Making Finite State Machines Simpler

~~~~~

FSM State Reduction
~~~~~

Review of Objectives

After this lecture, you should be able to…

- **Declare the 2 conditions necessary for states in an FSM to be EQUIVALENT**
- **Use a state reduction to lower the number of required FSM states so that the design flow takes the form shown:-**
- STEPS IN REDUCED FSM DESIGN
  - **Draw a state transition diagram**
  - **Reduce number of states if possible**
  - **Derive a state table from the state transition diagram**
  - **Assign states, determine number, type of FF's required**
  - **Draw up a truth table**
  - **Design logic for combinational logic unit**
  - **Draw the complete schematic**
- **Write D, RS, T and JK FF excitation tables**
- **Employ FF excitation tables to use any kind of FF in an FSM designs**

Definitions for FSM State Reduction

- Condition #1 : The outputs associated with 2 states are the SAME

- Condition #2 : Corresponding Next States are the SAME or EQUIVALENT

- With these 2 conditions satisfied, we can COMBINE the 2 states into 1 newly named state

State Reduction

- **Description of state machine M_0 to be reduced.**

Input Present state	*X*	
	0	1
A	C/0	E/1
B	D/0	E/1
C	C/1	B/0
D	C/1	A/0
E	A/0	C/1

© 1999 M. Murdocca and V. Heuring

State Reduction – Step 1

- **Look for output compatible transitions: Note that {A,B and E} and {C,D} transition to OUTPUT compatible states**

Input Present state	X	
	0	1
A	C/0	E/1
B	D/0	E/1
C	C/1	B/0
D	C/1	A/0
E	A/0	C/1

© 1999 M. Murdocca and V. Heuring

State Reduction – Step 2

- Take each collection of states and examine each pair within the set for possibly equivalent states with the different inputs X=0 and X=1

Possibly Equivalent Pairs	Next State pairing with X=0 [output incompatible] = [!]	Next State pairing with X=1 [output incompatible] = [!]
(A,B)	(C,D)	
(A,E)	(A,C) [!]	(C,E) [!]
(B,E)	(A,D) [!]	(C,E) [!]

Possibly Equivalent Pairs	Next State pairing with X=0 [output incompatible] = [!]	Next State pairing with X=1 [output incompatible] = [!]
(C,D)		(A,B)

© 1999 M. Murdocca and V. Heuring

State Reduction – Step 3

- Eliminate pairs that are not output compatible:

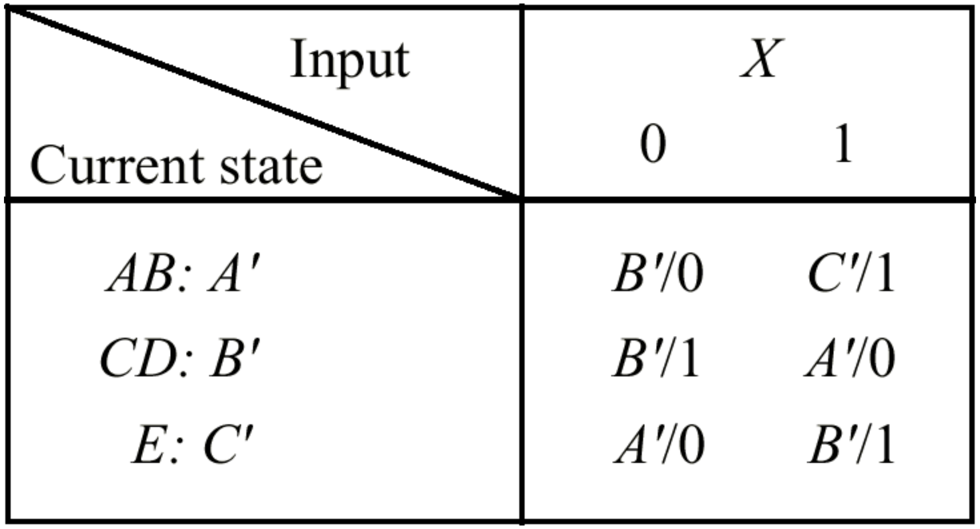
- We see that E is not compatible with A or B or by implication with (C,D) which are already seen to be output incompatible. So E cannot be eliminated nor combined with other states.

  So we have pairs {A,B}, {C,D} and {E} which represent 3 states which allow the FSM definition to remain unchanged.

Reduced State Table – Step 4

- We have {A,B}, {C,D} and {E} as the new set of states, which we will rename as A’ B’ and C’ for the reduced FSM’s internal states.

- We now write the “reduced state” state table for machine M_1.



Input / Current state	X	
	0	1
$AB: A'$	$B'/0$	$C'/1$
$CD: B'$	$B'/1$	$A'/0$
$E: C'$	$A'/0$	$B'/1$

© 1999 M. Murdocca and V. Heuring

The State Assignment Problem

- **Two state assignments for machine M_2.**

Input	X	
P.S.	0	1
A	*B*/1	*A*/1
B	*C*/0	*D*/1
C	*C*/0	*D*/0
D	*B*/1	*A*/0

Machine M_2

Input	X	
S_0S_1	0	1
A: 00	01/1	00/1
B: 01	10/0	11/1
C: 10	10/0	11/0
D: 11	01/1	00/0

