CSCE2303 - Computer Organization and Assembly Language Programming

RISC-V Simulator

Names & IDs

Jana Elfeky, 900215017

Andrew Antoine, 900211633

Abdelrahman Ihab, 900213468

Program Description: The program consists of the functions: extractBits, signExtend, parseInstruction, execute, loadAssemblyProgram, loadData, printProgram and the main function. The execute function contains most of the programs logic as it takes the given instructions and divides it to dedicated functions in order to input the 32 bit binary instruction

The program utilizes the use of the maps data structure to store the binary representations of the RISC-V instructions with various RISC-V parameters (opcode, func3, etc..)

We implemented 2 bonus features in our program:

2. Implementing and integrating an assembler that will convert the input program into machine code and use these values to properly initialize the corresponding memory locations (which will be displayed as part of the relevant memory locations output).

7. Including a larger set of test programs (at least 6 meaningful programs) and their equivalent C programs.

For the first bonus feature, we used the function parseInstruction to convert the assembly instructions to machine code which are then displayed onto the screen when the user enters the instruction they want to execute. Moreover, for the second bonus feature, we created 6 meaningful C programs and included their equivalent assembly programs to use for testing.

Bugs/Issues: There weren't any major bugs/issues encountered while working on the simulator.

User Guide: After running, we need to input the starting address in the memory, afterwards the program receives the instructions until the word END is given. The program requests pair values (address, value written in the address)