UNIT 15

Reduction of State Tables
State Assignment

This chapter includes:

Learning Objectives

1. Define equivalent states, state several ways of testing for state equivalence, and determine if two states are equivalent.
2. Define equivalent sequential circuits and determine if two circuits are equivalent.
3. Reduce a state table to a minimum number of rows.
4. Specify a suitable set of state assignments for a state table, eliminating those assignments which are equivalent with respect to the cost of realizing the circuit.

Learning Objectives

5. State three guidelines which are useful in making state assignments, and apply these to making a good state assignment for a given state table.
6. Given a state table and assignment, form the transition table and derive flip-flop input equations.
7. Make a one-hot state assignment for a state graph and write the next-state and output equations by inspection.

Elimination of Redundant States

TABLE 15-1
State Table for Sequence Detector

© Cengage Learning 2014

Input Sequence	Present State	Next State $X = 0$	Next State $X = 1$	Present Output $X = 0$	Present Output $X = 1$
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

Eliminating Redundant States from Table 15-1:

- Looking at the table, we see that there is no way of telling states H and I apart. That is, if we start in state H, the next state is A and the output is 0; similarly, if we start in state I, the next state is A and the output is 0. We say that H is equivalent to I(H≡I).
- Similarly, rows K, M, N, and P have the same next state and output as H, so K, M, N, and P can be replaced by H, and these rows can be deleted.
- Also, the next states and outputs are the same for rows J and L, so J≡L. Thus, L can be replaced with J and eliminated from the table. The result is shown in Table 15-2.
- Having made these changes in the table, rows D and G are identical and so are rows E and F. Therefore, D≡G, and E≡F, so states F and G can be eliminated.

Elimination of Redundant States

TABLE 15-2
State Table for Sequence Detector

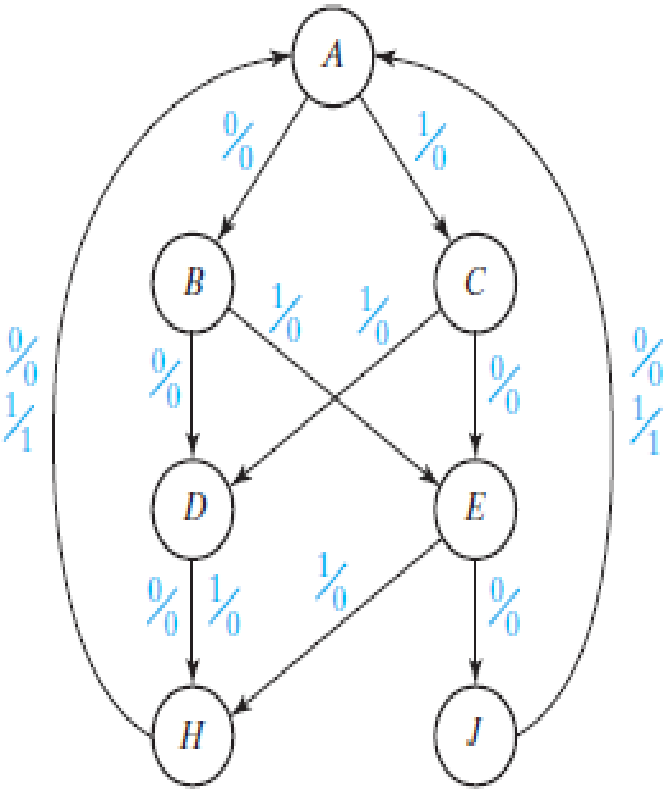
Present State	Next State $X = 0$	Next State $X = 1$	Present Output $X = 0$	Present Output $X = 1$
A	B	C	0	0
B	D	E	0	0
C	~~F~~ E	~~G~~ D	0	0
D	H	~~I~~ H	0	0
E	J	~~K~~ H	0	0
~~F~~	~~L J~~	~~M H~~	~~0~~	~~0~~
~~G~~	~~N H~~	~~P H~~	~~0~~	~~0~~
H	A	A	0	0
~~I~~	~~A~~	~~A~~	~~0~~	~~0~~
J	A	A	0	1
~~K~~	~~A~~	~~A~~	~~0~~	~~0~~
~~L~~	~~A~~	~~A~~	~~0~~	~~1~~
~~M~~	~~A~~	~~A~~	~~0~~	~~0~~
~~N~~	~~A~~	~~A~~	~~0~~	~~0~~
~~P~~	~~A~~	~~A~~	~~0~~	~~0~~

Elimination of Redundant States

FIGURE 15-1 Reduced State Table and Graph for Sequence Detector

Present State	Next State $X = 0$	$X = 1$	Output $X = 0$	$X = 1$
A	B	C	0	0
B	D	E	0	0
C	E	D	0	0
D	H	H	0	0
E	J	H	0	0
H	A	A	0	0
J	A	A	0	1

(a)



(b)

Equivalent States

State Equivalence:

- Two states are equivalent if there is no way of telling them apart through observation of the circuit inputs and outputs.

