What is the series of inputs that will produce the most TRUEs on the output?

Review Problem 25
1. Understand the problem
2. Draw the state diagram
3. Use state diagram to produce state table
4. Implement the combinational control logic
Vending Machine Example

Vending Machine:

1. Must be on edge that matches input pattern (0 or 01)
2. Cannot be on edge that tells you what to do.
3. Reset: State: Search for coin

PSM Rules

1. Single coin slot for dimes, nickels
2. NO change returned
3. Deliver package of gum after >= 10 cents deposited

State Diagram:

Nicky: $0.10
Dime: 10¢
Vending Machine Example (cont.)

State Table:

<table>
<thead>
<tr>
<th>ps</th>
<th>D</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\(\text{N} = \overline{\text{ps}} \cdot N + \text{ps} \cdot \overline{D} + \text{ps} \cdot D\)
Vending Machine Example (cont.)

Implementation:
Previous 2 cycles: m has true so the output is true if...

enum { A, B, C } ps, ns; // Present state, next state

output logic out;
input logic clk, reset, w';

module simple (clk, reset, w', out);

FSMs in Verilog - Declarations
assign out = (ps == C) !
or part of next-state logic. //
// output logic - could also be "always"
endcase

endcase

case (ps)
always_comb begin
Next State Logic //

C: if (w) ns = C;
else ns = A;
end

B: if (w) ns = C;
else ns = A;
end

A: if (w) ns = B;
end
endmodule

end

ps : us;

else

ps : A;

if (reset)

always @ (posedge clk)

begin

sequential logic (DFFs)

end

always @ (posedge clk) begin

sequential logic (DFFs)

end

FMS in Verilog - DFFs