

# Digital Systems Design and Laboratory

## [ 15. Reduction of State Tables and State Assignment ]

Chung-Wei Lin

[cwlin@csie.ntu.edu.tw](mailto:cwlin@csie.ntu.edu.tw)

CSIE Department

National Taiwan University

# Sequential Logic Design

- ❑ Unit 11: Latches and Flip-Flops
- ❑ Unit 12: Registers and Counters
- ❑ Units 13--15: Finite State Machines
- ❑ Unit 16: Summary
  
- ❑ Designing a sequential circuit
  - Construct a state graph or state table (Unit 14)
  - Simplify it (Unit 15)
  - Derive flip-flop input equations and output equations (Unit 12)

# Outline

## ☒ **Elimination of Redundant States**

- ☐ Equivalent States
- ☐ Implication Table
- ☐ Equivalent Sequential Circuits
- ☐ Incompletely Specified State Tables
- ☐ Derivation of Flip-Flop Input Equations
- ☐ Equivalent State Assignments
- ☐ Guidelines for State Assignment
- ☐ One-Hot State Assignment

# "0101/1001" Detector

❑ Output "1" if detecting "0101" or "1001"

❑ Reset after every 4 inputs

❑ Example

➤ Input X     0 1 0 1 0 0 1 0 1 0 0 1 0 1 0 0

➤ Output Z    0 0 0 **1** 0 0 0 0 0 0 0 0 **1** 0 0 0 0

❑ Complete state graph

➤  $S_0$ : Reset

➤  $S_1$ : 0

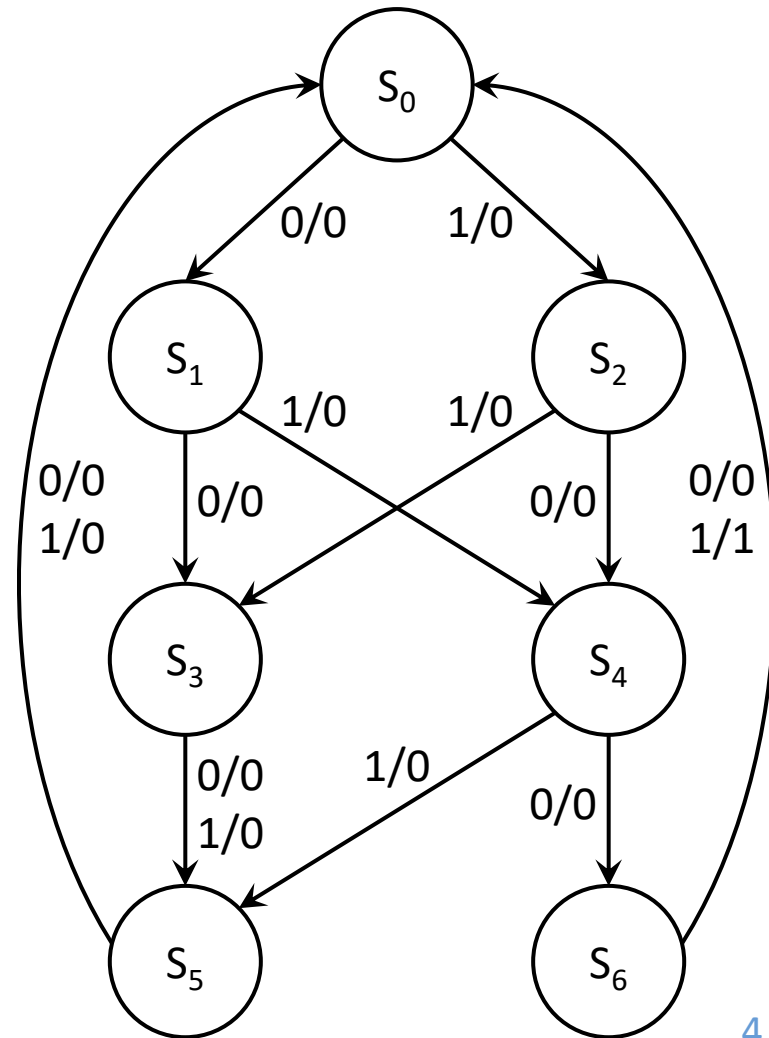
➤  $S_2$ : 1

➤  $S_3$ : Two inputs received; Z must be 0

➤  $S_4$ : 01 or 10

➤  $S_5$ : Three inputs received; Z must be 0

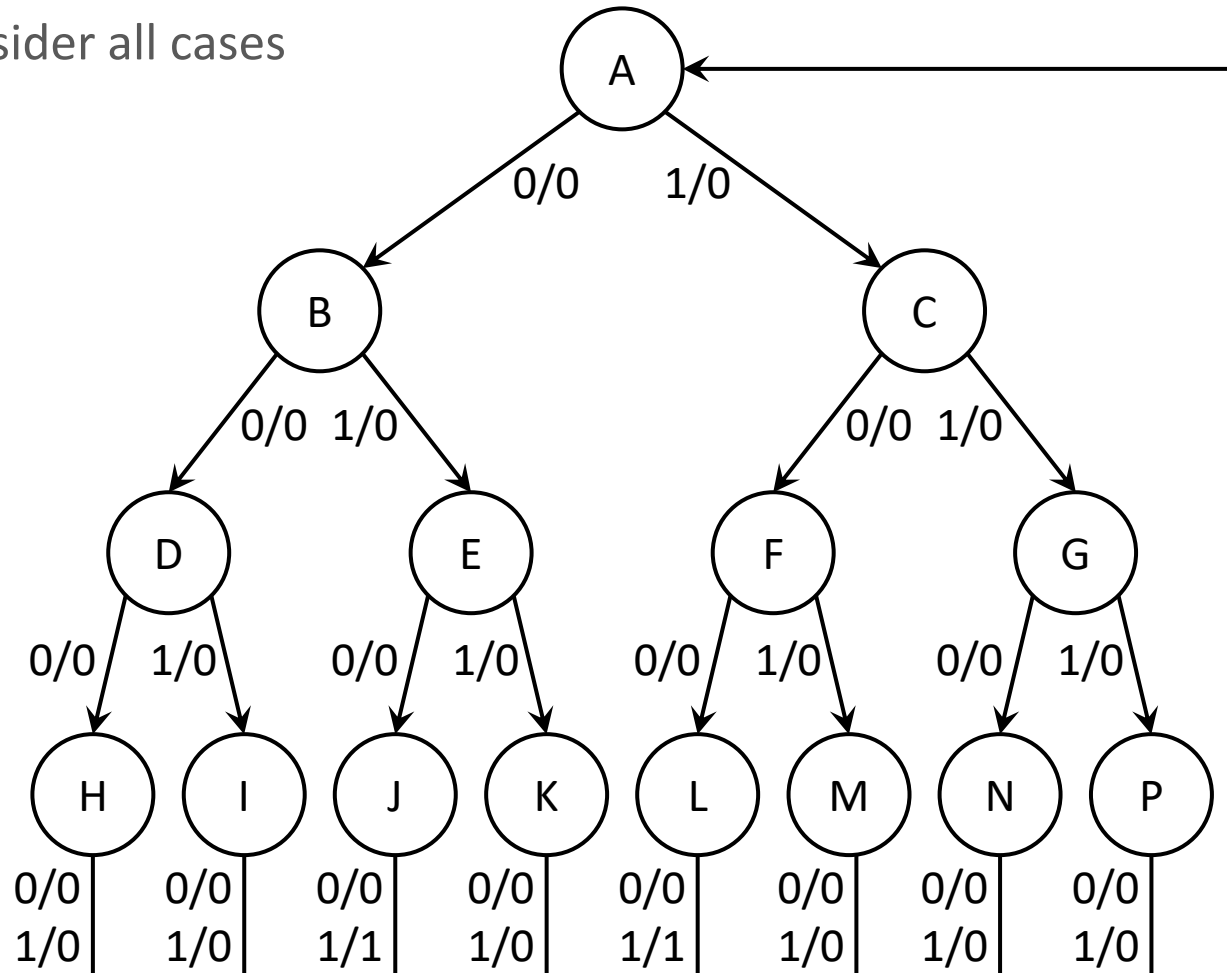
➤  $S_6$ : 010 or 100



# Elimination of Redundant States (1/4)

❑ "Another" state graph for "0101/1001" sequence detector

➤ Consider all cases



# Elimination of Redundant States (2/4)

- ❑ Many similar cases

- ❑ Equivalent state

  - Same next states

  - Same outputs

- ❑ Check H

  - $H = I = K = M = N = P$

- ❑ Check J

  - $J = L$

Input Sequence	Present State	Next State		Present Output	
		X = 0	X = 1	X = 0	X = 1
Reset	A	B	C	0	0
0	B	D	E	0	0
1	C	F	G	0	0
00	D	H	I	0	0
01	E	J	K	0	0
10	F	L	M	0	0
11	G	N	P	0	0
000	H	A	A	0	0
001	I	A	A	0	0
010	J	A	A	0	1
011	K	A	A	0	0
100	L	A	A	0	1
101	M	A	A	0	0
110	N	A	A	0	0
111	P	A	A	0	0

