



Amna Farooq
01-134141-011
Group ID-06

Adder Multiplexer Simulator

Bachelor of Science in Computer Science

Supervisor: DR. MUHAMMAD ASFAND-E-YAR

Department of Computer Science
Bahria University, Islamabad

Certificate

The accepted work contained in this report titled ADDER MULTIPLEXER SIMULATOR, written by AMNA FAROOQ under the supervision of DR. MUHAMMAD ASFAND-E-YAR” as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

Abstract

In this modern age of technology, where every conceptual thing is eased with the help of technology, the concepts and conceptual understandings have made easy for the students. Therefore, this system is designed for the students to get the conceptual knowledge systematically. AMS is a DLD course based educational tool system on logical programming which provides the easy concept learning related to DLD mainly logic gates. Complex hardware structure of circuits and designing them manually has been replaced with this tool. It is beneficial for the students for understanding, learning and solving their DLD related problems by entering the values into the gates of the selected circuit and output will be calculated and displayed. All the arithmetic computation is performed by this system and generates the outputs for both, half adder and the full adder, multiplexers depending upon their given inputs.

Acknowledgement

I would like to thank Allah (SWT) for his countless blessings on me. The most Merciful, who gave me strength to accomplish my goal.

I would also like to express my utmost gratitude for the help and guidance of my respected supervisor, Mr Muhammad Asfand-e-Yar, who has helped me throughout my journey with this project by not only providing me with the support and advice needed to accomplish this project, but for being there for everything required by me. No words, how rich, would do justice to his contributions.

I would also like to thank my friends who motivated me to complete this project. At the end I would like to thank my parents for their love, affection, prayers and support.

Amna farooq Rawalpindi, Pakistan December 2017

Dedication

The Work is dedicated to my parents and my teachers without their hard work and support; I would not have been able to complete this project.

6

**Just because something doesn't do what you planned it to do
doesn't mean it's useless.**

Thomas A. Edison

Contents

1	Introduction	15
1.1	Introduction	15
1.2	Objective	15
1.3	Problem Description	16
1.4	Project Scope	16
1.5	Feasibility Study	16
2	Literature Review	17
2.1	Existing Applications	17
2.1.1	Logisim	17
2.1.2	Logic Gates	18
2.1.3	Logic Gate Simulator	18
2.2	Comparison Table	19
3	Requirement Specifications	21
3.1	Purpose of Document	21
3.2	System Overview and Scope	21
3.3	User Characteristics	21
3.4	Existing System	21
3.5	Functional Requirements	22
3.6	Non-Functional Requirements	22
3.6.1	Reliability	22
3.6.2	Security	23
3.6.3	Performance	23
3.6.4	Maintainability	23
3.6.5	State Maintenance	23
3.6.6	Availability	23
3.6.7	Correctness	23
3.6.8	Usability	23
3.6.9	Reusability	24
3.6.10	Efficiency	24

3.7	Use Cases	24
3.7.1	Use Case Specification: Lets Build	25
3.7.2	Use Case Specification: Half Adder	25
3.7.3	Use Case Specification: Full Adder	26
3.7.4	Use Case Specification: Multiplexer	27
3.7.5	Use Case Specification: My Workspace	28
3.7.6	Use Case Specification: Enter Values	29
4	System Design	31
4.1	System Architecture	31
4.2	Design Methodology	32
4.2.1	Waterfall Model Diagram	32
4.3	Deployment Diagram	32
4.4	Data flow Diagram	33
4.5	System Sequence Diagram [5]	34
4.5.1	Opens Main Workspace	34
4.5.2	Select operation	34
4.5.3	Enter Values	35
5	Implementation	37
5.1	Tool and technologies	37
5.1.1	Unity	37
5.1.2	MonoDevelop	38
5.2	Language	38
5.3	Software Design Pattern	38
5.3.1	Singleton pattern	39
5.3.2	Observer pattern	39
5.3.3	Flyweight pattern	39
5.3.4	Script machine pattern	39
5.4	Methodology	40
5.5	For Usage	40
6	System Testing and Evaluation	41
6.1	Introduction	41
6.2	Testing Evaluation	41
6.3	Testing Technique	41
6.4	Test Cases	43
6.4.1	Installing Application	43
6.4.2	Lets Build	43
6.4.3	Half adder	44
6.4.4	Full adder	45

6.4.5	Multiplexer	45
6.4.6	Checking the Standard half Adder circuit	45
6.4.7	Checking the Standard Full Adder circuit	46
6.4.8	Checking the Standard Multiplexer circuit	47
6.4.9	My Workspace	47
6.4.10	Undo	48
7	Conclusion	49
7.1	Introduction	49
7.2	Achievements and Improvements	49
7.3	Critical Review	49
7.4	Future Recommendations	50
7.5	Future Improvement	50
7.5.1	DLD History	50
7.5.2	Other Smart Phone Applications	50
7.5.3	Updating AMS	50
7.5.4	Other Portals	50

List of Figures

2.1	Logic Gate Simulator	18
2.2	Comparison	19
3.1	Use case for AMS	24
3.2	Use case for selecting Lets Build Button	25
3.3	Use case for selecting Half Adder	26
3.4	Use case for selecting Full Adder	27
3.5	Use case for selecting Multiplexer	28
3.6	Use case for selecting My Workspace	29
3.7	Use case for Entering Values	30
4.1	System Architecture Diagram	31
4.2	Waterfall Model Diagram	32
4.3	Deployment Diagram	33
4.4	Data Flow Diagram	33
4.5	Open Main Workspace Sequence Diagram	34
4.6	Select Operation Sequence Diagram	34
4.7	Enter values Sequence Diagram	35
5.1	Unity Model	38
5.2	Fly weight Pattern	39
6.1	Lets build	43

List of Tables

3.1	Use case for selecting Lets Build button	25
3.2	Use case for selecting Half Adder	26
3.3	Use case for selecting Full Adder	27
3.4	Select Multiplexer	28
3.5	Use case for selecting My Workspace	29
3.6	Enter Values	30
6.1	Tests the Starting Page Button	44
6.2	Half Adder Test case	44
6.3	Full adder Test case	45
6.4	Multiplexer Test case	45
6.5	Testing the Half adder circuit builder	46
6.6	Testing the Full Adder circuit builder	46
6.7	Testing the Multiplexer circuit builder	47
6.8	Testing my workspace button	48
6.9	Functioning of the undo button	48

Chapter 1

Introduction

1.1 Introduction

AMS (Adder Multiplexer Simulator) is designed for DLD course based on the logical programming. DLD course is basically designed for the students to get the knowledge about the fundamentals of digital logic. Digital logic design is basics in the field of electrical engineering and computer engineering. The course provides the concepts of gates how they operate and its application and how to design the logic circuits. Logic gates includes AND, OR, NOT, XOR, NOR, NAND and XNOR [2]. DLD is taught in the starting semester of electrical and computer engineering.

AMS is designed keeping the concept of DLD in mind. In modern age technology is replacing a lot of things. Instead of designing the circuits on pages or designing the complex hardware structures the user have to just enter the inputs into the gates and the desired output will be generated.

