University of New Hampshire
Department of Electrical and Computer Engineering
ECE 562 – Computer Organization
Fall 2019
Lab #2

Writing a Complete Assembly Program for Blinking an LED

In this assignment, you will develop your assembly programming skills and learn about bit manipulation, memory-mapped I/O, and controlling GPIO.

Step 1 – Prepare for the Lab

Make sure you download and read the following files: `Lab2-notes.pdf`, `Makefile`, and `lab2.s`. You might need to refer to the `BCM2837-ARM-Peripherals-<version>.pdf` for following the comments in `lab2.s`. You will also need `ft232h.cfg` and `rpi3.cfg` installed in the same directory.

Step 2 – Write the Assembly Code

Read the comments in `lab1.s` carefully and fill in your code. You will write over 30 instructions. Each time you add a few instructions, try running `make` to see if the assembler or the linker give any errors.

Step 3 – Emulation (optional)

Once you have completed your code, you can try executing it under the emulator, attach a debugger to step through instructions and view the register contents etc. to see if your code is working as expected.

Step 4 – Setup in the Lab

Only some of the computers in the lab have the necessary board next to them. Find an available computer and login as user `ece562` with password `ece562`. Transfer or download your files to `c:\ece562`. If you find the lab files of another student, please delete them without opening. Make sure to always keep your copy on either a freshly formatted flash drive, a remote machine (e.g. FTP), or in cloud storage (e.g. Box). DO NOT expect to find your files if you come back to the lab.

Step 5 – In the Lab

You can edit your files by right-clicking and selecting `Open With->Notepad++`. When you want to run your program on the Raspberry Pi, turn the board on using the switch on the cable, connect the USB cable to the computer, open three Ubuntu terminals. In each one, switch to the directory with `cd /mnt/c/ece562` command. On the first one, you can build your program by running `make`. On the second one, you can run OpenOCD by running `make run-openocd`. On the third one, you can run `aarch64-elf-gdb` and use the GDB commands to connect to OpenOCD, load the program into memory, set the program counter, and execute the program.

If you then make changes to the `lab1.s` under `c:\ece562`, you will need to run `make` on the first window again, then switch back to third window to load the program into memory again and set the program counter again before you can run the updated program.

Once you have completed running your program on the board, close the terminal windows, then turn off the board from the switch on the cable and unplug the USB from the computer. Again, make sure to transfer, or upload your modified files back to a safe place and delete the copies of your files from the computer. Finally shut down the computer.

Step 6 – Submit

Submit your `lab2.s` file online.