5 Regular (Category 1) State Machines

5.1 Introduction

We know that, from a hardware perspective, state machines can be classified into two types, based on their *input connections*, as follows.

1) *Moore machines:* The input, if it exists, is connected only to the logic block that computes the next state.
2) *Mealy machines:* The input is connected to both logic blocks, that is, for the next state and for the actual output.

In Section 3.6 we introduced a new classification, also from a hardware point of view, based on the *transition types* and *nature of the outputs*, as follows (see figure 5.1).

1) *Regular (category 1) state machines:* This category, illustrated in figure 5.1a and studied in chapters 5 to 7, consists of machines with only untimed transitions and outputs that do not depend on previous (past) output values.
2) *Timed (category 2) state machines:* This category, illustrated in figure 5.1b and studied in chapters 8 to 10, consists of machines with one or more transitions that depend on time (so they can have all four transition types: conditional, timed, conditional-timed, and unconditional). However, all outputs are still independent from previous (past) output values.
3) *Recursive (category 3) state machines:* This category is illustrated in figure 5.1c and studied in chapters 11 to 13. It can have all four types of transitions, but one or more outputs depend on previous (past) output values. Recall that the outputs are produced by the FSM's *combinational* logic block, so the current output values are "forgotten" after the machine leaves that state; consequently, to implement a recursive (recurrent) machine, some sort of extra memory is needed.

As seen in this and in upcoming chapters, the classifications mentioned above (no other classification is needed) will immensely ease the design of hardware-based

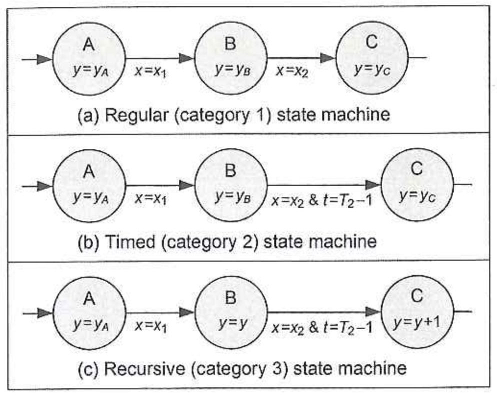


Figure 5.1
State machine categories (from a hardware perspective).

state machines. The two fundamental decisions before starting a design are then the following:

1) Decide the state machine category (regular, timed, or recursive).
2) Next, decide the state machine type (Moore or Mealy).

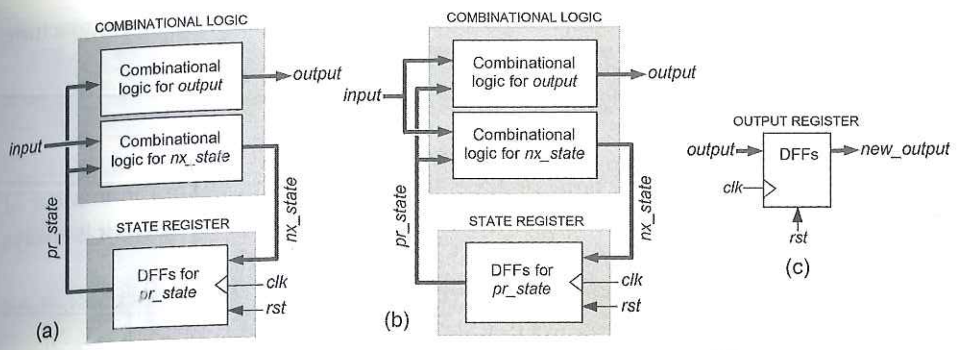
It is important to recall, however, that regardless of the machine category and type, the state transition diagram must fulfill three fundamental requisites (seen in section 1.3):

1) It must include all possible system states.
2) All state transition conditions must be specified (unless a transition is unconditional) and must be truly complementary.
3) The list of outputs must be exactly the same in all states (standard architecture).

5.2 Architectures for Regular (Category 1) Machines

The architectures for category 1 machines are summarized in figure 5.2. These representations follow the style of figures 3.1b,d, but the style of figures 3.1a,c could be used equivalently. The output register (figure 5.2c) is optional. The four possible constructions, listed in figure 5.2d, are summarized below.

Regular Moore machine (figure 5.2a): In this case, the input (if it exists) is connected only to the logic block for the next state. Consequently, the output depends only on the state in which the machine is (in other words, for each state, the output value in unique), resulting a synchronous behavior (see details in section 3.5). Because modern designs are generally synchronous, this implementation is preferred whenever the application permits.



(d)

Construction	Resulting category 1 circuit	Behavior
(a) only	Pure Moore machine (preferred)	Order-1 synchronous
(b) only	Pure Mealy machine	Asynchronous
(a) + (c)	Out-registered (pipelined) Moore machine	Order-2 synchronous
(b) + (c)	Out-registered (pipelined) Mealy machine	Order-1 synchronous

Figure 5.2
Regular (category 1) state machine architectures for (a) Moore and (b) Mealy types. (c) Optional output register. (d) Resulting circuits.

Regular Mealy machine (figure 5.2b): In this case, the input is connected to both logic blocks, so it can affect the output directly, resulting an asynchronous behavior. Therefore, the machine can have more than one output value for the same state (section 3.5).

Out-registered (pipelined) Moore machine: This consists of connecting the register of figure 5.2c to the output of the Moore machine of figure 5.2a. As seen in sections 2.5 and 2.6, two fundamental reasons for doing so are glitch removal and pipelined construction. As a result, the final circuit's output will be delayed with respect to the original machine's output by either one clock period (if the same clock edge is employed in the state register and in the output register) or by one-half of a clock period (if different clock edges are used). Note that the resulting circuit is order-2 synchronous because the original Moore machine was already a registered circuit (in other words, the input–output transfer occurs after two clock edges—see details in section 3.5). If in a given application this extra register is needed but its consequent extra delay is not acceptable, the next alternative can be used.

Out-registered (pipelined) Mealy machine: This consists of connecting the register of figure 5.2c to the output of the Mealy machine of figure 5.2b. The reasons for doing so are the same as for Moore machines. The resulting circuit is order-1 synchronous because the original Mealy machine is asynchronous. Consequently, the overall