Université d'Ottawa
Faculté de génie

École de science informatique
et de génie électrique

uOttawa

L'Université canadienne
Canada's university

University of Ottawa
Faculty of Engineering

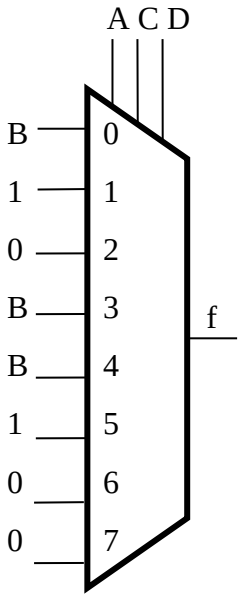
School of Electrical Engineering
and Computer Science

Digital Logic Design: Principles and Practices
ELG5195 (EACJ5705), Carleton CRN: 18371
Assignment #2

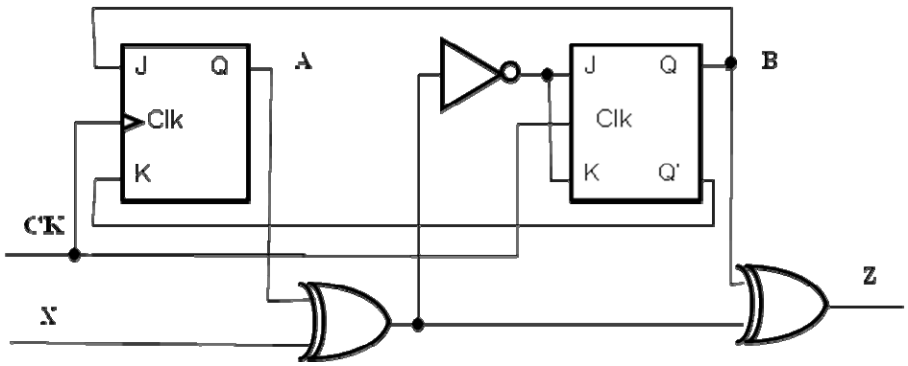
Q1. Implement $f(A,B,C,D) = \Sigma(1,4,5,7,9,12,13)$ using a MUX(8x1) where *A*, *C*, and *D* are connected to the multiplexor's select inputs.

	A	B	C	D	f
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	0

	A	C	D	B	**f**	
0	0	0	0	0	**0**	B
	0	0	0	1	**1**	
1	0	0	1	0	**1**	1
	0	0	1	1	**1**	
2	0	1	0	0	**0**	0
	0	1	0	1	**0**	
3	0	1	1	0	**0**	B
	0	1	1	1	**1**	
4	1	0	0	0	**0**	B
	1	0	0	1	**1**	
5	1	0	1	0	**1**	1
	1	0	1	1	**1**	
6	1	1	0	0	**0**	0
	1	1	0	1	**0**	
7	1	1	1	0	**0**	0
	1	1	1	1	**0**	



Q2. A sequential circuit has two *JK* flip-flops *A* and *B*, one input *X* and one output *Z*. The circuit **logic diagram is shown below**. Derive the circuit **State Table** as well as the **State diagram**.



$JA = B,\ KA = B'$,
$JB = KB = (A \oplus X)' = AX + A'X'$,
$Y = (A \oplus X) \oplus B$
A_{t+1} and B_{t+1} can be derived using the JK characteristic equations.
Ex., $A_{t+1} = A_tK_A' + A_t'J_A = A_t(B')' + A_t'(B) = A_t(B) + A_t'(B) = (A_t + A_t')B = B$

You can also use the characteristic table instead of the equation:

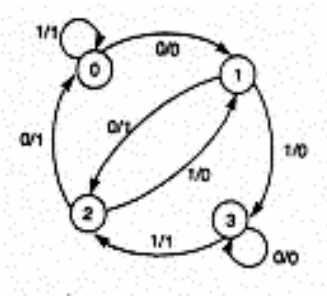
J	K	Q_{t+1}
0	0	Q_t
0	1	0
1	0	1
1	1	Q'_t

A	B	X	JA	KA	$\mathbf{A_{t+1}}$	JB	KB	$\mathbf{B_{t+1}}$
0	0	0	0	1	**0**	1	1	**1**
0	0	1	0	1	**0**	0	0	**0**
0	1	0	1	0	**1**	1	1	**0**
0	1	1	1	0	**1**	0	0	**1**
1	0	0	0	1	**0**	0	0	**0**
1	0	1	0	1	**0**	1	1	**1**
1	1	0	1	0	**1**	0	0	**1**
1	1	1	1	0	**1**	1	1	**0**

State Table:

Present State		In	**Next State**		**Out**
A	B	X	$\mathbf{A_{t+1}}$	$\mathbf{B_{t+1}}$	**Y**
0	0	0	**0**	**1**	**0**
0	0	1	**0**	**0**	**1**
0	1	0	**1**	**0**	**1**
0	1	1	**1**	**1**	**0**
1	0	0	**0**	**0**	**1**
1	0	1	**0**	**1**	**0**
1	1	0	**1**	**1**	**0**
1	1	1	**1**	**0**	**1**

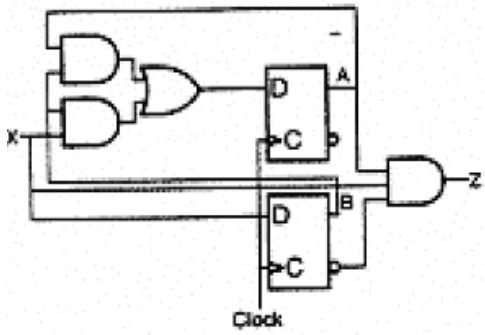
State Diagram



Q3. Using *D flip-flops*, design and implement (devise the logic diagram) the sequential circuit specified by the following state table:

Present State		In	**Next State**		**Out**
A	B	X	$\mathbf{A^{n+1}}$	$\mathbf{B^{n+1}}$	**Z**
0	0	0	**0**	**0**	**0**
0	0	1	**0**	**1**	**0**
0	1	0	**0**	**0**	**0**
0	1	1	**1**	**1**	**0**
1	0	0	**0**	**0**	**0**
1	0	1	**0**	**1**	**1**
1	1	0	**1**	**0**	**0**
1	1	1	**1**	**1**	**0**

$A^{n+1} = AB + BX$ $B^{n+1} = X$



Q4. Using *JK flip-flops*, design and implement (devise the logic diagram) the sequential circuit specified by the following transition table:

Present State		Input	Next State		FF's Inputs (Excitations)			
A	B	X	A^+	B^+	JA	KA	JB	KB
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

JA

A \ BX	00	01	11	10
0	0	0	0	1
1	X	X	X	X

JA = BX'

