3) Register Design using AND, OR, INV gates

Register in Fig. 1.13 is composed of two back-to-back latches which operate on opposite phases of the clock; together, these two latches form a positive edge−triggered D flip−flop.

In this question, we are asked to replace the action of the switches with a sea of combinational gates: AND, OR, INV. This is shown above by circling the two identical regions in the latches above.
Intuitively, we want the latch switches to have the following behavior:

\[ \begin{align*}
\text{PASS} = 1 & \Rightarrow \text{Switch output should be new value} \\
\text{PASS} = 0 & \Rightarrow \text{Switch output should be stored value}
\end{align*} \]

\[ \text{Switch output} = (\text{PASS} \cdot \text{NEW}) + (\overline{\text{PASS}} \cdot \text{STORED}) \]

Substituting this into our diagram from the previous page, we get the following diagram.
Shown below is a timing diagram which describes the behavior of the above flip-flop:

- Clock (clk)
- Data (d)
- Qn
- Q

<table>
<thead>
<tr>
<th>Clock (clk)</th>
<th>Pass</th>
<th>Store</th>
<th>Pass</th>
<th>Store</th>
<th>Pass</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data (d)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td>XXXX</td>
<td>XXXX</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- XXXXX – Assuming we’re on our first clock cycle and we don’t know what the previous stored value was (after all, it’s the 1st cycle...)

Q <= D only at positive clock edge and stores this value until the next positive edge of clk... where Q then takes on a new value of D at this edge, and so on.

- [Handwritten date: 7/3/2003]