UVSim SRS Document

Purpose

UVSim is an easy-to-use program to help computer science students learn the basics of machine language and computer architecture.

Description

The desired outcome for the UVSim is that it will be a lightweight machine language interpreter for the language MLBasic capable of running programs written in that language.

Definitions

- Word: A 4-digit signed (+ or -) number that represents either an instruction, a value, or an empty space.
    - Operator: A two-digit instruction for the program to act on a location in memory. The operator is the first half of an instructional word, optionally preceded by a sign.
    - Operand: The two-digit location in memory of information to be acted upon by the operator. The operand is the second half of an instructional word.
- Accumulator: A memory register that temporarily holds a word for use in mathematical operations or for writing to the screen.

Requirements

Functional

Loading the Program:

1. The system will allow users to load a program into memory.

2. The system will validate that each instruction in the program is a valid BasicML instruction

3. The system will display an error message if an invalid instruction is used during loading.

Program Execution:

4. The system will fetch instructions from memory in the sequence it was given.

5. The system will decode and execute the fetched instruction.

6. The system will support the following BasicML instructions: READ, WRITE, LOAD, STORE, ADD, SUBTRACT, MULTIPLY, DIVIDE, BRANCH, BRANCHNEG, BRANCHZERO, HALT.

Input and Output:

7. The system will prompt the user to enter an integer when executing the READ instruction.

8. The system will output the value stored in the specified memory slot when executing a WRITE instruction.

Arithmetic Operations:

9. The system will add the value from a specified memory location to the accumulator for the ADD instruction.

10. The system will subtract the value from a specified memory location to the accumulator for the SUBTRACT instruction.

11. The system will multiply the accumulator by the value from a specified memory location for the MULTIPLY instruction.

12. The system will divide the accumulator by the value from a specified memory location for the DIVIDE instruction.

Branching Operations:

13. The system will set the instruction counter to the specified address for BRANCH instructions.

14. The system will set the instruction counter to the specified address if the accumulator is negative for BRANCHNEG instructions.

15. The system shall set the instruction counter to the specified address if the accumulator is zero for BRANCHZERO instructions.

Non-Functional Requirements

Performance Requirements:

1. The system will execute each instruction within 10 milliseconds to ensure responsive interaction with the user.

Usability/Help Requirements:

2. The system shall provide clear error messages to help users debug their programs.

Reliability Requirement:

3. The system shall handle unexpected input, ensuring the virtual machine doesn't crash due to invalid operations.