

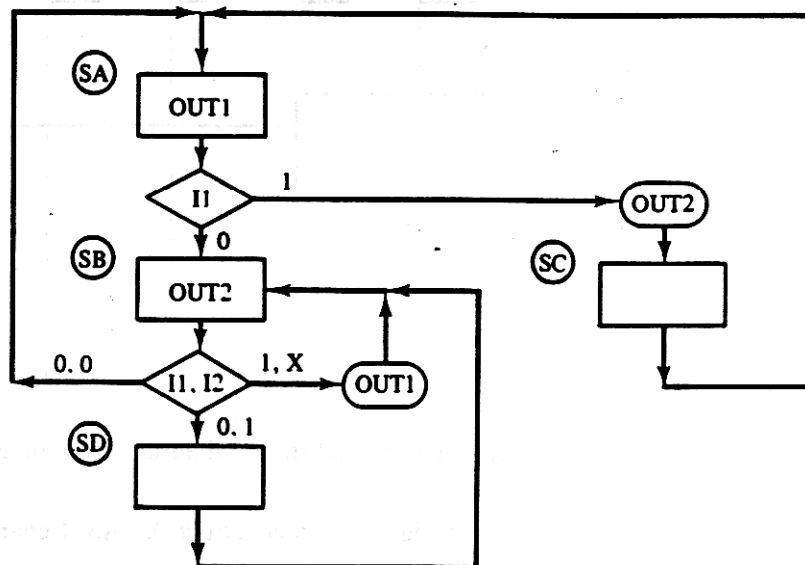
has been to present and illustrate the general principles of digital design and to consider various tools that are useful in the design process. These design principles and tools form a solid foundation on which to build experience and to exercise creativity.

SUPPLEMENTARY READING (see Bibliography)

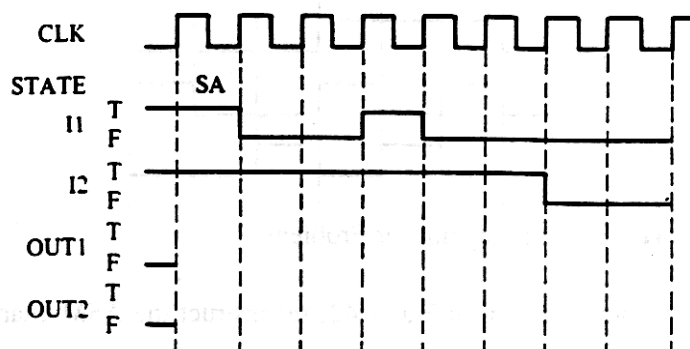
[Clare 73], [Fletcher 80], [Kline 83], [Mano 79], [Mano 84], [Peatman 80], [Prosser 87], [Wiatrowski 80]

PROBLEMS

- 7.1. Explain the general model for a digital circuit design shown in Fig. 7.1.
- 7.2. The digital design process can be divided into three major phases: the preliminary design phase, the refinement phase, and the realization phase. At the conclusion of each phase, what are the expected end products in terms of the controller and the controlled circuit elements?
- 7.3. Given the ASM chart of Fig. 7.39(a), complete the corresponding timing diagram of Fig. 7.39(b).



(a)



(b)

Figure 7.39 ASM chart and timing diagram for Problem 7.3.

