ECE 2300
Digital Logic & Computer Organization
Spring 2017

More FSMs
Timing, Clocking
Announcements

• Prelim 1 graded
  – High: 75 (out of 75); Mean: 57; Median: 58

• Prelab 3(B) due tomorrow

• Lab 4 to be released tonight
  – Stick to the same group from this lab
Review: Moore Transition/Output Table

- Uses state binary encodings

<table>
<thead>
<tr>
<th>( S_1 \ S_0 )</th>
<th>( S_1^* \ S_0^* )</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 00 )</td>
<td>( 00 )</td>
<td>01</td>
</tr>
<tr>
<td>( 01 )</td>
<td>( 00 )</td>
<td>10</td>
</tr>
<tr>
<td>( 10 )</td>
<td>( 00 )</td>
<td>11</td>
</tr>
<tr>
<td>( 11 )</td>
<td>( 00 )</td>
<td>11</td>
</tr>
</tbody>
</table>

\[
\text{Init: } S_0 = 0, \quad \text{Out: } S_1 = 0 \\
\text{Got1: } S_0 = 0, \quad \text{Out: } S_1 = 0 \\
\text{Got11: } S_0 = 0, \quad \text{Out: } S_1 = 0 \\
\text{Got111: } S_0 = 0, \quad \text{Out: } S_1 = 1
\]
Minimized Equations for $S^*$ and $Out$

$S_1^* = S_0 \cdot In + S_1 \cdot In$

$S_0^* = S_0' \cdot In + S_1 \cdot In$

<table>
<thead>
<tr>
<th>$S_1 \cdot S_0$</th>
<th>$S_1^*$</th>
<th>$S_0^*$</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$In = 0$</td>
<td>$In = 1$</td>
<td></td>
</tr>
<tr>
<td>0 0</td>
<td>0 0</td>
<td>0 1</td>
<td>0</td>
</tr>
<tr>
<td>0 1</td>
<td>0 0</td>
<td>1 0</td>
<td>0</td>
</tr>
<tr>
<td>1 0</td>
<td>0 0</td>
<td>1 1</td>
<td>0</td>
</tr>
<tr>
<td>1 1</td>
<td>0 0</td>
<td>1 1</td>
<td>1</td>
</tr>
</tbody>
</table>

Out = $S_1 \cdot S_0$
Review: Mealy Transition/Output Table

- Uses state binary encodings

<table>
<thead>
<tr>
<th>$S_1 , S_0$</th>
<th>$S_1^* , S_0^*$, Out</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>In = 0</td>
</tr>
<tr>
<td>0 0</td>
<td>0 0, 0</td>
</tr>
<tr>
<td>0 1</td>
<td>0 0, 0</td>
</tr>
<tr>
<td>1 0</td>
<td>0 0, 0</td>
</tr>
</tbody>
</table>

Diagram:

- States: Init, Got1, Got11
- Transitions:
  - Reset: $S_0 = 0, S_1 = 0$
  - $S_1 = 0, S_0 = 0$: $\text{In} = 0, \text{Out} = 0$
  - $S_1 = 1, S_0 = 0$: $\text{In} = 1, \text{Out} = 0$
  - $S_1 = 0, S_0 = 1$: $\text{In} = 0, \text{Out} = 0$
  - $S_1 = 1, S_0 = 1$: $\text{In} = 1, \text{Out} = 1$
Minimized Equations for $S^*$ and $Out$

\[ S_1^* = S_0 \cdot In + S_1 \cdot In \]
\[ S_0^* = S_1' \cdot S_0' \cdot In \]
\[ Out = S_1 \cdot In \]

<table>
<thead>
<tr>
<th>$S1$</th>
<th>$S0$</th>
<th>$S1^<em>$ $S0^</em>$, Out</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$In = 0$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$In = 1$</td>
</tr>
<tr>
<td>0 0</td>
<td>0 0</td>
<td>0 0, 0</td>
</tr>
<tr>
<td>0 0</td>
<td>0 1</td>
<td>0 1, 0</td>
</tr>
<tr>
<td>0 1</td>
<td>0 0</td>
<td>1 0, 0</td>
</tr>
<tr>
<td>1 0</td>
<td>0 0</td>
<td>1 0, 1</td>
</tr>
<tr>
<td>1 0</td>
<td>0 0</td>
<td>1 0, 1</td>
</tr>
</tbody>
</table>
FSM for Toggle Flip-Flop

