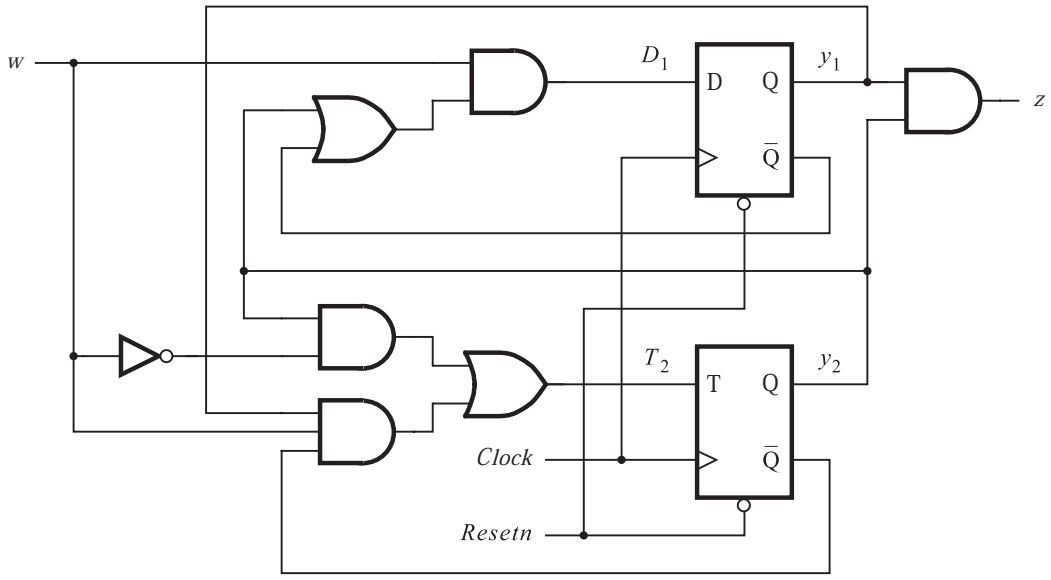

Figure 8.84 Circuit for Example 8.10.

Present state y_2y_1	Flip-flop inputs $w = 0$ T_2D_1	Flip-flop inputs $w = 1$ T_2D_1	Output z
00	00	01	0
01	00	10	0
10	10	01	0
11	10	01	1

Figure 8.85 The excitation table for the circuit in Figure 8.84.

8.10 ALGORITHMIC STATE MACHINE (ASM) CHARTS

The state diagrams and tables used in this chapter are convenient for describing the behavior of FSMs that have only a few inputs and outputs. For larger machines the designers often use a different form of representation, called the *algorithmic state machine (ASM) chart*.

An ASM chart is a type of flowchart that can be used to represent the state transitions and generated outputs for an FSM. The three types of elements used in ASM charts are depicted in Figure 8.86.

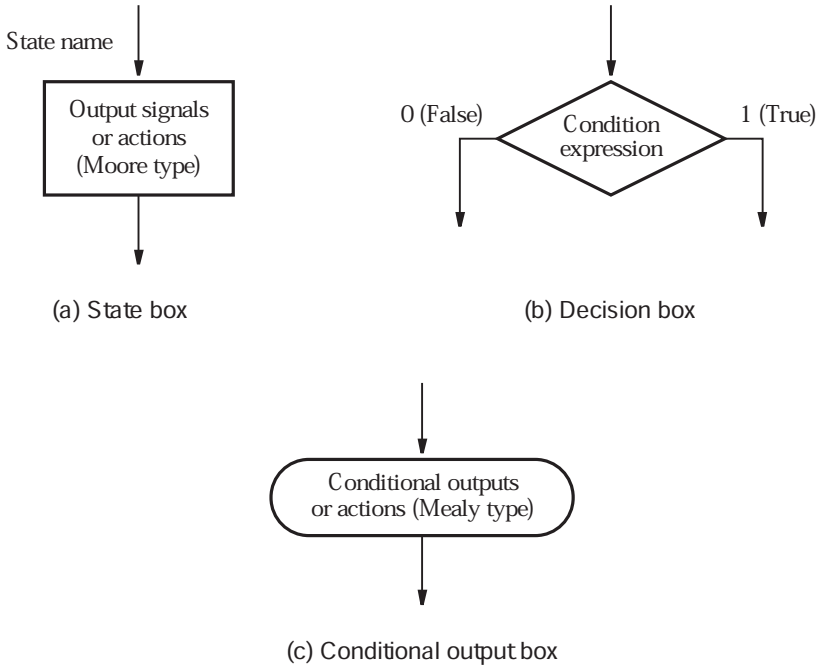

Figure 8.86 Elements used in ASM charts.

