Software Requirement Specifications

For

8051 Emulator

Version 1.0

Last Updated: June 30, 2013

Under the guidance of

Prof. Deepak B. Phatak

Principal Investigator

Project In-charge:

Dr. Madhuri Sawant

Mentors:

Pankaj Patil

M. Barani

Group Name: 8051 Emulator

T.M.S.K. Alekhya

Prayasee Pradhan

Priyanka Padmanabhan

Sumanth Myrala

Anupam Sanghi

Sweksha Tripathi

Department of Computer Science and Engineering IIT Bombay, Powai

CONTENTS

1.	**INTRODUCTION**	**1**
1.1	DOCUMENT PURPOSE	1
1.2	PRODUCT SCOPE	1
1.3	INTENDED AUDIENCE AND DOCUMENT OVERVIEW	1
1.4	DOCUMENT CONVENTIONS	2
1.5	REFERENCES AND ACKNOWLEDGMENTS	2
2.	**OVERALL DESCRIPTION**	**3**
2.1	PRODUCT PERSPECTIVE	3
2.2	PRODUCT FUNCTIONALITY	3
2.3	USERS AND CHARACTERISTICS	3
2.4	OPERATING ENVIRONMENT	4
2.5	DESIGN AND IMPLEMENTATION CONSTRAINTS	4
3.	**SPECIFIC REQUIREMENTS**	**5**
3.1	EXTERNAL INTERFACE REQUIREMENTS	5
3.2	FUNCTIONAL REQUIREMENTS	5
3.3	BEHAVIOUR REQUIREMENTS	6
3.3.1	APPLICATION LAUNCH	6
3.3.2	FILE MENU	7
3.3.3	WORKBENCH	9
3.3.4	CIRCUIT SCHEMATIC DESIGN	10
3.3.5	INTERNALS 8051	11
3.3.6	EXECUTE	12
3.3.7	OSCILLOSCOPE	13
3.3.8	EDITOR	14
3.3.9	CLEAR	15
3.3.10	ASSEMBLE	16
4.	**OTHER NON-FUNCTIONAL REQUIREMENTS**	**17**
4.1	PERFORMANCE REQUIREMENTS	17
4.2	SOFTWARE QUALITY ATTRIBUTES	17
5.	**FUTURE SCOPE**	**18**

List of Figure

Fig 3.3.1	Use Case for Application Launch	6
Fig 3.3.2	Use Case for File Menu	7
Fig 3.3.3	Use Case for WorkBench	9
Fig 3.3.4	Use Case for Circuit Schematic Design	10
Fig 3.3.5	Use Case for Internals 8051	11
Fig 3.3.6	Use Case for Execute	12
Fig 3.3.7	Use Case for Oscilloscope	13
Fig 3.3.8	Use Case for Assembly Editor	14
Fig 3.3.9	Use Case for Clear	15
Fig 3.3.10	Use Case for Assemble	16

1. Introduction

1.1 Document Purpose

The purpose of this document is to present the detailed description of software requirements to build 8051 Microcontroller Emulator. It gives the developers, the overview of developing the application. It explains the purpose and features of the system, the interfaces of the system, what the system does, the constraints under which it must operate and how the system reacts to external stimuli.

1.2 Product Scope

It is useful for conducting academic experiments on Microcontrollers and gives a hands on learning experience to the students. This product is also helpful for embedded hobbyists without using hardware.

1.3 Intended Audience and Document Overview

This document is intended for the developers as well as the students who are the users of this software.

The second chapter, named 'Overall Description' gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter named 'Specific Requirements' is written primarily for the developers and describes in technical terms the details of the functionality of the product.

The above two chapters describe the same software product in its entirety, but are intended for different audiences and thus use different language.

The fourth chapter, named 'Other Non-Functional Requirements' specifies the criteria that can be used to judge the operation of the system, rather than specific behaviours. This give information about the performance, safety requirements and software quality attributes.

The fifth chapter, named 'Future Scope' describes about the improvements that can be incorporated in the project.

1.4 Document Conventions

This Document follows the IEEE formatting of requirements. It has used Calibri font size 12 throughout the document for text.

1.5 Reference and Acknowledgments

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications*.