7.4. Given the ASM chart and block diagram of Fig. 7.40(a), complete the corresponding timing diagram of Fig. 7.40(b).

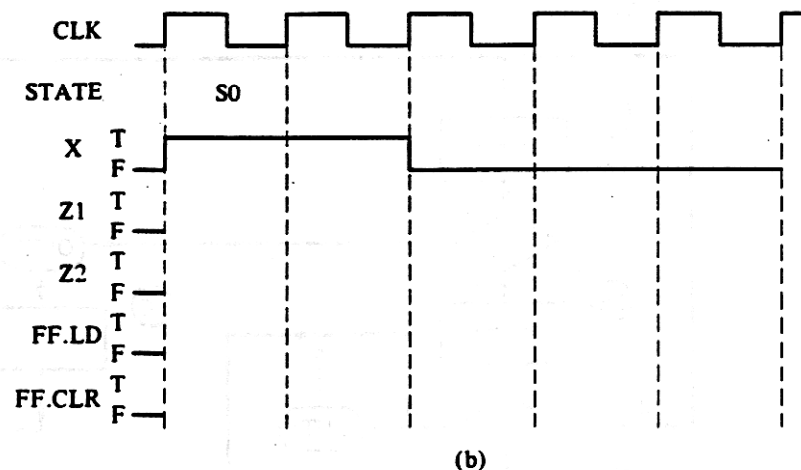
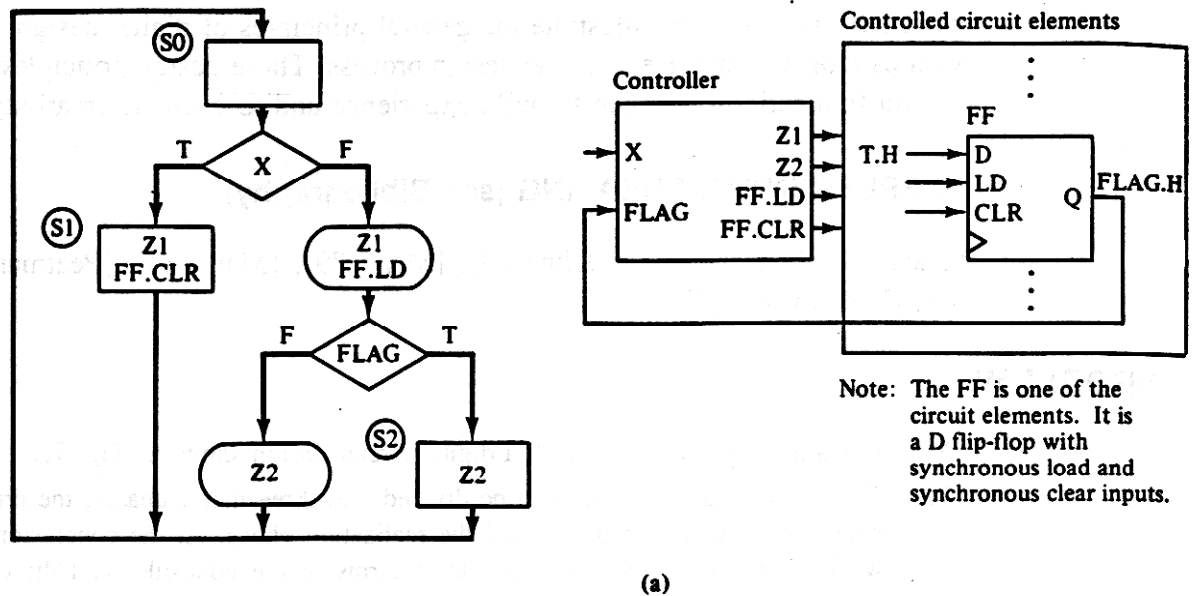


Figure 7.40 ASM chart, block diagram, and timing diagram for Problem 7.4.

7.5. Given the timing diagram of Fig. 7.41, reconstruct the ASM chart that corresponds to it.

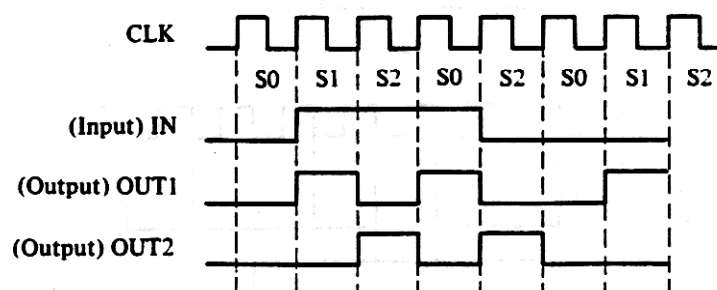


Figure 7.41 Timing diagram for Problem 7.5.

