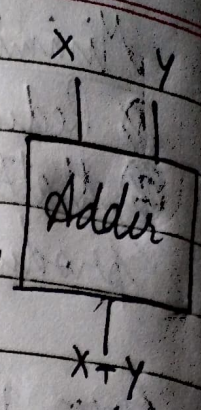
POST MIDSEM

1-1

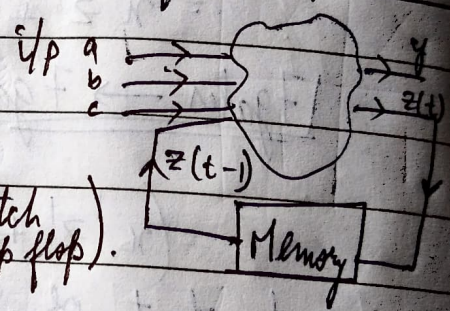


time needed to generate the output is called 'gate propagation delay'.

Sampling output is to be done after the last output is evaluated (longest path — critical path)

Memory elements keep record of the previous output (state of circuit)

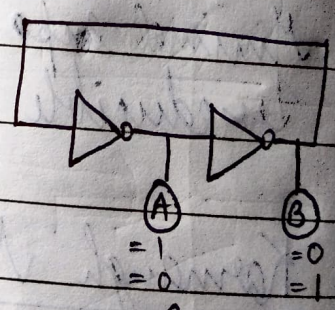
o/p depends on i/p + state.



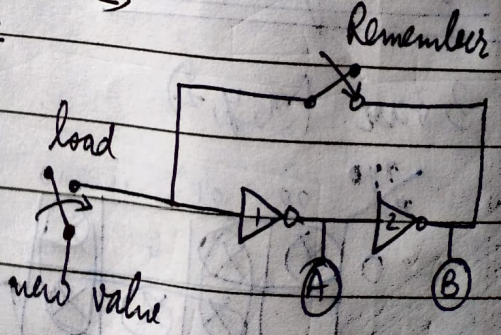
Sequential Circuits (eg:- latch, flip flop).

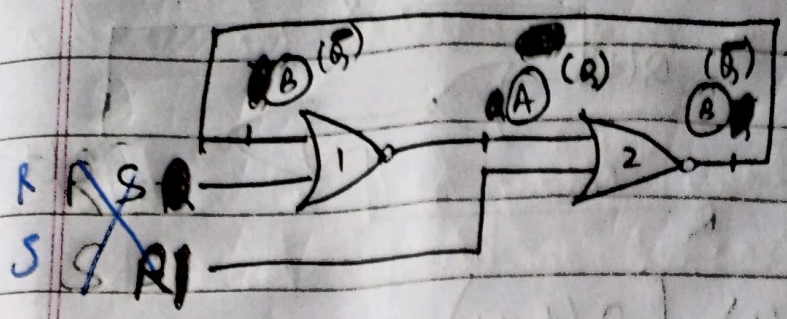
Uses state of circuit.

Make convention like always sample A (or B).



load	remember	store
close	open	new
open	close	old.



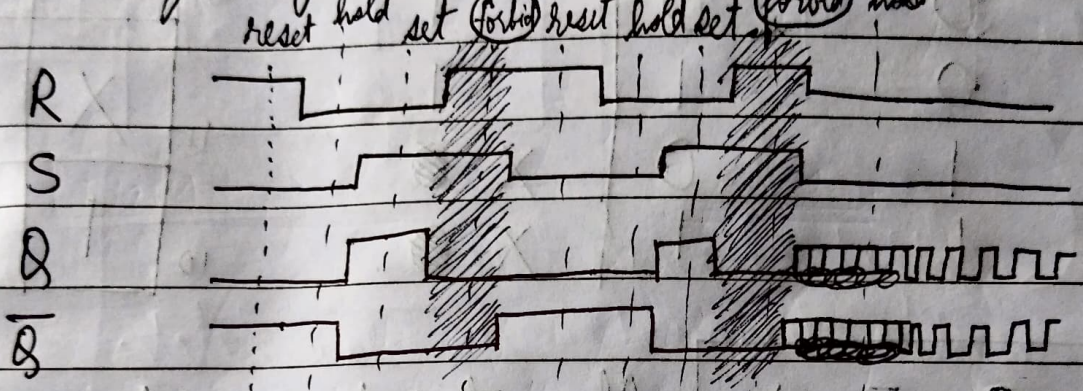


S	R	A	B
0	0	Q	$\bar{Q}$
1	0	1	0
0	1	0	1
1	1	0	0

forbidden input

↳ they are same

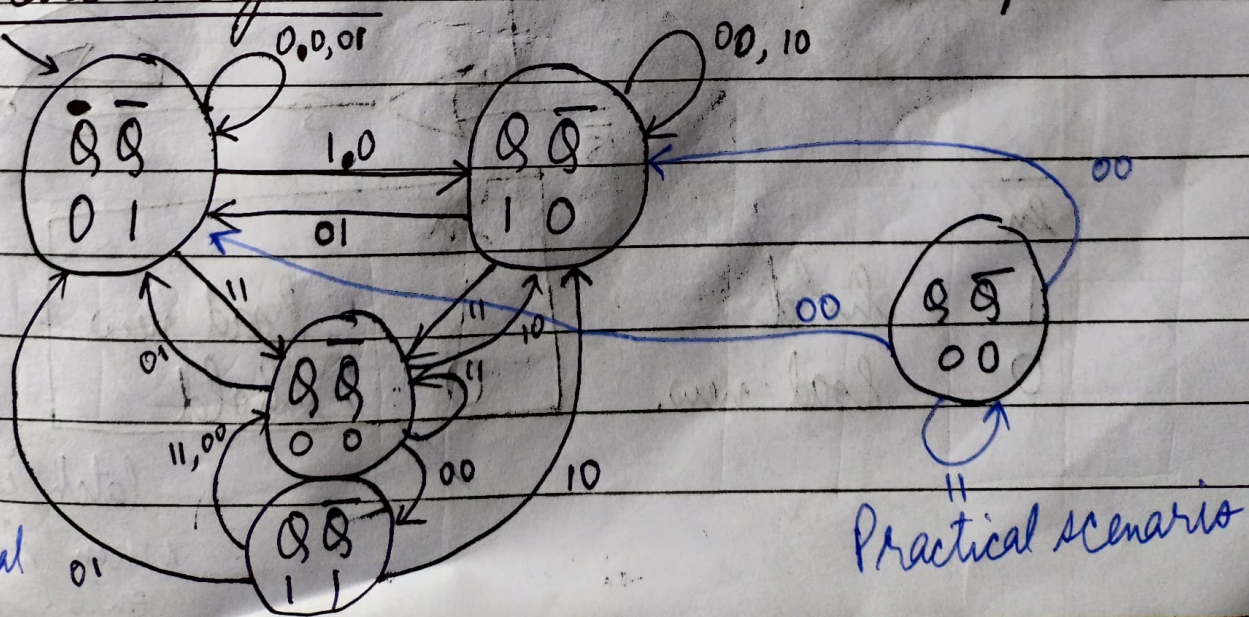
Timing Diagram R-S latch



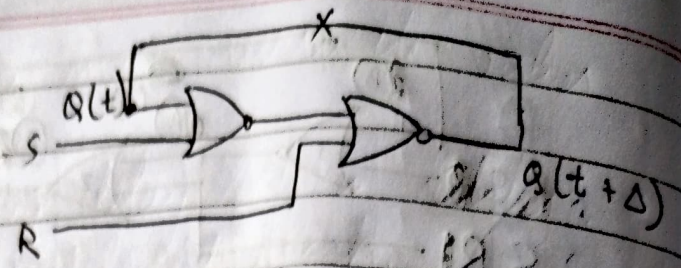
In practical scenarios, latch comes out of this situation ~~due to~~ because the ~~clo~~ delay in the gates is not "exact".

State Diagram

Input → SR



$\overline{\overline{R} + \overline{S}} = R$ em.



S	R	Q(t)	Q(t+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

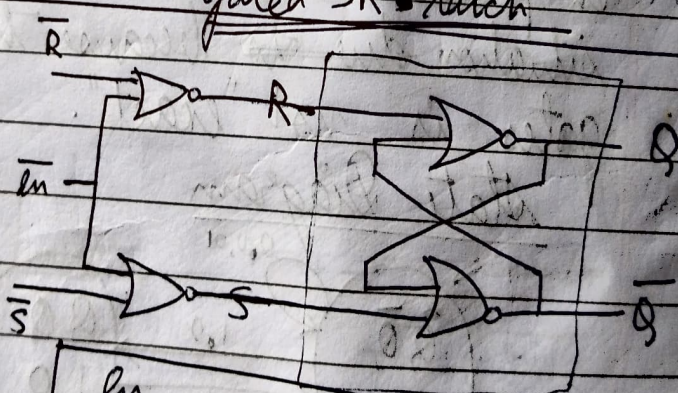
SR \ Q(t)	0	1
00		1
01		
11	X	X
10	1	1

$\boxed{Q(t+\Delta) = S_t + \overline{R}_t Q(t),}$ Characteristic equation of S-R latch.

gated SR latch

$\overline{en}$. active low

$\overline{en}$ = asserted = 1



$\overline{en}$	
1	hold
0	load new.

en	
1	load new
0	hold

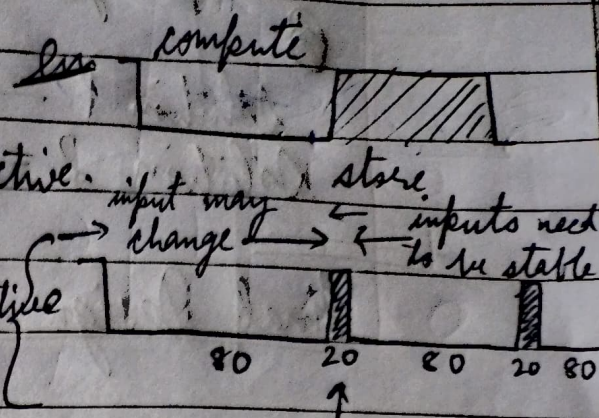
level sensitive latch

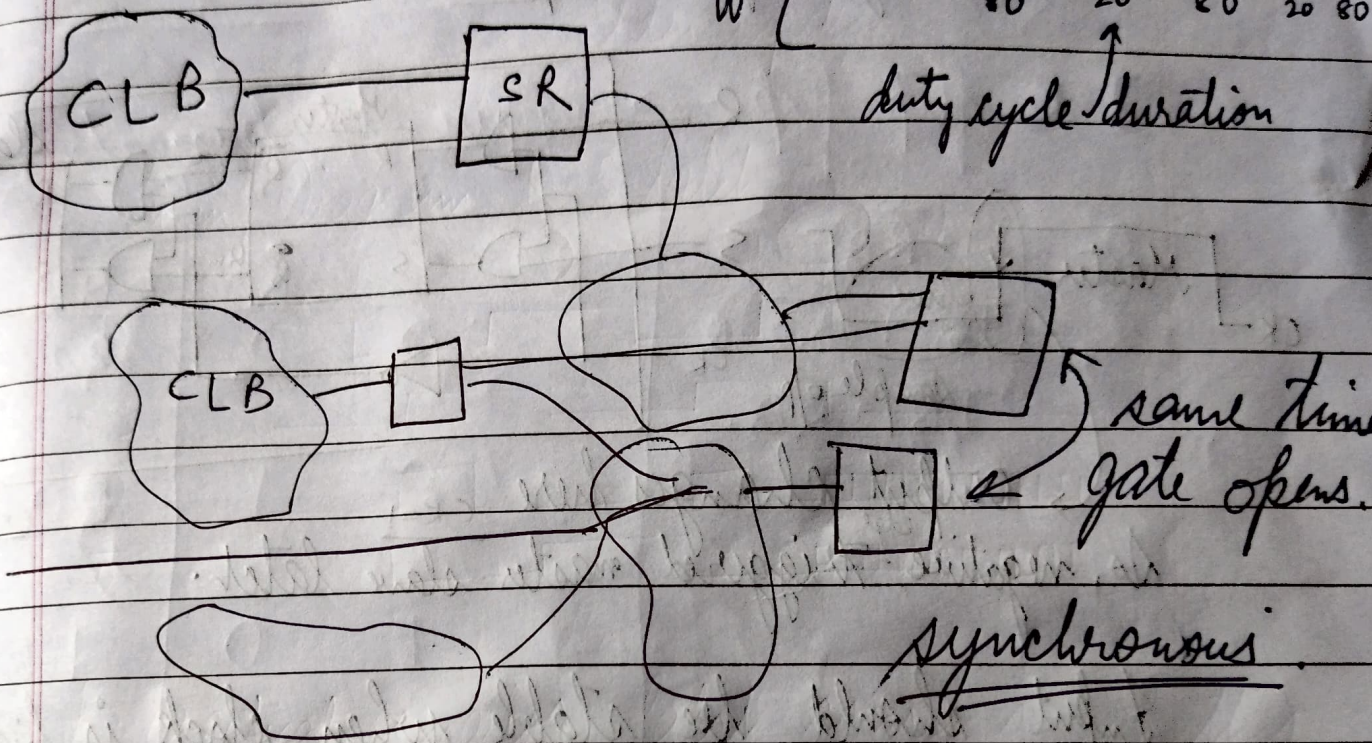
latch is sensitive ~~with~~ when clock is at high level only.