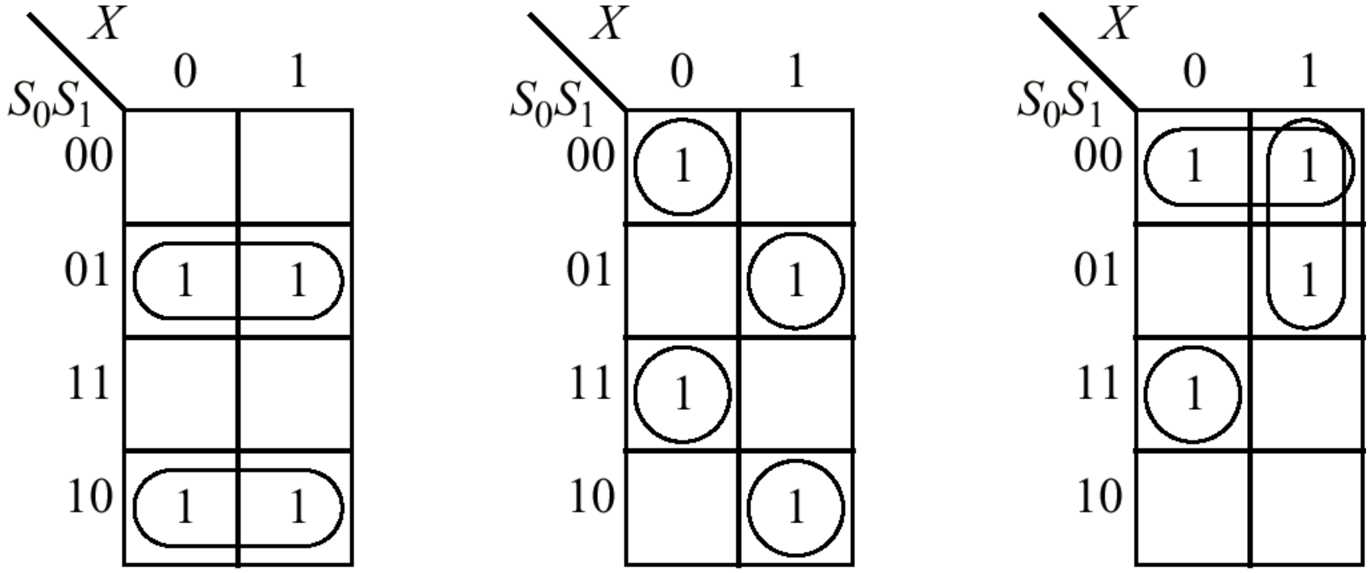
State assignment SA_0

Input	X	
S_0S_1	0	1
A: 00	01/1	00/1
B: 01	11/0	10/1
C: 11	11/0	10/0
D: 10	01/1	00/0

State assignment SA_1

State Assignment SA$_0$

- Boolean equations for machine M_2 using state assignment SA$_0$.



$$S_0 = \overline{S_0}S_1 + S_0\overline{S_1}$$

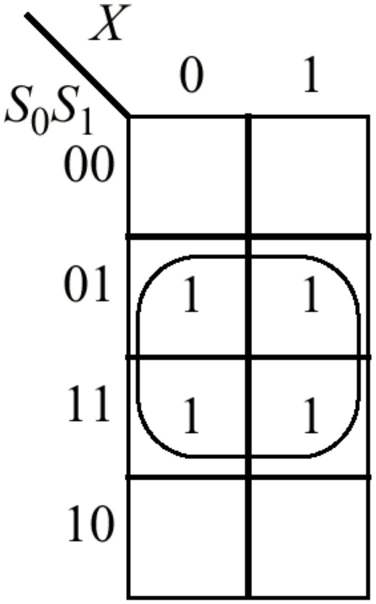
$$S_1 = \overline{S_0}\,\overline{S_1}\bar{X} + \overline{S_0}S_1X + S_0S_1\bar{X} + S_0\overline{S_1}X$$

$$Z = \overline{S_0}\,\overline{S_1} + \overline{S_0}X + S_0S_1\bar{X}$$

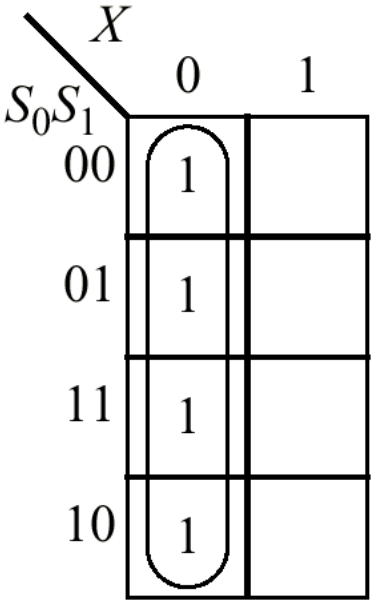
© 1999 M. Murdocca and V. Heuring

State Assignment SA$_1$

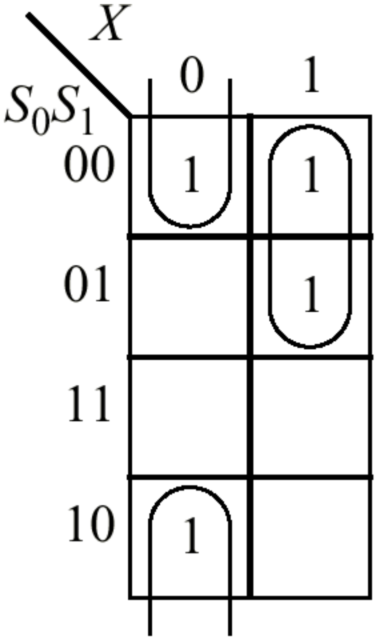
- **Boolean equations for machine *M*$_2$ using state assignment SA$_1$.**



