Digital Logic Suite Proposal

Academic Project Proposal

Department of Electronics and Communication Engineering

IOE, Pulchowk Campus

July 1, 2025

Acknowledgment

We would like to express our sincere gratitude to our project supervisor, faculty members, and peers for their invaluable guidance and support throughout the development of the Digital Logic Suite project. Their encouragement and constructive feedback have been instrumental in shaping the direction and scope of this proposal.

Contents

1 Introduction

Digital logic design is fundamental to computer engineering and electronics. The 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 leverages C++ object-oriented programming concepts and integrates with Graphviz for graphical representation.

2 Objectives

- Develop a robust CLI tool for digital logic analysis and simulation.
- Implement Boolean expression parsing and evaluation.
- Generate truth tables for arbitrary Boolean expressions.
- Perform Karnaugh map minimization for logic simplification.
- Simulate digital circuits based on user-defined logic.
- Visualize logic circuits and truth tables using Graphviz.
- Demonstrate effective use of C++ OOP principles.

3 Existing System

Several tools exist for digital logic simulation and analysis, such as Logisim and online Boolean calculators. However, most are GUI-based and lack extensibility or CLI integration. Few open-source solutions provide a unified command-line interface with advanced features like expression parsing, minimization, and circuit visualization tailored for educational and research purposes.

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.

5 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.

6 System Block Diagram

7 Methodology

The project will follow an iterative, modular development approach:

System Block Diagram Placeholder

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

1. Requirements analysis and system design.
2. Implementation of core modules (expression parser, truth table generator).
3. Integration of minimization and simulation features.
4. Development of CLI and user documentation.
5. Testing, debugging, and optimization.
6. Integration with Graphviz for visualization.

8 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.

9 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

10 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.

11 Header Files List

The following C++ header files will be used throughout the project:

```
#include <iostream>
#include <string>
#include <vector>
#include <map>
#include <algorithm>
#include <stack>
#include <queue>
#include <set>
#include <fstream>
#include <sstream>
```

Appendices

Class Diagrams

Sample Outputs