We can then state formally the definition of state equivalence as follows:

Definition 15.1 Let N_1 and N_2 be sequential circuits (not necessarily different). Let $\underline{X}$ represent a sequence of inputs of arbitrary length. Then state p in N_1 is equivalent to state q in N_2 iff $\lambda_1(p, \underline{X}) = \lambda_2(q, \underline{X})$ for every possible input sequence $\underline{X}$.

Equivalent States

Theorem15.1[1] Two states p and q of a sequential circuit are equivalent iff for every single input X, the outputs are the same and the next states are equivalent, that is,

$$\lambda(p, X) = \lambda(q, X) \qquad \text{and} \qquad \delta(p, X) \equiv \delta(q, X)$$

where $\lambda(p, X)$ is the output given the present state p and input X, and $\delta(p, X)$ is the next state given the present state p and input X. Note that the next states do not have to be equal, just equivalent. For example, in Table 15-1, $D \equiv G$, but the next states (H and N for $X = 0$, and I and P for $X = 1$) are not equal.

Determination of State Equivalence Using An Implication Table

Implication Table Method Procedure:

1. Construct a chart which contains a square for each pair of states.

2. Compare each pair of rows in the state table. If the outputs associated with states *i and j are different, place an X in square i-j to indicate that i≢j. If the outputs are the same, place the implied pairs in square i-j. (If the next states of i and j are m and n for some input x, then m-n is an implied pair.) If the outputs and next states are the same (or if i-j only implies itself), place a check (Ë) in square i-j to indicate that i≡j.*

Determination of State Equivalence Using An Implication Table

Implication Table Procedure (continued):

3. Go through the table square-by-square. If square *i-j contains the implied pair m-n, and square m-n contains an X, then i≢j, and an X should be placed in square i-j.*

4. If any X's were added in step 3, repeat step 3 until no more X's are added.

5. For each square *i-j which does not contain an X, i≡j.*

Determination of State Equivalence Using An Implication Table

TABLE 15-3

Present State	Next State $X = 0$	1	Present Output
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

