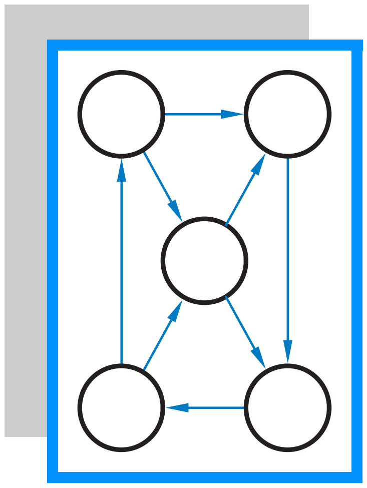
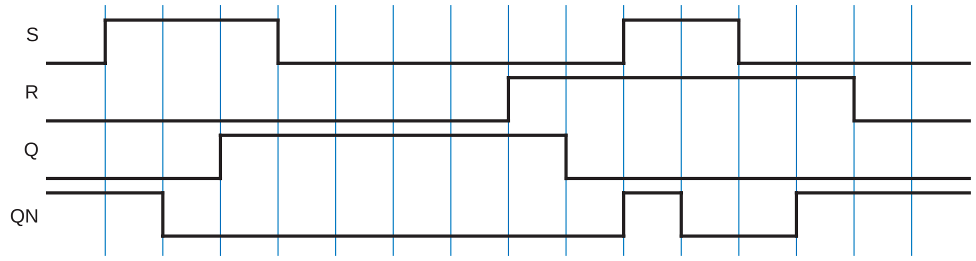


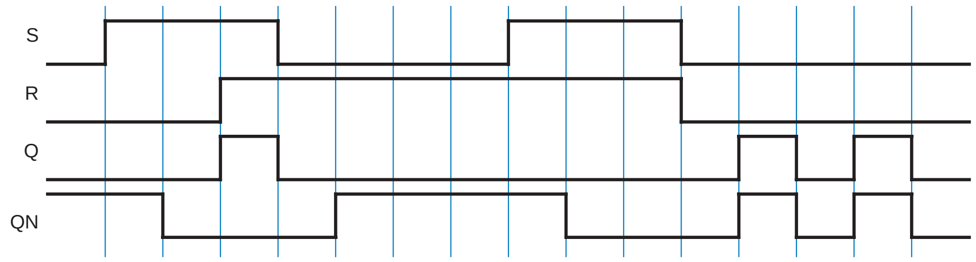
7 Sequential Logic Design Principles

EXERCISE SOLUTIONS

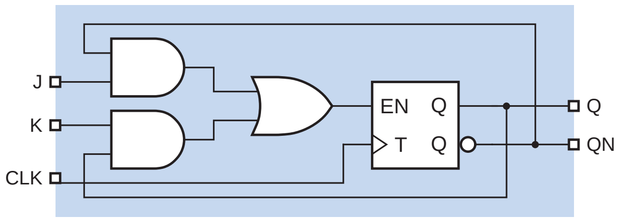
7.2



7.3 The latch oscillates if S and R are negated simultaneously. Many simulator programs will exhibit this same behavior when confronted with such input waveforms.



7.5



7.8 Just tie the J and $\overline{K}$ inputs together and use as the D input.

7.9 Excitation and output equations:

$$D1 = Q1' + Q2$$
$$D2 = Q2' \cdot X$$
$$Z = Q1 + Q2'$$

Excitation/transition table; state/output table:

	EN	
Q1 Q2	0	1
00	10	11
01	10	10
10	00	01
11	10	10
	Q1* Q2*	

	EN		
S	0	1	Z
A	C	D	1
B	C	C	0
C	A	B	1
D	C	C	1
	S*		

7.15 Excitation equations:

$$D2 = (Q1 \oplus Q0) \oplus (Q1' \cdot Q2')$$
$$D1 = Q2$$
$$D0 = Q1$$

Excitation/transition table; state table:

