1. Introduction

1.1 Purpose

This document provides the Software Requirement Specification (SRS) for the UVSim program, a simplified machine language interpreter. It includes functional and nonfunctional requirements to assist in the development of the program and its graphical user interface (GUI).

1.2 Scope

UVSim program is designed to execute a simple set of machine language instructions in the form of 5 character signed digits. The program includes memory, a program counter, an accumulator acting as a single register, and a console for user input and output. A GUI will provide a user-friendly interface to interact with these items.

1.3 Acronyms

* SRS: Software Requirement Specification
* GUI: Graphical User Interface

1. Functional Requirements
   1. Memory
2. The system shall initialize with 100 memory locations, each containing a word of 5 characters.
3. The system shall allow reading a word from the console and storing it in a specified memory location.
4. The system shall allow writing a word from a specified memory location to the console.
5. The system shall allow loading a word from a specified memory location into the accumulator
6. The system shall allow storing a word from the accumulator into a specified memory location.
   1. Arithmetic Operations
7. The system shall allow adding a word from a specified memory location to the word in the accumulator.
8. The system shall allow subtracting a word from a specified memory location from the word in the accumulator.
9. The system shall allow multiplying a word from a specified memory location by the word in the accumulator.
10. The system shall allow dividing the word in the accumulator by a word in a specified memory location.
11. The system shall handle overflows by keeping the same sign and truncating the digits to ensure a 5 digit word.
    1. Control Flow
12. The system shall allow branching to a specified memory location if the accumulator value is positive.
13. The system shall allow branching to a specified memory location if the accumulator value is negative.
14. The system shall allow branching to a specified memory location if the accumulator value is zero.
15. The system shall terminate execution when a halt instruction is encountered.
    1. Program Execution
16. The program shall prompt for user input for a file, then read and store the contents into system memory.
17. Nonfunctional Requirements
    1. Usability
18. The GUI shall be user friendly and intuitive, allowing the user to easily load a file and access the write method, as well as monitor the memory, pointer, and accumulator values.
    1. Reliability
19. The system shall gracefully handle any invalid inputs, providing informative error messages to the user without crashing.
20. Maintainability
21. The system shall be constructed in a way that allows easy modification to the structure to allow implementation of new features.