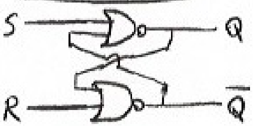
NEELU SARASWATIBHATLA (SRN52)

1. A synchronous FSM has states, and its current state determines its next state. The two types are:
   - Mealy machine – Inputs and outputs appear on transitions
   - Moore machine – Only inputs appear on transitions

2. **SR-FF**

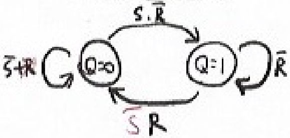
SR-Latch diagram



Truth table

S	R	Q'	$\overline{Q}'$
0	0	Q	$\overline{Q}$
0	1	0	1
1	0	1	0
1	1	0	0

State Diagram



Excitation Table

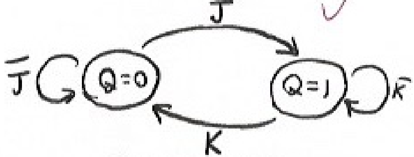
Q	Q'	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	x	0

JK-FF

Truth Table

J	K	Q'	$\overline{Q}'$
0	0	Q	$\overline{Q}$
0	1	0	1
1	0	1	0
1	1	$\overline{Q}$	Q

State Diagram



Excitation Table

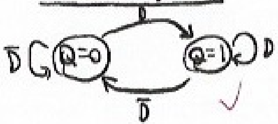
Q	Q'	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

D-FF

Truth Table

D	Q'	$\overline{Q}'$
0	0	1
1	1	0

State Diagram



Excitation Table

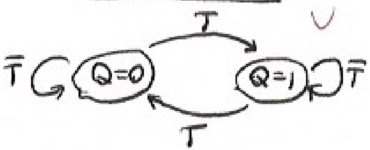
Q	Q'	D
0	0	0
0	1	1
1	0	0
1	1	1

T-FF

Truth Table

T	Q'	$\bar{Q}'$
0	Q	$\bar{Q}$
1	$\bar{Q}$	Q

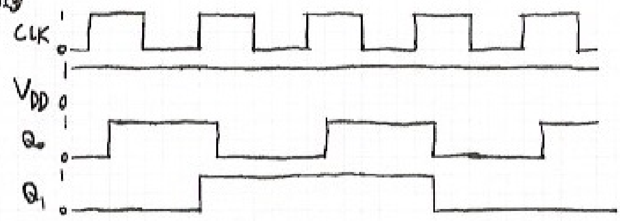
State Diagram



Excitation Table

Q	Q'	T
0	0	$\bar{T}$ 0
0	1	T 1
1	0	T 1
1	1	$\bar{T}$ 0

3 ⓐ



ⓑ

$Q_3 Q_2 Q_1 Q_0$	$Q_3' Q_2' Q_1' Q_0'$	$T_3 T_2 T_1 T_0$
0 0 0 0	0 0 0 1	0 0 0 1
0 0 0 1	0 0 1 0	0 0 1 1
0 0 1 0	0 0 1 1	0 0 0 1
0 0 1 1	0 1 0 0	0 1 1 1
0 1 0 0	0 1 0 1	0 0 0 1
0 1 0 1	0 1 1 0	0 0 1 1
0 1 1 0	0 1 1 1	0 0 0 1
0 1 1 1	1 0 0 0	1 1 1 1
1 0 0 0	1 0 0 1	0 0 0 1
1 0 0 1	1 0 1 0	0 0 1 1
1 0 1 0	1 0 1 1	0 0 0 1
1 0 1 1	1 1 0 0	0 1 1 1
1 1 0 0	1 1 0 1	0 0 0 1
1 1 0 1	1 1 1 0	0 0 1 1
1 1 1 0	1 1 1 1	0 0 0 1
1 1 1 1	0 0 0 0	1 1 1 1

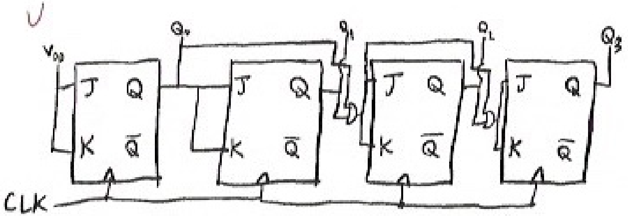
$T_0 = 1$

$T_1 = Q_0$

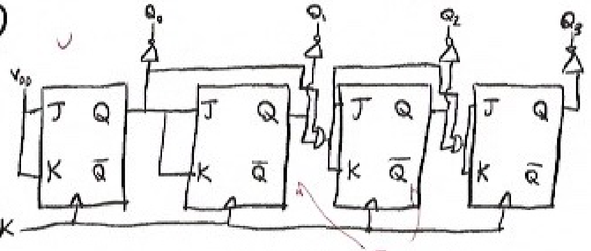
$T_2 = Q_0 \cdot Q_1$

$T_3 = Q_0 \cdot Q_1 \cdot Q_2$

(Continued on next page)

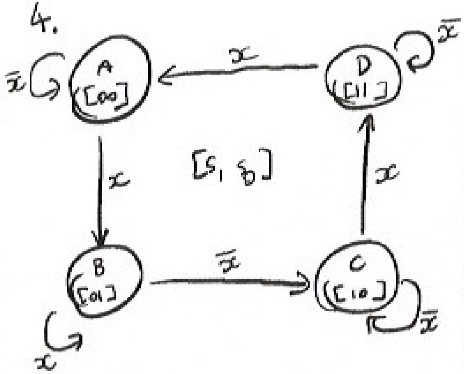


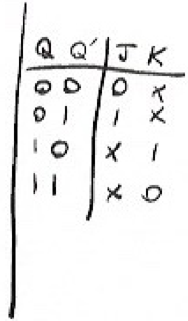
(c)



or you could use $\bar{Q}$ outs

4.





Q	Q′	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

S_1	S_0	x	S_1'	S_0'	J_1	K_1	J_0	K_0
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

J_1

$x \backslash S_1S_0$	00	01	11	10
0	0	1	X	X
1	0	0	X	X

$J_1 = S_0 \cdot \bar{x}$ ✓

K_1

$x \backslash S_1S_0$	00	01	11	10
0	X	X	0	0
1	X	X	1	0

$K_1 = S_0 \cdot x$ ✓

J_0

$x \backslash S_1S_0$	00	01	11	10
0	0	X	X	0
1	1	X	X	1

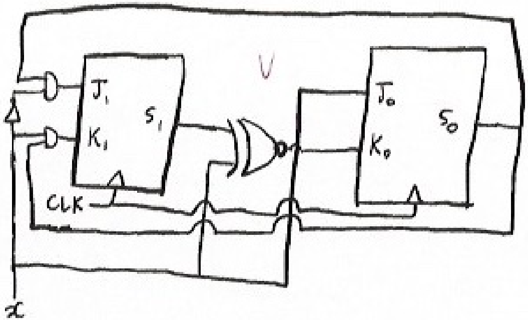
$J_0 = x$ ✓

K_0

$x \backslash S_1S_0$	00	01	11	10
0	X	1	0	X
1	X	0	1	X

$K_0 = \bar{S}_1 \cdot \bar{x} + S_1 \cdot x$

$= \overline{S_1 \oplus x}$ ✓



5. $J_0 = \bar{Q}_1$ $\quad J_1 = Q_0$

$K_0 = 1$ $\quad K_1 = 1$

Q	Q'	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

K=1 $\Rightarrow$

Q	Q'	J
0	0	0
0	1	1
1	0	0
1	0	1

$\Rightarrow$ When $J=0$, $Q'=0$

When $J=1$, $Q' = \bar{Q}$

If $Q_0 = Q_1 = 0$ at the start

CLOCK CYCLE	Q_1	Q_0	J_1	J_0
0	0	0	0	1
1	0	1	1	1
2	1	0	0	0
3	0	0	0	1
4	0	1	1	1
5	1	0	0	0

ctr mod 3

This goes through $(Q_1, Q_0) = (0,0)$, $(0,1)$ and $(1,0)$ ✓ repeats. Q_1 and Q_0 are each on for one clock cycle (out of phase with each other) and out for two. Now $(Q_1, Q_0) = (1,1)$ be investigated.

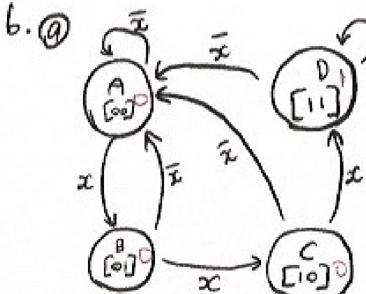
If $Q_1 = Q_0 = 1$,

CLOCK CYCLE	Q_1	Q_0	J_1	J_0
0	1	1	1	0
1	0	0	1	1

... and this then goes into the cycle from before.

$\therefore Q_0$ is on for one clock cycle and off for two and so is Q_1 but is one cycle after Q_0.

6. ⓐ



$[S_1 S_0]$

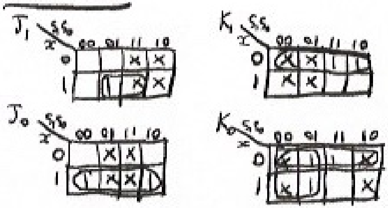
output $= S_1 . S_0$ ok !

ⓑ Moore machine

ⓒ

S_1	S_0	x	S_1'	S_0'	J_1	K_1	J_0	K_0	D_1	D_0
0	0	0	0	0	0	X	0	X	0	0
0	0	1	0	1	0	X	1	X	0	1
0	1	0	0	0	0	X	X	1	0	0
0	1	1	1	0	1	X	X	1	1	0
1	0	0	0	0	X	1	0	X	0	0
1	0	1	1	1	X	0	1	X	1	1
1	1	0	0	0	X	1	X	1	0	0
1	1	1	1	1	X	0	X	0	1	1

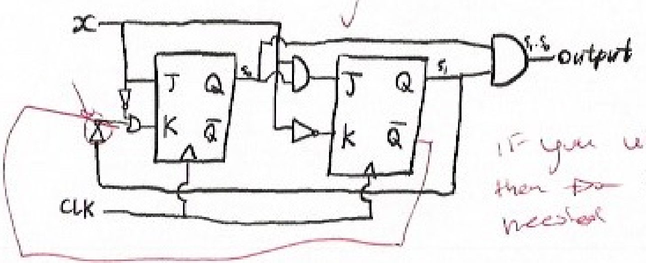
ⓓ JK - FFs



$J_1 = S_0 . x$ ✓
$K_1 = \bar{x}$ ✓

$J_0 = x$ ✓
$K_0 = \bar{x} + \bar{S_1}$ ✓

✓



If you use $\bar{Q}$ then ▷o is not needed

D-FFs

D_1

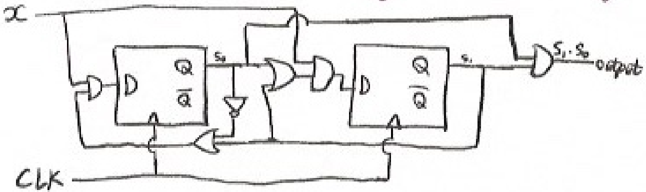
$x \backslash S_1S_0$	00	01	11	10
0				
1		1	1	1

$D_1 = S_0 . x + S_1 . x$
$= (S_0 + S_1) . x$

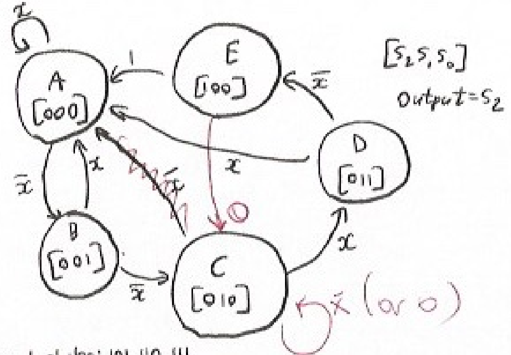
D_0

$x \backslash S_1S_0$	00	01	11	10
0				
1	1		1	1

$D_0 = S_1 . x + \overline{S_0} . x$
$= (\overline{S_0} + S_1) . x$



7.(a)



$[S_2 S_1 S_0]$

Output $= S_2$

unused states: 101, 110, 111

(b)

Current State	Next State $x=0$	Next State $x=1$
A	B	A
B	C	A
C	A	D
D	E	A
E	C	A

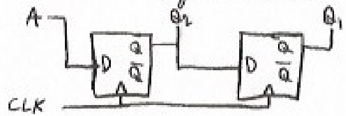
There aren't any redundant states here...

8.(a)

Q_2	Q_1	A	Q_2'	Q_1'	D_2	D_1
0	0	0	0	0	0	0
0	0	1	1	0	1	0
0	1	0	0	0	0	0
0	1	1	1	0	1	0
1	0	0	0	1	0	1
1	0	1	1	1	1	1
1	1	0	0	1	0	1
1	1	1	1	1	1	1

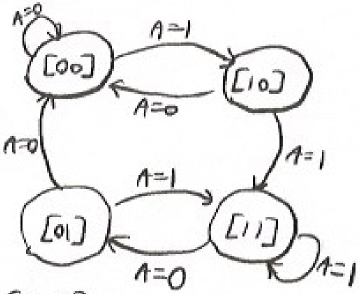
$D_2 = A$
$D_1 = Q_2$

∴ This can be implemented by connecting the D-flip flops in the form of a shift register as follows:



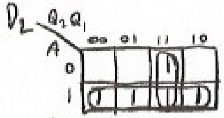
(b) As the clock rate increases, the D-flip flops have less time to both switch as there is a propagation delay in a shift register, so output errors are more likely, and the dotted line transitions are less likely to happen. +/-

(c) ①

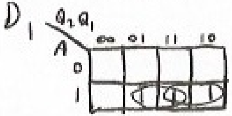


[Q2 Q1]

Q_2	Q_1	A	Q_2'	Q_1'	D_2	D_1
0	0	0	0	0	0	0
0	0	1	1	0	1	0
0	1	0	0	0	0	0
0	1	1	1	1	1	1
1	0	0	0	0	0	0
1	0	1	1	1	1	1
1	1	0	0	0	1	0
1	1	1	1	1	1	1

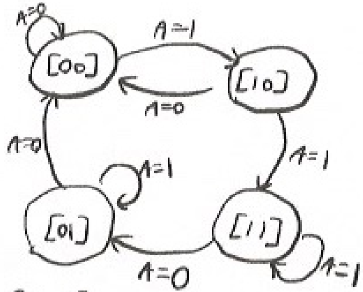


$D_2 = A + Q_2 . Q_1$



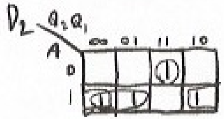
$D_1 = A.Q_2 + A.Q_1$
$= A.(Q_2 + Q_1)$

②



[Q2 Q1]

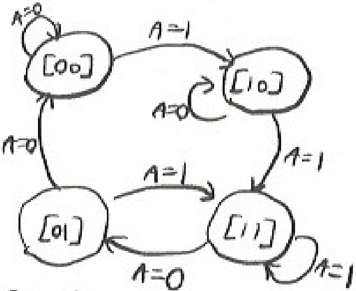
Q_2	Q_1	A	Q_2'	Q_1'	D_2	D_1
0	0	0	0	0	0	0
0	0	1	1	0	1	0
0	1	0	0	0	0	0
0	1	1	0	1	0	1
1	0	0	0	0	0	0
1	0	1	1	1	1	1
1	1	0	0	0	1	0
1	1	1	1	1	1	1



$D_2 = A.\overline{Q_2} + A.\overline{Q_1} + \overline{A}.Q_2.Q_1$
$= A.(\overline{Q_2} + \overline{Q_1}) + \overline{A}.Q_2.Q_1$

$D_1 = A.Q_2 + A.Q_1$
$= A.(Q_2 + Q_1)$

③



[Q_2 Q_1]

Q_2	Q_1	A	Q_2'	Q_1'	D_2	D_1
0	0	0	0	0	0	0
0	0	1	1	0	1	0
0	1	0	0	0	0	0
0	1	1	1	1	1	1
1	0	0	1	0	1	0
1	0	1	1	1	1	1
1	1	0	0	0	1	0
1	1	1	1	1	1	1

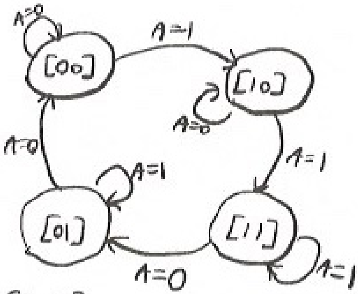
D_2

A \ Q_2Q_1	00	01	11	10
0			1	1
1	1	1	1	1

$D_2 = A + Q_2$

$D_1 = A.Q_2 + A.Q_1$
$= A.(Q_2 + Q_1)$

④



[Q_2 Q_1]

Q_2	Q_1	A	Q_2'	Q_1'	D_2	D_1
0	0	0	0	0	0	0
0	0	1	1	0	1	0
0	1	0	0	0	0	0
0	1	1	0	1	0	1
1	0	0	1	0	1	0
1	0	1	1	1	1	1
1	1	0	0	0	1	0
1	1	1	1	1	1	1

D_2

A \ Q_2Q_1	00	01	11	10
0			1	1
1	1		1	1

$D_2 = Q_2 + A.\overline{Q_1}$?

$D_1 = A.Q_2 + A.Q_1$?
$= A.(Q_2 + Q_1)$