Clock Period

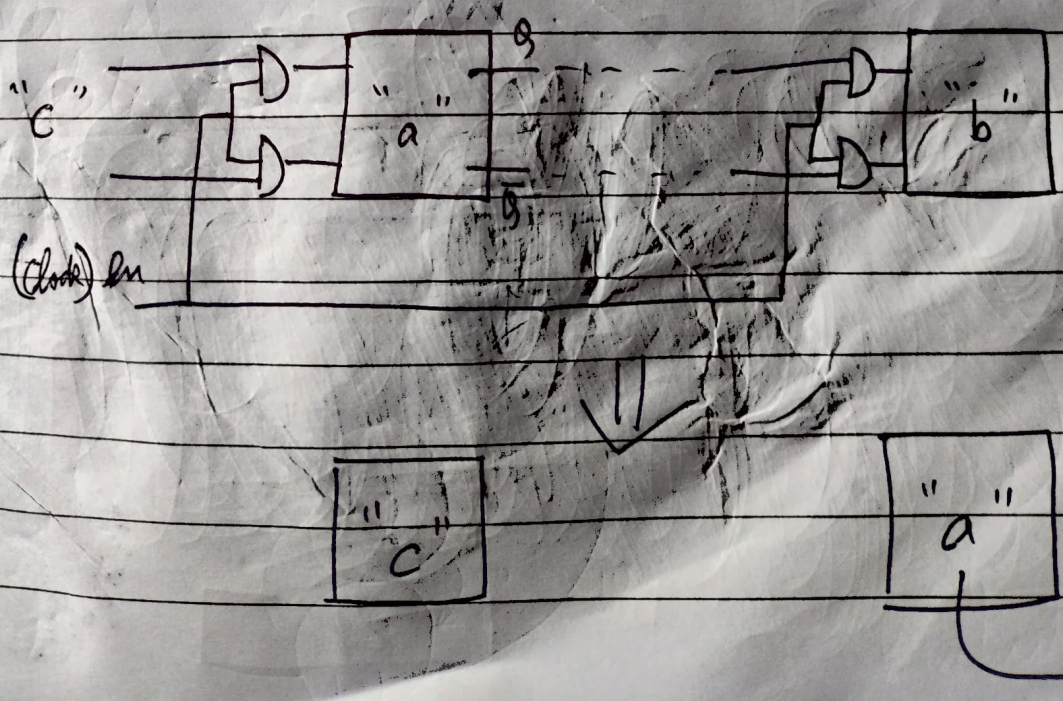
Time to compute

Duty cycle = % time ck effective.





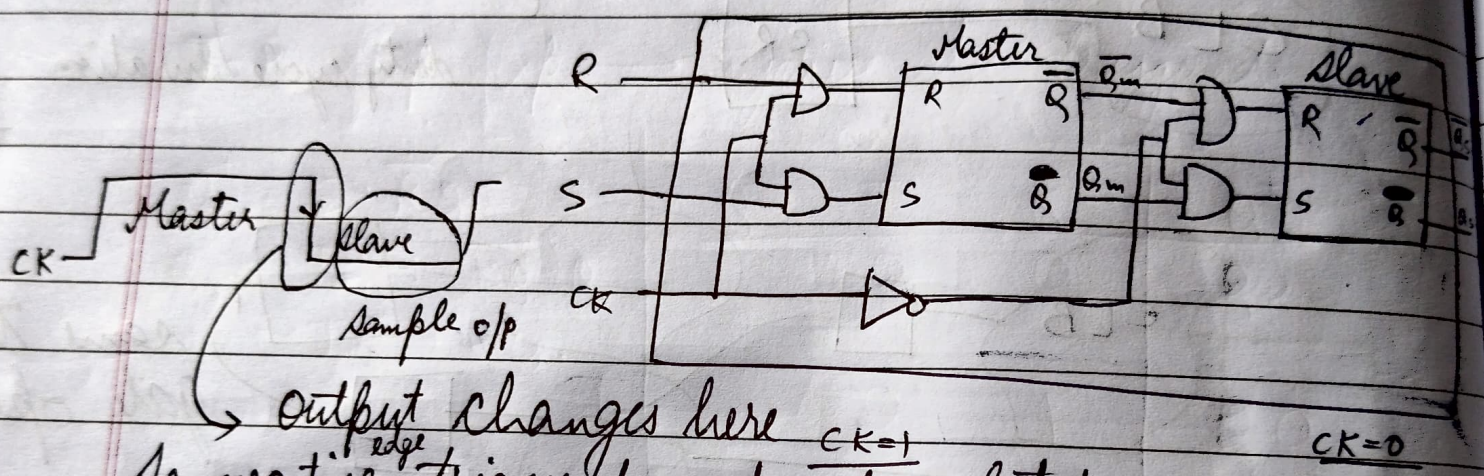
synchronous.



Connecting latches may overwrite values

Solution 1 → long delay between latches.
Solution 2 → complement clock.

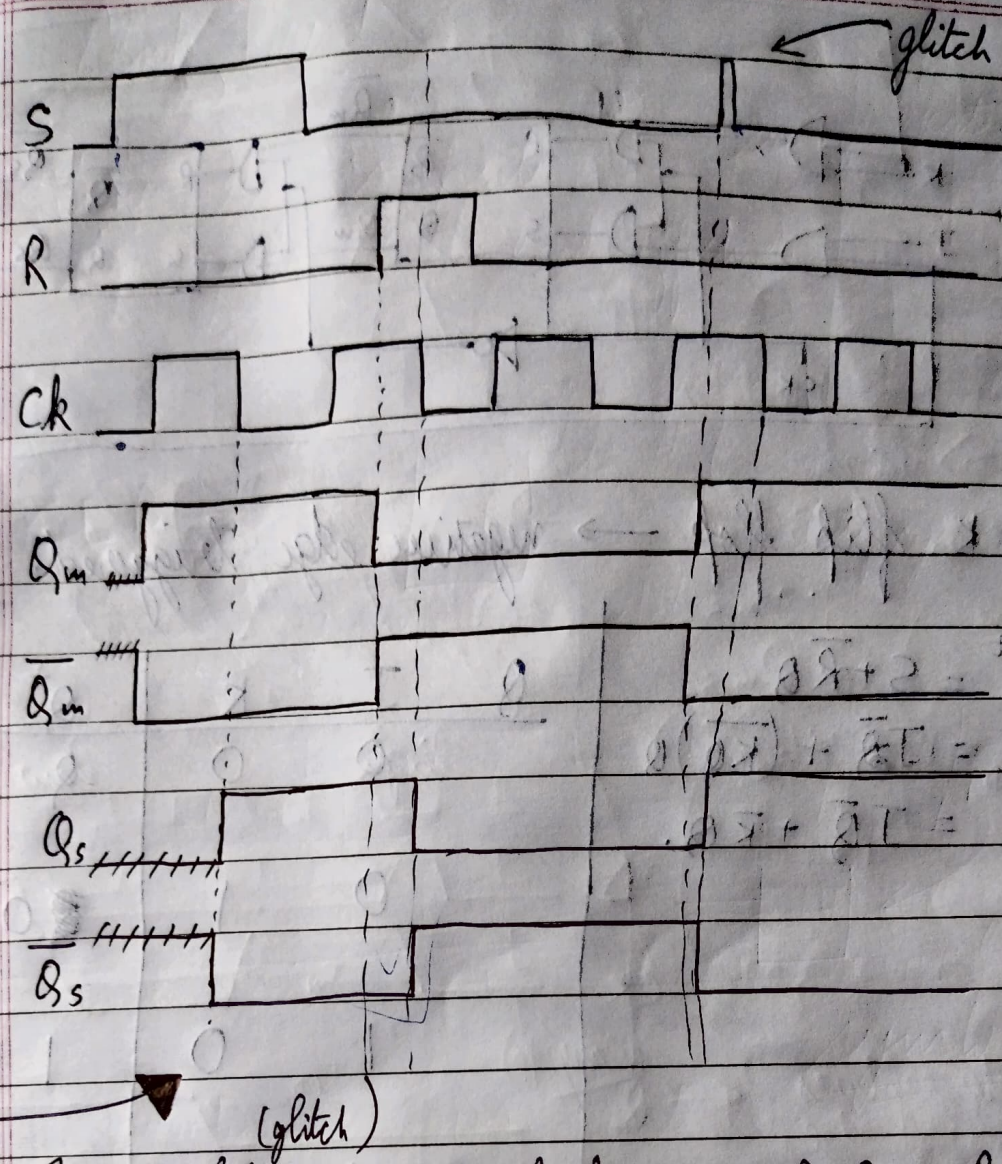
Final Design for 2 inputs –



→ output changes here

So, negative edge-triggered master-slave latch.

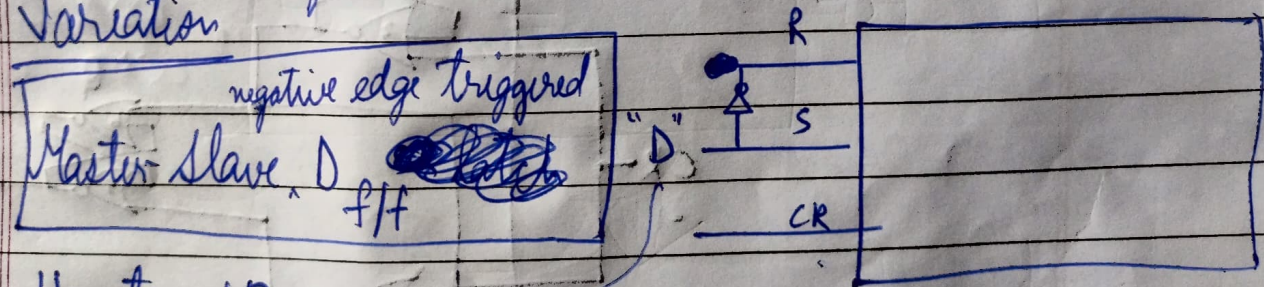
Input should be stable when clock is high so that master can capture value.



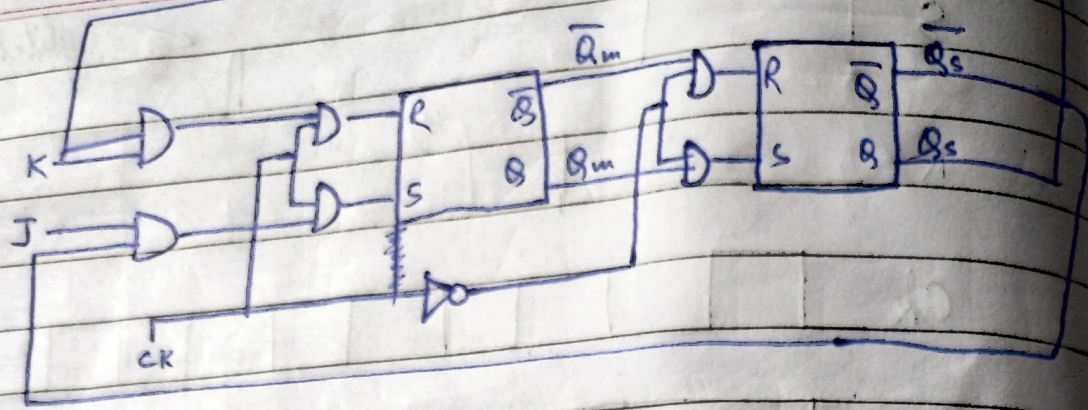
If problem is just before S=1, R=0 or R=1, S=0, it would rectify itself auto.