Q2 Q1 Q0	Q2* Q1* Q0*
000	100
001	000
010	101
011	001
100	010
101	110
110	111
111	011

S	S*
A	E
B	A
C	F
D	B
E	C
F	G
G	H
H	D

7.18 Excitation and output equations:

$$J0 = K0 = EN$$
$$J1 = K1 = Q0 \cdot EN$$
$$MAX = EN \cdot Q1 \cdot Q0$$

Note that the characteristic equation for a J-K flip-flop is $Q* = J \cdot Q' + K' \cdot Q$. Thus, we obtain the following transition equations:

$$Q0* = EN' \cdot Q0 + EN \cdot Q0'$$

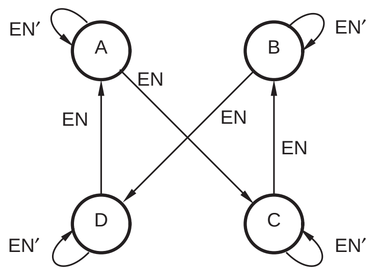
$$Q1* = EN' \cdot Q1 + EN \cdot Q0 \cdot Q1' + EN \cdot Q0' \cdot Q1$$

Transition/output table; state/output table:

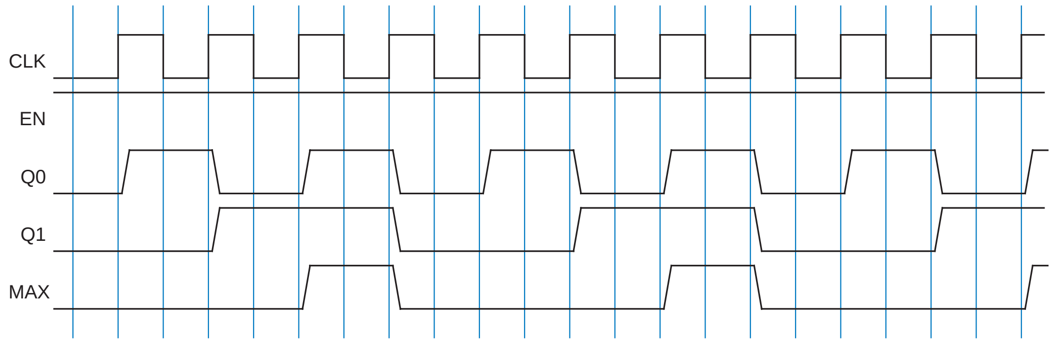
	EN	
Q1 Q2	0	1
00	00,0	01,0
01	01,0	10,0
10	10,0	11,0
11	11,0	00,1
	Q1* Q2*, MAX	

	EN	
S	0	1
A	A,0	B,0
B	B,0	C,0
C	C,0	D,0
D	D,0	A,1
	S*, MAX	

State diagram:



Timing diagram:



7.20 This can be done algebraically. If all of the input combinations are covered, the logical sum of the expressions on all the transitions leaving a state must be 1. If the sum is not 1, it is 0 for all input combinations that are uncovered. For double-covered input combinations, we look at all possible pairs of transitions leaving a state. The product of a pair of transition equations is 1 for any double-covered input combinations.

(a) State D, Y = 0 is uncovered.

(b) State A, (X+Z′) = 0 is uncovered. State B, W = 1 is double-covered; (W+X) = 0 is uncovered. State C, (W+X+Y+Z) = 0 is uncovered; (W·X + W·Y + Z·Y + Z·X) = 1 is double covered. State D, (X·Y + ·X′·Z + W·Z) = 0 is uncovered; (W·X′·Z + W·X·Y·Z) = 1 is double-covered;

