We practice the standard interconnect module designs and applications.

1. (Decoders) Given four four-input Boolean functions
   \[ f_1(a, b, c, d) = \sum m(0, 1, 4, 7) + \sum d(2, 3, 5), \]
   \[ f_2(a, b, c, d) = \sum m(1, 4, 11) + \sum d(3, 14), \]
   \[ f_3(a, b, c, d) = \sum m(0, 8, 10, 11) + \sum d(9, 12, 15). \]
   \[ f_4(a, b, c, d) = \sum m(2, 12) + \sum d(8, 11, 14). \]
   
   1.1. Implement the functions using a minimal network of 4:16 decoders and OR gates.
   1.2. Implement the functions using a minimal network of 3:8 decoders and OR gates.
   1.3. Implement the functions using a minimal network of 2:4 decoders and OR gates.

2. (Encoders) A high bit priority encoder inputs \( 2^n \) bits from \( 2^n \) devices and outputs \( n \) bit as the index of the asserted input line with the highest priority (largest in binary code) as shown in page 25 of lecture 11. Implement a high bit 16:4 priority encoder using 4:2 high bit priority encoders and minimal networks of NAND gates.

3. (Multiplexers) Assume a dual-railed system, where you have access to any variable and its complement. Implement the following four-input Boolean function as indicated in each of the following subproblems.
   \[ f(a, b, c, d) = \sum m(1, 2, 3, 5, 8, 13) + \sum d(0, 7, 10, 14). \]
   
   3.1. Implement the function using a minimal network of 8:1 multiplexers.
   3.2. Implement the function using a minimal network of 4:1 multiplexers.
   3.3. Implement the function using a minimal network of 2:1 multiplexers.

4. Assume a dual-railed system, where you have access to any variable and its complement. Given a four-input Boolean function
   \[ f(a, b, c, d) = \sum m(0, 3, 4, 7, 10, 12) + \sum d(5, 11, 14). \]
   
   4.1. Implement the function using a minimal network of 2:4 decoders and OR gates.
   4.2. Implement the function using a minimal network of 4:1 multiplexers.
   4.3. Implement the function using a minimal network of 2:1 multiplexers.