

Sequence Detector using D-flt

3 or more 1's (consecutive)

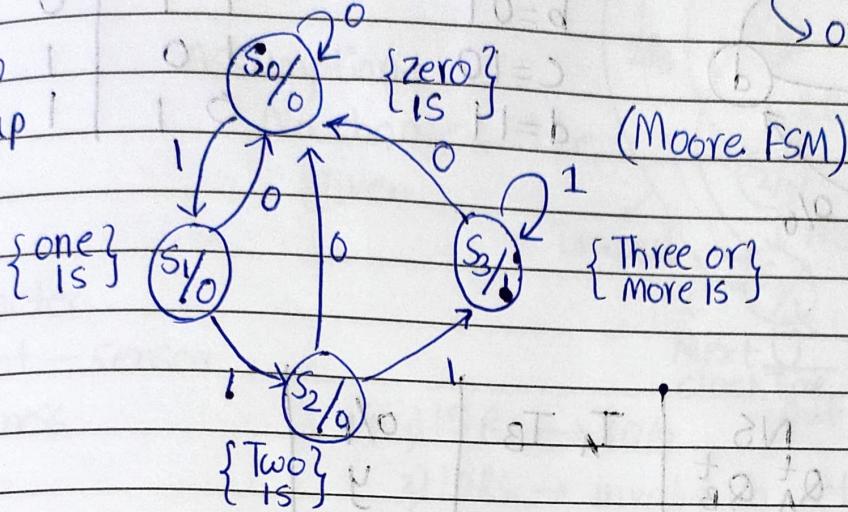
- flip the output

$\overrightarrow{00110011} \{ 00111100$

output=0 output=1

change state - output=1
overlapping sequence

allow overlap



State encoding

$$S_0 = 00$$

$$S_1 = 01$$

$$S_2 = 10$$

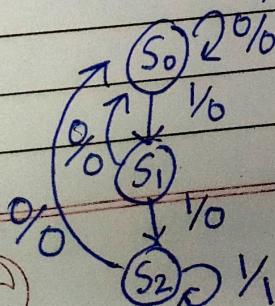
$$S_3 = 11$$

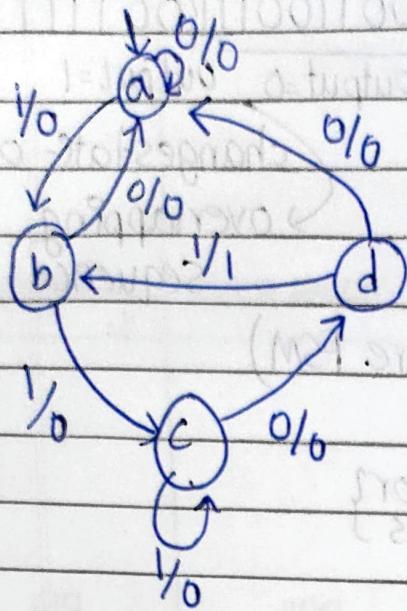
PS	I/P	Q _A Q _B	NS	O/P	Q _A	Q _B
0 0	0	0 0	0 0	0	0	0
0 0	1	0 1	0 1	0	0	1
0 1	0	0 0	0 0	0	1	0
0 1	1	1 0	1 0	0	1	0
1 0	0	0 0	0 0	0	1	1
1 0	1	1 1	1 1	1	1	1
1 1	0	0 0	0 0	0	0	0
1 1	1	1 1	1 1	1	1	1

$$D_A = aQ_A + a\bar{Q}_B$$

$$D_B = a\bar{Q}_A + aQ_B$$

Mealy:





State encoding

$Q_A \cdot Q_B^t$	T
a=00	0 0 1 0
b=01	1 1 0 0
c=10	1 0 1 1
d=11	0 1 1 1

PS $Q_A Q_B$	i/p x	NS $Q_A^t Q_B^t$	T _A T _B	o/p. y
0 0	0	0 0	0 0	0
0 0	1	0 1	0 1	0
0 1	0	0 0	0 1	0
0 1	1	1 0	1 1	0
1 0	0	1 1	0 1	0
1 0	1	1 0	0 0	0
1 1	0	0 0	1 1	0
1 1	1	0 1	1 0	1

