Colton Evans
Group C

Personal SRS Document

Functional Requirements

1. The UVSim shall support basic math operations such as addition, subtraction, multiplication, and division.
2. The UVSim shall be equipped with a 100-word memory.
3. The UVSim shall perform these operations accurately and update the contents of the accumulator and memory as needed.
4. The UVSim shall correctly evaluate the conditions for branching (negative or zero accumulator) and jump to the specified memory location accordingly.
5. The UVSim shall detect and report errors like: division by zero, invalid memory addresses, or unsupported instructions.
6. The UVSim shall handle errors by providing informative error messages to users in case of invalid input, unsupported operations, or runtime errors.
7. The UVSim shall provide a GUI for interacting with the simulator, allowing users to input machine language, view program output, and control the program execution.
8. The UVSim shall display relevant information such as: program state, memory contents, accumulator value, and execution status to the user during program execution.
9. The UVSim shall be capable of executing machine language programs written in BasicML.
10. The UVSim shall load the BasicML program to the main memory starting at location 00 before it executes.
11. The UVSim shall support the following BasicML instructions: READ, WRITE, LOAD, STORE, ADD, SUBTRACT, DIVIDE, MULTIPLY, BRANCH, BRANCHNEG, BRANCHZERO, and HALT.
12. The UVSim shall correctly read and execute each BasicML instruction from the program according to the instructions given.
13. The UVSim shall be able to read input from the keyboard (displayed from GUI) using the READ instruction then store it in a memory location.
14. The UVSim shall be able to display output to the screen using the WRITE instruction, then display the contents of the memory location.
15. The UVSim shall terminate as soon as the HALT instruction is read into The UVSim.

Non functional requirements:

1. The UVSim's GUI shall be simple with clear instructions on how to interact with it.
2. The UVSim's code shall be written and designed in a way that'll minimize the execution time.
3. The UVSim shall be structured and documented in a way that'll provide minimal difficulty for maintenance.