KA

A \ BX	00	01	11	10
0	X	X	X	X
1	0	0	1	0

KA = BX

JB

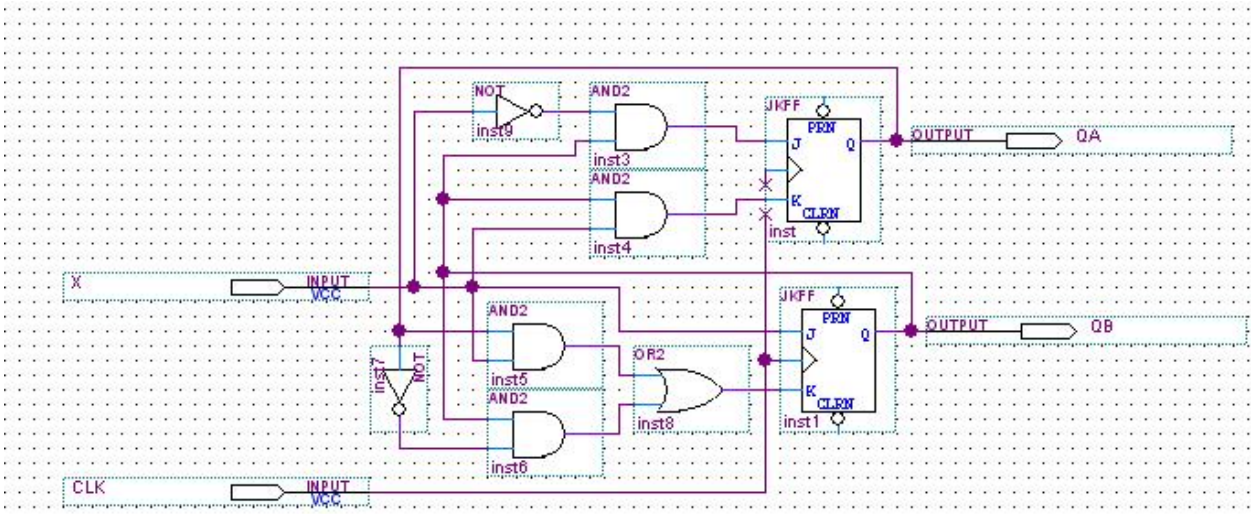
A \ BX	00	01	11	10
0	0	1	X	X
1	0	1	X	X

JB = X

KB

A \ BX	00	01	11	10
0	X	X	0	1
1	X	X	1	0

KB = A'B + AX



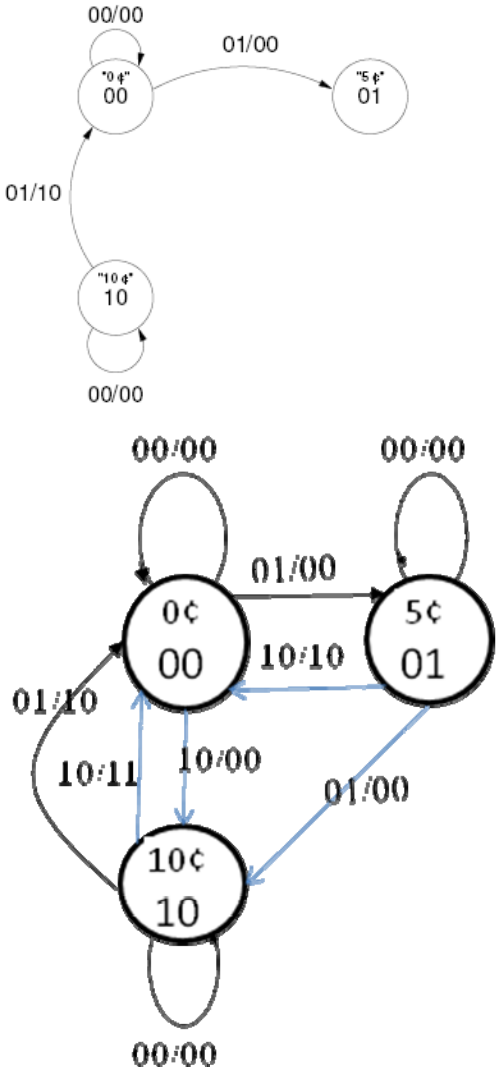
Q5. Design of a vending machine

Design and draw the logical diagram of a vending machine for pop drinks. Each can cost 15 cents. Only coins of 5 cents and of 10 cents are accepted. The circuit has two inputs X_1X_0, and two outputs Y_1Y_0. The behaviour of the circuit is described below:

X_1X_0	Description
0 0	No money is deposited in the machine
0 1	1 nickel is deposited in the machine
1 0	1 dime is deposited in the machine

Y1 = 1 => Dispense a can
Y0 = 1 => Give change a nickel

1. Assuming that the machine starts from state 00, complete the state diagram of the sequential circuit given in the following figure:



2. Derive the state table …

Next state S^{n+1} / Output = $Q_1^+ Q_0^+$ / $Y_1 Y_0$

Q_1Q_0 \ X_1X_0	0 0	0 1	1 1	1 0
00	0 0 : 0 0	0 1 : 0 0	x x : x x	1 0 : 0 0
01	0 1 : 0 0	1 0 : 0 0	x x : x x	0 0 : 1 0
11	x x : x x	x x : x x	x x : x x	x x : x x
10	1 0 : 0 0	0 0 : 1 0	x x : x x	0 0 : 1 1

Present state S^n				Next state S^{n+1}/Output			
Q_1	Q_0	X_1	X_0	Q_1^+	Q_0^+	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0
0	0	1	1	x	x	x	x
0	1	0	0	0	1	0	0
0	1	0	1	1	0	0	0
0	1	1	0	0	0	1	0
0	1	1	1	x	x	x	x
1	0	0	0	1	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	0	1	1
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

… and then the excitation table, given that JK flip-flops are used for the state register of this sequential circuit.

Present state S^n				Next state S^{n+1}/Output				Q_1 input		Q_0 input	
Q_1	Q_0	X_1	X_0	Q_1^+	Q_0^+	Y_1	Y_0	J_1	K_1	J_0	K_0
0	0	0	0	0	0	0	0	0	x	0	x
0	0	0	1	0	1	0	0	0	x	1	x
0	0	1	0	1	0	0	0	1	x	0	x
0	0	1	1	x	x	x	x	x	x	x	x
0	1	0	0	0	1	0	0	0	x	x	0
0	1	0	1	1	0	0	0	1	x	x	1
0	1	1	0	0	0	1	0	0	x	x	1
0	1	1	1	x	x	x	x	x	x	x	x
1	0	0	0	1	0	0	0	x	0	0	x
1	0	0	1	0	0	1	0	x	1	0	x
1	0	1	0	0	0	1	1	x	1	0	x
1	0	1	1	x	x	x	x	x	x	x	x
1	1	0	0	**x**	**x**	**x**	**x**	x	x	x	x
1	1	0	1	**x**	**x**	**x**	**x**	x	x	x	x
1	1	1	0	**x**	**x**	**x**	**x**	x	x	x	x
1	1	1	1	**x**	**x**	**x**	**x**	x	x	x	x

Next state S^{n+1} / Output $= Q_1^+ Q_0^+$ / $Y_1 Y_0$

Q_1Q_0 \ X_1X_0	00	01	11	10
00	00 / 00	01 / 00	xx / xx	10 / 00
01	01 / 00	10 / 00	xx / xx	00 / 10
11	xx / xx	xx / xx	xx / xx	xx / xx
10	10 / 00	00 / 10	xx / xx	00 / 11

3. Derive the simplified excitation equations of the JK flip-flops and the output equations.

$Q_1^+ = Q_0 X_0 + Q'_1 Q'_0 X_1 + Q_1 X'_1 X'_0$

Q_1Q_0 \ X_1X_0	00	01	11	10
00	0	0	x	1
01	0	1	x	0
11	**x**	**x**	**x**	**x**
10	1	0	x	0

$Q_0^+ = Q'_1 Q'_0 X_0 + Q_0 X'_1 X'_0$

Q_1Q_0 \ X_1X_0	00	01	11	10
00	0	1	x	0
01	1	0	x	0
11	**x**	**x**	**x**	**x**
10	0	0	x	0