IEEE Computer Society, 1998

Technical Reference:

(i) The 8051 microcontroller and embedded systems by Mazidi, Edition II, Pearson Education

(ii) The 8051 Microcontroller by Kenneth-J-Ayala, Edition III, Thomson, Delmar Learning

2. Overall Description

2.1 Product Perspective

8051 Emulator is an integrated development environment with an objective to replicate the 8051 Microcontroller development board on Aakash tablet.

Major components:

1. **Workbench:** It facilitates the user to build circuits. It contains the following
    i. Circuit Panel: It contains various built in circuits of interfacing devices that the user connects to the different ports of microcontroller. On selecting interface device. Application places selected interface device on the specified port of Microcontroller.
    ii. Execute Button: It is a button that is used to execute the assembled code and view the corresponding animations on the final circuit.
    iii. Oscilloscope Button: It is a button used to view the V-t graph generated when the user runs a program on the circuit that includes digital to analog converter.
    iv. Grid: It is the workspace that is available for the user for interfacing the complete circuit using circuits from the circuit panel. It is scrollable.
2. **Editor:** It provides workspace to the user to code in assembly language for the 8051 microcontroller.
3. **Internals 8051**: It displays the contents of the various registers of the 8051 Microcontroller after execution of the code.

2.2 Product Functionality

1. Circuit designing: The users designs the Microcontroller circuit based on requirements by selecting different built-in circuits like the LED Circuit, the Stepper Motor Circuit, 7 Segment Display, DAC Circuit, etc.
2. Execute Assemble Code: The user writes the assembly code in editor for programming the 8051.
3. View results similar to that which appears on real development board of 8051.
4. View the register contents of the 8051 during the execution of the code.

2.3 Users and Characteristics

The target users of this application are students of Embedded System courses and anyone who is interested in learning the basics of Microcontrollers. The users should have adequate knowledge about circuit designing, assembly programming, and microcontrollers particularly 8051.

2.4 Operating Environment

The application is designed for the Aakash tablet with Android version 4.0.3 (OS Ice-cream sandwich).

2.5 Design and Implementation Constraints

Tablet Screen: Aakash tablet screen size is 960 X 510. Therefore building complex circuit on is not viable.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interface

8051Emulator user interface should have following:

1. Interface to circuit design bench.
2. Interface to write assembly code.
3. Interface to display errors occurred in the assembly code.
4. Interface to display ROM contents and ROM space usage
5. Interface to display the Internals of the 8051, like accumulator, working register, program counter, special function register, etc.
6. Interface for 8051 microcontroller learning material for user.
7. Interface to perform new experiments, save experiments and open already saved experiments.
8. Buttons used in application should be easily accessible and touchable for the user.
9. Application should be easily navigable across the different modules.

3.2 Functional Requirements

1. Building Circuit:
   i. User opens workspace for designing circuits.
   ii. User selects various circuits from the Circuit Panel corresponding to the ports provided in a Dialogue Box provided in the interface.
   iii. Application places the interface circuit in the workbench area on selected port of the micro controller.
   iv. Workbench facilitates user to add multiple interface circuits to the microcontroller.
   v. User saves the circuit diagram.
2. Writing Assembly Code:
   i. User writes the assembly code, based on different interface circuits mounted on ports of the Microcontroller
   ii. User assembles, the code written.
   iii. If user has made errors in writing assembly code, application displays the error message.
   iv. Application displays micro-controller ROM contents, ROM Usage.
   v. User saves the assembly code written.

vi. User edits assembly code in the editor, if user finds inappropriate output after executing of the previously written code.

3. Execution:
   i. User executes the assembly code.
   ii. Application displays appropriate behavioural animations on the interface devices.
   iii. Application displays the internal contents of microcontroller after the execution of code.
4. File Handling:
   i. User creates new project.
   ii. User opens the previously saved files.
   iii. User saves the project, which contains the .asm file for the assembly code and the .sch file for the circuit.
   iv. User deletes the saved file.

