ECE 331 – Digital System Design

Derivation of State Graphs and State Tables

(Lecture #22)

The slides included herein were taken from the materials accompanying Fundamentals of Logic Design, 6th Edition, by Roth and Kinney, and were used with permission from Cengage Learning.
Sequential Circuit Design

1. Understand specifications
2. Draw state graph (to describe state machine behavior)
3. Construct state table (from state graph)
4. Perform state reduction (if necessary)
5. Encode states (aka. state assignment)
6. Create state-assigned table
7. Select type of Flip-Flop to use
8. Derive Flip-Flop input equations and FSM output equation(s)
9. Draw logic diagram
 FSM Design: Mealy

Example: Design a sequence detector.

The circuit is of the form:
Example: A sequence detector (Mealy)

Suppose we want to design the sequence detector so that any input sequence ending in 010 will produce an output of $Z = 1$ coincident with the last 0.

The circuit does not reset when a 1 output occurs.

A typical input sequence and the corresponding output sequence are:

\[
\begin{align*}
X &= 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \\
Z &= 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0
\end{align*}
\]

(time: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15)
Example: A sequence detector (Mealy)

Initially, we do not know how many flip-flops will be required, so we will designate the circuit states as $S_0$, $S_1$, etc.

We will start with a reset state designated $S_0$. 
Example: A sequence detector (Mealy)

State Graph for the Mealy Machine
Example: A sequence detector (Mealy)

Convert the state graph to a state table:

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next State $X = 0$</th>
<th>Next State $X = 1$</th>
<th>Present Output $X = 0$</th>
<th>Present Output $X = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

How many Flip-Flops are required for this sequential logic circuit?
Example: A sequence detector (Mealy)

Convert the state table to a transition table:

<table>
<thead>
<tr>
<th>AB</th>
<th>$A^+B^+$</th>
<th>$Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$X = 0$</td>
<td>$X = 0$</td>
</tr>
<tr>
<td>0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

What about $AB = 11$?
Example: A sequence detector (Mealy)

From the state transition table, plot the next-state maps for the flip-flops and the map for the output function $Z$:

\[
\begin{align*}
A^+ &= \ & \ & X \\
B^+ &= \ & \ & X \\
Z &= \ & \ & X
\end{align*}
\]
Example: A sequence detector (Mealy)

Using the derived equations, draw the corresponding circuit diagram:
Example: Design a sequence detector.

The circuit (again) is of the form:

```
X  ->  Clock  ->  Z

serial bit stream (input)  output (serial bit stream)
```
Example: A sequence detector (Moore)

The sequential logic circuit (aka. FSM) should produce an output (Z) of a 1 only for an input sequence (X) ending in 010. The circuit does not reset when a 1 output occurs.

A typical input sequence and the corresponding output sequence are:

\[ X = 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ ]

\[ Z = 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ ]

(time: 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15)
Example: A sequence detector (Moore)

State Graph for the Moore Machine
Example: A sequence detector (Moore)

Convert the state graph to a state table:

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next State $X = 0$</th>
<th>Next State $X = 1$</th>
<th>Present Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example: A sequence detector (Moore)

Convert the state table to a transition table:

<table>
<thead>
<tr>
<th>AB</th>
<th>X = 0</th>
<th>X = 1</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example: A sequence detector (Moore)

From the state transition table, plot the next-state maps for the flip-flops and the map for the output function $Z$:

$$A^+ = \begin{array}{c|c|c|c|c|c|c|c|c} \hline A & B & X & \hline \hline A & X & B & X \end{array}$$

$$B^+ = \begin{array}{c|c|c|c|c|c|c|c|c} \hline A & B & X & \hline \hline A & X & B & X \end{array}$$

$$Z = \begin{array}{c|c|c|c|c|c|c|c|c} \hline A & B & X & \hline \hline A & X & B & X \end{array}$$
Example: A sequence detector (Moore)

Using the derived equations, draw the corresponding circuit diagram:
Example: Design a more complex sequence detector.

The circuit (again) is of the form:
Example: Complex sequence detector (Moore)

The sequential logic circuit (aka. FSM) should produce an output (Z) of a 1 for an input sequence (X) ending in either 010 or 1001; the output (Z) should be 0 otherwise.

