Dr. Chuck Brown
Engineering and Computer Information Science
Folsom Lake College
Outline for Today's Lecture

• Moore & Mealy State Machines
• Factoring
Examples of Finite State Machines

• Vending Machine
• Combination Lock
• Elevator Controller
• Light Controller
• Alarm System
• Pattern Recognition
• Network Controller
• Interface Controller
• and more
Moore State Machines

- **Moore** is a finite-state machine whose output values are determined solely by its current state.
  - Moore machines are safer to use: Outputs change at clock edge (always one cycle later)
  - Additional logic may be necessary to decode state into outputs—more gate delays after clock edge
  - Some sequential circuits can only be implemented as Moore machines

This system takes in a stream of zeros and ones and outputs a 1 any time it gets the input sequence 011
Mealy State Machines

- **Mealy**¹ machine is a finite-state machine whose output values are determined both by its current state and the current inputs
  - Mealy machines tend to have fewer states
  - React faster to inputs in same cycle—don't need to wait for clock
  - Input change can cause output change as soon as logic is done—a big problem when two machines are interconnected
  - Not all sequential circuits can be implemented using the Mealy model

---

¹ named after George H. Mealy, who presented the concept in a 1955 paper, “A Method for Synthesizing Sequential Circuits”.

This system takes in a stream of zeros and ones and outputs a 1 any time it gets the input sequence 011
FSM Design Procedure

1. Identify inputs and outputs
2. Sketch state transition diagram
3. Write state transition table
4. Select state encodings
5. For Moore machine:
   1. Rewrite state transition table with state encodings
   2. Write output table
6. For a Mealy machine:
   1. Rewrite combined state transition and output table with state encodings
7. Write Boolean equations for next state and output logic
8. Sketch the circuit schematic
Design a Mealy and Moore FSM of a sequence detector that takes a serial data input stream of zeros and ones and outputs a one any time the input sequence ends in 01

- Inputs: $CLK$, $Reset$, $A$
- Outputs: $Y$

Diagram:
- Input: $CLK$
- Input: $A$
- Input: $Reset$
- Outputs: $Y$
State Transition Diagrams

Moore FSM

Moore FSM: arcs indicate input and state indicates output

Mealy FSM

Mealy FSM: arcs indicate input/output

This system takes in a stream of zeros and ones and outputs a 1 any time it gets the input sequence 01
Recap - FSM State Encoding

- **Binary encoding:**
  - i.e., for four states, 00, 01, 10, 11

- **One-hot encoding**
  - One state bit per state
  - Only one state bit HIGH at once
  - i.e., for 4 states, 0001, 0010, 0100, 1000
  - Requires more flip-flops
  - Often next state and output logic is simpler
Moore FSM State Transition Table

<table>
<thead>
<tr>
<th>Current State</th>
<th>Inputs</th>
<th>Next State</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>$S_0$</td>
<td>$A$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td>00</td>
</tr>
<tr>
<td>S1</td>
<td>01</td>
</tr>
<tr>
<td>S2</td>
<td>10</td>
</tr>
</tbody>
</table>
Moore FSM State Transition Table

<table>
<thead>
<tr>
<th>Current State</th>
<th>Inputs</th>
<th>Next State</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>$S_0$</td>
<td>$A$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

$S_1' = S_0A$

$S_0' = \overline{A}$
Moore FSM Output Table

<table>
<thead>
<tr>
<th>Current State</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>$S_0$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Moore FSM Output Table

<table>
<thead>
<tr>
<th>Current State</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>$S_0$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

$Y = S_1$
Moore FSM Schematic
# Mealy FSM State Transition & Output Table

<table>
<thead>
<tr>
<th>Current State</th>
<th>Input</th>
<th>Next State</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>A</td>
<td>$S'_0$</td>
<td>$Y$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td>00</td>
</tr>
<tr>
<td>S1</td>
<td>01</td>
</tr>
</tbody>
</table>
Mealy FSM State Transition & Output Table

<table>
<thead>
<tr>
<th>Current State</th>
<th>Input</th>
<th>Next State</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$A$</td>
<td>$S'_0$</td>
<td>$Y$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
S'_0 = \overline{A} \\
Y = S_0A
\]
Mealy FSM Schematic

\[ S' \quad 0 \quad Y \quad \text{CLK} \quad \text{Reset} \quad A \quad S_0 \quad r \quad \text{Reset} \quad Y \]
• Break complex FSMs into smaller interacting FSMs

• Example: Modify traffic light controller to have Parade Mode.
  – Two more inputs: $P$, $R$
  – When $P = 1$, enter Parade Mode & Bravado Blvd light stays green
  – When $R = 1$, leave Parade Mode
Traffic Light FSM Parade Mode

Unfactored FSM

Factored FSM
Unfactored FSM

S0
L_A: green
L_B: red

S1
L_A: yellow
L_B: red

S2
L_A: red
L_B: yellow

S3
L_A: red
L_B: yellow

S4
L_A: green
L_B: red

S5
L_A: yellow
L_B: red

S6
L_A: red
L_B: green

S7
L_A: red
L_B: yellow

Reset

PT_A
RT_A
PT_B
RT_B
Factored FSM

Lights FSM

Mode FSM

ENGR 303
Recap - FSM Design Procedure

1. Identify inputs and outputs
2. Sketch state transition diagram
3. Write state transition table
4. Select state encodings
5. For Moore machine:
   1. Rewrite state transition table with state encodings
   2. Write output table
6. For a Mealy machine:
   1. Rewrite combined state transition and output table with state encodings
7. Write Boolean equations for next state and output logic
8. Sketch the circuit schematic

ENGR 303