرة ال binary counter قائمة على فكرة Toggle أو Complement ونستخدم خرج كل FF كلوك هو نفسه clk لل FF الذي يليه ونكرر (-ve edge)

Ex : Counter 0 → 7 (Up counter) :- 3bit counter (A B C)

(a) using T-FF

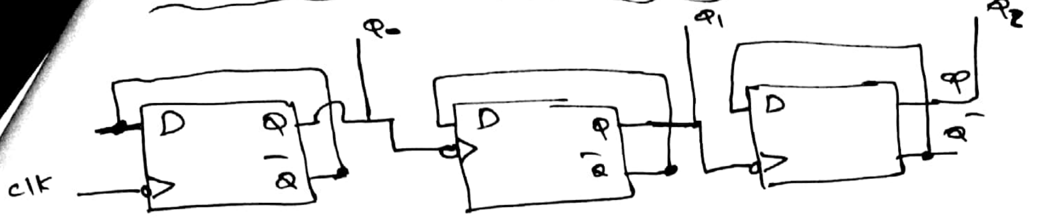
A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

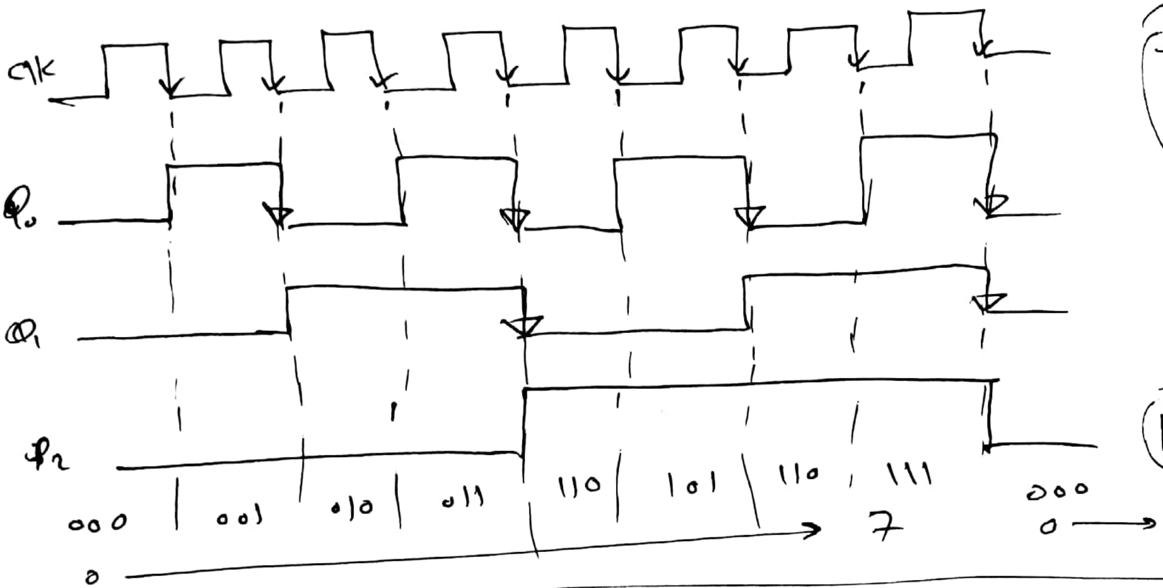




تعاد من الصفر →

3 bit Counter D-flipflop