$S_0 = S_1$



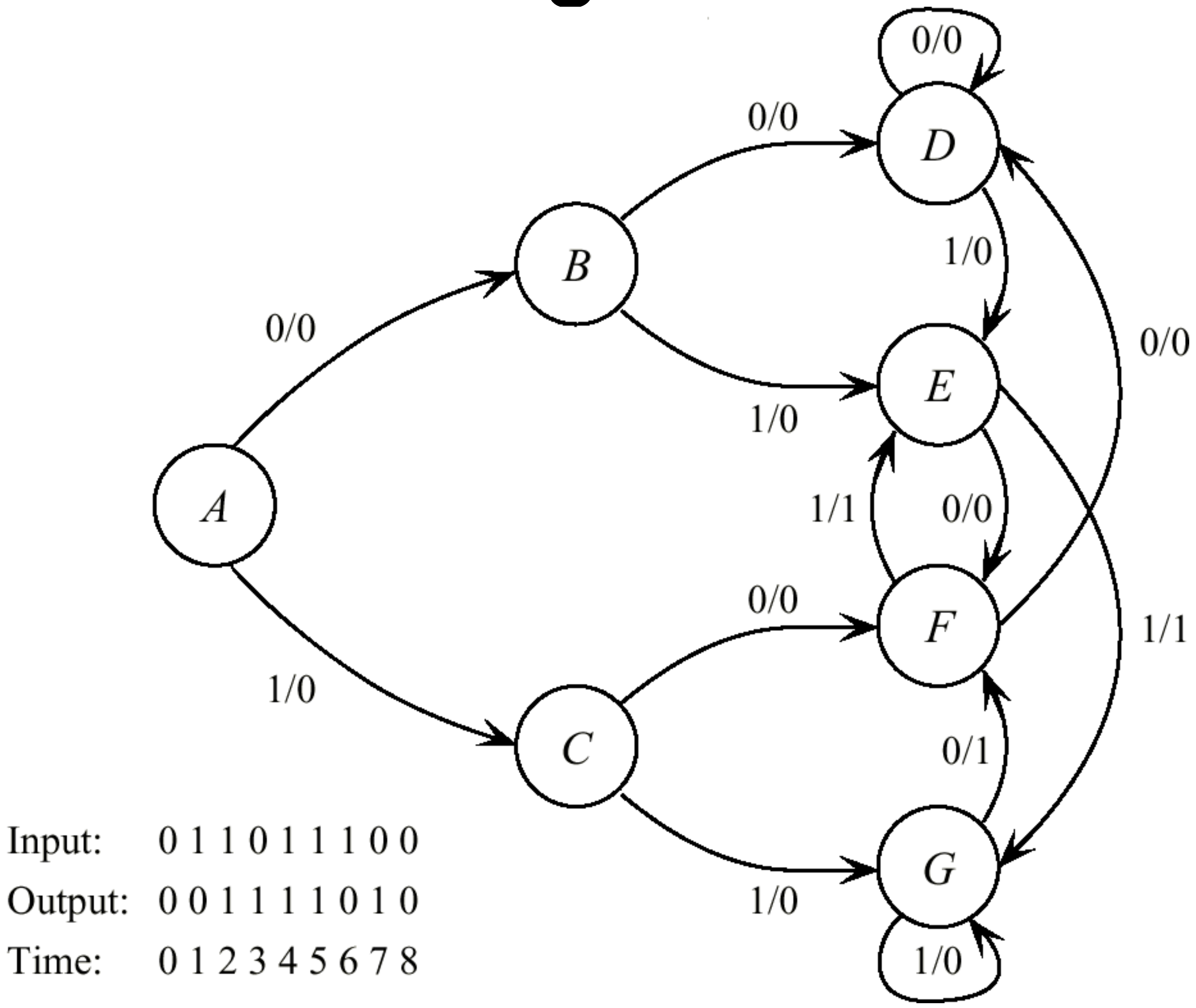
$S_1 = \bar{X}$



$Z = \overline{S_1}\bar{X} + \overline{S_0}X$

© 1999 M. Murdocca and V. Heuring

Sequence Detector State Transition Diagram



Input:	0 1 1 0 1 1 1 0 0
Output:	0 0 1 1 1 1 0 1 0
Time:	0 1 2 3 4 5 6 7 8

© 1999 M. Murdocca and V. Heuring

Sequence Detector State Table

Input / Present state	*X*	
	0	1
A	*B*/0	*C*/0
B	*D*/0	*E*/0
C	*F*/0	*G*/0
D	*D*/0	*E*/0
E	*F*/0	*G*/1
F	*D*/0	*E*/1
G	*F*/1	*G*/0

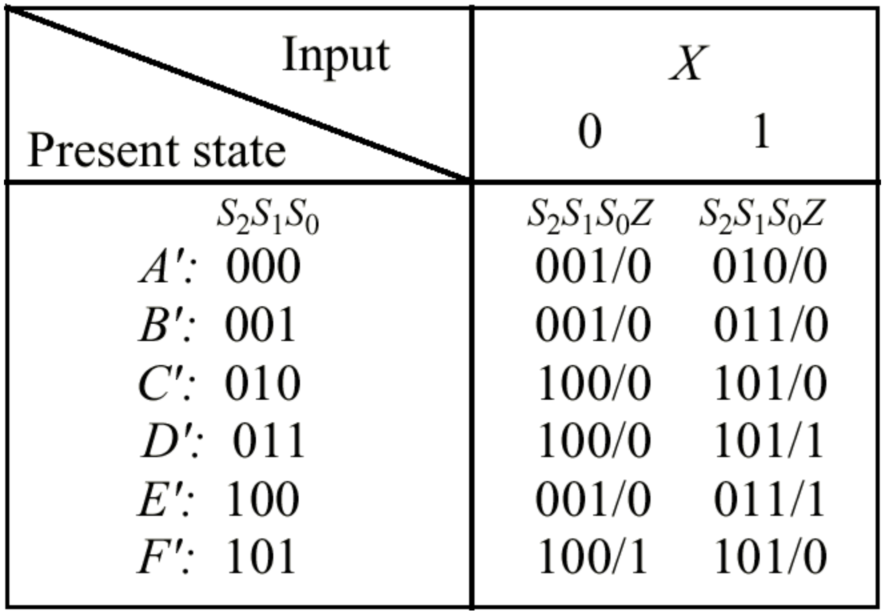
© 1999 M. Murdocca and V. Heuring

Sequence Detector Reduced State Table

Input Present state	X	
	0	1
A: A'	$B'/0$	$C'/0$
BD: B'	$B'/0$	$D'/0$
C: C'	$E'/0$	$F'/0$
E: D'	$E'/0$	$F'/1$
F: E'	$B'/0$	$D'/1$
G: F'	$E'/1$	$F'/0$

© 1999 M. Murdocca and V. Heuring

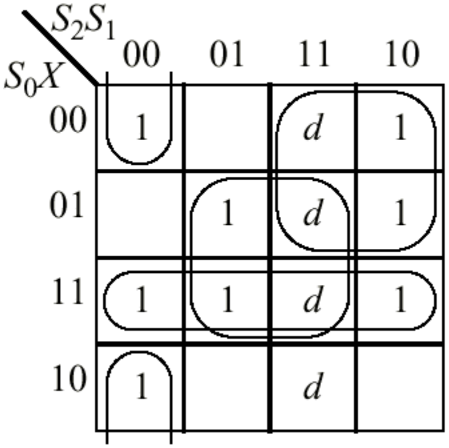
Sequence Detector State Assignment



Input Present state	X	
	0	1
$S_2S_1S_0$	$S_2S_1S_0Z$	$S_2S_1S_0Z$
A': 000	001/0	010/0
B': 001	001/0	011/0
C': 010	100/0	101/0
D': 011	100/0	101/1
E': 100	001/0	011/1
F': 101	100/1	101/0