AMS an educational tool is basically designed for students which will help them understand how to design the circuits and what result will be generated by the each gate by just entering the input value. Complex and larger circuits can also be designed on this application. The user will enter the value in 0s and 1s and the output will be generated in less than 5 seconds.

1.2 Objective

To design software that will generate an output and circuit of half adder, full adder and multiplexer depending upon the inputs. The objective of this project is to make easy and convenient DLD circuit for user for not wasting time on large process of diagrams through gates and truth tables and easily get results by putting values. Most people expect computers to do some kind

of arithmetic computation, and thus, most people expect computers to do the operations of add, subtract, multiply and divide. It will also help user to match results with the hardware result that it is correct or not. This will help an individual person as well as for larger projects.

1.3 Problem Description

Digital Logic Design is used to develop hardware, such as circuit boards and microchip processors. Basic programming skills are essential for every student to build their logic but most of them have difficulty in building basic logic. Programming is essential in computer science and information technology. Students have to learn DLD and for the purpose of learning mostly hardware, specifically DLDs hardware or manually using the concepts of DLD and this manual work is avoided by using the programming concepts and logic used in this project which helps the user to solve their logical problems just by entering the values, and will give the required results processed through programming, hence this is a much easier way of learning data logic and designing more easily and with chances of more accurate results.

1.4 Project Scope

The scope of the project is to construct a program that will help the DLD students in the combinational logic circuits that perform binary addition with half adder, full adder and multiplexer. The project is for the beginners of DLD course work.

User will drag and drop the gate on work space and will give the input values into the gate in the form of 0s and 1s and the desired output will be generated in the result box. Our purpose of using unity platform is to easily convert the code for other Smart phone. Basically the application can be developed as a desktop application and android application.

1.5 Feasibility Study

The proposed project is for undergraduate level students and almost all the technologies and tools to be used in designing and implementation of the application have been taught in separate courses unity software and monodevelop text editor have been taught in computer graphics course. Therefore, the project will be doable within the given time.

Chapter 2

Literature Review

2.1 Existing Applications

2.1.1 Logisim

Logisim is an educational tool it is used for simulating and designing the digital logic. It helps the younger students in learning the basic concepts that are related to logical circuits such as gate operations (AND, OR, NOT, XOR, NAND, NOR). It has a simple GUI (graphical user interface). It has the capacity to build larger and complex circuits from the smaller circuits and it can draw a bundle of wires with a single drag and drop option. Logisim software can also be used to simulate and design entire CPUs for the educational purpose.

Logisim is basically used by the students of college and university level all around the world in many different type of classes, mainly in computer science field the students have to learn DLD subject and for this purpose hardware mostly wires the processors are being used and this all manual work is being avoided by using the programming and logic concept. By just entering the values the required result is being obtained. Hence this is a very easy way of learning data logic and designing the complex circuits.

Logisim is a very simple tool totally portable and free to use it as an educational tool for learning the basic logical circuits. It also helps you to learn how the circuits work. Logisim allows the user to change the circuit during the simulation. Circuits can also be saved as a program file, exported to GIF file or can be printed on a printer.

Logisim is a java application that can run on many platforms for example MacOS X, Linux and Windows. It uses the combinational analysis module. It is a free software. The drawing interface is an intuitive toolbar. Color coded wires can be used in debugging and simulating a circuit. The lines

can be drawn in any way horizontal or vertical depend upon the requirement and they can be automatically connected to the components and other wires. Circuit drawing is very easy. Circuits layouts can be used as the sub circuits of the other circuit allowing the hierarchical circuit design. Its combinational analysis module allows the conversion between the truth tables, circuits and Boolean expressions.

2.1.2 Logic Gates

Logic gates application is for new readers who want to make concepts about Digital logic design. Application consists of detailed knowledge of AND, OR, XOR, NOT, NAND, XNOR but not facilitate user with emulator where user can learn the concepts with using these gates[3]. Our application provides user a workspace where two inputs are given by user and select the operation, in result output will display.

2.1.3 Logic Gate Simulator

Logic gate simulator is a graph circuit activity, which are used by drag and drop layout and wiring. A variety of visual modes including isolating partial circuits to show use, hiding intermediate gates, and not showing different colors for true/false useful for preparing circuits for use in documents. Most of its gates are not working properly as NAND gate is not working the way it should perform its operation[4].

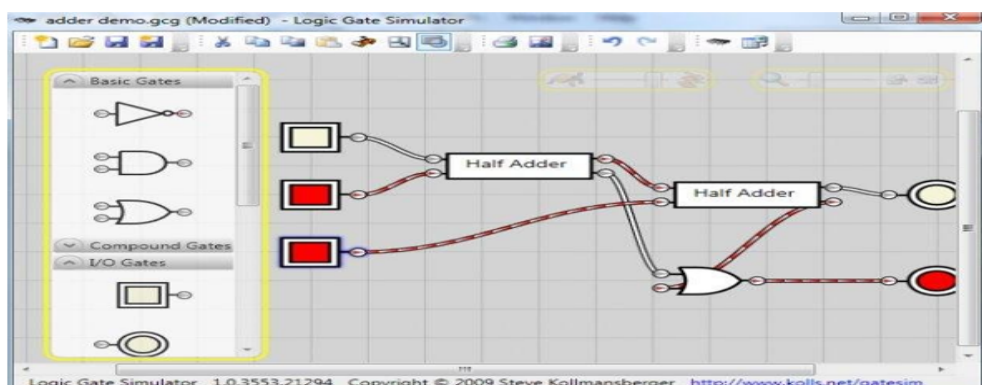


Figure 2.1: Logic Gate Simulator

2.2 Comparison Table

Following is the short comparison between existing system and our system. The purpose is to find out how our system is dominant on existing system, and facilitate to user as shown in below table.

Features of AMS	Logisim	Logic Gates	Logic Gate Simulator
User friendly Interface	Logisim is not as much user friendly, as application AMS:8bit ALU provides user friendly interface	App is user friendly but user doesn't found any functionality as our app.	Software is not as much user friendly because for new user it takes some time to make understandability
User reviews	Logisim have not well and value able customer's reviews. Our system's features compel people to make good reviews about our system	User reviews are not satisfied able because user doesn't found require functionality	User reviews is satisfactory.
Visual Interaction	Existing system have less visual representation for user, our system provides user friendly and well color visual representation to users.	Logic gates has normal visual representation for user, AMS provides authentic well graphics representation.	Visual representation is complex for new user.

Figure 2.2: Comparison

Chapter 3

Requirement Specifications

3.1 Purpose of Document

The purpose of document is to explain requirement specifications which are used in making this application. This chapter includes working of application, its specification, functionalities, and constraints of the half adder, Full adder and multiplexer under which the system will operate and use cases. This document describes all functional and non-functional requirements of the system.

3.2 System Overview and Scope

The software is based on logic that will help students to learn the basic concepts of DLD in the combinational logical circuits by just giving the input values that will perform binary addition with half adder, full adder and multiplexer.

3.3 User Characteristics

There are no such major user characteristics involved related to the user, in order for him/her to operate the application. The user just needs to know how to open the application and to enter the binary values.