Problem only when CK=1, R=0, S=0

Variation



#gates = 10



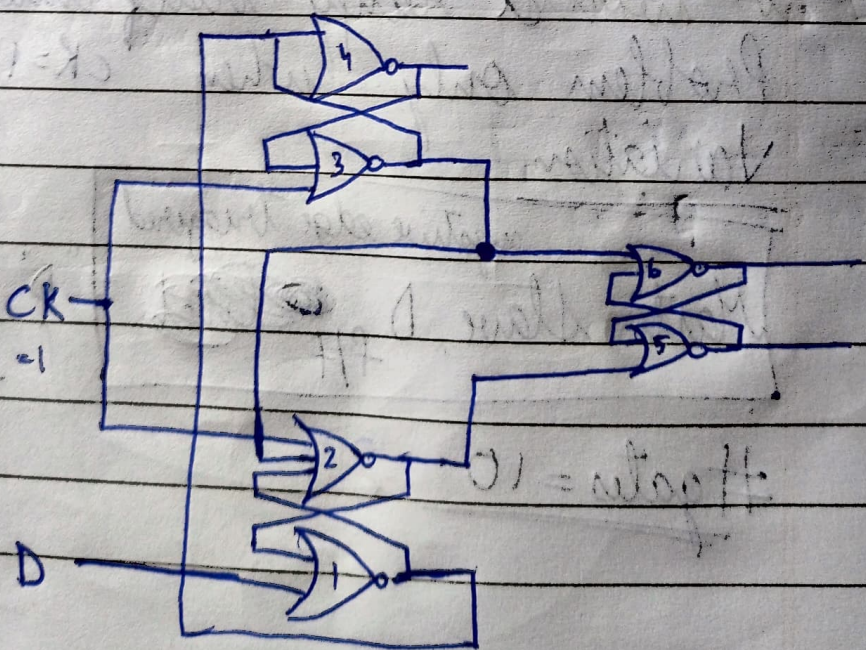
J-K flip flop → negative edge triggered:

$Q^+ = S + \overline{R}Q$

$Q^+ = J\overline{Q} + (\overline{KQ})Q$

$= J\overline{Q} + \overline{K}Q.$

Q	J	K	Q^+	
	0	0	Q	Hold
	0	1	0	Reset
	1	0	1	Set
	1	1	$\overline{Q}$	Toggle



classmate
Date
Page

CK=1, "D"

gate 2,3 → o/p = 0 } **hold**

gate 1, 4 → inverters

1 → $\overline{D}$

4 → D } sample D

CK = 0

2, 3 → inverters

3 → $\overline{D}$

2 → D } 'D' given to RS

$Q = D$

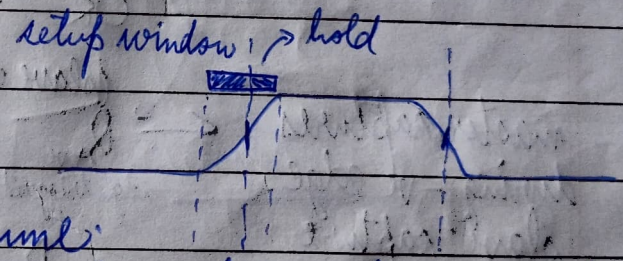
$\overline{Q} = \overline{D}$

CK = D, D = D_{new}

1 → 0

4 → D (no change) } No change in RS.

Clock in real life

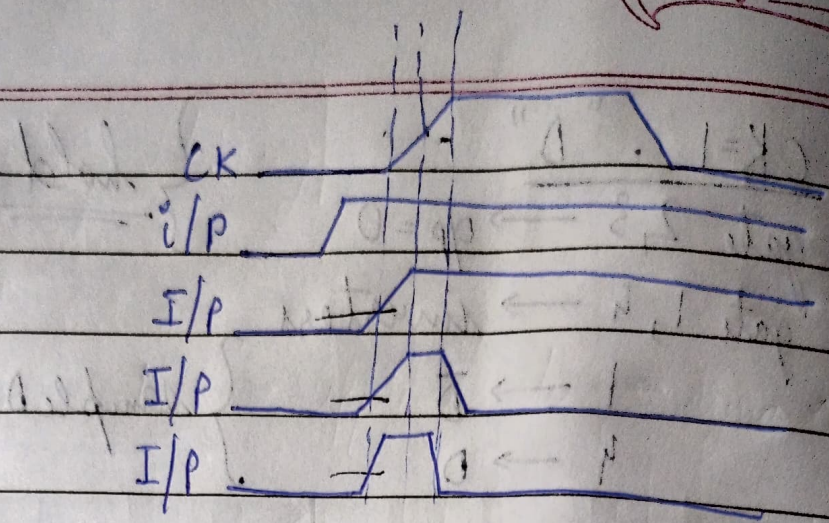


setup time

- t_{su}, min time ~~travel~~ interval immediately preceding the active clock transition during which the inputs should be maintained stable to be validly recognised.

hold time

- t_h, " " " " immediately ~~preceding~~ after



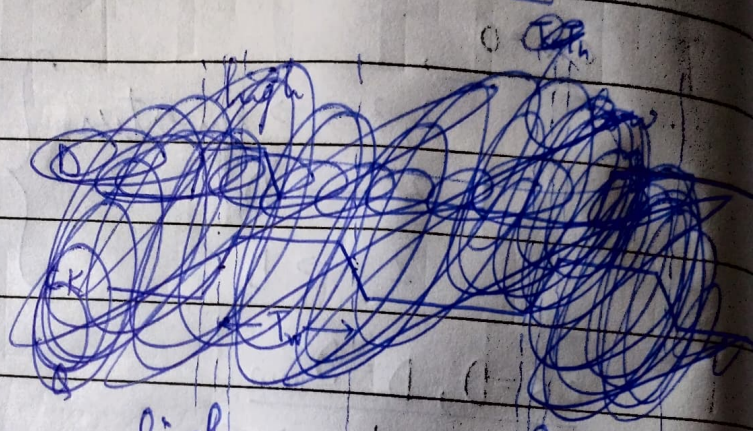
D ⇒ high

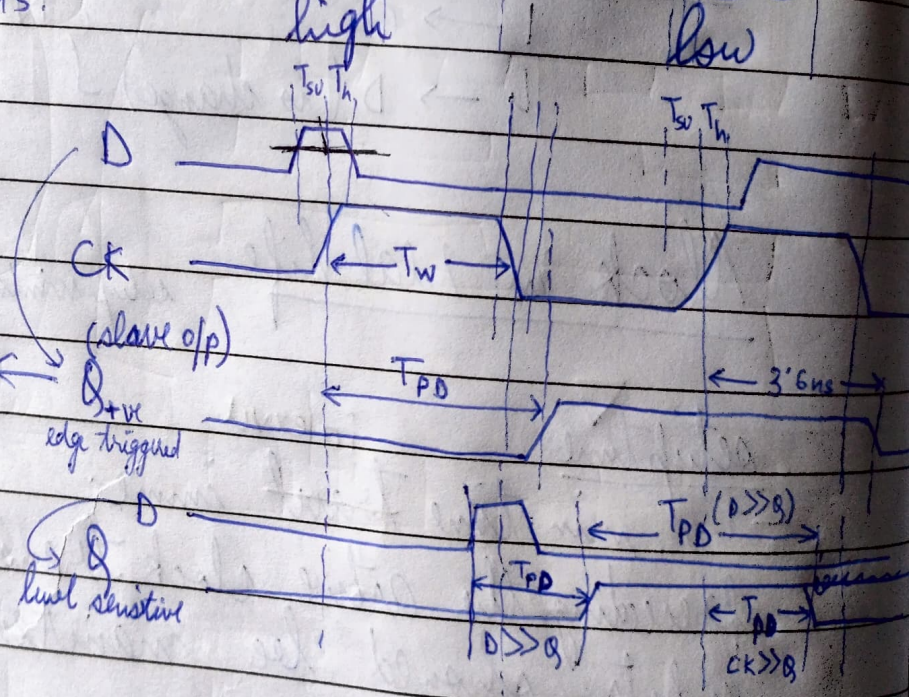
$T_{width} = 3.3\ ms$

$T_{sw} = 1.8\ ms$

$T_h = 0.5\ ms.$

$T_{PD} = 1.1 - 3.6\ ms.$

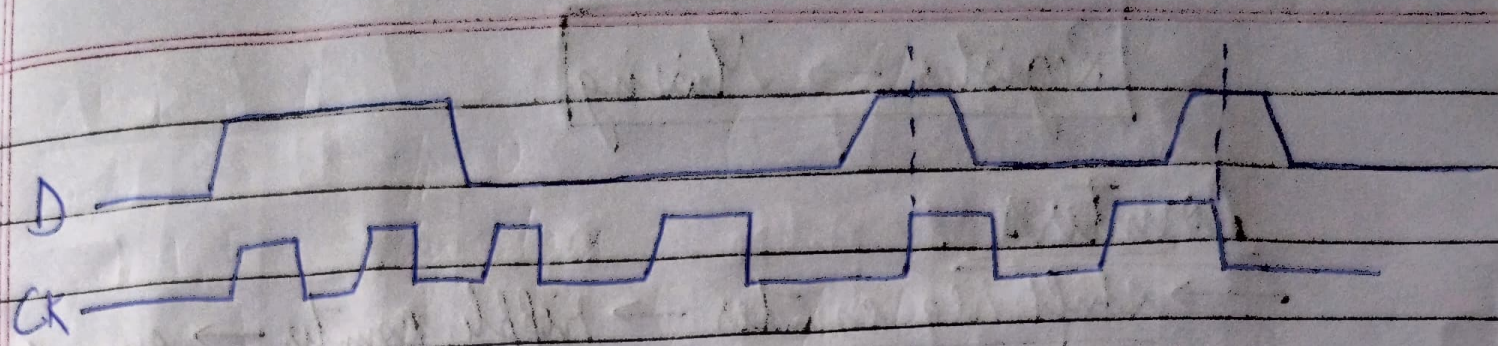




master captures during -ve edge, slave capture during +ve edge.

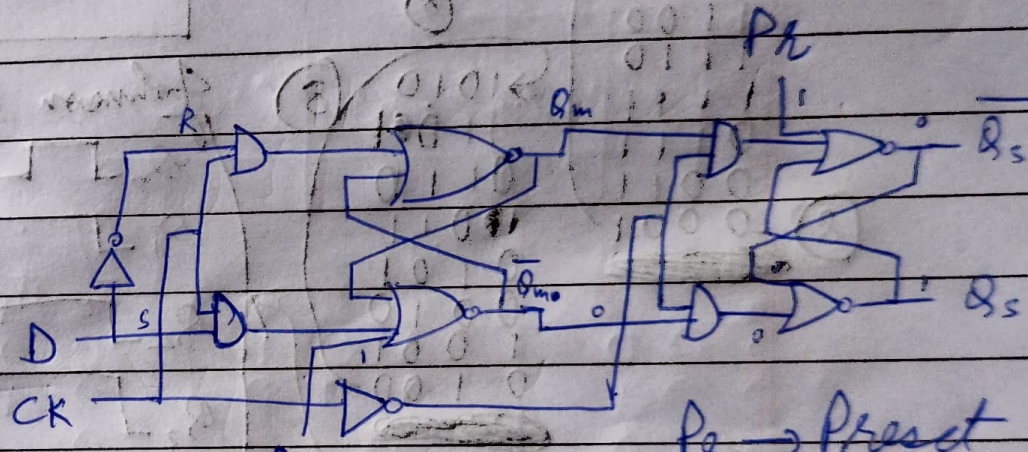
T_{PD} D>>Q 1.0 - 3.6 ms

CK>>Q 1.0 - 3.3 ms



Q = FF

Q = latch



Pr → Preset

synchoroms and asynchronous preset :

If preset = 1,
Qs = 1.

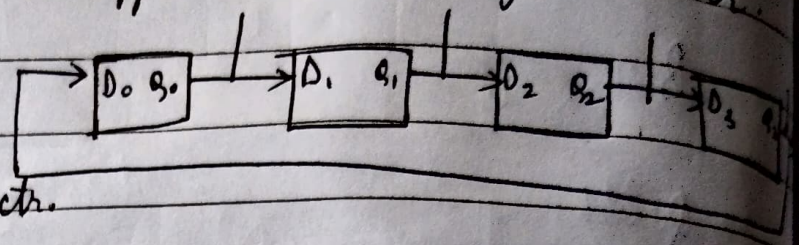
Register → Missing.

Counters.

→ shift reg counter → ripple counter → sync counter.

1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1

4 states



Ring ctr.

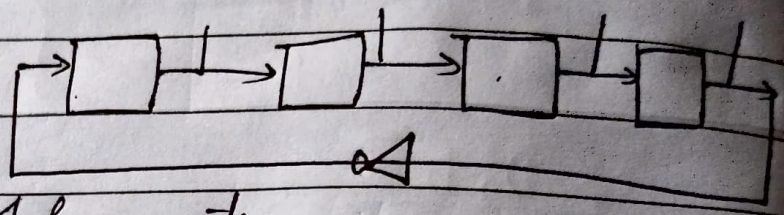
states = 1 → 4.