7.6. Given the timing diagram of Fig. 7.42, reconstruct the ASM chart that corre

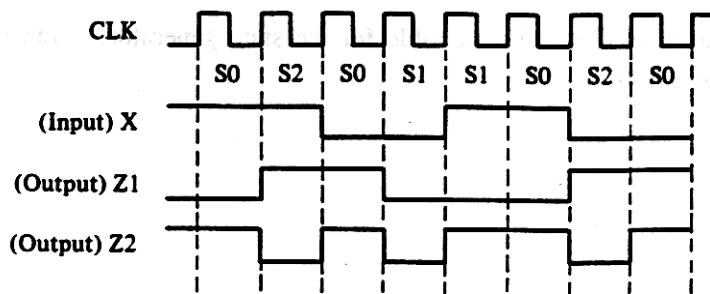


Figure 7.42 Timing diagram for Problem 7.6.

7.7. For the ASM chart specified in Fig. 7.40(a),

- Make a block diagram design (similar to the one shown in Fig. 7.3) of the corresponding state generator, specifying the inputs, outputs, and the state flip-flops (D type).
- Construct the next-state table for the state generator, given the following state code assignment:

	C1	C0
S0	1	0
S1	0	0
S2	0	1

- Realize the controller by the traditional method. Use D flip-flops, and draw a detailed circuit diagram of the controller.

7.8. For the ASM chart of Fig. 7.43,

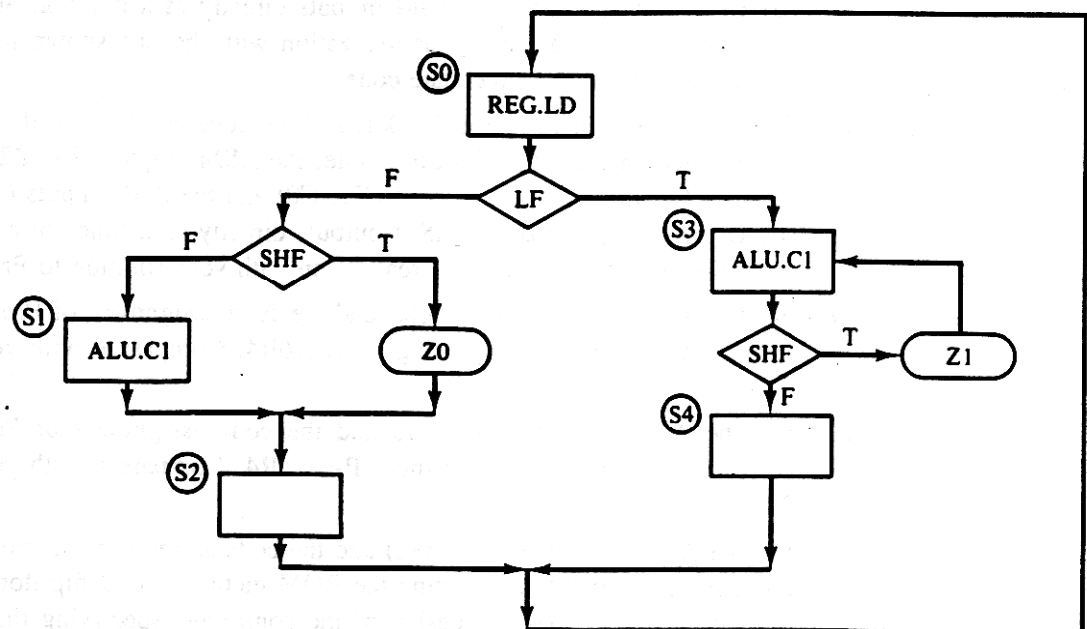


Figure 7.43 ASM chart for Problem 7.8.