3.3 Behaviour Requirements

3.3.1 Application Launch:

Diagram:

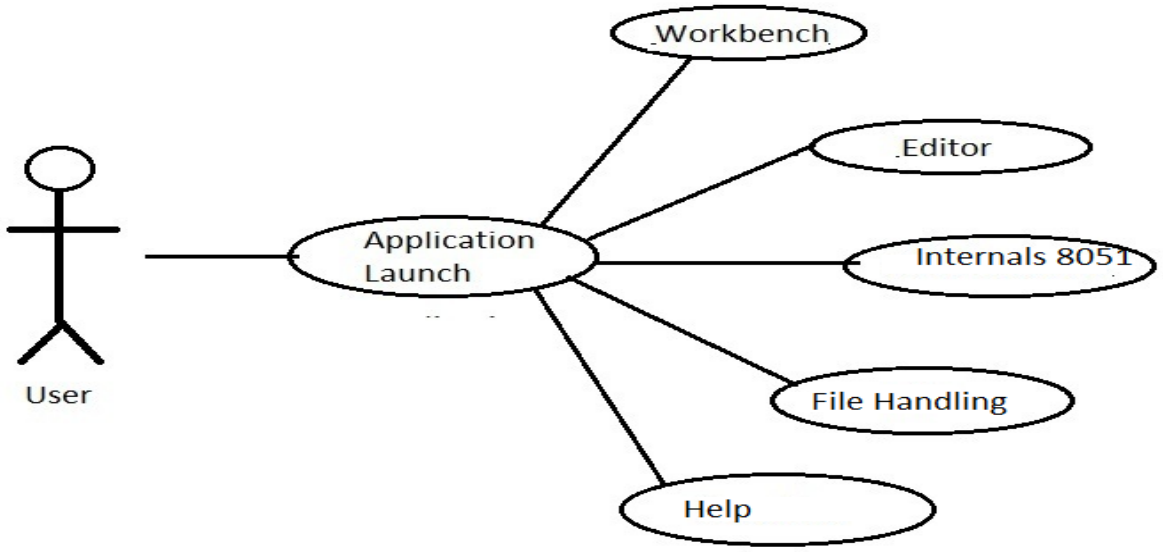

Fig 3.3.1 Use Case for Microcontroller Emulator

Brief Description:

The user accesses the Application Launcher, wherein user further accessess Workbench, Editor, Internals 8051, File handling and Help.

PreCondition:

The application should be installed with the proper version and the proper format.

Initial Step-By-Step Description:

This use case provides the user with the following functionalities:

1. The User accesses the Workbench which consists of a Microcontroller along with Reset, Power, Clock and Pull-up Circuit. In this module the user designs different circuits by selecting the desired one from the Circuit Panel.
2. The User accesses the Editor wherein user writes assembly code to program 8051 microcontroller.
3. The User accesses the Internals 8051 interface which displays the contents of the register.
4. The file option provides facility for the user to save circuit, open an existing circuit, create a new circuit and delete a circuit.
5. The User also views the contents of the help, which includes the User Manual, the 8051 Learning Manual and the list of experiments.

PostCondition:

The application is accessed by the user.

3.3.2 File Menu:

Diagram:

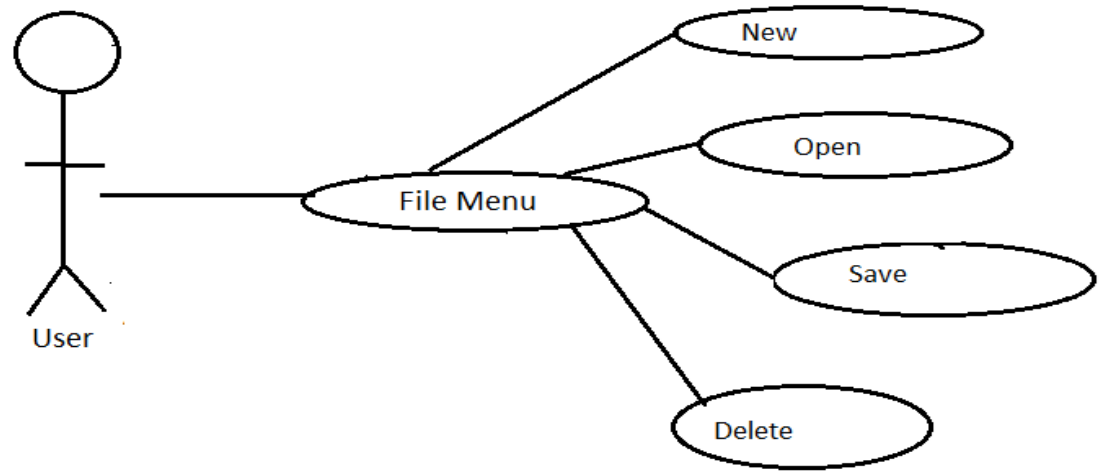

Fig 3.3.2 Use Case for File Menu

Brief Description:

The User accesses the File menu to open a new project, open previously created project, save the project and delete the project.