0000
1000
1100
1110
1111
0111
0011
0001

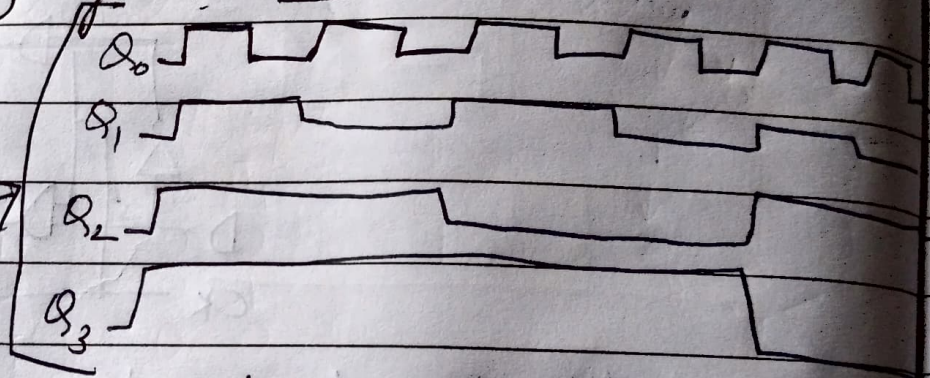
8

1000
1010
1101
0110
1011
0101
0010
1001
0100

8

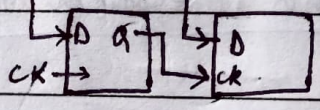


Johnson ctr.



Ripple counter

4-bit up counter



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

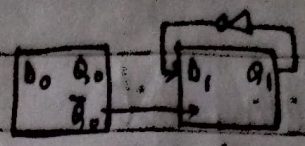
Q_0 1 → 0

⇓

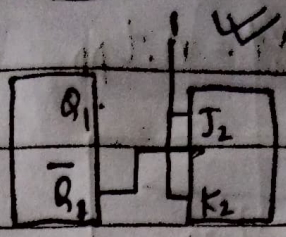
Toggle Q_1.

-ve edge triggered

If only +ve edge triggered f/f available, use $\overline{Q}$ in register.



J-K flip-flop is better as toggle is already available



	+ve edge	-ve edge
up-ctr	$\bar{Q}_i$ to CK_{i+1}	Q_i to CK_{i+1}
down-ctr	Q_i to CK_{i+1}	$\bar{Q}_i$ to CK_{i+1}

Decade up-counter

$\boxed{\rightarrow 0 \rightarrow 1 \rightarrow \ldots \rightarrow 9}$ 1010

$Q_3 Q_1 \longrightarrow CLR$.

$CLR_0 = CLR_1 = CLR_2 = CLR_3 = Q_3 Q_1$

$P_0 = P_1 = P_2 = P_3 = 0$.

Excess-3 up-counter

$\boxed{\rightarrow 3 \rightarrow 4 \rightarrow \ldots \rightarrow 12}$

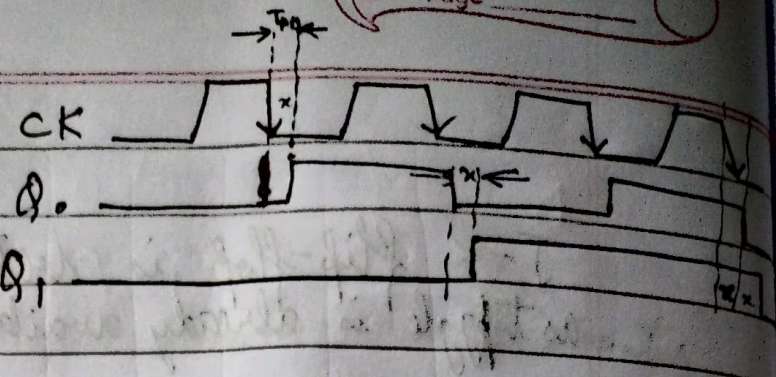
1101
↓
0011

$\boxed{C_3 = C_2 = Q_3 Q_2 Q_0 = P_1 = P_0}$

0010
↓
1100

down-counter $\longrightarrow$

$C_1 = C_0 = \bar{C}_3 \bar{C}_2 \bar{C}_0 = P_3 = P_2$.



Total delay at final output $= n.t_{PD}$.

$$\boxed{f_{max} = \frac{1}{N.\ t_{PD}}}$$

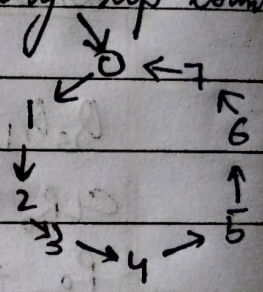
4-bit

$T = 1000$ ns → CK time period → 100 ns.

$T_{PD} = 50$ ns → 50 ns

Synchronous counters (3-bit binary up-counter)

PS Q_2	PS Q_1	PS Q_0	NS Q_2	NS Q_1	NS Q_0	f/f inputs T_2	T_1	T_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



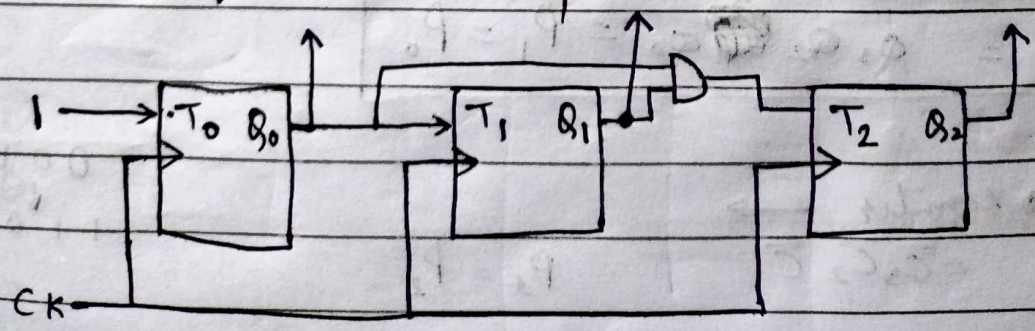
Using D f/f,

$D_0 = \overline{Q_0}$

$D_1 = \overline{Q_1}\,Q_0 + Q_1\overline{Q_0}$

$D_2 = \overline{Q_2}\,Q_1 Q_0 + Q_2(\overline{Q_1} + \overline{Q_0})$

$= \overline{Q_2}\,Q_1 Q_0 + Q_2\overline{Q_1} + Q_2\overline{Q_0}$



f/f excitation table → tell about inputs for f/fs.

Q	Q^+	J	K	
0	0	0	$\phi = 0$ or 1	hold/reset
0	1	1	ϕ 1 or 0	toggle/set
1	0	ϕ	1	toggle/reset
1	1	ϕ	0	hold/set

Q	Q^+	T
0	0	0
0	1	1
1	0	1
1	1	0

PS	NS	J_3K_3	J_2K_2	J_1K_1	J_0K_0
$Q_3Q_2Q_1Q_0$	$Q_3Q_2Q_1Q_0$				
0000	0001	0 x	0 x	0 x	1 x
0001	0010	0 x	0 x	1 x	x 1
0010	0011	0 x	0 x	x 0	1 x
0011	0100	0 x	1 x	x 1	x 1
0100	0101	0 x	x 0	0 x	1 x
0101	0110	0 x	x 0	1 x	x 1
0110	0111	0 x	x 0	x 0	1 x
0111	1000	1 x	x 1	x 1	x 1
1000	1001	x 0	0 x	0 x	1 x
1001	1010	x 0	0 x	1 x	x 1
1010	1011	x 0	0 x	x 0	1 x
1011	1100	x 0	1 x	x 1	x 1
1100	1101	x 0	x 0	0 x	1 x
1101	1110	x 0	x 0	1 x	x 1
1110	1111	x 0	x 0	x 0	1 x
1111	0000	x 1	x 1	x 1	x 1

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	0X	0X	0X	0X
01	0X	0X	1X	0X
11	X0	X0	X1	X0
10	X0	X0	X0	X0

$J_3 = Q_2Q_1Q_0$

$K_3 = Q_2Q_1Q_0$

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	0X	0X	1X	0X
01	X0	X0	X1	X0
11	X0	X0	X1	X0
10	0X	0X	1X	0X

$J_2 = Q_1Q_0$

$K_2 = Q_1Q_0$

Q_3Q_2 \ Q_1Q_0				

$J_1 = Q_0$

$K_1 = Q_0$

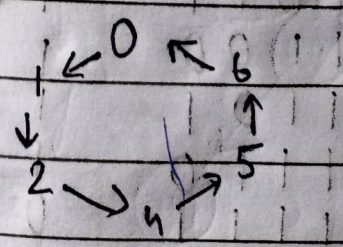
$J_0 = 1$

$K_0 = 1$

4-bit sync up-ctr J-K ✓ T ✓

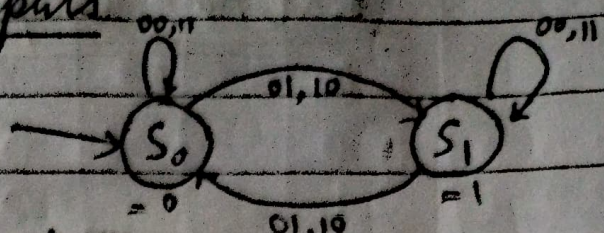
BCD up-ctr $0 \to 1 \to 2 \to \dots \to 9$

Q/ Counter with unused state



CKts with external inputs

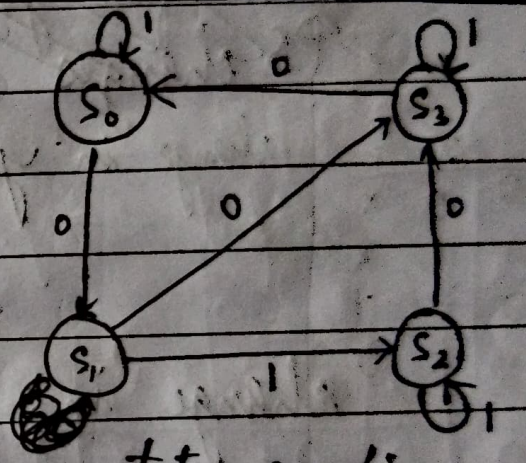
PS(Q_0)	i/p xy	NS(Q_0)	f/f i/p D
0	00	0	0
0	01	1	1
0	10	1	1
0	11	0	0
1	00	1	1
1	01	0	0
1	10	0	0
1	11	1	1



Q_0 \ xy	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$D = Q_0 \oplus x \oplus y.$

PS Q_1 Q_0	i/p x	NS Q_1^+	Q_0^+	J_1K_1	J_0K_0
0 0	0	0	1	0 X	1 X
0 0	1	0	0	0 X	0 X
0 1	0	1	1	1 X	X 0
0 1	1	1	0	1 X	X 1
1 0	0	1	1	X 0	1 X
1 0	1	1	0	X 0	0 X
1 1	0	0	0	X 1	X 1
1 1	1	1	1	X 0	X 0



state encoding

$S_0 = 00$

$S_1 = 01$

$S_2 = 10$

$S_3 = 11$

x \ Q_1Q_0	00	01	11	10
0	0X	1X	X1	X0
1	0X	1X	X0	X0

$J_1 = Q_0$

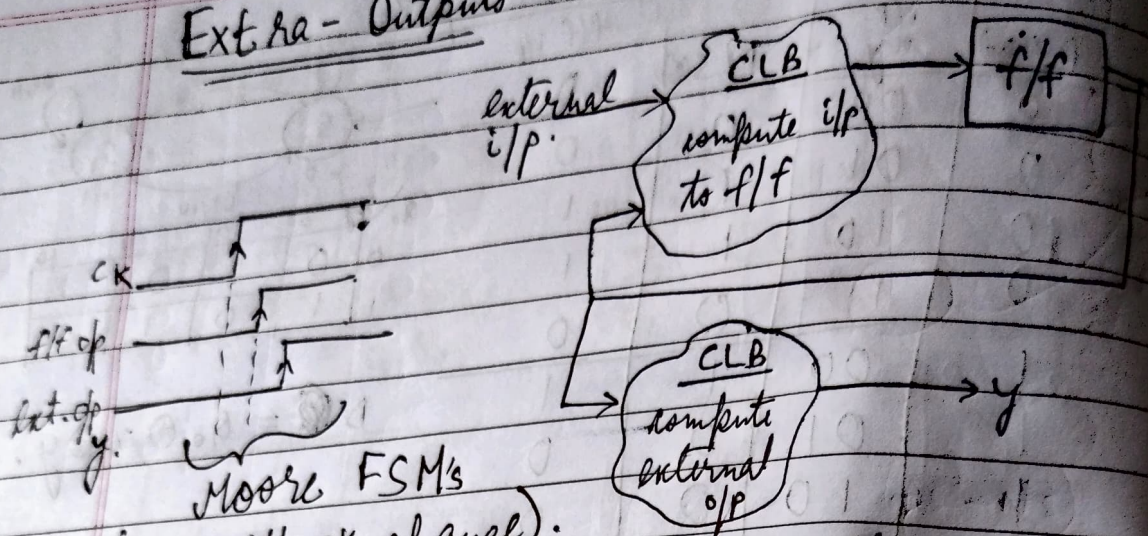
$K_1 = \bar{x} Q_0$

x \ Q_1Q_0	00	01	11	10
0	1X	X0	X1	1X
1	0X	X1	X0	0X

$J_0 = \bar{x}$

$K_0 = x\bar{Q}_1 + \bar{x}Q_1$

Extra - Outputs — When ? and What ?



