Lecture 8:  
Finite State Machines  
And  
Sequential circuit Design  

CSE 140: Components and Design Techniques for Digital Systems  

Diba Mirza  
Dept. of Computer Science and Engineering  
University of California, San Diego
Generalized Model of Sequential Circuits
Canonical Form: Mealy and Moore Machines

Mealy Machine: $y_i(t) = f_i(X(t), S(t))$
Moore Machine: $y_i(t) = f_i(S(t))$

$s_i(t+1) = g_i(X(t), S(t))$

Mealy Machine

Moore Machine

$x(t)$: input (s) (external)
$s(t)$: state (multibit)
$y(t)$: output

Asynchronous

Synchronous output
Differences in State Diagram: Mealy vs. Moore Machines

Mealy Machine

If $S(t) = S_3$ and $x(t) = 1$, then $S(t+1) = S_1$ and $y(t) = 1$

Moore Machine

If $S(t) = S_1$, then $y(t) = 0$

If $S(t) = S_1$ and $x(t) = 1$, then $S(t+1) = S_2$
This Counter Design Is:

A. Moore machine  
B. Mealy machine
C. None of the above

Output depends only on state.

No external inputs are present in this case.
Life on Mars?

Mars rover has a binary input $x$. When it receives the input sequence $x(t-2, t) = 001$ from its life detection sensors, it means that it has detected life on Mars 😊 and the output $y(t) = 1$, otherwise $y(t) = 0$ (no life on Mars ☹).

Implement the Life-on-Mars Pattern Recognizer!

This pattern recognizer should have
A. One state because it has one output
B. One state because it has one input
C. Two states because the input can be 0 or 1
D. More than two states because ….
E. None of the above
Life on Mars?

Mars rover has a binary input $x$. When it receives the input sequence $x(t-2, t) = 001$ from its life detection sensors, it means that the it has detected life on Mars 😊 and the output $y(t) = 1$, otherwise $y(t) = 0$ (no life on Mars ☹️).
Mars Life Recognizer FSM

Which of the following diagrams is a correct Mealy solution for the 001 pattern recognizer on the Mars rover?

A. 1/0

B. 1/0

C. Both A and B are correct

D. None of the above
What does state table need to show to design controls of C1?

A. (current input $x(t)$, current state $S(t)$ vs. next state, $S(t+1)$)
B. (current input, current state vs. current output $y(t)$)
C. (current input, current state vs. current output, next state)
D. None of the above
State Diagram => State Table with State Assignment

State Assignment

<table>
<thead>
<tr>
<th>S(t)</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td>S1,0</td>
<td>S0,0</td>
</tr>
<tr>
<td>S1</td>
<td>S2,0</td>
<td>S0,0</td>
</tr>
<tr>
<td>S2</td>
<td>S2,0</td>
<td>S0,1</td>
</tr>
</tbody>
</table>

S0: 00
S1: 01
S2: 10

Mealy Machine

\[ Q_{1(t+1)} Q_{0(t+1)}, y \]

\[ S(t+1) \]
State Diagram => State Table => Excitation Table => Circuit

<table>
<thead>
<tr>
<th>Q₁(t) Q₀(t) \ x</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>01,0</td>
<td>00,0</td>
</tr>
<tr>
<td>01</td>
<td>10,0</td>
<td>00,0</td>
</tr>
<tr>
<td>10</td>
<td>10,0</td>
<td>00,1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>id</th>
<th>Q₁Q₀x</th>
<th>D₁</th>
<th>D₀</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>000</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>001</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>010</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>011</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>110</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>7</td>
<td>111</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Mealy Machine

Circuit Diagram
State Diagram => State Table => Excitation Table => Circuit

<table>
<thead>
<tr>
<th>id</th>
<th>Q_1Q_0x</th>
<th>D_1</th>
<th>D_0</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>000</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>001</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>010</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>011</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>110</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>7</td>
<td>111</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

D_1(t):  

\[
D_1(t) = x'Q_0 + x'Q_1
\]

D_0(t) = Q_1Q_0'x'

y = Q_1x

State Diagram:

State Table:

Excitation Table:

Circuit:
State Diagram => State Table => Excitation Table => Circuit

\[ D_1(t) = x'Q_0 + x'Q_1 \]
\[ D_0(t) = Q'_1Q'_0x' \]
\[ y = Q_1x \]
Summary: Implementation

- Set up canonical form
  - Mealy or Moore machine
- Identify the next states
  - state diagram $\Rightarrow$ state table
  - state assignment
- Derive excitation table
  - Inputs of flip flops
- Design the combinational logic
  - don’t care set utilization