Lab Demo Requirements

ECE532S Digital Systems Design

January 10, 2007

For the lab demo on February 7, 2007, you should do the following two tasks. Although you may work on the tasks as a group, each person in the group should be prepared to demonstrate any part of any of the tasks. This is worth 10% of your final grade.

1. Build a MicroBlaze system that can blink an LED at the rate of once per second based on interrupts from a timer. The program should run until a character is typed on the serial port.

   Add the snoopy core from Module m05 to determine:

   (a) The time it takes to execute the interrupt service routine;

   (b) The number of times the interrupt service routine is called.

2. Using the HDL of your choice, develop a synthesizable block that implements a simple 16 by 16 sequential multiplier. See Figure 6.7, page 379, in *Computer Organization*, Hamacher, Vranesic and Zaky, 5th ed.

   The block has the following ports:

   clk Input: the clock

   reset_b Input: an active low reset signal

   mer[15:0] Input: the 16-bit multiplier

   mand[15:0] Input: the 16-bit multiplicand

   go Input: the signal to start the operation

   product[31:0] Output: the 32-bit product

   done Output: the signal that indicates the product is valid

   The block operates as follows:

   (a) The **reset_b** signal is set to 0 for a few cycles to reset the state machine.

   (b) The operands are made available at the inputs.

   (c) When **go** goes high, the multiplier and multiplicand are stored in internal registers.

   (d) After the operands are stored, the block uses a sequence of shift-and-add operations to do the multiplication.

   (e) When the multiplication is completed, the **done** signal is raised.

   Your task is to:

   (a) Simulate this block using ModelSim.

   (b) Show a simple testbench using either ModelSim scripts or using an HDL.

   (c) Develop enough tests to convince someone that it is working.

   (d) Show the results of some tests with a waveform viewer.

   (e) Use ISE to synthesize the block and show that it can synthesize without errors. You do not need to actually load the chip and show it working in the hardware.