NEW:

PreCondition:

The user should have made the circuit on the workbench or should have written the assembly code in the editor.

Initial Step-By-Step Description:

1. The user access the new option.
2. The previously opened project is saved.
3. A new workspace and a blank Assembly Editor is available for the user.

PostCondition:

The new project is loaded.

OPEN:

PreCondition:

The user should have saved some projects in the application.

Initial Step-By-Step Description:

1. The user accesses the open option.
2. The previously opened project is saved.
3. The user accesses the list of all the saved projects.
4. The user opens one of the projects from the list.

PostCondition:

The previously saved project is opened.

SAVE:

PreCondition:

The user should have performed certain experiment.

Initial Step-By-Step Description:

1. The user accesses the save option.
2. The user provides the experiment name.
3. The project is saved with the given experiment name.

PostCondition:

The project is saved.

DELETE:

PreCondition:

The user should have selected a project existing in the application.

Initial Step-By-Step Description:

1. The user accesses the list of projects.
2. The user long presses the project to be deleted.

PostCondition:

The selected project is deleted.

3.3.3 WorkBench:

Diagram:

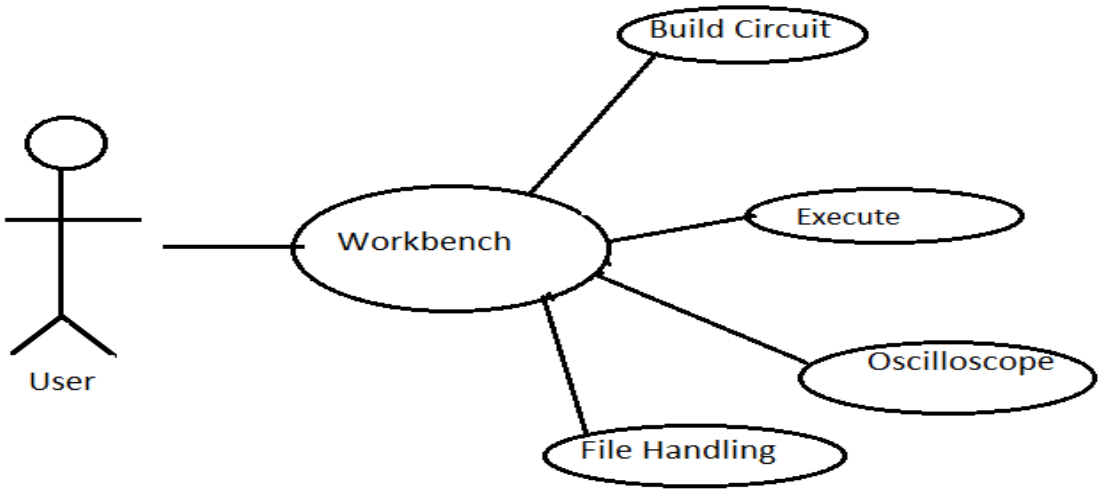

Fig 3.3.3 Use Case for WorkBench

Brief Description:

The user creates the schematic of the circuit, executes the assembly program, accesses file menu for file operations, uses oscilloscope for viewing graph if Digital to Analog Convertor is used in the circuit.

PreCondition:

The application should have been launched and the user has accessed WorkBench interface.

Initial Step-By-Step Description:

1. The User accesses the Workbench to design circuit schematic.
2. The user accesses to the Create Circuit option, which facilitates the user to design the desired circuit schematic by choosing the available circuit options from the Circuit Panel.
3. The user accesses the execute option after writing the assembly code.
4. The user views voltage vs time graph on the Oscilloscope by accessing it.
5. The User accesses to the File Menu which provides new, open, save and delete circuit options.

Post Condition:

The user has created the circuit schematic in workbench.