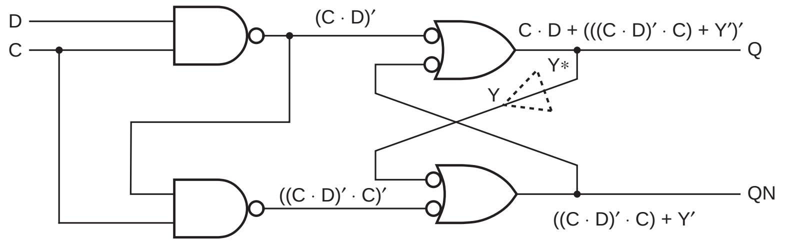
7.21 Table 9–4 on page 804 shows an output-coded state assignment. Here is a corresponding transition list:

S	L3Z	L2Z	L1Z	R1Z	R2Z	R3Z	Transition expression	S*	L3Z*	L2Z*	L1Z*	R1Z*	R2Z*	R3Z*
IDLE	0	0	0	0	0	0	(LEFT + RIGHT + HAZ)′	IDLE	0	0	0	0	0	0
IDLE	0	0	0	0	0	0	LEFT · HAZ′ · RIGHT′	L1	0	0	1	0	0	0
IDLE	0	0	0	0	0	0	HAZ + LEFT · RIGHT	LR3	1	1	1	1	1	1
IDLE	0	0	0	0	0	0	RIGHT · HAZ′ · LEFT′	R1	0	0	0	1	0	0
L1	0	0	1	0	0	0	HAZ′	L2	0	1	1	0	0	0
L1	0	0	1	0	0	0	HAZ	LR3	1	1	1	1	1	1
L2	0	1	1	0	0	0	HAZ′	L3	1	1	1	0	0	0
L2	0	1	1	0	0	0	HAZ	LR3	1	1	1	1	1	1
L3	1	1	1	0	0	0	1	IDLE	0	0	0	0	0	0
R1	0	0	0	1	0	0	HAZ′	R2	0	0	0	1	1	0
R1	0	0	0	1	0	0	HAZ	LR3	1	1	1	1	1	1
R2	0	0	0	1	1	0	HAZ′	R3	0	0	0	1	1	1
R2	0	0	0	1	1	0	HAZ	LR3	1	1	1	1	1	1
R3	0	0	0	1	1	1	1	IDLE	0	0	0	0	0	0
LR3	1	1	1	1	1	1	1	IDLE	0	0	0	0	0	0

The excitation equations and circuit diagram follow directly from this transition list.

7.25 The minimum setup time is the clock period times the duty cycle. That is, the minimum setup time is the time that the clock is 1.

7.27 As shown in Section 7.9.1, the excitation equation for the latch of Figure 7–72 is $Y* = C \cdot D + C' \cdot Y + D \cdot Y$ Below, we analyze Figure X7.27 in the same way:



The feedback equation is

$$\begin{aligned} Y* &= C \cdot D + (((C \cdot D)' \cdot C) + Y')' \\ &= (C \cdot D) + ((C \cdot D)' \cdot C)' \cdot Y \\ &= C \cdot D + ((C \cdot D) + C') \cdot Y \\ &= C \cdot D + (D + C') \cdot Y \\ &= C \cdot D + D \cdot Y + C' \cdot Y \end{aligned}$$

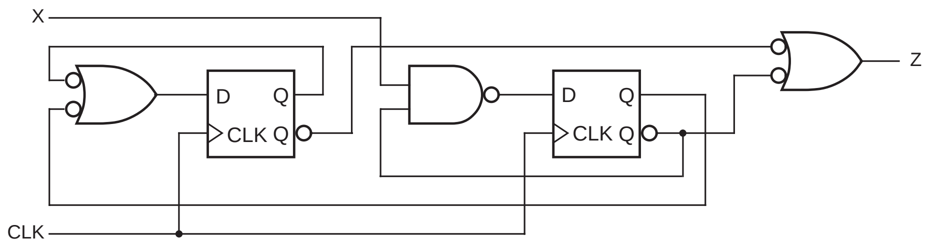
The feedback equations are the same, and so the circuits have identical steady-state behavior.

The circuit in Figure X7.27 is better in two ways. It uses one less gate, and it has one less load on the D input.

