6.

10.
24. 

\[
\begin{align*}
S_1 S_0 &= 01_{2} = 1 \text{ (decimal)} \\
Y &= D_1 = 1
\end{align*}
\]

28. 

Even parity happens 5 times

\[
Y = (A_0 \oplus A_1) \oplus (A_2 \oplus A_3)
\]

\[
Y = 1 \text{ when the input has an odd number of } 1.
\]
A. Consider two 4-bit words \( X = x_3x_2x_1x_0 \) and \( Y = y_3y_2y_1y_0 \). Design a logic circuit whose output \( Z = 1 \) when the following conditions are simultaneously true:

- \( x_3 \) is not equal to \( y_2 \cdot y_0 \)
- \( x_2 \) is equal to \( y_3 \oplus y_1 \)
- \( x_1 \) is equal to \( x_3 \cdot y_2 \)

B. Using a 8:1 MUX (8-input data selectors) implement the Boolean functions:

\[
G = (A' \cdot B \cdot C) + (A \cdot B \cdot \overline{C}) + (B' \cdot C) + (A + \overline{A})
\]

\[
H(W, X, Y, Z) = \prod M(1, 2, 4, 7, 8, 9, 11, 13)
\]

\[
G(A, B, C) = (A \cdot B \cdot C) + (A \cdot B \cdot \overline{C}) + (\overline{B} \cdot C) + (A + \overline{A})
\]

\[
\sum m(1, 3, 5, 6)
\]
We will follow Example 6-17 from the textbook.

C. Using only 4:1 MUXes (4 input data selectors) design a 8:1 MUX. Label the all inputs and outputs. Using the 8:1 MUX that you designed, design the following Boolean function:

\[ F(A, B, C) = \Sigma m(2, 3, 6, 7) \]

Assume: 4:1 MUX has an enable input EN-L

To build an 8:1 MUX, we would need two 4:1 MUXes & one inverter & an OR gate. (This is what was done in Project 3). However, since the design must be done using only 4:1 MUXes, here is the correct design.
\[ F(A, B, C) = \sum m(2, 3, 6, 7) \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

D. Using Table 6.7, convert the BCD number 10001001 to binary.

\[ \begin{array}{c}
78 \ 40 \ 28 \ 16 \ 6 \ 4 \ 2 \\
\hline
1000 \ 1001 \ (89)
\end{array} \]