3.3.4 Circuit Schematic Design:

Diagram:

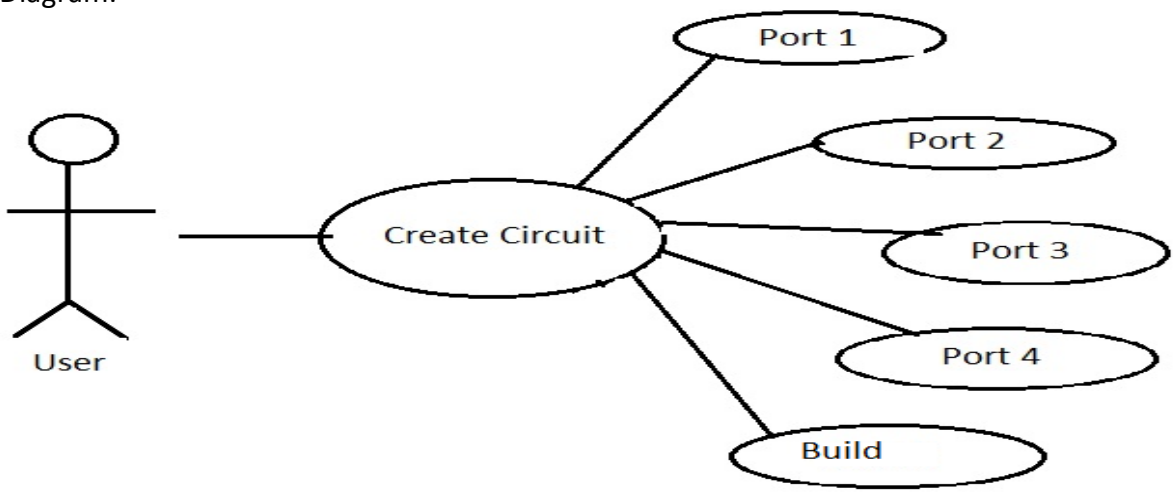

Fig 3.3.4 Use Case for Circuit Schematic Design

Brief Description:

The user accesses the Create Circuit option to mount different circuits on different ports.

PreCondition:

The user should have a thorough understanding of the requirements of a particular project.

Initial Step-By-Step Description:

1. The user accesses this option to create the desired circuit
2. The User selects available circuits for ports of the Microcontroller.
3. Application places selected circuit on specified port of the MicroController.

Post Condition:

The user can build required circuit in WorkBench Interface.

3.3.5 Execute:

Diagram:

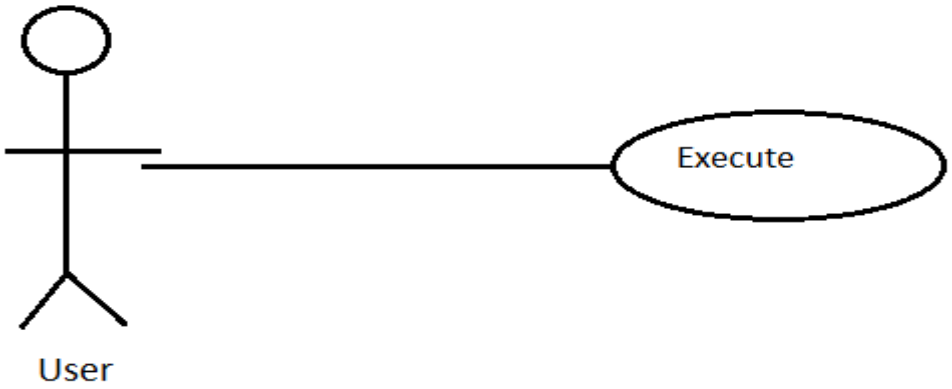

Fig 3.3.5 Use Case for Execute

Brief Description:

The User accesses the Execute option to view the desired result on the circuit created in the workbench during runtime.

PreCondition:

The user should have written assembly code and it should be assembled.

Initial Step-By-Step Description:

1. The user writes the assembly code in the editor, assembles it and executes it from the workbench.
2. During execution application shows behavioural animations on the interface devices based on assembly code written and also updates the 8051 Internals interface.

Post Condition:

The user has built required circuit in WorkBench interface.

3.3.6 Internals 8051:

Diagram:

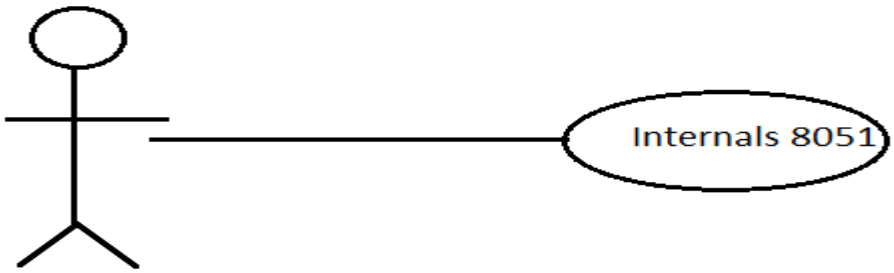

Fig 3.3.6 Use Case for Internal Contents of 8051

Brief Description:
The User accesses the Internals 8051 interface to view the contents of the registers.

