Using D flip-flops and logic gates, design a Mealy machine that detects the sequence 0100.

A. First, list all the states (s0, s1, etc.) that we need to define, and briefly define them.

s0: first 0 not detected
s1: 0 detected
s2: 01 detected
s3: 010 detected

B. (Four points.) Write out the state diagram (not the circuit diagram).

C. Write out the necessary truth table, showing how Next State and Output depend on Present State and Input.

<table>
<thead>
<tr>
<th>q1</th>
<th>q0</th>
<th>In</th>
<th>D1</th>
<th>D0</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>s0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>s0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>s1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>s1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>s2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>s2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>s3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>s3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
D. (Three points.) Write a logic equation for each bit’s Next State and for the Output.

\[
D_0 = \sim In
\]
\[
D_1 = q_0 \& In \mid q_1 \& \sim q_0 \& \sim In
\]
\[
Out = q_1 \& q_0 \& \sim In
\]

E. (Three points.) Write out the circuit diagram.

![Circuit Diagram](image)

F. (Ten points.) Verilog module

```verilog
module test2 (input clk, input In, output reg LatchedOutput);
reg q0;
reg q1;
always@(posedge clk)
begin
```

```verilog```
q0 <= ~In;
q1 <= q0&In | q1&~q0&~In;
LatchedOutput <= q1 & q0 & ~In;
end
endmodule