FIGURE 15-3
Implication Chart for Table 15-3

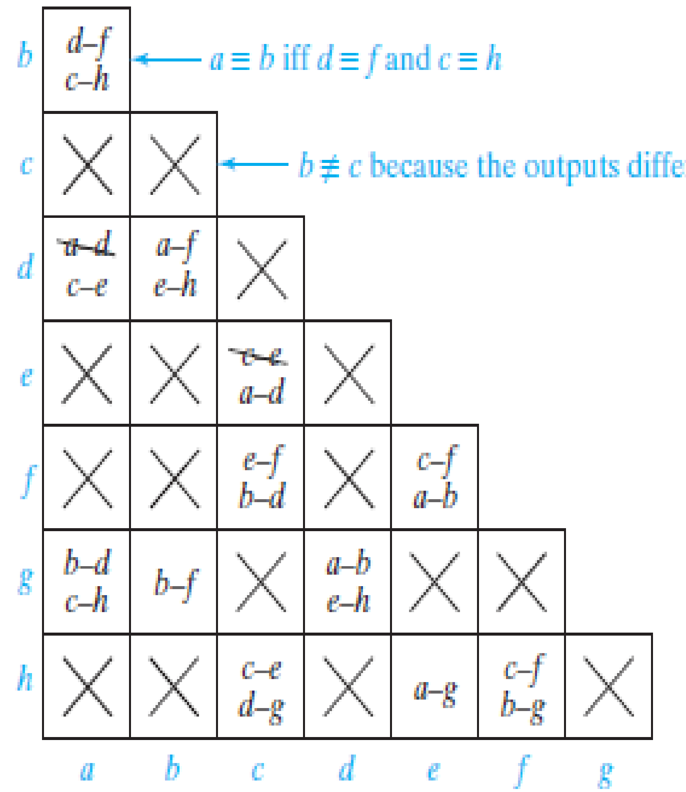
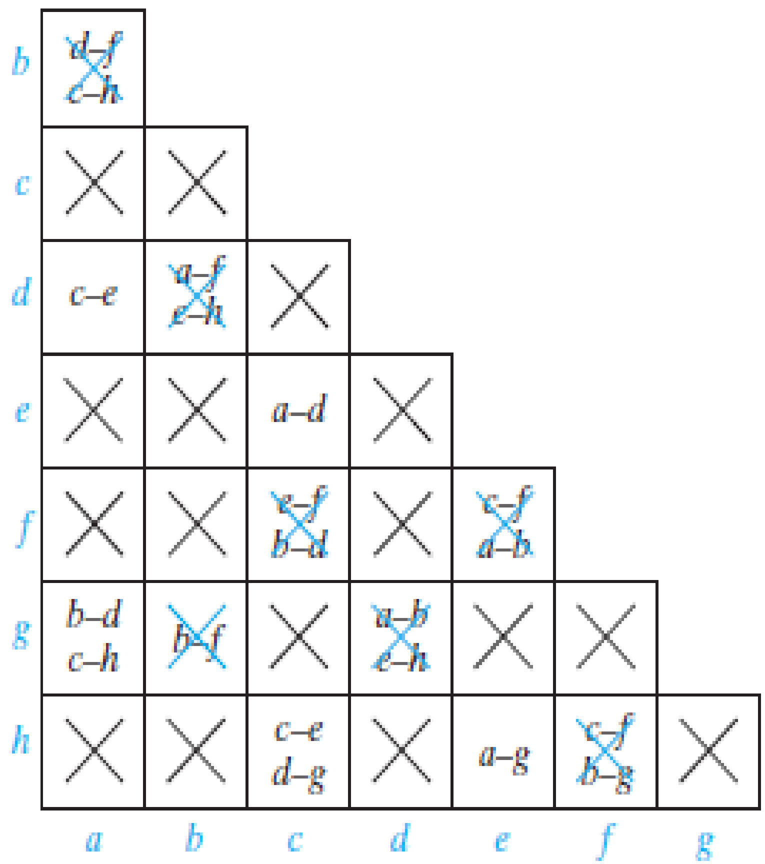


TABLE 15-4

Present State	Next State $X = 0$	1	Output
a	a	c	0
b	f	h	0
c	c	a	1
f	f	b	1
g	b	h	0
h	c	g	1

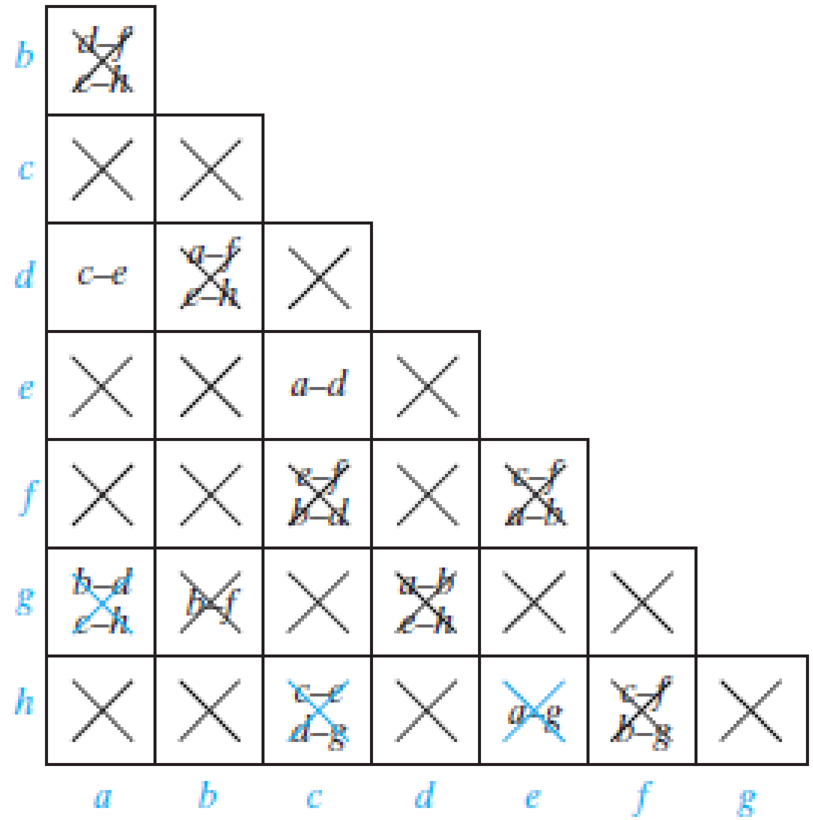
Determination of State Equivalence Using An Implication Table