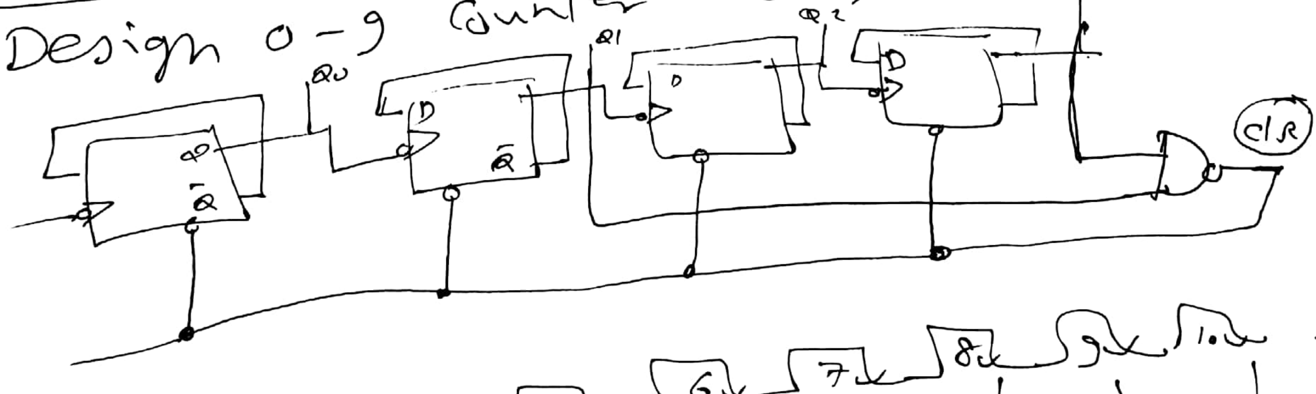


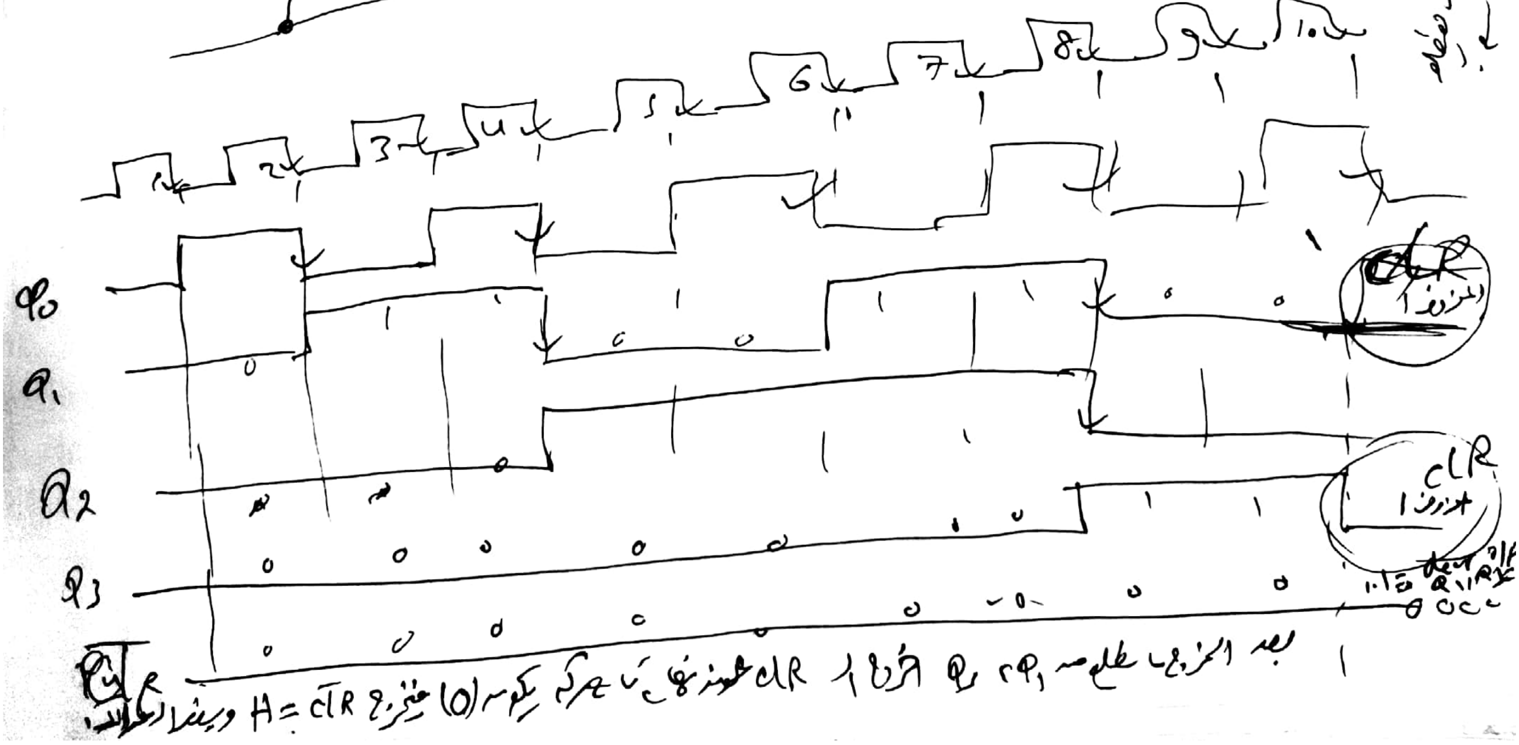


لو عاوز عداد من 0 ← 9 استخدم 4 flipflop ممكن نفطر 15 بس
هنحط بـ nand ~~clk~~ عند ما يعد 9 ويرجع للصفر

Design 0-9 Counter

0 → 9 → 0





بعد العشرة ما طلع من Q1 و Q3 ... clR [illegible] ... (0) ... H = clR

يعني الـ clk لما مرة بتكون آخر صفر لـ Q_1 هيا $Q_2 = 1$

∴ وهي الـ nand = صفر تخلي الـ

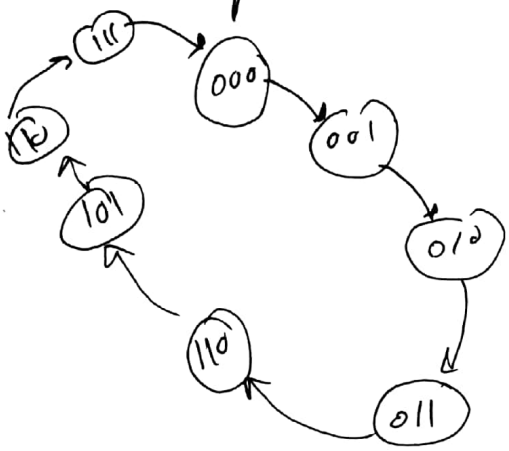
$H = \overline{clk}$ ، $Q = \overline{clk}$

Q_1, Q_3 هيفرقوا ∴

ودي عند اللحظة اللي بعد وهي [0] CC

Analyze 2- Synchronous Counter

Design 3 bit binary counter using T flip flop



Present state	Next state	F.F input
$A_2 A_1 A_0$	$A_2 A_1 A_0$	TA_2 TA_1 TA_0
0 0 0	0 0 1	0 0 1
0 0 1	0 1 0	0 1 1
0 1 0	0 1 1	0 0 1
0 1 1	1 0 0	1 1 1
1 0 0	1 0 1	0 0 1
1 0 1	1 1 0	0 1 1
1 1 0	1 1 1	0 0 1
1 1 1	0 0 0	1 1 1