Moore FSM's
(o/ps with ck change).

~~even after giving external o/p, y changes after PS changes~~

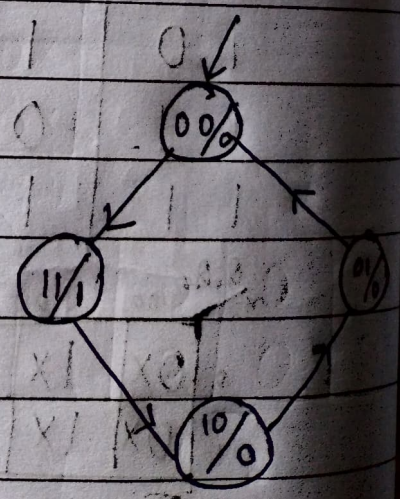
change in y → after PS changes (after ck ↑)
→ after i/p " (whenever i/p changes)

	J	K
0 0	0	x
0 1	1	x
1 0	x	1
1 1	x	0

change in y at clock → Moore machine
" not " " → Mealy machine

Moore FSM

PS $Q_1 Q_0$	NS $Q_1^+ Q_0^+$	$J_1 K_1$	$J_0 K_0$	y
0 0	1 1	1 x	1 x	0
1 1	1 0	x 0	x 1	1
1 0	0 1	x 1	1 x	0
0 1	0 0	0 x	x 1	0



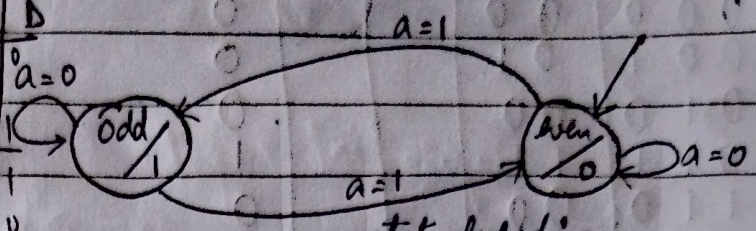
$J_1 = \overline{Q_0}$ $\quad J_0 = 1$ $\quad y = Q_1 Q_0$

$K_1 = \overline{Q_0}$ $\quad K_0 = 1$

Moore FSM - Parity checker

Odd-parity checker.

	PS	a	NS	J K	y	D
input → "a"	0	0	0	0 X	0	0
output → "y"	0	1	1	1 X	0	1
	1	0	1	X 0	1	1
	1	1	0	X 1	1	0



state encoding
even → 0
odd → 1

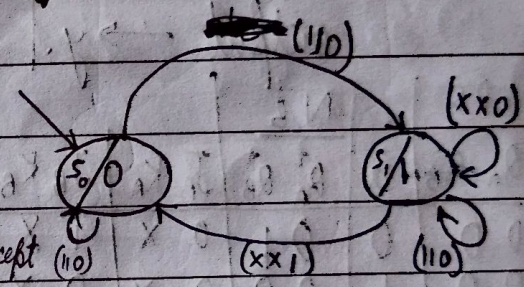
$J = a$
$K = a$, $y = Q_0$.

or

$D = Q_0 \oplus a$

Q.) 3 inputs → a, b, c

output → $y = \begin{cases} 1 & , ab\bar{c} = 1 \\ 1 & , c == 0 , y_{t-1} == 0 \\ 0 & , \text{o.w.} \end{cases}$



$D_A = ab\bar{c} + Q_A\bar{c}$

$y = Q_A$

state encoding
$S_0 \rightarrow 0$
$S_1 \rightarrow 1$

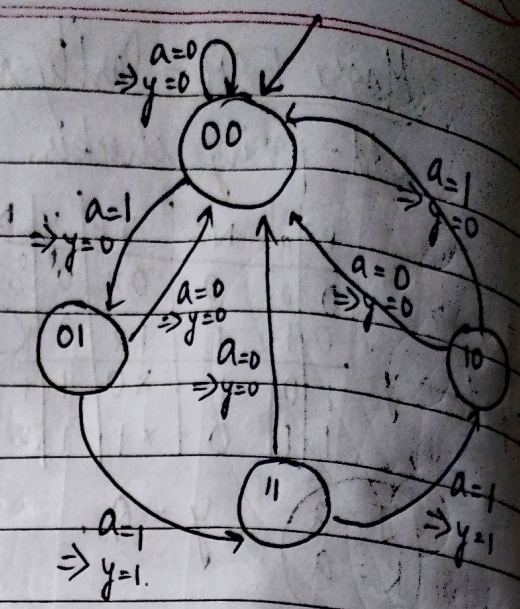
Mealy FSM

Q.) Input → "a"

Output → y { up-counting, 2-bit grey code, at clocktick, (if a = 1)
grey-code becomes 00. (a = 0)

$y = \begin{cases} 0 & , \text{entering/exiting/staying initial state.} \\ 1 & \text{o.w.} \end{cases}$

PS			NS					
Q_A	Q_B	a	Q_A^+	Q_B^+	$J_A K_A$	$J_B K_B$	y
0	0	0	0	0			0
0	0	1	0	1			0
0	1	0	0	0			0
0	1	1	1	1			1
1	1	0	0	0			0
1	1	1	1	0			1
1	0	0	0	0			0
1	0	1	0	0			0



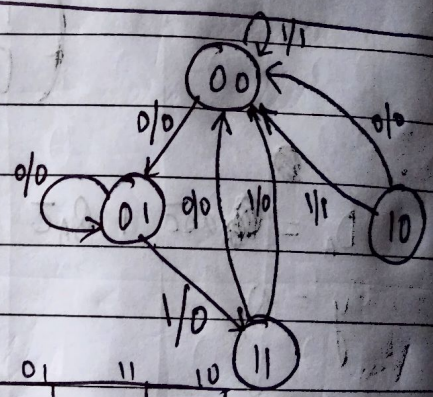
f/f in ← (PS − NS)

Moore o/p ← PS

Mealy o/p ← {PS, in}.

Q.) input → a, output → y. | using JK.

PS			NS					
Q_A	Q_B	a	Q_A^+	Q_B^+	$J_A K_A$	$J_B K_B$	y
0	0	0	0	1	0 X	1 X	0
0	0	1	0	0	0 X	0 X	1
0	1	0	0	1	0 X	X 0	0
0	1	1	1	1	1 X	X 0	0
1	1	0	0	0	X 1	X 1	0
1	1	1	0	0	X 1	X 1	0
1	0	0	0	0	X 1	0 X	0
1	0	1	0	0	X 1	0 X	1



a \ Q_AQ_B	00	01	11	10
0	1X	X0	X1	0X
1	0X	X0	X1	0X

$J_B = \overline{Q_A}\,\overline{a}$

$K_B = Q_A$

a \ Q_AQ_B	00	01	11	10
0	0	0	0	0
1	1	0	0	1

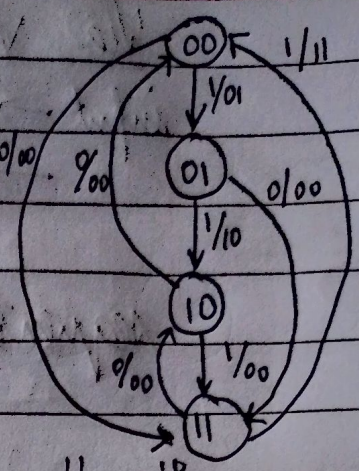
$y = a\overline{Q_B}$

a \ Q_AQ_B	00	01	11	10
0	0X	0X	X1	X1
1	0X	1X	X1	X1

$J_A = aQ_B$ $K_A = 1$

Q6 input → a Use D-f/f.
output → xy.

PS $Q_A Q_B$	a	NS $Q_A^+ Q_B^+$	D_A	D_B	x	y
00	0	11	1	1	0	0
00	1	01	0	1	0	1
01	0	11	1	1	0	0
01	1	10	1	0	1	0
10	0	00	0	0	0	0
10	1	11	1	1	0	0
11	0	10	1	0	0	0
11	1	00	0	0	1	1



a \ $Q_A Q_B$	00	01	11	10
0	11	11	10	00
1	01	10	00	11

$D_A = aQ_A\overline{Q_B} + \overline{a}\,\overline{Q_A} + \overline{a}\,\overline{Q_B} + \overline{Q_A}\,Q_B$

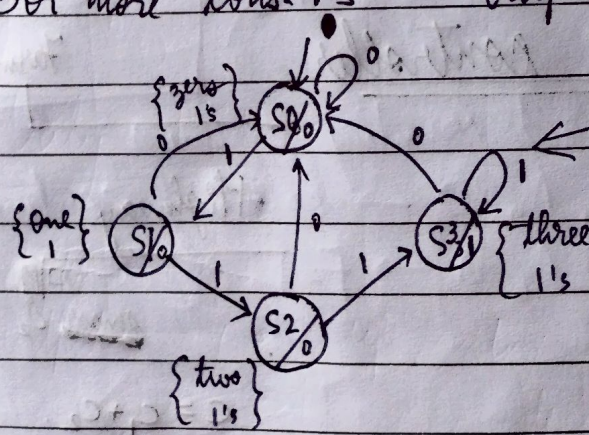
$D_B = aQ_B + \overline{a}\,Q_A$

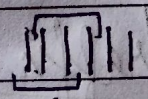
a \ $Q_A Q_B$	00	01	11	10
0	00	00	00	00
1	01	10	11	00

$x = aQ_B$, $y = a\overline{Q_A}\,\overline{Q_B} + aQ_AQ_B$

Sequence Detector using D-f/f

Ex: 3 or more cons. 1's → output = 1.





if overlapping valid → 001111

if " not valid → 001001

Q_B	Q_A	PS	a	NS	$J_A K_A$	$J_B K_B$	y
0	0	00	0	00	0 x	0 x	0
1	0	00	1	01	0 x	1 x	0
0	0	01	0	00	0 x	x 1	0
0	1	01	1	10	1 x	x 1	0
0	0	10	0	00	x 1	0 x	0
1	1	10	1	11	x 0	1 x	0
0	0	11	0	00	x 1	x 1	1
1	1	11	1	11	x 0	x 0	1

$D_A = aQ_A + aQ_B$

$D_B = aQ_A + a\overline{Q_B}$

$y = Q_A Q_B$.

AA, AB, BB, BC, CC, CD, DD
10 9 8 7 6 5 4

Ex—

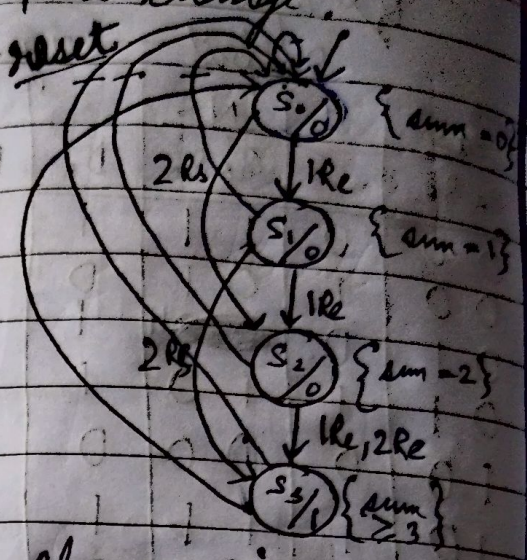
Tea Vending Machine — 3 Rs, no change.

input → 1 Re = 'a'
→ 2 Rs = 'b'

sequential input

a → b
or b → a.

reset sensor.



