Mini Project Report – Applied System Development

./



Life Cycle and Software Testing

LAB-07



### Document History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SI.**  **No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **To be approved By** | **Remarks/Revision Details** |
| 1 | 24/08/21 | 99005726 | 99005726 | Kartik Mudaliar |  |
| 2 | 24/08/21 | 99005729 | 99005729 | Kartik Mudaliar |  |
| 3 | 24/08/21 | 99005721 | 99005721 | Kartik Mudaliar |  |
| 4 | 24/08/21 | 99005713 | 99005713 | Kartik Mudaliar |  |
| 5 | 24/08/21 | 99005708 | 99005708 | Kartik Mudaliar |  |
| 6 | 24/08/21 | 99005714 | 99005714 | Kartik Mudaliar |  |
| 7 | 24/08/21 | 99005734 | 99005734 | Kartik Mudaliar |  |
| 8 | 24/08/21 | 99005758 | 99005758 | Kartik Mudaliar |  |
| 9 | 24/08/21 | 99005735 | 99005735 | Kartik Mudaliar |  |
| 10 | 24/08/21 | 99005741 | 99005741 | Kartik Mudaliar |  |
| 11 | 24/08/21 | 99005746 | 99005746 | Kartik Mudaliar |  |



**TABLE OF CONTENTS**

* 1. Introduction
  2. Research
  3. State of Art
  4. 4W's and 1'H
  5. SWOT Analysis
  6. Features
  + High Level Requirements:
  + Low level Requirements:
  1. 7. Architecture
  + High Level Diagram:
  + Use Case Diagram:
  + Component Diagram:
  + Low Level Diagram:
  + Filter use case diagram:
  + Logic gate Activity diagram:
  + Oscillators Use Case Diagram:
  + Rectifier Activity diagram:
  + Flip Flop Activity Diagram:
  + Opamp Part –B Behavioral Diagram:
  + Capacitor Activity Diagram:
  1. Implementation
  2. TEST PLAN
     + High level test plan:
     + Low level test plan:
  3. Git Inspector
  4. Output:
  5. **Daily Stand Up**



# Introduction

This product is a powerful tool which helps in designing and implementing electronic circuits. It has various components as features like Filters, Oscillators, Amplifiers, Logic gates, etc which are related to electronics. It takes input from the user through Command line and gives the output through the same. In this project the programming language used is C.

# Research

Many websites were available which would implement only one of the features of our product alone. For example, there are pages or sites for calculating filters RC values or implementing logic gates. But we could not find anything integrated to this extent.

# State of Art

**Ageing - Time**

* Usually these types of calculations were done manually, handwritten. Even in colleges, we still practice all these manually. This project is an idea for automating all the calculations and design which will reduce the time consumed and errors committed.

**Ageing - Cost**

* The cost parameter for this project was none. All the softwares and features were freely available on the internet.

# 4W's and 1'H

**Who:**

* Electrical, Electronics, control systems, instrumentation engineers, students and people who need to design or use circuits related to electronics.

**What:**

* This project gives out the parameters or the values required to design or implement the type of electronic component according to the user inputs.

**When:**

* This project can be used whenever there is a need to implement or design electronic components.

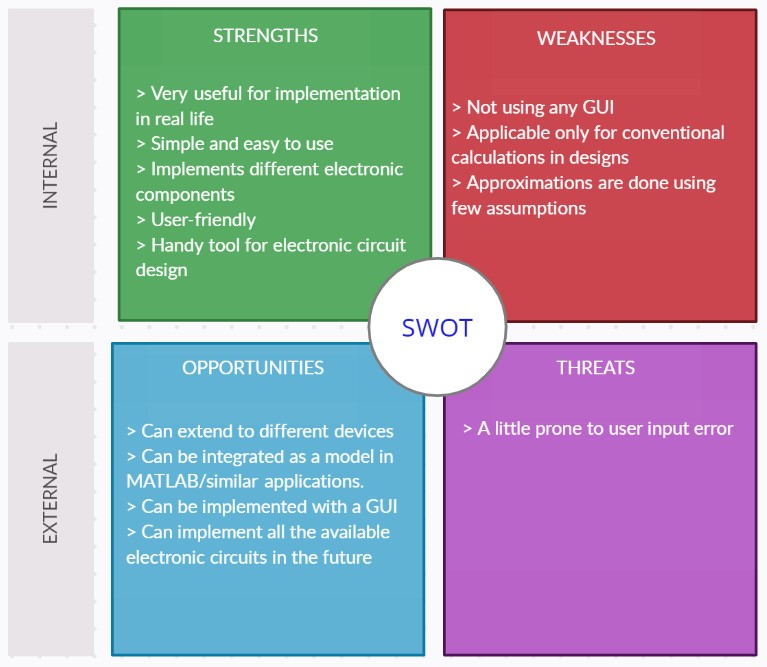
**Where:**

* This project can be used colleges/universities, labs or even for individual uses.