$Y_1 = \cancel{Q_0 X_0} + Q_0 X_1 + Q_1 X_0 + Q_1 X_1$

Q_1Q_0 \ X_1X_0	00	01	11	10
00	0	0	x	0
01	0	0	x	1
11	**x**	**x**	**x**	**x**
10	0	1	x	1

$Y_0 = Q_1 X_1$

Q_1Q_0 \ X_1X_0	00	01	11	10
00	0	0	x	0
01	0	0	x	0
11	**x**	**x**	**x**	**x**
10	0	0	x	1

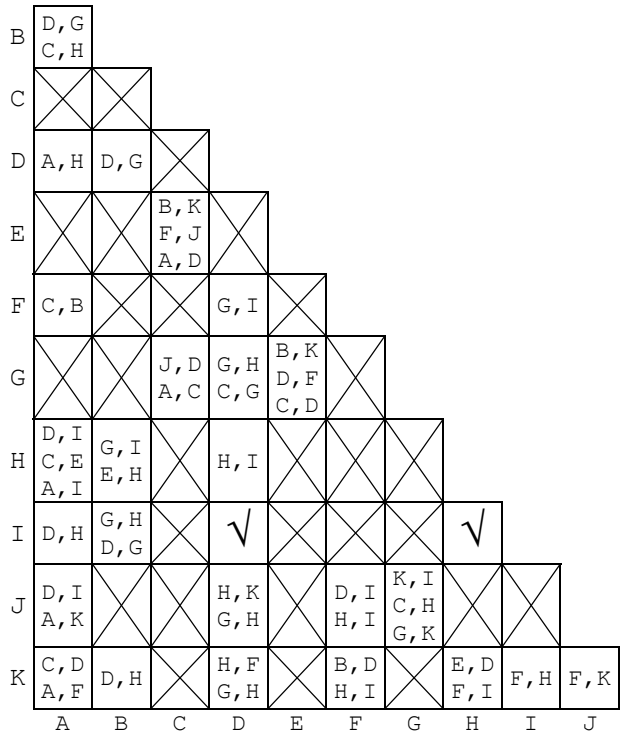
4. Draw the logic diagram of the circuit, using only NAND gates and JK flip flops.

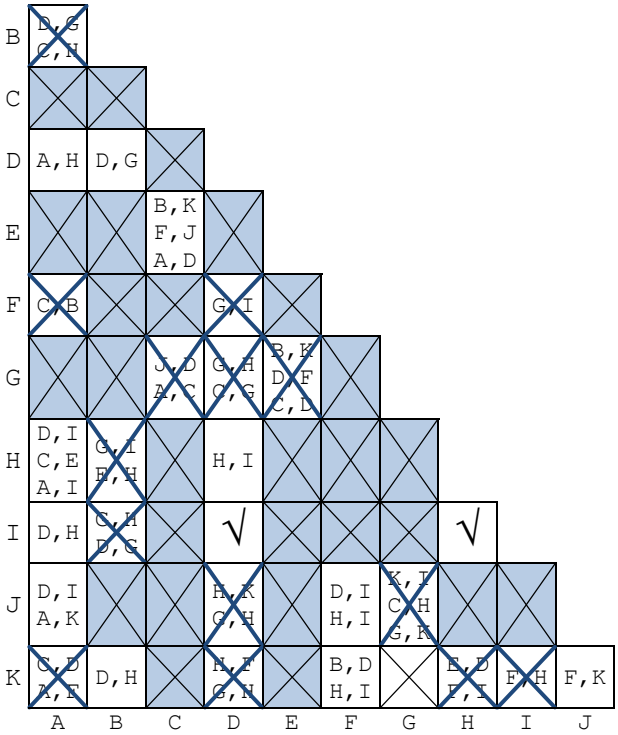
Question 6:

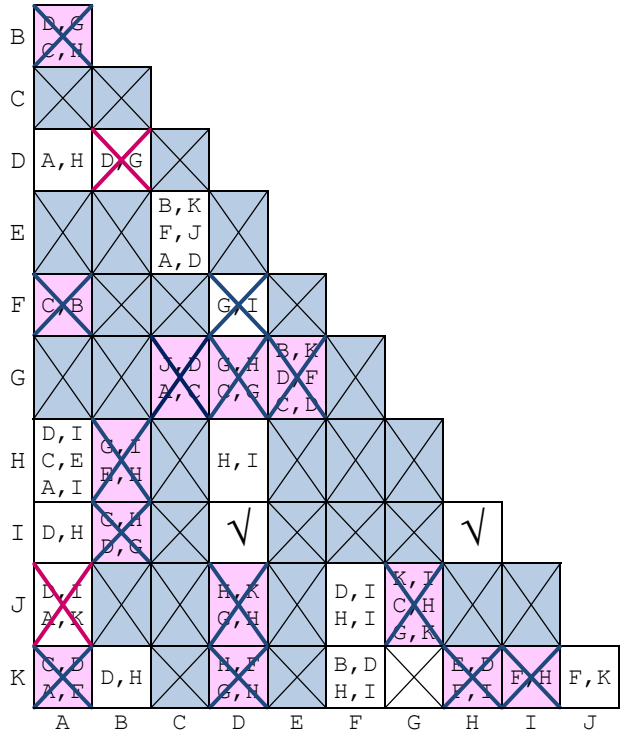
An incompletely specified FSM is specified in the following state table.

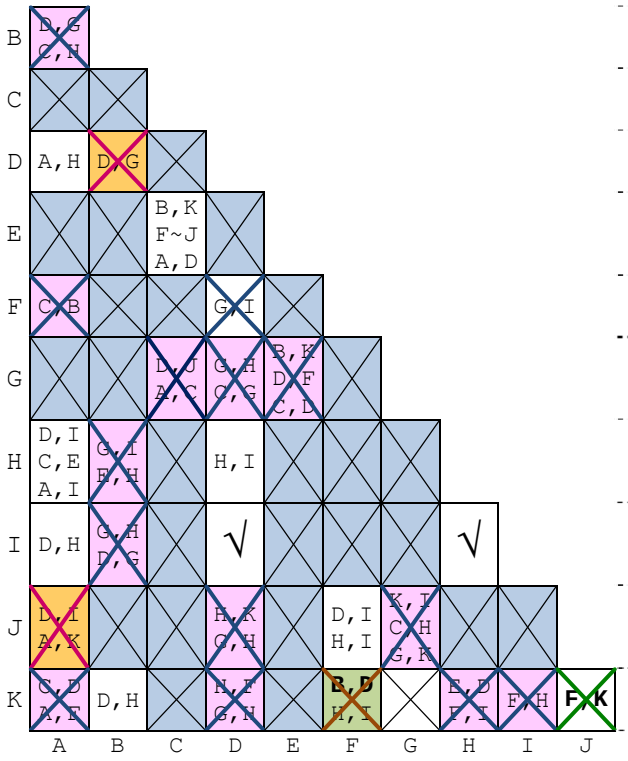
Present state	Next state/output			
	xy=00	xy=01	xy=10	xy=11
A	D/-	C/0	A/-	-/1
B	G/0	H/0	-/-	D/1
C	K/-	J/1	-/-	A/0
D	-/-	-/-	H/0	G/1
E	B/1	F/1	-/-	D/0
F	D/1	B/0	-/-	I/1
G	K/1	D/1	G/-	C/-
H	I/0	E/0	I/0	-/-
I	H/0	-/-	-/-	G/1
J	I/1	-/-	K/0	H/1
K	-/-	D/0	F/0	H/1

1. Use the implication chart method to identify all pairs of compatible states, along with their implied compatibilities.