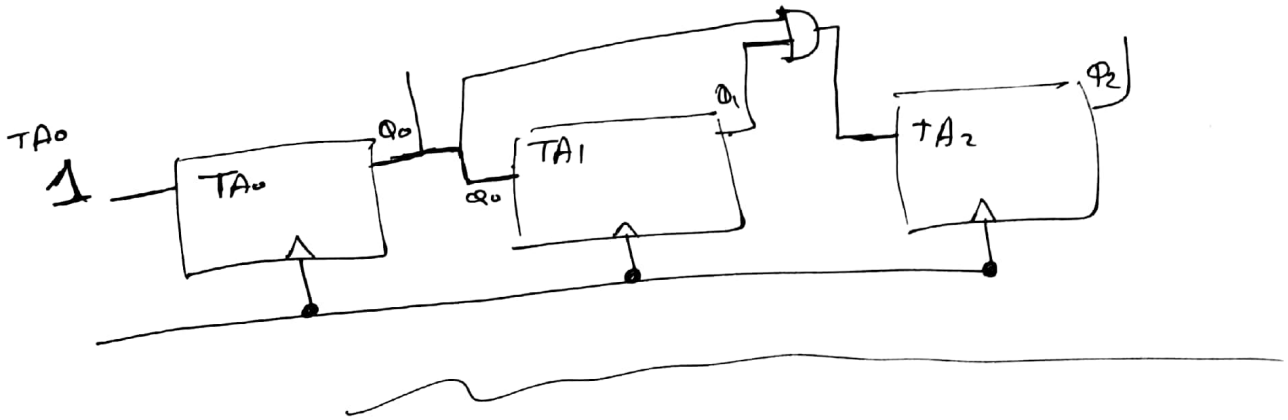
$$TA_2 = A_0 A_1 \bar{Q}_2 + A_0 A_1 A_2$$

$$= A_0 A_1 (\bar{A}_2 + A_2) = A_0 A_1$$

$$\rightarrow 1$$

$TA_0 = 1$

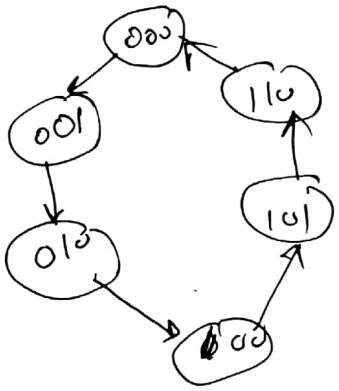
$TA_1 = A_0$



Ex Counter with non binary sequence using JK

000 → 001 → 010 → 100 → 101 → 110 → 000

There is un used state (011) ③



Present State	Next State	F.f inputs		
A_2 B_1 C_0	A_2' B_1' C_0'	J_{A_2} K_{A_2}	J_{B_1} K_{B_1}	J_{C_0} K_{C_0}
0 0 0	0 0 1	0 X	0 X	1 X
0 0 1	0 1 0	0 X	1 X	X 1
0 1 0	1 0 0	1 X	X 1	0 X
1 0 0	1 0 1	X 0	0 X	1 X
1 0 1	1 1 0	X 0	1 X	X 1
1 1 0	0 0 0	X 1	X 1	0 X
~~0 1 1~~				