3.4 Existing System

There are a number of softwares on Circuits. One of them is Logic gates, its basically an educational tool. Drawbacks of the mentioned application are

following:

- Interface is not user friendly.
- User reviews are not satisfactory.
- It just gives the knowledge about the gates.
- Lack of visual interaction.

3.5 Functional Requirements

1. Start application

- User must have started the application.
- User must have click on the lets build button.

2. Select operation

- User must select the operation he/she wants to perform i.e. Adder, Full Adder, and Multiplex or simple.

3. Input values

- User must be able to input the binary values in the combo box.

4. View the result

- View the result according to the input.

5. Close application

- Close the application.

3.6 Non-Functional Requirements

The Requirements that system must have to accomplish along the side of the functional requirements. The non-functional requirements of the system are following.

3.6.1 Reliability

The application should be reliable the user will just enter the input value and the expected result should be displayed. If in some state a bug occurs the system should work properly at users end.

3.6.2 Security

The system will limit access to authorized users. The user should not make changes other than the privileges assigned to them.

3.6.3 Performance

The system should perform accordingly to the use cases and other flow diagrams, as the system will give an alert at the time if the circuit is not according to the given operation.

3.6.4 Maintainability

The system needs to be cost-effective to maintain. The system has the capability to be enhanced and improve further. The system has capability to further expand.

3.6.5 State Maintenance

Backup copies of different versions of the system should be kept to avoid any accidental loss of data.

3.6.6 Availability

The system does not require any Internet connectivity and will be available once the application has been installed on a desktop.

3.6.7 Correctness

The system should provide all the functionalities as they are stated in requirements.

3.6.8 Usability

The system should be easy to use by any of the user who can operate the Smart phone and use computer. If she/he has little knowledge about the both devices they can use the application easily. It shouldnt require any training for using the application

3.6.9 Reusability

The systems should be developed in a way that it would ensure high reusability in enhancement of the application in future or reuse of modules in any other related future projects.

3.6.10 Efficiency

This application should be efficient regarding speed as well as memory usage. It should fast enough to satisfy expectations of the users. It should have less response time and quick in completing the tasks.

3.7 Use Cases

The case scenario will describe overall working or functionality of the application. How the user will interact with the application [6].

Actors

- User
- System

Use case Diagram OF AMS

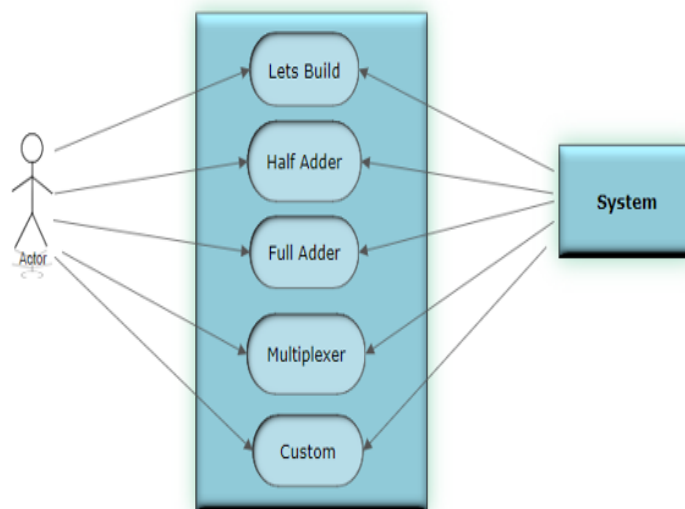


Figure 3.1: Use case for AMS

Figure 3.1 Use case diagram of AMS have been shown the overall interaction of the user with the application.

3.7.1 Use Case Specification: Lets Build

Figure 3.2 and Table 3.1 show the interaction of the user with the system as he/she selects the button. The user clicks the lets build Button.

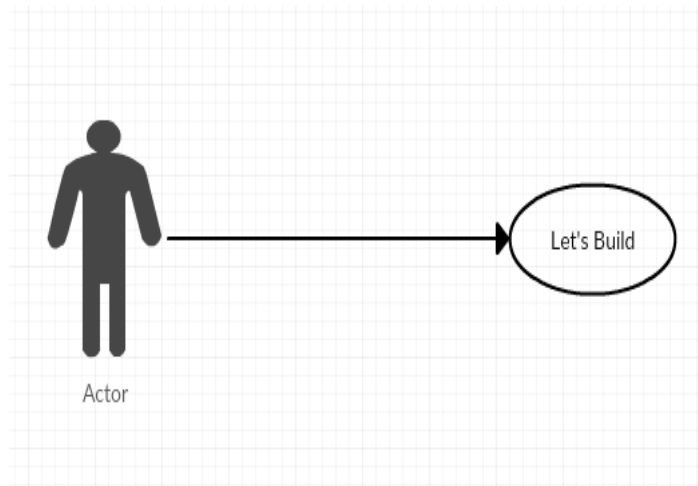


Figure 3.2: Use case for selecting Lets Build Button

Use Case ID	UC01
Name	Lets Build
Description	User can click lets build button.
Primary Actor	User.
Pre-Condition	User must visit the application.
Stake holders	User
Post-condition	Successfully clicks the button.
Main Scenario	<ol style="list-style-type: none"> 1. Select Lets build button. 2. System initiates use case. 3. Successfully main workspace appears.
Exception	<ol style="list-style-type: none"> 1. If user does not click the button next main workspace will not appear.

Table 3.1: Use case for selecting Lets Build button

3.7.2 Use Case Specification: Half Adder

Figure 3.2 and Table 3.1 it has been shown the interaction of the user with the as he/she selects the operation which has to be perform. The user clicks

the button and the Half Adder workspace gets opened.

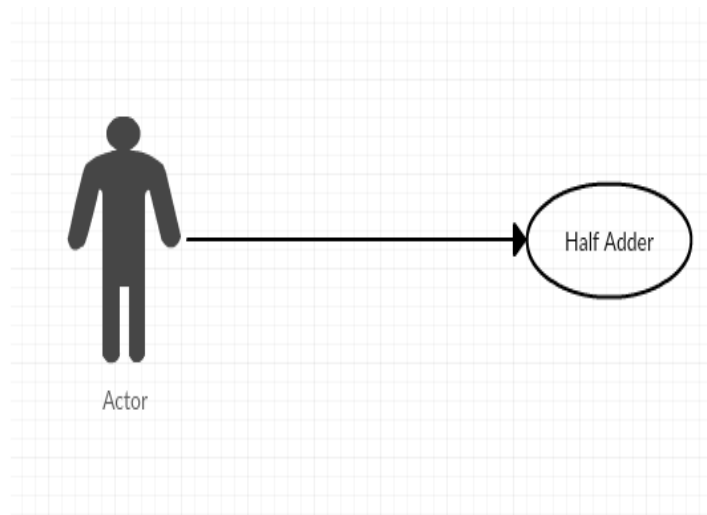


Figure 3.3: Use case for selecting Half Adder

Use Case ID	UC02
Name	Half Adder
Description	User selects the Half Adder
Primary Actor	User.
Pre-Condition	User must select half adder
Stake holders	User
Post-condition	Successfully selected
Main Scenario	<ol style="list-style-type: none"> 1. User selects the operation Half Adder. 2. System opens the workspace of Half Adder. 3. User designs half adder.
Exception	None

Table 3.2: Use case for selecting Half Adder

3.7.3 Use Case Specification: Full Adder

Figure 3.2 and Table 3.3 it has been shown the interaction of the user with the as he/she selects the operation which has to be perform. The user clicks the button and the Half Adder workspace gets opened.