7.29 The AND gate in the original circuit is replaced with a NAND gate. As a result, the second flip-flop stores the opposite of the value stored in the original circuit; to compensate, swap connections to its Q and QN outputs.

The OR gates in the original circuit are also replaced with NAND gates. As a result, each input must be connected to a signal of the opposite polarity as before, that is, to the complementary flip-flop output. In the case of connections to the second flip-flop, we swapped outputs twice, so the connections remain the same.

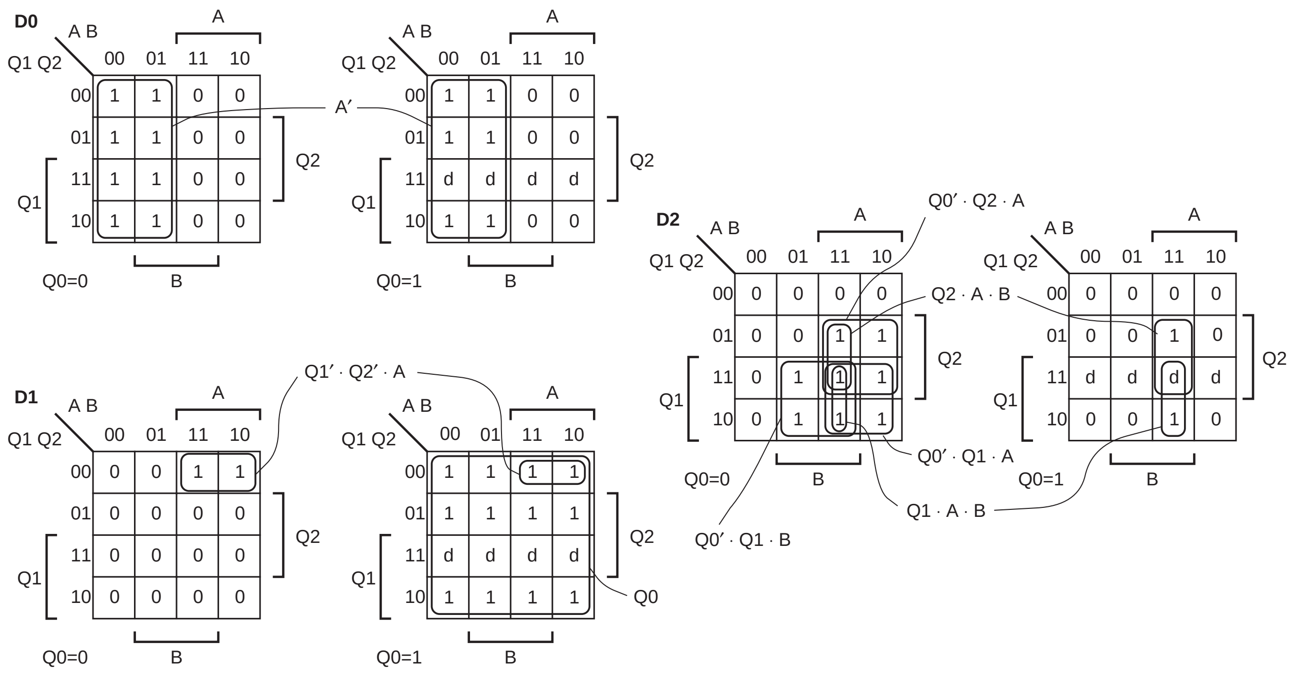
The final circuit below uses three 2-input NAND gates.



7.45 A transition table corresponding to the state table is shown below:

	A B				
Q2 Q1 Q0	00	01	11	10	Z
000	001	001	010	010	0
001	011	011	010	010	0
010	001	001	100	100	0
011	011	011	110	010	1
100	001	101	100	100	1
101	011	011	110	010	1
110	001	101	100	100	1
	Q2* Q1* Q0*				

This table leads to the following Karnaugh maps for the excitation logic, assuming a "minimal cost" treatment of unused states.



