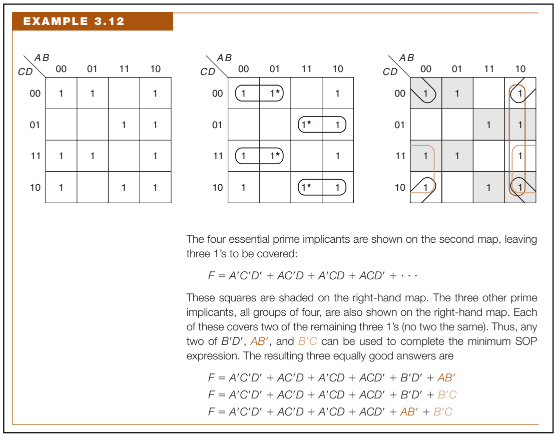
WALK THROUGH

Introduction to Logic Design is written with the student in mind. The focus is on the fundamentals and teaching by example. The author believes that the best way to learn logic design is to study and solve a large number of design problems, and that is what he gives students the opportunity to do. In keeping with the student focus, the following features contribute to this goal.

Examples Numerous easy-to-spot examples that help make concepts clear and understandable are integrated throughout each chapter.



EXAMPLE 3.12

CD \ AB	00	01	11	10
00	1	1		1
01			1	1
11	1	1		1
10	1		1	1

CD \ AB	00	01	11	10
00	1	1*		1
01			1*	1
11	1	1*		1
10	1		1*	1

CD \ AB	00	01	11	10
00	1	1		1
01			1	1
11	1	1		1
10	1		1	1

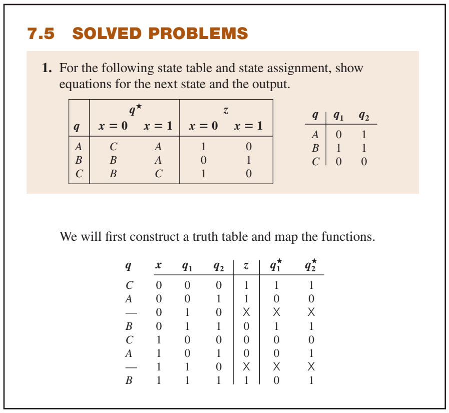
The four essential prime implicants are shown on the second map, leaving three 1's to be covered:

$$F = A'C'D' + AC'D + A'CD + ACD' + \cdots$$

These squares are shaded on the right-hand map. The three other prime implicants, all groups of four, are also shown on the right-hand map. Each of these covers two of the remaining three 1's (no two the same). Thus, any two of $B'D'$, AB', and $B'C$ can be used to complete the minimum SOP expression. The resulting three equally good answers are

$$F = A'C'D' + AC'D + A'CD + ACD' + B'D' + AB'$$
$$F = A'C'D' + AC'D + A'CD + ACD' + B'D' + B'C$$
$$F = A'C'D' + AC'D + A'CD + ACD' + AB' + B'C$$

Solved Problems A hallmark feature of this book, the extensive set of solved problems found at the end of every chapter gives students the advantage of seeing concepts applied to actual problems.



7.5 SOLVED PROBLEMS

1. For the following state table and state assignment, show equations for the next state and the output.

	$q^\star$		z	
q	$x = 0$	$x = 1$	$x = 0$	$x = 1$
A	C	A	1	0
B	B	A	0	1
C	B	C	1	0

q	q_1	q_2
A	0	1
B	1	1
C	0	0

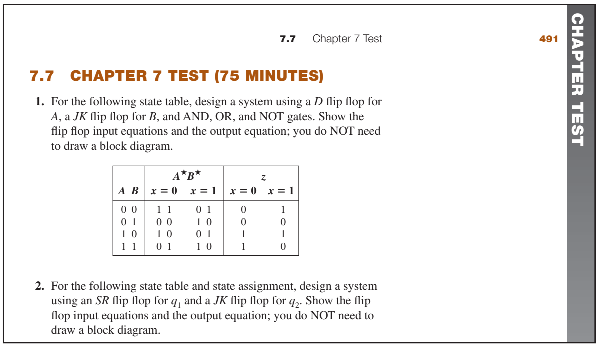
We will first construct a truth table and map the functions.

q	x	q_1	q_2	z	$q_1^\star$	$q_2^\star$
C	0	0	0	1	1	1
A	0	0	1	1	0	0
—	0	1	0	X	X	X
B	0	1	1	0	1	1
C	1	0	0	0	0	0
A	1	0	1	0	0	1
—	1	1	0	X	X	X
B	1	1	1	1	0	1

Color Color is used as a powerful pedagogical aid throughout.

