Name: _____

For this assignment you may not use any built-in **decoder**, **encoder**, **mux**, or **demux** components. You may build and use your own implementations of any of these components using `And`, `Or`, and `Not`.

1. An *encoder* is a combinatorial circuit that takes some number of inputs, say 16, and encodes them as a binary number.

   You are to implement a `16To4BitPriorityEncoder` with 16 inputs and a 4-bit output.

   When *no* inputs are set to 1 (asserted), then the output should be a binary 0. When just input i is asserted, the output is the binary representation of i.

   The *priority* part is what happens if more than one wire is asserted simultaneously: the output is the binary encoding of the *highest* asserted wire.

   The circuit has 16 1-bit inputs and 1 4-bit output. Label the inputs as `i0` through `iF` (0123456789ABCDEF) and the output `out`.

   Write a *complete* test case for your encoder for all inputs with only one wire asserted. Also test every wire simultaneously with `i0` and simultaneously with `iA`.

   Save your correct `16To4BitPriorityEncoder.dig` file.

2. A *decoder* is the opposite of an encoder: given $\log_2 n$ inputs, interpret that as a binary number and assert one of n wires. You are to implement a `4BitToOneOf16Decoder`.

   The circuit has a 4-bit binary input (`i`) and sixteen 1-bit outputs (`out0, out1, ..., outF`).

   Write a *complete* test case for your encoder for all possible input values.

   Save your correct `4BitToOneOf16Decoder.dig` file.

Deliverables: When all the tests pass, upload a `git` repository to Gitea with your two files. No need for a `README` for this assignment.

Points earned: _____
out of a possible 0 points