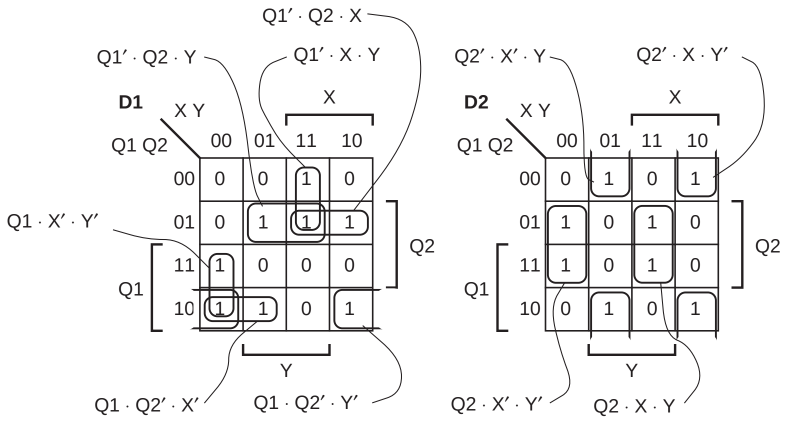
The resulting excitation equations are

$$D0 = A'$$
$$D1 = Q1'\cdot Q2'\cdot A + Q0$$
$$D2 = Q2\cdot A\cdot B + Q0'\cdot Q2\cdot A + Q0'\cdot Q1\cdot A + Q1\cdot A\cdot B + Q0'\cdot Q1\cdot B$$

Ignoring inverters, a circuit realization with the new equations requires one 2-input gate, six 3-input gates, and one 5-input gate. This is more expensive than Figure 7–54, by four gates.

7.49 The new state assignment yields the following transition/excitation table and Karnaugh maps:

	X Y				
Q1 Q0	00	01	11	10	Z
00	00	01	11	01	1
01	01	11	10	11	0
11	11	10	00	10	0
10	10	00	01	00	0
	Q2* Q1* or D1 D2				



This yields the following excitation equations:

$$D1 = Q1'\cdot Q2\cdot X + Q1'\cdot Q2\cdot Y + Q1'\cdot X\cdot Y + Q1\cdot Q2'\cdot X' + Q1\cdot Q2'\cdot Y' + Q1\cdot X'\cdot Y'$$
$$D2 = Q2\cdot X\cdot Y + Q2'\cdot X\cdot Y' + Q2'\cdot X'\cdot Y + Q2\cdot X'\cdot Y'$$

Compared with the results of original state assigment, these equations require two more 3-input AND gates, plus a 6-input OR gate inplace of a 4-input one. However, if we are not restricted to a sum-of-products realization, using the fact that $D2 = Q2 \oplus X \oplus Y$ might make this realization less expensive when discrete gates are used.

7.57 Here is the transition list:

S	Q2	Q1	Q0	Transition expression	S*	Q2*	Q1*	Q0*
IDLE	0	0	0	(LEFT+RIGHT+HAZ)′	IDLE	0	0	0
IDLE	0	0	0	LEFT	L1	0	0	1
IDLE	0	0	0	HAZ	LR3	1	0	0
IDLE	0	0	0	RIGHT	R1	1	0	1
L1	0	0	1	1	L2	0	1	1
L2	0	1	1	1	L3	0	1	0
L3	0	1	0	1	IDLE	0	0	0
R1	1	0	1	1	R2	1	1	1
R2	1	1	1	1	R3	1	1	0
R3	1	1	0	1	IDLE	0	0	0
LR3	1	0	0	1	IDLE	0	0	0

The transition/excitation and output equations below follow directly from the transition list.

