TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

A PROJECT PROPOSAL ON

DIGITAL LOGIC SUITE

CLI BASED TOOL

SUBMITTED BY:

NABARAJ BHANDARI (081BCT041)

NIKUNJ BHUSAL (081BCT043)

NIRDESH JOSHI (081BCT044)

SUPERVISED BY:

SENIOR FACULTY MEMBER, DAYA SAGAR BARAL

SUBMITTED TO:

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

INSTITUTE OF ENGINEERING, PULCHOWK CAMPUS

SUBMISSION DATE:

2082 Asar 17, Tuesday

Acknowledgment

We would like to express our sincere gratitude to our supervisor, **Mr. Daya Sagar Baral**, Faculty Member, Department of Electronics and Computer Engineering, IOE Pulchowk Campus, for his continuous guidance, invaluable suggestions, and encouragement throughout the development of this project.

This project, **Digital Logic Suite**, has been carried out as part of the subject **Object-Oriented Programming with C++**, and it would not have been possible without the support provided by our supervisor. His expertise and insights have greatly contributed to the successful design and implementation of this project.

We would also like to acknowledge the Department of Electronics and Computer Engineering at Pulchowk Campus for providing the resources and environment necessary for the completion of this work.

Table of Contents

1 Introduction

Digital logic design is fundamental to both electronics and computer engineering. **Digital Logic Suite** aims to provide a comprehensive, command-line-based toolkit for parsing and evaluating Boolean expressions, generating truth tables, minimizing logic using Karnaugh maps, simulating digital circuits, and visualizing logic structures.

The suite uses C++ object-oriented programming concepts and integrates with Graphviz for graphical representation.

2 Objectives

i. Develop a powerful CLI tool for digital logic analysis and simulation.

ii. Implement Boolean expression parsing and evaluation.

iii. Generate truth tables for arbitrary Boolean expressions.

iv. Perform Karnaugh map minimization for logic simplification.

v. Simulate digital circuits based on user-defined logic.

vi. Visualize logic circuits and truth tables using Graphviz.

vii. Demonstrate use of OOP principles using C++.

3 Existing Systems

Several existing tools provide digital logic design, analysis, and simulation functionalities. Some notable systems include:

i. **Logisim**: A user-friendly interface for building digital logic systems but lacks command-line capabilities.

ii. **Digital Works**: Supports circuit construction and basic simulation, but does not provide features for Boolean expression parsing or Karnaugh map minimization.

iii. **Logic Friday**: A Windows-based application for simplifying and analyzing Boolean equations, generating truth tables, and Karnaugh maps but, limited to a GUI environment.

Despite the availability of these tools, there is a lack of a lightweight, extensible, command-line-based tool designed specifically for learning, experimenting with, and automating digital logic analysis. This gap inspired us for the development of **Digital Logic Suite** project.

4 Proposed System

The proposed **Digital Logic Suite** is a modular, extensible CLI application written in C++. It supports parsing Boolean expressions, generating truth tables, Karnaugh map minimization, circuit simulation, and Graphviz-based visualization. The system is designed for ease of use, extensibility, and integration into academic workflows.

4.1 Description

The suite consists of several core modules:

- **CLI Module:** Handles user input, command parsing, and help documentation.
- **Expression Parser:** Parses and evaluates Boolean expressions using recursive descent or shunting yard algorithms.
- **Truth Table Generator:** Produces truth tables for given expressions.
- **Karnaugh Map Minimizer:** Simplifies Boolean expressions using K-map techniques.
- **Circuit Simulator:** Simulates logic circuits based on parsed expressions.
- **Graphviz Visualizer:** Generates .dot files and invokes Graphviz to visualize logic circuits and truth tables.

The suite is implemented using C++ OOP concepts, with each module encapsulated as a class or set of classes.

4.2 System Block Diagram

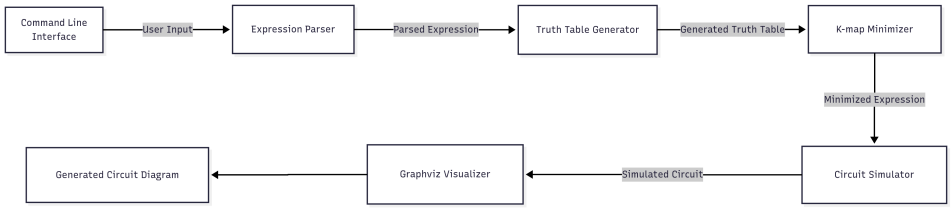

Figure 1: System Block Diagram: CLI → Expression Parser → Truth Table Generator → Circuit Simulator → Graphviz Visualizer

5 Methodology

The project will follow an iterative, modular development approach:

i. Requirements analysis and system design.

ii. Implementation of core modules (expression parser, truth table generator).

iii. Integration of minimization and simulation features.

iv. Development of CLI and user documentation.

v. Testing, debugging, and optimization.

vi. Integration with Graphviz for visualization.

6 Project Scope

The **Digital Logic Suite** is intended for academic and educational use.

It focuses on combinational logic analysis, minimization, and simulation.

Sequential logic and hardware synthesis are outside the current scope but may be considered for future extensions.

7 Project Schedule

Week	Planned Activities
Week 1	Requirements gathering, literature review, and system design.
Week 2	Implementation of CLI and Boolean expression parser.
Week 3	Truth table generation and Karnaugh map minimization modules.
Week 4	Circuit simulation and Graphviz visualization integration.
Week 5	Testing, documentation, and final report preparation.

Table 1: Project Schedule for Digital Logic Suite

8 Libraries and Tools

- **C++ Standard Template Library (STL):** Data structures and algorithms.
- **Graphviz:** Visualization of logic circuits and truth tables.
- **Visual Studio:** Development environment.
- **Git:** Version control and collaboration.