**How:**

* The user can follow the (add link for implementation). Then the user enters their requirements and the output will be displayed.

# SWOT Analysis

[](https://github.com/TanmayBhilkar/SDLC_AUG_TEAM_7/blob/main/1_Requirements/SWOT.jpg)





# Features

|  |  |  |
| --- | --- | --- |
| Sl. No. | Component | Description |
| 01 | Design of filters | Taking input like frequency and gain from the user and returning the circuit parameters for design |
| 02 | OP-AMP | Just to check Whether the given parameter forms the given Amplifier or Not. and Find out the output Voltage |
| 03 | Logic Gates | To implement different Logic gates like AND, OR, NOT, NAND, NOR, XOR, and XNOR Gates |
| 04 | Design of Oscillators | Taking input like oscillating frequency, oscillators amplitude gain and feedback fraction from the user and returning the circuit parameters for design of that oscillator circuit |
| 05 | Flip Flops | Implement different Flip Flops and give the next state to the user |
| 06 | Number system conversions | To implement conversion of numbers from one system to the other i.e binary,decimal,octal,hexadecimal |
| 07 | Rectifier | To design different rectifiers by talking input from the user for various parameters |
| 08 | OP-AMP\_Part-B | To check Wheteher the given parameter forms the given Amplifier or Not. and Find out the output Voltage |
| 09 | Adders and Subtractors | To implement addition and subtraction on binary numbers |
| 10 | Digital converters | To perform binary to gray code conversion and vice versa as well as BCD to excess 3 code and vice versa |

# High Level Requirements:

|  |  |  |
| --- | --- | --- |
| ID | Description | Status |
| HR01 | Application should implement various electronic component design | Implemented |
| HR02 | Application should take input from the user | Implemented |
| HR03 | Application should print output to the user | Implemented |
| HR04 | Application should work on both Linux and Windows | Implemented |
| HR05 | Application should follow clean code rules | Implemented |
| HR06 | Op-Amp Application Should Print Voltage to User | Implemented |
| HR07 | Adders must add binary inputs and display the output to user | Implemented |

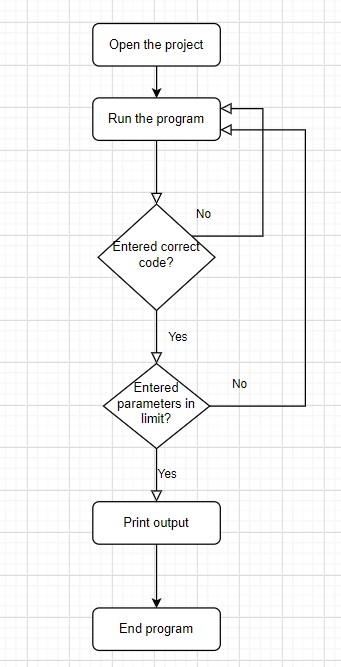
# Low level Requirements:

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | HR ID | Status |
| LR01 | Application should be able to run on the CLI of the user | HR02, HR03 | Implemented |
| LR02 | Application should use makefile option for different commands | HR04 | Implemented |
| LR03 | Application should comply with different checks like valgrind, cppcheck, build etc. | HR05 | Implemented |
| LR04 | Application to be used with a GUI to provide inputs and print outputs | HR02, HR03 | Implemented |
| LR1.1 | Application should consider all the different types of active filters available to design | HR01 | Implemented |
| LR2.1 | Implementation of different Logic Gates | HR01 | Implemented |
| LR2.2 | Implement test cases to check the implementation of Logic Gates | HR05 | Implemented |
| LR3.1 | Application should take the Flip Flop present states as input | HR02 | Implemented |
| LR3.2 | Application should print the next state as output | HR03 | Implemented |
| LR4.1 | Application should be able to give an output in specific number system, when user enters an input | HR012,HR03, HR04 | Implemented |
| LR5.1 | Application should be able to give an output of rectifier when user enters input | HR02, HR03 | Implemented |
| LR5.2 | Application should be able to calculate efficiency when user opts for it | HR02, HR03 | Implemented |
| LR5.3 | Application should be able to take input of Vm in the given range | HR02 | Implemented |
| LR6.0 | Application Should be Able to Take correct Op\_AMP | HR06 | Implemented |
| LR6.1 | Application Should Be Able to Give Output According to the Op-Amp | HR06 | Implemented |
| LR7.1 | Op-Amp\_Part-B Should Print output Voltage by taking inputs for input votages and Resistances | HR02, HR03 | Implemented |
| LR7.2 | Op-Amp\_Part-B should give the choice to user to choose various opamps available | HR01 | Implemented |
| LR8.1 | Oscillators: Application should consider the different types of oscillators to design using Op Amp | HR01 | Implemented |
| LR8.2 | Oscillators: Application takes oscillating frequency, oscillator amplitude gain, feedback fraction as inputs depending on the oscillator circuit selected | HR02 | Implemented |
| LR8.3 | Oscillators: Application should print the circuit design parameters of the selected oscillator circuit | HR03 | Implemented |
| LR9.1 | Adders, Application should add binary numbers and produce output | HR07 | Implemented |
| LR9.2 | Subtractors, Application should subtract binary numbers and produce output | HR07 | Implemented |
| LR10.1 | Application should be able to able to convert binary code to gray code and vice versa upon user input | HR02, HR03 | Implemented |
| LR10.2 | Application should be able to able to convert BCD to Excess 3 code and vice versa upon user input | HR02, HR03 | Implemented |
| LR10.2 | Application should be able to able to show equivalent capacitance in series and parallel form | HR02 | Implemented |
| LR10.2 | Application should be able to able to show output according to the input given | HR02, HR03 | Implemented |