$$\begin{aligned}
\mathrm{D2} = \mathrm{Q2*} &= \mathrm{Q2'\cdot Q1'\cdot Q0'\cdot HAZ} \\
&\quad + \mathrm{Q2'\cdot Q1'\cdot Q0'\cdot RIGHT} \\
&\quad + \mathrm{Q2\cdot Q1'\cdot Q0} \\
&\quad + \mathrm{Q2\cdot Q1\cdot Q0} \\
&= \mathrm{Q2'\cdot Q1'\cdot Q0'\cdot (HAZ + RIGHT) + Q2\cdot Q0}
\end{aligned}$$

$$\begin{aligned}
\mathrm{D1} = \mathrm{Q1*} &= \mathrm{Q2'\cdot Q1'\cdot Q0} \\
&\quad + \mathrm{Q2'\cdot Q1\cdot Q0} \\
&\quad + \mathrm{Q2\cdot Q1'\cdot Q0} \\
&\quad + \mathrm{Q2\cdot Q1\cdot Q0} \\
&= \mathrm{Q0}
\end{aligned}$$

$$\begin{aligned}
\mathrm{D0} = \mathrm{Q0*} &= \mathrm{Q2'\cdot Q1'\cdot Q0'\cdot LEFT} \\
&\quad + \mathrm{Q2'\cdot Q1'\cdot Q0'\cdot RIGHT} \\
&\quad + \mathrm{Q2'\cdot Q1'\cdot Q0} \\
&\quad + \mathrm{Q2\cdot Q1'\cdot Q0} \\
&= \mathrm{Q2'\cdot Q1'\cdot Q0'\cdot (LEFT + RIGHT) + Q1'\cdot Q0}
\end{aligned}$$

Starting from the IDLE state, the following transitions may be observed:

S	Q2	Q1	Q0	LEFT	RIGHT	HAZ	Q2*	Q1*	Q0*	S*
IDLE	0	0	0	1	0	1	1	0	1	R1
IDLE	0	0	0	0	1	1	1	0	1	R1
IDLE	0	0	0	1	1	0	1	0	1	R1
IDLE	0	0	0	1	1	1	1	0	1	R1

For each input combination, the machine goes to the R1 state, because R1's encoding is the logical OR of the encodings of the two or three next states that are specified by the ambiguous state diagram.

The behavior aboveis not so good and is a result of synthesis choices—state encoding and logic synthesis method. If a different state encoding were used for R1, or if a different synthesis method were used (e.g., product-of-s-terms), then the results could be different. For example, starting with the transition list given earlier, we can obtain the following set of transition equations using the product-of-s-terms method:

$$\begin{aligned} D2 = Q2^* &= (Q2 + Q1 + Q0 + LEFT + RIGHT + HAZ) \\ &\cdot(Q2 + Q1 + Q0 + LEFT') \\ &\cdot(Q2 + Q1 + Q0') \\ &\cdot(Q2 + Q1' + Q0') \\ &\cdot(Q2 + Q1' + Q0) \\ &\cdot(Q2' + Q1' + Q0) \\ &\cdot(Q2' + Q1 + Q0) \\ &= (Q2 + Q1 + RIGHT + HAZ)\cdot(Q2 + Q1 + LEFT')\cdot(Q2 + Q0')\cdot(Q1' + Q0)\cdot(Q2' + Q0) \end{aligned}$$

$$\begin{aligned} D1 = Q1^* &= (Q2 + Q1 + Q0 + LEFT + RIGHT + HAZ) \\ &\cdot(Q2 + Q1 + Q0 + LEFT') \\ &\cdot(Q2 + Q1 + Q0 + HAZ') \\ &\cdot(Q2 + Q1 + Q0 + RIGHT') \\ &\cdot(Q2 + Q1' + Q0) \\ &\cdot(Q2' + Q1' + Q0) \\ &\cdot(Q2' + Q1 + Q0) \\ &= (Q2 + Q1 + Q0)\cdot(Q1' + Q0)\cdot(Q2' + Q0) \end{aligned}$$