# Elimination of Redundant States (3/4)

## ☐ Check E

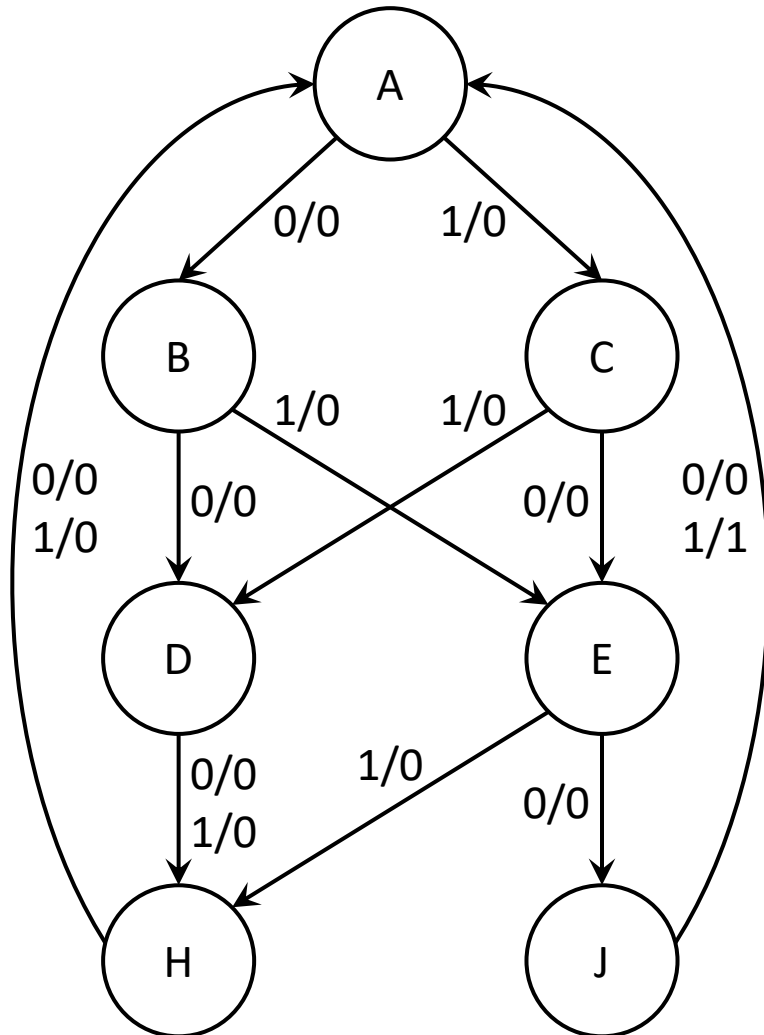
➤  $E = F$

## ☐ Check D

➤  $D = G$

Input Sequence	Present State	Next State		Present Output	
		X = 0	X = 1	X = 0	X = 1
Reset	A	B	C	0	0
0	B	D	E	0	0
1	C	F	G	0	0
00	D	H	I → H	0	0
01	E	J	K → H	0	0
10	F	L → J	M → H	0	0
11	G	N → H	P → H	0	0
000	H	A	A	0	0
001	I	A	A	0	0
010	J	A	A	0	1
011	K	A	A	0	0
100	L	A	A	0	1
101	M	A	A	0	0
110	N	A	A	0	0
111	P	A	A	0	0

# Elimination of Redundant States (4/4)



Input Sequence	Present State	Next State		Present Output	
		X = 0	X = 1	X = 0	X = 1
Reset	A	B	C	0	0
0	B	D	E	0	0
1	C	F → <b>E</b>	G → <b>D</b>	0	0
00	D	H	H	0	0
01	E	J	H	0	0
10	F	J	H	0	0
11	G	H	H	0	0
000	H	A	A	0	0
001	I	A	A	0	0
010	J	A	A	0	1
011	K	A	A	0	0
100	L	A	A	0	1
101	M	A	A	0	0
110	N	A	A	0	0
111	P	A	A	0	0



# Outline

- ❑ Elimination of Redundant States
- ❑ **Equivalent States**
- ❑ Implication Table
- ❑ Equivalent Sequential Circuits
- ❑ Incompletely Specified State Tables
- ❑ Derivation of Flip-Flop Input Equations
- ❑ Equivalent State Assignments
- ❑ Guidelines for State Assignment
- ❑ One-Hot State Assignment