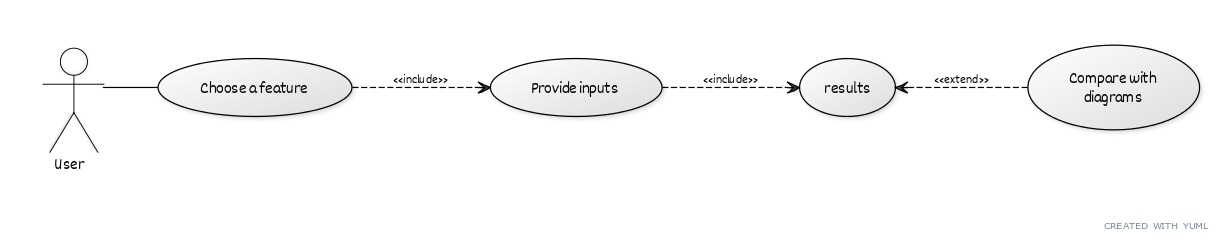
# Architecture

# High Level Diagram:

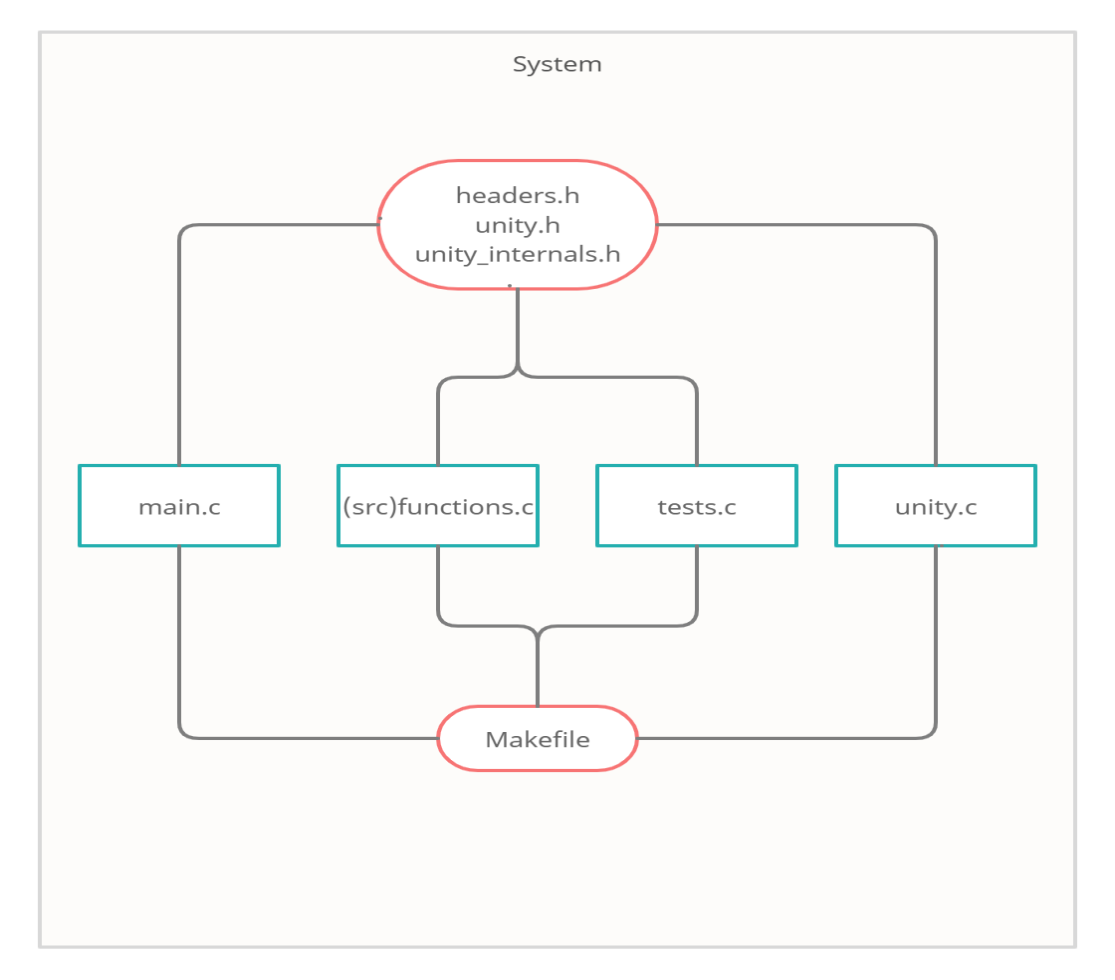




# Use Case Diagram:

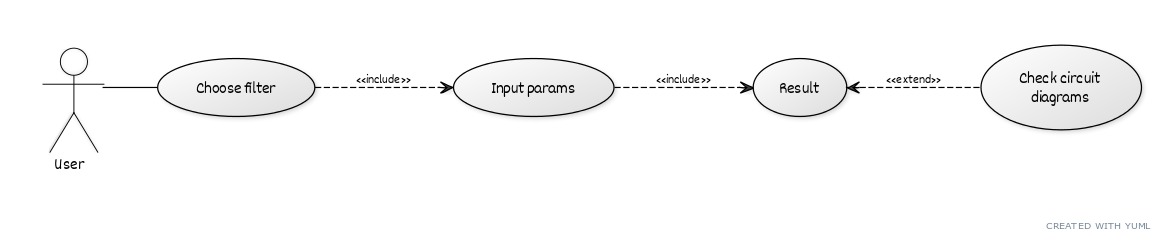


# Component Diagram:



# Low Level Diagram:

# Filter use case diagram:

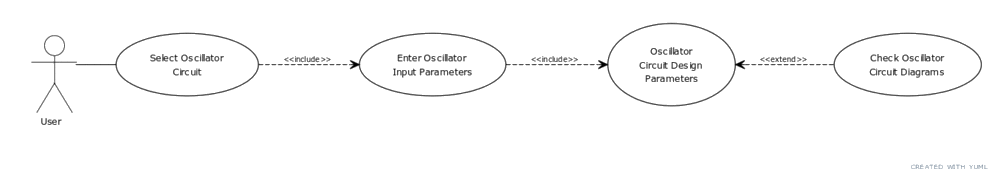


