

Exercises for “Computer Science II” — SS 2003

No. 3

Due: May 24, 2003

Computer Systems Organization

1. Explain why Program Counter is changed in the second step of instruction execution. 1) What would happen if this step is missing? 2) What happens if an interrupt comes? **(3 points)**
2. A computer has a bus with a 25 nsec cycle time, during which it can read or write a 32-bit word from memory. The computer has an Ultra-SCSI disk that uses the bus and runs at 40 Mbytes/sec. The CPU normally fetches and executes one 32-bit instruction every 25 nsec. How much does the disk slow down the CPU? **(1 points)**
3. Suppose that a CPU has a level 1 cache and a level 2 cache, with access time of 5 nsec and 10 nsec, respectively. The main memory access time is 50 nsec. If 20% of the accesses are level 1 cache hits and 60% are level 2 cache hits, what is the average access time? **(1 points)**

Introduction to MIPS Programming

Download and install SPIM from <http://www.cs.wisc.edu/~larus/spim.html>.

Study the realplayer tutorial files from

<http://users.ece.gatech.edu/~sudha/2030/temp/spim/spim-tutorial.html>,

and the demo program `first-spim.s`. Following the steps below and answer subsequent questions. **(5 points)**

A. Download a program `second-spim.s` from

<http://user.informatik.uni-goettingen.de/~fu/teaching/Info-II/SS2003/second-spim.s>,

print it out and read it. Pay attention to the comments, and familiarize yourself with how it works. This program should look similar to a simple example done in lecture when discussing datapaths.

B. Run SPIM. Select Simulator->Setting. Make sure all 3 Display Settings (save window positions, general registers in hexadecimal, floating point registers in hexadecimal) are checked. Make sure the only execution setting that is set is “Allow pseudo-instructions.”

All Others (especially load trap file) should be unchecked. If using the UNIX version, run the program by typing `xspim -notrap`. Note that the instructions below are for PCSPIM, the xspim interface has all the necessary windows already showing.

C. Make sure all of your windows (Message, Register, Data Segment, Text Segment) are visible. If not, select each one using the Window Menu. Be careful: if no windows have been selected, SPIM will load successively selected windows on top of each other.

D. Load `second-spim.s`.

E. Look at the program code in the Text Segment starting at `0x00400000`. The left side of each line shows the address, instruction code, and corresponding assembler code. After the semicolon (a comment) is the assembler code that was specified in the source code for `second-spim.s`.

F. Notice that in some cases the specified assembler instruction is not the same as the instruction generated by the assembler! These are pseudo-instructions – instructions that make it "easier" for you to write assembler that are actually translated into an equivalent instruction. For example, look at line `0x00400008`.

4. What address range is the data segment for this program? Where does the text segment start?
5. What is the original instruction specified in the source code?
6. What is the instruction that was actually generated?
7. Why do you think this particular substitution might have been made? (Hint: recall your single-cycle datapath)