# State Equivalence

## □ Definition

- $N_1, N_2$ : sequential circuits (not necessarily different)
- $\underline{X}$ : a sequence of inputs of arbitrary length
- Then, state  $p$  in  $N_1 \equiv$  state  $q$  in  $N_2$  if and only if  $\lambda_1(p, \underline{X}) = \lambda_2(q, \underline{X})$  for every possible input sequence  $\underline{X}$ 
  - $\lambda$ : output
- Difficult to check the equivalence using this definition!
  - Infinite number of input sequences

## □ Theorem

- Two states  $p$  and  $q$  of a sequential circuit are equivalent if and only if for every single input  $X$ , the outputs are the same and the next states are equivalent, i.e.,  $\lambda(p, X) = \lambda(q, X)$  and  $\delta(p, X) \equiv \delta(q, X)$ 
  - $\delta$ : next state
  - Note that the next state do not have to be equal, just equivalent

# Example: State Equivalence

❑ The following state table has no equivalent states

➤ The only possible pair of equivalent states is  $S_0$  and  $S_2$

•  $S_0 \equiv S_2$  if and only if  $S_3 \equiv S_3$ ,  $S_2 \equiv S_0$ ,  $S_1 \equiv S_1$ , and  $S_0 \equiv S_1$

➤ However  $S_0 \equiv S_1$  is not true due to different outputs

Present State	Next State				Present Output			
	$X_1X_2 = 00$	01	10	11	$X_1X_2 = 00$	01	10	11
$S_0$	$S_3$	$S_2$	$S_1$	$S_0$	00	10	11	01
$S_1$	$S_0$	$S_1$	$S_2$	$S_3$	10	10	11	11
$S_2$	$S_3$	$S_0$	$S_1$	$S_1$	00	10	11	01
$S_3$	$S_2$	$S_2$	$S_1$	$S_0$	00	00	01	01

# Outline

- ☐ Elimination of Redundant States
- ☐ Equivalent States
- ☒ **Implication Table**
- ☐ Equivalent Sequential Circuits
- ☐ Incompletely Specified State Tables
- ☐ Derivation of Flip-Flop Input Equations
- ☐ Equivalent State Assignments
- ☐ Guidelines for State Assignment
- ☐ One-Hot State Assignment

# Implication Table Construction (1/3)

- Draw an empty table, where each square represents a pair
- If outputs are different, give it an X (impossible!)
- Write down the implied pair in the square
- Delete self-implied pairs (redundant)

Present State	Next State		Present Output
	X = 0	X = 1	
A	D	C	0
B	F	H	0
C	E	D	1
D	A	E	0
E	C	A	1
F	F	B	1
G	B	H	0
H	C	G	1

B	D-F C-H						
C	X	X					
D	<del>A-D</del> C-E	A-F E-H	X				
E	X	X	<del>C-E</del> A-D	X			
F	X	X	E-F B-D	X	C-F A-B		
G	B-D C-H	B-F	X	A-B E-H	X	X	
H	X	X	C-E D-G	X	A-G	C-F B-G	X
	A	B	C	D	E	F	G

# Implication Table Construction (2/3)

## Iteratively compare states by the implied pairs

- Only for the same output
- First pass (column-by-column in this case) ~~✗~~
- Second pass ~~✗~~

Present State	Next State		Present Output
	X = 0	X = 1	
A	D	C	0
B	F	H	0
C	E	D	1
D	A	E	0
E	C	A	1
F	F	B	1
G	B	H	0
H	C	G	1

B	<del>D-F</del> <del>C-H</del>						
C	X	X					
D	C-E	<del>A-F</del> <del>E-H</del>	X				
E	X	X	A-D	X			
F	X	X	<del>E-F</del> <del>B-D</del>	X	<del>C-F</del> <del>A-B</del>		
G	<del>B-D</del> <del>C-H</del>	<del>B-F</del>	X	<del>A-B</del> <del>E-H</del>	X	X	
H	X	X	<del>C-E</del> <del>D-G</del>	X	<del>A-G</del>	<del>C-F</del> <del>B-G</del>	X
	A	B	C	D	E	F	G

# Implication Table Construction (3/3)

## Find equivalent states

➤ For each square  $i-j$  which does not contain an X,  $i \equiv j$

## The same procedure for Moore and Mealy machines

Present State	Next State		Present Output
	X = 0	X = 1	
A	D → <b>A</b>	C	0
B	F	H	0
C	E → <b>C</b>	D → <b>A</b>	1
D	A	E	0
E	C	A	1
F	F	B	1
G	B	H	0
H	C	G	1