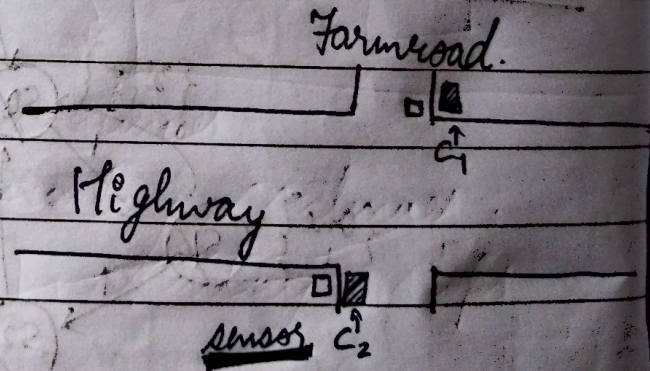
Features.

1. Tea dispensed
2. Reset → super-user
3. Fake coin check
4. display ~~fake~~ partial sum
5. Tea supply finished
6. Amount of tea { small/large, 3Rs 5Rs }
7. User input of # cups.

If change given

→ Track available coins to return
→ keep balance.
→ alarm seller

Traffic light controller



Outputs

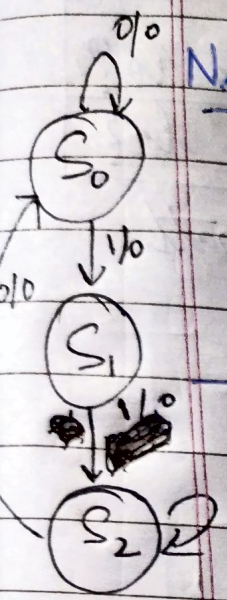
FR HG, HY
FY, FG HR

$$\boxed{\begin{aligned} HG + HY &= FR \\ FY + FG &= HR \end{aligned}}$$

→ Output

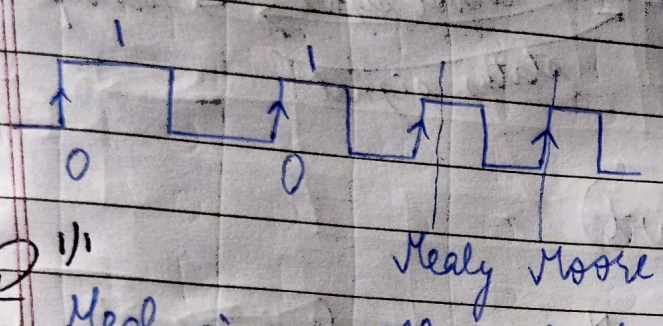
$C = C_1 + C_2$.

Moore vs Mealy FSMs

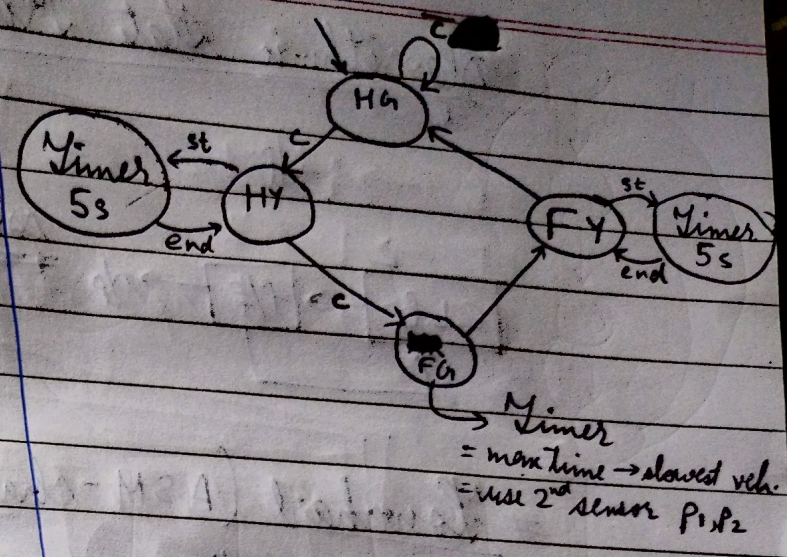


Number of states:

Mealy ≤ Moore



Mealy gives output faster.



Output signal duration → Moore = 1 clock-cycle.

Mealy = min{input change, 1 clock cycle}

susceptible to glitches on i/p.

Controller design

↓

stable

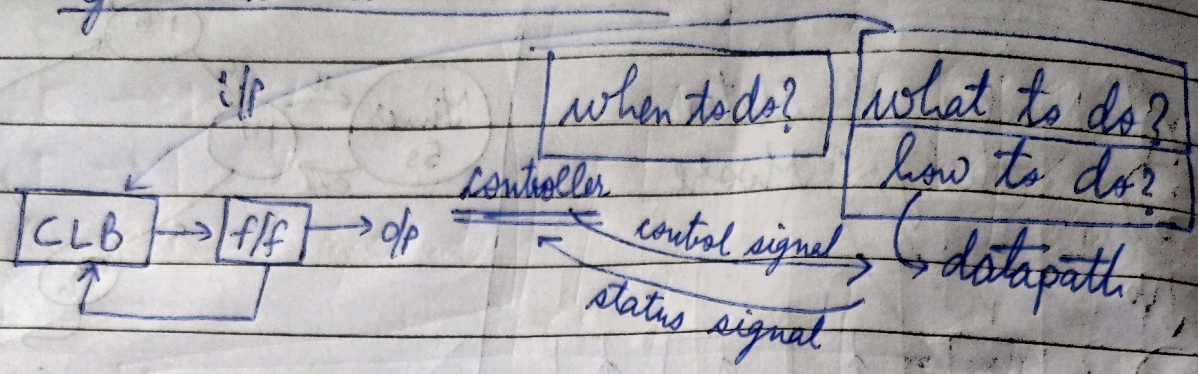
↓

level-sensitive

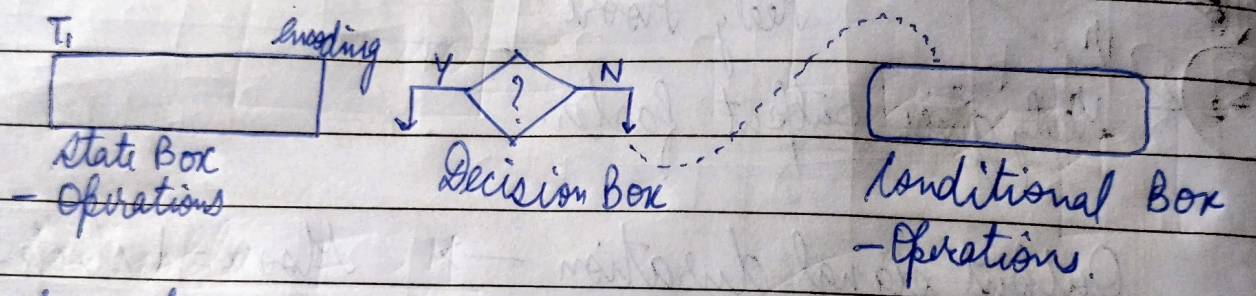
↓

write enable (WE) signal of SRAM.

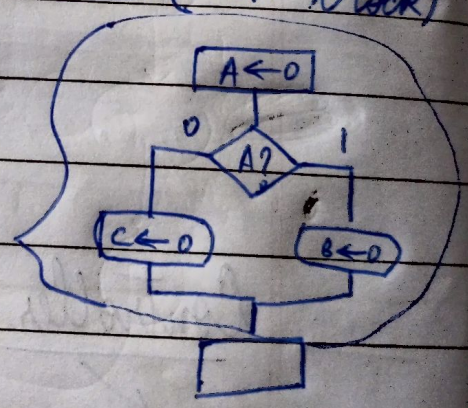
Algorithmic State Machines



Flowchart (ASM-chart)



1 state starts from a ▭ until next ▭ (ASM block)



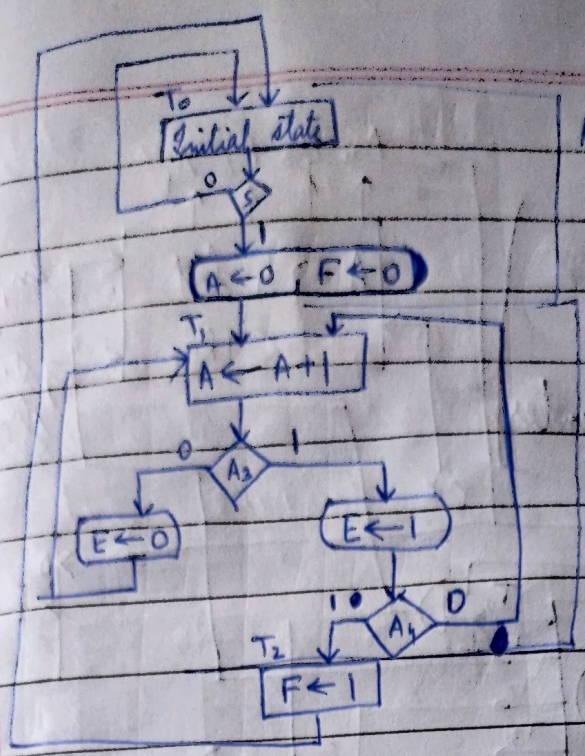
4-bit binary counter – A ($A_4 A_3 A_2 A_1$)

start signal S ⟶ clear A (=0) and F (=0) f/f.

If $A_3 = 0$ ⟶ E=0 and counting continues

$A_3 = 1$ ⟶ E=1, $A_4 = 0$ ⟶ counting continues.

$A_4 = 1$ ⟶ F=1, counting stops



ASM block for state T_0.

ASM block for state T_1.

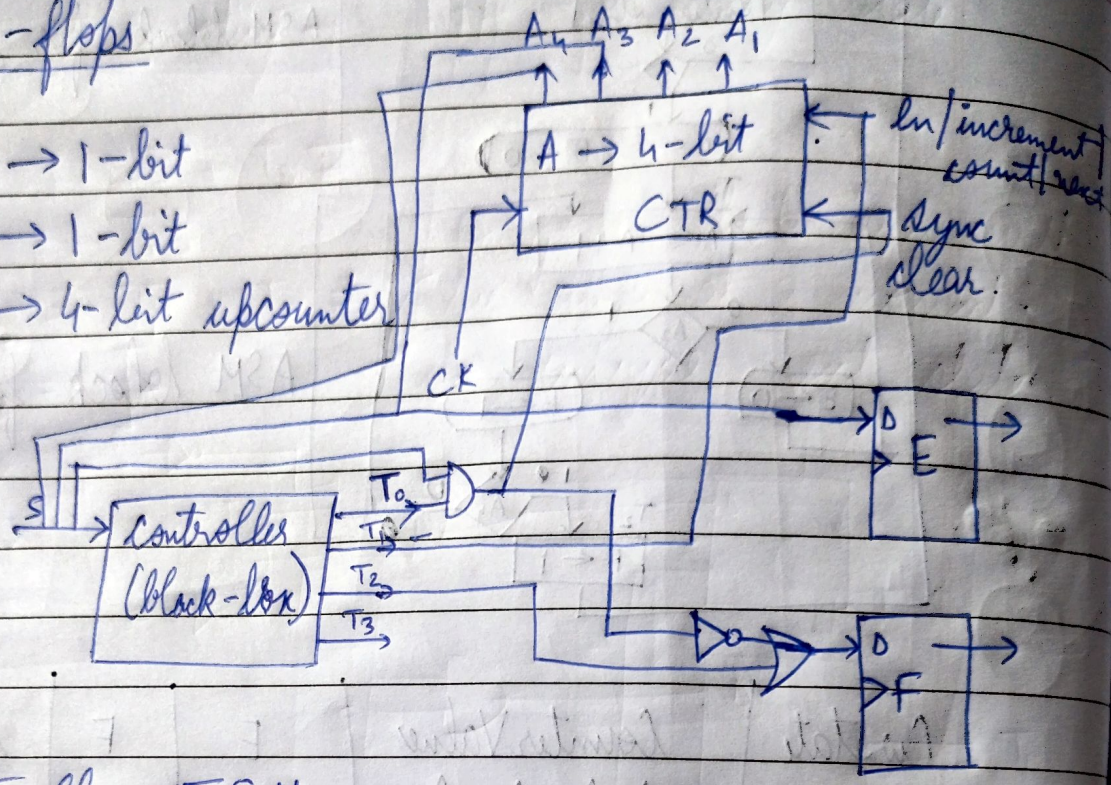
CK	Curstate	Counter Value A_4 A_3 A_2 A_1	E	F	Conditions A_3	A_4	Next State
1	T_1	0 0 0 0	~~X~~ 1 (from previous cycle)	0	0	0	T_1
2		0 0 0 1	0	0	0	0	
3		0 0 1 0	0	0	0	0	
4		0 0 1 1	0	0	0	0	
5		0 1 0 0	0	0	0	0	T_1
6		0 1 0 1	1	0	1	0	
7		0 1 1 0	1	0	1	0	
8		0 1 1 1	1	0	1	0	
9		1 0 0 0	1	0	1	0	T_1
10		1 0 0 1	0	0	0	1	
11		1 0 1 0	0	0	0	1	
12		1 0 1 1	0	0	0	1	
13		1 1 0 0	0	0	0	1	T_1
14		1 1 0 1	1	0	1	1	T_2
15		1 1 0 1	1	1			T_0