J_A 0 1 B ترتيب

K_A 0 1 ترتيب

J_B 1 ⇒ C

K_B = 1 ⇐ 1 ⇒ 1 — B

J_C 0 0 B | $Q_2 \to Q_2'$ 0 → 0 ↘ no change ↘ Reset

$J_A = 0$ $J_K = 0$

$J_A = 0$ $J_K = 1$

$\therefore J_A = 0$ $J_K = X$

K_C 1 — Remember excitation table of JK

Q	Q_{next}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

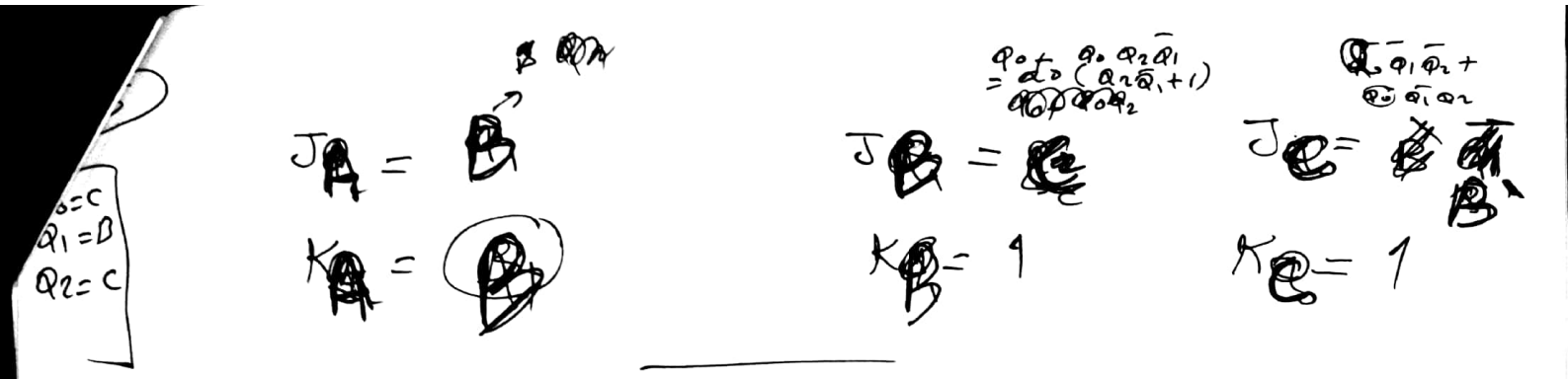
for T

Q	Q_{next}	T
0	0	0
0	1	1
1	0	1
1	1	0

Q	Q_{next}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

for D

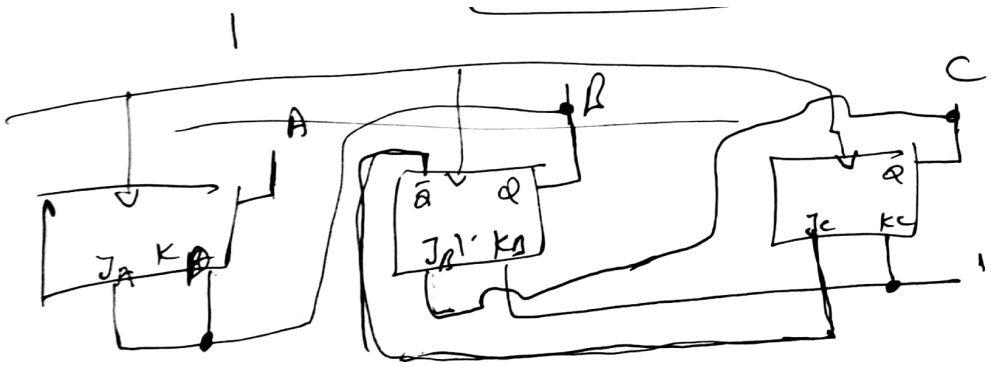
Q	Q_{next}	D
0	0	0
0	1	1
1	0	0
1	1	1



$J_C = Q_0' Q_1' Q_2' + Q_0' Q_1' Q_2 = Q_0' Q_1' (Q_2 + Q_2')$

$= Q_0' Q_1'$

$= \overline{Q_0 + Q_1} = \overline{Q_1}$



3] Analysis of clocked sequential ckt

Steps

1] Determine no: of Flip Flops

2] " " " inputs

3] " The input eqn for ff (State / transition eqn) (ch1)

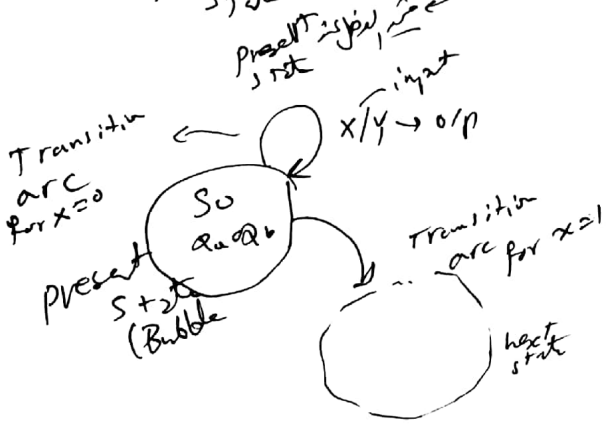
4] Draw transition table
Excitation

5] Draw State transition Diagram (STD)

