Unit 4 (1 - 12)

Sequential circuit design and Analysis.

synchronous / clocked seq ckts : They are represented by two models :

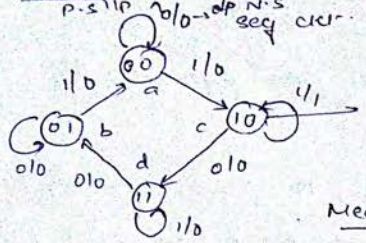
- moore circuits — o/p depends only on present state of FF
- mealy ckt - o/p depends on both P.S of FF & on I/p's

* sequential ckts can also called as Finite State machines {ie finite no. of states}

steps in syn seq ckt design:

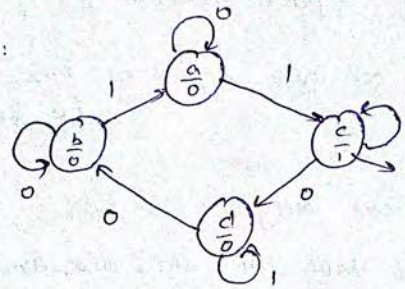
1. Obtain state table (or) state dia, timing dia / pertinent info.
2. Reduce the no. of state using state reduction tech.
3. Assign binary values.
4. Determine no. of FF needed & assign a letter sym to each.
5. choose the type of FF to be used
6. Derive ckt excitation & o/p tables from state table.
7. using K-map, derive o/p ckt fun & FF I/p fun
8. Draw logic dia :

State dia: It is a pictorial rep of a behaviour of a p.s i/p o/p → o/p N.S seq ckt.



Mealy ckt.

Moore ckt: Olp doesn't depend on i/p.



State table: Mealy:

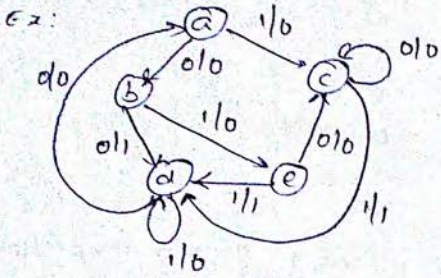
P.S	N.S		olp.	
A B	x=0	x=1	x=0	x=1
0 0	a	c	0	0
0 1	b	a	0	0
1 0	d	c	0	1
1 1	b	d.	0	0

Moore:

P.S	N.S		olp.
A B	x=0	x=1	
0 0 = a	a	c	0
0 1 = b	b	a	0
1 0 = c	d	c	1
1 1 = d	b	d.	0

State Reduction:

Ex:



State dia.

State table:

1)

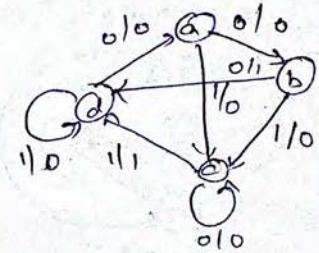
P.S	N.S		O/P	
AB	x=0	x=1	x=0	x=1
a	b	c	0	0
b	d	e	1	0
c	c	d	0	1
d	a	d	0	0
e	c	d	0	1

2) State reduction table:

P.S	N.S		O/P	
AB	x=0	x=1	x=0	x=1
a	b	c	0	0
b	d	e=c	1	0
c	c	d	0	1
d	a	d	0	0

(Repetition can be eliminated)

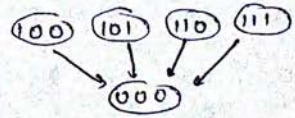
3) Reduced state dia:



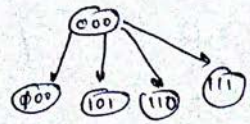
4) State Assignment: (State dia with binary states).

Rules: 1. Ex:

a = 00
b = 01
c = 10
d = 1



2.



The state 100, 101, 110, 111 can be grouped into logically adjacent cells in K-map,

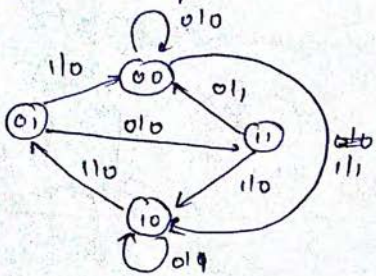
5) State table with assigned states.

a: 00
b: 01
c: 10
d: 11

P.S	N.S x=0	N.S x=1	o/p x=0	o/p x=1
00	b = 01	c = 10	0	0
01	d = 11	c = 10	1	0
10	c = 10	d = 11	0	1
11	a = 00	d = 11	0	0

6) Choose FF & derivation of N.S & o/p Expressions:

Ex2: A seq ckt has i/p & o/p. The state dia is

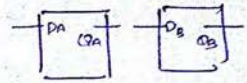


design a seq ckt with
a) DFF b) TFF c) RS FF
d) JKFF

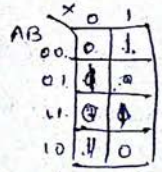
1. State table:

P.S AB	N.S x=0 AB	N.S x=1	o/p x=0	o/p x=1
00	00	10	0	1
01	11	00	0	0
10	10	01	1	0
11	00	10	1	0

using DFF:



for FF A:



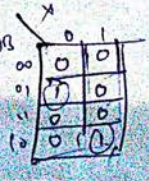
$A = \bar{A}Bx + \bar{A}B\bar{x} + A\bar{B}x + A\bar{B}\bar{x}$

000 = 0
010 = 0
100 = 1
000 = 1
101 = 1
001 = 0
011 = 0
101 = 0

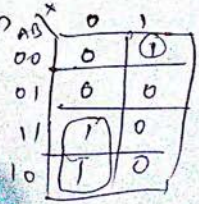
[State table shows that ckt goes through 4 states ∴ 2 FF
no of states = 2^m
m = no. of FF)

000 = 0
001 = 1
010 = 0
011 = 0
100 = 1
101 = 0
110 = 1
111 = 0

for B:



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

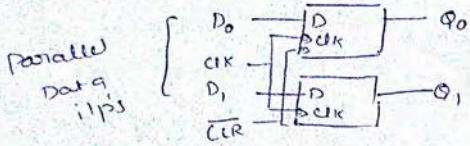


o/p: $A\bar{x} + \bar{A}\bar{B}x$

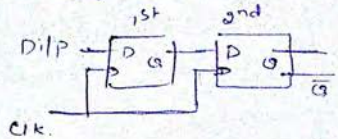
* The situation where the value of Q is ambiguous (ie) change from 0 to 1 & 1 to 0 is known as a "race-around condition".