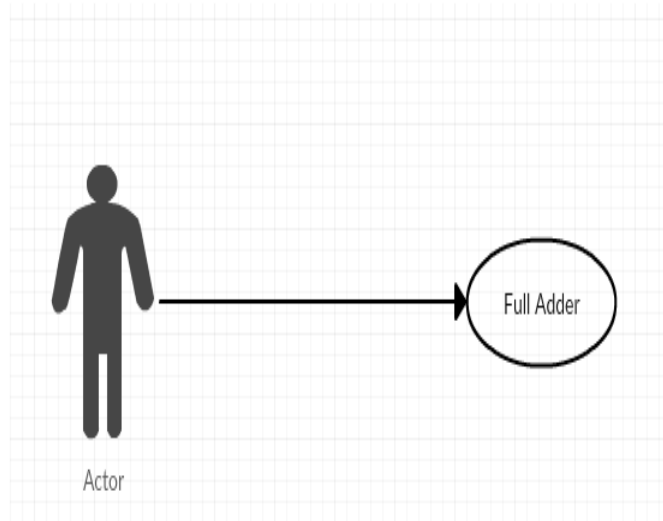


Figure 3.4: Use case for selecting Full Adder

Use Case ID	UC03
Name	Full Adder
Description	User selects the Full Adder
Primary Actor	User.
Pre-Condition	User must select Full adder
Stake holders	User
Post-condition	Successfully selected
Main Scenario	<ol style="list-style-type: none"> 1. User selects the operation Full Adder. 2. System opens the workspace of Full Adder. 3. User designs Full adder.
Exception	None

Table 3.3: Use case for selecting Full Adder

3.7.4 Use Case Specification: Multiplexer

Figure 3.2 and Table 3.4 it has been shown the interaction of the user as he/she selects the operation which has to be performed. The user clicks the button and the multiplexer workspace gets opened.

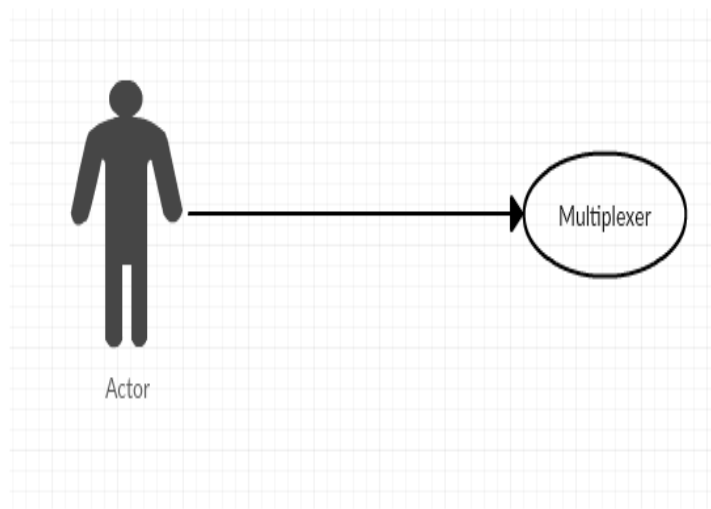


Figure 3.5: Use case for selecting Multiplexer

Use Case ID	UC04
Name	Multiplexer
Description	User selects the Multiplexer
Primary Actor	User.
Pre-Condition	User must select Multiplexer
Stake holders	User
Post-condition	Successfully selected
Main Scenario	<ol style="list-style-type: none"> 1. User selects the operation Multiplexer. 2. System opens the workspace of Multiplexer. 3. User designs Multiplexer.
Exception	None

Table 3.4: Select Multiplexer

3.7.5 Use Case Specification: My Workspace

Figure 3.2.5 and Table 3.1 it has been shown the interaction of the user as he/she selects the operation which has to be performed. The user clicks the button and the My Workspace gets opened.

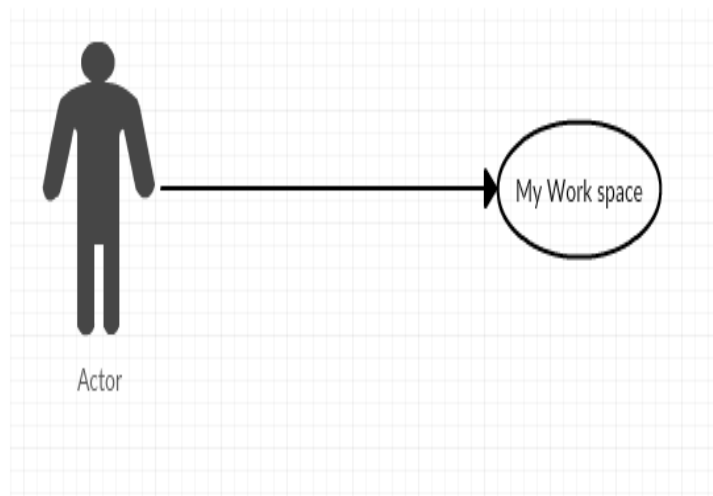


Figure 3.6: Use case for selecting My Workspace

Use Case ID	UC05
Name	My Work space
Description	User selects the My workspace
Primary Actor	User.
Pre-Condition	User must select My workspace
Stake holders	User
Post-condition	Successfully selected
Main Scenario	<ol style="list-style-type: none"> 1. User selects the operation my workspace. 2. System opens the workspace of My workspace. 3. User design Circuit.
Exception	None

Table 3.5: Use case for selecting My Workspace

3.7.6 Use Case Specification: Enter Values

Figure 3.6 and Table 3.1 it has been shown the interaction of the user as how he/she enter values into the combo box.

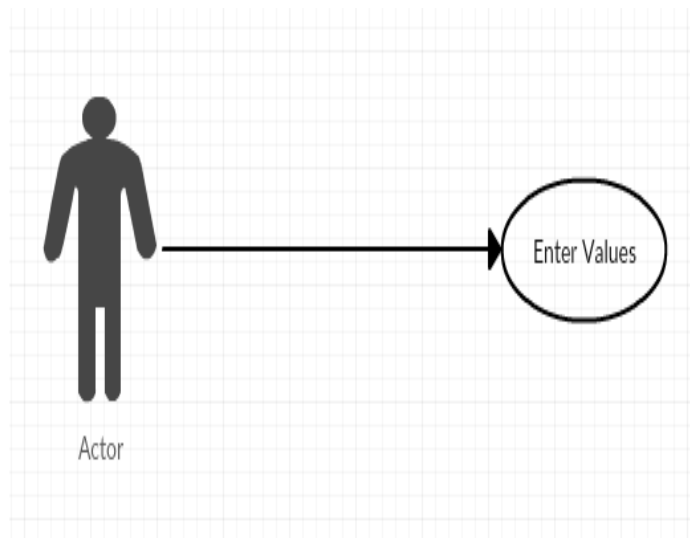


Figure 3.7: Use case for Entering Values

Use Case ID	UC06
Name	Enter values
Description	User enters the values in combo boxes.
Primary Actor	User.
Pre-Condition	User must be selects the operations
Stake holders	User
Post-condition	Successfully enters the values and get the output
Main Scenario	<ol style="list-style-type: none"> 1. User enters the values in combo boxes. 2. System initiates the use case. 3. Successfully get the result.
Exception	<ol style="list-style-type: none"> 1. If user enters invalid values, System shows prompt error Invalid value

Table 3.6: Enter Values

Chapter 4

System Design

The design and architecture of the project and overview of how different processes will be carry out is discussed in this chapter.

