CS M151B/EE M116C - Homework #2, Due 4/17/2017

Reading Assignment:

- *Readings from hw #1 — Appendix B: pp. B-26 – B-37, Chapter 4: pp. 244-272*
- Chapter 2: pp. 96-106 (procedure calls)
- Preparation for next week
  - Appendix B: pp. B72-B74 (timing)
  - Chapter 5, of the **3rd Edition**: pp. 318 - 330, 331 - 339 (skip example on pp. 330-331)
    Note: This is from the **Third** Edition of the Patterson & Hennessy book. This chapter is available in the ‘‘Supplementary Readings’’ part of the class web site.

Problems:

(1)-(2) B.24, B.25 (on page B-83, Appendix B).

(3) The single cycle implementation of the MIPS processor uses for the main ALU the implementation shown in Figures B.5.10-B.5.12 (pp. B-33 – B-36) in the book. Due to a circuit malfunction, one of the four ALU control lines is always one (stuck-at-1). Specifically, refer to Figure B.5.13, number the control bits in little endian order, with the least-significant bit being bit-0. The faulty bit described above is **bit-1**.
The rest of the circuitry of the ALU and the rest of the processor operates normally.

Explain **in full detail** what will be the consequences of this fault when the processor executes programs — how will it change the behavior of the processor **as observed by a user/programmer** who does not know and does not care how the processor is implemented internally? Be sure to clearly identify **each and every** consequence of this fault.

(4) Problem 4.6.4 in the textbook.
Note that you need to read Problem 4.6, above 4.6.1 on page 359.

(5) Problem 4.6.5 in the textbook.
Note that you need to read Problem 4.6, above 4.6.1 on page 359.

(6) Consider the single cycle implementation shown in Figure 4.24 (page 271). Assume that during one particular cycle, regardless of the inputs to the control unit, the control unit generates the following values for the control signals:

`0001000001`

The values of the control signals are specified in order from top to bottom, as they appear in the figure: the left-most bit is the value of the RegDst signal and the right-most bit is the value of the RegWrite signal. The most-significant bit of ALUOp appears to the left of the least-significant bit of ALUOp.

Specify all the state changes that will occur over this one cycle. Be sure to indicate which (if any) state elements will change their state and to what will be the new value of each of those state elements. Your answer must be as specific as possible. Explain your answer.

(7) Note: Review the solutions to practice problems 11 and 12 as examples for how to solve problems like this one.

Add a new instruction to the single-cycle implementation described in Chapter 4. Note that this new instruction is **not** part of the standard MIPS ISA. The new instruction is `lwrr` (load word register-register). This is an R-format instruction that has similar functionality to the standard `lw` instruction. The difference is that the address is the sum of the value in the `rs` register and the `rt` register. The value is loaded from memory to the register specified in the `rd` field.

Your starting point is the datapath shown in Figure 4.17 on page 265 and the control shown in Figure 4.18 on page 266.

Key building blocks of the datapath are implemented as shown in Appendix B: the ALU in Section B.5 and the register file in Section B.8.

All the instructions supported by the design in Figures 4.17 and 4.18 must continue to work correctly. You should minimize the changes to the datapath and to the control. You need to show all the required changes to Figure 4.17 and Figure 4.18. In case you need them, you can find these figures on the class Web page (''Useful Figures'').

If any modifications to the datapath building blocks (ALU, register file, etc) are necessary, you must show the implementation of the modified building blocks **in detail**.

(8) Consider the **single** cycle MIPS implementation with the datapath shown in Figure 4.17 (page 265) in the book and the control logic implemented based on the truth table in Figure 4.22 (page 269).

Your task is to modify this implementation to add support for the `jr` (jump register) instruction, which is part of the MIPS ISA. This instruction is described on page A-64 in the book.

A) Show any required modifications to the datapath. You can show the modifications on a copy of Figure 4.17 (available on the class web page, *Useful Figures*).
Also explain the modifications in words, to make sure your modifications are clear.
In addition, if you need to modify any of the datapath building blocks, draw the implementation of the modified building blocks and explain the modifications in full detail.

B) Are any new control signals required? If so, list them with an explanation and identify them on the datapath diagram.

C) Show any necessary changes to the control logic by presenting a modified version of the control truth table in Figure 4.22 (available on the class web page, *Useful Figures*). For any entries that you add, mark all ''don't cares'' with an **X**.

Practice problems: You do not need to hand in a solution to the problems below.

(9) Consider the ALU shown in Figure B.5.12 (page B-36). What function will this ALU perform if the ALU control lines are set to `0101`. Explain your answer.

(10) Consider the MIPS ALU implementation shown in Figures B.5.10-B.5.12 (pp. B-33 – B-36). A change in the implementation is made to Figure B.5.11: the `Less` input to the `ALU3` cell is set to 1 instead of 0. Explain **in detail** what will be the consequences of making this change when the processor executes programs (i.e., how will it change the behavior of the processor **as observed by a user/programmer** who does not know and does not care how the processor is implemented internally?).

(11) Similar to problem (7) above, except that you need to add the `addi` instruction.

(12) Similar to problem (7) above except that you need to add a new instruction, which is not part of the standard MIPS ISA.

```
swap $5,$9
```

After the `swap` instruction is executed, register $5 contains the original value of register $9 and register $9 contains the original value of register $5.

Note that the first step is to decide on the instruction format to be used for the `swap` instruction. Choose a format that will lead to a simple implementation.

(13) Consider a proposal to eliminate the control signal MemtoReg. The multiplexor that has MemtoReg as an input will instead use the control signal MemRead. Will this modification work with the single cycle implementation? Explain your answer.

(14) Consider the single cycle MIPS implementation shown in Figure 4.24 (page 271). Due to a hardware fault (malfunction) in the ALU, the Zero output from the ALU is **always 1**.
Explain **in full detail** what will be the consequences of this fault when the processor executes programs — how will it change the behavior of the processor **as observed by a user/programmer** who does not know and does not care how the processor is implemented internally? Be sure to clearly identify **each and every** consequence of this fault and be as specific as possible.

(15) Similar to problem (7) above, except that you need to add the bne instruction (described on page A-62).

(16) Problem 4.6.1 from the textbook. Note that you need to read Problem 4.6, above 4.6.1 on page 359.

(17) Problem 4.6.2 from the textbook. Note that you need to read Problem 4.6, above 4.6.1 on page 359.