Flip-flops

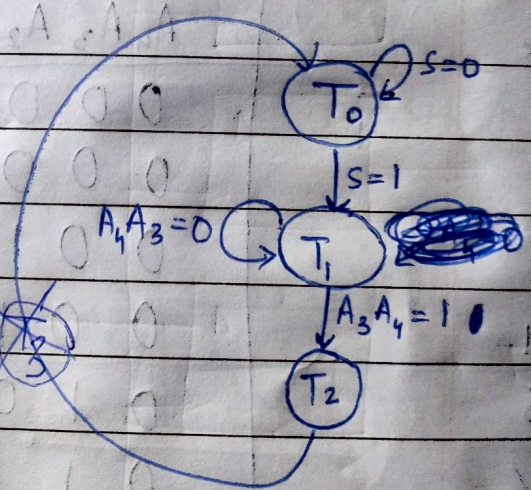
E → 1-bit

F → 1-bit

A → 4-bit upcounter

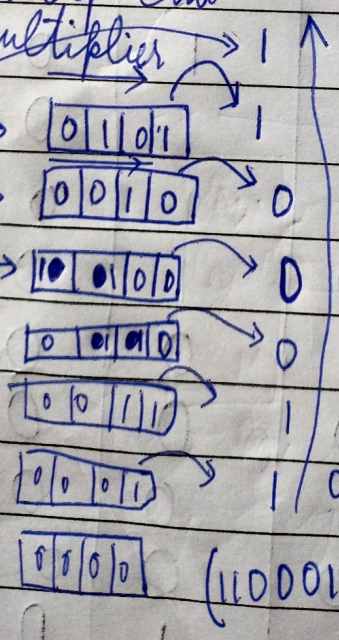


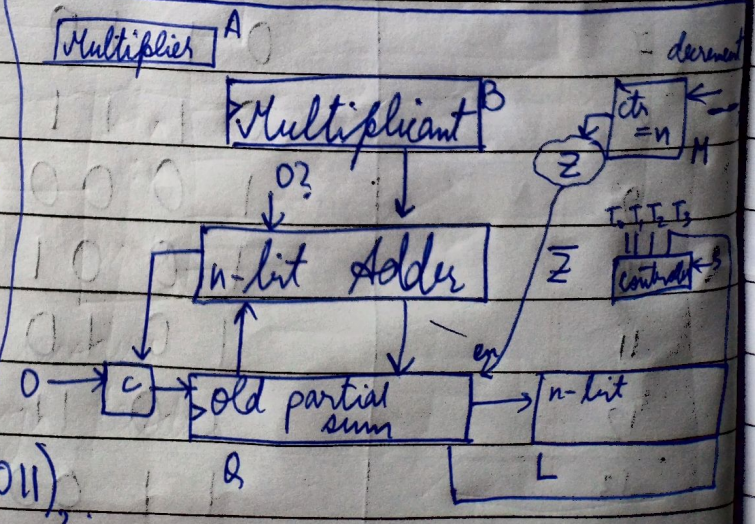
Controller FSM



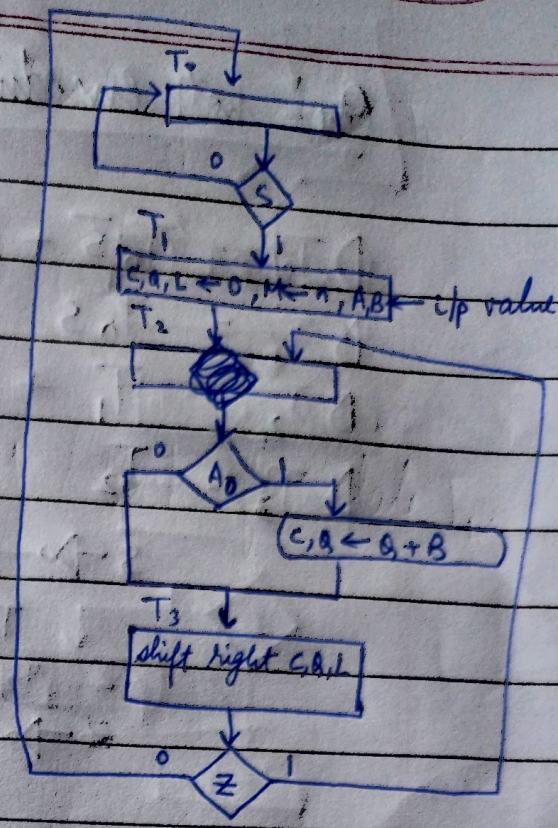
Sequential Multiplier

```
  1011   Multiplicand
x 1001   Multiplier
-------
  1011
 0000x
0000xx
1011xxx
-------
1100011
```

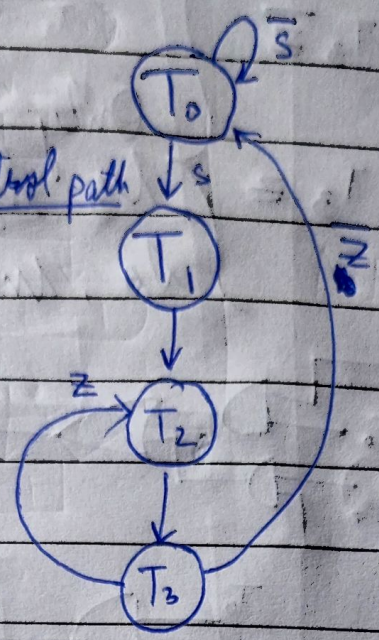




ASM chart →



FSM chart



Registers Transfer Level

Control path

Data path (RTL)

$T_1 : C, Q, L \leftarrow 0$

$M \leftarrow n$

$T_2 : M = M - 1$

if (A_0) then $\{C, Q\} \leftarrow Q + B$.

T_3 : shift-right C, Q, A.

PS $Q_1 Q_0$	i/p S, Z	NS $Q_1 Q_0$
0 0	0 0	0 0
0 0	0 1	0 0
0 0	1 0	0 1
0 0	1 1	0 1
0 1	0 0	1 0
0 1	0 1	1 0
0 1	1 0	1 0
0 1	1 1	1 0
1 0	0 0	1 1
1 0	0 1	1 1
1 0	1 0	1 1
1 0	1 1	1 1
1 1	0 0	0 0
1 1	0 1	1 0
1 1	1 0	0 0
1 1	1 1	1 0

SZ \ Q_1Q_0	00	01	11	10
00	00	10	00	11
01	00	10	10	11
11	01	10	10	11
10	01	10	00	11

$D_1 = \bar{Q}_1 Q_0 + Q_1 \bar{Q}_0 + Q_1 Z$.

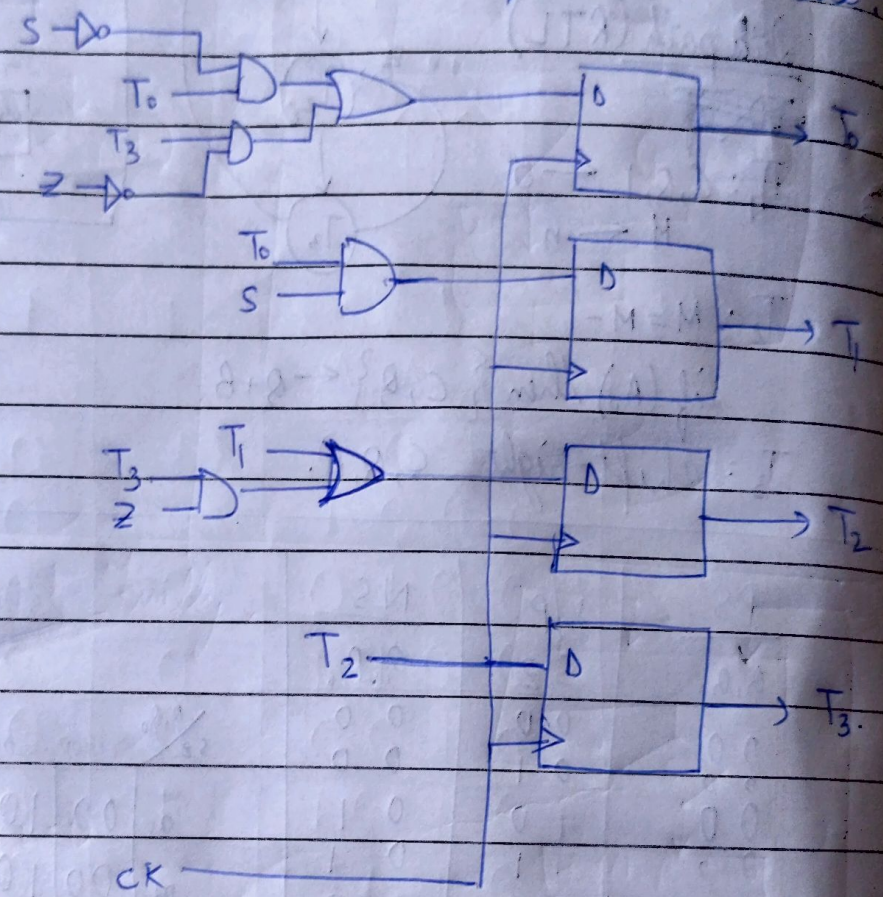
$D_2 = Q_1 \bar{Q}_0 + Q_0 S$.

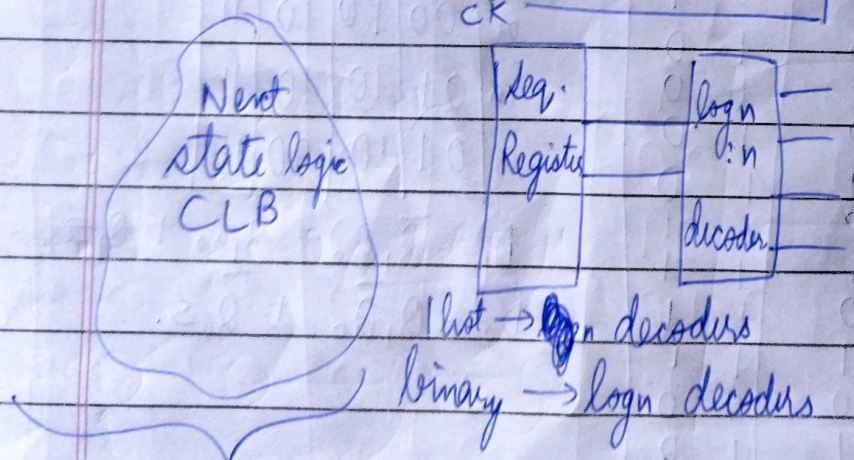
classmate
Date
Page

One-hot encoding – 1 f/f for every state.

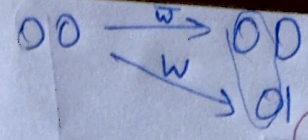
$$\left.\begin{aligned} DT_0 &= T_0\bar{S} + T_3\bar{Z} \\ DT_1 &= T_0 S \\ DT_2 &= T_1 + T_3 Z \\ DT_3 &= T_2 \end{aligned}\right\}$$

Only 1 of the 6 terms will be high at a moment. So, unique next state.