- Make a block diagram of the corresponding controller, specifying the controller inputs and outputs.
- Make a block diagram design (similar to the one shown in Fig. 7.3) of the corresponding state generator, specifying the inputs, outputs, and the state flip-flops (D type).

- (c) Determine the next-state table for the state generator, given the following state code assignment:

	C2	C1	C0
S0	0	0	0
S1	0	0	1
S2	0	1	0
S3	0	1	1
S4	1	0	0

- (d) Realize the controller by the traditional method. Use D flip-flops, and draw a detailed circuit diagram of the controller.

7.9. Given the ASM chart of Fig. 7.4,

- (a) Specify the ASM outputs as a function of the state code and the ASM inputs. In other words, complete the following truth table:

C1	C0	IN. BIT	BUF. FULL	COUNT. EN	REG. LD	OUT. FLAG
0	0	0	0			
0	0	0	1			
0	0	1	0			
1	1	1	1			

- (b) Using gates, realize the ASM outputs directly as a function of the state code and the ASM inputs. Compare your realization with the one shown in Fig. 7.6, which has a circuit for decoding the state code.

7.10. Given the ASM chart of Fig. 7.40(a) and the code assignment of Problem 7.7(b),

- (a) Specify, in the form of a truth table, the ASM outputs (Z1, Z2, FF.LD, and FF.CLR) as a function of the state code (C1, C0) and the ASM inputs (X, FLAG).

- (b) Using gates, realize the ASM outputs directly as a function of the state code and the ASM inputs. Compare this realization with your solution to Problem 7.7.

7.11. For the ASM chart of Fig. 7.40(a) and the code assignment of Problem 7.7(b), realize the corresponding controller by using a PAL16R4. Compare it with your solution to Problem 7.7.

7.12. For the ASM chart of Fig. 7.43 and the code assignment of Problem 7.8, realize the corresponding controller by using a PAL16R4. Compare it with your solution to Problem 7.8.

7.13. For the ASM chart of Fig. 7.40(a) and the code assignment of Problem 7.7(b), realize the corresponding controller by using the ROM method and D flip-flops. That is,

- (a) Make a block diagram design of the controller, specifying the state flip-flops and all the appropriate connections to the inputs and outputs of the ROM. In other words, make a block diagram design similar to the one shown in Fig. 7.8, but without the actual ROM contents.

- (b) Specify the ROM contents in hexadecimal.

7.14. Repeat Problem 7.13 for the ASM chart of Fig. 7.43 and the code assignment of Problem 7.8(c).

- 7.15. Figure 7.44 shows a block diagram design of a controller based on the ROM method and with D flip-flops. The ROM contents are also given. Derive the corresponding ASM chart.

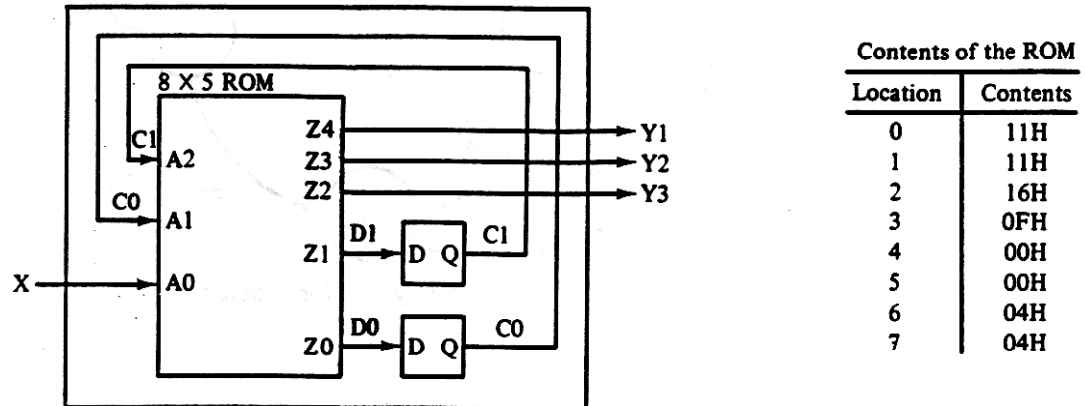


Figure 7.44 Controller block diagram and ROM contents for Problem 7.15.

