CSCI210 Computer Architecture and Organization
Lab Assignment

Lab: Write ARM assembly to toggle 4 LEDs in sequence . . . back and forth.

Cross Compiler: A cross compiler is a compiler capable of creating executable code for a platform other than the one on which the compiler is running. For example, a compiler that runs on a Windows 7 PC but generates code that runs on Android smartphone is a cross compiler.

A cross compiler is necessary to compile code for multiple platforms from one development host. Direct compilation on the target platform might be infeasible, for example on a microcontroller of an embedded system, because those systems contain no operating system. In paravirtualization, one computer runs multiple operating systems and a cross compiler could generate an executable for each of them from one main source.

Download ARM's cross compiler here:

https://developer.arm.com/open-source/gnu-toolchain/gnu-rm/downloads

Raspberry Pi Boot Process:

https://wiki.beyondlogic.org/index.php?title=Understanding_RaspberryPi_Boot_Process

https://www.raspberrypi.org/documentation/configuration/config-txt/README.md

start.elf	=> 3rd stage bootloader
bootcode.bin	=> 2nd stage bootloader

bootcode.bin loads start.elf

Steps:

1. Install the cross compiler
2. Erase and Format SD card to FAT32
   a. **Mac:** https://www.michaelcrump.net/the-magical-command-to-get-sdcard-formatted-for-fat32/
   b. **Linux:** Use gparted to make one FAT32 partition and ensure that the Boot Flag is enabled.
   c. **Windows:** Use SD Formatter https://sd-card-formatter.en.uptodown.com/windows
3. Study documentation to determine appropriate registers and bit locations for the available GPIO pins. GPIO descriptions begin on page 89 of the **BCM2837 ARM Peripherals** document
4. Write the ARM assembly on a non-Pi machine
5. Use the provided make file to build the kernel image
6. Copy the following files to the SD card

   a. kernel.img
   b. bootcode.bin
   c. start.elf
   d. fixup.dat
   e. kernel.ld

7. Eject the SD card from the development machine and insert into Raspberry Pi
8. Cross your fingers and boot the Pi . . . you will either see LEDs blinking or not
9. Lather, rinse, repeat until you are successful

Code Requirements:

- No redundant code: Generalize your code as much as possible by writing procedures to manipulate the GPIO registers. Pass pin numbers as arguments.

Final Submission:

- Demonstrate to me in person that your code works
- Submit your source file to Blackboard