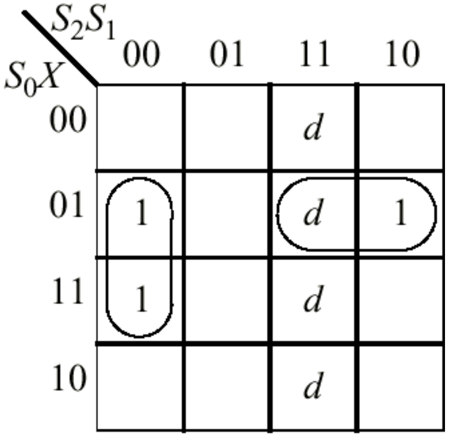
Sequence Detector K-Maps

- **K-map reduction of next state and output functions for sequence detector.**



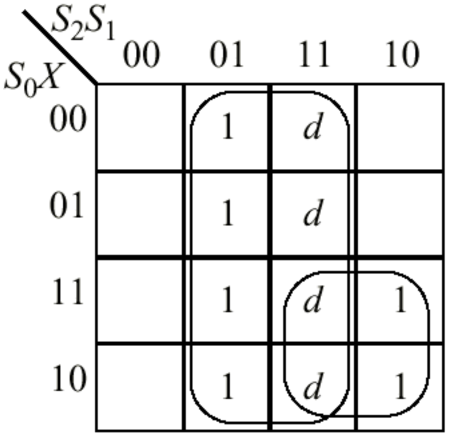
S_0X \ S_2S_1	00	01	11	10
00	1		d	1
01		1	d	1
11	1	1	d	1
10	1		d	

$$S_0 = \overline{S_2}\,\overline{S_1}\,\overline{X} + S_0X + S_2\overline{S_0} + S_1X$$



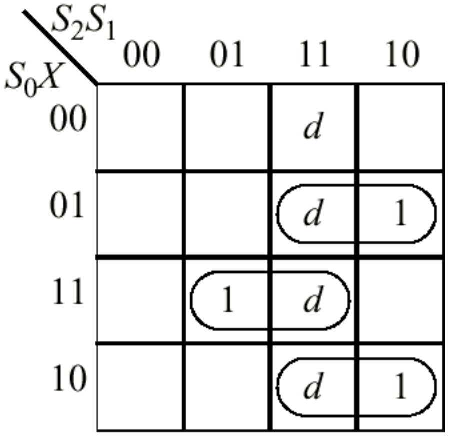
S_0X \ S_2S_1	00	01	11	10
00			d	
01	1		d	1
11	1		d	
10			d	

