

ELEC 5250/6250 VHDL Project #4

First draft – Due Thursday, September 15 (for instructor feedback only – no grade)

Final version, including test bench and simulation – Due Tuesday, September 20

Design a hierarchical VHDL register-transfer-level (RTL) model of a circuit that performs the arithmetic division operation on two unsigned integer numbers. A is to be an 8-bit unsigned “divisor”, and B is to be a 16-bit unsigned “dividend”. The quotient, Q, and remainder, R, are each to be 8-bit unsigned integer values. The algorithm to be implemented is to be an iterative (sequence of 8 subtract/add and shift operations) “restoring division” or “non-restoring division” algorithm (refer to various computer architecture text books for descriptions.)

1. Design four separate VHDL models: (1) register component (one common model to be instantiated for all registers), (2) the arithmetic unit, (3) the controller, and (4) a top-level model that instantiates these components.
2. Use “register-transfer-level” design, i.e. do not model individual gates and flip flops in your components. A “register” should be a relatively simple design, with data inputs and outputs of type “std_logic_vector”, and other signals of type “std_logic”.
3. For the registers, design a single multifunction register (shift, load, etc.), and instantiate that as needed for each register in the divide circuit.
4. The arithmetic unit should be a behavioral model, using arithmetic operators (add, subtract) between vectors (Std_Logic_Vector or UNSIGNED types); do not design a binary adder at the bit level.
5. The controller should be a behavioral model of a finite state machine.
6. The top-level should simply instantiate the registers, arithmetic unit, and controller.
7. **ELEC 5250** students: You may assume that only valid operands are to be divided (no overflow detection needed.)
ELEC 6250 students: You are to detect “overflow” conditions – those that would generate invalid results (quotient greater than 8 bits, divide by 0, etc.) There should be an OVERFLOW output signal to indicate the occurrence of such a condition.
8. The functionality of each of the components should be tested individually in ModelSim, in addition testing the top-level component. For the components, you may use either a force file or a test bench. For the top-level test, use a “test bench” to exercise the circuit with at least 50 pairs of operands, automatically check the result, and print a message if any errors are detected.

To be submitted:

1. The four VHDL models.
2. The top-level test bench, plus the force files or test benches used to test the three components.
3. An “edited” file of the top-level results from a **List Window**. Save the List Window, and then edit out all but a few lines at the start and conclusion of each of the 50 operations. **(Use minimal paper.)** You do not need to submit simulation results for the three individual components.