User Stories

User Story 1: As a Computer Science Student
Title: Executing Basic Machine Language Programs for Learning Purposes

Narrative:
As a computer science student,
I want to be able to load and execute BasicML programs using the UVSim simulator,
So that I can learn how machine language instructions are executed in a computer architecture.

Acceptance Criteria:

- The student can load a BasicML program from a text file into the UVSim simulator.
- The simulator executes the loaded program, correctly processing each BasicML instruction.
- The student can view the results of each instruction, including any changes to the memory and accumulator.
- The simulator provides clear error messages for invalid instructions or runtime errors.

User Story 2: As a Computer Science Instructor
Title: Providing an Interactive Learning Tool for Machine Language Concepts

Narrative:
As a computer science instructor,
I want to provide my students with an interactive tool like the UVSim simulator,
So that they can experiment with and understand the concepts of machine language and basic computer operations.

Acceptance Criteria:

- The instructor can use the simulator to demonstrate the execution of BasicML programs.
- The simulator includes features to step through each instruction, allowing for instructional pauses and explanations.
- The simulator accurately represents key concepts of machine language execution, such as memory management and arithmetic/logical operations.
- The tool is user-friendly and intuitive for students who are new to machine language concepts.

Use Cases

1. Use Case: Load Program from File

Actor: User (Student/Instructor)
Description: Load a BasicML program from a text file into UVSim's memory.
Steps:
User selects the option to load a program.
User provides the file path of the BasicML program.
The program is loaded into UVSim's memory.

2. Use Case: Execute Program

Actor: User
Description: Execute the loaded BasicML program.
Steps:
User selects the option to execute the program.
UVSim processes each instruction sequentially.

3. Use Case: View Memory Contents

Actor: User
Description: View the contents of the UVSim memory.
Steps:
User selects the option to view memory.
UVSim displays the contents of its memory.

4. Use Case: Input Data

Actor: User
Description: Input data when a READ instruction is executed.
Steps:
READ instruction is encountered.
User is prompted to enter data.
Data is stored in the specified memory location.

5. Use Case: Output Data

Actor: User
Description: Output data to the screen when a WRITE instruction is executed.

Steps:

WRITE instruction is encountered.

Data from the specified memory location is displayed on the screen.

6. Use Case: Perform Arithmetic Operations

Actor: System

Description: Execute arithmetic operations like ADD, SUBTRACT, MULTIPLY, DIVIDE.

Steps:

An arithmetic instruction is encountered.

UVSim performs the operation on the accumulator and memory.

7. Use Case: Modify Accumulator

Actor: System

Description: Load or store data in the accumulator.

Steps:

LOAD or STORE instruction is encountered.

UVSim loads data into the accumulator or stores data from the accumulator into memory.

8. Use Case: Branch Execution

Actor: System

Description: Alter the execution flow based on BRANCH instructions.

Steps:

BRANCH instruction is encountered.

UVSim jumps to the specified memory location for the next instruction.

9. Use Case: Handle Errors

Actor: System

Description: Handle errors like invalid instructions or memory overflow.

Steps:

An error condition is detected.

UVSim displays an appropriate error message and halts or handles the error gracefully.

10. Use Case: Halt Execution

Actor: System

Description: Halt the execution of the program.
Steps:
HALT instruction is encountered.
UVSim stops executing further instructions.

11. Use Case: Step Through Execution

Actor: User
Description: Execute the program one instruction at a time.
Steps:
User selects the option to step through the program.
UVSim executes one instruction and waits for the user to proceed.

12. Use Case: Reset Simulator

Actor: User
Description: Reset the UVSim to its initial state.
Steps:
User selects the option to reset the simulator.
UVSim clears its memory and accumulator, ready for a new program.

13. Use Case: Save Program State

Actor: User
Description: Save the current state of the program and memory.
Steps:
User selects the option to save the state.
UVSim saves the current memory and program state.

14. Use Case: Load Saved State

Actor: User
Description: Load a previously saved state of the program and memory.
Steps:
User selects the option to load a saved state.
UVSim loads the saved memory and program state.

15. Use Case: View Execution History

Actor: User

Description: View the history of executed instructions and their results.
Steps:
User selects the option to view execution history.
UVSim displays a log of executed instructions and changes to memory and accumulator.