T _A	x	$Q_A Q_B$	00	01	11	10	
0	0	0	0	0	1	0	0 0 0 0
1	0	1	0	1	0	1	0 1 0 0

$$= x Q_B + Q_A Q_B$$

T _B	x	$Q_A Q_B$	00	01	11	10	
0	0	0	0	0	0	0	0 0 0 0
1	1	1	0	1	0	1	1 1 0 0

$$= x \bar{Q}_A + \bar{x} Q_B + \bar{x} Q_A$$

Vending machine controller \rightarrow 3Rs

inputs = 1Rs

coins = 2Rs

Assumption-

No change will be given

Diameter

Weight - sensor

Thickness

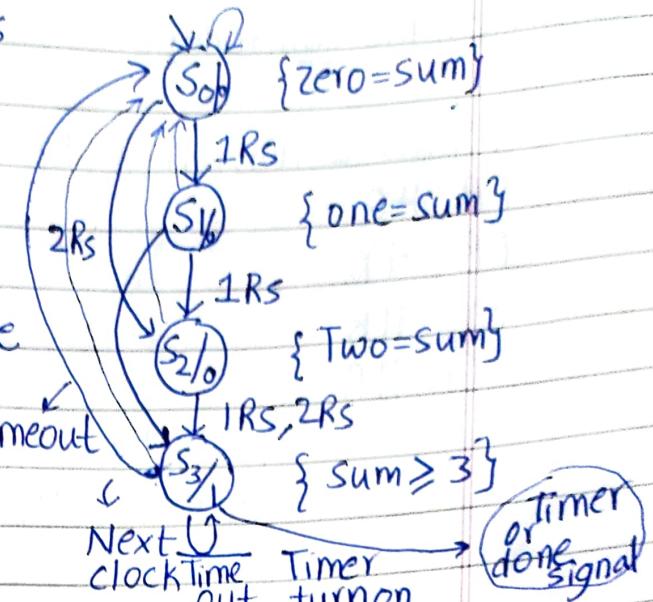
i/p 1) 10 Rs \rightarrow 30/p

2) 10Rs \rightarrow invalid in return coin.

Sensor a = high

Sensor b = high \rightarrow valid

$a \rightarrow b$ reset sensor



Features

inputs \rightarrow ① 1Rs = a

② 2Rs = b

③ Reset

1. Tea

2. Reset

3. Fake coin check

4. Display partial sum

If change given \rightarrow Track available coins to return / keep balance

5Rs

10Rs

4Rs

5. Return change

6. Tea Supply finish
{Ingredients}

7. Amount of Tea

small / large

3Rs 5Rs

8. User input of # of cups.

9. Tea + Coffee

Traffic light controller

outputs = HG, HY, FG, FY

FR HG

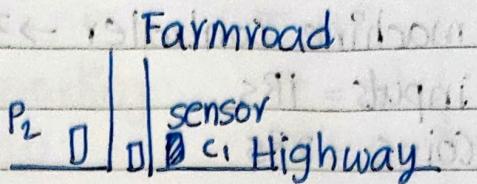
FR HY

FY HR

FG HR

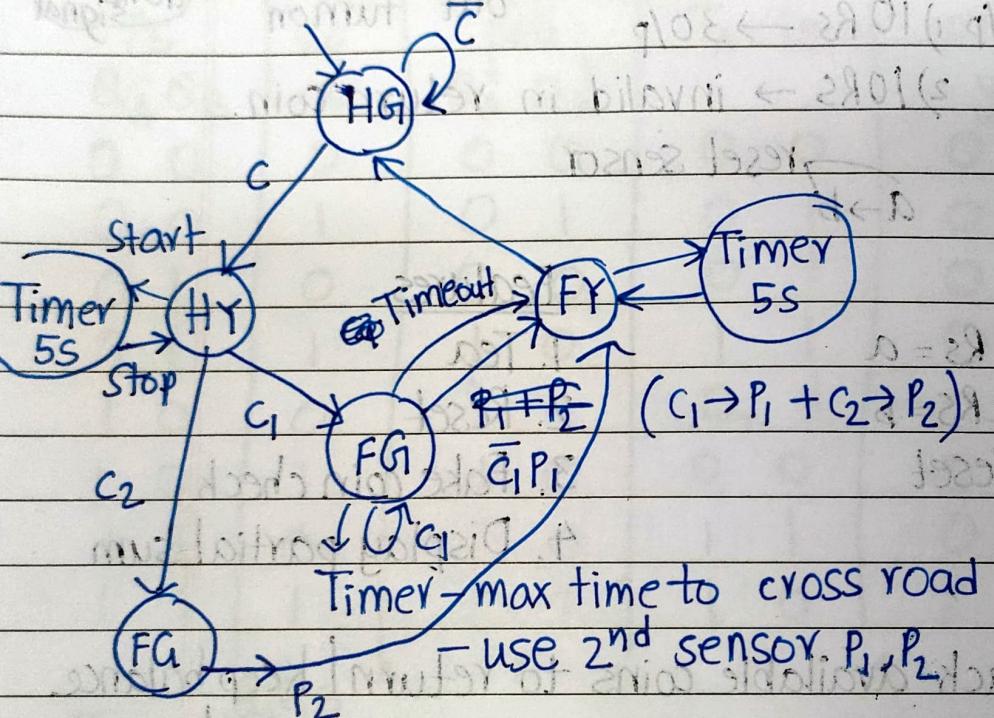
$$HG + HY = FR$$

$$HY + FG = HR$$



$$C = C_1 + C_2$$

$C_2 \rightarrow$ not sensitive
to car coming
downward



Count farm carts

at C_1

Then check
at P_1 .

(similar at C_2, P_2)

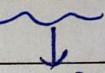
So build Two
FSMs

Mealy, Moore FSMs

No. of States: Mealy < Moore faster output in Mealy.