APP of FF:

- Parallel Data Storage:

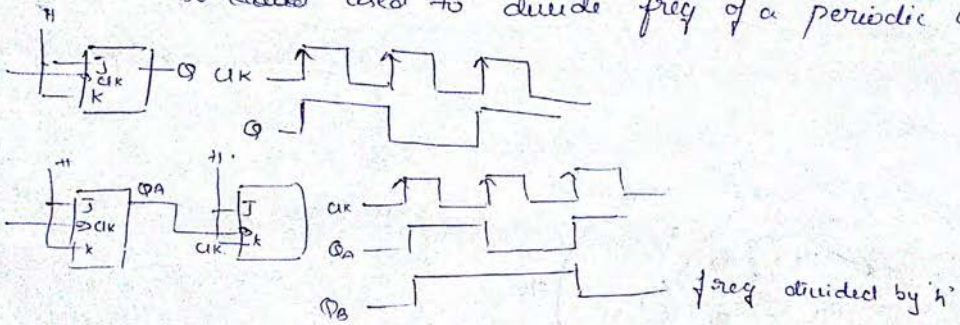


- shift Reg:



Frequency Division

FF are also called used to divide freq of a periodic wave/sm.

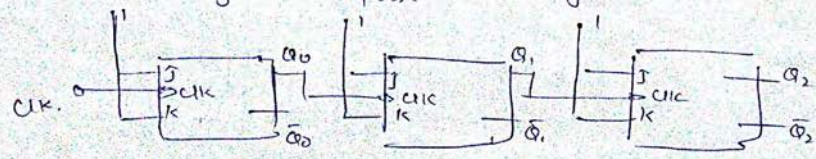


Counters: It operates as a binary counter.

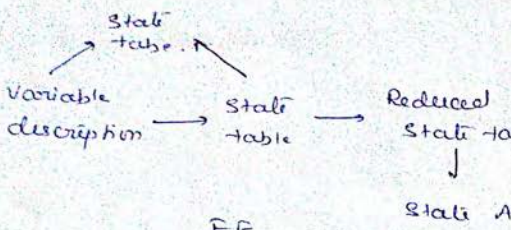
* FF are initially RESET.
* for 3 bit data => 000 to 111 the sequence flows.
* After first clk pulse FF are in 001

  seventh clk pulse FF = 111

  eighth clk pulse FF again is at 000

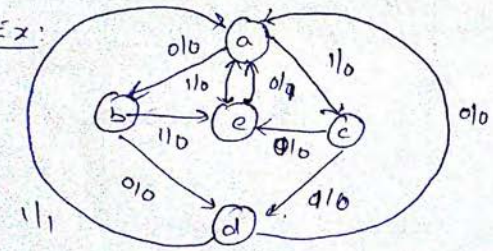


- Steps in synchronous sequential circuit design:



1. word statement.
2. state di (state graph).
3. state table.
4. Reduced state table.
5. State Assignment & Transition table.
6. choose type of FF & form excitation table.
7. Excitation maps & logic fun
8. Draw ckt for the machine.

Ex:



1. state table:

P.S	N.S		o/p.	
	x=0	x=1	x=0	x=1
a	b	e	0	0
b	d	e	0	0
c	d	e	0	0
d	a	a	0	1
e	a	a	1	0

State Assignment:

let a: 000
b: 001
c: 010
d: 011
e: 100
f

P.S			i/p	N.S			o/p.
A_n	B_n	C_n	x	A_{n+1}	B_{n+1}	C_{n+1}	
a 0	0	0	0	b 0	0	1	0
b 0	0	1	1	e 1	0	0	0
c 0	1	0	0	d 0	1	1	0
d 0	1	1	1	a 0	0	0	1
e 1	0	0	0	a 0	0	0	0
f 1	0	1	1				
1	1	0	0				
1	1	1	1				

Excitation table :

	P.S $A_n B_n C_n$	i/p x	N.S $A_{n+1} B_{n+1} C_{n+1}$	o/p Y
a	0 0 0	0b	0 0 1	0
	0 0 0	1c	0 1 0	0
b	0 0 1	0d	0 1 1	0
	0 0 1	1e	1 0 0	0
c	0 1 0	1d	0 1 1	0
	0 1 0	0e	1 0 0	0
d	0 1 1	0a	0 0 0	0
	0 1 1	1a	0 0 0	0
e	1 0 0	0a	0 0 0	0
	1 0 0	1a	0 0 0	0

General K-map :

A_{n+1} : Cx \ AB	00	01	11	10
00	0	0	1	0
01	1	0	0	0
11	x	x	x	x
10	0	0	x	x

$A_{n+1} = Cx\bar{B} + \bar{C}\bar{x}B$.

$B_{n+1} = \bar{A}\bar{C}x + \bar{B}Cx$

$C_{n+1} = \bar{A}\bar{B}\bar{x} + B\bar{C}x$.

o/p : $Y = A\bar{x} + BC\bar{x}$

state d & e } must be adjacent
c & b. }

(1) d 011 (3) (N.S) 001 (b) 010 (c) 100 (N.S) e(4)

001 b(1) 010 (p.s) c(2)

a - 000
b - 001
c - 011 - 3
d - 101 - 5
e - 110 - 7.

2+1 100

p.s (d) 011 100 e.
N.S 000 (a)

000 N.S.
100 110 111 101 p.s

b(1) → 3 → 5

0 1 3 2
4 5 7 6

0 - 000
1 - 001
2 - 010
3 - 011
4 - 100
5 - 101
6 - 110
7 - 111

New excitation table

	P.S $A_n B_n C_n$	i/p x	N.S $A_{n+1} B_{n+1} C_{n+1}$	o/p Y
	0 0 0	0b	0 0 1	0
	0 0 0	1c	0 1 0	0
	0 0 1	0d	ϕ 0 1	0
	0 0 1	1e	1 1 1	0
	0 1 1	ϕd	1 0 1	0
	0 1 1	0e	1 1 1	0
	1 0 1	0a	0 0 0	0
	1 0 1	1a	0 0 0	0
	1 1 1	0a	0 0 0	0
	1 1 1	1a	0 0 0	0

d, e → adjacent
1, 4
001 100

AB \ Cx	00	01	11	10
00	0	0	1	1
01	0	0	ϕ	ϕ
11	x	x	x	x
10	0	0	x	x

$B_{n+1} = \bar{A}B\bar{x} + \bar{A}B\bar{x}$

$C_{n+1} = \bar{A}$

$Y = \bar{A}\bar{B}x + \bar{A}B\bar{x}$.