B	<del>D-F</del> <del>C-H</del>						
C	X	X					
D	C-E	<del>A-F</del> <del>E-H</del>	X				
E	X	X	A-D	X			
F	X	X	<del>E-F</del> <del>B-D</del>	X	<del>C-F</del> <del>A-B</del>		
G	<del>B-D</del> <del>C-H</del>	<del>B-F</del>	X	<del>A-B</del> <del>E-H</del>	X	X	
H	X	X	<del>C-E</del> <del>D-G</del>	X	<del>A-G</del>	<del>C-F</del> <del>B-G</del>	X
	A	B	C	D	E	F	G

# Outline

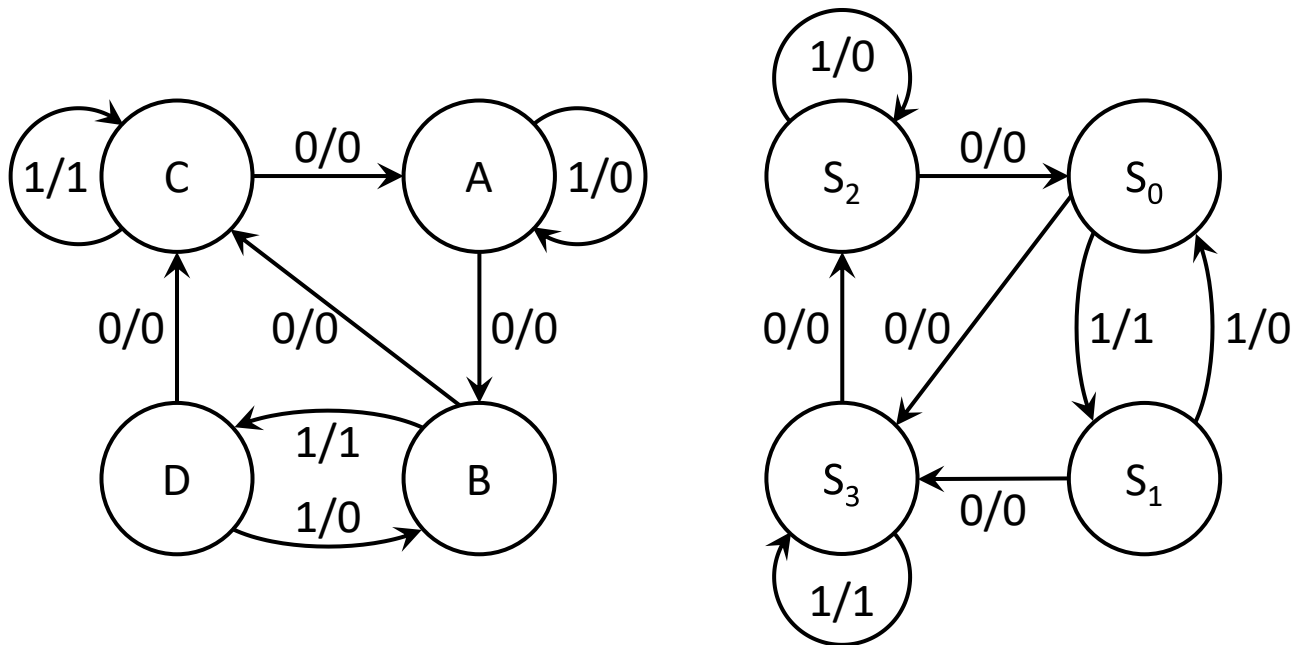
- ☐ Elimination of Redundant States
- ☐ Equivalent States
- ☐ Implication Table
- ☒ **Equivalent Sequential Circuits**
- ☐ Incompletely Specified State Tables
- ☐ Derivation of Flip-Flop Input Equations
- ☐ Equivalent State Assignments
- ☐ Guidelines for State Assignment
- ☐ One-Hot State Assignment



# Equivalent Sequential Circuits

## □ Definition

- Two sequential circuits are equivalent:  $N_1 \equiv N_2$  if
- For each state  $p$  in  $N_1$ , there is a state  $q$  in  $N_2$  such that  $p \equiv q$
- For each state  $s$  in  $N_2$ , there is a state  $t$  in  $N_1$  such that  $s \equiv t$



# Implication Table

□  $N_1 \equiv N_2$ ?