4.1 System Architecture

The desktop application which is intended to be developed has very simple architecture design. First proposed system will get the Input values from the user and then process on the values by using C sharp to give the desired output. Below is the context diagram showing the boundary of the proposed system, entities their interaction, and its environment.

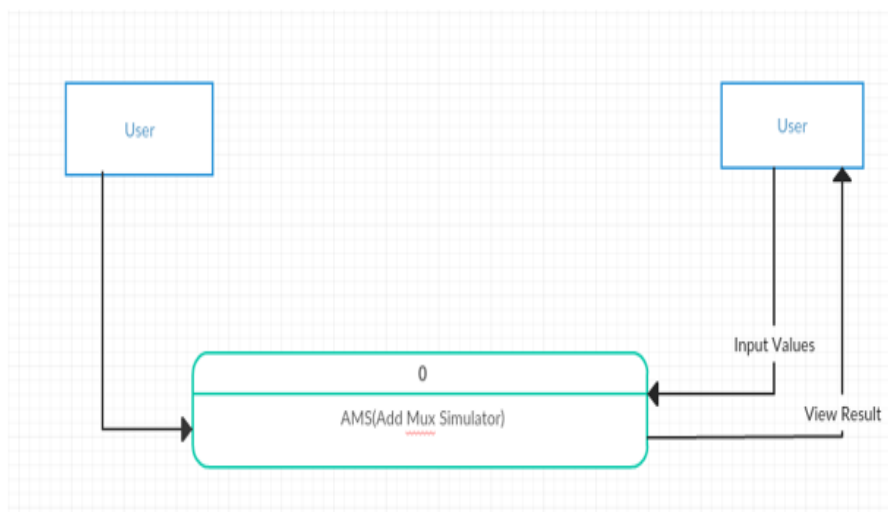


Figure 4.1: System Architecture Diagram

4.2 Design Methodology

Waterfall model develop this project as our requirement are clear and there were no ambiguous requirement as I followed the water fall model to implement the project. According to requirements of our project, the resources are assembled to complete the project.

4.2.1 Waterfall Model Diagram

First phase of model consists of requirement document and use cases thatll help us identifying and gathering requirements, in designing phase I will work on software architecture, UML, and Graphical User Interface. Implementation will start after designing that contains code generation, GUI construction, integration of components, verification phase consists of code installation, testing and debugging.

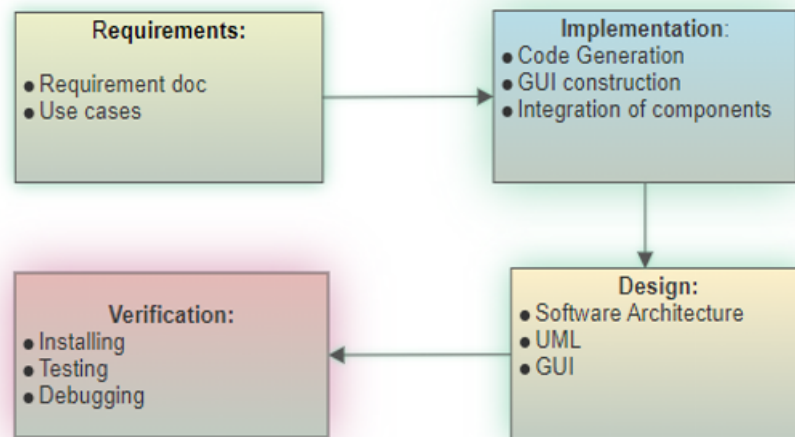


Figure 4.2: Waterfall Model Diagram

4.3 Deployment Diagram

Deployment diagram shows how system will work when it is deployed in its domain and the interaction of components in the system. In the system C sharp language and basic libraries are used and are deployed on desktop system.

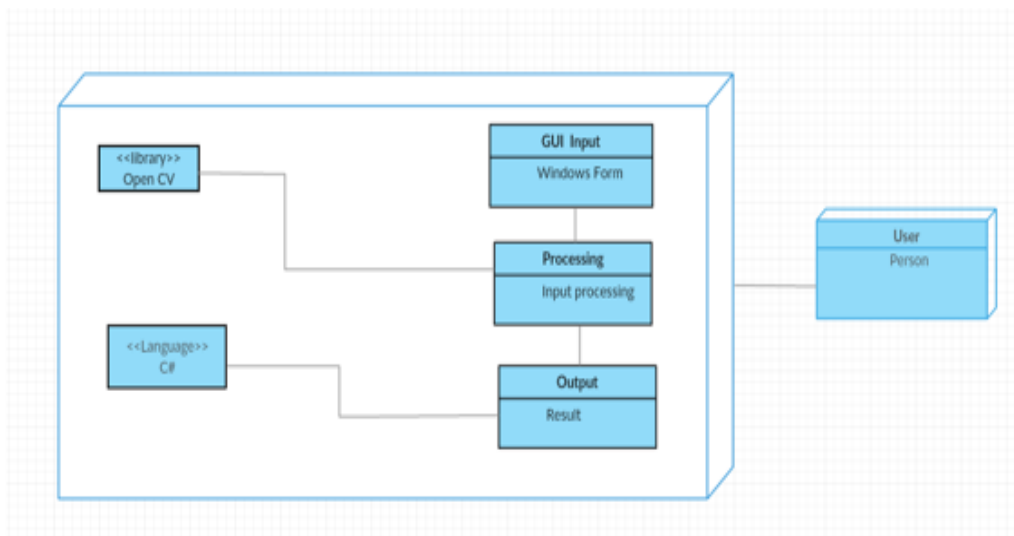


Figure 4.3: Deployment Diagram

4.4 Data flow Diagram

Data flow diagram (DFD) interprets how data is processed by a framework in terms of inputs and outputs. DFD emphasis on the flow of data, where input comes from, where it goes and how it gets processed.

