COMP1047 Lab Week 02

Part 1: Computer Performance

Q1. Consider three different processors P1, P2, and P3, executing the same set of instructions. P1 has 3GHz clock rate and a CPI of 1.5; P2 has a 2.5GHz clock rate and a CPI of 1.0; P3 has a 4.0GHz clock rate and has a CPI of 2.2.

(i) Which processor has the highest performance expressed in instructions per second?

(ii) If the processor each execute a program in 10 seconds, find the number of cycles and the number of instructions.

(iii) We are trying to reduce the execution time by 30% but this leads to an increase of 20% in the CPI. What clock rate should we have to get this time reduction?

Q2. Compliers can have profound impact on the performance of an application. Assume that for a program, complier A results in a dynamic instruction count of 1.0E9 and has an execution time of 1.1s, while complier B results in a dynamic instruction count of 1.2E9 and an execution time of 1.5s.

(i) Find the average CPI for each program given that the processor has a clock cycle time of 1ns.

(ii) Assume the compiled programs run on two different processors. If the execution times on the two processors are the same, how much faster is the clock of the processor running complier A's code versus the clock of the processor running complier B's code?

(iii) A new complier is developed that uses only 6.0E8 instructions and has an average CPI of 1.1. What is the speedup of using this new complier versus using complier A or B on the original processor?

Part 2: Getting Familiar with QtSpim

We will program the MIPS assembly code using an IDE named QtSpim. Please download and install from (https://sourceforge.net/projects/spimsimulator/files/). Some useful information can be found at (http://spimsimulator.sourceforge.net/). You may also see that QtSpim has already been installed on the computers in our lab room.

Try your first MIPS program

Once you are familiar with the software, you can now try to execute some simple MIPS programs. The first one you could try is the famous "Hello World", which almost every programmer will use as their first program. To do this, firstly, create a new text file and named it "HelloWorld.asm" or "HelloWorld.s". Open the file using any text editor (e.g. notepad) and copy/paste the following text into the file:

```
        .data 0x10010010
msg:  .asciiz "Hello world!\n"
        .text
        .globl main
main:
         la $a0, msg     #load label msg
         li $v0, 4          #load immediate
         syscall          # print it
         li $v0, 10
         syscall          # exit
```

Save the file before you close the text editor. Now load the program that you just created into QtSpim by clicking the Load File in File menu. Click the Run/Continue button (i.e. the green triangle icon ▶) on the menu bar and observe what you get in the console window.

In the next part, you are supposed to write some MIPS code using the instructions learnt in this week's lecture. But before that, you are strongly suggested to:

- View the demo video from Moodle page (QtSpim_Demo), to familiarize yourself with the QtSpim tool and some example MIPS code.
- Look at the "QtSpim Syscall" file in the Moodle page to get more experience from the syscall that is used together with your MIPS instructions.

At the end of this lab manual, find some common Q&A regarding the QtSpim.

Part 3: MIPS Programming Questions

Q1. Write a MIPS program to load two integers, 45 and 1026, into registers $s0 and $s1 as unsigned numbers. You can place the binary value directly in the data memory segment and then use `lw` instruction to load them into registers. For example, the following program stores two unsigned integers 0000000A16 and 1000000016 in the data segment of the memory, and then loads the first integer in $s0 using `lw` instruction.

```
      .data
uint: .word 0x0000000A 0x10000000
      .text
      .globl main
main:
      la $t0, uint          #load the base address
      lw $s0, 0($t0)        #load the first integer into $s0
```

Note that here we use assembler directive **.word**. It means each number in the following is treated as a 32-bit word. Now print out both numbers to the console using the **syscall** function. Check the output to see whether it is expected.

Q2. Write a program in MIPS assembly language which reads two integer numbers x and y from the console, calculates, then prints x – 2y – 40. *Hint: no multiplication is necessary and proper user prompts are expected.*

To read an integer from the console:

```
li  $v0, 5   # read_int
syscall
# after syscall, $v0 contains the number just entered
```

To print an integer to the console:

```
# $a0 contains the number to be printed
li  $v0, 1   # print_int
syscall
# after syscall, console with print the value contained in $a0
```

Some Common Q&A for Part 2 – Sorted from current and previous semesters

Q1: Where can I write code, and how to run it?

A: You can write code with any text editor, e. g. notepad, sublime, vim, and use QtSpim->file->open to load the file, then click '>' to run. The output will show in console.
If you close your console, find it by 'window->console'.

Q2: Can I use MARS (or other software) but not QtSpim to run MIPS code?

A: You can use it but we still suggest for QtSpim, because in very few cases, MARS could get different result with same code. So if you prefer MARS, remember to double-check with QtSpim before submitting your coursework.

Q3: Why I get error message - 'double define 'main'' when I want to load and run my program again?

A: That's caused by double define of .asm/.s file. Try 'file->reinitialize and open' to load your code file and run it again.

Q4: Why I get compile error?

A: It depends, firstly I suggest you to check if you use full version of comma in Chinese. Please use ',' (English comma) but not '，' (Chinese comma).

Q5: I change the msg from 'helloworld' to 'hellocomp1047', but output didn't change in QtSpim console.

A: Please save the code file, then reload & rerun.

Q6: I finish today's lab, but still confused about those code, like 'li' and 'syscall'.

A: That's OK for the first lab, as we are just begining to learning how to write MIPS code. Details for each instruction are introduced in 'MIPS Reference Card', which I will uploaded to moodle. And more information will be included in future lectures & labs, like how 'syscall' works and what does 'li $v0, 1' means.

Q7: I always get strange results or errors, but my code seems all good and correct.

A: That might be caused by number encoding standard of input method. Please switch your keyboard input method to English, not English style on other method (e. g. English mode of Microsoft Chinese Keyboard).

Q8: In QtSpim, backspace doesn't work. A strange char is inserted instead.

A: Simply, do not use it.

This issue has been proposed online, and the maintainer of QtSpim - James Larus have replied in 2011 by "This is by design. The console runs in raw mode, not allowing editing." on https://sourceforge.net/p/spimsimulator/bugs/16/

For more details about so-called 'raw mode', see:
https://unix.stackexchange.com/questions/21752/