Present State	Next State		Present Output	
	X = 0	X = 1	X = 0	X = 1
A	B	A	0	0
B	C	D	0	1
C	A	C	0	1
D	C	B	0	0

Present State	Next State		Present Output	
	X = 0	X = 1	X = 0	X = 1
$S_0$	$S_3$	$S_1$	0	1
$S_1$	$S_3$	$S_0$	0	0
$S_2$	$S_0$	$S_2$	0	0
$S_3$	$S_2$	$S_3$	0	1

$S_0$	X	C- $S_3$ D- $S_1$	<del>A-<math>S_3</math></del> <del>C-<math>S_1</math></del>	X
$S_1$	<del>B-<math>S_3</math></del> <del>A-<math>S_0</math></del>	X	X	C- $S_3$ B- $S_0$
$S_2$	B- $S_0$ <del>A-<math>S_2</math></del>	X	X	<del>C-<math>S_0</math></del> <del>B-<math>S_2</math></del>
$S_3$	X	<del>D-<math>S_2</math></del> <del>D-<math>S_3</math></del>	<del>A-<math>S_2</math></del> <del>C-<math>S_3</math></del>	X
	A	B	C	D

$\Rightarrow$ 
 $A \equiv S_2$   
 $B \equiv S_0$   
 $C \equiv S_3$   
 $D \equiv S_1$ 
 $\Rightarrow N_1 \equiv N_2$

# Outline


- ❑ Elimination of Redundant States
- ❑ Equivalent States
- ❑ Implication Table
- ❑ Equivalent Sequential Circuits
- ❑ **Incompletely Specified State Tables**
- ❑ Derivation of Flip-Flop Input Equations
- ❑ Equivalent State Assignments
- ❑ Guidelines for State Assignment
- ❑ One-Hot State Assignment

# How about Don't Cares?

❑ A state table is incompletely specified if don't cares are present

➤ Certain sequences will never occur as inputs

Present State	Next State		Present Output	
	X = 0	X = 1	X = 0	X = 1
A	B	D	0	---
B	C	D	---	0
C	B	A	1	0
D	C	D	0	1



Present State	Next State		Present Output	
	X = 0	X = 1	X = 0	X = 1
A	B	A	0	1
B	B	A	1	0
C	B	A	1	0
D	C	D	0	1

# Outline

- ❑ Elimination of Redundant States
- ❑ Equivalent States
- ❑ Implication Table
- ❑ Equivalent Sequential Circuits
- ❑ Incompletely Specified State Tables
- ❑ **Derivation of Flip-Flop Input Equations**
- ❑ Equivalent State Assignments
- ❑ Guidelines for State Assignment
- ❑ One-Hot State Assignment

# Recap: Sequential Logic Design

## □ Designing a sequential circuit

- Construct a state graph or state table (Unit 14)
- Simplify it (Unit 15)
  - State reduction
  - State assignment
  - Choice of flip-flops
- Derive flip-flop input equations and output equations (Unit 12)

# State Assignment and Transition Table

## □ Given a state table

- 7 states
- 3 flip-flops

## □ State assignment

- Many possible ways
- Example

- $S_0 = 000, S_1 = 110,$   
 $S_2 = 001, S_3 = 111,$   
 $S_4 = 011, S_5 = 101, S_6 = 010$

## □ Transition table

- State table + state assignment

## □ If using D flip-flops

- By Karnaugh maps

- $A^+ = D_A = X', B^+ = D_B = X'C' + A'C + A'B, C^+ = D_C = A + XB$

	X = 0	X = 1	X = 0	X = 1
$S_0$	$S_1$	$S_2$	0	0
$S_1$	$S_3$	$S_2$	0	0
$S_2$	$S_1$	$S_4$	0	0
$S_3$	$S_5$	$S_2$	0	0
$S_4$	$S_1$	$S_6$	0	0
$S_5$	$S_5$	$S_2$	1	0
$S_6$	$S_1$	$S_6$	0	1

	$A^+B^+C^+$		Z	
	X = 0	X = 1	X = 0	X = 1
000	110	001	0	0
110	111	001	0	0
001	110	011	0	0
111	101	001	0	0
011	110	010	0	0
101	101	001	1	0
010	110	010	0	1
100	---	---	---	---

# Recap: Derivation of Flip-Flop Input Equations

- Determine the flip flop input equations from the next-state equations using K-maps

