Indian Institute of Technology Tirupati

CS2610 Assembly Language Programming
Credit: L-T-P-C : 0-0-3-2
Instructor: Jaynarayan T Tudu <jtt@iittp.ac.in>
Experiment 3: Assembly Programming in MIPS for Arithmetic and Logic Instruction

1 Guidelines

The objective of the Experiment 3 is to understand the MIPS registers and the ALU instruction set. The Experiment 3 involves the following two different task:

1. To learn to execute the program written in MIPS assembly code in the QtSPIM simulator. As part of this exercise you are required to execute the program that has been done as part of the Experiment 2 and collect the values of each registers as the program moves step by step. At least five programs needs to executed. This exercise should be finished in the first one hour of your laboratory time.
2. To learn to write the assembly program in MIPS ISA (instruction set architecture). The program to be written are listed in the following section.

2 Programs

In this Experiment you need to use only the arithmetic and logic instructions, you may use call and return instruction as part of housekeeping task.

1. Swapping the register content: Write a program to swap the content of two registers (you have to chose any two general purpose registers from \$t0 to \$t7, \$s0 to \$s7, \$t8 to \$t9) using the following methods:
    - Using third intermediate register but without using any *move* instructions (think if you can use add instruction to move data).
    - Without using third temporary register and using only the addition and subtraction instructions.
    - Without using third temporary register and without using addition and subtraction instruction (give a thought if you can use any of logical operations).
2. Play with polynomial: Write a program using only the arithmetic and logic instruction set to compute the following polynomial: $ax^2 - bx + c$ where value for a, b, c, and x can be any positive integer number.
3. Doing without:
    - Write a program to multiply two numbers a and b using only the shift instructions. Assume that one of the number is either 2, 4, or 8. Write two version program where in the first one you use the register addressing mode (i.e R type instruction format) and in the second one you use immediate addressing mode (i.e I type instruction format).

    Write a program to perform division operation on a as dividend and b as divisor using shift instructions assuming that the dividend is either 2, 4, or 8. Check (manually) for what value of a you obtain the exact division results.
4. The Simple Logic:

    Write a program or a set of programs to implement the bit-wise logical operations: AND, OR, XOR, XNOR, AND Masking, Or Masking, Reset to zero with XOR, one's complement, (try write 2's complement but don't use add instruction).

3 QtSPIM Tools

The Experiment 3 will be carried out using QtSPIM tool. QtSPIM is a MIPS simulator which could be run on x86 machines. To start with QtSPIM following the guideline below:

1. open the terminal and type **qtspim** to open the tool
2. In the QtSPIM window you will the code window and register window by default. It will also open a console which is not needed so you can close it. In the register window you will see a list of registers which are a part of MIPS32 architecture. You may need to explore by reading the Help manual which will appear in top menu bar. In the program window you will see a set of MIPS32 assembly instructions which are in-built for housekeeping task like initialising the user program for execution etc. Your program will be accommodated somewhere in the list of instructions.
3. Next, use the load program menu to load your assembly program to the QtSPIM simulator. Try to run by clicking on run menu (the arrow symbol), you may see error message if your program is not written in proper format. Correct those errors and reload the program, before reloading you need to re-initialise the registers.
4. When you execute the program that is compiled using mips cross compiler you may find a lot of error messages with respect the levels (all the levels starts with . (dot)). You may disable those levels one by one using # symbol.
5. One your program is loaded successfully now you are ready run the program. There are two ways to run the program: one is direct run and the other is step. In this lab we will use step (the grid symbol in menu bar) such that the intermediate value can be observed.

For further details on use and other stuff you may visit the QtSPIM home page: updated one [1], the old one:[2].

The assembler and QtSPIM details could be found in your reference book [3]

4 Report

You have to write a report for two programs with the following details. The report should be handwritten with the assembly program and a table which depicts the steps and change in register values as follow.

	Step 1	Step 2	Step 3	Step 4	Step n
PC	0x02ab78	0x02ab7c			
R0	10	20			
R1	...	...			
R2					
R10					
R11					
R13					
R12					
R14					
R15					
R28					
R29					
R30					
R31					
..					

[1] http://spimsimulator.sourceforge.net/

[2] http://pages.cs.wisc.edu/ larus/spim.html

[3] Appendix A, Computer Organisation and Design The Hardware Software Interface, 5th Edition, MIPS version

5 Evaluation

Demonstration + Viva: 2 + 1 points,
Report: 2 points

6 Timeline

Demonstration: 5:30 pm
Uploading of programs in github: 6:00 pm
Report: 15th Feb, 2018