Risc-V Simulator

**Team Members:**

Ahmed ElAswar - 900211265 - ahmed-farid@aucegypt.edu

Reem Hamada – 900196110 - Reem.abdelaty@aucegypt.edu

John Ibrahim - 900226449 - johnashraf@aucegypt.edu

**Brief Explanation:**

Initially we have the names of all supported 42 instructions, register names, and 5 halting instructions in the constants file.

registers.py has the logic for collaborating with the memories' registers.

All instructions are defined in functions.py which reads from registers.py to define

how registers will be utilized within each instruction.

in parse.py, the assembly code file is read and cleaned into data structures that then populates the memory

within the same file.

The overall connection and running of the program then happens in the main file.

**Decisions/Assumptions:**

Brief Description of Implementation:

1- pc is initialized to zero.

2- We write into the memory starting from the first address which is initialized to zero.

3- We are using 2 separate data structures

-byte arrays for the data memory.

-a list for the instructions memory.

4- The size of our data memory is 128 bytes.

**Bugs/Issues:**

No known bugs or issues.

**User Guide:**

Start by running main.py where a prompt will then be given asking for the name of the assembly input file you want to run. Once given, The final output is displayed in the terminal.

Example on testcase1

This is part of the output since the output is so long (displaying memory again for every single command.)

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

There are many more lines that we can’t display as screenshots

**List of programs:**

We have 4 short testcases and each has around 5 to 7 assembly instructions. Each has one or 2 halting to test the termination of the program upon reading one of the 5 halting functions. As well as 3 relatively longer testcases.

1- test1.s, test2.s, test3.s (the relatively longer files)

2- Test1\_Arithmetic\_Operations\_EBREAK

3- Test2\_LogicalOperations\_PAUSE

4- Test5\_loop

5- Test\_4\_Branch&JumpOperations\_FENCE