$$\begin{aligned} D0 = Q0^* &= (Q2 + Q1 + Q0 + LEFT + RIGHT + HAZ) \\ &\cdot(Q2 + Q1 + Q0 + HAZ') \\ &\cdot(Q2 + Q1' + Q0') \\ &\cdot(Q2 + Q1' + Q0) \\ &\cdot(Q2' + Q1' + Q0') \\ &\cdot(Q2' + Q1' + Q0) \\ &\cdot(Q2' + Q1 + Q0) \\ &= (Q2 + Q0 + LEFT + RIGHT)\cdot(Q2 + Q0 + HAZ')\cdot(Q1')\cdot(Q2' + Q0) \end{aligned}$$

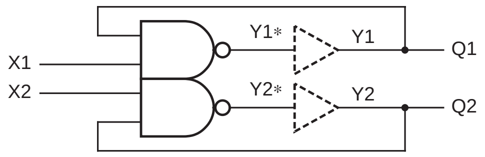
These equations yield the following transitions:

S	Q2	Q1	Q0	LEFT	RIGHT	HAZ	Q2*	Q1*	Q0*	S*
IDLE	0	0	0	0	0	0	0	0	0	IDLE
IDLE	0	0	0	0	1	1	1	0	0	LR3
IDLE	0	0	0	1	1	0	0	0	1	L1
IDLE	0	0	0	1	1	1	0	0	0	IDLE

This is obviously different and still not particularly good behavior.

7.58 Let E(SB), E(SC), and E(SD) be the binary encodings of states SB, SC, and SD respectively. Then $E(SD) = E(SB) + E(SC)$, the bit-by-bit logical OR of E(SB) and E(SC). This is true because the synthesis method uses the logical OR of the next values for each state variable and, by extension, the logical OR of the encoded states.

7.68 As far as I know, I was the first person to propose BUT-flops, and Glenn Trewitt was the first person to analyze them, in 1982. To analyze, we break the feedback loops as shown in the figure to the right.



The excitation and output equations are

$$\begin{aligned}
Y1 &= [(X1 \cdot Y1) \cdot (X2 \cdot Y2)']' \\
&= X1' + Y1' + X2 \cdot Y2 \\
Y2 &= [(X2 \cdot Y2) \cdot (X1 \cdot Y1)']' \\
&= X2' + Y2' + X1 \cdot Y1 \\
Q1 &= Y1 \\
Q2 &= Y2
\end{aligned}$$

The corresponding transition/state table is

	X1 X2			
Y1 Y2	00	01	11	10
00	11	11	11	11
01	11	10	10	11
11	$\boxed{11}$	10	$\boxed{11}$	01
10	11	11	01	01
	Y1* Y2*			

The two stable total states are circled. Notice that state 00 is unreachable.

When X1 X2 = 00 or 11, the circuit generally goes to stable state 11, with Q1 Q2 = 11. The apparent oscillation between states 01 and 10 when X1 X2 = 11 may not occur in practice, because it contains a critical race that tends to force the circuit into stable state 11.

When X1 X2 = 01 or 10, the Q output corresponding to the HIGH input will oscillate, while the other output remains HIGH .

Whether this circuit is useful is a matter of opinion.

7.71 When X=1, the circuit was supposed to "count" through its eight states in Gray-code order. When X=0, it remains in the current state. If this were the case, I suppose it could be used as a 3-bit random number generator. However, I messed up on the logic diagram and the circuit actually does something quite different and completely useless, compared to what I intended when I wrote the problem. Someday I'll fix this problem. Also, metastability may occur when X is changed from 1 to 0.

7.79 Figure X5.59 requires two "hops" for each input change. Figure 7–66 is faster, requiring only one hop for each input change. On the other hand, Figure 7–66 cannot be generalized for *n*>2.

7.90 Either this exercise is a joke, or a correct answer is much too dangerous to publish. Nevertheless, Earl Levine offers two possible answers:

(Stable output)	Was the last answer to this question "yes"?
(Oscillating output)	Was the last answer to this question "no"?