FIGURE 15-4
Implication Chart After First Pass



Determination of State Equivalence Using An Implication Table

FIGURE 15-5
Implication Chart After Secong Pass



Equivalent Sequential Circuits

Equivalent Sequential Circuits Formal Definition:

Definition 15.2 Sequential circuit N_1 is equivalent to sequential circuit N_2 if for each state p in N_1, there is a state q in N_2 such that $p \equiv q$, and conversely, for each state s in N_2, there is a state t in N_1 such that $s \equiv t$.

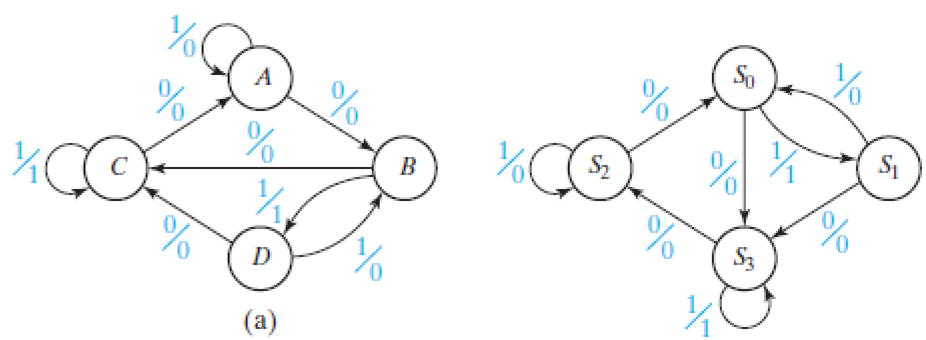
FIGURE 15-6 Tables and Graphs for Equivalent Circuits

N_1

	$X = 0$	1	$X = 0$	1
A	B	A	0	0
B	C	D	0	1
C	A	C	0	1
D	C	B	0	0

N_2

	$X = 0$	1	$X = 0$	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



(a) (b)

Equivalent Sequential Circuits

FIGURE 15-7
Implication Tables for Determining Circuit Equivalence

© Cengage Learning 2014

	A	B	C	D
S_0	X	C–S_3 D–S_1	A–S_3 C–S_1	X
S_1	B–S_3 A–S_0	X	X	C–S_3 B–S_0
S_2	B–S_0 A–S_2	X	X	C–S_0 B–S_2
S_3	X	C–S_2 D–S_3	A–S_2 C–S_3	X

(a)

	A	B	C	D
S_0	X	C–S_3 D–S_1	$\cancel{A\text{–}S_3}$ $\cancel{C\text{–}S_1}$	X
S_1	$\cancel{B\text{–}S_3}$ $\cancel{A\text{–}S_0}$	X	X	C–S_3 B–S_0
S_2	B–S_0 A–S_2	X	X	$\cancel{C\text{–}S_0}$ $\cancel{B\text{–}S_2}$
S_3	X	$\cancel{C\text{–}S_2}$ $\cancel{D\text{–}S_3}$	A–S_2 C–S_3	X

(b)

Equivalent State pairs:

$$A \equiv S_2 \qquad B \equiv S_0 \qquad C \equiv S_3 \qquad D \equiv S_1$$

Since each state in N_1 has an equivalent state in N_2 and conversely, $N_1 \equiv N_2$.