- **State box** – A rectangle represents a state of the FSM. It is equivalent to a node in the state diagram or a row in the state table. The name of the state is indicated outside the box in the top-left corner. The Moore-type outputs are listed inside the box. These are the outputs that depend only on the values of the state variables that define the state; we will refer to them simply as *Moore outputs*. It is customary to write only the name of the signal that has to be asserted. Thus it is sufficient to write z, rather than $z = 1$, to indicate that the output z must have the value 1. Also, it may be useful to indicate an action that must be taken; for example, $Count \leftarrow Count + 1$ specifies that the contents of a counter have to be incremented by 1. Of course, this is just a simple way of saying that the control signal that causes the counter to be incremented must be asserted. We will use this way of specifying actions in larger systems that are discussed in Chapter 10.
- **Decision box** – A diamond indicates that the stated condition expression is to be tested and the exit path is to be chosen accordingly. The condition expression consists of one or more inputs to the FSM. For example, w indicates that the decision is based on the value of the input w, whereas $w_1 \cdot w_2$ indicates that the true path is taken if $w_1 = w_2 = 1$ and the false path is taken otherwise.
- **Conditional output box** – An oval denotes the output signals that are of Mealy type. These outputs depend on the values of the state variables and the inputs of the FSM; we will refer to these outputs simply as *Mealy outputs*. The condition that determines whether such outputs are generated is specified in a decision box.

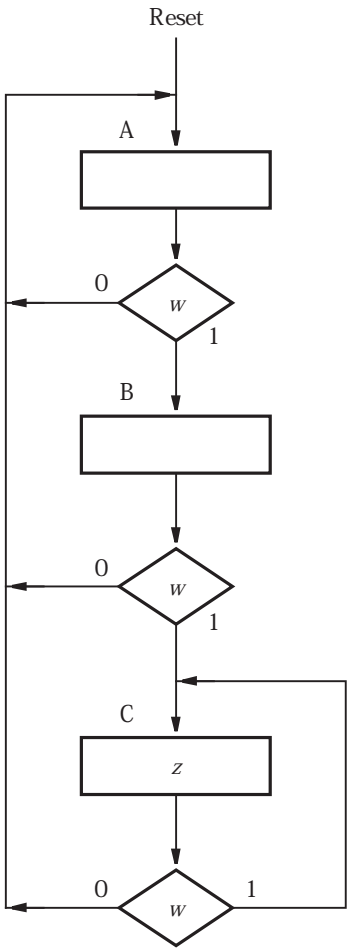

Figure 8.87 ASM chart for the FSM in Figure 8.3.

Figure 8.87 gives the ASM chart that represents the FSM in Figure 8.3. The transitions between state boxes depend on the decisions made by testing the value of the input variable w. In each case if $w = 0$, the exit path from a decision box leads to state A. If $w = 1$, then a transition from A to B or from B to C takes place. If $w = 1$ in state C, then the FSM stays in that state. The chart specifies a Moore output z, which is asserted only in state C, as indicated in the state box. In states A and B, the value of z is 0 (not asserted), which is implied by leaving the corresponding state boxes blank.

Figure 8.88 provides an example with Mealy outputs. This chart represents the FSM in Figure 8.23. The output, z, is equal to 1 when the machine is in state B and $w = 1$. This is indicated using the conditional output box. In all other cases the value of z is 0, which is implied by not specifying z as an output of state B for $w = 0$ and state A for w equal to 0 or 1.

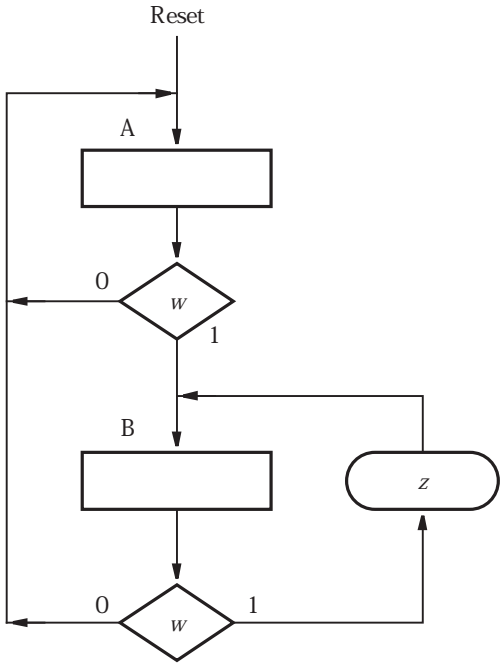

Figure 8.88 ASM chart for the FSM in Figure 8.23.

