Finite State Machine (FSM)

- Consists of:
  - **State register**
    - Stores current state
    - Loads next state at clock edge
  - **Combinational logic**
    - Computes the next state
    - Computes the outputs
Finite State Machines (FSMs)

- Next state determined by current state and inputs
- Two types of finite state machines differ in output logic:
  - **Moore FSM**: outputs depend only on current state
  - **Mealy FSM**: outputs depend on current state and inputs

Moore FSM

Mealy FSM
1. Identify inputs and outputs
2. Sketch state transition diagram
3. Write state transition table
4. Select state encodings
5. Rewrite state transition table with state encodings
6. Write output table
7. Write Boolean equations for next state and output logic
8. Sketch the circuit schematic
• Traffic light controller
  – Traffic sensors: $T_A$, $T_B$ (TRUE when there’s traffic)
  – Lights: $L_A$, $L_B$
1. Identify inputs and outputs
2. Sketch state transition diagram
3. Write state transition table
4. Select state encodings
5. Rewrite state transition table with state encodings
6. Write output table
7. Write Boolean equations for next state and output logic
8. Sketch the circuit schematic
• Inputs: $CLK$, $Reset$, $T_A$, $T_B$
• Outputs: $L_A$, $L_B$
1. Identify inputs and outputs
2. Sketch state transition diagram
3. Write state transition table
4. Select state encodings
5. Rewrite state transition table with state encodings
6. Write output table
7. Write Boolean equations for next state and output logic
8. Sketch the circuit schematic

FSM Design Procedure
- **Moore FSM**: outputs labeled in each state
- **States**: Circles
- **Transitions**: Arcs

**FSM State Transition Diagram**

- \( S_0 \)
- \( L_A: \) green
- \( L_B: \) red

**Reset**

Academic Ave.
Bravo Blvd.
Dorms
Fields
Dining Hall
Labs
• **Moore FSM**: outputs labeled in each state
• **States**: Circles
• **Transitions**: Arcs

![Diagram](image)
1. Identify inputs and outputs
2. Sketch state transition diagram
3. Write state transition table
4. Select state encodings
5. Rewrite state transition table with state encodings
6. Write output table
7. Write Boolean equations for next state and output logic
8. Sketch the circuit schematic
FSM State Transition Table

<table>
<thead>
<tr>
<th>Current State</th>
<th>Inputs</th>
<th>Next State</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S$</td>
<td>$T_A$</td>
<td>$T_B$</td>
</tr>
<tr>
<td>S0</td>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td>S0</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>S1</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>S2</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>S2</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>S3</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
### FSM State Transition Table

<table>
<thead>
<tr>
<th>Current State</th>
<th>Inputs</th>
<th>Next State</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S$</td>
<td>$T_A$</td>
<td>$T_B$</td>
</tr>
<tr>
<td>S0</td>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td>S0</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>S1</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>S2</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>S2</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>S3</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
FSM Design Procedure

1. Identify inputs and outputs
2. Sketch state transition diagram
3. Write state transition table
4. Select state encodings
5. Rewrite state transition table with state encodings
6. Write output table
7. Write Boolean equations for next state and output logic
8. Sketch the circuit schematic
### FSM Encoded State Transition Table

<table>
<thead>
<tr>
<th>Current State</th>
<th>Inputs</th>
<th>Next State</th>
<th>State</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>$S_0$</td>
<td>$T_A$</td>
<td>$T_B$</td>
<td>$S'_1$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Two bits required for 4 states
# FSM Encoded State Transition Table