Reducing Incompletely Specified State Tables

Incompletely Specified Examples:

TABLE 15-5 Incompletely Specified Examples

© Cengage Learning 2014

(a)

Present State	Next State $X=0$	Next State $X=1$	Output $X=0$	Output $X=1$
S_0	S_1	S_3	0	$-^2$
S_1	S_2	S_3	$-^1$	0
S_2	S_1	S_0	1	0
S_3	S_2	S_3	0	1

(b)

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

Reducing Incompletely Specified State Tables

FIGURE 15-8 Implication Charts for Table 15-5(a)

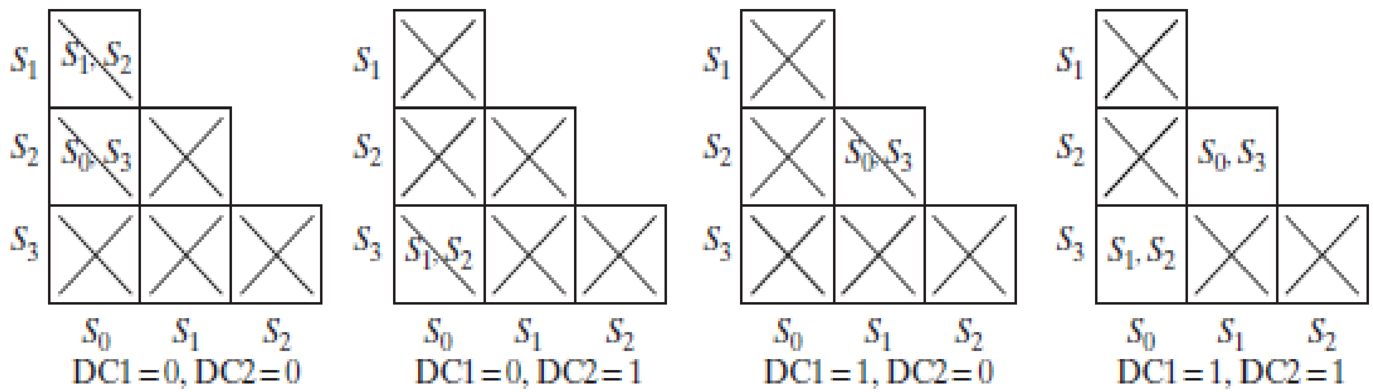


TABLE 15-6 Modified Table 15-5(b)

Present State	Next State $X = 0$	Next State $X = 1$	Output $X = 0$	Output $X = 1$
S_0	S_2	S_1^1	0	1
S_1^1	S_1^2	S_0	0	1
S_1^2	S_1^2	S_0	1	1
S_2	S_2	S_1^1	1	1

Reducing Incompletely Specified State Tables

Procedure to Reduce an Incompletely Specified Table:

- To reduce an incompletely specified table, a minimum of the maximal compatibles are selected, say $C_1, C_2, \ldots, C_k$, *so that*

(1) each state of the table appears in at least one of the C_i, *and*

(2) for each input combination *x and each* C_i, *the next states of the states in* C_i *are contained in some* C_j. *(It may be that* $j=i$*.)*

Reducing Incompletely Specified State Tables

Even Parity Detector for 0 Through 5:

TABLE 15-7
Even Parity Detector For 0 Through 5

	Present State	Next State $X=0$	Next State $X=1$	Output $X=0$	Output $X=1$
	S_0	S_1	S_2	–	–
0	S_1	S_3	S_4	–	–
1	S_2	S_4	–	–	–
00	S_3	S_0	S_0	1	0
01, 10	S_4	S_0	S_0	0	1

FIGURE 15-9
Implication Chart for Table 15-7

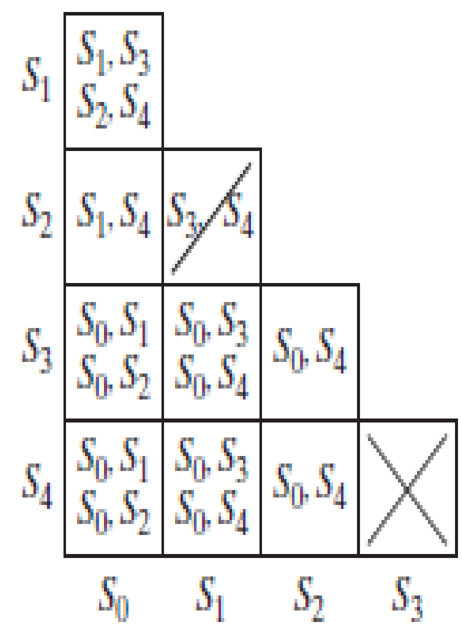


TABLE 15-8
Reduced Table for Table 15-7

	Present State	Next State $X=0$	Next State $X=1$	Output $X=0$	Output $X=1$
$S_0\ S_1\ S_3$	A	A	B	1	0
$S_0\ S_2\ S_4$	B	C	B	0	1
$S_0\ S_1\ S_4$	C	A	B	0	1

Derivation of Flip-Flop Input Equations

Procedure to Derive Flip-Flop Input Equations:

1. Assign flip-flop state values to correspond to the states in the reduced table.

2. Construct a transition table which gives the next states of the flip-flops as a function of the present states and inputs.