$$S_1 = \overline{S_2}\,\overline{S_1}X + S_2\overline{S_0}X$$



S_0X \ S_2S_1	00	01	11	10
00		1	d	
01		1	d	
11		1	d	1
10		1	d	1

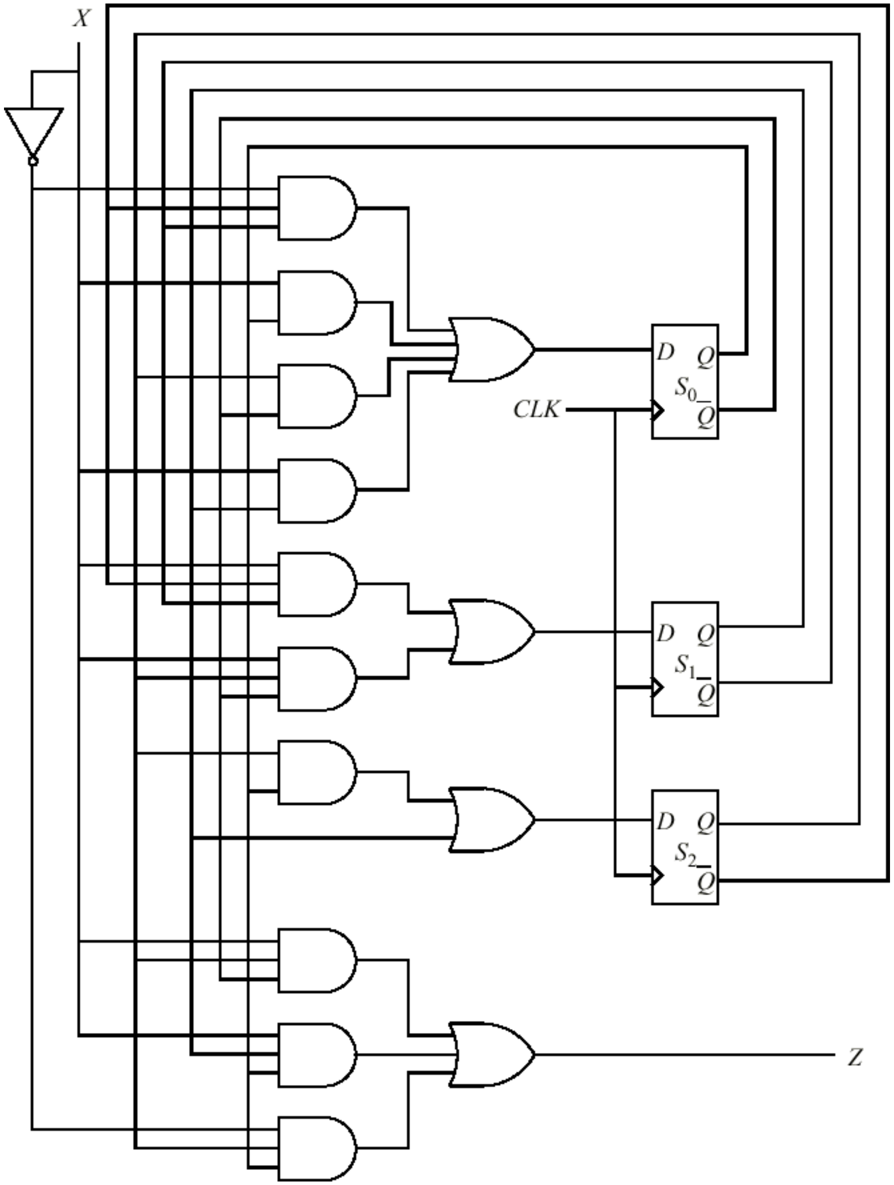
$$S_2 = S_2S_0 + S_1$$



S_0X \ S_2S_1	00	01	11	10
00			d	
01			d	1
11		1	d	
10			d	1

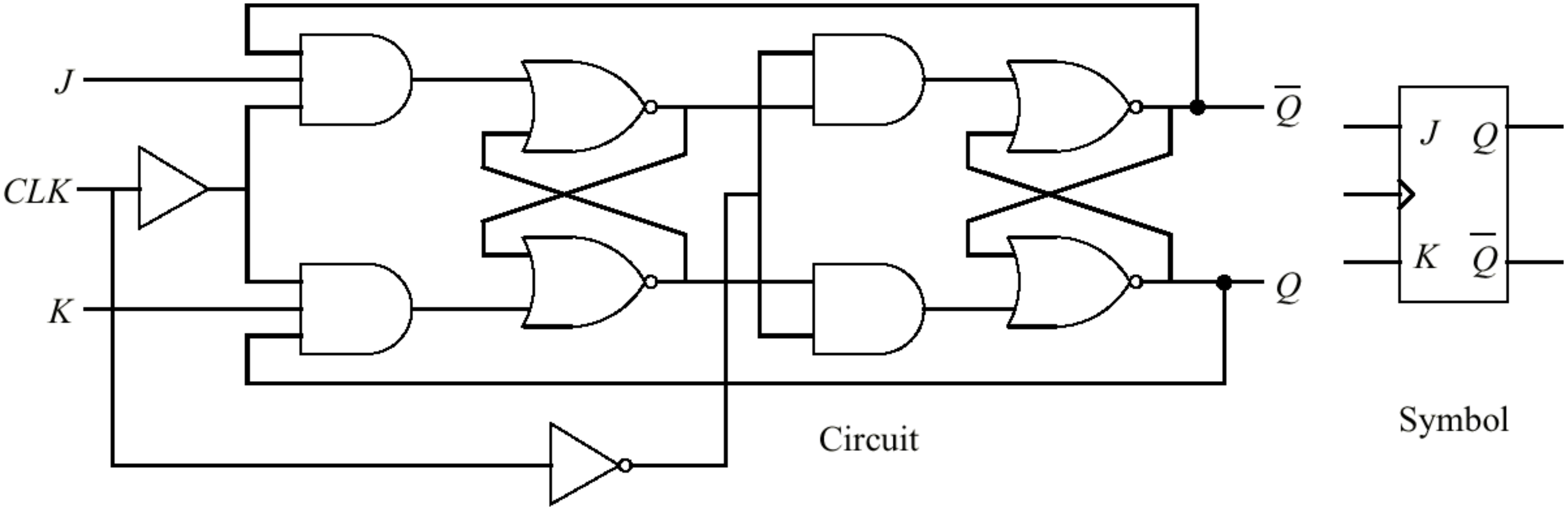
$$Z = S_2\overline{S_0}X + S_1S_0X + S_2S_0\overline{X}$$

Sequence Detector Circuit



Excitation Tables

- In addition to the D flip-flop, the S-R, J-K, and T flip-flops are used as delay elements in finite state machines.
- A Master-Slave J-K flip-flop is shown below.



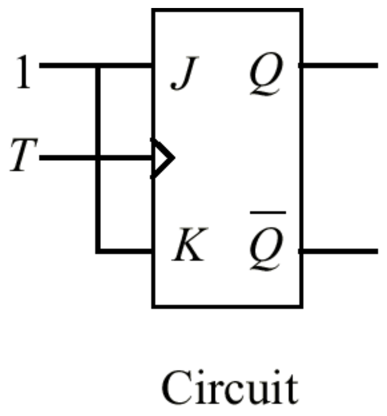
Circuit

Symbol

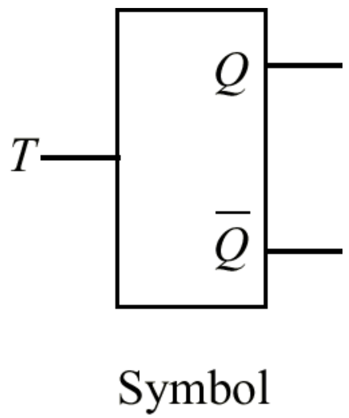
© 1999 M. Murdocca and V. Heuring

Clocked T Flip-Flop

- **Logic diagram and symbol for a T flip-flop.**



Circuit



Symbol

© 1999 M. Murdocca and V. Heuring

Excitation Tables

- **Each table shows the settings that must be applied at the inputs at time *t* in order to change the outputs at time *t*+1.**

S-R flip-flop

Q_t	Q_{t+1}	S	R
0	0	0	0
0	1	1	0
1	0	0	1
1	1	0	0

D flip-flop

Q_t	Q_{t+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

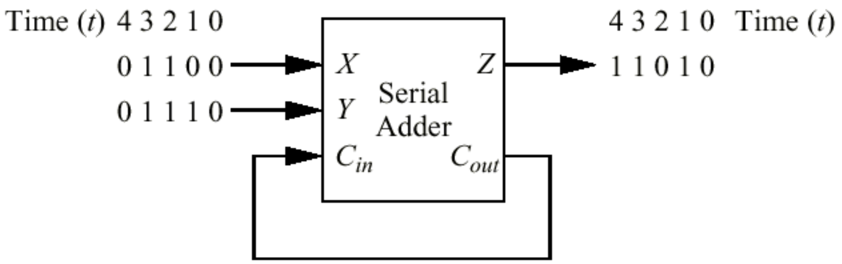
J-K flip-flop

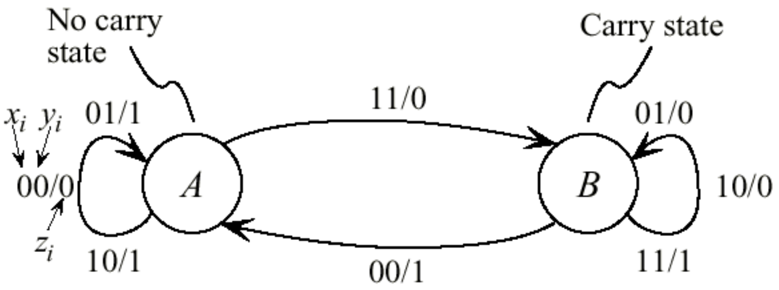
Q_t	Q_{t+1}	J	K
0	0	0	*d*
0	1	1	*d*
1	0	*d*	1
1	1	*d*	0

T flip-flop

Q_t	Q_{t+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Serial Adder





- **State transition diagram, state table, and state assignment for a serial adder.**

