**Design Procedure — 3.3 in Text**

Below is a seven step design process for state machines.

1. Define the state machine requirements
2. State diagram
3. State assignment
4. State table
5. Minimizations
6. State Equations
7. Logic Diagram

An example of a state machine could be a bidirectional stepper motor controller. A stepper motor controller is used to the timing and control logic necessary to rotate a stepper motor clockwise or counter clockwise.

1. Define the state machine requirements
   There is one input to the stepper motor controller, a single bit that controls the direction of the stepper motor controller. If Dir = 1 the motor steps clockwise, but if Dir = 0 the motor steps counter-clockwise. The clock input will be 1 KHz. There will be four states, for moving the stepper to A, B, C, or D. There are four outputs, W, X, Y, and Z, control currents going through the stepper motor coils. Figure 1 shows the general template for organizing the details of the state machine. Figures 2, 3, and 4 all show how W, X, Y, and Z can be manipulated to control the stepper motor positions.
Figure 1: A typical template for a Moore State Machine.

Figure 2: Setting $W = 1$ and $X = 0$ causes current to flow through coils in the stepper motor, which creates a magnetic field to move the stepper motor to position A.

Figure 3: Setting $Y = 1$ and $Z = 0$ causes current to flow through coils in the stepper motor, which creates a magnetic field to move the stepper motor clockwise from position A to position B.

Figure 4: A stepper motor has more than four steps per revolution, most have 200 steps per revolution.

2. State diagram
   Translate the information in step 1 into a state diagram as shown in figure 5.
3. State assignment
   Assign each state in the state diagram a binary representation.

<table>
<thead>
<tr>
<th>State</th>
<th>BinaryValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>00</td>
</tr>
<tr>
<td>B</td>
<td>01</td>
</tr>
<tr>
<td>C</td>
<td>10</td>
</tr>
<tr>
<td>D</td>
<td>11</td>
</tr>
</tbody>
</table>

4. State table

<table>
<thead>
<tr>
<th>State</th>
<th>BinaryValue</th>
<th>Input</th>
<th>Dir</th>
<th>NextState</th>
<th>NextStateBinaryValue, $Q_1^<em>, Q_0^</em>$</th>
<th>WXYZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>00</td>
<td>CCW</td>
<td>0</td>
<td>D</td>
<td>11, 1000</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>00</td>
<td>CW</td>
<td>1</td>
<td>B</td>
<td>01, 1000</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>01</td>
<td>CCW</td>
<td>0</td>
<td>A</td>
<td>00, 0010</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>01</td>
<td>CW</td>
<td>1</td>
<td>C</td>
<td>10, 0010</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>10</td>
<td>CCW</td>
<td>0</td>
<td>B</td>
<td>01, 0100</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>10</td>
<td>CW</td>
<td>1</td>
<td>D</td>
<td>11, 0100</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>11</td>
<td>CCW</td>
<td>0</td>
<td>C</td>
<td>10, 0001</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>11</td>
<td>CW</td>
<td>1</td>
<td>A</td>
<td>00, 0001</td>
<td></td>
</tr>
</tbody>
</table>

5. Minimizations
   Find and minimize equations for $Q_1^*$, $Q_0^*$, W, X, Y, and Z. That will be 6 three variable K-maps. The inputs are $Q_1$, $Q_0$, and Dir.

6. State Equations
   Write down the 6 State Equations.

7. Logic Diagram
   Put the logic gates for $Q_1^*$ and $Q_0^*$ into the next state logic of figure 5. These outputs will go into the D inputs of 2 D flip flops. The logic gates for W, X, Y, and Z go into the output logic of 6.
Figure 6: A typical template for a Moore State Machine.

Figure 7: A Moore State Machine for controlling a stepper motor.

Figure 8: A simulation of the stepper motor state machine.