ECE M116C - CS M151B Computer Assignment 1: RISC-V CPU Simulator

In this assignment, we are tasked with creating a C++ simulator for a simple RISC-V CPU. We use a 32-bit version of the RISC-V ISA and focus on implementing only 10 instructions:

{ADD, SUB, AND, OR, ADDI, ORI, ANDI, LW, SW, BEQ}

The code is written only with the standard libraries. First, the instructions are loaded into memory from the specified path, and then they are copied into the CPU class's instruction memory. Then, an object from the CPUStat class is initialized to keep track of CPU instructions. A main execution loop is run that first fetches then decodes each decimal-encoded instruction from the instruction memory, converting it into binary and checking its opcode and updating the Instruction struct accordingly.

The results of cpusim.cpp on trace/test/instMem1.txt are:

200					
47	74	41	38	0	0
41	42	16			

The results of cpusim.cpp on trace/test/instMem2.txt are:

200					
46	72	47	35	0	0
47	30	10			

The values are mapped accordingly to:

# of fetched instructions					
# of R-type	# of I-type	# of S-type	# of B-type	# of U-type	# of J-type
# of SW inst.	# of LW inst.	# of ADD inst.			

Included in the files is a Makefile which has targets *make*, *make debug*, *make test1*, *make test2*, and *make clean*, which compile, run, and clean up the associated targets. The simulator is all inside of cpusim.cpp, which requires the C++ 11 compiler to run.