PreCondition:
The assembly code written on the Assembly Editor should have been assembled and the program should have been executed.

Initial Step-By-Step Description:
The user accesses Internals 8051 interface to view the contents of the internal registers like R1, R2, R3, etc, the value of the accumulator, the value of the Stack Pointer, Program Counter, Data Pointer, the values of the ports, interrupt enable, Interrupt priority, etc.

Post Condition:
The code is executed and the output is displayed in the WorkBench and the Internals 8051 interface.

3.3.7 Oscilloscope:

Diagram:

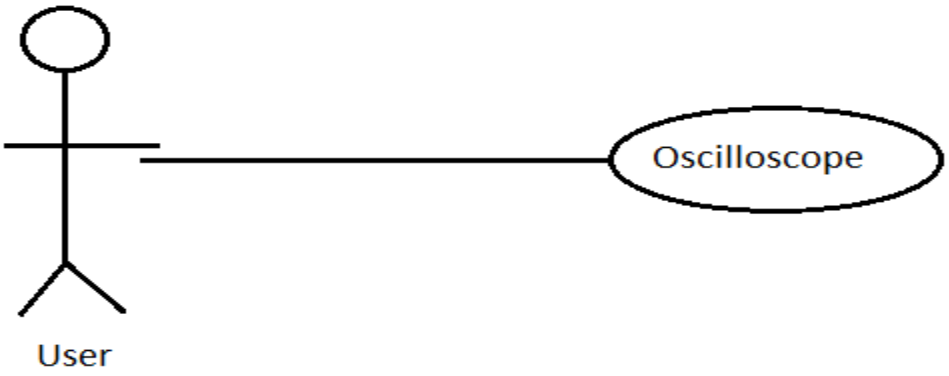

Fig 3.3.7 Use Case for Oscilloscope

Brief Description:

The User accesses the Oscilloscope option to view the graph generated based on assembly code if DAC has been interfaced with Microcontroller.

PreCondition:

The user should have mounted DAC circuit and should have executed assembly code.

Initial Step-By-Step Description:

1. User interfaced DAC with the Microcontroller.
2. User writes the assembly code in Editor.
3. User executed the code written.
4. User accesses oscilloscope to view voltage-time graph generated based on assembly code written.

Post Condition:

The graph is generated on a popup Dialogue Box.

3.3.8 Assembly Editor:

Diagram

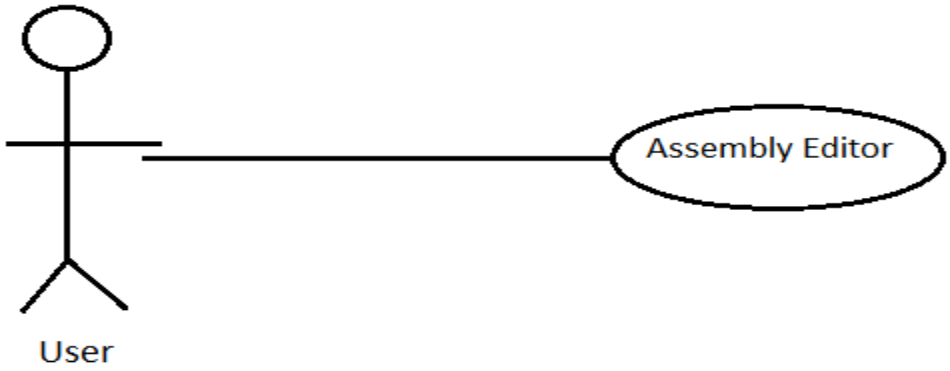

Fig 3.3.8 Use Case for Assembly Editor

Brief Description:

The User accesses the Assembly Editor to write the assembly code.

PreCondition:

The user should have selected the Assembly Editor tab.

Initial Step-By-Step Description:

1. The user writes the assembly instructions in the Assembly Editor.
2. The user assembles the code.
3. The user views the status of the assembler in the log cat.
4. The user views the internal ROM contents and total memory usage.

