

Team SRS Document

Functional Requirements

1. UVSIM shall allow users to select a text file containing instructions for the UVSIM.
2. UVSIM shall execute read operations, prompting the user to input values for specific memory locations in the GUI read program text input box. .
3. UVSIM shall execute write operations, displaying the value stored at a specified memory location.
4. UVSIM shall execute load operations, loading a four-digit integer from a memory array index into the accumulator.
5. UVSIM shall execute store operations, storing the value of the accumulator variable into a specific index in the memory array.
6. UVSIM shall support basic math operations such as addition, subtraction, multiplication, and division of an integer value from a specific memory index from the value of the accumulator variable.
7. UVSIM shall correctly evaluate the conditions for branching (negative or zero accumulator) and jump to the specified memory location accordingly.
8. UVSIM shall handle file loading errors, such as invalid file formats or inaccessible files, and in case of invalid input, unsupported operations, or runtime errors it'll notify users by providing informative error messages to users.
9. UVSIM shall provide a GUI for interacting with the simulator, allowing users to input machine language, view program output, and control the program execution.
10. UVSIM shall read a four-digit integer from the GUI read text input box into a specific index in the memory array and write a four-digit integer from a memory array index to the GUI output text box.
11. UVSIM shall support the execution of programs with varying lengths and complexities, accommodating different input file sizes and instruction sets.
12. UVSIM shall allow users to export the current state of the UVSIM machine, including memory contents and program counter, for analysis or sharing
13. UVSIM shall provide functionality for users to create and save their own UVSIM programs within the application for future use.
14. UVSIM shall allow users to pause, resume, and reset program execution at any point.

15. UVSim shall provide options for users to step through the program, executing one instruction at a time.
16. UVSim shall maintain the integrity of program execution by preventing unauthorized modifications to critical components, such as the program counter and accumulator.
17. UVSim shall log program execution details, including executed instructions and memory changes, for debugging and analysis purposes.
18. UVSim shall increment the pc variable for each loop iteration in the `execute_program` function until the loop stops.

Non functional requirements:

1. UVSim's GUI shall be implemented in the Python coding language with clear instructions on how to interact with it.
2. UVSim's code shall be written and designed in a way that'll minimize the execution time and provide ease for maintenance.
3. UVSim shall provide clear and informative error messages to assist users in troubleshooting.
4. UVSim shall be scalable, capable of handling increasing user loads and expanding functionality without significant performance degradation.
5. UVSim shall be software-independent and shall be runnable on Windows 10 or above without requiring additional dependencies.

