

# EE480 Assignment 0: Maximum Of Two 8-Bit Unsigned Binary Integers

## Implementor's Notes

Hank Dietz

Department of Electrical and Computer Engineering  
University of Kentucky, Lexington, KY USA  
hankd@engr.uky.edu

### ABSTRACT

This project was a simple combinatorial design problem to ensure that students were somewhat comfortable with basic use of Verilog, including the concept of writing an exhaustive testbench.

### 1. GENERAL APPROACH

This assignment required construction of a synthesizable, combinatorial, 8-bit unsigned maximum unit. I've provided two very different solutions, one using each of the two algorithms suggested in the project assignment.

The first algorithm works down from the MSB to the LSB tracking two 1-bit flags at each stage: `d` is one if the max is already decided and `w` is who is the max. It doesn't matter who we pretend is max until it is decided, because both values match to that point, but once it is decided, who simply propagates down to the lower bits unchanged. Each bit of the max is copied from who was selected at that position.

The second algorithm basically guesses that `a` is the maximum and then propagates guesses from the LSB up, adjusting our guess whenever the bits in `a` and `b` differ. The `whomax` module holds the logic determining how to update the guess for one bit position, and it is instantiated 8 times in `max`. Of course, the guesses at lower bit positions very often are wrong. That doesn't matter, because only the final guess, `w[7]`, is used to select to copy all the bits of `a` or `b` as the max.

The `refmax` (oracle) and `testbench` modules are identical in both solutions.

### 2. ISSUES

It was a little confusing how each variable should be declared and when it should be updated.... The `#1` in `testbench` is a hack to ensure that each pair of inputs is processed in sequence with nothing missed. There are various other ways to do this, but I didn't want to implement a clock for testing what is inherently a combinatorial circuit.

Everything was tested and apparently worked correctly the first time it made it through the WWW-form compiler and simulator. However, to confirm that the error detection code in `testbench` worked, I deliberately introduced an error into each version (deliberately using the wrong subscript for one of the references to `b`). The assignment did not make clear what format the faulty outputs should be listed in, so I just composed a format where each line starts with `Wrong:`.