$A_{n+1} = \bar{A}\epsilon$

Realization of circuit using various FF:

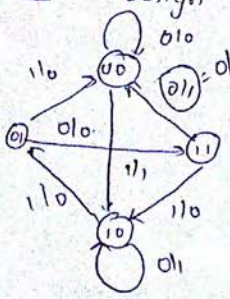
DFF → Transfer of data in shift reg.

TFF → involving complementing. (binary counters).

JK FF → General app.

Ex: Design seq ckt using TFF, DFF & JK FF

1.



2. State table:

P.S	N.S		O/P	
A B	x=0	x=1	x=0	x=1
0 0	00=a	10=c	0	1
0 1	11=d	00=a	0	0
1 0	10=c	01=b	1	0
1 1	00=a	10=c	1=0	0

Excitation table:

	P.S		i/p	N.S		FF i/p's		o/p's
	A	B	x	A	B	T_A	T_B	Y
0 —	0	0	0	0	0	0	0	0
1 —	0	0	1	1	0	1	0	1
2 —	0	1	0	1	1	1	0	0
3 —	0	1	1	0	0	0	1	0
4 —	1	0	0	1	0	0	0	1
5 —	1	0	1	0	1	0	1	0
6 —	1	1	0	0	0	1	1	1
7 —	1	1	1	1	0	1	1	0

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

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

$T_A = \Sigma_d(1, 2, 5, 6)$

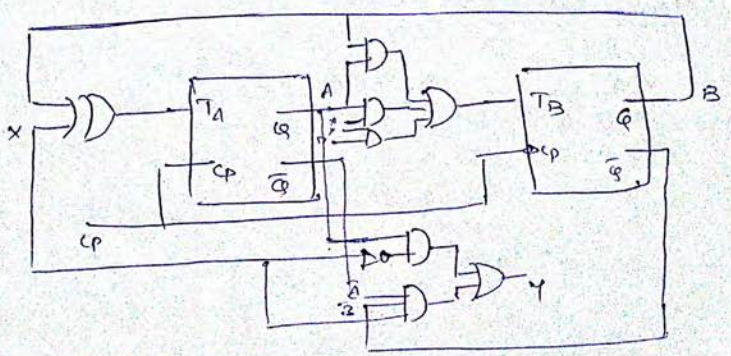
$T_B = \Sigma_m(3, 5, 6, 7)$

$Y = \Sigma(1, 4, 6)$

$T_A = B\bar{x} + \bar{B}x$

$T_B = AB + Ax + Bx$

$Y = A\bar{x} + \bar{A}\bar{B}x$



DFF:

Excitation table:

P.S		I/P	N.S		FF I/ps		o/p's
A	B	x	A	B	D_A	D_B	Y
0	0	0	0	0	0	0	0
0	0	1	1	0	1	0	1
0	1	0	1	1	1	1	0
0	1	1	0	0	0	0	0
1	0	0	1	0	1	0	1
1	0	1	0	1	0	1	0
1	1	0	0	0	0	0	1
1	1	1	1	0	1	0	0

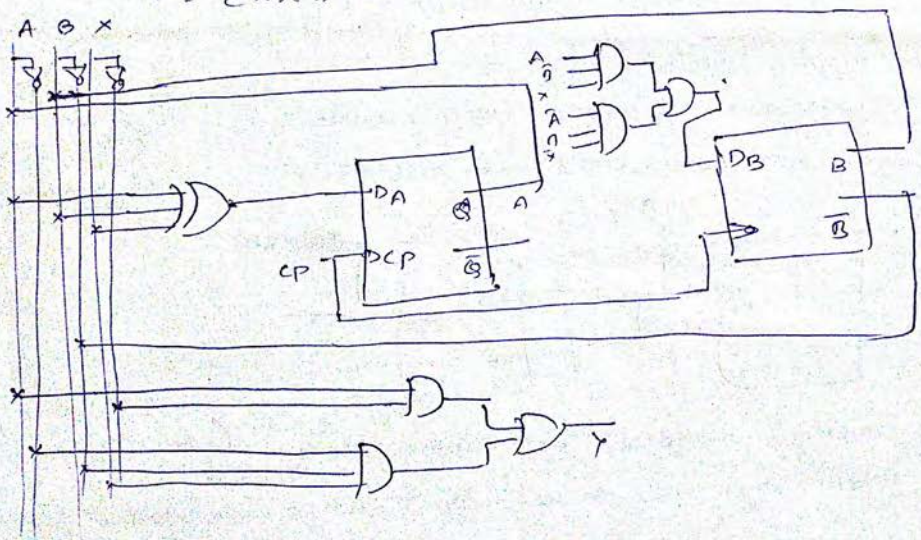
D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

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

$D_A = \Sigma m(1,2,4,7) = A \oplus B \oplus x$

$D_B = \Sigma m(2,5) = A\bar{B}x + \bar{A}B\bar{x}$

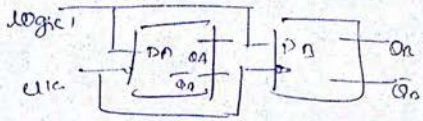
$Y = \Sigma m(1,4,6) = A\bar{x} + \bar{A}\bar{B}x$

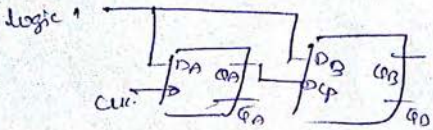


Design of counters: [upon the manner in which FF's are triggered, counters can be classified as]:

Asynchrono-now [ripple] counter
- easy to design
- speed is low.

synchronous [parallel counter)



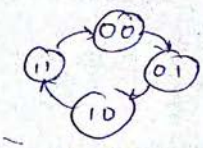


Syn:

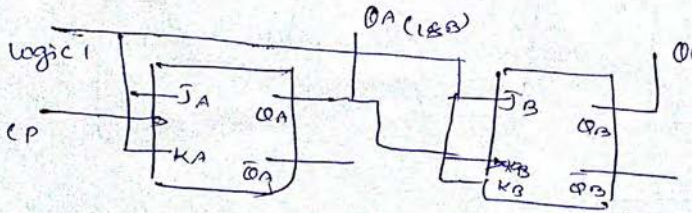
counters
- up counter: If counts such a way that the decimal eqv of O/P ↑ with cp.
- down counter → O/P ↓ with cp.
- up/down counter: any direction.

