CS 2450 002

David Mora-Bravo, Ethan Kidd, Garrison Brown, Travis Cooper

November 15, 2024

Design Document

User Stories

As a user,

I want to be able to load my BasicML program into the UVSim and have it perform I/O operations such as reading, writing, loading, and storing.

So that I can control the programs input and output functions.

As a user,

I want to be able to load my BasicML program into the UVSim and have it perform arithmetic operations like adding, subtracting, dividing, and multiplying.

so that I can perform calculations within the program.

As a user,

I want to be able to load my BasicML program into the UVSim and have it perform basic control operations like branching to specific memories based on specific criteria and halting the program

so that I can have better control of how the program behaves.

As a user,

I want to be able to control the color scheme of the program window by clicking a button that will open a menu with options of different schemes to choose from.

So that I can control how the program looks.

Use Cases

1. Read operation

Actor: User

System: UVSim

Goal: Read a word to a specific memory register.

- The user runs the program

- The program prompts the user to input a .txt file
- The user inputs a .txt file that contains a word beginning with ‘10’
- The .txt file is loaded into the registers of the program
- The program iterates through the registers until it encounters that word
- The program prompts the user to enter a word to be read into memory
- The user inputs a word and hits ‘Enter’
- The program reads that word into the specified location in memory

2. Write operation

Actor: User

System: UVSim

Goal: Write a word from a specific memory register to the screen.

- The user runs the program
- The program prompts the user to input a .txt file
- The user inputs a .txt file that contains a word beginning with ‘11’
- The .txt file is loaded into the registers of the program
- The program iterates through the registers until it encounters that word
- The program writes a word from the specified register to the screen

3. Load operation

Actor: User

System: UVSim

Goal: Load a word from a specific memory register to the accumulator.

- The user runs the program
- The program prompts the user to input a .txt file
- The user inputs a .txt file that contains a word beginning with ‘20’
- The .txt file is loaded into the registers of the program
- The program iterates through the registers until it encounters that word
- The program loads a word from the specified location in memory into the accumulator

4. Store operation

Actor: User

System: UVSim

Goal: Store a word from the accumulator into the specified memory location.

- The user runs the program
- The program prompts the user to input a .txt file
- The user inputs a .txt file that contains a word beginning with ‘21’

- The .txt file is loaded into the registers of the program
- The program iterates through the registers until it encounters that word
- The program stores a word from the accumulator into the specified memory location

5. Add operation

Actor: User

System: UVSim

Goal: Add a word from a specific location in memory to the word in the accumulator.

- The user runs the program
- The program prompts the user to input a .txt file
- The user inputs a .txt file that contains a word beginning with ‘30’
- The .txt file is loaded into the registers of the program
- The program iterates through the registers until it encounters that word
- The program adds the word at the specified memory location to the word in the
- accumulator
- The result is left in the accumulator

6. Subtract operation

Actor: User

System: UVSim

Goal: Subtract a word from a specific location in memory from the word in the accumulator.

- The user runs the program
- The program prompts the user to input a .txt file
- The user inputs a .txt file that contains a word beginning with ‘31’
- The .txt file is loaded into the registers of the program
- The program iterates through the registers until it encounters that word
- The program subtracts the word at the specified memory location from the word in the
- accumulator
- The result is left in the accumulator

7. Divide operation

Actor: User

System: UVSim

Goal: Divide a word in the accumulator by a word at the specified location in memory.

- The user runs the program
- The program prompts the user to input a .txt file
- The user inputs a .txt file that contains a word beginning with ‘32’

- The .txt file is loaded into the registers of the program
- The program iterates through the registers until it encounters that word
- The program divides the word in the accumulator by the word at the specified memory
- location
- The result is left in the accumulator

8. Multiply operation

Actor: User

System: UVSim

Goal: Multiply a word at a specific location in memory by the word in the accumulator.

- The user runs the program
- The program prompts the user to input a .txt file
- The user inputs a .txt file that contains a word beginning with ‘33’
- The .txt file is loaded into the registers of the program
- The program iterates through the registers until it encounters that word
- The program multiplies the word at the specified memory location by the word in the
- accumulator
- The result is left in the accumulator

9. Branch operation

Actor: User

System: UVSim

Goal: Branch to a specified location in memory.

- The user runs the program
- The program prompts the user to input a .txt file
- The user inputs a .txt file that contains a word beginning with ‘40’
- The .txt file is loaded into the registers of the program
- The program iterates through the registers until it encounters that word
- The program branches to the specified location in memory

10. Branch Neg. operation

Actor: User

System: UVSim

Goal: Branch to a specified location in memory if the accumulator is negative.

- The user runs the program
- The program prompts the user to input a .txt file
- The user inputs a .txt file that contains a word beginning with ‘41’

- The .txt file is loaded into the registers of the program
- The program iterates through the registers until it encounters that word
- The program branches to the specified location in memory if the accumulator is negative

11. Branch Zero operation

Actor: User

System: UVSim

Goal: Branch to a specified location in memory if the accumulator is zero.

- The user runs the program
- The program prompts the user to input a .txt file
- The user inputs a .txt file that contains a word beginning with ‘42’
- The .txt file is loaded into the registers of the program
- The program iterates through the registers until it encounters that word
- The program branches to the specified location in memory if the accumulator is zero

12. Halt operation

Actor: User

System: UVSim

Goal: Stop the program

- The user runs the program
- The program prompts the user to input a .txt file
- The user inputs a .txt file that contains a word beginning with ‘43’
- The .txt file is loaded into the registers of the program
- The program iterates through the registers until it encounters that word
- The program exits

13. Change Color Scheme

Actor: User

System: UVSim

Goal: Change color of program window

- User runs the program
- User selects Change Theme button
- Menu pops up with .json files that contain different color schemes
- User selects file to be used
- Program restarts displaying selected color scheme

14. Edit .txt file

Actor: User

System: UVSim

Goal: Edit instructions in .txt file in the application

- User runs the program
- User selects Load Instructions button
- User selects instructions from file to be loaded into application
- User the views or change instructions as needed