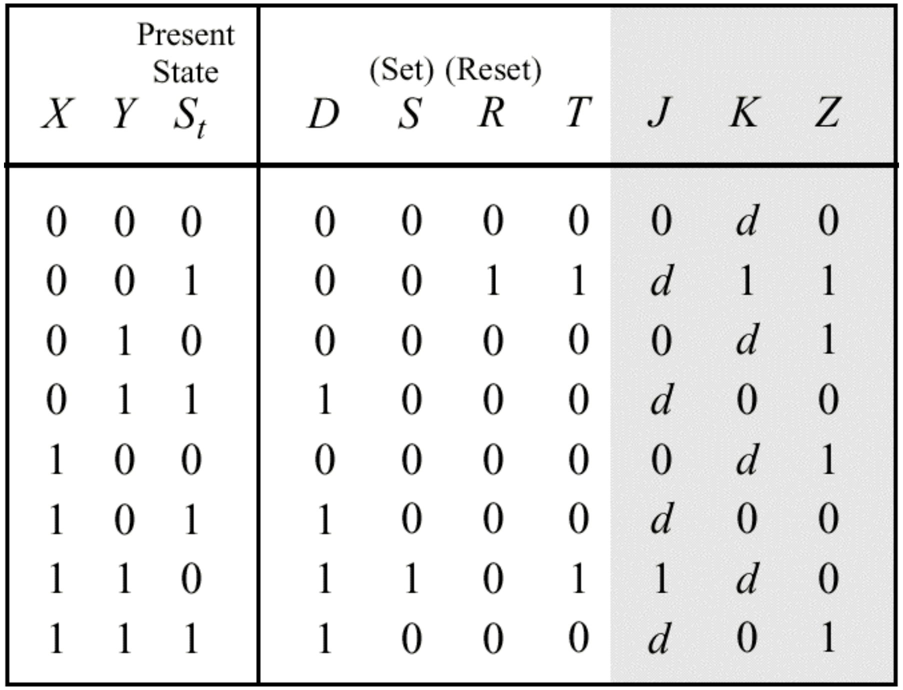
Present state \ Input	XY			
	00	01	10	11
A	A/0	A/1	A/1	B/0
B	A/1	B/0	B/0	B/1

Next state Output

Present state (S_t) \ Input	XY			
	00	01	10	11
A:0	0/0	0/1	0/1	1/0
B:1	0/1	1/0	1/0	1/1

Serial Adder Next-State Functions

- **Truth table showing next-state functions for a serial adder for D, S-R, T, and J-K flip-flops. Shaded functions are used in the example.**



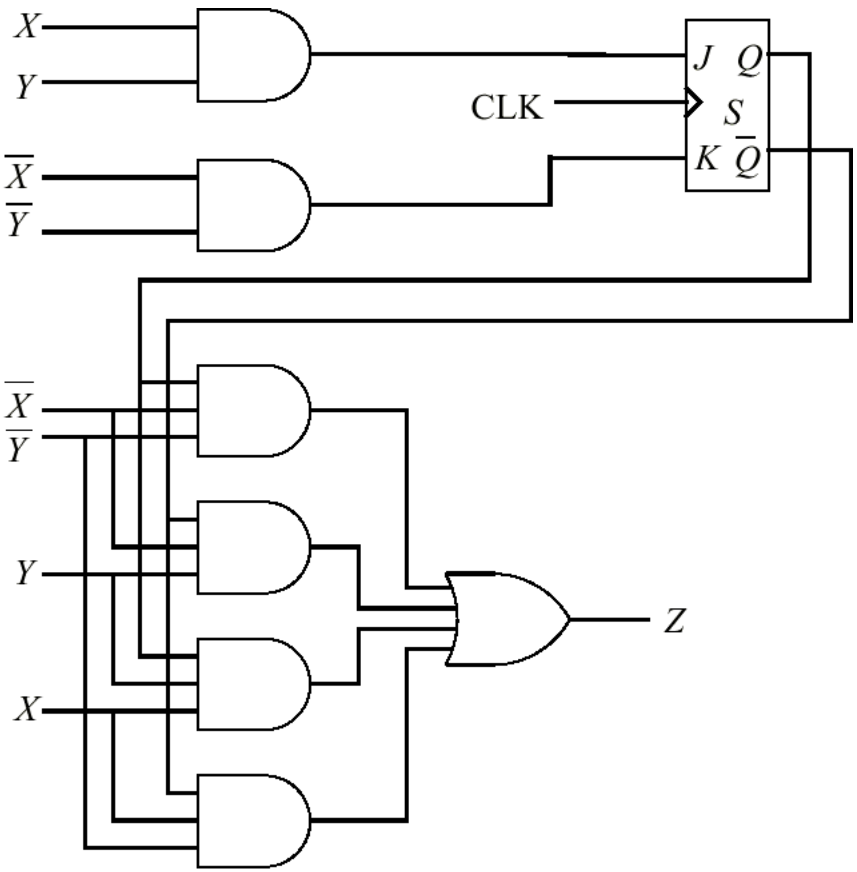
		Present State		(Set)	(Reset)				
X	Y	S_t	D	S	R	T	J	K	Z
0	0	0	0	0	0	0	0	d	0
0	0	1	0	0	1	1	d	1	1
0	1	0	0	0	0	0	0	d	1
0	1	1	1	0	0	0	d	0	0
1	0	0	0	0	0	0	0	d	1
1	0	1	1	0	0	0	d	0	0
1	1	0	1	1	0	1	1	d	0
1	1	1	1	0	0	0	d	0	1