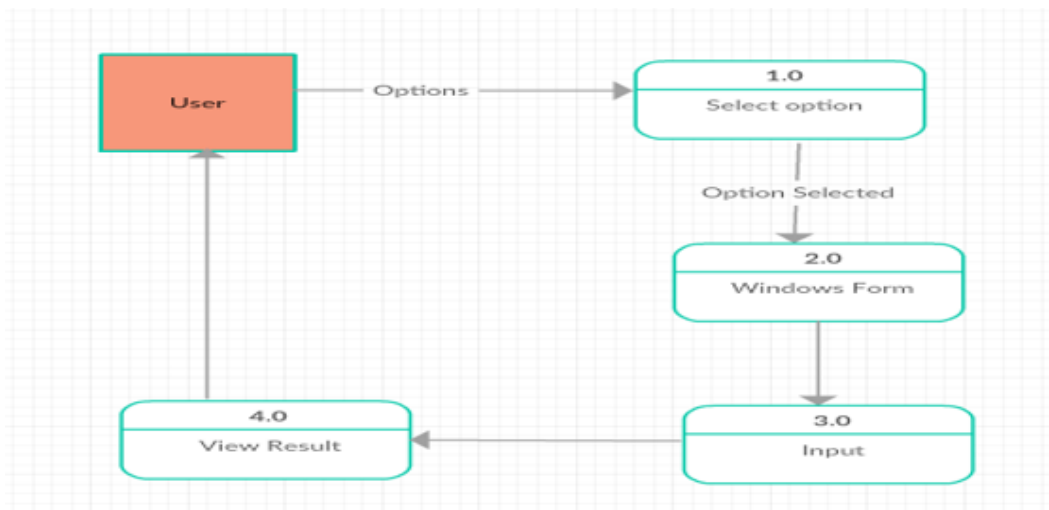


Figure 4.4: Data Flow Diagram

4.5 System Sequence Diagram [5]

4.5.1 Opens Main Workspace



Figure 4.5: Open Main Workspace Sequence Diagram

4.5.2 Select operation

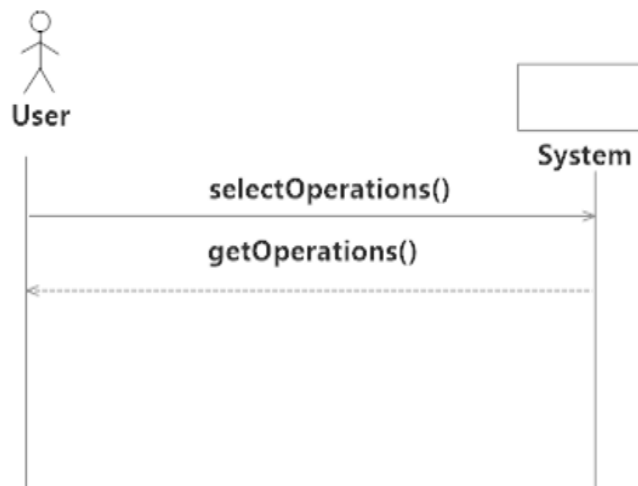


Figure 4.6: Select Operation Sequence Diagram

4.5.3 Enter Values

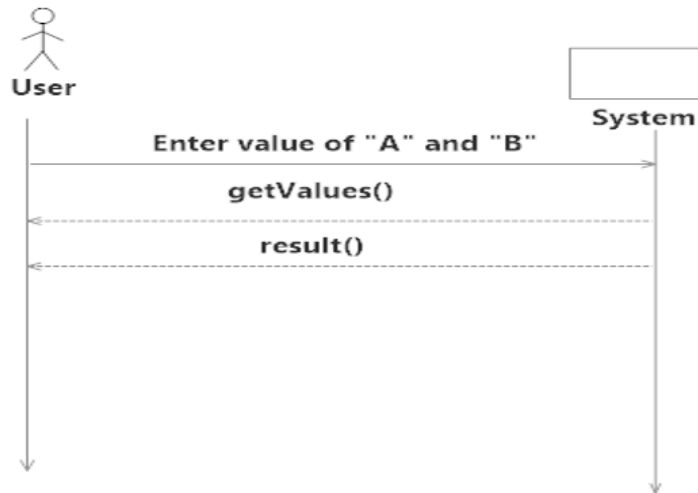


Figure 4.7: Enter values Sequence Diagram

Chapter 5

Implementation

Implementation chapter will provide a complete overview of tools, technologies, libraries and development techniques used to develop the application.

5.1 Tool and technologies

Following are the tool and technologies which have been used for the development of the application.

5.1.1 Unity

The project is constructed in Unity 5.6.0. Unity is across platform game engine developed by unity technologies for the purpose of developing gaming application and stimulation software for PC, consoles and mobile devices. Unity supports multiple languages like C sharp and JavaScript and comes with a default cross-platform compiler Mono Develop. The basic purpose of using this technology was to make an application that runs successfully on windows based computers and android phones due to the simple integration of third party code and libraries allowed here. I am using this software for my interface designing.

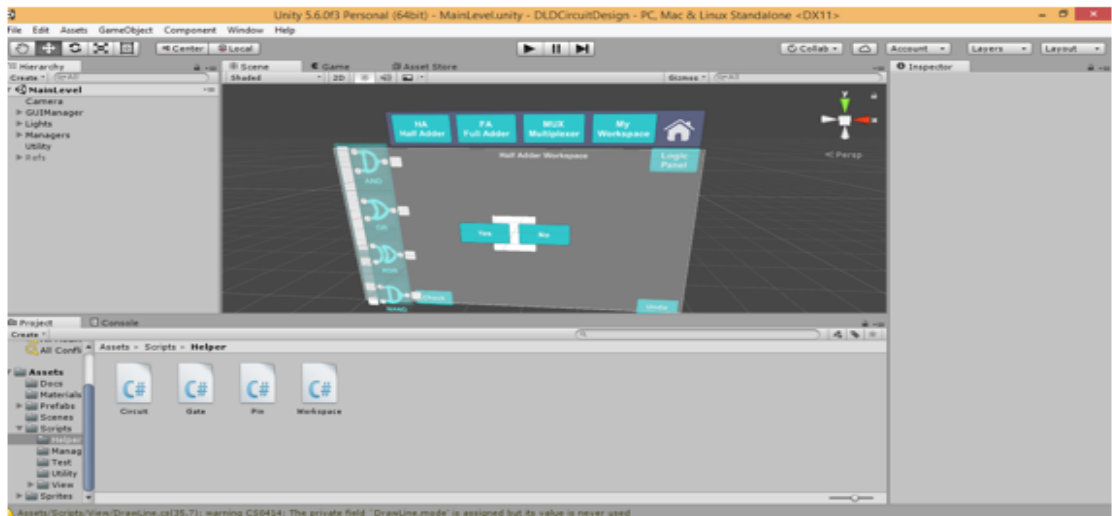


Figure 5.1: Unity Model

5.1.2 MonoDevelop

The external editor of unity is MonoDevelop. I have used MonoDevelop editor with C sharp language. C sharp being fast and robust was opted for the development of this project. Unity and MonoDevelop both are basically separate applications but unity has the capability to synchronize MonoDevelop with itself. The basic gate operation which it will perform has a coding that is done in MonoDevelop.

5.2 Language

The language that has been used for the development of the application is C sharp. C sharp is a significant language to use for logic because its so powerful. C sharp being fast and robust was opted for the development of the project. With this logic, you can drag and drop the gate on the workspace, provide the input and output will get generated according to the gate and much more.

5.3 Software Design Pattern

Four types of software design patterns are being used in the application:

5.3.1 Singleton pattern

It is used for global communication. It controls the workspace order.

5.3.2 Observer pattern

Observer pattern, this pattern is implemented to broadcast events. The events can be subscribed and the subsequent actions are implemented to maintain the flow of the program. The output from the output pins were broadcasted so the subscriber pins can update their inputs accordingly in real-time.