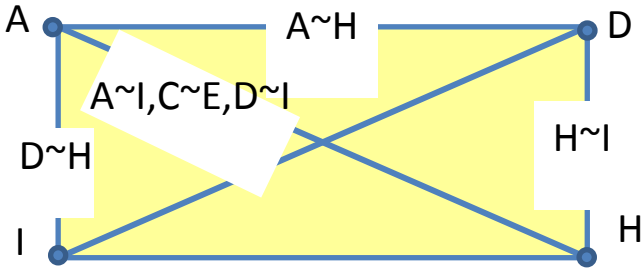


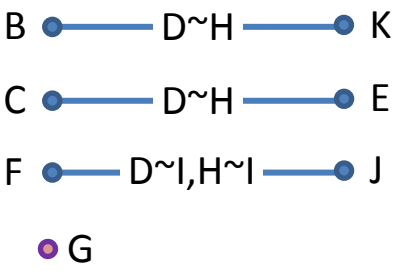




Compatible states?	Implied compatibilities
A~D	A~H
C~E	B~K F~J A~D
A~H D~I	D~I C~E A~I H~I
A~I D~I H~I	D~H - -
F~J	D~I H~I
B~K	D~H

2. Represent the compatible state pairs on a merger diagram and find all classes of maximal compatibility. Find all minimal closed sets of *compatible classes* that cover the original FSM.





3. Use these set(s) to devise a reduced table that is compatible with the original FSM.

	Pr. St.	Next state/output 00	01	10	11
P	A	D/-	C/0	A/-	-/1
	D	-/-	-/-	H/0	G/1
	H	I/0	E/0	I/0	-/-
	I	H/0	-/-	-/-	G/1
Q	B	G/0	H/0	-/-	D/1
	K	-/-	D/0	F/0	H/1
R	C	K/-	J/1	-/-	A/0
	E	B/1	F/1	-/-	D/0
S	F	D/1	B/0	-/-	I/1
	J	I/1	-/-	K/0	H/1
G	G	K/1	D/1	G/-	C/-

or a different table layout:

Pr. St	Next state/output xy 00	01	10	11
P	P/0	R/0	P/0	G/1
Q	G/0	P/0	S/0	P/1
R	Q/1	S/1	-/-	P/0
S	P/1	Q/0	Q/0	P/1
G	Q/1	P/1	G/-	R/-

Pr. St.	Next state xy 00	01	10	11	Output xy 00	01	01	11
P	P	R	P	G	0	0	0	1
Q	G	P	S	P	0	0	0	1
R	Q	S	-	P	1	1	-	0
S	P	Q	Q	P	1	0	0	1
G	Q	P	G	R	1	1	-	-

Draw the state diagram of the reduced FSM.

4. Create the adjacency groups for the reduced state table.

Pr. St	Next state/output xy				Common successor for the same inputs (Rule 1)	Common ancestor for adjacent inputs (Rule 2)	Same output (e.g., "1" to minimize SoP) for the same input
	00	01	11	10			
P	P/0	R/0	G/1	P/0	P-S	P-R, G-P,G-R	P-Q, P-S
Q	G/0	P/0	P/1	S/0	Q-R, Q-S, Q-G	G-P, P-S, G-S	Q-S
R	Q/1	S/1	P/0	-/-	R-S,G-R	Q-S, P-S	R-S, 2xG-R
S	P/1	Q/0	P/1	Q/0		4xP-Q	G-S
G	Q/1	P/1	R/-	G/-		P-Q, P-R, G-R, G-Q	

	Rule 1	Rule 2	Rule 3
P-S	1	2	1
G-R	1	2	2
G-Q	1	1	
Q-R	1		
Q-S	1	1	1
R-S	1		1
P-R		2	
G-P		2	
P-Q		5	1
G-S		1	1

Ordered:

	Rule 1	Rule 2	Rule 3
G-R	1	2	2
P-S	1	2	1
Q-S	1	1	1
G-Q	1	1	
R-S	1		1
Q-R	1		
P-Q		5	1
P-R		2	
G-P		2	
G-S		1	1

5. Plot the state assignment map for the reduced state table if binary encoding of the states is employed.

Variant 1

Q_1Q_0 / Q_2	00	01	11	10
0	**P**			
1	**S**	**Q**	**G**	**R**

Variant 2

Q_1Q_0 / Q_2	00	01	11	10
0	P	S	Q	
1	G	R		

Variant 3

Q_1Q_0 / Q_2	00	01	11	10
0	P	S	Q	R
1			G	

Variant 4

Q_1Q_0 / Q_2	00	01	11	10
0		**P**		
1	**S**	**Q**	**G**	**R**

Rules observed

Rule 1	Rule 2	Rule 3
5	6	5
4	6	5
4	6	2
4	9	5

Derive the transition table.

Pr. St	Next state/output xy			
	00	01	10	11
P	P/0	R/0	P/0	G/1
Q	G/0	P/0	S/0	P/1
R	R/1	S/1	-/-	P/0
S	P/1	Q/0	Q/0	P/1
G	Q/1	P/1	G/-	R/-

Pr. St	Next state/output xy			
	00	01	10	11
P=000	000/0	010/0	000/0	111/1
Q=011	111/0	000/0	001/0	000/1
R=010	010/1	001/1	-/-	000/0
S=001	000/1	011/0	011/0	000/1
G=111	011/1	000/1	111/-	010/-

Pr. St	Next state/output xy			
	00	01	10	11
P=000	000/0	010/0	000/0	111/1
S=001	000/1	011/0	011/0	000/1
R=010	010/1	001/1	-/-	000/0
Q=011	111/0	000/0	001/0	000/1
100	-/-	-/-	-/-	-/-
101	-/-	-/-	-/-	-/-
110	-/-	-/-	-/-	-/-
G=111	011/1	000/1	111/-	010/-

6. Implement the state machine using J-K flip-flops.

Q_2

Pr. St	Next state/output xy			
	00	01	11	10
P=000	0	0	1	0
S=001	0	0	0	0
R=010	0	0	0	-
Q=011	1	0	0	0
100	-	-	-	-
101	-	-	-	-
110	-	-	-	-
G=111	0	0	0	1

J_2

Pr. St	Next state/output xy			
	00	01	11	10
P=000	0	0	1	0
S=001	0	0	0	0
R=010	0	0	0	-
Q=011	1	0	0	0
100	d	d	d	d
101	d	d	d	d
110	d	d	d	d
G=111	d	d	d	d

K_2

Pr. St	Next state/output xy			
	00	01	11	10
P=000	d	d	d	d
S=001	d	d	d	d
R=010	d	d	d	d
Q=011	d	d	d	d
100	d	d	d	d
101	d	d	d	d
110	d	d	d	d
G=111	1	1	1	0

$J_2 = xy\,Q_1'\,Q_0' + x'y'\,Q_1\,Q_0$

Q_2 \ Q_1Q_0	00	01	11	10
0	xy	0	x'y'	0
1	d	d	d	d

$K_2 = x' + y$

Q_2 \ Q_1Q_0	00	01	11	10
0	d	d	d	d
1	d	d	(xy')'	d

Q_1

Pr. St	Next state/output xy			
	00	01	10	11
P=000	0	1	0	1
S=001	0	1	1	0
R=010	1	0	-	0
Q=011	1	0	0	0
100	-	-	-	-
101	-	-	-	-
110	-	-	-	-
G=111	1	0	1	1

J_1

Pr. St	Next state/output xy			
	00	01	10	11
P=000	0	1	0	1
S=001	0	1	1	0
R=010	d	d	d	d
Q=011	d	d	d	d
100	d	d	d	d
101	d	d	d	d
110	d	d	d	d
G=111	d	d	d	d