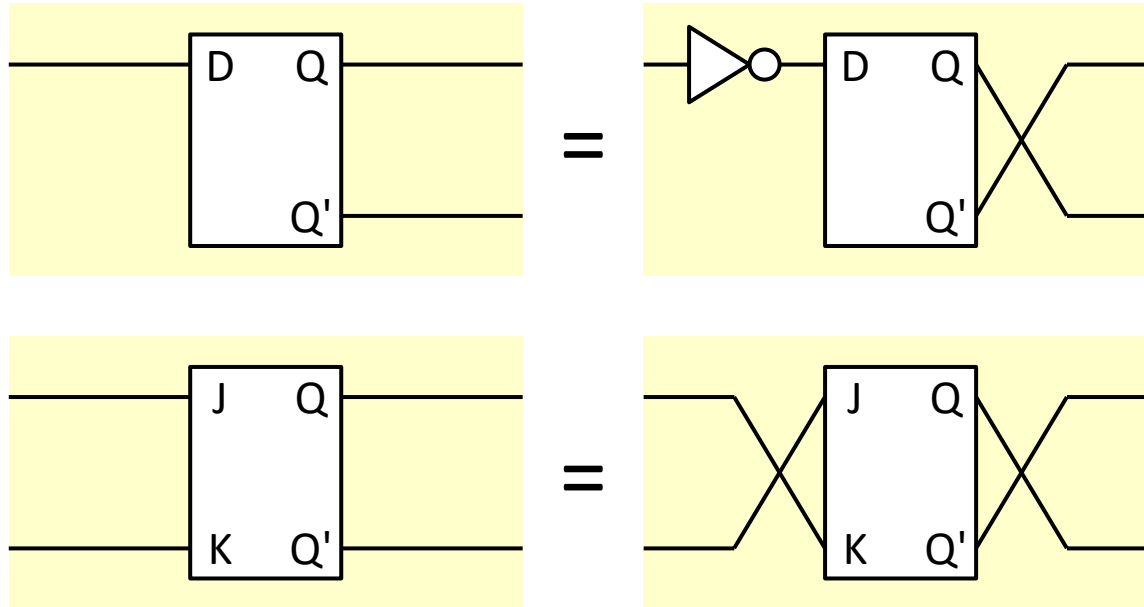
Type of FF	Input	Q = 0		Q = 1		Rules for forming input map from next state map	
		Q <sup>+</sup> = 0	Q <sup>+</sup> = 1	Q <sup>+</sup> = 0	Q <sup>+</sup> = 1	Q = 0 Half of Map	Q = 1 Half of Map
D	D	0	1	0	1	No change	No change
T	T	0	1	1	0	No change	Complement
S-R	S	0	1	0	X	No change	Replace 1's with X's
	R	X	0	1	0	Replace 0's with X's Replace 1's with 0's	Complement
J-K	J	0	1	X	X	No change	Fill in with X's
	K	X	X	1	0	Fill in with X's	Complement



# Outline

- ❑ Elimination of Redundant States
- ❑ Equivalent States
- ❑ Implication Table
- ❑ Equivalent Sequential Circuits
- ❑ Incompletely Specified State Tables
- ❑ Derivation of Flip-Flop Input Equations
- ❑ **Equivalent State Assignments**
- ❑ Guidelines for State Assignment
- ❑ One-Hot State Assignment

# Equivalent State Assignments



# Outline

- ☐ Elimination of Redundant States
- ☐ Equivalent States
- ☐ Implication Table
- ☐ Equivalent Sequential Circuits
- ☐ Incompletely Specified State Tables
- ☐ Derivation of Flip-Flop Input Equations
- ☐ Equivalent State Assignments
- ☒ **Guidelines for State Assignment**
- ☐ One-Hot State Assignment

# State Assignment

- ❑ State assignment determines the cost of the logic required to realize a sequential circuit
  - A good state assignment leads to a Karnaugh map that can easily be simplified and results in few terms
- ❑ Idea: Place 1's together (or 0's together) on the next-state maps