<table>
<thead>
<tr>
<th>Current State</th>
<th>Inputs</th>
<th>Next State</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$ $S_0$</td>
<td>$T_A$  $T_B$</td>
<td>$S'_1$ $S'_0$</td>
</tr>
<tr>
<td>0 0</td>
<td>0 X</td>
<td>0 1</td>
</tr>
<tr>
<td>0 0</td>
<td>1 X</td>
<td>0 0</td>
</tr>
<tr>
<td>0 1</td>
<td>X X</td>
<td>1 0</td>
</tr>
<tr>
<td>1 0</td>
<td>X 0</td>
<td>1 1</td>
</tr>
<tr>
<td>1 0</td>
<td>X 1</td>
<td>1 0</td>
</tr>
<tr>
<td>1 1</td>
<td>X X</td>
<td>0 0</td>
</tr>
</tbody>
</table>

$S'_1 = S_1 \oplus S_0$

$S'_0 = \overline{S_1} \overline{S_0} T_A + S_1 \overline{S_0} T_B$

**State**   | **Encoding**
---|---
S0 | 00
S1 | 01
S2 | 10
S3 | 11

Two bits required for 4 states
FSM Design Procedure

1. Identify inputs and outputs
2. Sketch state transition diagram
3. Write state transition table
4. Select state encodings
5. Rewrite state transition table with state encodings
6. Write output table
7. Write Boolean equations for next state and output logic
8. Sketch the circuit schematic
### FSM Output Table

<table>
<thead>
<tr>
<th>Current State</th>
<th>Outputs</th>
<th>Output</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$L_{A1}$</td>
<td>$L_{A0}$</td>
<td>$L_{B1}$</td>
</tr>
<tr>
<td>$S_1$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$S_0$</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$S_{11}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$S_{31}$</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$S_{21}$</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$S_{22}$</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Two bits required for 3 outputs
## FSM Output Table

<table>
<thead>
<tr>
<th>Current State</th>
<th>Outputs</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>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

- \( L_{A1} = S_1 \)
- \( L_{A0} = \overline{S_1}S_0 \)
- \( L_{B1} = S_1 \)
- \( L_{B0} = \overline{S_1}S_0 \)

### Output Encoding

<table>
<thead>
<tr>
<th>Output</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>green</td>
<td>00</td>
</tr>
<tr>
<td>yellow</td>
<td>01</td>
</tr>
<tr>
<td>red</td>
<td>10</td>
</tr>
</tbody>
</table>

Two bits required for 3 outputs
1. Identify inputs and outputs
2. Sketch state transition diagram
3. Write state transition table
4. Select state encodings
5. Rewrite state transition table with state encodings
6. Write output table
7. Write Boolean equations for next state and output logic
8. Sketch the circuit schematic
FSM Schematic: State Register

\[ S'_1 = S_1 \oplus S_0 \]
\[ S'_0 = S_1 S_0 T_A + S_1 S_0 T_B \]

\[ L_{A1} = S_1 \]
\[ L_{A0} = S_1 S_0 \]
\[ L_{B1} = S_1 \]
\[ L_{B0} = S_1 S_0 \]
\[ S'_1 = S_1 \oplus S_0 \]
\[ S'_0 = S_1 S_0 T_A + S_1 S_0 T_B \]
FSM Schematic: Output Logic

\[ \begin{align*}
L_{A1} &= S_1 \\
L_{A0} &= S_1S_0 \\
L_{B1} &= S_1 \\
L_{B0} &= S_1S_0
\end{align*} \]
FSM Timing Diagram

CLK
Reset
TA
TB
S'1:0
S1:0
LA1:0
LB1:0

Cycle 1
Cycle 2
Cycle 3
Cycle 4
Cycle 5
Cycle 6
Cycle 7
Cycle 8
Cycle 9
Cycle 10

S0 (00)
S1 (01)
S2 (10)
S3 (11)
S0 (00)
S1 (01)

S0 (00)
S1 (01)
S2 (10)
S3 (11)
S0 (00)

Green (00)
Red (10)
Yellow (01)
Red (10)
Green (00)

LA: green
LB: red
LA: yellow
LB: red
LA: red
LB: yellow
LA: red
LB: green

Reset
TA
TB
• **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
FSM Design Procedure

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