CS2310 Lab 1: Digital Logic Design

Semester: Jul-Nov 2024

Due: August 13, 2024 (in class)

1 Circuit Design

Design a Verilog module that implements the following logic function:

$$\mathbf{F} = \mathbf{XY} + \overline{\mathbf{XY}} + \overline{\mathbf{Y}}\mathbf{Z}$$

where **X**, **Y**, and **Z** are **input** bits, and **F** is the **output** bit. Implement the module in three different ways:

a) Using **AND**, **NOT**, and **OR** gates. **[10]**

b) Using **ONLY OR**, and **NOT** gates. **[15]**

c) Using **ONLY AND**, and **NOT** gates. **[15]**

Each of these modules should have the signature `module q1i(output f, input x, input y, input z)`, where **i** corresponds to the sub-part letter (a, b, or c) of the module.

Additionally, write a test bench that tests each part for **at least five different sets** of input values (`X`, `Y`, `Z`).

Note: You can use the same test bench for all three sub-parts, with only minor modifications needed for each.

Provide the **complete** Verilog code for each of the implementations **as well as the test bench**. Ensure that your modules correctly reflect the logic function specified above.

---

2 Boolean Simplification

Consider the Boolean expression provided below in canonical Sum-Of-Products form. Here $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ are the input bits and **F** is the output bit.

$$\mathbf{F}(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}) = \sum(\mathbf{0}, \mathbf{1}, \mathbf{4}, \mathbf{5}, \mathbf{10}, \mathbf{11}, \mathbf{14}, \mathbf{15})$$

1. Implement the above Boolean expression as a module in Verilog. The module should have the signature `module q2a(output f, input a, input b, input c, input d)`. For this part, you may use multiple (more than 2) input gates. **[10]**
2. Implement the above Boolean expression as a module in Verilog **with only 2 input gates**. The module signature should be `module q2b(output f, input a, input b, input c, input d)`. **[15]**

3. Compute the simplified expression for **F**. Implement this simplified expression as a module in Verilog **with atmost 2 input gates**, with the signature `module q2c(output f, input a, input b, input c, input d)`. **[15]**
   Hint: For full credit, the final simplified expression and implementation should have at most 2 AND gates, 2 NOT gates, and 1 OR gate.

The testbench for this question will be provided to you during the lab. Your Verilog files should be named `q2a.v, q2b.v` and `q2c.v` with the corresponding modules in them.

---

3 Single-bit Comparator

Design a Verilog module for a single-bit comparator. The module should take two single-bit inputs, `A` and `B`, and provide three single-bit outputs, `X`, `Y`, and `Z`. **[20]**

- `X` is 1 **only** when `A` $>$ `B`, and 0 otherwise.
- `Y` is 1 **only** when `A` $=$ `B`, and 0 otherwise.
- `Z` is 1 **only** when `A` $<$ `B`, and 0 otherwise.

Name the module `comparator`. Write the Verilog code for this module, ensuring that it correctly implements the described functionality. Additionally, **write a test bench** that tests the module **for all possible combinations** of inputs `A` and `B`.

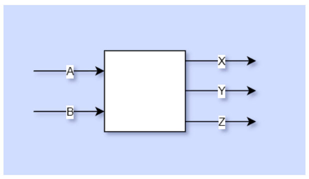

Figure 1: Single-bit comparator

Bonus

Implement the single-bit comparator from part (2) using **only** minimum number of **NAND** gates.

---

4 Bonus Question: The Universal Gates

Consider the Boolean expression for **F** given below:

$$\mathbf{F} = \mathbf{XY} + \overline{\mathbf{XY}} + \overline{\mathbf{W}}\mathbf{Z} + \mathbf{W}\overline{\mathbf{Z}}$$

Implement this expression in Verilog in two different ways:

1. Use only NAND gates. Name this module `F_nand`.
2. Use only NOR gates. Name this module `F_nor`.