CSE 140 Homework 3
Prof. Tajana Rosing

1. Implement the given Boolean function using a 4-input multiplexer and minimum number of logic gates. You may use any desired gate. Use AB for the select lines.

\[ F(A,B,C,D) = \Sigma(1,2,5,7,8,10,11,13,14). \]

| \( \begin{array}{cccc}
AB & CD & 00 & 01 & 11 & 10 \\
00 & & & & & 1 \\
01 & 1 & 1 & 1 & & \\
11 & 1 & & 1 & & \\
10 & 1 & 1 & 1 & & \\
\end{array} \) |
|---|

- \( AB = 00 \rightarrow F = C'D + CD' = C \oplus D \)
- \( AB = 01 \rightarrow F = C'D + CD = D \)
- \( AB = 10 \rightarrow F = (C'D)' = C + D' \)
- \( AB = 11 \rightarrow F = C'D + CD' = C \oplus D \)
2. Build an 8-bit comparator that compares unsigned numbers \( A = a_7 \cdots a_0 \) and \( B = b_7 \cdots b_0 \) and outputs 1 if \( A > B \). First build a smaller unit (using K-map) with logic gates that compares two bit numbers \( X = x_1 x_0 \) and \( Y = y_1 y_0 \). Then, use sufficient number of these elements with required additional gates to build the final circuit.

\[
G = F(X > Y) = x_1 y'_1 + x_0 y'_1 y'_0 + x_1 x_0 y_0'
\]

\[
E = F(X = Y) = x'_1 x'_0 y'_1 y'_0 + x'_1 x_0 y'_1 y_0 + x_1 x_0 y'_1 y_0' + x_1 x_0 y_1 y_0
\]
3. Implement a circuit with minimum number of half adders and full adders that outputs $10 \times A$ where $A = a_3 a_2 a_1 a_0$ is a 4-bit binary number (give priority to using half adder whenever possible).
4. For a 4-to-2 encoder, first show that if more than one input is active (one), it produces incorrect output. Then, design a 4-input priority encoder using:

(a) A typical 4-to-2 encoder and additional logic gates.
(b) Minimum number of 2-to-1 multiplexers.

Note that a priority encoder produces correct output according to the most significant input bit, ignoring the rest. For instance, when both \( I_2 \) and \( I_0 \) are 1 (active), the output becomes \( Y = 00 \), denoting input \( I_0 \) (which has higher priority than \( I_2 \)).

The equations for an encoder with output \( Y = Q_1Q_0 \):

\[
Q_0 = I_1 + I_3 \quad (Q_0 = 1 \text{ means an input with “odd” index is active})
\]
\[
Q_1 = I_2 + I_3 \quad (Q_1 = 1 \text{ means an input with index larger than 1 is active})
\]

When \( I_0 = 0, I_1 = 1, I_2 = 1, I_3 = 0 \rightarrow Q_1Q_0 = 11 \), denoting \( I_3 \) is active, which is wrong.

The solution of part (b) is not unique. For output “A”, we check whether \( Q_1Q_0 = 11 \) is happened because of \( I_3 = 1 \), or because all inputs were 0. This helps us save 1 multiplexer than cascading 3 multiplexers to check if all inputs are 0.