COMP3211/9211 Computer Architecture
2004 S2 Laboratory Exercise 1 (VHDL1)
Memory Elements (Draft 01/08)

Oliver Diessel

1 Goals

This laboratory exercise has two goals:

1. To (re-)acquaint you with ISE, the integrated design environment used in our computing labs to describe and simulate hardware structures using VHDL; and

2. To practice the description and testing of memory structures needed for the design of the Pipelined MIPS Processor you will implement for Assignment 1.

2 General procedures

This and future exercises are self-paced. You may need to refer to additional documentation that may not be explicitly provided to you. The demonstrators may be able to guide you to additional useful sources.

Future exercises will require you to submit one of the expected deliverables for the laboratory session. However, this exercise has no required hand-in. Please do not hesitate to ask for assistance from the demonstrators if you need help. They should be available for about 1.5 hours each lab session.

Attendance at labs will be noted.

3 VHDL Lab 1

3.1 Getting started with ISE and VHDL

Please refer to the VHDL references link on the course web-page for background information on VHDL and how to use the ISE design environment.

If you have not used `vmware` before, please refer to the instructions entitled “Getting started with vmware” under the “Using ISE in the labs” heading of the VHDL references page.

The Quick Start Tutorial is a good place to start if you have no experience with using ISE. At the very least, if you have no prior experience, you should complete this first. The In-Depth Tutorial offers a lot more background. The VHDL manual page is also a reasonable reference for the VHDL language if you do not have access to a reference text. The source files used by the tutorials are also available on-line from the VHDL references page.

If you have no prior experience with VHDL, you should study some reference material. The DSS lecture slides are a reasonable start, as are the reference texts, particularly Yalamanchili, as well as the last two references provided under the “On-line Documentation” heading on the VHDL references page.

3.2 D-type flip-flop

Implement and test a behavioural model of a positive edge-triggered D-type flip-flop with enable and asynchronous reset.

Hint: take a look at the VHDL Synthesis Templates for Flip-flops provided by the Language Assistant (click on the light-bulb icon at the right end of the ISE Project Manager toolbar icons).

Checkpoint: demonstrate the simulation of your flip-flop to your demonstrator.

3.3 32-bit Register

A register is a vector of flip-flops. Implement and test a 32-bit register with positive edge-triggered clock, enable, asynchronous reset, 32-bit data input, and 32-bit output. Try implementing as a behavioural model rather than a structural model composed of flip-flops.

3.4 Memory block

Implement and test an 8 x 32-bit RAM with enable, R/~W (read/not write) control, address and data inputs and data output.

Hints: Try figuring out what you need first, then take a look at the VHDL Synthesis Templates for ideas and download the model provided on the VHDL models page of the course web-site. How do these models differ?

3.5 Integration

Build a small structure consisting of the memory and register components, in which you store the output of the memory in a register when enabled. Test your design.

3.6 Questions

1. Why is it usual to provide a register or memory with an “enable” or “load” input?
2. What is the purpose of the R/~W control input on the memory block?
3. For the electronically minded: In contrast to the flip-flop and register you implemented, the memory block you explored is asynchronous (does not have a clock input or respond to clock transitions). Why so?