- 7.16. For the ASM chart of Fig. 7.39(a),
- Draw a block diagram of the corresponding controller, specifying the controller inputs and outputs.
 - Make a block diagram design (similar to the one of Fig. 7.10) for the corresponding state generator, specifying the inputs, outputs, and the state flip-flops (J-K type).
 - Determine the next-state table for the state generator, given the following state code assignment:

	C1	C0
SA	0	0
SB	0	1
SC	1	0
SD	1	1

- Realize the controller by the traditional method. Use J-K flip-flops, and draw a detailed circuit diagram of the controller.
- 7.17. For the ASM chart of Fig. 7.39(a) and the state code assignment of Problem 7.16(c), realize the corresponding controller by using the ROM method and J-K flip-flops: that is,
- Make a block diagram design of the controller, specifying the state flip-flops and all the appropriate connections to the ROM inputs and outputs.
 - Specify the ROM contents in hexadecimal.
- 7.18. Convert the Mealy state graph of Fig. 7.45 into an equivalent ASM chart. The inputs are X1 and X2, and the outputs are Z1 and Z2.
- 7.19. Convert the Moore state graph of Fig. 7.46 into an equivalent ASM chart. The inputs are X1 and X2, and the outputs are Z1, Z2, and Z3.
- 7.20. Assume that the design of the dynamic RAM controller circuit described in Sec. 7.8.1 is to be modified. In addition to the already specified functions, if the RD and WR signals are both true, then the function of the circuit is to be described by the timing diagram of Fig. 7.47. Specifically,

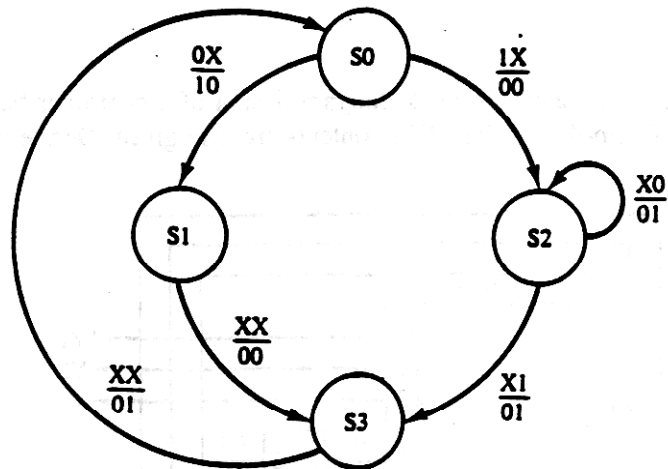


Figure 7.45 Mealy state graph for Problem 7.18.

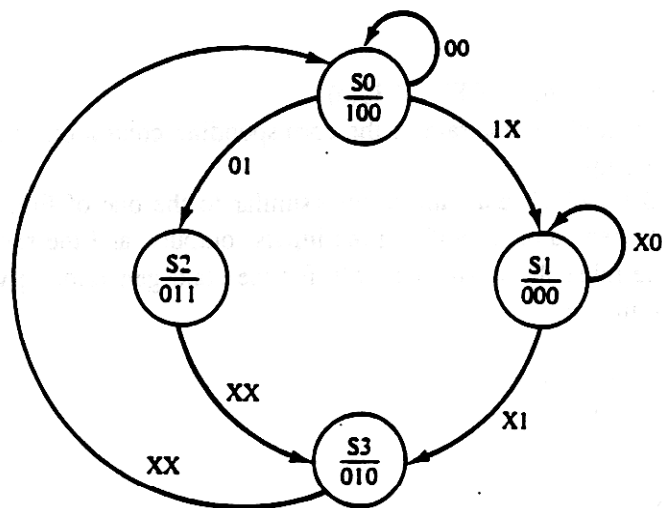


Figure 7.46 Moore state graph for Problem 7.19.