3. Derive the next-state maps from the transition table.

4. Find flip-flop input maps from the next-state maps using the techniques developed in Unit 12 and find the flip-flop input equations from the maps.

❖ See pages 516-520 for applications of this procedure.

Equivalent State Assignments

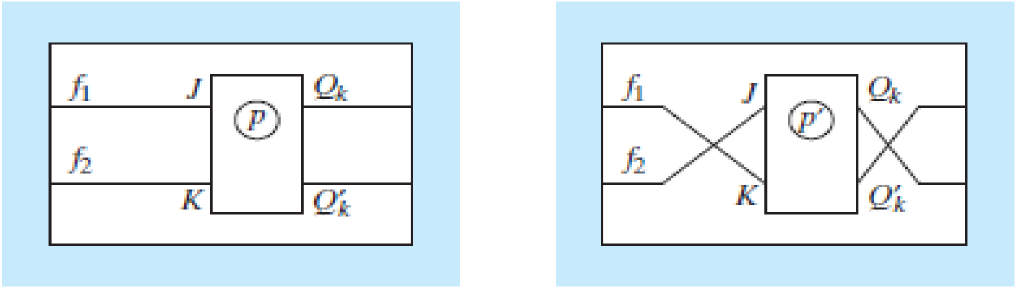
Equivalent State Assignments:

- Now we must assign flip-flop states to correspond to the states in the table.

TABLE 15-11
State Assignments for 3-Row Tables

	1	2	3	4	5	6	7		19	20	21	22	23	24
S_0	00	00	00	00	00	00	01	. . .	11	11	11	11	11	11
S_1	01	01	10	10	11	11	00		00	00	01	01	10	10
S_2	10	11	01	11	01	10	10		01	10	00	10	00	01

FIGURE 15-13
Equivalent Circuits Obtained by Complementing Q_k

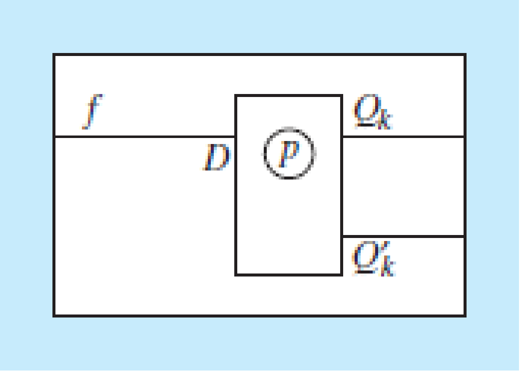


(a) Circuit A

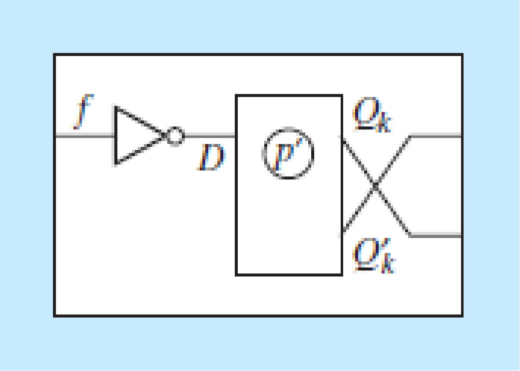
(b) Circuit B
(identical to A except leads to flip-flop Q_k are crossed)

Equivalent State Assignments

FIGURE 15-14
Equivalent Circuits Obtained by Complemeting Q_k



(a) Circuit A



(b) Circuit B (identical to A except for connections to flip-flop Q_k)

TABLE 15-12

Assignments			Present State	Next State		Output	
A_3	B_3	C_3		$X = 0$	1	0	1
00	00	11	S_1	S_1	S_3	0	0
01	10	10	S_2	S_2	S_1	0	1
10	01	01	S_3	S_3	S_3	1	0

Equivalent State Assignments

J and K Input Equations:

Assignment A	Assignment B	Assignment C
$J_1 = XQ_2'$	$J_2 = XQ_1'$	$K_1 = XQ_2$
$K_1 = X'$	$K_2 = X'$	$J_1 = X'$
$J_2 = X'Q_1$	$J_1 = X'Q_2$	$K_2 = X'Q_1'$
$K_2 = X$	$K_1 = X$	$J_2 = X$
$Z = X'Q_1 + XQ_2$	$Z = X'Q_2 + XQ_1$	$Z = X'Q_1' + XQ_2'$
$D_1 = XQ_2'$	$D_2 = XQ_1'$	$D_1 = X' + Q_2'$
$D_2 = X'(Q_1 + Q_2)$	$D_1 = X'(Q_2 + Q_1)$	$D_2 = X + Q_1Q_2$

Equivalent State Assignments

Equivalent and Distinct State Assignments:

- Two state assignments are equivalent if one can be derived from the other by permuting and complementing columns.
- Two state assignments which are not equivalent are said to be distinct.

TABLE 15-13 Nonequivalent Assignments for Three and Four States

	3-State Assignments			4-State Assignments		
States	1	2	3	1	2	3
a	00	00	00	00	00	00
b	01	01	11	01	01	11
c	10	11	01	10	11	01
d	–	–	–	11	10	10

Equivalent State Assignments

TABLE 15-14
Number of Distinct (Nonequivalent) State Assignments

Number of States	Minimum Number of State Variables	Number of Distinct Assignments
2	1	1
3	2	3
4	2	3
5	3	140
6	3	420
7	3	840
8	3	840
9	4	10,810,800
⋮	⋮	⋮
16	4	$\approx 5.5 \times 10^{10}$

Guidelines for State Assignment

Guidelines:

Assignments for two states are said to be adjacent if they differ in only one variable. Thus, 010 and 011 are adjacent, but 010 and 001 are not. The following *guidelines* are useful in making assignments which will place 1's together (or 0's together) on the next-state maps:

1. States which have the same next state for a given input should be given adjacent assignments.
2. States which are the next states of the same state should be given adjacent assignments.

A third guideline is used for simplification of the output function:

3. States which have the same output for a given input should be given adjacent assignments.

Guidelines for State Assignment

Derivation of State Assignment:

FIGURE 15-15

ABC		X = 0	1	0	1
000	S_0	S_1	S_2	0	0
110	S_1	S_3	S_2	0	0
001	S_2	S_1	S_4	0	0
111	S_3	S_5	S_2	0	0
011	S_4	S_1	S_6	0	0
101	S_5	S_5	S_2	1	0
010	S_6	S_1	S_6	0	1

(a) State table

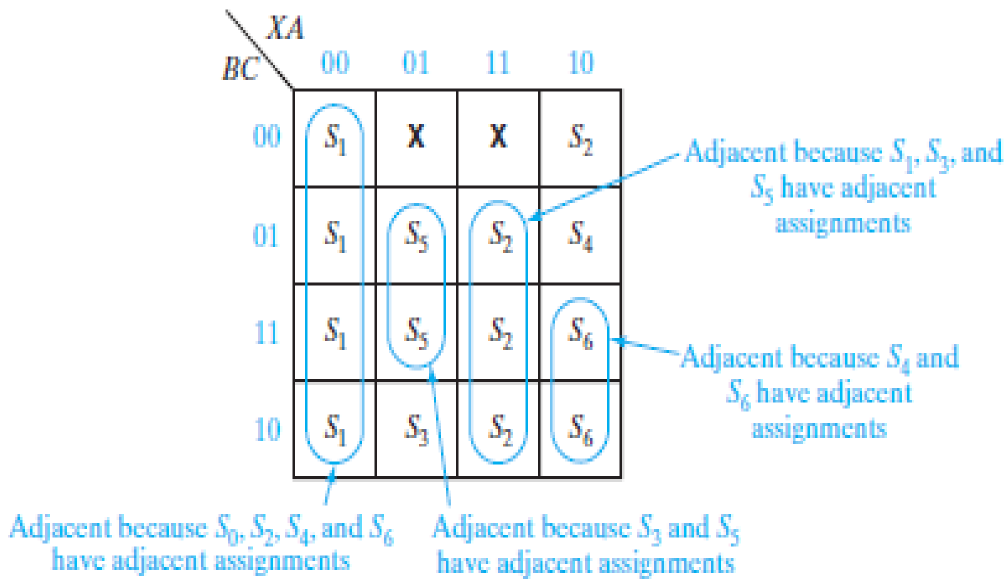
BC \ A	0	1
00	S_0	
01	S_2	S_5
11	S_4	S_3
10	S_6	S_1