Asynchronow: (Ripple counter)

- 2 bit Asynchronow up-counter (mod-4 counter)
- It counts from 0 to 3, 0 to 3 - - (∞) 00, 01, 10, 11, 00 - -



State dia



* - consider initially FF = logic 0 state ($Q_A : Q_B = 0$)
$Q_B Q_A$ eg: 00

- when $CP_A = 1$ FFA causes $-Q_A$ to change from 0 to 1

CP_B = this gen sgn at clk i/p FFB

$Q_B Q_A = 0\ 1$ ∴ FFB ⇒ not affected $Q_B = 0$

- At -ve clk pulse edge → $Q_A \rightarrow$ 1 to 0.

$Q_B \rightarrow$ 0 to 1

∴ $Q_B Q_A = 1\ 0$

Illy it continues.

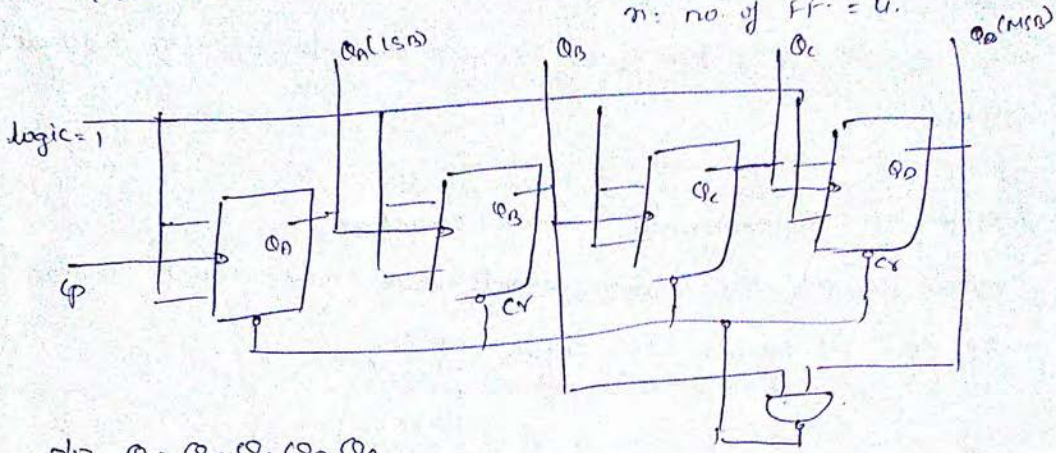
mod-10 ⇒ BCD counter { Asynchronous up. counter } { decade counter } (3)

0000 → 0001 → 0010 → 0011 → 0100 → 0101 → 0110 → 0111 → 1000 → 1001 → 0000

No. of FF ⇒

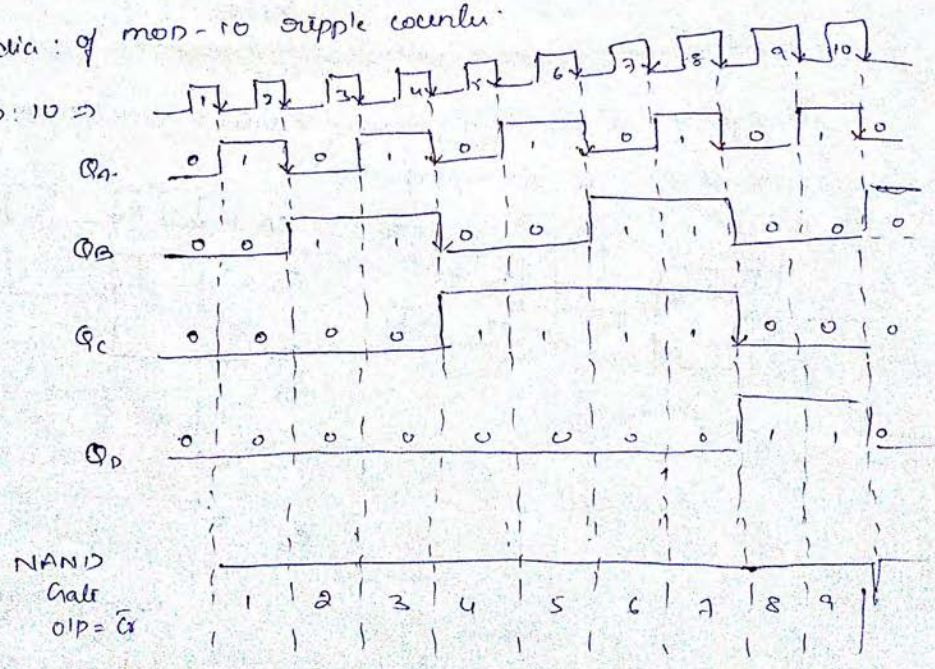
$2^{n-1} < 6 \leq 2^n$

n: no of FF = 4.



o/p $Q = Q_D Q_C Q_B Q_A$.

Timing dia: of mod-10 ripple counter:



o/p ⇒ Q_D Q_C Q_B Q_A

0 0 0 0

0 0 0 1

...

when it reaches to 1010 ⇒ the 10th clk pulse occurs, then the NAND gate o/p goes low & all F.F's are in clear condition.

- when 10th pulse occurs, the counter o/p Q is 0000 instead of 1010.

- The BCD counter is a decade counter ∵ it counts from 0 to 9

- If counting has to be done blw 00 to 99 the two decade counters.

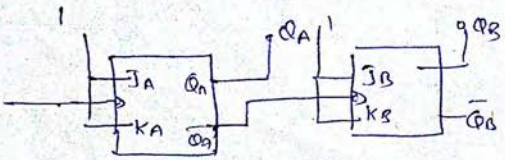
- Design of Asynchronous Down counters:-

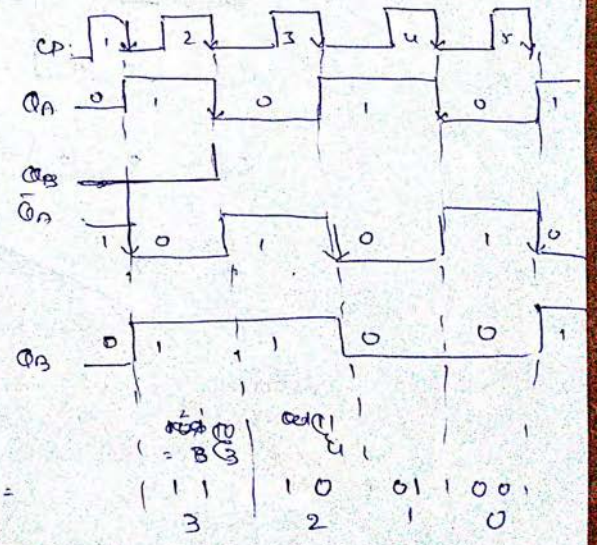
- it counts ~~from~~ down words from max count to zero.