5.3.3 Flyweight pattern

In the project this pattern basically works as if a gate is dropped from the component panel into the workspace it works as the child is dropped into the workspace but the parent remains at its place so that again a child can be dropped into the workspace.

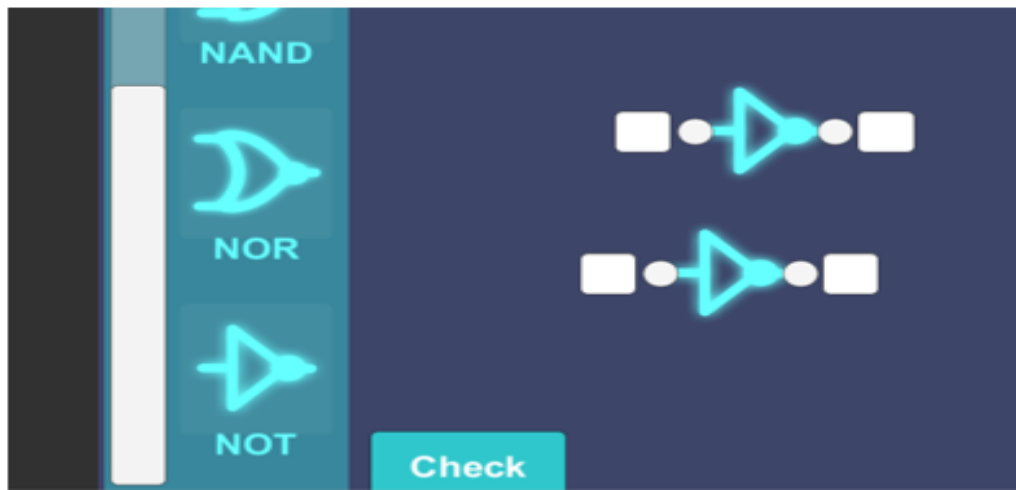


Figure 5.2: Fly weight Pattern

5.3.4 Script machine pattern

Which basically controls the state of the workspace.

5.4 Methodology

In the proposed project user, will enter the values and then gate will process the values for the desired output that will be through a half adder or full adder and for multiplexer. There are 7 gates AND, OR, NOT, XOR, NOR, NAND, and XNOR [1]. User will select which gate he/she wants to use for the desired output. According to the requirement gate is dragged and dropped by the user. The combo box can only take binary values like 0 and 1, if the user puts value greater than 1 in the combo box it will automatically change into 1 and if the user puts less than 0 it will change into 0. For Designing and implementation of proposed project requirements plays a very important part. For starting the project requirements and information about the gates I will gather. To achieve the desire target I will use appropriate logic, schemes and model (i.e. waterfall model) from the different resources. Once the application is ready, it will be tested for bugs and errors shortly after which it will be handed over to the user for reviews. Updates will be made keeping in consideration the users view point.

5.5 For Usage

- Desktop
- Android

Chapter 6

System Testing and Evaluation

6.1 Introduction

The main objective of the system is to developing software that will generate circuit of half adder, full adder and multiplexer depending user demand. The main purpose of the project is to make easy and convenient DLD circuits for user for not wasting time on large process of diagrams through gates easily get results by putting values. System will also help user to match results with the hardware result that it is correct or not in just a second. This chapter includes how the system is tested.

6.2 Testing Evaluation

Software testing is the process to find out the bugs and errors from developed software, whether the Software evaluations purpose is to verify that developed software is meet with clients requirements that are to be define at early stage of development.

Purpose:The purpose of Software testing and evaluation is to determine the system whether it perform accurately as expected or not. The developed software is exactly build according to clients requirements. To find out these all aspects some testing and evaluation techniques are performed, to validate the system.

6.3 Testing Technique

- **Black Box Testing** Black box testing also known as behavioral testing, to know the system behavior toward users. This is a testing method

in which the internal structure, design and functionality of the system is to be tested.

- **Test case Scenario** A scenario is a story that describes a hypothetical situation. In testing, you check how the program copes with some specific hypothetical situation.

The following types of testing that should be considered during system testing.

1. Graphical User Interface

- The interface of the system is carefully designed and should be user friendly if the user has minimum knowledge of using this application.
- New user can easily understand the graphical representation of the application.

2. Usability Testing

- The usability testing is not the only option to check the user interaction according to his/her least knowledge its about testing each gate is operating the way it has to be operating.
- Hallway testing is performed. According to Hallway testing, random students are picked and ask to try out the product. Random students had performed the test of the application and were asked to give the inputs to the gates so that they can check the output by that gate whether it is correct according to the given gate. Minor errors are identified by the students using the interface of both desktop and android application. These errors are fixed after the test.

3. Software performance Testing

- System takes maximum 5 seconds to give output.
- System is more efficient to perform its core functionality.

4. Load Testing

- System performs normally on load testing. Can put different functionalities at the same time, but the system performance should be balanced.

5. Installation Testing

- User must install application on his/her system as it is desktop application.
- For Android users they must install app through Google play store.

6.4 Test Cases

In these section details of the system testing was carried out by designing a number of test cases as discussed in the following sections.

6.4.1 Installing Application

The application is to be installed on windows or android.

6.4.2 Lets Build

This Test Case is used to define the testing of the lets Build Button.

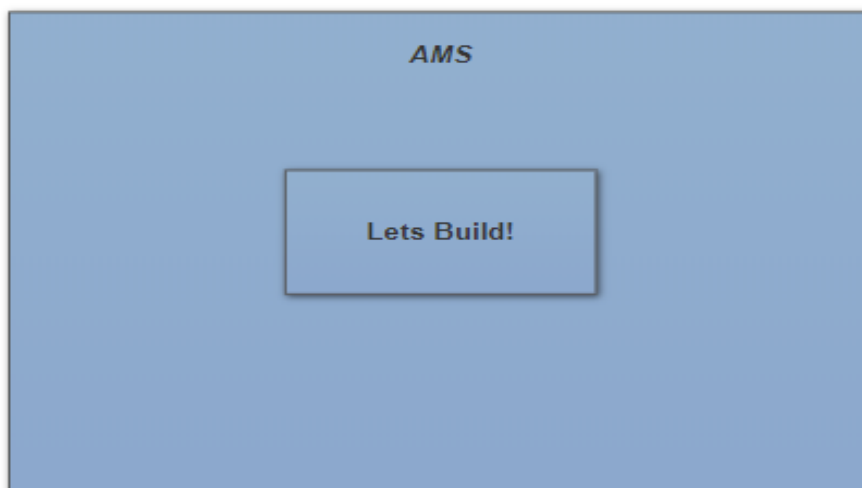


Figure 6.1: Lets build

Test Case ID	02
Description	Tests the Starting page button
Application for	Window desktop systems and android phone.
Requirement	Application is to be installed on windows with required condition
Steps to be Taken	1. Open the home screen 2. Enter the lets build button
Step	Procedural Steps 1)Turn on the computer 2)start the application 3) Click the lets build button.
Step	Task and expected result 1)Main button option if clicked correctly 2) If the system powers off, the same interface on which it was closed reappears on switching on the application.
Remarks	N/A

Table 6.1: Tests the Starting Page Button

6.4.3 Half adder

This Test Case is use to define the testing of the Half Adder Button.