J-K Flip-Flop Serial Adder Circuit

$$J = XY$$

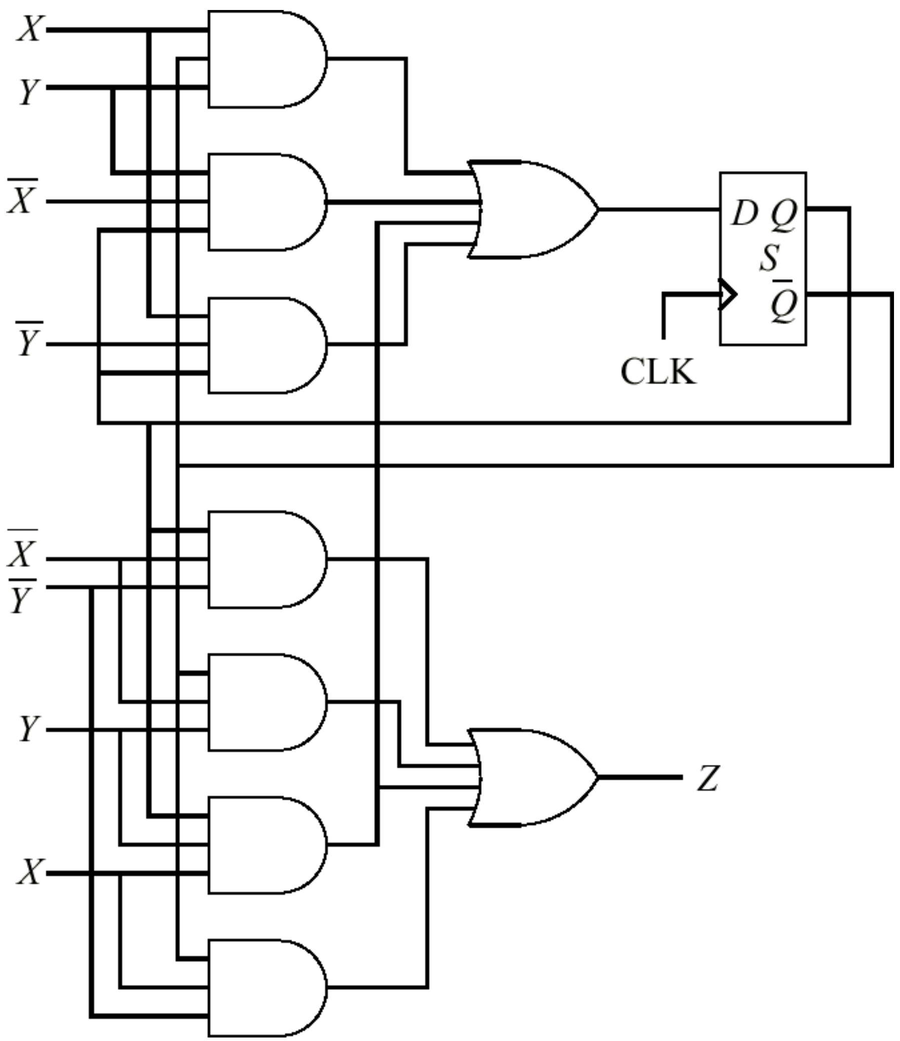
$$K = \overline{X}\,\overline{Y}$$

$$Z = \overline{X}\,\overline{Y}S + \overline{X}Y\overline{S} + XYS + X\overline{Y}\,\overline{S}$$



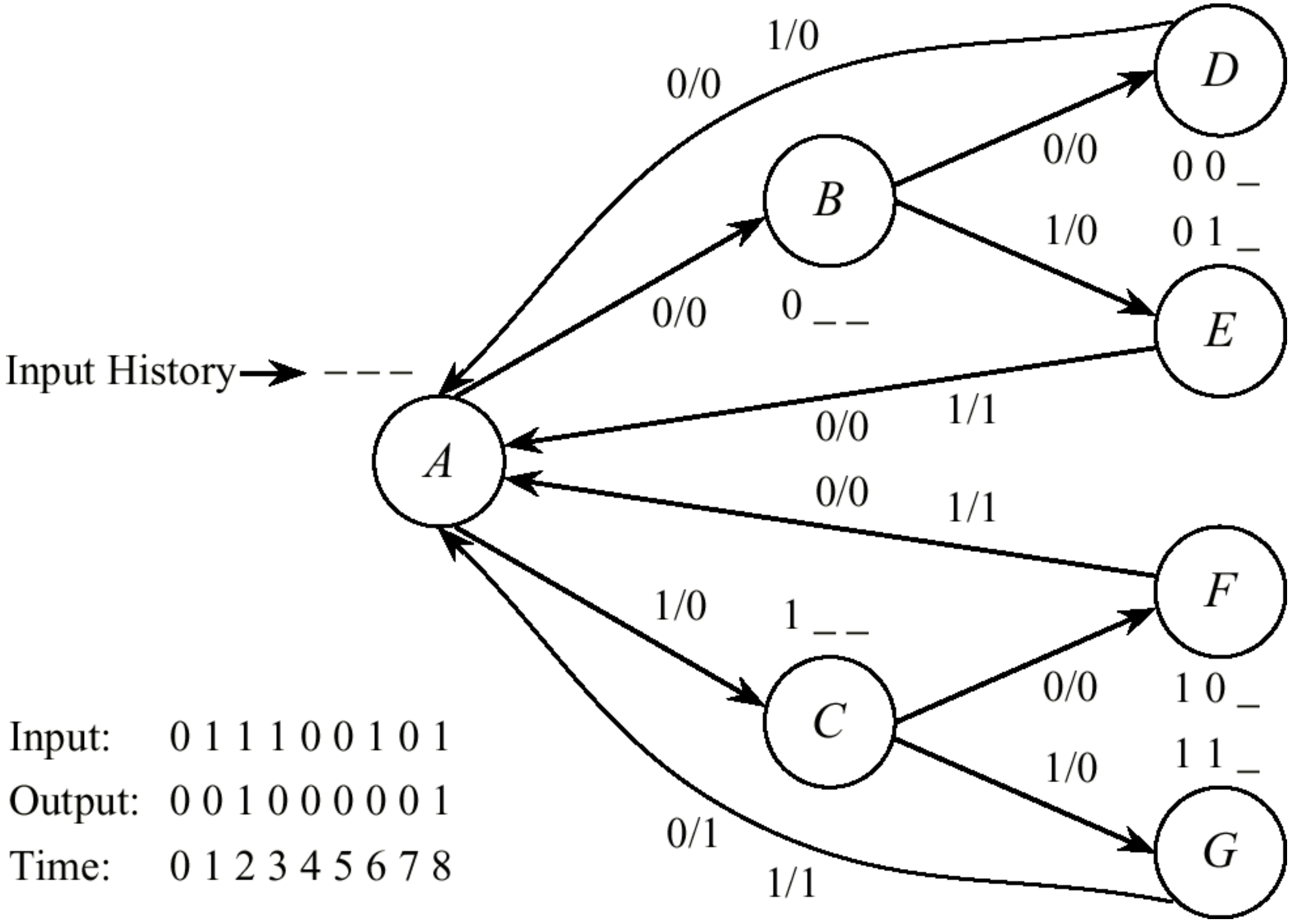
© 1999 M. Murdocca and V. Heuring

D Flip-Flop Serial Adder Circuit



© 1999 M. Murdocca and V. Heuring

Majority Finite State Machine



Input:	0	1	1	1	0	0	1	0	1
Output:	0	0	1	0	0	0	0	0	1
Time:	0	1	2	3	4	5	6	7	8

Majority FSM State Table

- **(a) State table for majority FSM; (b) partitioning; (c) reduced state table.**

Input	X	
P.S.	0	1
A	$B/0$	$C/0$
B	$D/0$	$E/0$
C	$F/0$	$G/0$
D	$A/0$	$A/0$
E	$A/0$	$A/1$
F	$A/0$	$A/1$
G	$A/1$	$A/1$

(a)

$$P_0 = (ABCDEFG)$$
$$P_1 = (ABCD)(EF)(G)$$
$$P_2 = (AD)(B)(C)(EF)(G)$$
$$P_3 = (A)(B)(C)(D)(EF)(G)$$
$$P_4 = (A)(B)(C)(D)(EF)(G) \surd$$

(b)