K_1

Pr. St	Next state/output xy			
	00	01	10	11
P=000	d	d	d	d
S=001	d	d	d	d
R=010	1	0	d	0
Q=011	1	0	0	0
100	d	d	d	d
101	d	d	d	d
110	d	d	d	d
G=111	1	0	1	1

$J_1 = y\,Q_0' + (x \oplus y)\,Q_0\,Q_0$

Q_1Q_0 / Q_2	00	01	11	10
0	y	$x \oplus y$	d	d
1	d	d	d	d

$K_1 = (x + y')\,Q_2 + x'y'\,Q_2'$ or, since x'y'=1 ➔ y'=1➔(x+y')=1,➔ $K_1 = (x + y')Q_2 + x'y$

Q_1Q_0 / Q_2	00	01	11	10
0	d	d	x'y'	x'y'
1	**d**	**d**	**x+y'**	**d**

Q_0

Pr. St	Next state/output xy			
	00	01	10	11
P=000	0	0	0	1
S=001	0	1	1	0
R=010	0	1	-/-	0
Q=011	1	0	1	0
100	-	-	-	-
101	-	-	-	-
110	-	-	-	-
G=111	1	0	1	0

J_0

Pr. St	Next state/output xy			
	00	01	10	11
P=000	0	0	0	1
S=001	d	d	d	d
R=010	0	1	-/-	0
Q=011	d	d	d	d
100	d	d	d	d
101	d	d	d	d
110	d	d	d	d
G=111	d	d	d	d

K_0

Pr. St	Next state/output xy			
	00	01	10	11
P=000	d	d	d	d
S=001	1	0	0	1
R=010	d	d	d	d
Q=011	1	0	1	0
100	d	d	d	d
101	d	d	d	d
110	d	d	d	d
G=111	0	1	0	1

$J_0 = xy\,Q_1' + x'y\,Q_1$

Q_1Q_0 / Q_2	00	01	11	10
0	xy	d	d	x'y
1	d	d	d	d

$K_0 = (x \oplus y)'Q_1'Q_0 + y'\,Q_2'\,Q_1 + y\,Q_2$

Q_1Q_0 / Q_2	00	01	11	10
0	d	$(x \oplus y)'$	y'	d
1	d	d	y	d

Z

Pr. St	Next state/output xy			
	00	01	10	11
P=000	0	0	0	1
S=001	1	0	0	1
R=010	1	1	-	0
Q=011	0	0	0	1
100	-	-	-	-
101	-	-	-	-
110	-	-	-	-
G=111	1	1	-	-

$Z = Q_2 + (x \oplus y)'\,Q_1'Q_0 + xy\,(Q_1 \oplus Q_0) + x'\,Q_1\,Q_0'$

Q_1Q_0 / Q_2	00	01	11	10
0	xy	$(x \oplus y)'$	xy	x'
1	d	d	1	d

OR:

Q_2	Q_1	Q_0	x	y	Q_2^+	Q_1^+	Q_0^+	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	0	0	0	0	d	0	d	0	d
0	0	0	0	1	0	1	0	0	d	1	d	0	d
0	0	0	1	0	0	1	0	0	d	1	d	0	d
0	0	0	1	1	1	0	1	1	d	0	d	1	d
0	0	1	0	0	0	0	0	0	d	0	d	d	0
0	0	1	0	1	0	1	1	0	d	1	d	d	1
0	0	1	1	0	0	0	1	0	d	0	d	d	1
0	0	1	1	1	0	1	0	0	d	1	d	d	0
0	1	0	0	0	0	1	0	0	d	d	0	0	d
0	1	0	0	1	0	0	1	0	d	d	1	1	d
0	1	0	1	0	d	0	d	d	d	d	1	d	d
0	1	0	1	1	0	d	0	0	d	d	d	0	d
0	1	1	0	0	1	1	1	1	d	d	0	d	1
0	1	1	0	1	0	0	0	0	d	d	1	d	0
0	1	1	1	0	0	0	1	0	d	d	1	d	1
0	1	1	1	1	0	0	0	0	d	d	1	d	0
1	0	0	0	0	d	d	d	d	d	d	d	d	d
1	0	0	0	1	d	d	d	d	d	d	d	d	d
1	0	0	1	0	d	d	d	d	d	d	d	d	d
1	0	0	1	1	d	d	d	d	d	d	d	d	d
1	0	1	0	0	d	d	d	d	d	d	d	d	d
1	0	1	0	1	d	d	d	d	d	d	d	d	d
1	0	1	1	0	d	d	d	d	d	d	d	d	d
1	0	1	1	1	d	d	d	d	d	d	d	d	d
1	1	0	0	0	d	d	d	d	d	d	d	d	d
1	1	0	0	1	d	d	d	d	d	d	d	d	d
1	1	0	1	0	d	d	d	d	d	d	d	d	d
1	1	0	1	1	d	d	d	d	d	d	d	d	d
1	1	1	0	0	0	1	1	d	1	d	0	d	1
1	1	1	0	1	0	0	0	d	1	d	1	d	0
1	1	1	1	0	1	1	1	d	0	d	0	d	1
1	1	1	1	1	0	1	0	d	1	d	0	d	0

$J_2 = xy\,Q_1'\,Q_0' + x'y'Q_1Q_0$

$Q_2Q_1Q_0$ \ xy	00	01	11	10
000	0	0	1	0
001	0	0	0	0
011	1	0	0	0
010	0	0	0	d
100	d	d	d	d
101	d	d	d	d
111	d	d	d	d
110	d	d	d	d

$K_2 = x' + y$

$Q_2Q_1Q_0$ \ xy	00	01	11	10
000	d	d	d	d
001	d	d	d	d
011	d	d	d	d
010	d	d	d	d
100	d	d	d	d
101	d	d	d	d
111	1	1	1	0
110	d	d	d	d

$J_1 = y\,Q_0' + (x \oplus y)Q_0$

$Q_2Q_1Q_0$ \ xy	00	01	11	10
000	0	1	1	0
001	0	1	0	1
011	d	d	d	d
010	d	d	d	d
100	d	d	d	d
101	d	d	d	d
111	d	d	d	d
110	d	d	d	d

$K_1 = (x + y')\,Q_2 + x'y'$

$Q_2Q_1Q_0$ \ xy	00	01	11	10
000	***d***	d	d	***d***
001	d	d	d	d
011	1	0	0	0
010	***1***	0	0	***d***
100	***d***	d	**d**	***d***
101	**d**	d	**d**	**d**
111	**1**	0	**1**	**1**
110	***d***	d	**d**	***d***

$J_0 = xy\,Q_1' + x'y\,Q_1$

$Q_2Q_1Q_0$ \ xy	00	01	11	10
000	0	0	1	0
001	d	d	d	d
011	d	d	d	d
010	0	1	0	d
100	d	d	d	d
101	d	d	d	d
111	d	d	d	d
110	d	d	d	d

