

David J Tinley
09/24/2023
Lab 1 Report
Computer Organization

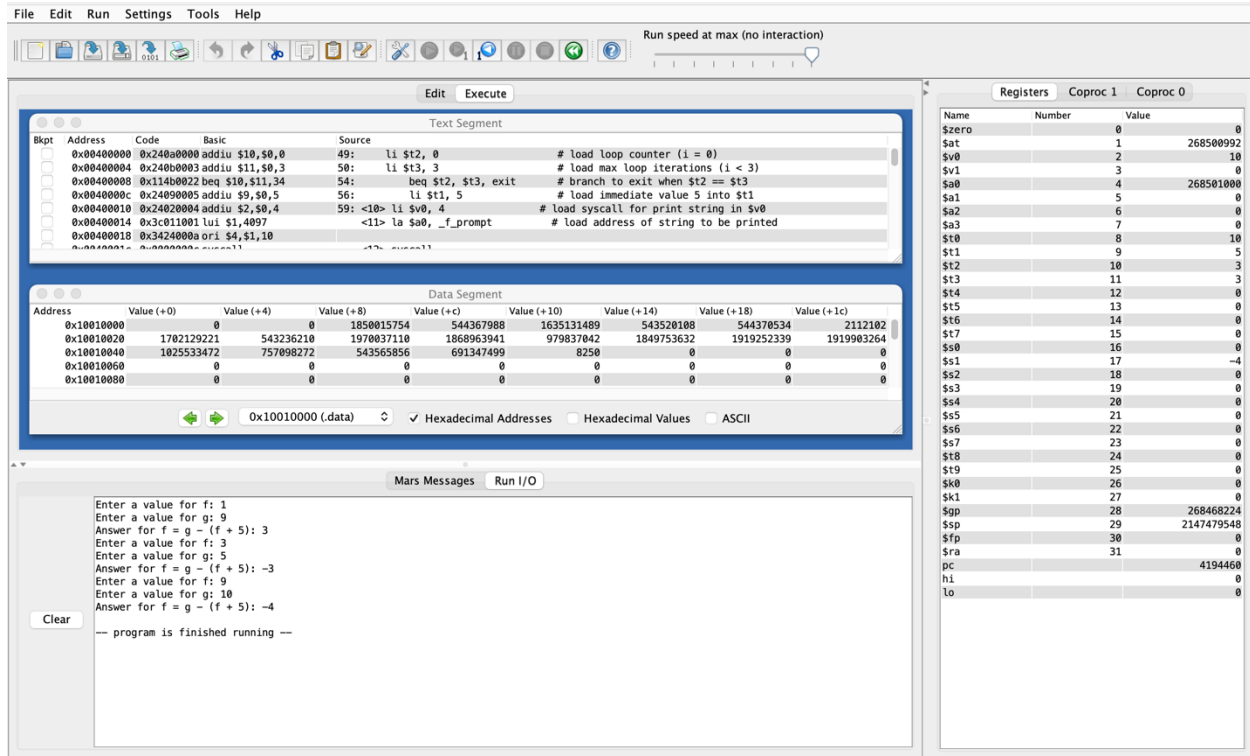
For this lab assignment I was able to utilize macro's that I had learned about online. They are essentially functions that you can write instead of having to write each line of assembly code over and over for things like print or read in the terminal. The second concept I used was a loop. The MIPS loops were like using for loops in C/C++. A big difference being you must declare your loop iterator and maximum loop counter outside of the loop. You also must explicitly tell the program where to branch to after the loop is finished.

The assembly code file for this project is called "lab1.asm"

The screenshot displays the MARS MIPS simulator interface. The top menu bar includes File, Edit, Run, Settings, Tools, and Help. Below the menu is a toolbar with various icons for file operations and execution. The main window is divided into several panes:

- Text Segment:** Displays assembly code with addresses, hex codes, and comments. The code includes instructions like `addiu $t0, $0, 0`, `addiu $t1, $0, 3`, `beq $t2, $t3, exit`, `li $t1, 5`, `li $v0, 4`, and `la $a0, _f_prompt`.
- Data Segment:** Shows a table of memory addresses and their corresponding values. The table has columns for Address, Value (+0), Value (+4), Value (+8), Value (+c), Value (+10), Value (+14), Value (+18), and Value (+1c).
- Registers:** A table on the right side of the window showing the state of MIPS registers. It includes registers like `$zero`, `$at`, `$v0`, `$v1`, `$a0`, `$a1`, `$a2`, `$a3`, `$t0`, `$t1`, `$t2`, `$t3`, `$t4`, `$t5`, `$t6`, `$t7`, `$s0`, `$s1`, `$s2`, `$s3`, `$s4`, `$s5`, `$s6`, `$s7`, `$s8`, `$s9`, `$k0`, `$k1`, `$gp`, `$sp`, `$fp`, `$ra`, `pc`, `hi`, and `lo`.
- Mars Messages:** A pane at the bottom left showing assembly messages, including "Assemble: assembling /Users/djt/Desktop/mips/source_files/lab1.asm" and "Assemble: operation completed successfully."

The simulator is running at "Run speed at max (no interaction)".



In conclusion, I think the project went well. The biggest issue I faced was understanding how memory alignment works within the code. I was eventually able to figure how the memory must be aligned differently for words, half-words, and so on. I do not fully understand why the alignment call had to be positioned exactly where I had put it in the code though.