BC \ A	0	1
00	S_0	
01	S_1	S_6
11	S_3	S_4
10	S_5	S_2

(b) Assignment maps

Guidelines for State Assignment

FIGURE 15-16
Next-State Maps for Figure 15-15



(a) Next-state maps for Figure 15-15

Guidelines for State Assignment

These pairs are adjacent because S_2 and S_5 have adjacent assignments

BC \ XA	00	01	11	10
00	1	X	X	0
01	1	1	0	0
11	1	1	0	0
10	1	1	0	0

$A^+ = D_A = X'$

BC \ XA	00	01	11	10
00	1	X	X	1
01	1	0	0	1
11	1	0	0	1
10	1	1	0	1

$B^+ = D_B = X'C' + A'C + A'B$

BC \ XA	00	01	11	10
00	0	X	X	1
01	0	1	1	1
11	0	1	1	0
10	0	1	1	0

$C^+ = D_C = A + XB'$

(b) Next-state maps for Figure 15-15 (cont.)

Guidelines for State Assignment

	$X=0$	1	$X=0$	1
a	*a*	*c*	0	0
b	*d*	*f*	0	1
c	*c*	*a*	0	0
d	*d*	*b*	0	1
e	*b*	*f*	1	0
f	*c*	*e*	1	0

(a)

$Q_2Q_3 \backslash Q_1$	0	1
00	*a*	*c*
01		*e*
11	*d*	*b*
10		*f*

$a = 000$
$b = 111$
$c = 100$
$d = 011$
$e = 101$
$f = 110$

(b)

$Q_2Q_3 \backslash Q_1$	0	1
00	*c*	*a*
01		*e*
11	*d*	*b*
10	*f*	

$a = 100$
$b = 111$
$c = 000$
$d = 011$
$e = 101$
$f = 010$

(c)

TABLE 15-15
Transition Table for Figure 15-17(a)

$Q_1Q_2Q_3$	$Q_1^+Q_2^+Q_3^+$ $X=0$	1	$X=0$	1
1 0 0	100	000	0	0
1 1 1	011	010	0	1
0 0 0	000	100	0	0
0 1 1	011	111	0	1
1 0 1	111	010	1	0
0 1 0	000	101	1	0

Guidelines for State Assignment

(Figure 15-18) from the transition table. The D flip-flop input equations can be read directly from these maps:

$$D_1 = Q_1^+ = X'Q_1Q_2' + XQ_1'$$
$$D_2 = Q_2^+ = Q_3$$
$$D_3 = Q_3^+ = XQ_1'Q_2 + X'Q_3$$

and the output equation is

$$Z = XQ_2Q_3 + X'Q_2'Q_3 + XQ_2Q_3'$$

The cost of realizing these equations is 10 gates and 26 gate inputs.

FIGURE 15-18 Next-State and Output Maps for Table 15-15

© Cengage Learning 2014

$Q_1^+ = D_1$

Q_2Q_3 \ XQ_1	00	01	11	10
00	0	1	0	1
01	X	1	0	X
11	0	0	0	1
10	0	X	X	1

$Q_2^+ = D_2$

Q_2Q_3 \ XQ_1	00	01	11	10
00	0	0	0	0
01	X	1	1	X
11	1	1	1	1
10	0	X	X	0

$Q_3^+ = D_3$

Q_2Q_3 \ XQ_1	00	01	11	10
00	0	0	0	0
01	X	1	0	X
11	1	1	0	1
10	0	X	X	1

Z

Q_2Q_3 \ XQ_1	00	01	11	10
00	0	0	0	0
01	X	1	0	X
11	0	0	1	1
10	1	X	X	0

Using a One-Hot State Assignment

One-Hot State Assignment:

- The one-hot assignment uses one flip-flop for each state, so a state machine with N states requires N flip-flops. Exactly one of the flip-flops is set to one in each state.
- For example, a system with four states (S_0, S_1, S_2, and S_3) could use four flip-flops (Q_0, Q_1, Q_2, and Q_3) with the following state assignment:

  S_0: Q_0 Q_1 Q_2 Q_3=1000, S_1: 0100, S_2: 0010, S_3: 0001

  The other 12 combinations are not used.
- In general, when a one-hot state assignment is used, each term in the next-state equation for each flip-flop contains exactly one state variable, and the reduced equation can be written by inspecting the state graph.