# FlowchartLogic gate Activity diagram:

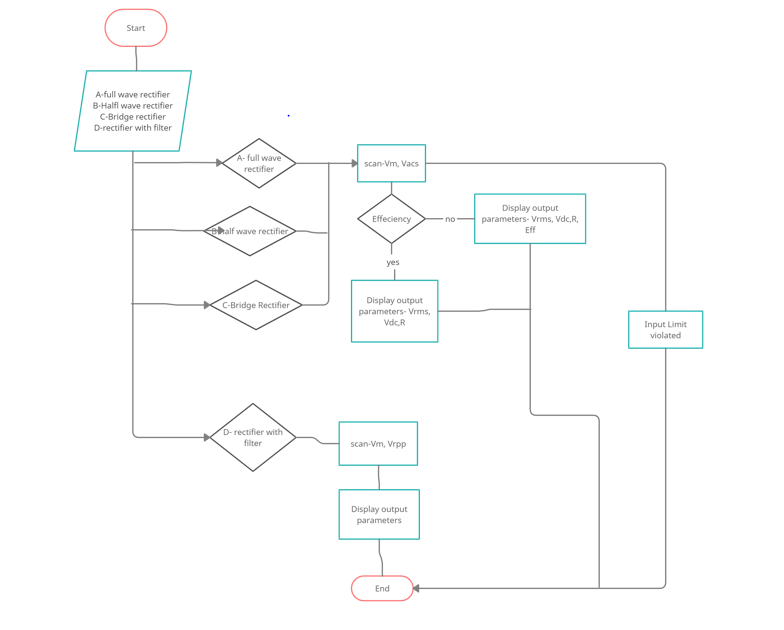




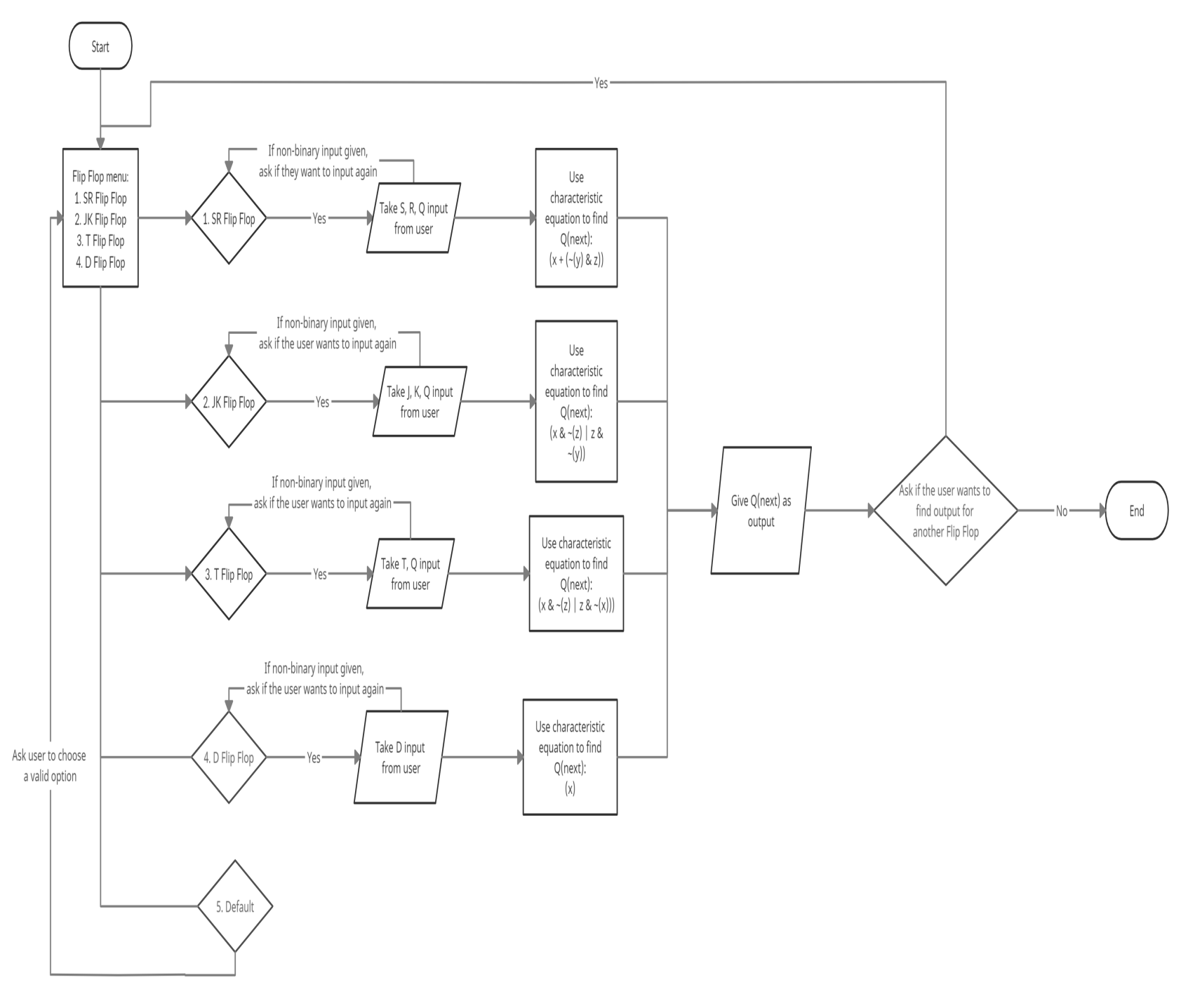
# Oscillators Use Case Diagram:



# Rectifier Activity diagram:

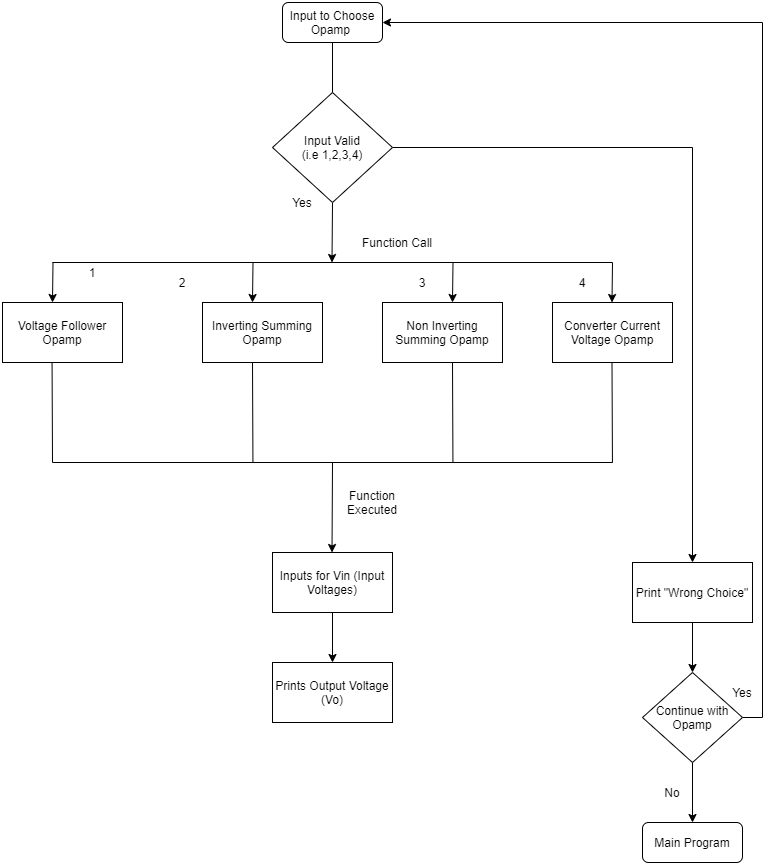


# Flip Flop Activity Diagram:



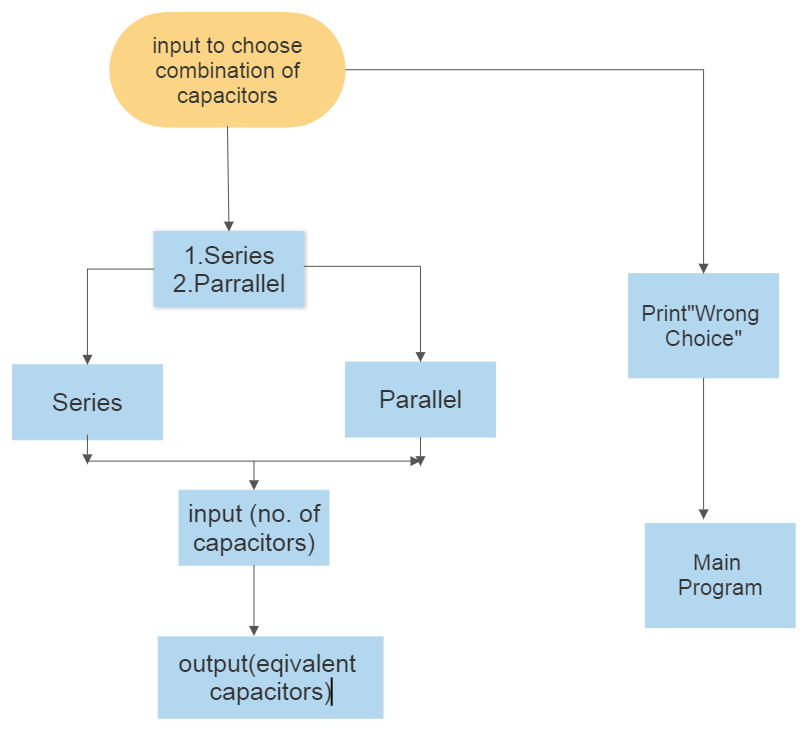


# Opamp Part –B Behavioral Diagram:



# 

# Capacitor Activity Diagram:







# Implementation

# To implement this project, follow the below steps:

1. Open the terminal in the location of the implementation folder.
2. Run the command "make" to build the project.
3. Now,run the command "make run" to choose from the features available in this project.
4. The parameters to design/implement and the results will be printed on the screen.
5. Open the images and videos folder in the project for the circuit diagrams if mentioned.

## Note:

1. To test, after step 1, run the command "make test".
2. To remove the built files and dirs, run the command "make clean".
3. If it shows "'make' is not recognized as an internal or external command" in your computer, try using "mingw32-make".





# TEST PLAN

# High level test plan:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Exp I/P** | **Exp O/P** | **Actual Out** | **Type Of Test** |
| HLT01 | Test various code quality checks | .c files, .out/.exe files, Implementation folder | TBD | TBD | Technical |
| HLT02 | Test various user inputs | User inputs | TBD | TBD | |  | | --- | | Scenario/  Technical | |  | |
| HLT03 | Test for various boundary conditions | TBD | TBD | Technical |  |
| HLT04 | Test for expected and actual design values | Input Parameters | Expected Values | Actual Values | Technical |
| HLT05 | Test for expected and actual output Voltage | Input Parameters | Expected Values | Actual Values | Technical |