Output signal duration

depends on ilp = Mealy



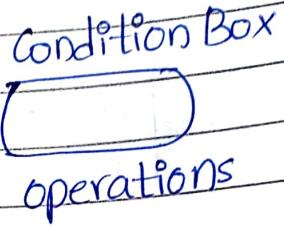
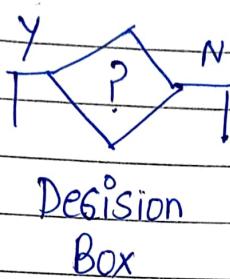
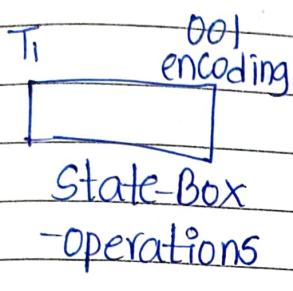
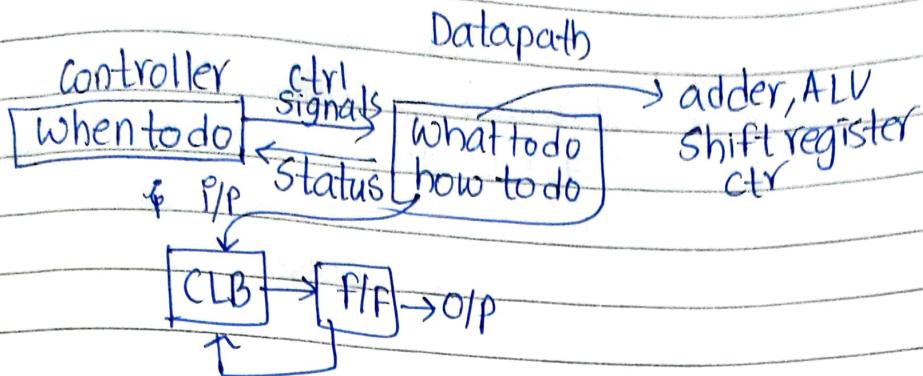
with clk-duration - Moore.

Susceptible
to glitches
on ilp.

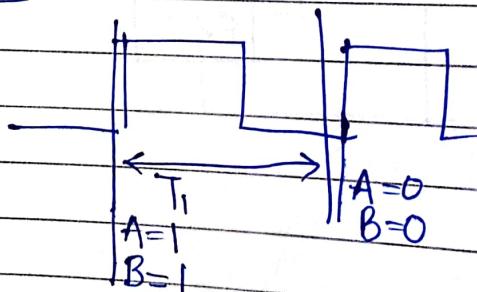
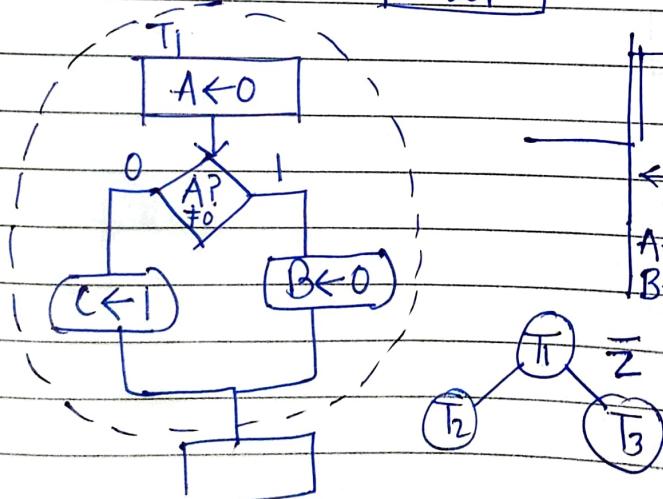
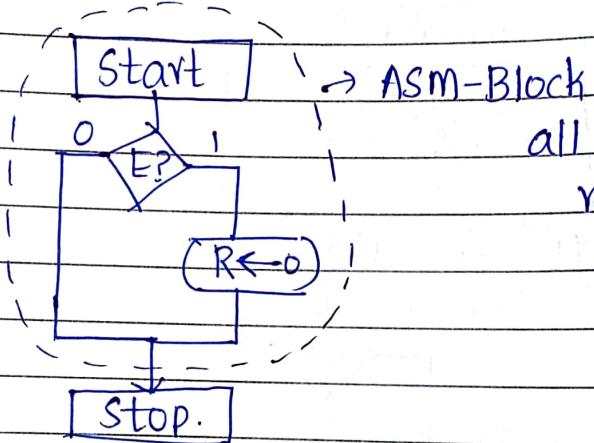
Edge sensitive - Mealy.

level sensitive - Moore.

Algorithmic State Machine.