Ex for 3 bit counter -

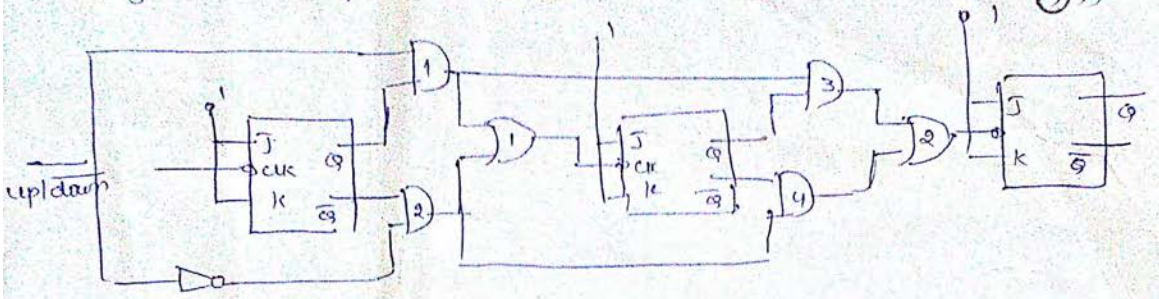
Count	C B A
7	1 1 1
6	1 1 0
5	1 0 1
4	1 0 0
3	0 1 1
2	0 0 1
1	0 0 0

- design of 2-bit Asyn Down-counter: with -ve edge triggered JK FF & its timing dia:





- Asynchronous up/down counter:



A logic 1 on up/$\overline{down}$ counter enables AND gates 1 & 3 & disables 2 & 4

$\therefore Q_A$ = o/p : Q_B ⇒ counter up as pulses are applied.

when low ⇒ counter counts downwards. {down counter}.

propagation delay:

These types (Asy/Ripple) are simplest type of binary counters as they require fewest

adv & disadv of asynchronous counters:

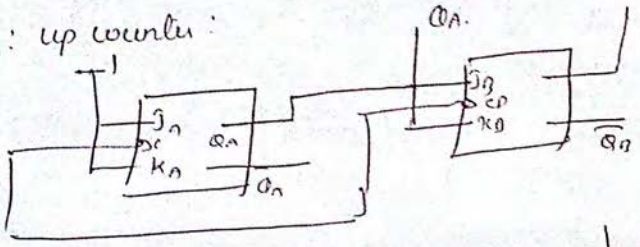
adv: 1. Very simple & straight forward ope
2. requires fewer components
3. easy to design any MOD-counter

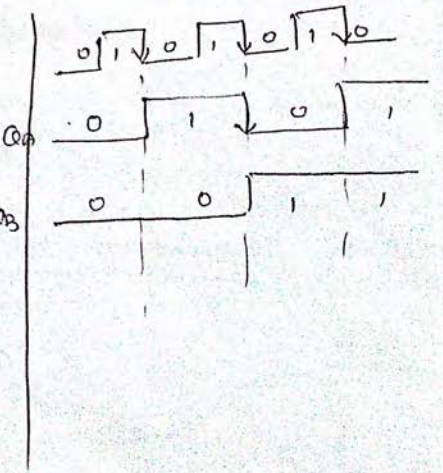
disadv: 1. speed of operation is low.
2. prop delay.
3.

Synchronous counters:

dis of prop delay can be rectified by syn/parallel counters.

2 bit : up counter:





$Q_A = Q_B : 0$

Design of syn counters:

1. Determine no. of FF to be used. [bits]
2. Choose type of FF
3. Draw state dia.
4. use state transitions
5. derive our excitation table
6. K-map for FF functions
7. logic dia.

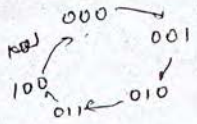
Ex:

- Design a mod-5 syn counter using JK FF & implement it.

1. $2^{n-1} < N \leq 2^n \Rightarrow N = \text{mod } 5 = 5$.

   Consider $n = 3$ = FF's.

2. State dia:



3. State transition table.

P.S			N.S		
Q_C	Q_B	Q_A	Q_{C+1}	Q_{B+1}	Q_{A+1}
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	0	0	0
1	0	1	x	x	x
1	1	0	x	x	x
1	1	1	x	x	x

4. Excitation table:

P.S			N.S			FF i/p's					
Q_C	Q_B	Q_A	Q_{C+1}	Q_{B+1}	Q_{A+1}	J_C	K_C	J_B	K_B	J_A	K_A
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	0	1	1	0	x	x	0	1	x
0	1	1	1	0	0	1	x	x	1	x	1
1	0	0	0	0	0	x	1	0	x	0	x
1	0	1	x	x	x	x	x	x	x	x	x
1	1	0	x	x	x	x	x	x	x	x	x
1	1	1	x	x	x	x	x	x	x	x	x

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

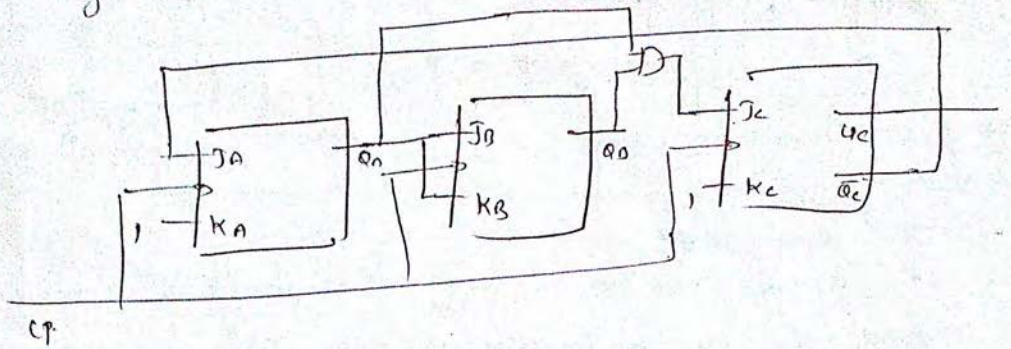
5. K. maps:

$J_C : Q_B Q_A$ $\quad K_C : 1$

$J_B : Q_A$ $\quad K_B : Q_A$

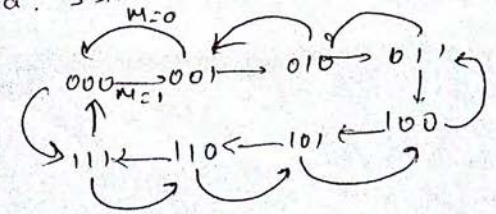
$J_A : \overline{Q_C}$ $\quad K_A : 1$

6. Logic dia:



up / $\overline{down}$ counter (MS). M:1 = Down
M:0 = up.