- **Input:** T
- **Output:** Q
- **Current state:** S
- **Next state:** S*

\[ S^* = T'S + TS' \]
\[ Q = S \]

**Moore state diagram**

**Transition/output table**

<table>
<thead>
<tr>
<th>S</th>
<th>T = 0</th>
<th>T = 1</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</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>
Analyzing the FSM

What does this FSM do?
Transition and Output Equations

\[ S_0^* = \]

\[ S_1^* = \]

\[ \text{Out} = \]

![Diagram of a digital circuit with inputs, outputs, and gates](image)
## Transition/Output Table

<table>
<thead>
<tr>
<th>( S_1 \ S_0 )</th>
<th>( S_1^* \ S_0^* )</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \quad 0 \ 0 )</td>
<td>( \quad )</td>
<td>( \quad )</td>
</tr>
<tr>
<td>( \quad 0 \ 1 )</td>
<td>( \quad )</td>
<td>( \quad )</td>
</tr>
<tr>
<td>( \quad 1 \ 0 )</td>
<td>( \quad )</td>
<td>( \quad )</td>
</tr>
<tr>
<td>( \quad 1 \ 1 )</td>
<td>( \quad )</td>
<td>( \quad )</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
S_0^* & = \text{In}' \\
S_1^* & = \text{In}' \cdot S_1 \cdot S_0' + \text{In} \cdot S_0 \\
\text{Out} & = S_1 \cdot S_0
\end{align*}
\]
State Diagram

010 Pattern Detector

$S_0^* = \text{In}'$

$S_1^* = \text{In}' \cdot S_1 \cdot S_0' + \text{In} \cdot S_0$

Out = $S_1 \cdot S_0$

<table>
<thead>
<tr>
<th>$S_1$ $S_0$</th>
<th>$S_1^<em>$ $S_0^</em>$</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>In = 0</td>
<td></td>
</tr>
<tr>
<td>0 0</td>
<td>0 1</td>
<td>0</td>
</tr>
<tr>
<td>0 1</td>
<td>0 1</td>
<td>0</td>
</tr>
<tr>
<td>1 0</td>
<td>1 1</td>
<td>0</td>
</tr>
<tr>
<td>1 1</td>
<td>0 1</td>
<td>1</td>
</tr>
</tbody>
</table>
Traffic Light Controller

- 4-way intersection with traffic lights
- Opposing lanes sequence together
  - 20 seconds dwell on green
  - 5 seconds dwell on yellow
  - 25 seconds dwell on red
Four Scenarios

- E & W Green / N & S Red for 20 seconds
- E & W Yellow / N & S Red for 5 seconds
- E & W Red / N & S Green for 20 seconds
- E & W Red / N & S Yellow for 5 seconds
Traffic Light Controller States

- **10 states**
  - E & W Green / N & S Red1 for 5 seconds
  - E & W Green / N & S Red2 for 5 seconds
  - E & W Green / N & S Red3 for 5 seconds
  - E & W Green / N & S Red4 for 5 seconds
  - E & W Yellow / N & S Red for 5 seconds
  - E & W Red / N & S Green1 for 5 seconds
  - E & W Red / N & S Green2 for 5 seconds
  - E & W Red / N & S Green3 for 5 seconds
  - E & W Red / N & S Green4 for 5 seconds
  - E & W Red / N & S Yellow for 5 seconds

Clock period is 5 seconds
Traffic Light Controller FSM

State (EW/NS)

Outputs

State (EW/NS)

Outputs
Factoring FSMs

- Break FSM into multiple communicating FSMs
- Simplifies large FSMs
- May result in fewer states
Traffic Light Controller Using 2 FSMs