Post Condition:

The code written by the user is assembled.

3.3.9 Clear:

Diagram:

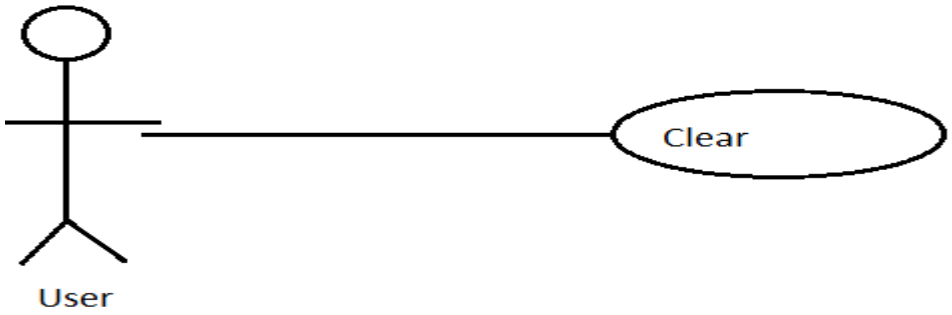

Fig 3.3.9 Use Case for Clear

Brief Description:

The User accesses the Clear option to clear the contents of the assembly editor.

PreCondition:

The user should have written the code in the assembly editor.

Initial Step-By-Step Description:

1. The user writes the code in the assembly editor.
2. If user needs to write code in different way, user clears the code.

Post Condition:

The ROM Usage, ROM Contents, Logcat and Assembly Code are cleared.

3.3.10 Assemble:

Diagram

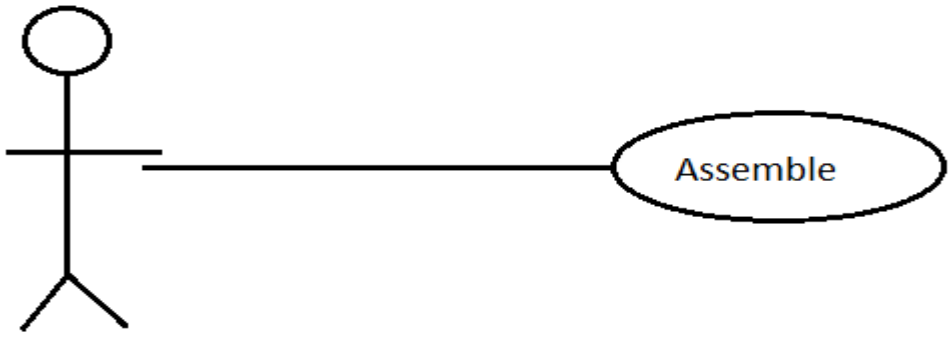

Fig 3.3.10 Use Case for Assemble

Brief Description:

The User accesses the Assemble option to assemble the assembly code written in the Assembly Editor.

PreCondition:

The user should have written the Assembly code on the Editor.

Initial Step-By-Step Description:

1. The user writes code in assembly editor and accesses assemble option
2. If error is present, the error message appears in the log cat, specifying the line numbers of the errors, else it is assembled successfully.
3. If assembled successfully, the ROM contents and the total ROM usage fields are updated.

Post Condition:

The ROM Usage, ROM Contents, LogCat and Assembly Code are updated.

4. Other Non-Functional Requirements

4.1 Performance Requirements

1. The application should have enough circuit design bench area, which should accommodate interface circuits.
2. Application should perfectly emulate the functionalities of the 8051 microcontroller.
3. Application should be easy to operate.
4. The processing time of the emulator should exactly duplicate that of the real 8051 microcontroller.

4.2 Software Quality Attributes

Availability: The application is open source and will be easily available on GitHub and SourceForge.

Correctness: Application should be designed to give accurate results for a variety of circuits and applications. The results produced should be equivalent to 8051 as much as possible.

Usability: The circuit designing should be easy to use. A thorough user manual should be provided to answer any queries. A list of experiments should be provided which the user can refer to for better understanding of the 8051 microcontroller.

5. Future Scope

Following features can be incorporated to enhance the usability of the product

- ✓ Making it executable for closed loop systems.
- ✓ Features of timers, serial communication and external interrupts can be included.
- ✓ A wide variety of circuits can be made available to the user.
- ✓ Embedded C coding feature can be included in the application.