1. Design a single 8-input MUX (with no enable) using only the below MUX’s. You MUST use **one and only one of each** of the two tri-state enabled MUX’s (in a non-trivial way). You may use **as many** of the un-enabled MUX’s as you need to complete the problem. **You do not need other components. DO NOT DRAW WIRES**, use **LABELS** to indicate a connection (e.g., two pins labeled by a signal name are connected together). Show **activation levels** (H or L) in the parentheses next to every pin name. Pins \{X_7-X_0\} are inputs, \{S_2-S_0\} are select and Y is the output. **(Hint: Design two 4-input MUX’s.)**

**As many** of these MUX’s as you need.

\[
\begin{align*}
X_0 & ( ) \\
X_1 & ( ) \\
X_2 & ( ) \\
X_3 & ( ) \\
X_4 & ( ) \\
X_5 & ( ) \\
X_6 & ( ) \\
X_7 & ( ) \\
\end{align*}
\]

\[
\begin{align*}
S_2 & ( ) \\
S_1 & ( ) \\
S_0 & ( ) \\
\end{align*}
\]
2. The figure below shows a block diagram design of a State Machine controller based on the ROM Method with D flip-flops. Assume that all inputs and outputs are active-high. The ROM contents are also given in the table below.

(50%) a) Derive the corresponding Algorithmic State Machine (ASM) chart. Show ALL work. (Do not miss part b below.)

<table>
<thead>
<tr>
<th>Addr Hex</th>
<th>Value Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$19</td>
</tr>
<tr>
<td>1</td>
<td>$08</td>
</tr>
<tr>
<td>2</td>
<td>$0B</td>
</tr>
<tr>
<td>3</td>
<td>$06</td>
</tr>
<tr>
<td>4</td>
<td>$10</td>
</tr>
<tr>
<td>5</td>
<td>$10</td>
</tr>
<tr>
<td>6</td>
<td>$00</td>
</tr>
<tr>
<td>7</td>
<td>$00</td>
</tr>
</tbody>
</table>

(50%) b) If Y2 in part (a) is changed to active-low, describe in detail all the changes in the ASM (if any) and redraw the ASM (with the changes).