Karnaugh Maps The liberal use of Karnaugh maps helps students grasp the basic principles of switching algebra.

End-of-Chapter Tests "Test Yourself" sections, also identifiable by a shaded bar, are designed to help students measure their comprehension of key material. Answers to tests can be found in Appendix C.



7.7 Chapter 7 Test **491**

CHAPTER TEST

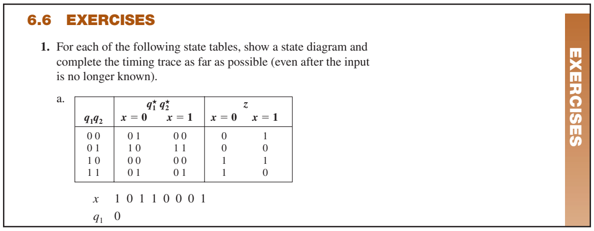
7.7 CHAPTER 7 TEST (75 MINUTES)

1. For the following state table, design a system using a *D* flip flop for *A*, a *JK* flip flop for *B*, and AND, OR, and NOT gates. Show the flip flop input equations and the output equation; you do NOT need to draw a block diagram.

	$A^\star B^\star$		z	
$A\ B$	$x = 0$	$x = 1$	$x = 0$	$x = 1$
0 0	1 1	0 1	0	1
0 1	0 0	1 0	0	0
1 0	1 0	0 1	1	1
1 1	0 1	1 0	1	0

2. For the following state table and state assignment, design a system using an *SR* flip flop for q_1 and a *JK* flip flop for q_2. Show the flip flop input equations and the output equation; you do NOT need to draw a block diagram.

Exercises Each chapter features a wide selection of exercises, identifiable by a colored bar, with selected answers in Appendix B.



6.6 EXERCISES

EXERCISES

1. For each of the following state tables, show a state diagram and complete the timing trace as far as possible (even after the input is no longer known).

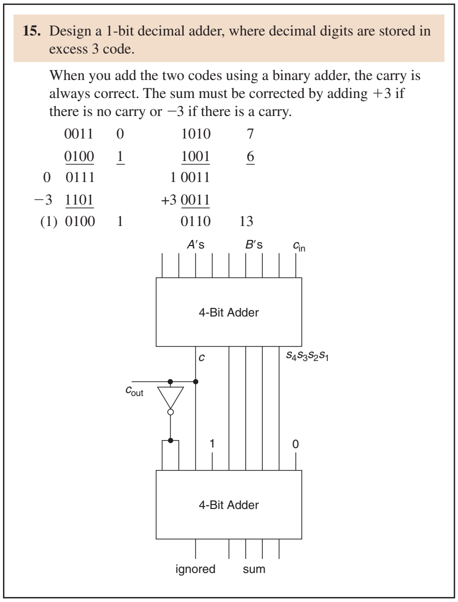
a.

	$q_1^\star\ q_2^\star$		z	
q_1q_2	$x = 0$	$x = 1$	$x = 0$	$x = 1$
0 0	0 1	0 0	0	1
0 1	1 0	1 1	0	0
1 0	0 0	0 0	1	1
1 1	0 1	0 1	1	0

x 1 0 1 1 0 0 0 1

q_1 0

Design Design using standard small- and medium-scale integrated circuit packages and programmable logic devices is a key aspect of the book.



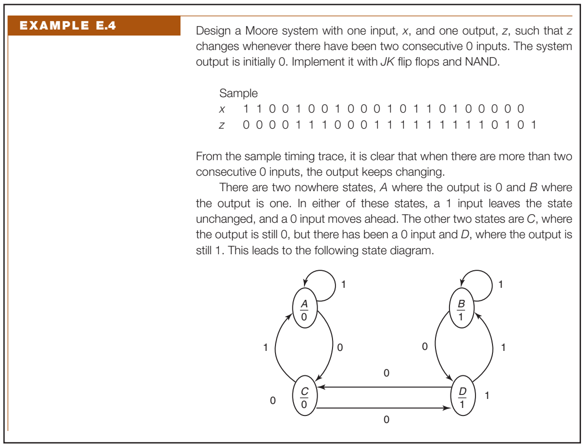
15. Design a 1-bit decimal adder, where decimal digits are stored in excess 3 code.