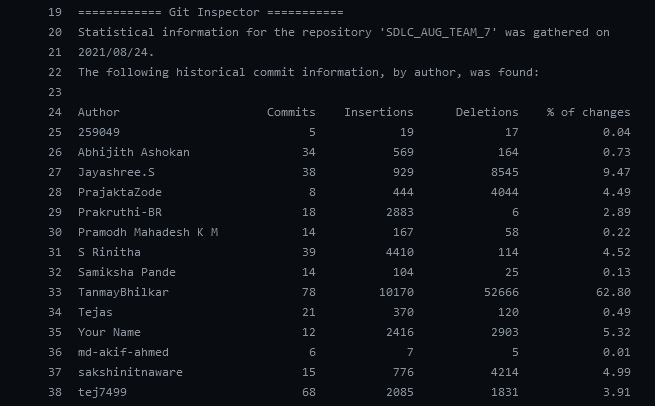
## 

# Low level test plan:

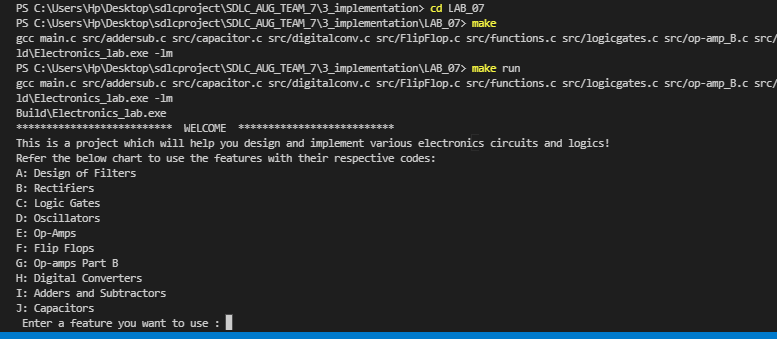
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **HLT ID** | **Description** | **Exp I/P** | **Exp O/P** | **Actual Out** | | Type Of Test | | --- | |  | |
| LLT01 | HLT01 | Static code analysis | Cppcheck on .c files | Passed | Passed | |  | | --- | | Technical | |  | |
| LLT02 | HLT01 | Dynamic code quality check | Valgrind on .out files | - | - | Technical |
| LLT03 | HLT02 | Check for wrong input variable for filter choosing | char with wrong code | Passed | Passed | Scenario/Technical |
| LLT04 | HLT03 | Upper and lower limit violation check for freq | Struct with limit violating values of freq | Passed | Passed | Scenario/Technical |
| LLT05 | HLT02 | Oscillators: Check for wrong input variable for oscillator choosing | char with wrong code | osc\_failure (enum) | osc\_failure (enum) | Scenario |
| LLT06 | HLT03 | Oscillators: Upper and lower limit violation check for oscillator frequency, amplitude gain and feedback fraction (one or more) | Structure with limit violating values of oscillator frequency, amplitude gain and feedback fraction (one or more) | osc\_limits\_error (enum) | osc\_limits\_error (enum) | Scenario/Technical |
| LLT07 | HLT04 | Oscillators: Circuit Design Parameters Check | Struct with the right input parameters for the given oscillator circuit | Structure containing the expected circuit design parametrs | Structure containing the calculated circuit design parameters | Technical |
| LLT08 | HLT05 | Operational Amplifier : check the various output of various Amplifier of voltage output parameter | Rf,Rin,Vin | Vout of different amplifiers | Vout of differnt Amplifier | Technical |
| LLT09 | HLT02 | Logic Gates: Prompt user to choose an option, and number of inputs and check against valid options | char and int | - | - | Scenario based |
| LLT10 | HLT03 | Logic Gates: Prompt for user inputs and check for a valid input | Binary values | Binary values | Binary values | Scenario based/Technical |
| LLT11 | HLT04 | Logic Gates: If invalid input, display error message | - | - | - | Scenario based/Technical |
| LLT12 | HLT02 | Flip Flops: Prompt user to chooce an option from a menu and check it's validity | char IN A-D or a-d | Respective case execution | Respective case execution | Scenario based |
| LLT13 | HLT03 | Flip Flops: Check for various user inputs (within limits) and it's respective output | Calculate Q(next) and return 0 | Calculates Q(next) and returns 0 | Calculates Q(next) and returns 0 | Scenario based |
| LLT14 | HLT04 | Flip Flops: Check for non-binary input | Print "Non-binary value" and return -1 | Prints "Non-binary value" and returns -1 | Prints "Non-binary value" and returns -1 | Scenario |
| LLT15 | HLT04 | Rectifier: Prompt user to enter Vm, Vac in the given range | float | Respective R,Efficiency | Respective R,Efficiency | Technical |
| LLT16 | HLT03 | Rectifier: Prompt user to enter Vm in given range (0>Vm && Vm<100) | float | Vm limits voilated | Vm limits voilated | Technical |
| LLT17 | HLT03 | Rectifier: Prompt user to enter non zero value for Vac | Float | Vac cant be zero | Vac cant be zero | Technical |
| LLT18 | HLT03 | Rectifier: Prompt user to enter Vac as Vdc/Vac should not be greater than 1 | Float | Vac Limits Voilated | Vac Limits Voilated | Technical |
| LLT19 | HLT02 | Capacitor: Prompt user to enter no. of capacitors for equivalent capacitance | Float | No. of capacitors | Equivalent capacitance | Technical |
| LLT20 | HLT03 | Capacitor: Prompt user to enter type of combination of capacitors | Int | type of combination | Series or parallel | Technical |