State dia: 3 bit

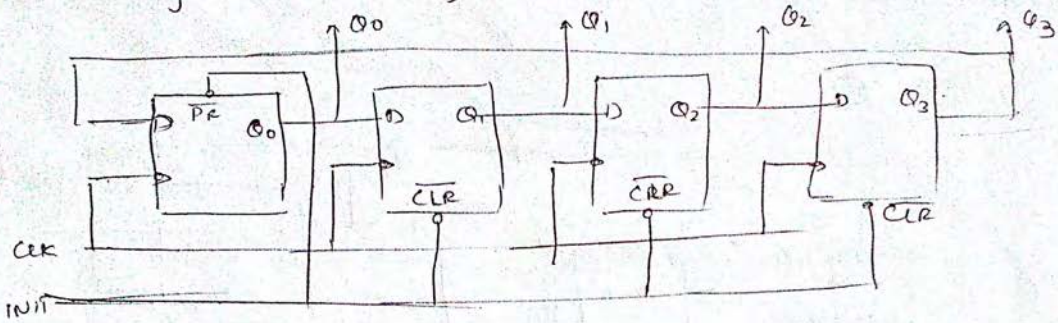


i/p	P.S			N.S			FF i/p's		
up / $\overline{down}$ = M	Q_C	Q_B	Q_A	Q_{C+1}	Q_{B+1}	Q_{A+1}	T_C	T_B	T_A
0	0	0	0	1	1	1			
0	0	0	1	0	0	0			
0	0	1	0	0	0	1			
0	0	1	1	0	1	0			
0	1	0	0	0	1	1			
0	1	0	1	1	0	0			
0	1	1	0	1	0	1			
0	1	1	1	1	1	0			
1	0	0	0	0	0	1			
1	0	0	1	0	1	0			
1	0	1	0	0	1	1			
1	0	1	1	1	0	0			
1	1	0	0	1	0	1			
1	1	0	1	1	1	0			

Shift register counters:

- They use feedback.
- The mostly used FF shift reg counters are:

1. Ring counter 2. Johnson counter.

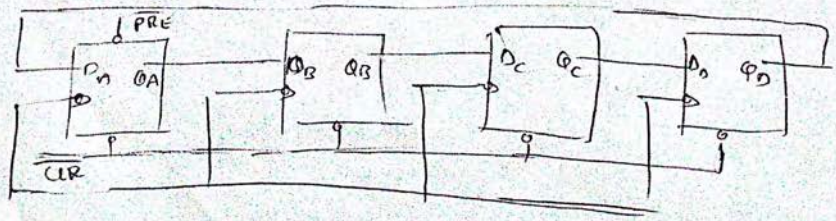


- 4 bit ring counter
- * A single 1 is in the reg & it is made to circulate around the reg as long as clk pulses are applied it's the name ring counter.
- If INIT i/p = set = 1 then $Q_0 = 1, Q_1 = Q_2 = Q_3 = 0 \Rightarrow Q_3 Q_2 Q_1 Q_0 = 0001$

& $Q_0 = 0$ 0010
$Q_1 = 1$ 0100
1000

4T: 4bit Ring counter:

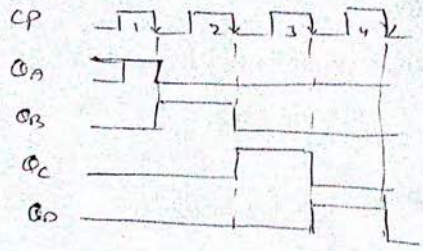
INIT	CLK	Q_3	Q_2	Q_1	Q_0
L	X	0	0	0	1
H	↑	0	0	1	0
H	↑	0	1	0	0
H	↑	1	0	0	0



Ring & shift counter: [MOD-N ⇒ $2^n \geq N$ ⇒ n: no. of FF]

Ring counter:

Timming dia.



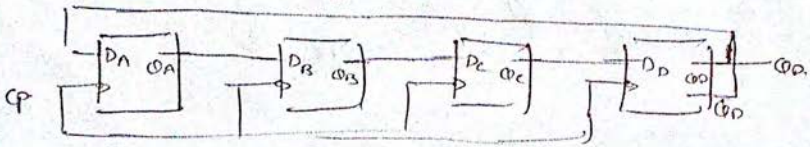
∴ there is 1 pulse at o/p for each of N clk pulses, so it is referred to as divide-by-N counter (or) N:1 scaler.

MOD N ⇒ N: no. of FF's.

adv: speed = fast,

disadv: for n no. of pulses/bit, n FF are required.

Shift / Johnson / Twisted Ring counter: [can be implemented with SR / JK]



— 1st reg = cleared, $Q_A, Q_B, Q_C, Q_D = 0$

∵ $Q_D = 0$, $\overline{Q_D} = 1$

∴ $Q_A = 1$ Q_B Q_C $Q_D = 0$ ∵ $D_B = D_C = D_D = 0$.

next clk pulse produces ⇒ $Q_A = Q_B = 1$

$Q_C = Q_D = 0$

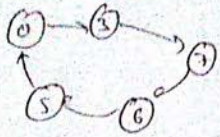
CP	Q_A	Q_B	Q_C	Q_D
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
8				

Applications of counters: Timers, sequence generators, Event counters

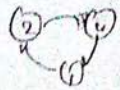
CB: AB: 3, 5, 8, 9, 14, 16, 20, 23, 28, 35, 31, 41, 53, 262, 9,

Note: unused state:

lockout condition: counter:

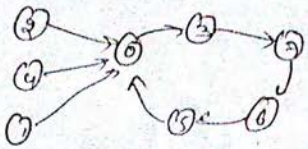


Desired sequence

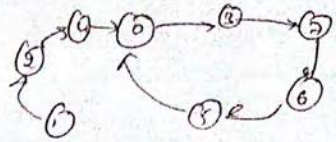


lock out occurring from unused state.