A SM charts



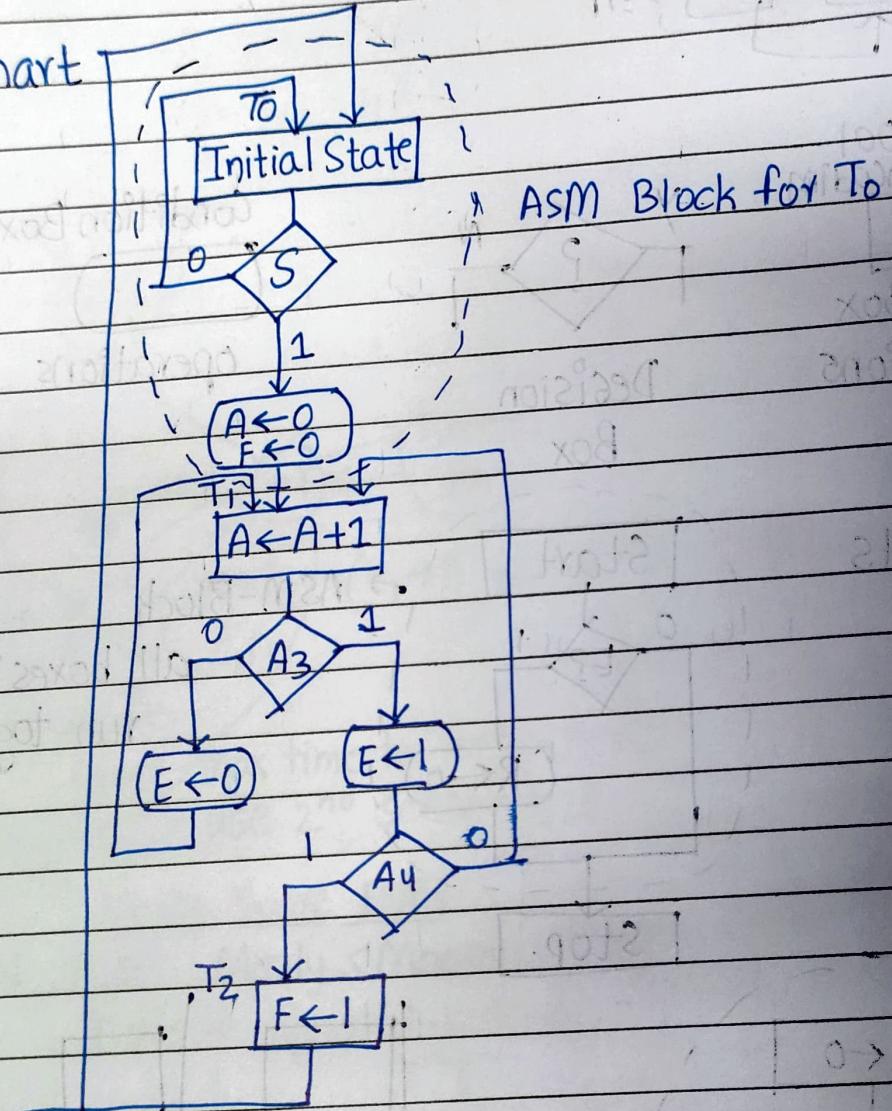
4 Flip Flop Binary counter, Start signal S; A, F clear
 $A_4 A_3 A_2 A_1$

↓
 4 bit counter

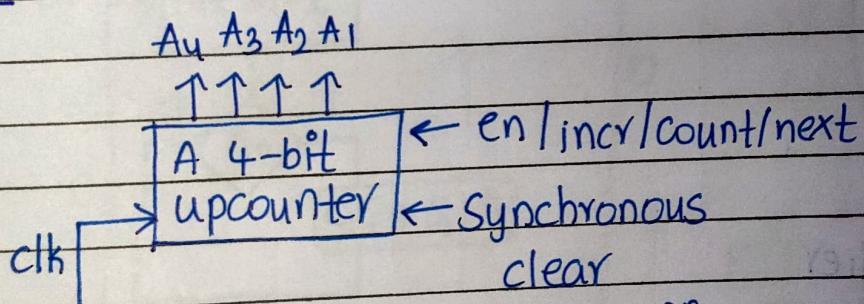
$A_3 = 0$ Then $E = 0$ and counting continues.

- $A_3 = \emptyset$ Then $E = 1$ If $A_4 = 0$, Then counting continues
 $A_4 = 1$, Then F is set to 1 in
 nxt clk pulse.

ASM chart

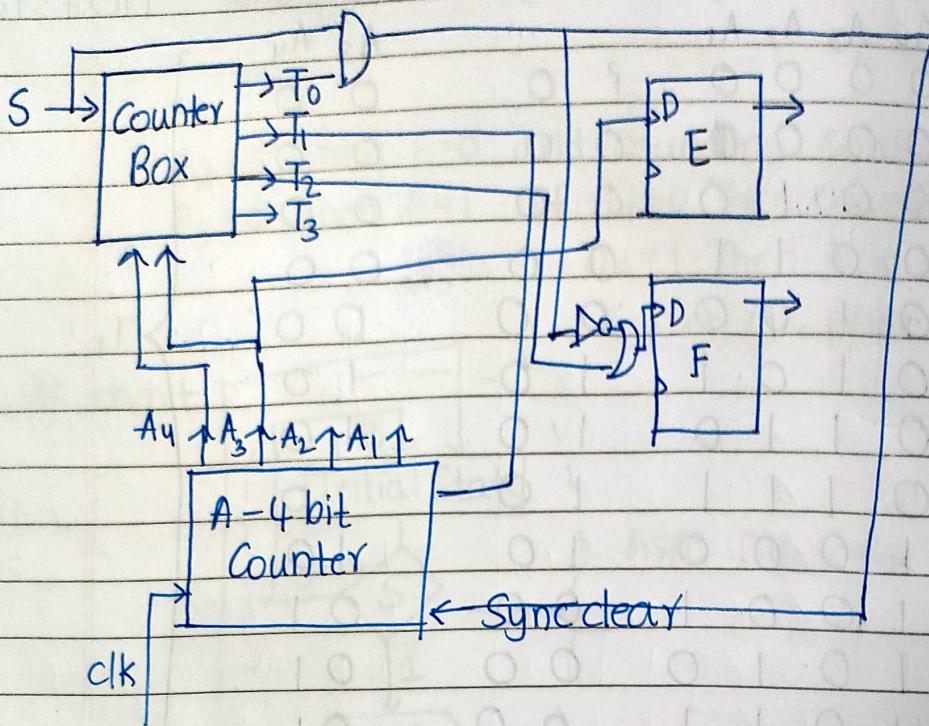


clk	current state	Counter Value	same time		Conditions	next state
			E	F		
1	T ₁	A ₄ A ₃ A ₂ A ₁	0 0 0 0	?	0 0	
2			0 0 0 0	1	0 0	
3			0 0 1 0	0 0	0 0	
4			0 0 1 1	0 0	0 0	
5			0 1 0 0	0 0	0 0	
6			0 1 0 1	1 0	1 0	
7			0 1 1 0	1 0	1 0	
8			0 1 1 1	1 0	1 0	
9			1 0 0 0	1 0	1 0	
10			1 0 0 1	0 0	0 1	
11			1 0 1 0	0 0	0 1	
12	T ₁		1 0 1 1	0 0	0 1	
13	T ₁		1 1 0 0	0 0	0 1	
14	T ₁		1 1 0 1	1 0	1 1	
15	T ₂		1 1 0 1	1 1	1 1	Initial state
16	Initial state	A ₄ A ₃ A ₂ A ₁	0 0 0 0	1 0 1	1 1	T ₁
17	T ₁		0 0 0 0	1 0	1 1	T ₂
18	T ₂		0 0 0 1	1 1	0 0	Initial state

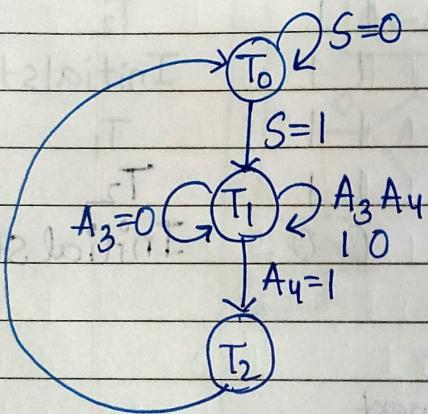


⇒ At every clk edge, if en=1, NS = PS+1

19.	Initial state	0 0 0 1	1 1	0 0	T ₁
20	T ₁	0 0 0 0	1 0	0 0	T ₁
21	T ₁	0 0 0 1	0 0	0 0	



Controller FSM



Sequential Multiplier

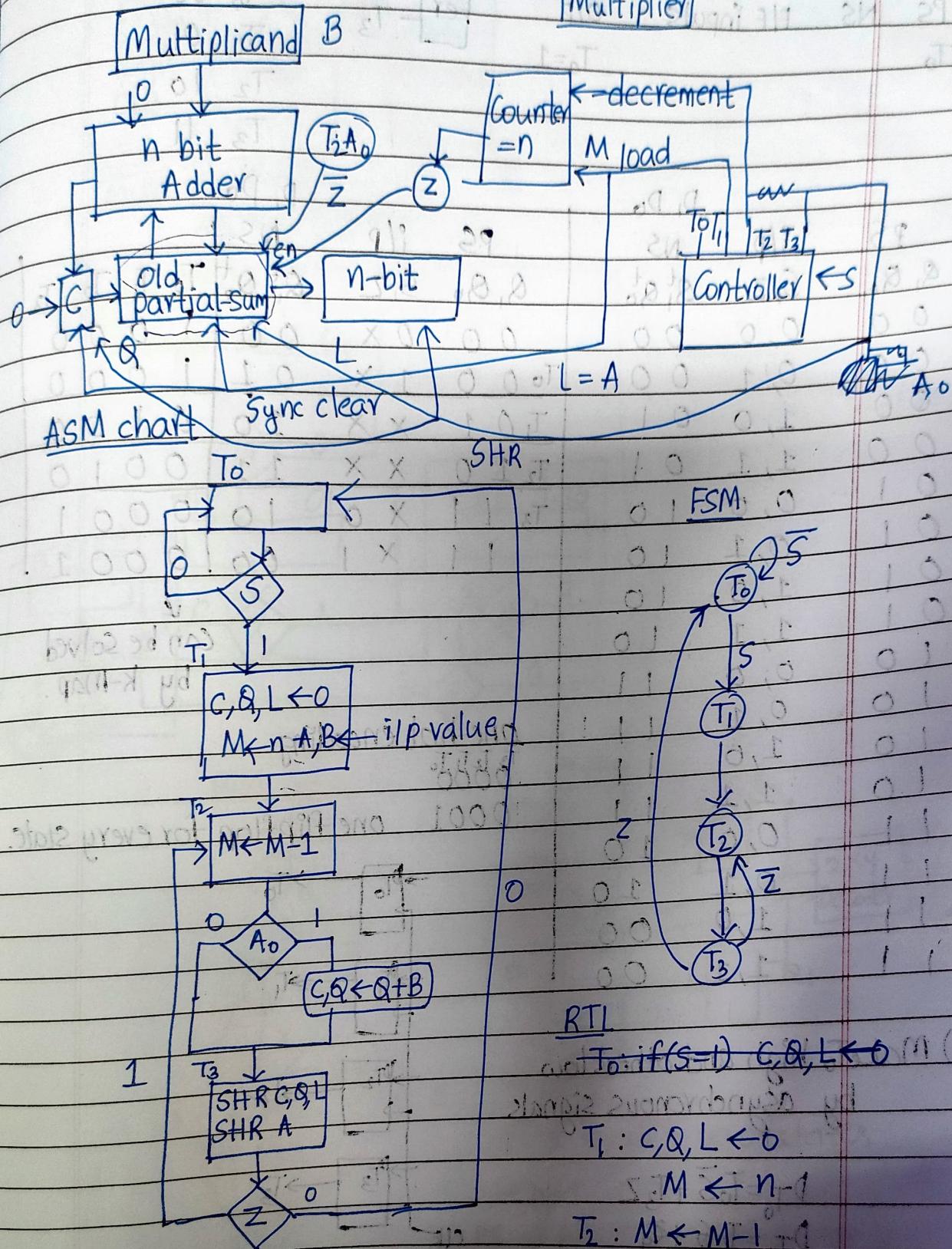
$$\begin{array}{r}
 1011 \\
 \times 1001 \\
 \hline
 1011 & 00 & 1011 \rightarrow 11 & 0000 \\
 1011 & 00 & 0101 & 0000 \\
 0000X & 00 & +0000 & 0000 \\
 0000XX & 00 & \hline
 10\cancel{0}1+XX & 11 & 0101 \rightarrow 11 & 0000 \\
 \hline
 1100011
 \end{array}$$

1011 → 11 0000
 0101 → 11 0000
 0010 → 011 0000
 0000 → 1100 0000
 1100 → 1100 0000

1100011 → 1100011

Because it is clock registered.

Multiplier



RTL

$T_0: \text{if}(S=1) \{ C, Q, L \leftarrow 0 \}$ $M \leftarrow n * A$

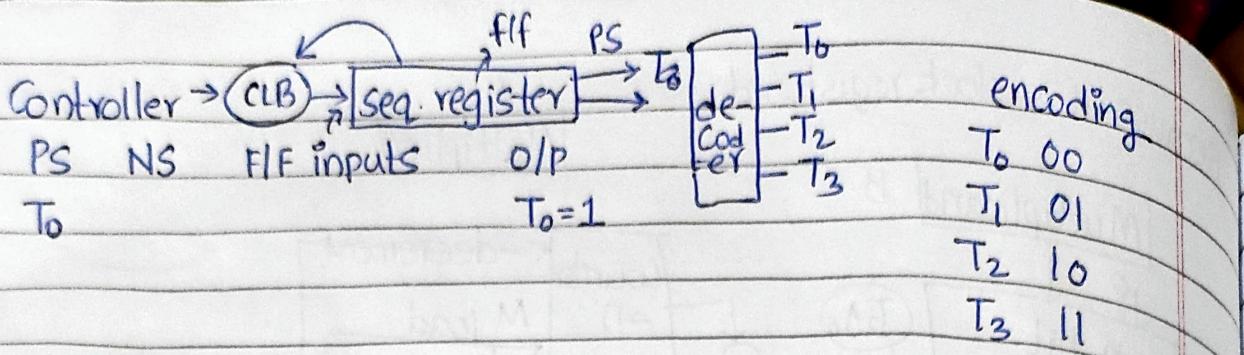
$T_1: C, Q, L \leftarrow 0$

$M \leftarrow M - 1$

$T_2: C, Q \leftarrow Q + B$

$T_3: \text{if}(A_0) \text{then } \{ C, Q \} \leftarrow Q + B$

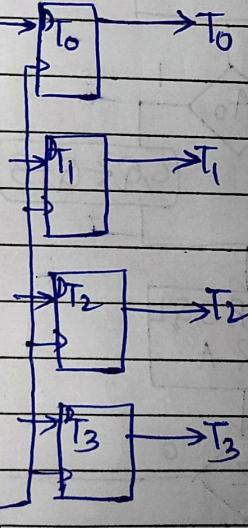
$T_3: \text{Shift-right } C, Q, A$



PS	i/p	NS	D ₁ , D ₀	PS	i/p	NS	D ₁ , D ₀
Q, Q ₀	S, Z	Q ₁ ^t , Q ₀ ^t		Q, Q ₀	S, Z	Q ₁ ^t , Q ₀ ^t	
00	0, 0	00		00	0 X	00	1 0 0 0
00	0, 1	00		<u>T₀</u>	0 0	1 X	0 1
00	1, 0	01		<u>T₁</u>	0 1	X X	1 0
00	1, 1	01		<u>T₂</u>	1 0	X X	1 1
01	0, 0	10		<u>T₃</u>	1 1	X 0	1 0
01	0, 1	10			1 1	X 1	0 0
01	1, 0	10					0 0 0 1
01	1, 1	10					0 0 0 1
10	0, 0	11					
10	0, 1	11					
10	1, 0	11					
10	1, 1	11					
11	0, 0	10					
11	0, 1	10					
11	1, 0	00					
11	1, 1	00					

one-hot encoding
 $T_3 T_2 T_1 T_0$
 0000

0001 one flip-flop for every state.



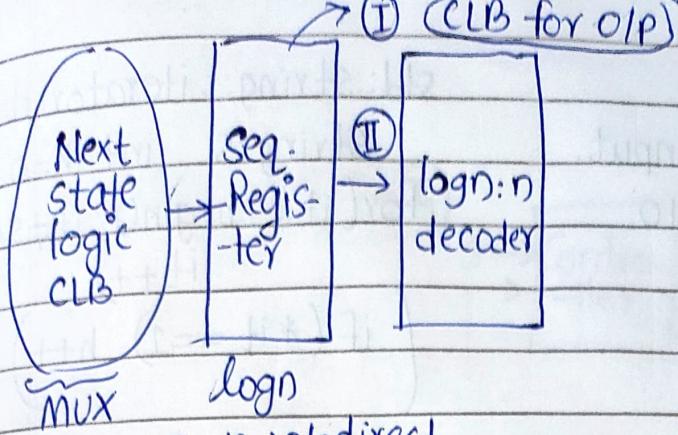
i) Make T_0 high and others low by asynchronous signals

$$D_{T_0} = T_0 \bar{S} + T_3 Z$$

$$D_{T_1} = T_0 S + \bar{T}_3$$

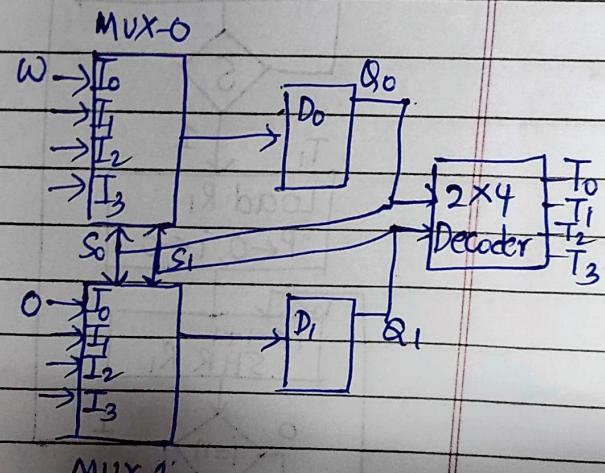
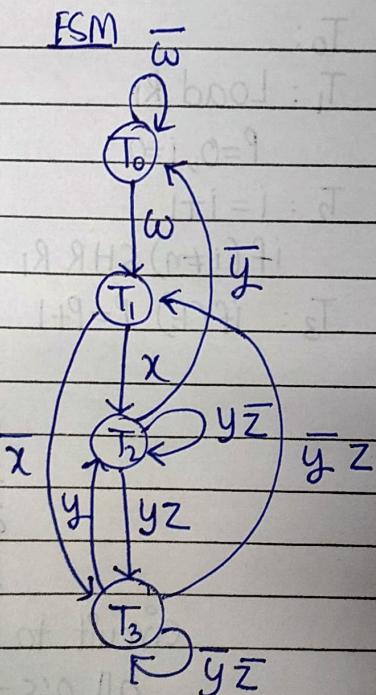
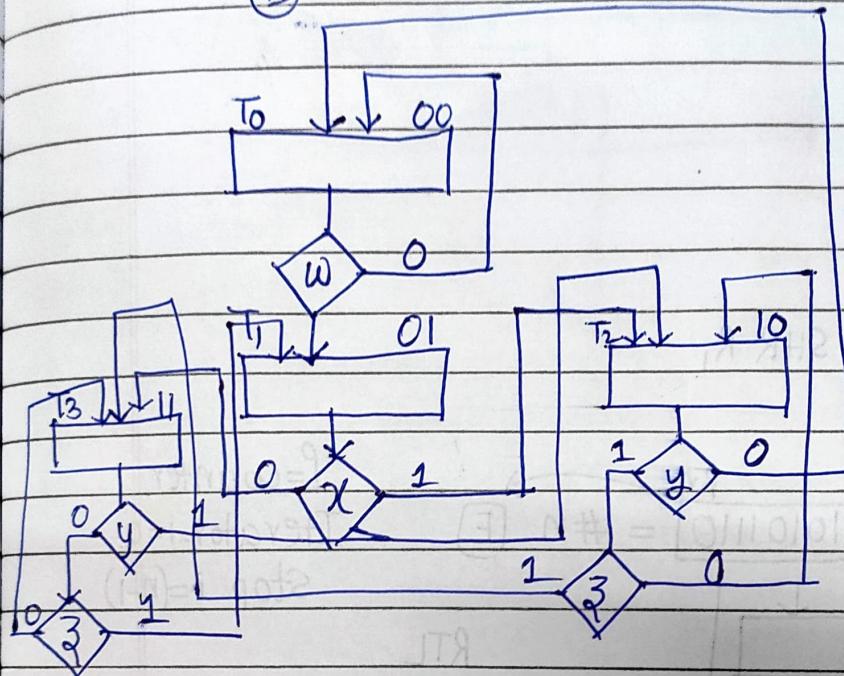
$$D_{T_2} = T_1 + T_2 + T_3 \bar{Z}$$