$K_0 = (x \oplus y)'Q_1'Q_0' + y'\,Q_2'\,Q_1 + \mathbf{y\,Q_2}$

$Q_2Q_1Q_0$ \ xy	00	01	11	10
000	d	d	d	d
001	1	0	1	0
011	1	0	0	1
010	d	d	d	d
100	d	**d**	**d**	d
101	d	**d**	**d**	d
111	0	**1**	**1**	0
110	d	**d**	**d**	d

or : $K_0 = x'y'Q_2' + xyQ_1 + y'Q_2'Q_1 + \mathbf{yQ_2}$

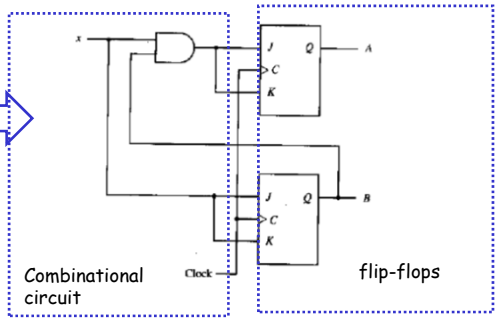
Schematics of the logic circuits

Sequential circuit design

DESIGN=

FROM problem specification
-Text description
-Graphic description

TO



"a sequential circuit is specified by a **time sequence** of **external inputs**, **external outputs**, and internal **flip-flop binary states**" – text book

Sequential circuit design procedure

Step 1:
Make a ***state table*** based on the problem statement. The table should show next states and outputs as function of the present states and inputs. (It may be easier to find a **state diagram** first, and then convert that to a **state table**)

Step 2:
Convert the **state table** into a **transition table**. Assign **binary codes** to the states in the state table, if you haven't already. If you have N states, your binary codes will have at least $\lceil \log_2 N \rceil$ digits, and your circuit will have at least $\lceil \log_2 N \rceil$ flip-flops

Step 3:
Convert the **transition table** into an **excitation table**. For each flip-flop and each row of your state table, find the flip-flop input values that are needed to generate the next state from the present state. You can use flip-flop excitation tables here.

Step 4:
Derive **excitation equations**. Find simplified equations for the **flip-flop inputs** and the **outputs**.

Step 5:
Build the circuit!

2

Flip-Flop Excitation Tables

- A table that lists the required inputs for a given change of state is known as an *excitation table.*

SR flip-flop

Q(t)	Q(t + 1)	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

D flip-flop

Q(t)	Q(t + 1)	D
0	0	0
0	1	1
1	0	0
1	1	1

JK flip-flop

Q(t)	Q(t + 1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

T flip-flop

Q(t)	Q(t + 1)	T
0	0	0
0	1	1
1	0	1
1	1	0

3

Sequence Detector

- A *sequence recognizer* or *detector* is a sequential circuit that looks for a special bit pattern occurring in a binary string
- The recognizer circuit has only one input, X
  - One input bit is supplied on every clock cycle
- For every bit applied to the input X, the circuit generates a bit at output Z.
  - Z = 1 when the most recent bits in the stream at input X completes the pattern, otherwise the output Z=0
- You are required to design a *sequence recognizer* that detects the bit pattern "1001"

Inputs: 11 1 00 1 1 0 1 0 0 1 00 1 10…
Outputs: 00000 1 00000 1 00 1 00…

- The *sequence recognizer* is implemented by a sequential circuit because the circuit has to "remember" the inputs from previous clock cycles, in order to determine whether or not a match was found

1. Design the above *sequence recognizer* using JK flip-flops
2. Redo the design with D, T and SR flip-flops

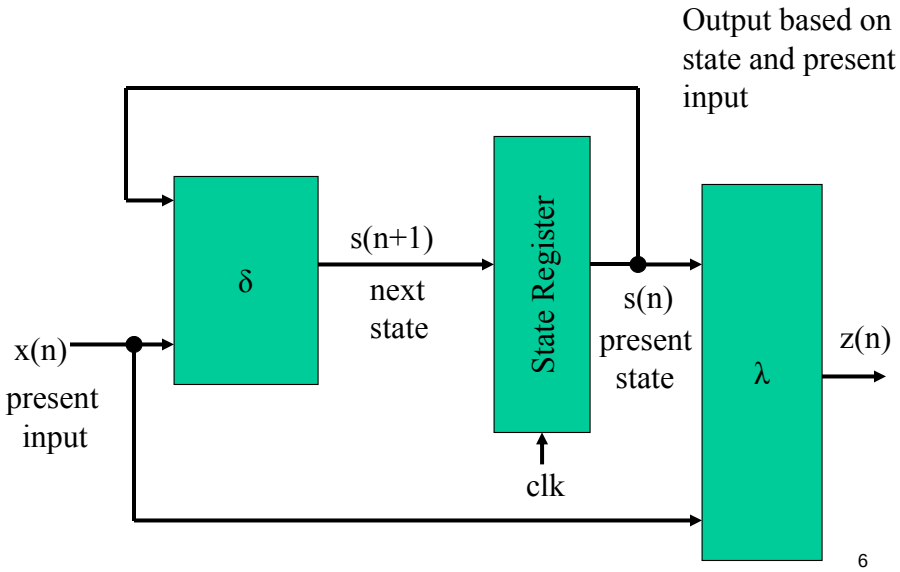
4

Step 1: Building a state table

- The first thing you have to figure out is precisely how the use of state will help you solve the given problem
  - Build the state table based on the problem statement. The table should show the present states, inputs, next states and outputs
  - Sometimes it is easier to first find a state diagram and then convert that to a table
- This is usually the most difficult step. Once you have the state table, the rest of the design procedure is the same for all sequential circuits

5

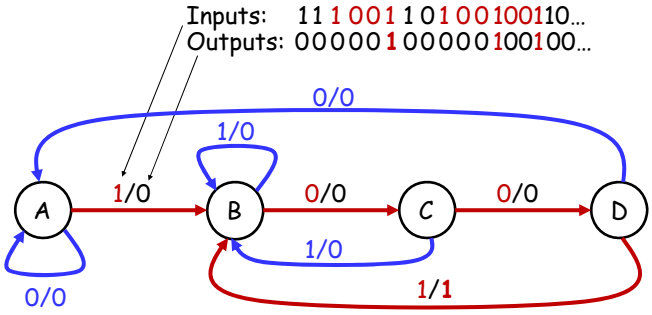
Mealy Machine



6

Step1.1 Building Mealy *state diagram*

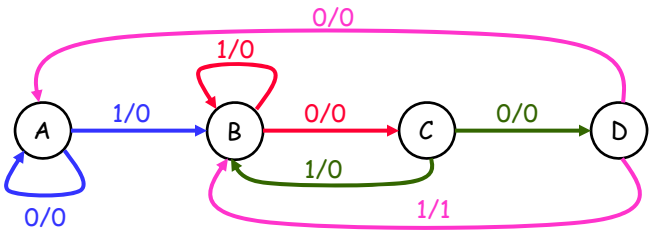
Example: design a sequential circuit that can detect the bit pattern "1001":



State	Meaning
A	None of the desired pattern (1001) has been input yet.
B	We've already seen the first bit (1) of the desired pattern.
C	We've already seen the first two bits (10) of the desired pattern.
D	We've already seen the first three bits (100) of the desired pattern.

7

Step1.2 Derive the *state table* from the *state diagram*



State: S={A,B,C,D}

Input: X={1,0}

Output: Z={1,0}

Initial State: A

Transitions: $S^{n+1}=\delta(X^n,S^n)$

$Z^n=\lambda(X^n,S^n)$

S^n	X^n	$S^{n+1}=\delta(X^n,S^n)$	$Z^n=\lambda(X^n,S^n)$
Present State	Input	Next State	Output
A	0	A	0
A	1	B	0
B	0	C	0
B	1	B	0
C	0	D	0
C	1	B	0
D	0	A	0
D	1	B	1

8

Step 2: Assigning binary codes to states

- We have four states ABCD, so we need at least two flip-flops Q_1Q_0
- The easiest thing to do is represent state A with Q_1Q_0 = 00, B with 01, C with 10, and D with 11. Other courses will show you how to assign codes to the FSM states.