Figure 8.89 gives the ASM chart for the arbiter FSM in Figure 8.73. The decision box drawn below the state box for *Idle* specifies that if $r_1 = 1$, then the FSM changes to state *gnt1*. In this state the FSM asserts the output signal g_1. The decision box to the right of the state box for *gnt1* specifies that as long as $r_1 = 1$, the machine stays in state *gnt1*, and when $r_1 = 0$, it changes to state *Idle*. The decision box labeled r_2 that is drawn below the state box for *Idle* specifies that if $r_2 = 1$, then the FSM changes to state *gnt2*. This decision box can be reached only after first checking the value of r_1 and following the arrow that corresponds to $r_1 = 0$. Similarly, the decision box labeled r_3 can be reached only if both r_1 and r_2 have the value 0. Hence the ASM chart describes the required priority scheme for the arbiter.

ASM charts are similar to traditional flowcharts. Unlike a traditional flowchart, the ASM chart includes timing information because it implicitly specifies that the FSM changes (flows) from one state to another only after each active clock edge. The examples of ASM charts presented here are quite simple. We have used them to introduce the ASM chart terminology by giving examples of state, decision, and conditional-output boxes. Another term sometimes applied to ASM charts is *ASM block*, which refers to a single state box and any decision and conditional-output boxes that the state box may be connected to. The ASM charts can be used to describe complex circuits that include one or more finite state machines and other circuitry such as registers, shift registers, counters, adders, and multipliers. We will use ASM charts as an aid for designing more complex circuits in Chapter 10.

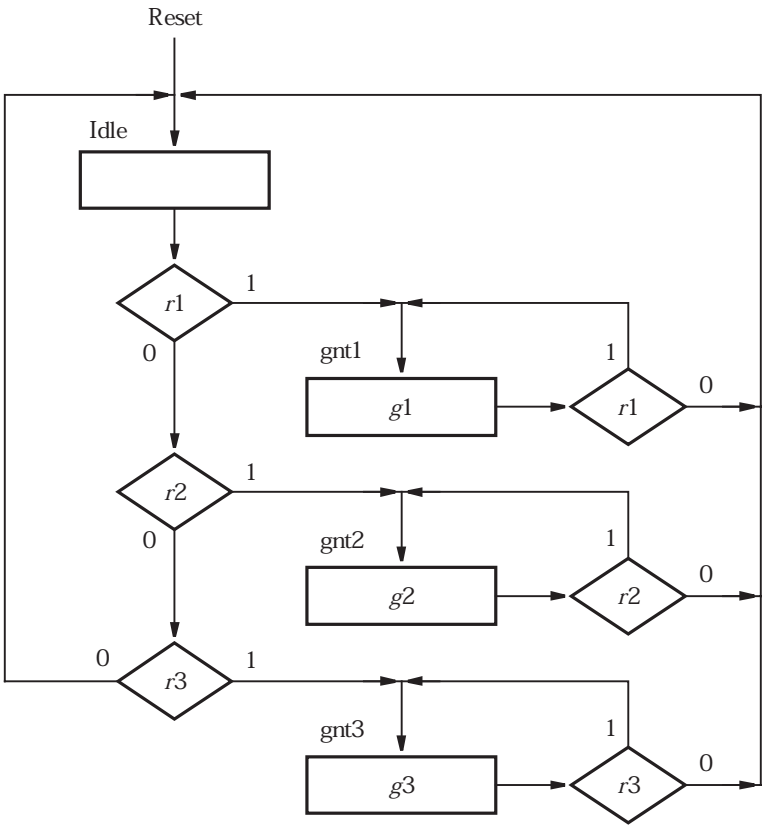

Figure 8.89 ASM chart for the arbiter FSM in Figure 8.73.

8.11 FORMAL MODEL FOR SEQUENTIAL CIRCUITS

This chapter has presented the synchronous sequential circuits using a rather informal approach because this is the easiest way to grasp the concepts that are essential in designing such circuits. The same topics can also be presented in a more formal manner, which has been the style adopted in many books that emphasize the switching theory aspects rather than the design using CAD tools. A formal model often gives a concise specification that is difficult to match in a more descriptive presentation. In this section we will describe a formal model that represents a general class of sequential circuits, including those of the synchronous type.

Figure 8.90 represents a general sequential circuit. The circuit has $W = \{w_1, w_2, \ldots, w_n\}$ inputs, $Z = \{z_1, z_2, \ldots, z_m\}$ outputs, $y = \{y_1, y_2, \ldots, y_k\}$ present-state variables, and $Y = \{Y_1, Y_2, \ldots, Y_k\}$ next-state variables. It can have up to 2^k states, $S = \{S_1, S_2, \ldots, S_{2^k}\}$. There are delay elements in the feedback paths for the state-variables which ensure that y