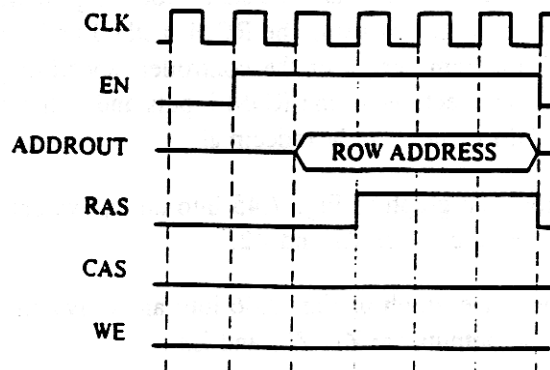


Figure 7.47 Timing diagram for Problem 7.20.

- (a) Modify the ASM chart of Fig. 7.20 to include this function.
- (b) Realize this modified ASM chart by using the ROM method and D flip-flops.

7.21. An 8-bit parallel-to-serial converter circuit, the block diagram of which is shown in Fig. 7.48, is to be designed and realized. This circuit remains in an idle state as long as the START input is false. But when this START input becomes true, the 8-bit data BYTE is loaded into the shift register and the right-shifting of the data begins. After the 8 bits are shifted out, the circuit returns to the idle state.

- (a) The major circuit elements for the parallel-to-serial converter circuit are given in Fig. 7.48. Make any reasonable assumptions for the circuit elements that are required and derive the ASM chart for the controller. (*Hint: A two-state ASM chart can be derived for this circuit.*)
- (b) Realize the circuit elements with commercially available chips.
- (c) Realize the ASM chart by using any of the methods presented in this chapter.

7.22. A lock to a certain safe can be opened with a correct combination or with a key. If the combination feature is used, then the lock must be supplied with the correct combination within three attempts. Otherwise an alarm will sound. Specifications for the lock circuit are as follows:

1. After the first unsuccessful attempt, output message 1 (MSG1).
2. After the second unsuccessful attempt,
 - (a) output message 2 (MSG2), and
 - (b) wait for 10 seconds before being ready to be tried again (READY).
3. After the third successive unsuccessful attempt, sound an alarm.
4. When the safe is opened, reset to allow three more tries.
5. When the key is used, stop the alarm and reset to allow three more tries.

Your part in this problem is to design the module of Fig. 7.49. Make a detailed design of this module, using a counter for the major circuit element. Realize the controller with the ROM method and D flip-flops.

7.23. Shown in Fig. 7.50(a) are the circuit elements for a digital circuit that functions as follows:

- If $OP = 00$, then $REGB \leftarrow REGA$; followed by $REGA \leftarrow IN + 1$.
- If $OP = 01$, then $REGB \leftarrow 0$; $REGA \leftarrow REGA + 1$.
- If $OP = 10$, $REGA \leftarrow IN$; $REGB \leftarrow IN$; followed by $REGB \leftarrow REGB + 1$.
- If $OP = 11$, then the contents of $REGA$ are doubled; $REGB \leftarrow REGB + 1$.

Notation: An arrow (\leftarrow) designates to load the data at the next active clock transition. For example, $REGA \leftarrow IN + 1$ designates to add 1 to the value of IN and load the sum into $REGA$ at the next active clock transition. In the function statements, more than one operation can be performed within a single state unless otherwise stated, as designated by "followed by."

The flowchart for the controller is given in Fig. 7.50(b). Derive the corresponding ASM chart for it. Optimize it by using the least number of states. In other words, if more than one operation can be performed within a single state, then do so. Also, make use of conditional outputs.