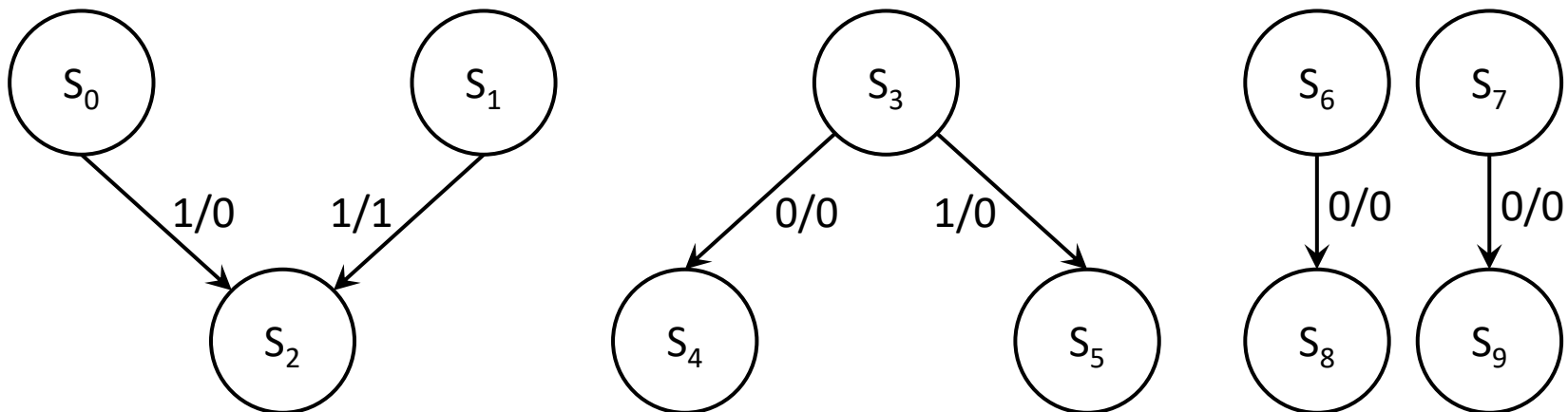
# Guidelines

## ❑ Guidelines

- No guarantee a minimum solution
- Work for D & J-K flip-flops, not for T and S-R flip-flops

## ❑ Adjacent assignments

- For a given input, states with the same next state ( $S_0$  and  $S_1$ )
- States which are next states of the same state ( $S_4$  and  $S_5$ )
- States which have the same output ( $S_6$  and  $S_7$ )
  - Place 1's together on the output maps



# Outline

- ❑ Elimination of Redundant States
- ❑ Equivalent States
- ❑ Implication Table
- ❑ Equivalent Sequential Circuits
- ❑ Incompletely Specified State Tables
- ❑ Derivation of Flip-Flop Input Equations
- ❑ Equivalent State Assignments
- ❑ Guidelines for State Assignment
- ❑ **One-Hot State Assignment**

# One-Hot State Assignment

## ❑ Different strategy

- Do not care about how many flip-flops used
  - Examples: Complex Programmable Logic Device (CPLD) and Field Programmable Gate Array (FPGA)
- Only care about the speed

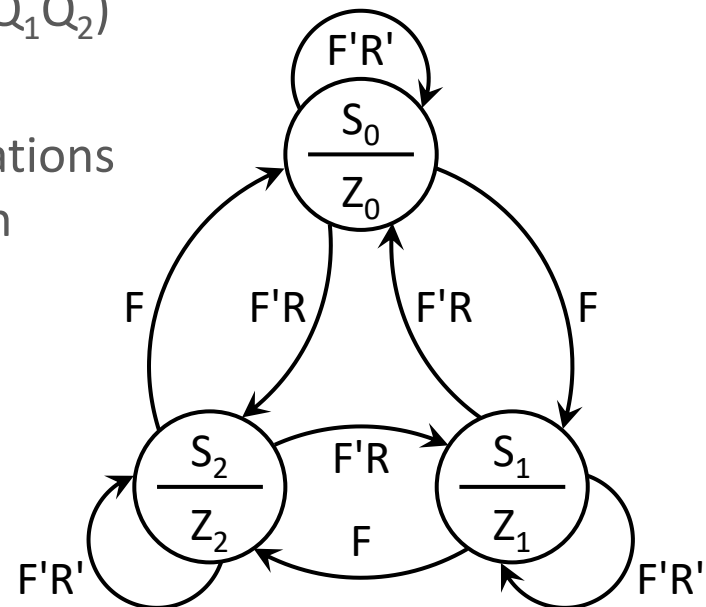
## ❑ One-hot state assignment: One flip-flop for each state

- Example: 3 flip-flops for 3 states ( $Q_0Q_1Q_2$ )

- $S_0 = 100, S_1 = 010, S_2 = 001$

- Write next-state and output (Z) equations directly by inspecting the state graph

- $Q_0^+ = F'R'Q_0 + F'RQ_1 + FQ_2$
  - $Q_1^+ = F'R'Q_1 + F'RQ_2 + FQ_0$
  - $Q_2^+ = F'R'Q_2 + F'RQ_0 + FQ_1$
  - $Z = Z_0Q_0 + Z_1Q_1 + Z_2Q_2$



# Q&A