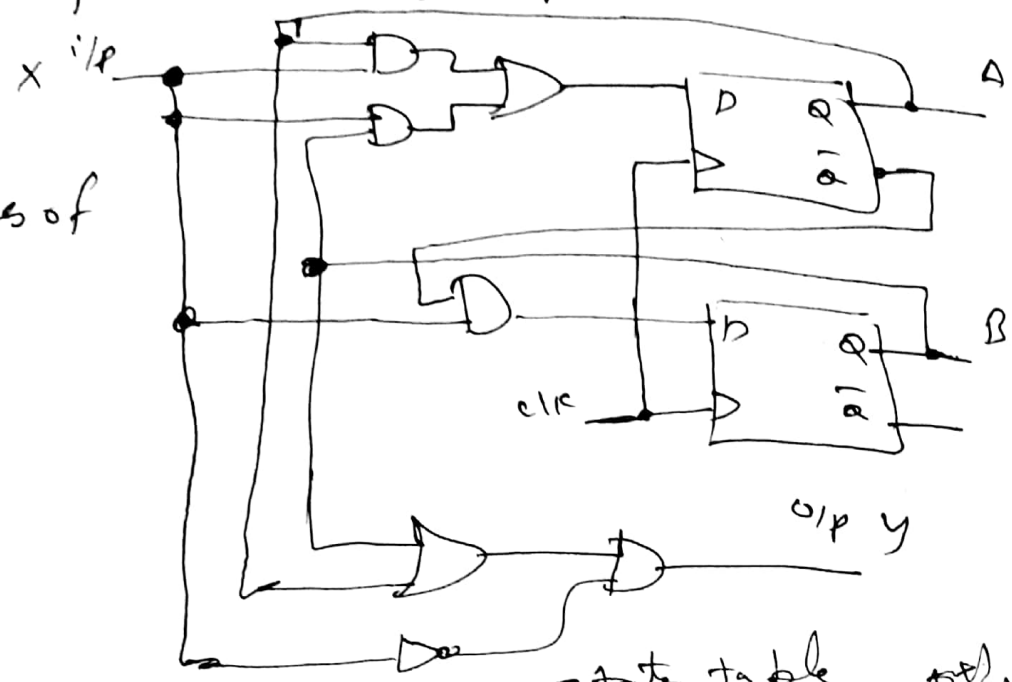
*State eqn ⇒ specifies the next state as f^n of present state

(Circuit Diagram → equations → State table → State Diagram.)

* State table
↳ Present state → input → Q_{n+1} next state → o/p



ex(1) Analysis of Following sequential ckt the behaviour



the state = Values of all f.f

Example

AB = 00
AB = 01
AB = 10
AB = 11

1-State eqn

$A_{next} = D_A = (A \cdot x + x B)$

$B_{next} = D_B = \bar{A} \cdot x$ ($\bar{a}_A$)

