CS151B/EE116C - Homework #4, Due 4/30/2018

Reading Assignment:

- *Readings from hw #3 — Ch 1: pp. 28-40, 46-48.*
- Chapter 5, **3rd Ed**: pp. 315-317, 330-331
- Chapter 4: pp. 272-325.
- Preparation for next week:
  - Section 2.21, pp. 2.21-1 – 2.21-6. This material can be found on the textbook web site. A copy is also available in the ''Supplementary Readings'' part of the class web page.
  - ''Instruction Set Styles'' and ''The IBM/Motorola PowerPC'' from the CD of the **3rd Edition** of the book. This material is available in the ''Supplementary Readings'' part of the class web page.

Problems:

(1) Problem 1.7 in the book.

(2) Consider the performance of a processor with a clock frequency of 1GHz on a suite of benchmarks. Two different sets of measurements are made: A) the programs are compiled by a compiler, and B) the programs are written in assembly by a highly-skilled programmer. The highly-skilled programmer manages to get a reduction of a factor of 2.45 in the CPI and a performance improvement of a factor of 10.1 compared to the compiled programs. What is the reduction in instruction count obtained by the highly-skilled programmer?

(3) When a processor executes program P_A, the execution time is 3.8 seconds and the CPI is 3. When the same processor executes program P_B, the execution time is 8.5 seconds and the CPI is 3.5.
A typical workload consists of executing P_A, then executing P_B. What is the CPI for this typical workload?

(4) Consider a processor that achieves an overall average CPI of 3.6 when executing a program P1. 18% of the instructions of P1 perform floating point operations. For these floating point instructions, the average CPI is 5.3.

By improving the design of the floating point ALU, the average CPI for floating point instructions is reduced from 5.3 to 3.

A) What is the overall average CPI for the improved processor?

B) If the original processor executed P1 in 220 seconds, how long will it take the improved processor to execute P1?

(5) Consider the multicycle MIPS implementation from chapter 5, **3rd Ed**, (Figure 5.28). Assume that the processor executes a long sequence of consecutive lw instructions or a long sequence of consecutive sw instructions. For each of these cases we would like to use pipelining to reduce the CPI by 1. The approach is to fetch the next instruction (instruction fetch step in Figure 5.30) during the execution of the previous instruction.

A) Explain the basic idea of your modifications in 2-4 clear sentences.

B) Show the necessary modifications to the datapath. If modifications are required, show the entire modified datapath (Figure 5.28 modified as necessary). If you need to modify any of the datapath building blocks, draw the modified building blocks and explain the modifications.

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

D) Modify the control unit state diagram (Figure 5.37) to reflect the changes.

(6) 4.8.2 in the book.
Note: The ‘‘nonpipelined processor’’ is the **single** cycle MIPS implementation.

(7) 4.8.3 in the book.

(8) 4.9.1 in the book.

For each dependence indicate:

- Which instruction is dependent on (must be executed after) which instruction. For this purpose, label the instructions within each of the program segments: i1, i2, i3.
- What is the type of the dependence: RAW, WAR, or WAW.
- What is the storage location (for example, the register number) through which there is the dependence.

(9)-(10) 4.9.2, 4.9.3 in the book.

(11) Consider the pipelined MIPS implementation shown in Figure 4.51 (page 304). For each of the following two instructions separately, for each of the pipeline stages, specify the values of the control signals that are asserted for the instruction, when the instruction reaches that stage. Note that you do not need to specify control signals that are not asserted.

a. `sw $14,44($3)`

b. `sub $21,$9,$2`

Practice problems: You do not need to hand in solutions for the problems below.

(12) Problem 1.5 in the book.

(13) In the embedded market, where cost is crucial, processors sometimes implement floating point only in software. We are interested in two implementations of a computer, one with and one without special floating-point hardware. Consider a program, P, with the following mix of operations:

Floating-point multiply	10%
Floating-point add	15%
Floating-point divide	5%
Integer instructions	70%

Computer MFP (computer with floating point) has floating-point hardware and can therefore implement the floating-point operations directly. It requires the following number of clock cycles for each instruction class:

Floating-point multiply	6
Floating-point add	4
Floating-point divide	20
Integer instructions	2

Computer MNFP (computer with no floating point) has no floating-point hardware and so must emulate the floating-point operations using integer instructions. The integer instructions all take 2 clock cycles. The number of integer instructions needed to implement each of the floating-point operations is as follows:

Floating-point multiply	30
Floating-point add	20
Floating-point divide	50

Both computers have a clock rate of 1000 MHz. Find the native MIPS ratings for both computers.

(14) If computer MFP in Problem 13 above needs 300 million instructions for this program, how many integer instructions does computer MNFP require for the same program?

(15) Assuming the instruction mix in Problem 13 above and a total of 5×10^9 instructions required on MFP for program P, what is the execution time (in seconds) for P on MFP and MNFP?

(16) Consider the multicycle MIPS implementation **from chapter 5** (Figure 5.28). Assume that the processor executes a long sequence of consecutive R-type instructions. We would like to use pipelining to reduce the CPI by 1. The approach is to fetch the next instruction (instruction fetch step in Figure 5.30) during the execution of the previous instruction.

A) Explain the basic idea of your modifications in 2-4 clear sentences.

B) Show the necessary modifications to the datapath. If modifications are required, show the entire modified datapath (Figure 5.28 modified as necessary). If you need to modify any of the datapath building blocks, draw the modified building blocks and explain the modifications.

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

D) Modify the control unit state diagram (Figure 5.37) to reflect the changes.

(17) 4.8.1 in the book.
Note: The ''nonpipelined processor'' is the **single** cycle MIPS implementation.

(18) Consider the MIPS implementation timing described in Figure 4.26, page 275.

a) If the time for an ALU operation is shortened by 25%, will this change affect the speedup obtained from pipelining? If yes, by how much? Otherwise, why?

b) Repeat part **a**, where the change is that the time for an ALU operation is increased by 25%.

(19) Using a drawing similar to Figure 4.29 on page 279, show the forwarding paths needed to execute the following program segment as quickly as possible:

```
add  $3, $4, $6
sub  $5, $3, $2
lw   $7, 100($5)
add  $8, $7, $2
```

(20) Identify all of the data dependences in the following code. Which dependences are data hazards that will be resolved via forwarding? Which dependences are data hazards that will cause a stall?

```
add  $3, $4, $2
sub  $5, $3, $1
lw   $6, 200($3)
add  $7, $3, $6
```

(21) We have a program of 10^3 instructions in the format of ''`lw, add, lw, add, ...`''. The `add` instruction depends (and only depends) on the `lw` instruction right before it. The `lw` instruction depends (and only depends) on the `add` instruction right before it. If the program is executed on the pipelined datapath of Figure 4.60 on page 316:

a. What would be the CPI?

b. Without forwarding, what would be the CPI?