EEL4713 Assignment #3
Fall 2014
Checkpoint Demo: 10/23/14 in Lab
Final Demo: 10/30/14 in Lab
Report Due: 11/1/14 @ 11:55pm Via Sakai
Developed by Scott Arnold and Ann Gordon-Ross Fall 2012

Overview:
In this assignment, you will be designing a single cycle implementation of a MIPS processor. Your processor will need to be able to successfully execute 29 instructions – all of the instructions listed under the Core Instruction Set on the green reference sheet from your book, except for ll (load linked) and sc (store conditional). Please read sections 4.3 and 4.4 of your textbook for insight on how this can be done. You will be using the mif file generated in the previous lab (Lab3.mif) to demonstrate the functionality of your processor for the in lab demo.

In this assignment, you will perform the following tasks:

DEMOS: checkpoint demo and final demo of working processor
2.1: Design and simulate a MIPS single cycle processor
2.2: Annotate the simulations for the Lab3 demo program
2.3: Review notes on the processor design for this lab
3: Review deliverables and write the report

Section 1: Setup
There is no setup in this assignment.

Section 2: Laboratory
2.1: Single-Cycle Processor
Begin your design by implementing one of the 29 instructions similarly to figure 4.17 in your textbook. Then, modify this hardware to implement the next instruction, adding any additional control signals needed for the new instruction to function correctly. Repeat this process until all 29 instructions are fully implemented. Note: If you use Altera's atlsyncram to implement your memory, be sure to use the "mclk" signal, which clocks your ram at 3 times the speed of your processor. Clocking the ram at a higher speed than the processor speed is required to ensure that the correct instruction is available from your instruction memory after your program counter is updated when branch and jump instructions execute.

Your report should include a separate section for each instruction listed in alphabetical order with the following:

- A number in the section heading that indicates the order in which you added the instructions' functionality to your processor. For example, ff you first implemented the ADD instruction, your section heading in your report will read "ADD – 1". If you implemented several simultaneously, just pick a logical ordering.
- A discussion of the modifications to the previously implemented instructions' hardware needed to implement the new instruction (including a screenshot or hand drawing of the added hardware)
- The control signals you had to add and all of the values the control signals must have for the new instruction to operate correctly
- An annotated simulation output that proves that the new instruction works correctly. ***Important!*** You can use the Lab3 mif you generated from the lab assignment 3 in your annotated simulations to show the instruction's functionality. However, it may be easier for you to write your own mif file to show the instruction's functionality. If you use your own file, make sure to cover all corner cases.

If an instruction can be implemented with the previous instruction's hardware with no modifications, state this but still discuss the values the control signals must have for the new instruction to operate correctly and provide an annotated simulation output that proves that the new instruction works correctly. This is the most important part of your report, *so please take it seriously*. Without this part of your report, there is no

way of knowing if you implemented each instruction correctly; more explicitly, without this part, there is no way to give you partial credit if your entire processor does not function perfectly.

Provide a detailed diagram of all interconnects in your system (similar to figure 4.17).

2.2: In-Lab DEMOS

In-Lab Checkpoint Demo: Implement at least one R-Type and at least one I-Type instruction and demonstrate their functionality no later than the **checkpoint demo date**.

Final Demo: Demonstrate the functionality of your processor using the Lab3.mif file you generated from your assembler in lab assignment 2.

Include a section in your report where you annotate the functional simulation output of your processor running the program.

Timing analysis of your processor should be done with the **Cyclone IV Gx - EP4CGX150DF3117** FPGA as your target device. Part of your demo grade will be based on the maximum clock rate listed in the timing analyzer report. Frequencies above 45MHz will receive extra credit. Anything below 25MHz will receive a zero for the final demo.

Question 2.2.1: How many cycles does your program take to execute?

Question 2.2.2: At the end of the program, what are the contents of the memory locations: 0x40000808, 0x4000080c, 0x40000810, 0x40000814.

Question 2.2.3: At the end of the program, what are the contents of registers: $8, $9, $30?

2.3: Notes for Design

Below is a list of things to keep in mind while designing your processor:

- Your instruction memory should be implemented with an "altsyncram" component, a 32-bit output bus, an 8-bit address bus, a 256 word capacity, and mapped to the memory block beginning at address 0x00400000.
- Your data memory should be implemented with an "altsyncram" component, a 32-bit input/output bus, an 8-bit address bus, a 256 word capacity, and mapped to the memory block beginning at address 0x10000000. Your data memory should also have ram enable, write enable, and byte enable signals.
- Register zero should always have the value zero so you need to alter your register file from assignment 3 so that register zero cannot be modified.
- Because the instruction memory is mapped to the 256 word memory block beginning at address 0x00400000, if you use your reg32 component from assignment 1 for your program counter you will need to modify it so that the value of the program counter is 0x00400000 after a reset.
- The shift instructions shift the contents of Rt not Rs as the green reference sheet suggests.
- The load instructions are I type instructions and therefore do not have an opcode of zero as the green reference sheet suggests; what the sheet lists as the function code is actually the opcode.

This list may be updated as questions arise.

Section 3: Report

Each of the remaining assignments should follow this format:

1. The report should contain the following sections
    - Cover page – with your name, date, and assignment number
    - Introduction – briefly comment on the assignment and what it accomplishes

- Instruction functionality (for each instruction):
    - Description of the instruction and its function.
    - Tests that were performed to verify that the components/instruction worked correctly. Provide functional simulation waveforms and make sure to describe the test they represent. In other words, ANOTATE!
- Demo Program:
    - Annotated simulation of the output of your processor from the lab3 mif file used in the final demo.
    - Answers to questions in the lab section.
- Appendix

2. The report needs to be typed. Annotations should not be scans of hand-drawn annotations and illegible annotations are almost as bad as no annotations.
3. Any waveform that is not labeled and annotated will not be considered as valid.
4. Make sure you use the names for inputs, outputs, and entities as outlined in the assignment. This will be essential for connecting these components together in later assignments.

Grade Breakdown

Section	**Deliverables**	**Percentage of Total grade**
Demo	Demo 2 instructions to the TA by the demo date (5 points each)	10%
Final Demo	Demonstrate your processor correctly running the Lab3.mif file	15%
2.1	Instruction functionality for all 29 instructions (2 points each)	58%
2.2	Lab3.mif decode	15%
Report	Report follows the format given in section 3 (1%), Report is neat and professional (1%).	2%