The circuit does not reset when a 1 output occurs.

A typical input sequence and the corresponding output sequence are:

\[
X = \begin{array}{cccccccccccccccc}
0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0
\end{array}
\]

\[
Z = \begin{array}{cccccccccccccccc}
0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0
\end{array}
\]

(time: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
Example: Complex sequence detector (Moore)
Example: Complex sequence detector (Mealy)

The state graph for the equivalent Mealy machine is derived in the textbook.
FSM Design: Moore

Example:

Another Moore Finite State Machine.
Example: Another FSM (Moore)

Design a Moore sequential circuit with one input $X$ and one output $Z$. The output $Z$ is to be 1 if the total number of 1’s received is odd and at least two consecutive 0’s have been received. A typical input and output sequence is:

$$X = \begin{array}{cccccc}
1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
\uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \\
\text{a} & \text{b} & \text{c} & \text{d} & \text{e} \\
\end{array}$$

$$Z = (0) 0 0 0 0 0 0 1 0 1$$
Example: Another FSM (Moore)
FSM Design: Mealy

Example:

Another *Mealy* Finite State Machine.
Example: Another FSM (Mealy)

A sequential circuit has one input ($X$) and one output ($Z$). The circuit examines groups of four consecutive inputs and produces an output $Z = 1$ if the input sequence 0101 or 1001 occurs. The circuit resets after every four inputs. Find a Mealy state graph. A typical input and output sequence is:

\[
\begin{align*}
X &= 0101 | 0010 | 1001 | 0100 \\
Z &= 0001 | 0000 | 0001 | 0000
\end{align*}
\]
Example: Another FSM (Mealy)
Constructing State Graphs

A set of guidelines for constructing state graphs is provided in the textbook.
Example: Multiple Inputs (Mealy)

A sequential circuit has two inputs ($X_1$, $X_2$) and one output ($Z$). The output remains a constant value unless one of the following input sequences occurs:
(a) The input sequence $X_1X_2 = 01, 11$ causes the output to become 0.
(b) The input sequence $X_1X_2 = 10, 11$ causes the output to become 1.
(c) The input sequence $X_1X_2 = 10, 01$ causes the output to change value.

(The notation $X_1X_2 = 01, 11$ means $X_1 = 0, X_2 = 1$ followed by $X_1 = 1, \ X_2 = 1$.)
### Example: Multiple Inputs (Mealy)

<table>
<thead>
<tr>
<th>Previous Input ($X_1X_2$)</th>
<th>Output ($Z$)</th>
<th>State Designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 or 11</td>
<td>0</td>
<td>$S_0$</td>
</tr>
<tr>
<td>00 or 11</td>
<td>1</td>
<td>$S_1$</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
<td>$S_2$</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>$S_3$</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>$S_4$</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>$S_5$</td>
</tr>
</tbody>
</table>
Example: Multiple Inputs (Mealy)

The state table for the Moore machine:

<table>
<thead>
<tr>
<th>Present State</th>
<th>Z</th>
<th>$X_1X_2 = 00$</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>0</td>
<td>$S_0$</td>
<td>$S_2$</td>
<td>$S_0$</td>
<td>$S_4$</td>
</tr>
<tr>
<td>$S_1$</td>
<td>1</td>
<td>$S_1$</td>
<td>$S_3$</td>
<td>$S_1$</td>
<td>$S_5$</td>
</tr>
<tr>
<td>$S_2$</td>
<td>0</td>
<td>$S_0$</td>
<td>$S_2$</td>
<td>$S_0$</td>
<td>$S_4$</td>
</tr>
<tr>
<td>$S_3$</td>
<td>1</td>
<td>$S_1$</td>
<td>$S_3$</td>
<td>$S_0$</td>
<td>$S_5$</td>
</tr>
<tr>
<td>$S_4$</td>
<td>0</td>
<td>$S_0$</td>
<td>$S_3$</td>
<td>$S_1$</td>
<td>$S_4$</td>
</tr>
<tr>
<td>$S_5$</td>
<td>1</td>
<td>$S_1$</td>
<td>$S_2$</td>
<td>$S_1$</td>
<td>$S_5$</td>
</tr>
</tbody>
</table>
Example: Multiple Inputs (Mealy)

The state graph for the Moore machine:
Questions?