

Project: Part #4 for ECE 152

Memory (25 points)

Must be submitted electronically by 10:00AM on Friday, March 7

In this part of the project, you will build the instruction memory and the data memory for the Duke 152/16 processor. *This assignment is the **only exception** to the rule about not using Quartus megafunctions—memories **must** be implemented through megafunctions.* You will use Quartus’s Megafunction Wizard to instantiate the two memories. Before proceeding, make sure you have the latest version of Quartus II (version 7.2 SP2, as of this writing). The Megafunction Wizard differs substantially from release to release.

Although the Duke152/16’s ISA has a single address space, your microarchitecture should have separate instruction and data memories (i.e., they should be separate blocks). This is essential for a high-performance pipelined processor, or else memory would present a structural hazard. To facilitate making a processor that can issue up to two instructions per cycle later this semester, you will make an instruction memory with two read ports. The data memory, however, will be single-ported.

To make the instruction memory (imem.vhd), first go to Tools > MegaWizard Plug-In Manager. On page 1, select Create and hit Next. On page 2a, select the Cyclone II device family, select VHDL as the output format, and set the target name to imem.vhd (in your project’s working directory). On the left side, select Installed Plugins > Memory Compiler > RAM: 2-PORT.

On page 3, select two read/write ports, and indicate that you wish to specify the memory size in words. On page 4, make sure “use different data widths” is not checked, then set the width (of the output bus) to 16 bits. Specify that there should be 8192 words of memory, and set the block type to M4K. Then, on page 5, indicate a single clock domain and no byte enables. Page 6 is automatically bypassed. On page 7, un-check read output port registers. No clock enables or clears are necessary. On page 8, select don’t care; page 9 will also be skipped automatically. Page 10 allows you to specify a memory initialization file; this is how you will load a program into the RAM later. For now, leave it blank, and hit Finish twice.

Follow the same steps to create the data memory (dmem.vhd). The megafunction name this time is RAM: 1-PORT and the size is 16384 words. Recall again that you should not register the output port.

Both memories are input-registered but not output-registered. This means that the memory will not latch the address provided to it until the next rising edge, so the read or write will not be performed until the subsequent rising edge. To verify this behavior, extensively test the data memory. Submit your single test waveform as dmem.vwf.

Part of your grade for this assignment will be determined by the coverage of your test cases. Your waveforms need not, and should not, be excessively long.

Submit this assignment in the same way in which you submitted previous project parts. You will have three files (imem.vhd, dmem.vhd, and dmem.vwf) that you must tar up into a single file for uploading.

You may re-submit as often as you like, but a re-submission will overwrite whatever you've previously submitted for this assignment. I will grade whatever has been submitted before 10:00AM on Friday, March 7.