STATE TABLE

Present State S^n	Input X^n	Next State $S^{n+1}=\delta(X^n,S^n)$	Output $Z^n=\lambda(X^n,S^n)$
A	0	A	0
A	1	B	0
B	0	C	0
B	1	B	0
C	0	D	0
C	1	B	0
D	0	A	0
D	1	B	1

TRANSITION TABLE

Present State Q_1	Q_0	Input X	δ Next State Q_1^+	Q_0^+	λ Output Z
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	0	1	0
1	1	0	0	0	0
1	1	1	0	1	1

9

Step 3: Finding JK flip-flop input values

Interpret the excitation table of the JK flip-flop as follows:

Q(n)	Q(n+1)	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

1. If Current State is 0 → J= Q(n+1) & K = x, i.e., **copy the next state to the J** column of the corresponding flip-flop
2. If Current State is 1 → J= x & K= $\overline{Q(n+1)}$, i.e., **copy the complement of the next state to the K** column of the corresponding flip-flop

Use the above 2 rules to derive efficiently the J K columns for Q_1 and Q_0

Present State (n) Q_1	Q_0	Input X	Next State (n+1) Q_1^+	Q_0^+	Flip flop inputs J_1	K_1	J_0	K_0	Output Z
0	0	0	0	0	0	x	0	x	0
0	0	1	0	1	0	x	1	x	0
0	1	0	1	0	1	x	x	1	0
0	1	1	0	1	0	x	x	0	0
1	0	0	1	1	x	0	1	x	0
1	0	1	0	1	x	1	1	x	0
1	1	0	0	0	x	1	x	1	0
1	1	1	0	1	x	1	x	0	1

10

Step 4: Find equations for the JK-FF inputs (1)

Present State (n) Q_1	Q_0	Input X	Next State $(n+1)$ Q_1^+	Q_0^+	Flip flop inputs J_1	K_1	J_0	K_0	Output Z
0	0	0	0	0	0	x	0	x	0
0	0	1	0	1	0	x	1	x	0
0	1	0	1	0	1	x	x	1	0
0	1	1	0	1	0	x	x	0	0
1	0	0	1	1	x	0	1	x	0
1	0	1	0	1	x	1	1	x	0
1	1	0	0	0	x	1	x	1	0
1	1	1	0	1	x	1	x	0	1

J_1

X \ $Q_1 Q_0$	00	01	11	10
0	0	1	x	x
1	0	0	x	x

$J_1 = X' Q_0$

K_1

X \ $Q_1 Q_0$	00	01	11	10
0	x	x	1	0
1	x	x	1	1

$K_1 = X + Q_0$

Step 4: Find equations for the JK-FF inputs (2)

Present State (n) Q_1	Q_0	Input X	Next State $(n+1)$ Q_1^+	Q_0^+	Flip flop inputs J_1	K_1	J_0	K_0	Output Z
0	0	0	0	0	0	x	0	x	0
0	0	1	0	1	0	x	1	x	0
0	1	0	1	0	1	x	x	1	0
0	1	1	0	1	0	x	x	0	0
1	0	0	1	1	x	0	1	x	0
1	0	1	0	1	x	1	1	x	0
1	1	0	0	0	x	1	x	1	0
1	1	1	0	1	x	1	x	0	1

J_0

X \ $Q_1 Q_0$	00	01	11	10
0	0	x	x	1
1	1	x	x	1

$J_0 = X + Q_1$

K_0

X \ $Q_1 Q_0$	00	01	11	10
0	x	1	1	x
1	x	0	0	x

$K_0 = X'$

Step 4: Find equations of the Sequential Circuit output

Present State (n) Q_1	Q_0	Input X	Next State $(n+1)$ Q_1^+	Q_0^+	Flip flop inputs J_1	K_1	J_0	K_0	Output Z
0	0	0	0	0	0	x	0	x	0
0	0	1	0	1	0	x	1	x	0
0	1	0	1	0	1	x	x	1	0
0	1	1	0	1	0	x	x	0	0
1	0	0	1	1	x	0	1	x	0
1	0	1	0	1	x	1	1	x	0
1	1	0	0	0	x	1	x	1	0
1	1	1	0	1	x	1	x	0	1

Z

X \ $Q_1 Q_0$	00	01	11	10
0				
1			1	

$Z = X Q_1 Q_0$

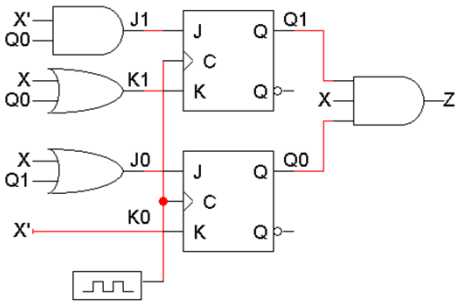
13

Step 5: Build the sequential circuit

$J_1 = X' Q_0$
$K_1 = X + Q_0$

$J_0 = X + Q_1$
$K_0 = X'$

$Z = Q_1 Q_0 X$



14

Building the same circuit with D flip-flops (Step 3)

- $D = Q^+$

i.e., copy the D-FF next state (Q^+) to the D column

excitation table

Q(n) Q	Q(n+1) Q^+	D
0	0	0
0	1	1
1	0	0
1	1	1

Present State$^{(n)}$ Q_1	Q_0	Input X	Next State$^{(n+1)}$ Q_1^+	Q_0^+	Flip flop inputs D_1	D_0	Output Z
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	0	1	0	0
0	1	1	0	1	0	1	0
1	0	0	1	1	1	1	0
1	0	1	0	1	0	1	0
1	1	0	0	0	0	0	0
1	1	1	0	1	0	1	1

15

Finding D-FF input equations (Step 4)

Present State$^{(n)}$ Q_1	Q_0	Input X	Next State$^{(n+1)}$ Q_1^+	Q_0^+	Flip flop inputs D_1	D_0	Output Z
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	0	1	0	0
0	1	1	0	1	0	1	0
1	0	0	1	1	1	1	0
1	0	1	0	1	0	1	0
1	1	0	0	0	0	0	0
1	1	1	0	1	0	1	1

D_1

X \ $Q_1 Q_0$	00	01	11	10
0		1		1
1				

$D_1 = Q_1 Q_0' X' + Q_1' Q_0 X'$

D_0

X \ $Q_1 Q_0$	00	01	11	10
0				1
1	1	1	1	1

$D_0 = X + Q_1 Q_0'$

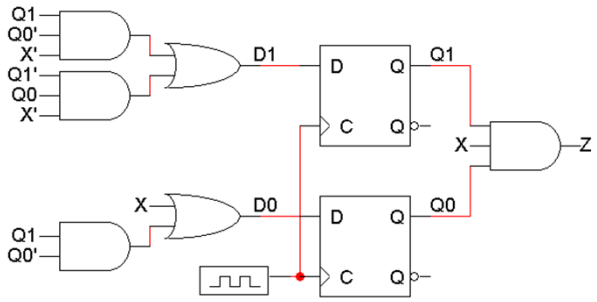
Z

X \ $Q_1 Q_0$	00	01	11	10
0				
1			1	

$Z = X Q_1 Q_0$

16

Building the circuit (Step 5)



17

Building the same circuit with T flip-flops (Step 3)

