Implement the following function in the specified technologies. You may use a minimum number of extra gates where needed:

\[ F = (A + B + D) + BCD + ACD + \overline{BCD} = \overline{A} \overline{B} \overline{D} + \overline{B} \overline{C} \overline{D} + \overline{A} \overline{C} \overline{D} + \overline{B} \overline{C} \overline{D} \]

a.) 4:16 decoder:

b.) 16:1 MUX

c.) 8:1 MUX
In class we built a 4:2 priority encoder, which required a VALID output signal and allowed multiple inputs to be on at a time. An alternative is to simply define address 0 as the invalid state, and not allow an input wire D0.

Design a 3:2 priority encoder. It will take inputs D3, D2, and D1, and produce A1 and A0. If exactly one of the inputs is true, the output is the binary number corresponding to the input number (D3 -> 11, D2 -> 10, D1 -> 01). If no input is true, or multiple inputs are true, the output will be 00.

\[
\begin{array}{c|cc|cc}
 D3 & D2 & D1 & A1 & A0 \\
--- & --- & --- & --- & --- \\
 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 1 \\
 0 & 1 & 0 & 1 & 0 \\
 0 & 1 & 1 & 0 & 0 \\
 1 & 0 & 0 & 0 & 0 \\
 1 & 0 & 1 & 0 & 0 \\
 1 & 1 & 0 & 0 & 0 \\
 1 & 1 & 1 & 0 & 0 \\
\end{array}
\]

\[
A_1 = D3 \overline{D2} D1 + \overline{D3} D2 \overline{D1}
\]

\[
A_0 = D3 \overline{D2} \overline{D1} + \overline{D3} \overline{D2} D1
\]