UNIVERSITY OF TEHRAN
Electrical and Computer Engineering Department
Digital Logic Design, ECE 367, Fall 1394
Computer Assignment 3
Expressions with Abstract Functionalities in Verilog
Week 9

Name:
Date:

In this assignment you will learn the design of complex circuits using functional units and existing packages. This is start of seeing how a design is put together at the RT level.

1. Using Verilog **assign** statement, and constructs such as **generate**, condition-operator (**?:**), shift-operator (>>, <<), write a description for a configurable barrel-shifter. Use #**parameter** to parametrize the shifter. Based on logic gate delay of the previous Computer Assignments and the structure of the barrel-shifter, estimate t_{plh} and t_{phl} for the shifter outputs and use the delays in the **assign** statements describing the shifter.

2. We define an *n-to-*2^n Inclusive Binary Decoder as an *n*-bit input binary decoder in which binary input value *b* causes all outputs numbered *b* and less than *b* to become active. (If we had first designed an XOR, we would have called it an OR gate, and then the regular OR gate would be named Inclusive-OR). Using Verilog **assign** statement, and constructs such as **generate**, condition-operator (**?:**), shift-operator (>>, <<), and necessary Boolean and logical operators, write a description for a configurable IBD. Use #**parameter** to parametrize this unit. Based on logic gate delay of the previous Computer Assignments and the structure of the IBD, estimate t_{plh} and t_{phl} for the outputs of this hardware, and use the delays in the **assign** statements describing it.

3. Using the component of Part 1, a decoder, and component of Part 2, show how a 16-bit barrel-shifter and a decoder can be used for a circuit for dividing *m* by 2^n (i.e., $m \div 2^n$) and calculating the quotient. Also show how such components can be used for calculating the remainder of division. You may need extra logic gates. Show a block diagram for fining the Quotient and Remainder.

4. The divider of Part 3 generates two 16-bit integers for the quotient and the remainder. In this part, you are to use these results and produce a 16-bit fixed point number eight bits for its fractional part and eight bits for the integer part. For calculating the fractional part, you will use the remainder part and a hardware similar to the part for calculating the quotient as in Part 2.

5. Write a testbench for testing your hardware.