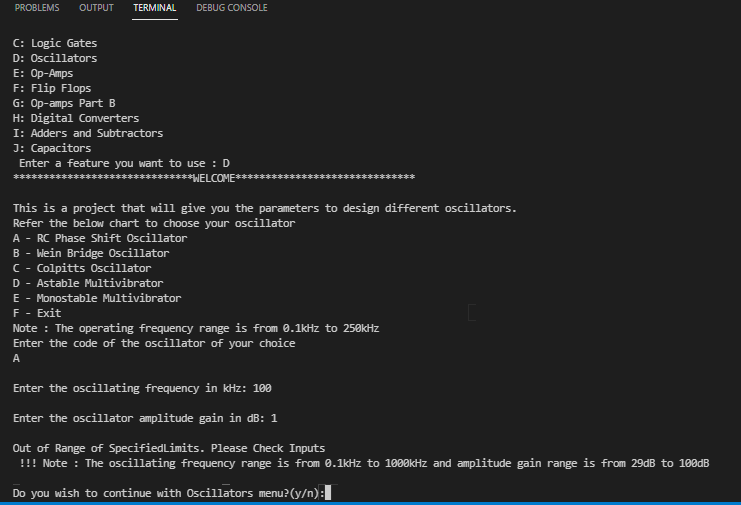
# Git Inspector

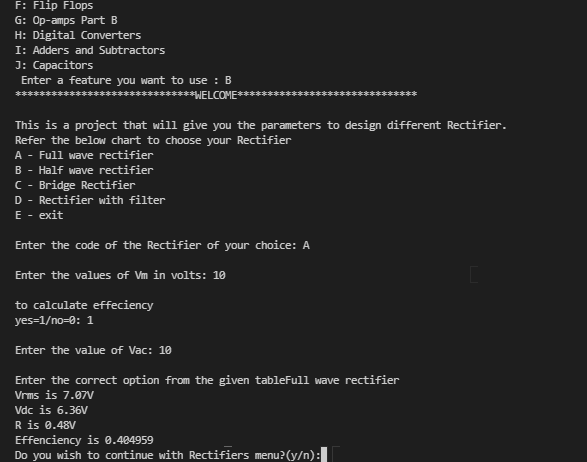
## GitInspector 2

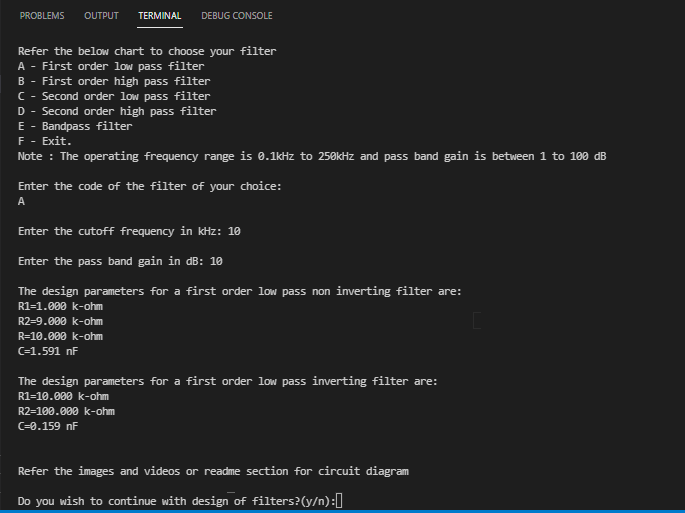


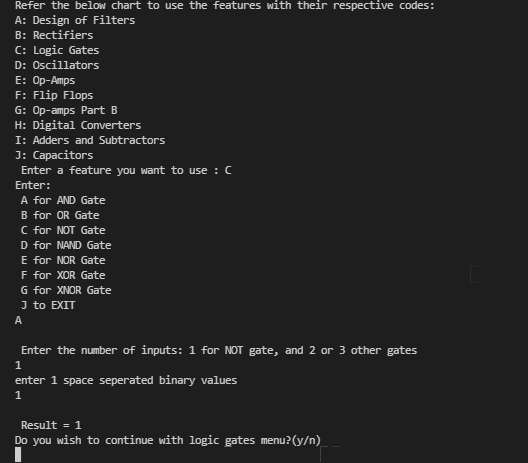
# Output:

****

****

****

****

****

# ****Daily Stand Up****

# 

# Date - 19/08/2021

* Came up with the idea for the project
* Laid out the high and low level requirements
* Completed the features, SWOT Analysis and 4Ws and 1H

# Date - 20/08/2021

* Discussed low and high level designs
* Went through the test plan and output
* Brifly beginning with implementation

# Date - 21/08/2021

* Working with implementation

# Date - 22/08/2021

* Implementation, solving issues and integration.

# Date - 23/08/2021

* Integrating all the various features
* Solving the issues raised
* Adding code quality badges

# Date - 24/08/2021

* Individual team review
* Group peer review















## 