$y_{next} = \bar{x} \cdot (A + B)$

State table

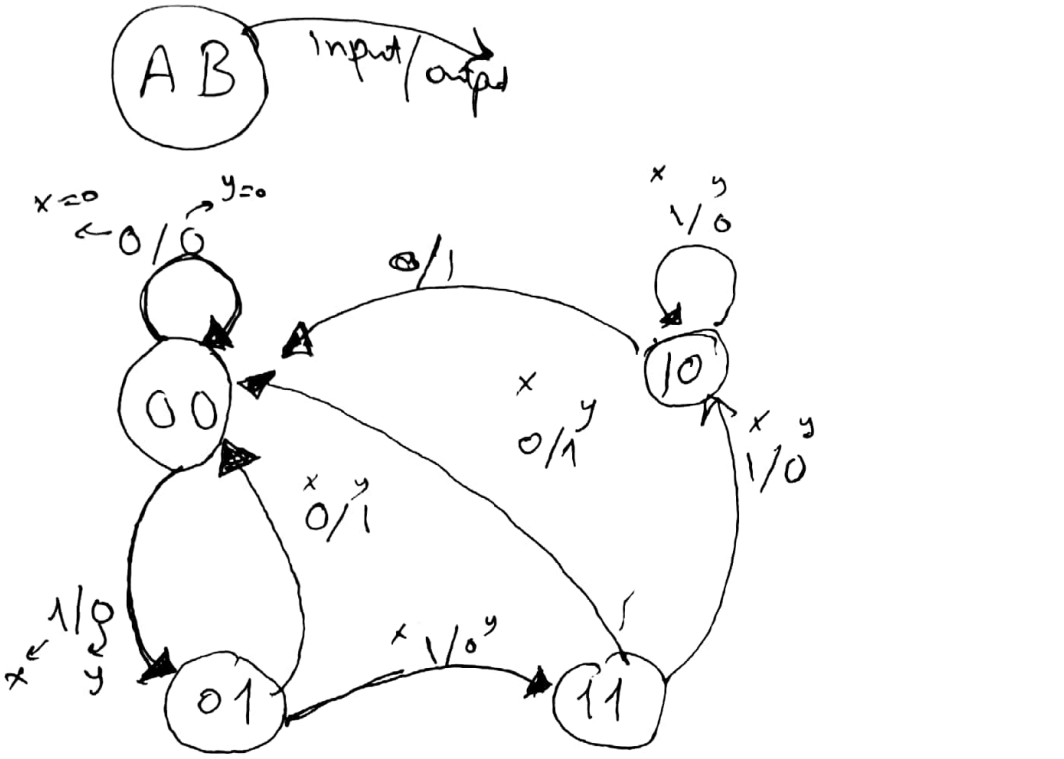
Present A B	input X	next A B	o/p y
0 0	0	0 0	0
0 0	1	0 1	0
0 1	0	0 0	1
0 1	1	1 1	0
1 0	0	0 0	1
1 0	1	1 0	0
1 1	0	0 0	1
1 1	1	1 0	0

Simplified

we can simplify table

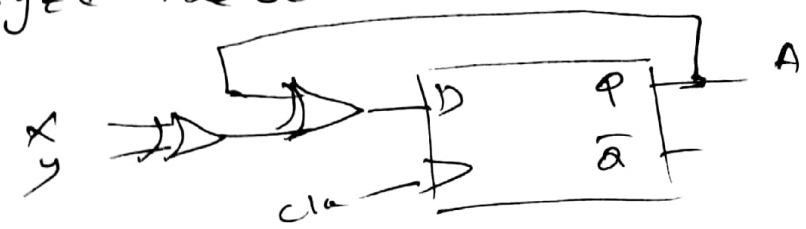
Present A B	X=0 A B	X=1 A B	o/p X=0 y	o/p X=1 y
0 0	0 0	0 1	0	0
0 1	0 0	1 1	1	0
1 0	0 0	1 0	1	0
1 1	0 0	1 0	1	0

(3) State Diagram



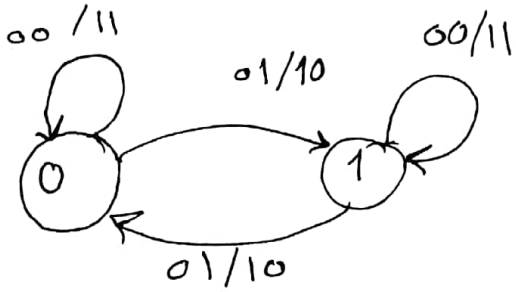
State Diagram

Ex(2) Analyze the behaviour of the following circuit?