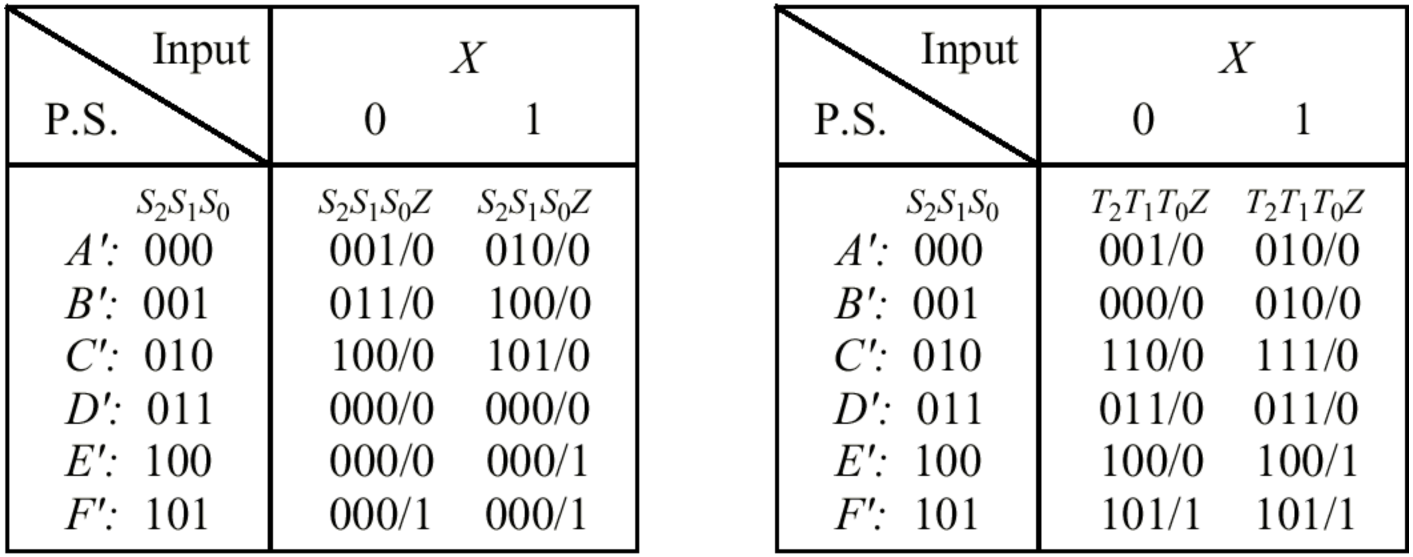
Input	X	
P.S.	0	1
A: A'	$B'/0$	$C'/0$
B: B'	$D'/0$	$E'/0$
C: C'	$E'/0$	$F'/0$
D: D'	$A'/0$	$A'/0$
EF: E'	$A'/0$	$A'/1$
G: F'	$A'/1$	$A'/$ 1

(c)

© 1999 M. Murdocca and V. Heuring

Majority FSM State Assignment

- **(a) State assignment for reduced majority FSM using D flip-flops; and (b) using T flip-flops.**



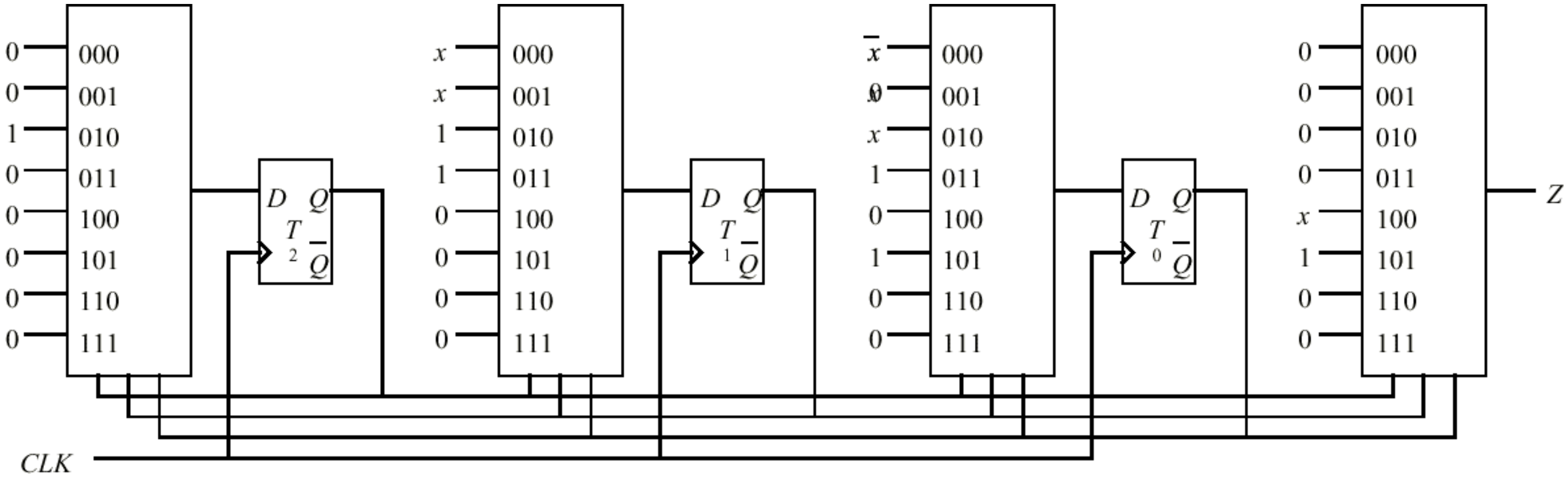
Input P.S.	X	
	0	1
$S_2S_1S_0$	$S_2S_1S_0Z$	$S_2S_1S_0Z$
A': 000	001/0	010/0
B': 001	011/0	100/0
C': 010	100/0	101/0
D': 011	000/0	000/0
E': 100	000/0	000/1
F': 101	000/1	000/1

(a)

Input P.S.	X	
	0	1
$S_2S_1S_0$	$T_2T_1T_0Z$	$T_2T_1T_0Z$
A': 000	001/0	010/0
B': 001	000/0	010/0
C': 010	110/0	111/0
D': 011	011/0	011/0
E': 100	100/0	100/1
F': 101	101/1	101/1

(b)

Majority FSM Circuit



© 1999 M. Murdocca and V. Heuring

Review of Objectives

- Declared the 2 conditions necessary for states in an FSM to be EQUIVALENT

- Used a state reduction to lower the number of required FSM states which slightly altered the FSM design flow

- Wrote D, RS, T and JK FF excitation tables

- Employed FF excitation tables to allow the application of any kind of FF in a given FSM design