CSE4372/5392 (Spring 2025)
Lab #1

This lab is due by January 28, with a 20% penalty per week day for being late.

In this lab, you will install Vivado ML and the RISC-V GNU cross compiler.

The following steps will guide you through this process:

1. Download and install Vivado ML from the AMD Download Center. This will take an hour after an account is setup.

2. Copy the provided files to a combo_logic directory.

3. Compile the code and program the device.

4. Verify the program operates as expected.

5. Clone the RISC-V GNU toolchain code
sudo apt install git
sudo git clone https://github.com/riscv/riscv-gnu-toolchain
(suggested target /opt/riscv-gnu-toolchain)

Configure the toolchain
sudo apt install gawk
cd /opt/riscv-gnu_toolchain
sudo ./configure --prefix=/opt/riscv --enable-multilib --with-arch=rv32i –with-abi=ilp32

Make the compiler tools (this will take up to an hour on a good connection)
sudo apt install autoconf automake autotools-dev curl python3 libmpc-dev libmpfr-dev libgmp-dev build-essential bison flex texinfo gperf libtool patchutils bc zlib1g-dev libexpat-dev
sudo make

Add a path to the /opt/riscv-gnu-toolchain folder by typing this command after each log-in or add to the ~/.bashrc file:
export PATH=/opt/riscv-gnu-toolchain/bin:$PATH

6. Compile the provided program (test.s) using the linker script (cse4372_riscv.ld) and verify the program assembles and the linker outputs the ELF file.

Verify that the GNU toolchain works for RV32i
Assemble a file test.s into a test ELF file using the default link script file:
riscv32-unknown-elf-gcc -o test test.s -ffreestanding -nostdlib -march=rv32i -mabi=ilp32

Show the disassembly of the test ELF file:
riscv32-unknown-elf-objdump -dr test > test.asm_dump

Extract out the binary from the test ELF file:
riscv32-unknown-elf-objcopy test -O binary test.bin

Dump the binary file:
hexdump test.bin

Note that the code is at an odd location in memory
(the startup address is implementation specific)

We will have RAM at low memory (0x00000000-0x0001FFFF) in our processor design with a reset PC value of 0x00000000

A custom link script file (cse4372_riscv.ld) will define the memory areas for the sections (.text, .data, .bss) and the stack top
Assemble test.s and create an ELF file:
riscv32-unknown-elf-gcc -o test test.s -march=rv32i -mabi=ilp32 -nostdlib -Tcse4372_riscv.ld

Create the Intel HEX file from the ELF file:
objcopy -O ihex test test.hex

7. Disassemble the code and verify that the .text section starts at address 0.

To disassemble the code and see the sections in the generated ELF file, run: riscv32-unknown-elf-objdump -dr test

8. Demonstrate steps 2-4 and 6-7 to the grader for credit.