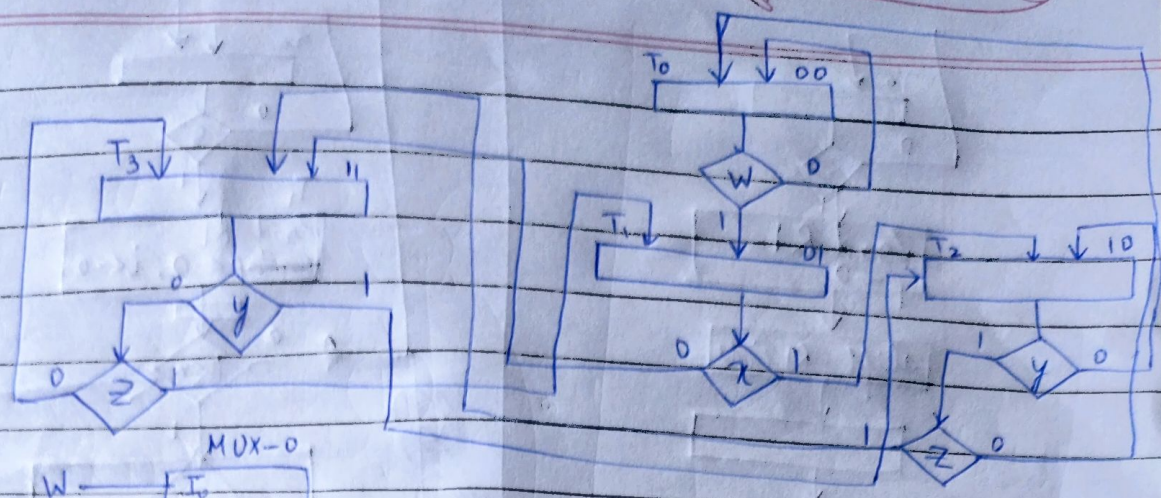


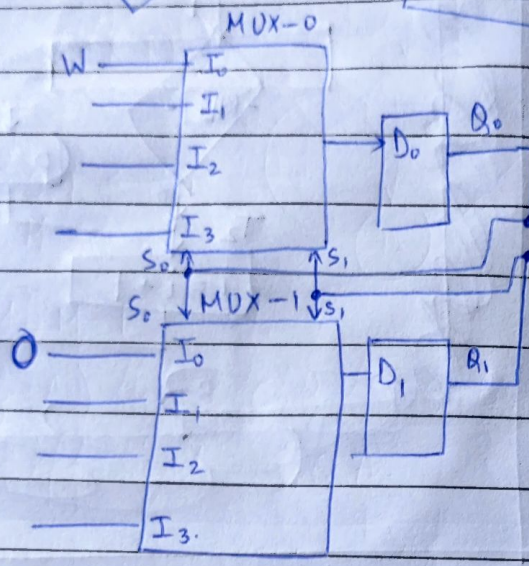


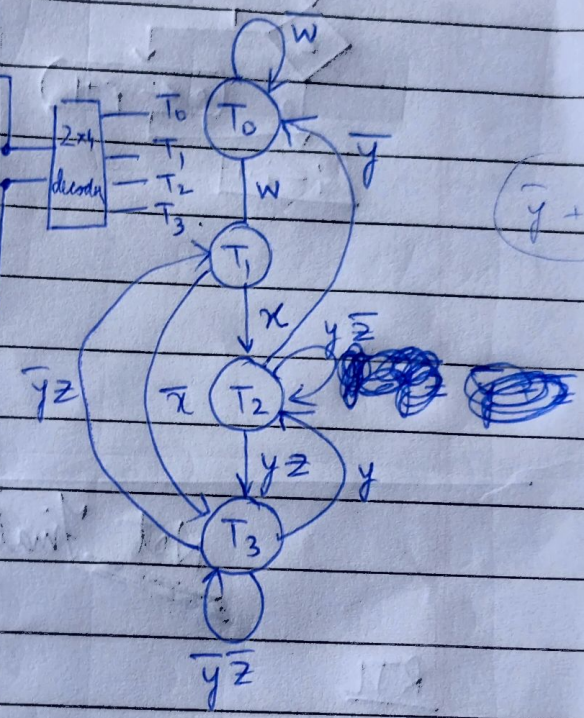
1 hot → log n decoders
binary → log n decoders



Q.1







$\bar{y} + y\bar{z}$

PS	NS	cond.
00	00	$\bar{w}$
00	01	w

D1 D0

No. of 1's counter

R1 = 10101110

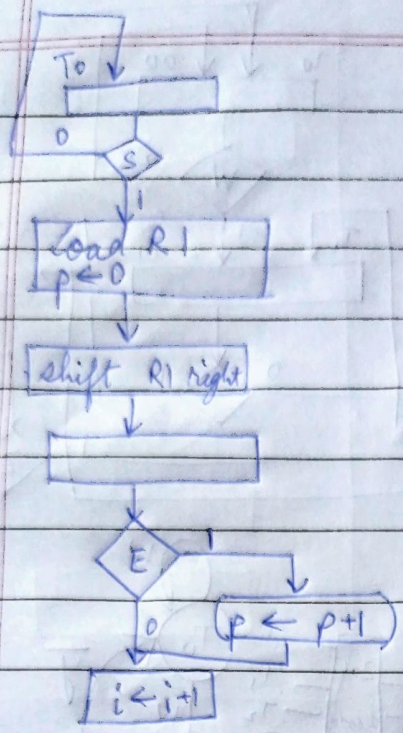
P = counter = 0

iterator = i = 0

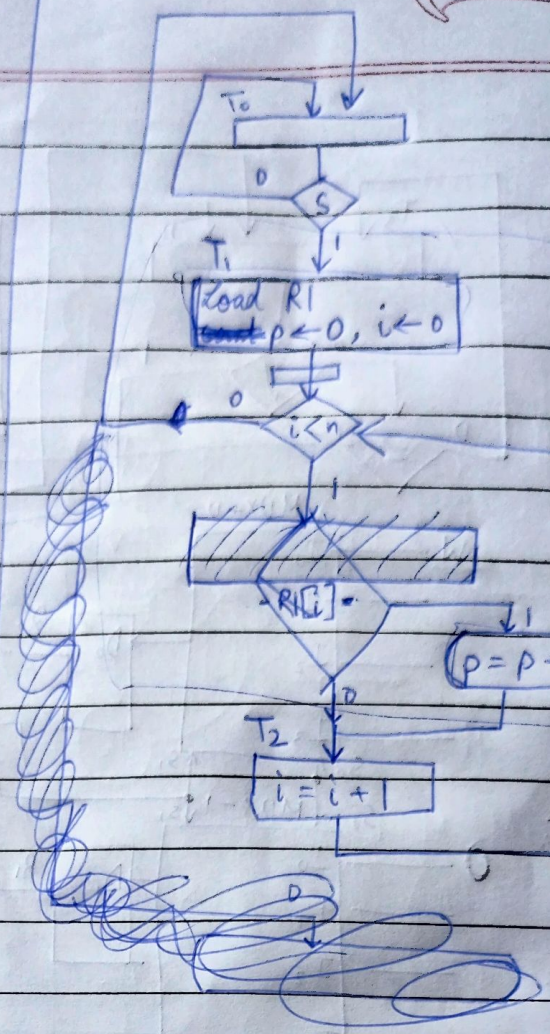
stop → i = (n-1)

```
int count = 0
string s;
cin >> s;
for (int i=0; i<s.length(); i++)
    count += s[i] - '0';
~~count~~ cout << count;
```

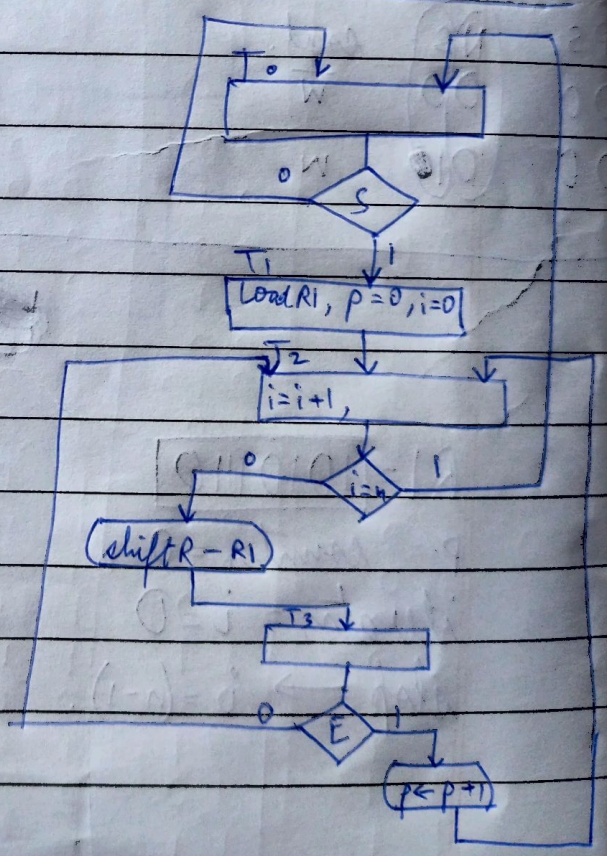
Madms



My



Not Finally,



RTL

T_0:

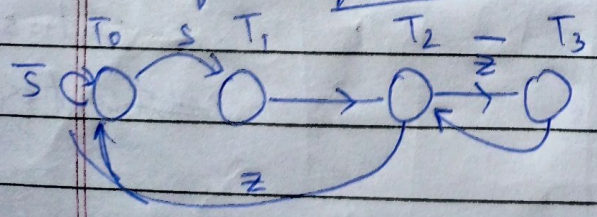
T_1: Load R1, $p=0, i=0$

T_2: $i=i+1$
if $(i \neq n)$ shR R1

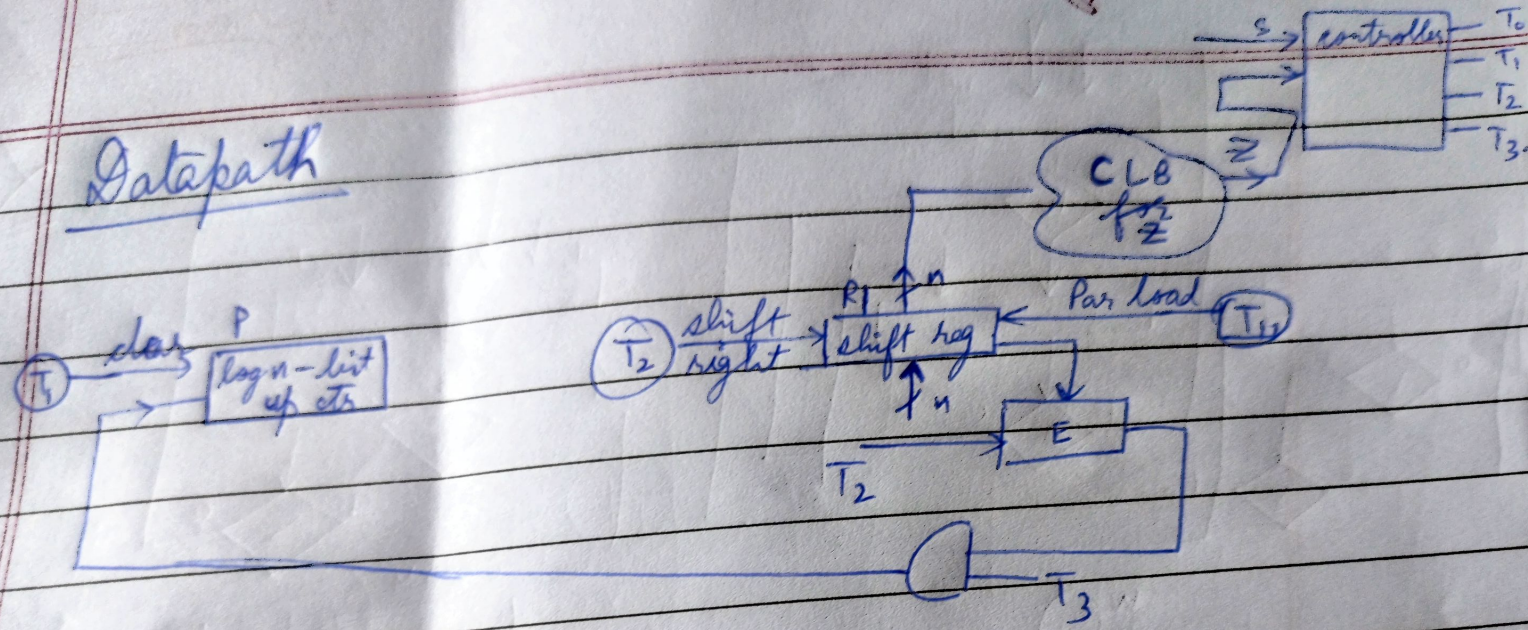
T_3: if (E) $p=p+1$

$Z_1 = 1$ if $R_1 = 0$

Control FSM



Datapath


controller
S
Z
T_0
T_1
T_2
T_3
CLB
f_z
R_1
n
Par load
T_1
T_2
shift right
shift reg
n
E
T_2
T_3
T_1
clear
P
log n-bit up ctr