University of California, Davis
Department of Electrical and Computer Engineering

EEC180 DIGITAL SYSTEMS II Winter Quarter 2021

# LAB 5: Design and Implementation of a Priority Arbiter

**Objective**:
A priority arbiter is an important circuit in networking applications, where the next packet to send is selected according to some Quality-of-Service policy that gives each packet a score.
In this lab you will design priority arbiters and use Verilog to implement them.

**Prelab**
Read the lab carefully and show the paper designs for Part II and III.

## I. Priority Arbiter

A four-input priority arbiter is shown below, which accepts four inputs and outputs the index of the input with the largest value. For example, if the four inputs are 28, 58, 19, and 37, the arbiter should output 1 (Out1 = 0, Out0 = 1), because input 1 has the highest value, 58.

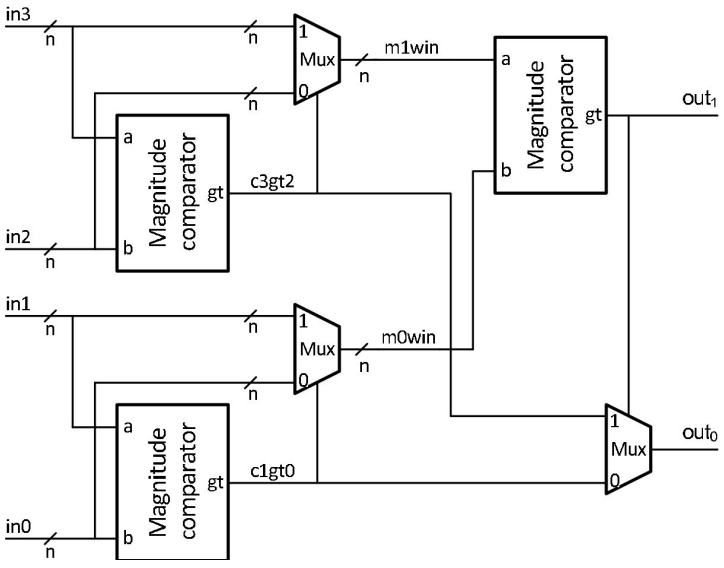


Fig. 1 Four-input priority arbiter design (See Page 193 and 194 from your textbook)

## II. Four-Input Priority Arbiter Design and Implementation

- Modify the design shown in Figure 1 to break the tie in favor of higher-numbered input for a four-input priority arbiter.

- Implement the design in Verilog.
- The size of the inputs, **n** should be Verilog parameter, so that the same design can be used to create priority arbiters of different size.

## III. Eight-Input Priority Arbiter Design and Implementation

- Modify the priority arbiter shown in Fig. 1 to take eight inputs and implement your eight-input priority arbiter design in Verilog using magnitude comparators and MUX.
- You should break ties in favor of the lower-numbered input.

## IV. Testing

- Write Verilog testbenches for both Part II and III.
- Verify your designs for n=8 and n=16 with at least 100 random test inputs. You can use the Verilog task $random to generate random numbers in your testbench.
- You must include test cases to verify the "ties" conditions.

## V. Synthesis

- Synthesize your design using Quartus
- Tabulate the resources used and delay for the priority arbiters for n=8 and n=16 in section II.
- Optimize your design to reduce the resources and delay [Extra Credit]. You can take an additional week to submit the extra credit.

## VI. Lab report

For your lab report, include the following:

- Paper designs for Part II and III.
- Complete Verilog source code.
- Simulation waveforms/screenshot of the transcript window from your functional simulation.
- Statement of contribution of each team member.

## VII. Grading Guidelines

| | |
|---|---|
| • Prelab | 10 points |
| • Part II Functional Simulation Check off | 25 points |
| • Part III Functional Simulation Check off | 25 points |
| • Synthesis | 20 points |
| • Lab Report | 20 points |
| • Extra Credit | 25 points |