Test Case ID	03
Description	Testing the half adder
Application for	Window desktop systems and android phone.
Requirement	Application is to be installed on windows with required condition
Step	Procedural Steps 1) Select the option by pressing button Half Adder
Step	Task and expected result 1) Half adder workspace appears if button half adder pressed.
Remarks	N/A

Table 6.2: Half Adder Test case

6.4.4 Full adder

This Test Case is use to define the testing of the Full Adder Button.

Test Case ID	04
Description	Testing the Full adder
Application for	Window desktop systems and android phone.
Requirement	Application is to be installed on windows with required condition
Step	Procedural Steps 1) Select the option by pressing button Full Adder
Step	Task and expected result 1) Full adder workspace appears if button Full adder pressed.
Remarks	N/A

Table 6.3: Full adder Test case

6.4.5 Multiplexer

This Test Case is use to define the testing of the Multiplexer Button.

Test Case ID	05
Description	Testing the Multiplexer
Application for	Window desktop systems and android phone.
Requirement	Application is to be installed on windows with required condition
Step	Procedural Steps 1) Select the option by pressing button Multiplexer
Step	Task and expected result 1) Multiplexer workspace appears if button Multiplexer pressed.
Remarks	N/A

Table 6.4: Multiplexer Test case

6.4.6 Checking the Standard half Adder circuit

In this test case we tested the circuit of half adder is correct or not correct by clicking the check button presents on half adder workspace.

Test Case ID	06
Description	Testing the Half adder circuit builder
Application for	Window desktop systems and android phone.
Requirement	Application is to be installed on windows with required condition
Step	Procedural Steps 1) Draw the circuit diagram. 2) Then press the check button.
Step	Task and expected result 1) Match the current diagram with the half adder diagram.
Remarks	N/A

Table 6.5: Testing the Half adder circuit builder

6.4.7 Checking the Standard Full Adder circuit

In this test case we test the circuit of Full adder is correct or not correct by clicking the check button presents on half adder workspace.

Test Case ID	07
Description	Testing the Full Adder circuit builder
Application for	Window desktop systems and android phone.
Requirement	Application is to be installed on windows with required condition
Initial condition	Full adder form will appear by pressing the half adder button
Step	Procedural Steps 1) Draw the circuit diagram. 2) Then press the check button.
Step	Task and expected result 1) Match the current diagram with the Full adder diagram.
Remarks	N/A

Table 6.6: Testing the Full Adder circuit builder

6.4.8 Checking the Standard Multiplexer circuit

In this test case we test the circuit of Multiplexer is correct or not correct by clicking the check button presents on multiplexer workspace.

Test Case ID	08
Description	Testing the Multiplexer circuit builder
Application for	Window desktop systems and android phone.
Requirement	Application is to be installed on windows with required condition
Initial condition	Multiplexer form will appear by pressing the Multiplexer button
Step	Procedural Steps 1) Draw the circuit diagram. 2) Then press the check button.
Step	Task and expected result 1) Match the current diagram with the Multiplexer diagram.
Remarks	N/A

Table 6.7: Testing the Multiplexer circuit builder

6.4.9 My Workspace

This test case is used to design the testing of My Workspace button.

Test Case ID	09
Description	Testing the My Workspace Button
Application for	Window desktop systems and android phone.
Requirement	Application is to be installed on windows with required condition
Initial condition	My Workspace will appear by pressing the My Workspace button.
Step	Procedural Steps 1) Select the option by pressing button My Workspace.
Step	Task and expected result 1) My workspace appears if button My Workspace pressed.
Remarks	N/A

Table 6.8: Testing my workspace button

6.4.10 Undo

This test case is used to design the testing of undo button whether the state changes to its previous state or not.

Test Case ID	10
Description	Functioning of the undo button
Application for	Window desktop systems and android phone.
Requirement	Application is to be installed on windows with required condition
Initial condition	Current state can be changed to previous state by pressing the undo button.
Step	Procedural Steps 1) Select the option by pressing button Undo
Step	Task and expected result 1) Current state is changed to previous state.
Remarks	N/A

Table 6.9: Functioning of the undo button

Chapter 7

Conclusion

7.1 Introduction

With the grace of Allah, the project is completed. This is the last chapter of my report in which I will describe the achievements, improvements that can be made to our project, critical review and future recommendations.

7.2 Achievements and Improvements

This section of document briefly describes the work done till now and the achievements which are made so far in development of application. I have improved my concepts about C sharp language and how to develop logic in C sharp and got a full control over unity. This project has also enabled me to learn several new tools which have only added to my skill repository. This, along with the challenges that I faced throughout the process, has helped me evaluate my skills and learn how to build upon them, which is something that will aid me in my professional career for a long time to come. Learnt about how to handle requirement changes, practically. Additionally, problem solving skill is also improved.

7.3 Critical Review

There are many existing systems that provide same functionality like our system, but every system has its own pros and cons which had been discussed in Chapter 2. So, the effort was to make the system to improve and enhance with more features to provide more ease to users.

7.4 Future Recommendations

I have planned to make improvements in terms of efficiency and usability and enhance the system by adding more Electronics functions in it. So, if user wants multiple operations he/she doesn't need to go to many applications, our system provides one platform where he/she can use multiple features regarding to his/her need. This Document covers detailed structure of our system through description, diagrams, Literature Survey. Application is almost completed with respect to coding and all Testing phases.

7.5 Future Improvement

Many improvements can be made to this project but because of time constraint and other course loads, it was unfortunately not possible to do more in the project. These improvements will be accomplished in future.

7.5.1 DLD History

The application right now do not maintains a complete DLD concept i.e. flip flop, subtractor etc But in future this application can be made more useful and more helpful to the users.

7.5.2 Other Smart Phone Applications

In future, the app can also be developed for iOS and Windows Smart phone users. The main purpose of using the Unity was to develop native applications for other platforms also.

7.5.3 Updating AMS

The application will be updated in future for all electronics functions, which will also include all types of DLD concepts as much possible.

7.5.4 Other Portals

The application will be used as portal for those students in universities which are in their higher semesters so this can help them by software means.

References:

- [1] M.Morris Mano,Digital Design,4th Ed,2014,cited on P. 32
- [2] Margaret Rouse May 2015 logic gate (AND, OR, XOR, NOT, NAND, NOR and XNOR)
Retrieved from <http://whatis.techtarget.com/definition/logic-gate-AND-OR-XOR-NOT-NAND-NOR-and-XNOR>
- [3] Application, Education, Logic Gates, cited on P. 16
Retrieved from <https://play.google.com/store/apps/details?id=logic.gates.app&hl=en>
- [4]Steven Kollmansberger, 2014-12-15 Source Forge, Logic Gate Simulator, cited on P. 16
Retrieved from <https://sourceforge.net/projects/gatesim>
- [5] System Sequence Diagrams, cited on P.28
<https://www.websequencediagrams.com>
- [6] Use Case Diagrams, cited on P. 21
<https://www.smartdraw.com>