Announcements

- Exam Tuesday February 22, in class.
  - Coverage is through Lecture #8 (last week).
  - Review session is Monday evening 5-7pm, 125 Cory.
  - Check the website before coming to class on Tuesday for your room assignment.
- Homework due next week posted within a couple of day.
  - Quiz next Thursday as usual.

Outline

- Moore versus Mealy style state machines.

Finite State Machines

- Example: Edge Detector
  Bit are received one at a time (one per cycle), such as: 000111010

  Design a circuit that asserts its output for one cycle when the input bit stream changes from 0 to 1.

  Try two different solutions.
State Transition Diagram Solution A

IN   PS    NS  OUT
0    00     00    0
1    00     01    0
0    01     00    1
1    01     11    1
0    11     00    0
1    11     11    0

Solution A, circuit derivation

<table>
<thead>
<tr>
<th>IN</th>
<th>PS</th>
<th>NS</th>
<th>OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>00</td>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>00</td>
<td>01</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>01</td>
<td>00</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>01</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>11</td>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>11</td>
<td>11</td>
<td>0</td>
</tr>
</tbody>
</table>

Solution B

Output depends not only on PS but also on input, IN

IN   PS   NS   OUT
0     00 00 0
0     01 01 1
1     00 01 0
1     01 10 1

Edge detector timing diagrams

- Solution A: output follows the clock
- Solution B: output changes with input rising edge and is asynchronous wrt the clock.
**Spring 2005 EECS150 – Lec10-fsm2**

### FSM Comparison

**Solution A**

**Moore Machine**
- output function only of PS
- maybe more states (why?)
- synchronous outputs
  - no glitches
  - one cycle “delay”
  - full cycle of stable output

**Solution B**

**Mealy Machine**
- output function of both PS & input
- maybe fewer states
- asynchronous outputs
  - if input glitches, so does output
  - output immediately available
  - output may not be stable long enough to be useful (below):

If output of Mealy FSM goes through combinational logic before being registered, the CL might delay the signal and it could be missed by the clock edge.

### FSM Recap

**Moore Machine**

- input value
- output value
- present state
- next state
- outputs

**Mealy Machine**

- input value/output values
- present state
- next state
- outputs

*Both machine types allow one-hot implementations.*

### FSMs in Verilog

**Mealy Machine**

```verilog
always @(posedge clk)
    if (rst) ps <= ZERO;
    else ps <= na;
    case (ps)
        ZERO: begin
            out = 1`b1;
            na = ONE;
            end
            else begin
                out = 1`b0;
                na = ZERO;
                end
            if (in) begin
                out = 1`b1;
                na = ONE;
                end
            else begin
                out = 1`b0;
                na = ZERO;
                end
            default begin
                out = 1`b0;
                na = default;
                end
        end
```

**Moore Machine**

```verilog
always @(posedge clk)
    if (rst) ps <= ZERO;
    else ps <= na;
    case (ps)
        ZERO: begin
            out = 1`b1;
            if (in) na = CHANGE;
            else na = ZERO;
            end
            end
            CHANGE: begin
                out = 1`b1;
                if (in) na = ONE;
                else na = ZERO;
                end
            ONE: begin
                out = 1`b0;
                if (in) na = ONE;
                else na = ZERO;
                end
            default begin
                out = 1`b0;
                na = default;
                end
        end
```

### Final Notes on Moore versus Mealy

1. A given state machine *could* have both Moore and Mealy style outputs. Nothing wrong with this, but you need to be aware of the timing differences between the two types.
2. The output timing behavior of the Moore machine can be achieved in a Mealy machine by “registering” the Mealy output values.