• Light Controller (LC) FSM has 4 states
  – G/R, Y/R, R/G, R/Y

• Timer FSM controls when the LC FSM advances to the next state
  – Keeps LC in Green states for 4 clock cycles

Next: tells LC FSM to advance to next state
Green: indicates the green light is currently on
Light Controller FSM

State (EW/NS)

- Green

Diagram:
- G/R Green = 1, Next = 1
- R/Y Green = 0, Next = 0
- Y/R Green = 0, Next = 0
- R/G Green = 1, Next = 1

Connections:
- Timer → LC
- Green from Timer to LC
- Next from LC to Timer
**Timer FSM**

**G**: wait for first Green before counting

**NG**: next Green period

- **G**: Next = 1
- **NG1**: Next = 0
- **NG2**: Next = 0
- **NG3**: Next = 0

Green = 0

Green = 1

Timer → Next → LC

Green → Timer
Traffic Light Controller Operation

Clk

Next

Green

LC state

Timer state

East

West

Lights

North

South

Lights

0s  5s  10s  15s  20s  25s  30s  35s  40s  45s  50s  55s  60s  65s
Moore vs. Mealy

• **States**
  – Mealy machines tend to have fewer states
    • Different outputs on arcs (up to $n^2$) rather than states ($n$)

• **Timing**
  – Mealy machines react faster to inputs
    • Need to avoid combinational loop (asynchronous feedback) when two FSMs are communicating
  – Moore machines are safer to use
    • No asynchronous feedback problem
    • Outputs change at clock edge (always one cycle later) – potentially higher clock frequency
Propagation Delay ($t_{pd}$)

- Time for change in input to change the output
- Typically specified between 50% points

Circuits have minimum and maximum propagation delays

- Minimum sometimes called the contamination delay and maximum the propagation delay
Timing Diagram

- Shows how outputs respond to changes in inputs over time

\[ S \cdot Y \]
\[ S' \cdot X \]

\[ X \]
\[ Y \]
\[ S \]

\[ t_p \text{ of AND gate} \]
Glitch (Hazard)

- Unplanned momentary switching of an output

- Types of glitches
  - Static 1-hazard: Input change causes output to go from 1 to 0 to 1 (should have stayed 1)

  \[
  \begin{array}{c}
  1 \\
  0 \\
  1 \\
  \end{array}
  \]

  - Static 0-hazard: Input change causes output to go from 0 to 1 to 0 (should have stayed 0)

  \[
  \begin{array}{c}
  0 \\
  1 \\
  0 \\
  \end{array}
  \]

  - Dynamic hazards: Input change causes a change from 0 to 1 to 0 to 1 or from 1 to 0 to 1 to 0 (there should be just one change)

  \[
  \begin{array}{c}
  0 \\
  1 \\
  0 \\
  1 \\
  \end{array}
  \]

  \[
  \begin{array}{c}
  1 \\
  0 \\
  1 \\
  0 \\
  \end{array}
  \]
Glitch Example

- Glitches are typically caused by unequal signal propagation delays through the circuit

Assume X and Y are 1
S changes from 1 to 0

Output signal should stay at 1,
but shows a transient 0 value
Timing Diagram Showing Glitch

- X
- Y
- S
- S'
- S\cdot Y
- S'\cdot X
- F

Glitch
Do Glitches Matter?

Glitches are fine, as long as they settle before the rising clock edge.

Must be glitch-free.
Sequential Circuit Timing

X
Y
S
S'
S \cdot Y
S' \cdot X
F
CLOCK

XX
YY
SS
SS'
S.Y
S'.X
FF
CLOCK

Lecture 10: 28
Glitch on F: No Problem

X
Y
S
S'
S\cdot Y
S'\cdot X
F
CLOCK

X
S
S\cdot Y
S'\cdot X
F
CLOCK
Synchronous Circuits

- On the triggering clock edge (clock tick), the input of a flipflop is transferred to the output and held

- Must ensure the output of the combinational logic has settled before the next clock tick
Before Next Class

• H&H 1.4, 3.4-3.5.5

Next Time

Timing Analysis
Binary Arithmetic