CSCI 6461 - User Guide for Project Part III

Team: Group 10 Date: April 12, 2025

1. Introduction

This user guide explains how to operate the CSCI 6461 Simulator with full instruction set support as required for Part III. The simulator allows users to execute machine-level programs written in the C6461 Instruction Set Architecture, including memory operations, branching, arithmetic, TRAP handling, and I/O operations.

2. System Requirements

- Java JDK 8 or higher
- Any OS (Windows, macOS, Linux)
- A Java IDE (Eclipse, IntelliJ) or terminal access
- Provided files: Simulator.jar, load.txt/test2.txt

3. Running the Simulator

Option 1 - IDE:

1. Open the project in your IDE.
2. Run GUI.java.
3. Use the GUI buttons to load and execute the program.

Option 2 - Terminal:

1. Open a terminal and navigate to the folder containing Simulator.jar.
2. Run: java -jar Simulator.jar
3. The GUI will appear allowing file loading and execution.

4. Using the GUI

- Load: Browse and select a machine code file (e.g., test2.txt or load.txt).
- Run: Executes all instructions continuously until halted.
- Step: Executes one instruction at a time.
- Halt: Stops the simulator.
- Display: Observe the current values of all registers, cache, and memory.

5. Program 2 - Word Search in Paragraph

This program demonstrates:

- Reading six sentences using device ID 2 (file input).
- Outputting the paragraph using device ID 1.
- Accepting a search word from the keyboard (device 0).
- Searching all six sentences using TRR (test equality).
- Displaying the sentence and position of the first match using OUT.

The program uses LDX, LDR, IN, OUT, JZ, TRR, and JMA to control flow and I/O.

6. Program File Format (load.txt / test2.txt)

Each line has:

[OCTAL ADDRESS] [OCTAL INSTRUCTION]

Example:

```
000150 104202   ; LDX 1, 200
000151 102102   ; IN R0, 2
...
```

Ensure no blank lines or invalid octal formatting.

7. Debugging and Tips

- Use the Step function to trace instruction-by-instruction behavior.
- Console output prints decoded instructions, register updates, and memory accesses.
- If nothing happens, check your input device IDs or file formatting.

8. Conclusion

This guide ensures you can run the simulator effectively and validate the execution of Program 2 under Part III specifications. For any additional diagnostics, enable debug print statements inside Control.java.