When you add the two codes using a binary adder, the carry is always correct. The sum must be corrected by adding +3 if there is no carry or −3 if there is a carry.

```
      0011   0        1010    7
      0100   1        1001    6
  0   0111           1 0011
 −3   1101          +3 0011
(1)   0100   1        0110   13
```

A's
B's
c_{in}
4-Bit Adder
c
$s_4s_3s_2s_1$
c_{out}
1
0
4-Bit Adder
ignored
sum

Complete Examples Marcovitz features six complete examples, from word problem to design, in Appendix E.



EXAMPLE E.4

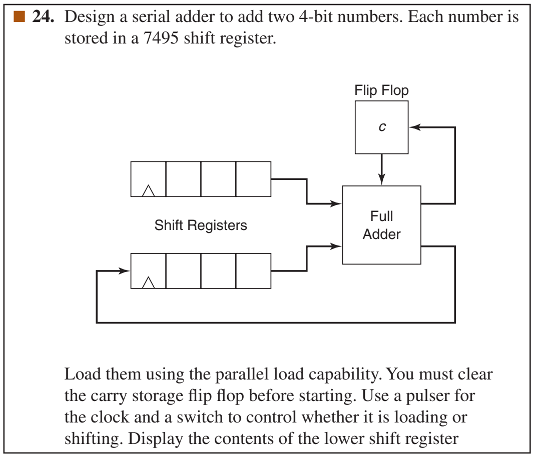
Design a Moore system with one input, *x*, and one output, *z*, such that *z* changes whenever there have been two consecutive 0 inputs. The system output is initially 0. Implement it with *JK* flip flops and NAND.

```
Sample
x    1 1 0 0 1 0 0 1 0 0 0 1 0 1 1 0 1 0 0 0 0 0
z    0 0 0 0 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 0 1 0 1
```

From the sample timing trace, it is clear that when there are more than two consecutive 0 inputs, the output keeps changing.

There are two nowhere states, *A* where the output is 0 and *B* where the output is one. In either of these states, a 1 input leaves the state unchanged, and a 0 input moves ahead. The other two states are *C*, where the output is still 0, but there has been a 0 input and *D*, where the output is still 1. This leads to the following state diagram.

Labs Four types of laboratory experiments help to integrate practical circuits with theory. Students can take advantage of traditional hands-on hardware experiments, experiments designed for WinBreadboard/MacBreadboard (a virtual breadboard), and simulation laboratory exercises using the circuit capture program LogicWorks.

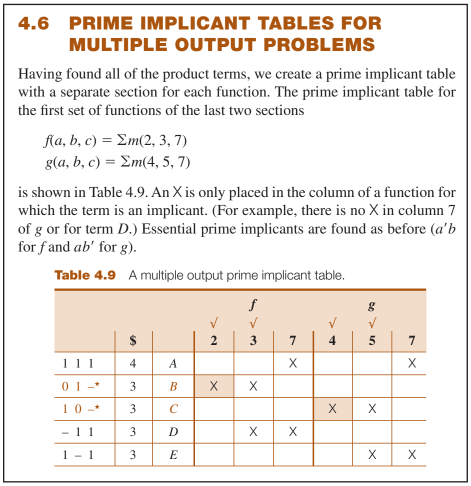


■ **24.** Design a serial adder to add two 4-bit numbers. Each number is stored in a 7495 shift register.

Flip Flop
c
Shift Registers
Full
Adder

Load them using the parallel load capability. You must clear the carry storage flip flop before starting. Use a pulser for the clock and a switch to control whether it is loading or shifting. Display the contents of the lower shift register

Multiple Output Problems Techniques for solving multiple output problems are shown using the Karnaugh map, Quine-McCluskey, and iterated consensus.



4.6 PRIME IMPLICANT TABLES FOR MULTIPLE OUTPUT PROBLEMS

Having found all of the product terms, we create a prime implicant table with a separate section for each function. The prime implicant table for the first set of functions of the last two sections

$$f(a, b, c) = \Sigma m(2, 3, 7)$$
$$g(a, b, c) = \Sigma m(4, 5, 7)$$

is shown in Table 4.9. An X is only placed in the column of a function for which the term is an implicant. (For example, there is no X in column 7 of *g* or for term *D*.) Essential prime implicants are found as before ($a'b$ for *f* and ab' for *g*).

Table 4.9 A multiple output prime implicant table.

				f			*g*	
			√	√		√	√	
	$		**2**	**3**	**7**	**4**	**5**	**7**
1 1 1	4	*A*			X			X
0 1 –*	3	*B*	X	X				
1 0 –*	3	*C*				X	X	
– 1 1	3	*D*		X	X			
1 – 1	3	*E*					X	X