State dia of removing lockout



a

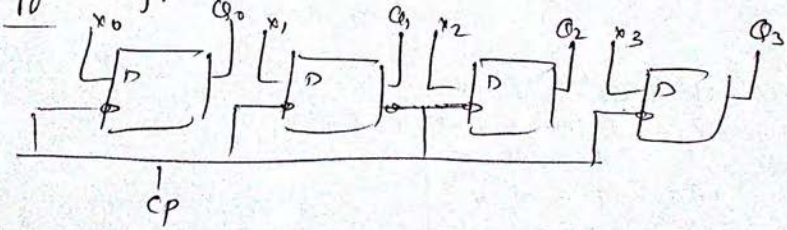


Design of Registers:

- The data can be entered serially / parallel.
- They are classified as - storage / Buffer reg.
  - Shift reg.

(Buffer: Temporary storage elements)

1. Buffer reg:



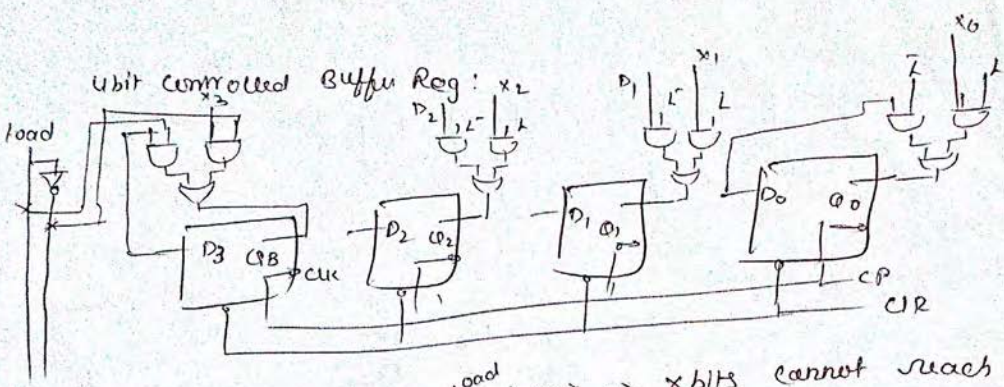
4-bit Buffer reg: $Q_0\ Q_1\ Q_2\ Q_3$: $x_0\ x_1\ x_2\ x_3$.

2. Controlled Buffer reg:

Gates → hold control information

Reg → have bits.

4bit controlled Buffer Reg :



when $load = L$ ckt ⇒ x bits cannot reach FF

Load: H ⇒ x-bits = FF o/p's

$Q_3 Q_2 Q_1 Q_0 = x_3 x_2 x_1 x_0$.

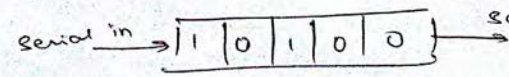
shift reg :

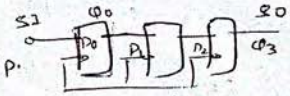
The data movement in reg is concerned with the two char.: 1. How binary inf is loaded into the reg.

2. How to read the data from reg.

- Acc to above reg are classified into

1. Serial In - serial out.

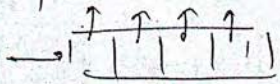




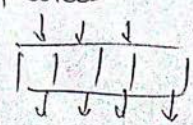
Ex: 1011

- $Q_3 = Q_2 = Q_1 = Q_0 = 0000$
- LSB = 1 ⇒ enter ⇒ $Q_0 = 1, Q_1 = Q_2 = Q_3 = 0$
- ⇒ 2nd bit = 1 ⇒ $Q_0 = 1, Q_1 = 1, Q_2 = 0, Q_3 =$
- ⇒ 3rd clk pulse : $Q_0 = 0, Q_1 = 1, Q_2 = 1, Q_3 = 0$
- ⇒ o/p = 1011

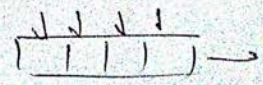
2. Serial In - parallel out :



3. Parallel In - parallel - out

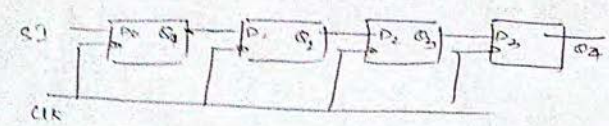


4. Parallel In - Serial out :



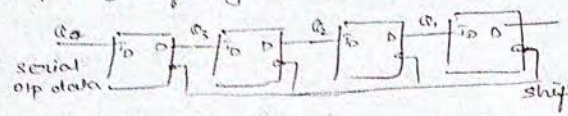
1011

1. SISO



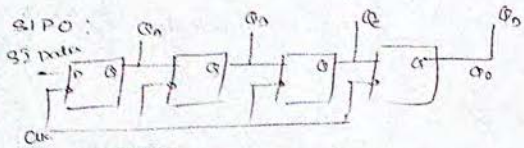
CP	Q_1	Q_2	Q_3	Q_4
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	0	1	1	0
4	1	0	1	1

2. Shift reg: left register

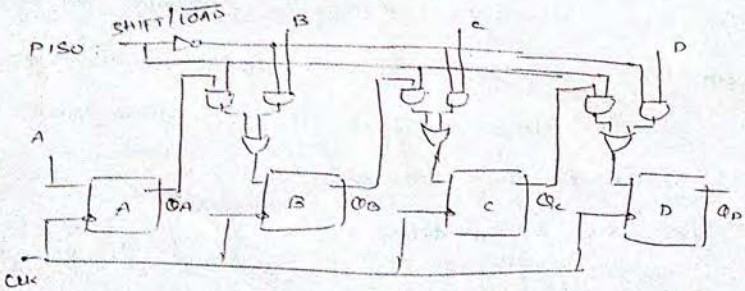


CP	Q_4	Q_3	Q_2	Q_1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	1	0	1
4	1	0	1	1

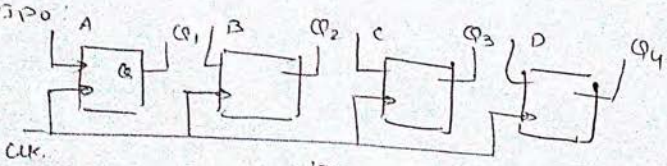
3. SIPO:



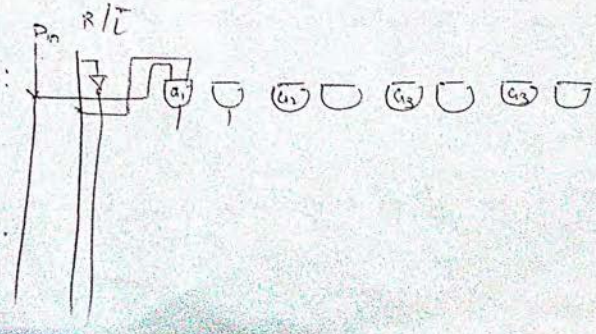
3 PISO:



4. PIPO:



Bi directional shift reg:

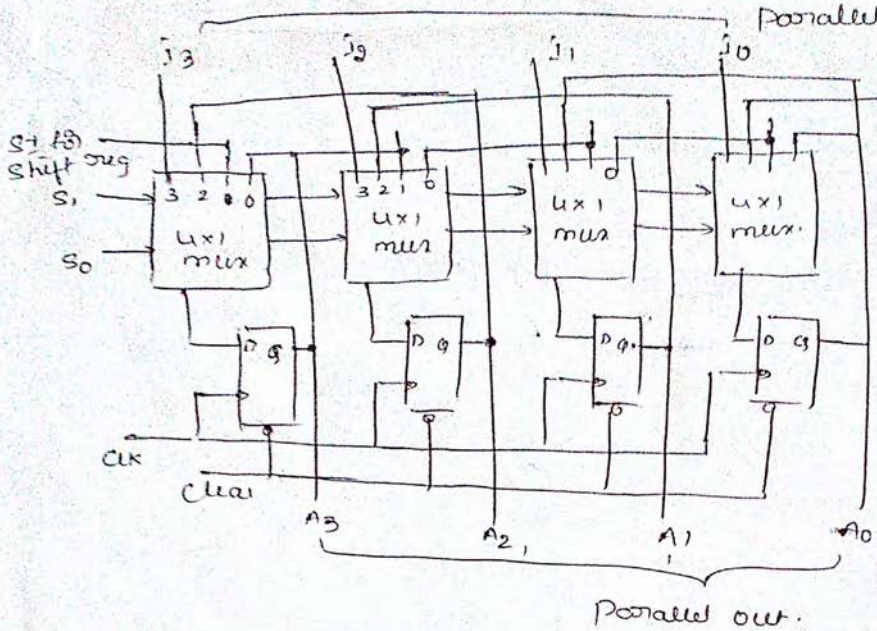


universal shift register : [if reg has both shift and parallel load capability].

Functional table :

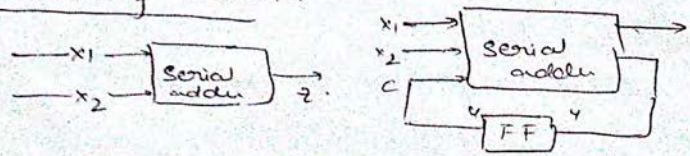
Select line		Data line selected	Register operation
S_0	S_1		
0	0	I_0	Hold
0	1	I_1	shift right
1	0	I_2	shift left
1	1	I_3	parallel load.

- It contains 4 F-F's & 4 mux.
- when $S_1S_0 = 00$
  i/p : 0 => I0ilp of F-F.
  no change in reg
- when $S_1S_0 = 01$,
  i/p: 1 => I1 F-F.
  Operates as shift right reg.
- when $S_1S_0 = 10$
  i/p: 2 =>
  it operate left shift reg.
- when $S_1S_0 = 11$,
  it operate parallel load ope.



Applications of shift reg:

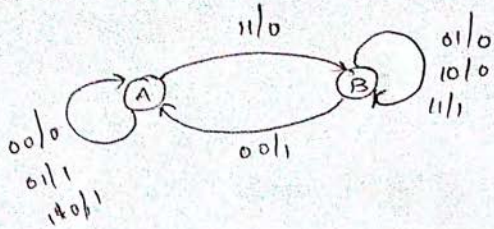
- Time delay :) $\Delta t = n \times \frac{1}{f}$.
- Ring counter
- Serial to parallel data conversion.
- parallel to serial data conversion.
- Serial Binary adder:



State table:

P.S	N.S,				o/p
	$X_1 X_2$				
	00	01	10	11	
A B	$A_{,0}$	$A_{,1}$	$A_{,1}$	$B_{,0}$	
B	$A_{,1}$	$B_{,0}$	$B_{,0}$	$B_{,1}$	

State dia:



Reduced standard form state table : no reduction.

State Assignment & Transition & o/p table :

for states A=0 & B=1, already assigned.

P.S	N.S x_1x_2 = 00	01	10	11	O/P x_1x_2 = 00	01	10	11
0	0	0	0	1	0	1	1	0
1	0	1	1	1	1	0	0	1

Choose type of FF & form Excitation table :

P.S y	I/P x_1	x_2	N.S Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

I/P of FF

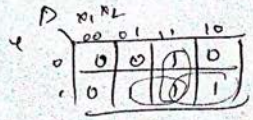
D
0
0
0
1
0
1
1
1

O/P.

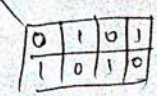
z
0
1
1
0
1
0
0
1

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

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



$D = y\bar{x}_2 + x_1\bar{x}_2 + \bar{y}x_1$



$O/P = y\bar{x}_1\bar{x}_2 + \bar{y}\bar{x}_1x_2 + y x_1 x_2 + \bar{y}x_1\bar{x}_2$

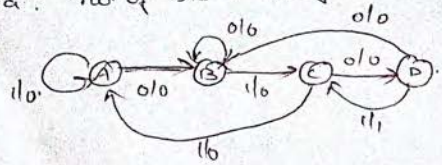
$= x_1 \oplus x_2 \oplus y$

→ sequence detectors : — A detector which detects overlapping ip seq
— Non overlapping ip sequence.

It is clocked sequential ckt used to detect the desired binary sequence.

Ex: detect 0101.

State dia: no. of bits = no. of states, at time T_i = initial state.



State table:

P.S	N.S		O/P	
	x=0	x=1	x=0	x=1
A	B	A	0	0
B	B	C	0	0
C	D	A	0	0
D	B	C	0	1

State Assignment table A=00, B=01 C=10 D=11

State Excitation table.

logic dia: