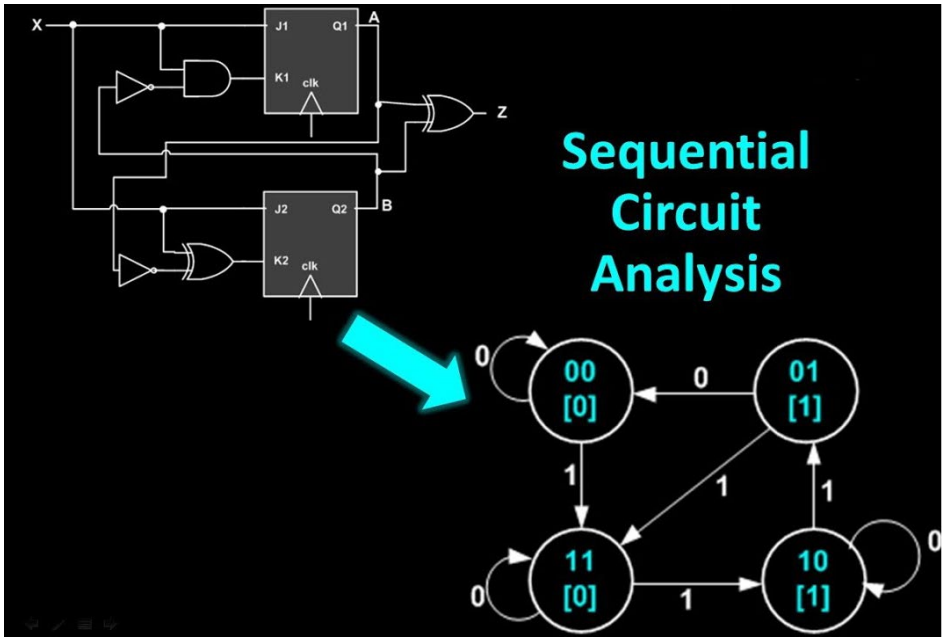
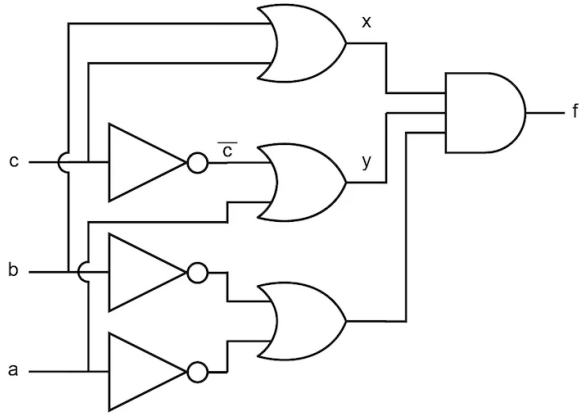
EE2000 Logic Circuit Design

Lecture 9 – Sequential Logic Circuit Design



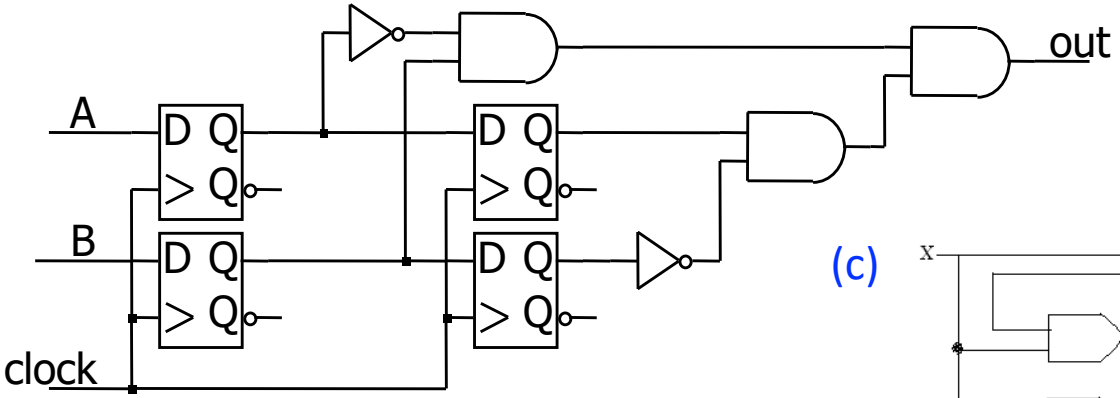
Examples (Mealy or Moore?)

(a)



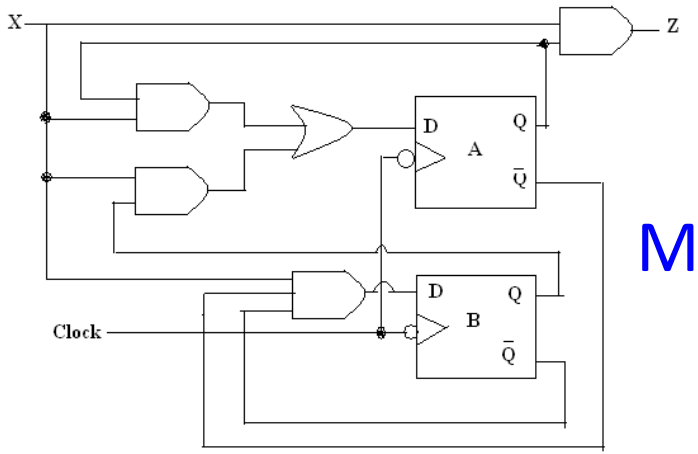
None

(b)



Moore

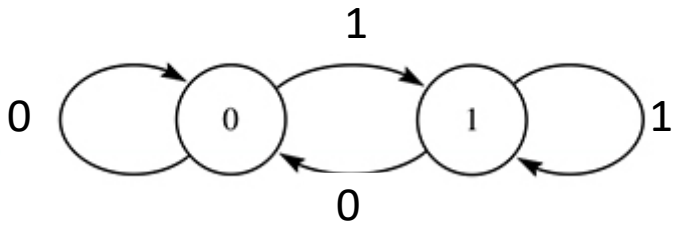
(c)



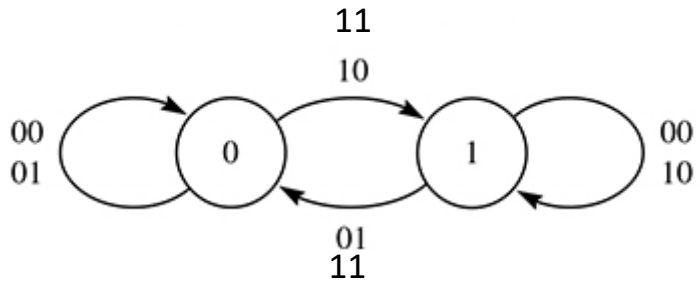
Mealy

Other FFs

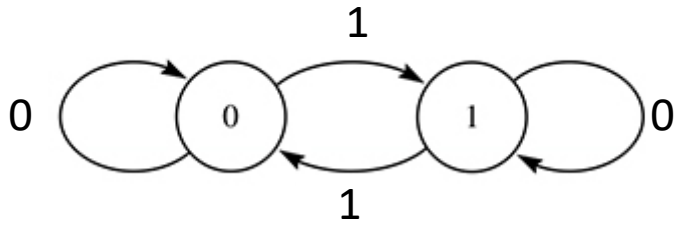
State diagram



D-FF



JK-FF



T-FF

Characteristic equation

$$Q_t^* = D$$

$$Q_t^* = JQ_t' + K'Q_t$$

Q_t^* (JK)	00	01	11	10
0	0	0	1	1
1	1	0	0	1

Q_t^* (T)	0	1
0		1
1	1	

$$Q_t^* = T \oplus Q_t$$

Example (Sequence Detector)

Design a Moore machine to detect the sequence “111” (Overlapping)

In other words,

Design a system with one input x and one output z such that z = 1 if x has been 1 for at least three consecutive clock times.

x	0	1	1	0	1	1	1	0	1	1	1	1	1	0
z	0	0	0	0	0	0	0	1	0	0	0	1	1	1

Question: How about no overlapping is allowed?

x	1	1	1	1	1	0
z	0	0	0	1	0	0

Example (Sequence Detector)

x	0	1	1	0	1	1	1	0	1	1	1	1	1	0
z	0	0	0	0	0	0	0	1	0	0	0	1	1	1

STEP 1: Determine what needs to be stored in memory and how to store them.

A: Input is ‘0’

B: one ‘1’ is detected

C: two ‘1’s are detected

D: three ‘1’s are detected and output 1

Example (Sequence Detector)

x	1	1	1	1	1	1	0
z	0	0	0	1	0	0	1

STEP 2: Work out the State Diagram

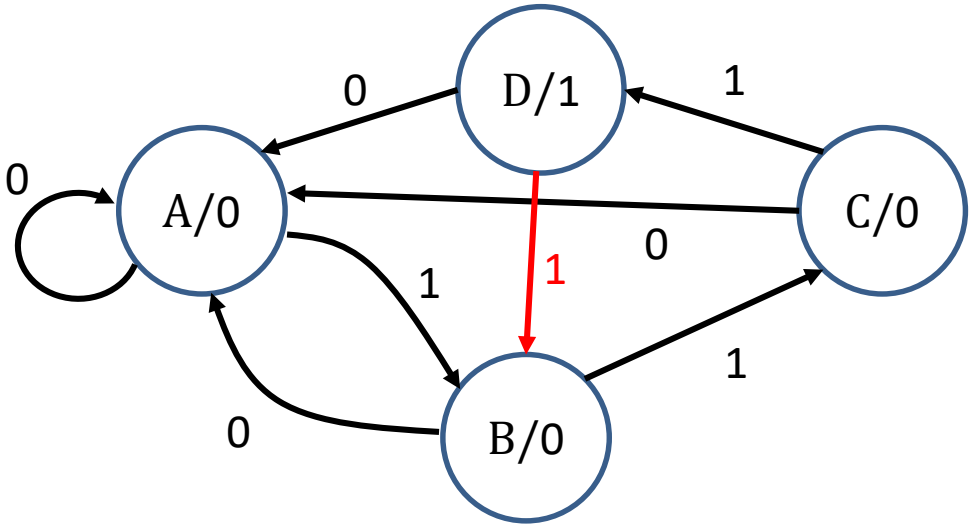
A: Input is '0'

B: one '1' is detected

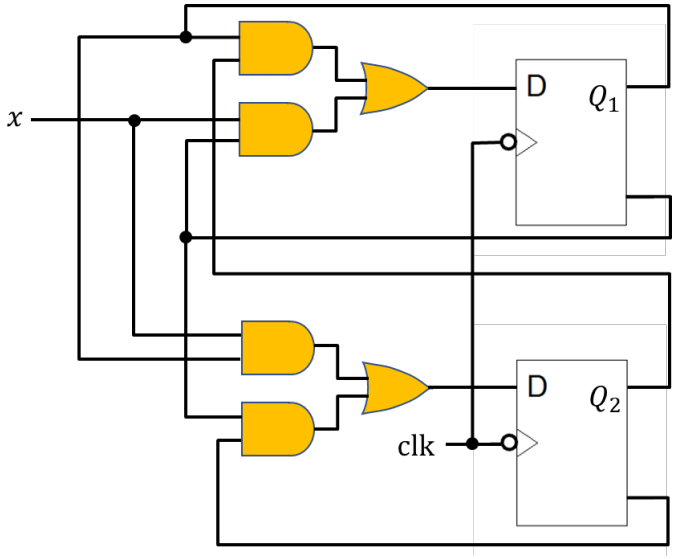
C: two '1's are detected

D: three '1's are detected and output 1

Question: How can we change the diagram if no overlapping is allowed?



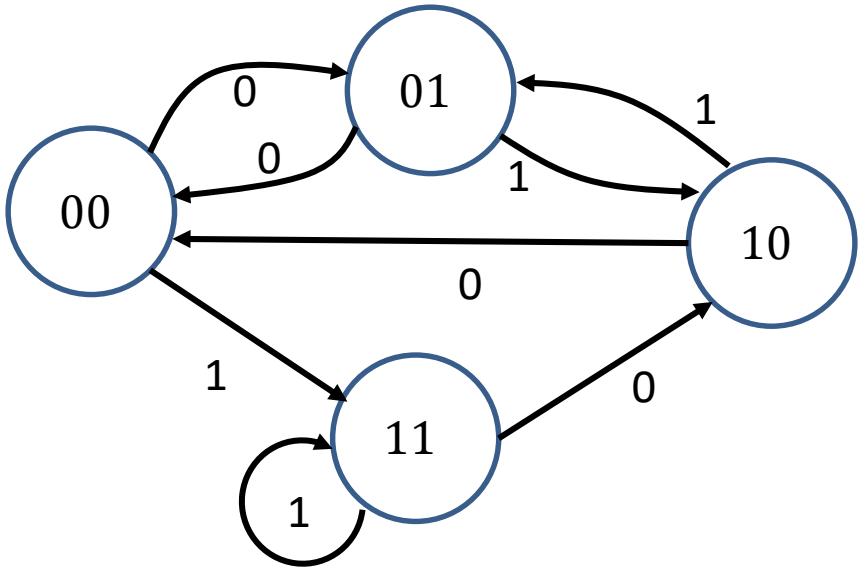
Exercise



Present State $Q_1\ Q_2$	Input *X* 0	Input *X* 1
(0 0)	(0 1)	(1 1)
(0 1)	(0 0)	(1 0)
(1 0)	(0 0)	(0 1)
(1 1)	(1 0)	(1 1)

$$Q_1^* = D_1 = Q_1Q_2 + xQ_1'$$

$$Q_2^* = D_2 = Q_1'Q_2' + xQ_1$$



Exercise (Sequence Detector)

Design a Mealy machine to detect the sequence “111” (Overlapping)

In other words,

Design a system with one input x and one output z such that z = 1 if x has been 1 for at least three consecutive clock times.

x	0	1	1	0	1	1	1	0	1	1	1	1	1	0
z	0	0	0	0	0	0	1	0	0	0	1	1	1	0

Exercise (Sequence Detector)

x	0	1	1	0	1	1	1	0	1	1	1	1	1	0
z	0	0	0	0	0	0	1	0	0	0	1	1	1	0

STEP 1: Determine what needs to be stored in memory and how to store them.

A: input is '0' *if next input is 0, remains at A else B

B: one '1' is detected *if next input is 0, back to A else C

C: two '1's are detected

* if next input is 0, back to A and output '0', else remains at C and output '1'

Exercise (Sequence Detector)

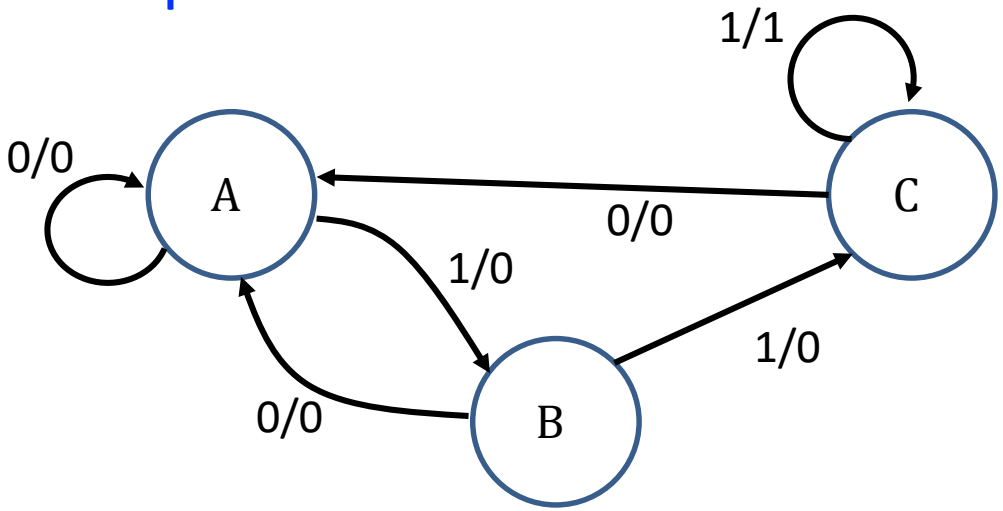
STEP 2: Work out the State Diagram

A: Input is '0' *if next input is 0, remains at A else B

B: one '1' is detected *if next input is 0, back to A else C

C: two '1's are detected

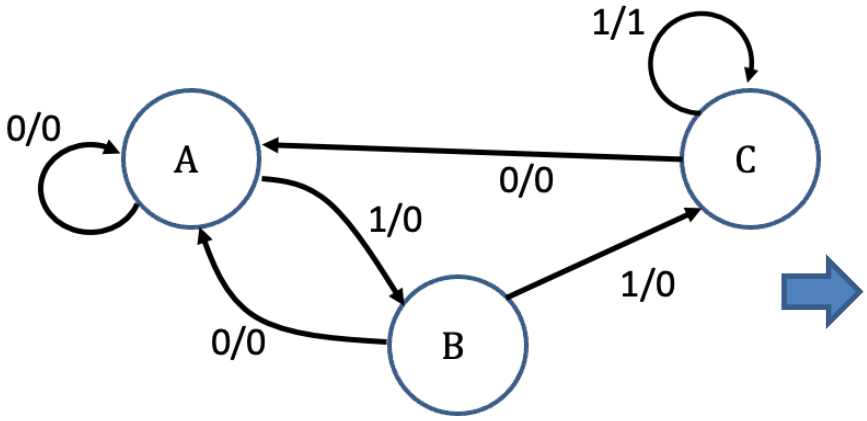
* if next input is 0, back to A and output '0', else remains at C and output '1'



Exercise (Sequence Detector)

STEP 3: Work out the analysis table with assigned FFs

3 states → 2 FFs (We use D-FFs in this example)



Present State (Q_1Q_2)	Input X	Next stage Q_1^*	Q_2^*	Present Output Z
A (0 0)	0	0	0	0
A (0 0)	1	0	1	0
B (0 1)	0	0	0	0
B (0 1)	1	1	1	0
(1 0)	x	x	x	x
C (1 1)	0	0	0	0
C (1 1)	1	1	1	1

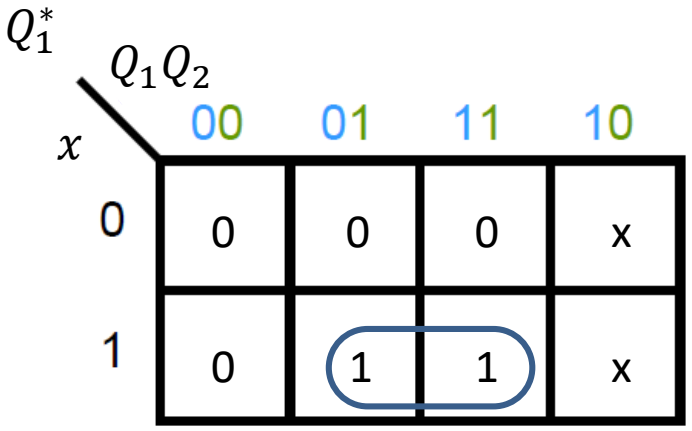
Assign State A:

$Q_1 \rightarrow 0$ and $Q_2 \rightarrow 0$ etc

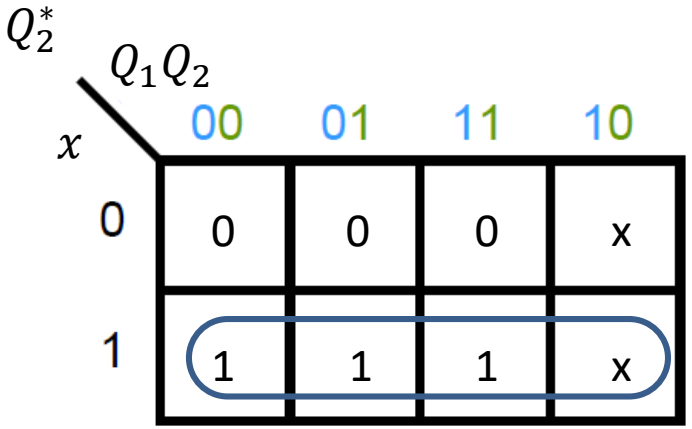
Exercise (Sequence Detector)

STEP 4: Work out D_1 and D_2

Present State (Q_1Q_2)	Input X	Next stage Q_1^*	Q_2^*	Present Output Z
A (0 0)	0	0	0	0
A (0 0)	1	0	1	0
B (0 1)	0	0	0	0
B (0 1)	1	1	1	0
(1 0)	x	x	x	x
C (1 1)	0	0	0	0
C (1 1)	1	1	1	1



$$D_1 = Q_1^* = xQ_2$$

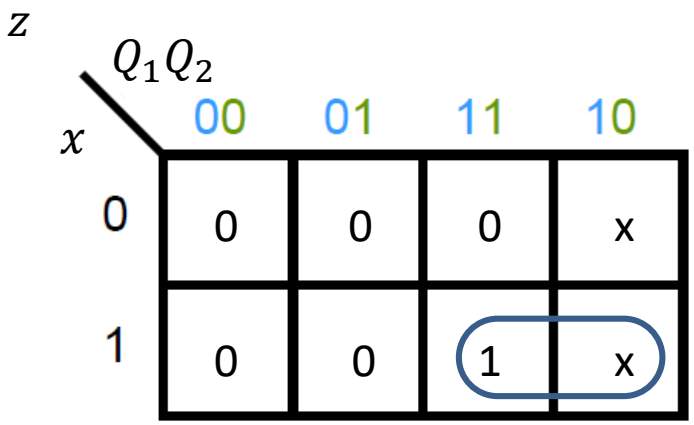


$$D_2 = Q_2^* = x$$

Exercise (Sequence Detector)

STEP 5: Work out z

Present State (Q_1Q_2)	Input X	Next stage Q_1^*	Q_2^*	Present Output Z
A (0 0)	0	0	0	0
A (0 0)	1	0	1	0
B (0 1)	0	0	0	0
B (0 1)	1	1	1	0
(1 0)	x	x	x	x
C (1 1)	0	0	0	0
C (1 1)	1	1	1	1

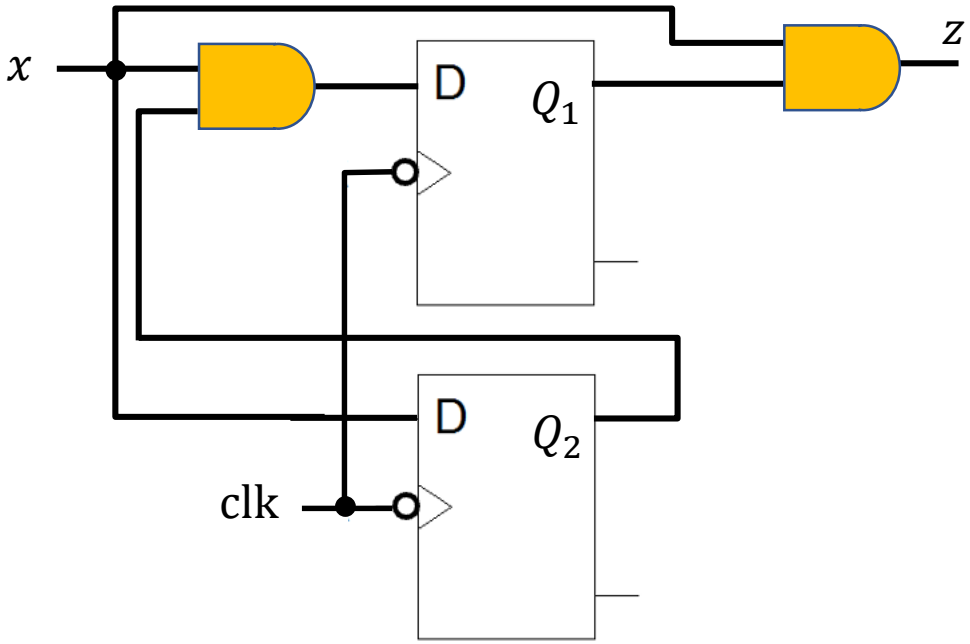


$$z = xQ_1$$

Exercise (Sequence Detector)

STEP 6: Draw the sequential logic circuits

$D_1 = xQ_2$ $D_2 = x$ $z = xQ_1$



Exercise (Sequence Detector)

Use T FFs to design a Mealy machine to detect the sequence “111” (Overlapping)

Present State (Q_1Q_2)	Input X	Next stage Q_1^*	Q_2^*	Flip-Flops T_1	T_2	Present Output Z
A (0 0)	0	0	0			0
A (0 0)	1	0	1			0
B (0 1)	0	0	0			0
B (0 1)	1	1	1			0
(1 0)	x	x	x			x
C (1 1)	0	0	0			0
C (1 1)	1	1	1			1

T	Q_{t+1}	$\overline{Q_{t+1}}$	State
0	Q_t	$\overline{Q_t}$	Hold
1	$\overline{Q_t}$	Q_t	Toggle

Exercise (Sequence Detector)

Use T FFs to design a Mealy machine to detect the sequence “111” (Overlapping)

Present State (Q_1Q_2)	Input X	Next stage Q_1^*	Q_2^*	Flip-Flops T_1	T_2	Present Output Z
A (0 0)	0	0	0	0	0	0
A (0 0)	1	0	1	0	1	0
B (0 1)	0	0	0	0	1	0
B (0 1)	1	1	1	1	0	0
(1 0)	x	x	x	x	x	x
C (1 1)	0	0	0	1	1	0
C (1 1)	1	1	1	0	0	1

T	Q_{t+1}	$\overline{Q_{t+1}}$	State
0	Q_t	$\overline{Q_t}$	Hold
1	$\overline{Q_t}$	Q_t	Toggle

Exercise (Sequence Detector)

Present State (Q_1Q_2)	Input X	Flip-Flops T_1	T_2
A (0 0)	0	0	0
A (0 0)	1	0	1
B (0 1)	0	0	1
B (0 1)	1	1	0
(1 0)	x	x	x
C (1 1)	0	1	1
C (1 1)	1	0	0

T_1

x \ Q_1Q_2	00	01	11	10
0	0	0	1	x
1	0	1	0	x

$$T_1 = xQ_1'Q_2 + x'Q_1$$

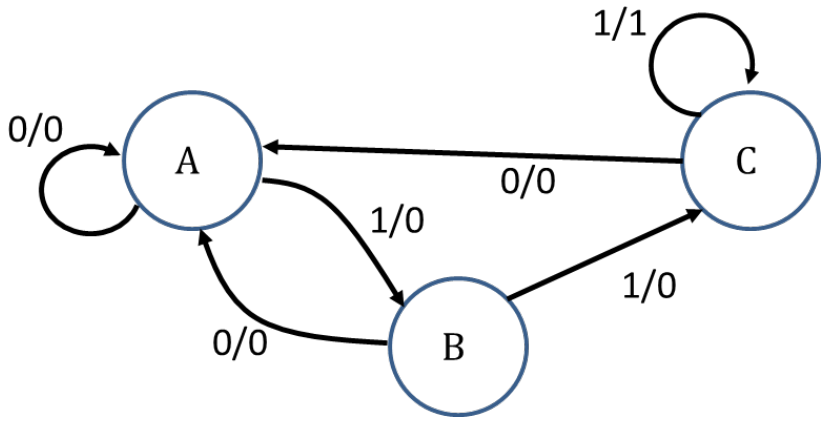
T_2

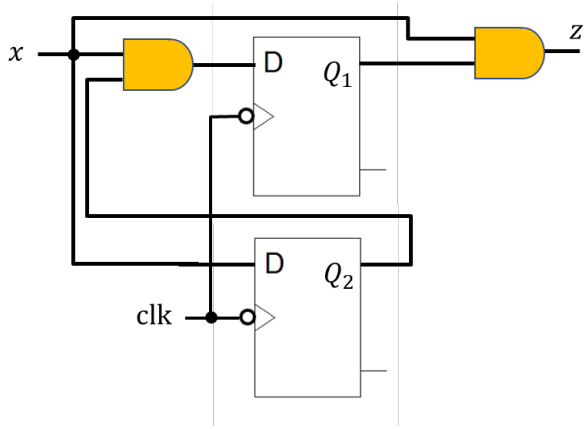
x \ Q_1Q_2	00	01	11	10
0	0	1	1	x
1	1	0	0	x

$$T_2 = xQ_2' + x'Q_2$$

Exercise (Timing Diagram)

$$z = xQ_1$$





Clk														
Present state	A	B	C	C	C	C	C	C	A	B	C	A	B	C
Input x	1	1	1	1	1	1	1	0	1	1	0	1	1	1
Next state	B	C	C	C	C	C	C	A	B	C	A	B	C	C
Q_1														
Output z	0	0	1	1	1	1	1	0	0	0	0	0	0	1

Exercise

Present State	Input *X*		Present Output *Z*
	0	1	
A	I	C	0
B	B	I	0
C	C	G	0
D	I	C	1
E	D	E	1
F	I	C	1
G	E	F	1
H	H	A	0
I	A	C	0

Reduce the state table using partitioning method

$$P_0 = (\mathrm{A\ B\ C\ D\ E\ F\ G\ H\ I})$$

Group states based on same output

Exercise

Present State	Input *X*		Present Output *Z*
	0	1	
A	I	C	0
B	B	I	0
C	C	G	0
H	H	A	0
I	A	C	0
D	I	C	1
E	D	E	1
F	I	C	1
G	E	F	1

Reduce the state table using partitioning method

$$P_1 = (\mathrm{A\ B\ C\ H\ I})(\mathrm{D\ E\ F\ G})$$

A B C H I → Output 0

D E F G → Output 1

Exercise

Present State	Input *X*		Present Output *Z*
	0	1	
A	I	C	0
B	B	I	0
H	H	A	0
I	A	C	0
C	C	G	0
D	I	C	1
F	I	C	1
E	D	E	1
G	E	F	1

Reduce the state table using partitioning method

$$P_2 = (\mathrm{A\ B\ H\ I})(\mathrm{C})(\mathrm{D\ F})(\mathrm{E\ G})$$

Exercise

Present State	Input *X*		Present Output *Z*
	0	1	
A	I	C	0
I	A	C	0
B	B	I	0
H	H	A	0
C	C	G	0
D	I	C	1
F	I	C	1
E	D	E	1
G	E	F	1

Reduce the state table using partitioning method

$$P_3 = (\mathrm{A\ I})(\mathrm{B\ H})(\mathrm{C})(\mathrm{D\ F})(\mathrm{E})(\mathrm{G})$$

Exercise

Present State	Input *X*		Present Output *Z*
	0	1	
A = I	A	C	0
B = H	B	A	0
C	C	G	0
D = F	A	C	1
E	D	E	1
G	E	D	1

Reduce the state table using partitioning method

$$P_3 = (\mathrm{A\ I})(\mathrm{B\ H})(\mathrm{C})(\mathrm{D\ F})(\mathrm{E})(\mathrm{G})$$