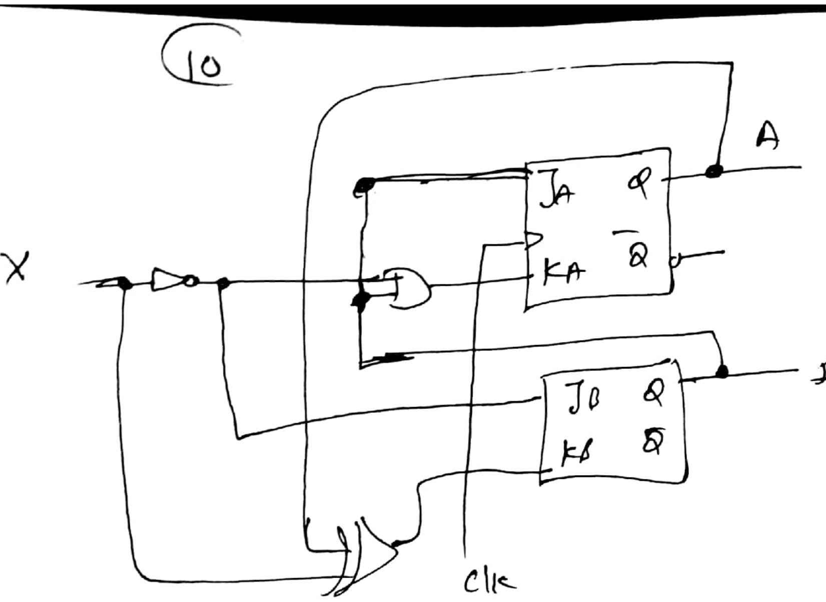
$A_{next} = A \oplus (x \oplus y)$

Present A	i/p x	i/p y	next A
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



Present	i/p	Next state	o/p (if required)

10



Ex 3

$J_A = B$

$K_A = Bx'$

$J_B = \bar{x}$

$K_B = A \oplus x$

present A	present B	i/p x	next A	next B	FF inputs J_A	K_A	J_B	K_B
0	0	0	0 (noch)	1 (set)	0	0	1	0
0	0	1	0	0 (Reset)	0	0	0	1
0	1	0	1 (toggle)	1 (set)	1	1	1	0
0	1	1	1 (set)	0 (Reset)	1	0	0	1
1	0	0	1 (nochange)	1 (toggle)	0	0	1	1
1	0	1	1 (nochange)	0 (toggle)	0	0	0	0
1	1	0	0 (toggle)	0 (toggle)	1	1	1	1
1	1	1	[illegible]	[illegible] (nochange)	1	0	0	0

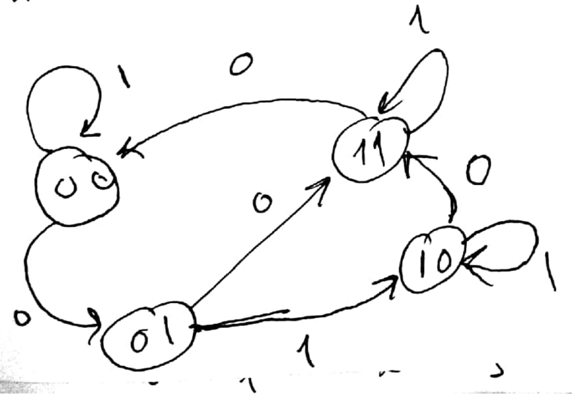
$A_{next} = J_A \bar{Q}_A + \bar{K}_A Q_A$

$B_{next} = J_B \bar{Q}_B + \bar{K}_B Q_B$

Present B / Bnext ... Apres, Anext — J_A K_A 0 0 nochange

Apres, Bpres ... 11 explained

... 00 ... no change ... Apres ... Anext = 0 ... 0 = ... 0



... 1 ... 2 ... 4 ... 0 1 1

Down Counter ① use +ve edge + Q
or $\bar{Q}$ + (-ve) edge