$$D_{T_3} = T_2$$



(II) logn

(III) n→o/p direct



4 inputs = no. of states

PS	D ₁	D ₀	NS	Cond ⁿ
00	00		00	w̄
00	01		01	w

ASM-based design for
Counting #1s in the input.

10101110

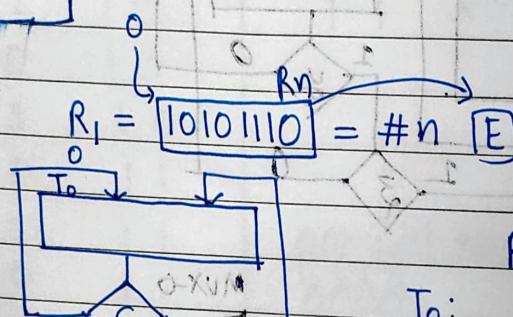
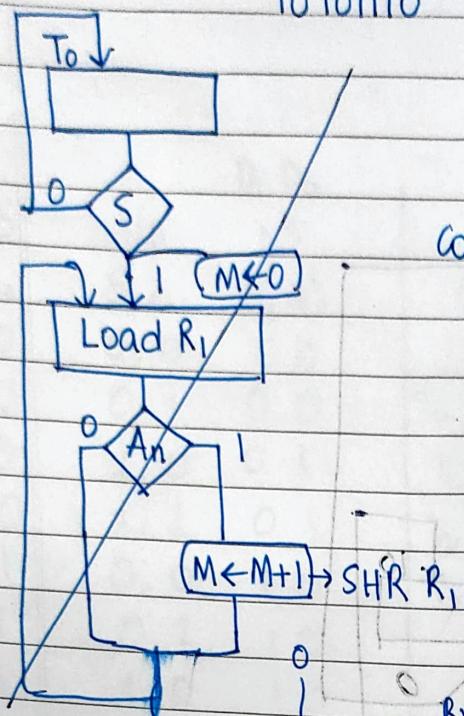
`std::string::iterator it.`

`String's int h=0`

`for(it=s.begin(); it!=s.end(); it++)`

`{ if (*it == 1) h++ }`

Counter M



$P = \text{Counter}$
 $\text{Iterator} = i = 0$
Stop $i = (n-1)$

RTL

$T_0:$

$T_1: \text{Load } R_1$

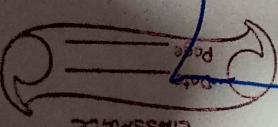
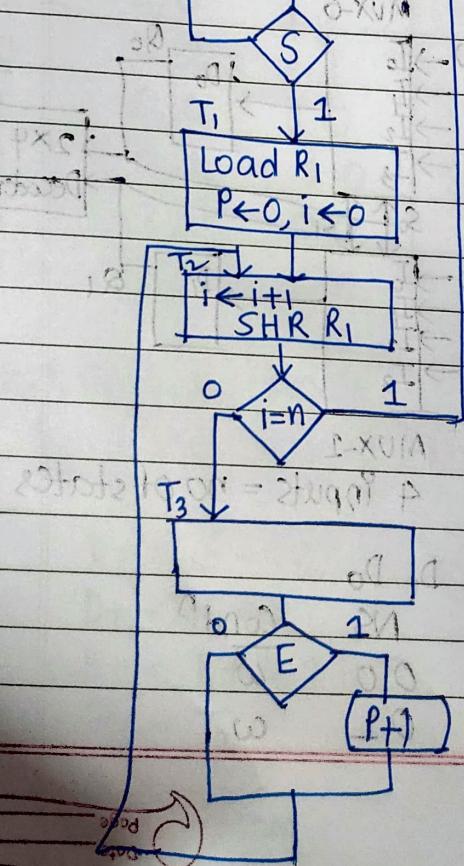
$P=0, i=0$

$T_2: i = i + 1$

$\text{if } (i \neq n) \text{ SHR } R_1$

$T_3: \text{if } (E) P = P + 1$

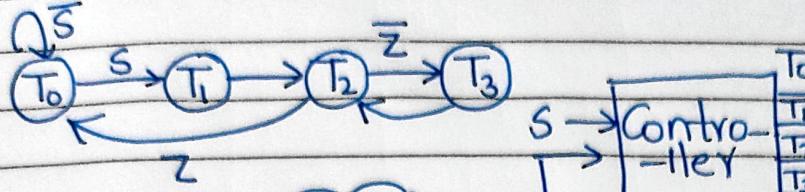
$n=3$



Circuit to check
all 0's $\rightarrow R_1$ becomes
all zero

Z
instead of
 $i=n$ replace
with

ASM



Datapath