T-FF characteristic equation

- $Q^+ = \overline{T}.Q + T.\overline{Q}$

excitation table

Q(n) Q	Q(n+1) Q^+	T
0	0	0
0	1	1
1	0	1
1	1	0

Use the excitation table of the T flip-flop as follows:
copy to the T column of the coressponding T-FF

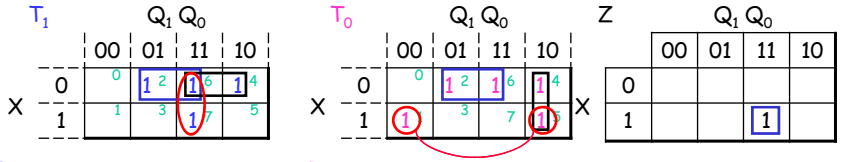
- the T-FF next state (Q^+) if Q = 0
- the complement of the T-FF next state ($\overline{Q}+$) if Q = 1

Present State$^{(n)}$ Q_1	Q_0	Input X	Next State$^{(n+1)}$ Q_1^+	Q_0	Flip flop inputs T_1	T_0	Output Z
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	0	1	1	0
0	1	1	0	1	0	0	0
1	0	0	1	1	0	1	0
1	0	1	0	1	1	1	0
1	1	0	0	0	1	1	0
1	1	1	0	1	1	0	1

18

Finding T-FF input equations (Step 4)

	Present State $Q_1^{(n)}$	Q_0	Input X	Next State $Q_1^{+(n+1)}$	Q_0^+	Flip flop inputs T_1	T_0	Output Z
0	0	0	0	0	0	0	0	0
1	0	0	1	0	1	0	1	0
2	0	1	0	1	0	1	1	0
3	0	1	1	0	1	0	0	0
4	1	0	0	1	1	0	1	0
5	1	0	1	0	1	1	1	0
6	1	1	0	0	0	1	1	0
7	1	1	1	0	1	1	0	1



$T_1 = Q_0 X' + Q_1 X' + Q_1 Q_0$ $T_0 = Q_0 X' + Q_1 Q_0 + X Q_0'$ $Z = X Q_1 Q_0$

19

Building the same circuit with SR flip-flops

Step 3: Finding flip-flop input values

Note:

1. For Current State Q(n) = 0 → S = Q(n+1) i.e., copy the next state to the S column of the corresponding flip-flop
2. If Current State Q(n) = 1 → R = $\overline{Q(n+1)}$, i.e., copy the complement of the next state to the R column of the corresponding flip-flop

Q(n)	Q(n+1)	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

Present State $Q_1^{(n)}$	Q_0	Input X	Next State $Q_1^{+(n+1)}$	Q_0^+	Flip flop inputs S_1	R_1	S_0	R_0	Output Z
0	0	0	0	0	0	x	0	x	0
0	0	1	0	1	0	x	1	0	0
0	1	0	1	0	1	0	0	1	0
0	1	1	0	1	0	x	x	0	0
1	0	0	1	1	x	0	1	0	0
1	0	1	0	1	0	1	1	0	0
1	1	0	0	0	0	1	0	1	0
1	1	1	0	1	0	1	x	0	1

20

Step 4: Find equations for the SR-FF inputs (1)

Present State (n) Q_1	Q_0	Input X	Next State $(n+1)$ Q_1^+	Q_0^+	Flip flop inputs S_1	R_1	S_0	R_0	Output Z
0	0	0	0	0	0	x	0	x	0
0	0	1	0	1	0	x	1	0	0
0	1	0	1	0	1	0	0	1	0
0	1	1	0	1	0	x	x	0	0
1	0	0	1	1	x	0	1	0	0
1	0	1	0	1	0	1	1	0	0
1	1	0	0	0	0	1	0	1	0
1	1	1	0	1	0	1	x	0	1

S_1

X \ $Q_1 Q_0$	00	01	11	10
0	0	1	0	x
1	0	0	0	0

$S_1 = X' Q_1' Q_0$

R_1

X \ $Q_1 Q_0$	00	01	11	10
0	x	0	1	0
1	x	x	1	1

$R_1 = X + Q_1 Q_0$

Step 4: Find equations for the SR-FF inputs (2)

Present State (n) Q_1	Q_0	Input X	Next State $(n+1)$ Q_1^+	Q_0^+	Flip flop inputs S_1	R_1	S_0	R_0	Output Z
0	0	0	0	0	0	x	0	x	0
0	0	1	0	1	0	x	1	0	0
0	1	0	1	0	1	0	0	1	0
0	1	1	0	1	0	x	x	0	0
1	0	0	1	1	x	0	1	0	0
1	0	1	0	1	0	1	1	0	0
1	1	0	0	0	0	1	0	1	0
1	1	1	0	1	0	1	x	0	1

S_0

X \ $Q_1 Q_0$	00	01	11	10
0	0	0	0	1
1	1	x	x	1

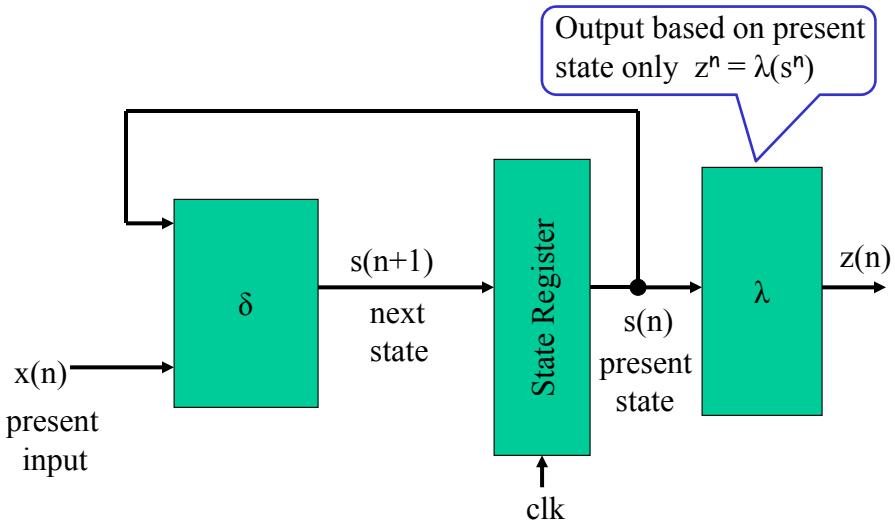
$S_0 = X + Q_0' Q_1$

R_0

X \ $Q_1 Q_0$	00	01	11	10
0	x	1	1	0
1	0	0	0	0

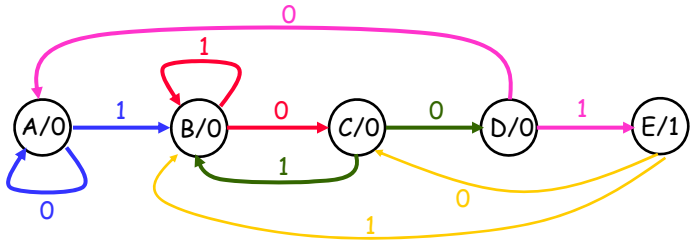
$R_0 = Q_0 X'$

Moore Machine



23

Moore state diagram & table



Present State	Input	Next State	Output
A	0	A	0
A	1	B	0
B	0	C	0
B	1	B	0
C	0	D	0
C	1	B	0
D	0	A	0
D	1	E	0
E	0	C	1
E	1	B	1

A: 000 D: 100
B: 001 E: 101
C: 010

Z

Q_0 \ $Q_2 Q_1$	00	01	11	10
0				
1				1

$Z = Q_2 Q_1' Q_0$

24