LC-3 Emulator: A Comprehensive Educational Platform

Development Proposal for the University of Auckland

Prepared by Jack Crump-Leys

jackcrumpleys@gmail.com

1. Executive Summary

This proposal presents a compelling case for the University of Auckland to continue development of our LC-3 emulator platform. The LC-3 architecture serves as a cornerstone for COMPSCI 210. Our emulator offers significant advantages over existing solutions through its textbook-accurate implementation, modern web-based interface, and comprehensive educational features.

The emulator is already operational in its current state, with a substantial foundation of features implemented. Continued development will focus on enhancing the educational value through improved visualization tools, expanded debugging capabilities, and integration with course materials.

2. Background

The Little Computer 3 (LC-3) is a simplified computer architecture designed for educational purposes. It provides students with a fundamental understanding of computer organization, assembly language programming, and low-level system operations. Despite its importance in computer science education, existing LC-3 emulators often present several limitations:

- Inconsistency with textbook specifications
- Poor user interface design
- Limited platform availability
- Lack of modern debugging tools
- Inadequate educational features

These limitations create barriers to learning and often require multiple different tools to achieve comprehensive understanding.

3. Competitive Analysis

Feature	Our Emulator[1]	Manitoba	LC3web	LC3Tutor	LC3UArch	PennSim	LC3Tools
Core Functionality & Conformity							
Textbook conformity	✓	✗	✗	✗	✗	✗	✓
Assembler	✓	✓	✓	✓	✗	✓	✓
Cycle-accurate simulation	✓	✗	✗	✗	✓	✗	✗
Object file support	✓	✗	✗	✗	✗	✓	✓
Debugging & Inspection							
Quality error messages	✓	✓	✓	✓	✗	✗	✓
Breakpoints	✓	✓	✓	✓	✗	✓	✓
Memory viewing	✓	✓	✓	✓	✓	✓	✓
Memory decompilation	✓	✓	✗	✗	✓	✓	✓
Step-by-step execution	✓	✓	✓	✓	✓	✓	✓
Register and memory editing	✓	✓	✓	✓	✗	✓	✓
Platform & User Experience							
Web platform	✓	✓	✓	✓	✗	✗	✗
Windows	✓	✗	✗	✗	✓	✓	✓
macOS	✓	✗	✗	✗	✓	✓	✓
Linux	✓	✗	✗	✗	✓	✓	✓
Syntax highlighting	✓	✓	✗	✗	✗	✗	✓
Documentation	✓	✓	✓	✓	✗	✗	✓
Console I/O	✓	✗	✓	✓	✗	✓	✓
Dark mode	✓	✓	✗	✗	✗	✗	✓

Table 1: Comparative analysis of LC-3 emulators

4. Assessment of Alternatives

- **Manitoba LC-3**: While offering an excellent web interface, it lacks cross-platform desktop support and has inconsistencies with the textbook implementation. Additionally, it doesn't support custom object files.

- **LC3web** and **LC3Tutor**: These platforms provide basic web functionality but lack critical features such as memory decompilation, syntax highlighting, and dark mode. They also contain implementation deviations from the textbook standards.

- **LC3UArch**: Primarily focuses on datapath visualization rather than providing a complete educational environment. It lacks many essential features like breakpoints, a run button, and console I/O. The datapath visualization feature is super useful and would be a great addition to our tool.

- **PennSim**: An outdated option with a clunky user interface that has fallen behind modern educational needs. It doesn't conform to the current textbook standard used at UoA. This is the current tool that the course uses.

- **LC3Tools**: While being the most comprehensive alternative and offering textbook conformity, it suffers from buggy input handling and lacks web platform support, limiting accessibility for students.

---

[1]Features either already implemented or trivial to implement.

5. Our Solution

Our LC-3 emulator represents a unique combination of technical accuracy, educational utility, and modern design principles:

5.1. Technical Implementation

Core Architecture

- Complete implementation of all LC-3 instructions
- Cycle-accurate simulation with visible micro-operations
- Privilege level support (User/Supervisor mode)
- Memory protection and exception handling
- Full conformity with the 3rd edition of "Introduction to Computer Systems" by Yale N. Patt and Sanjay J. Patel

Modern Development Stack

- Built with Rust for reliability and performance
- WASM compilation for cross-platform web support
- Native binaries for all major desktop platforms
- Modular design for easy extensibility

5.2. Educational Features

Debugging Capabilities

- Step-by-step execution at instruction or micro-operation level
- Integrated breakpoint system
- CPU state info for understanding processor cycles
- Detailed parsing errors for assembly programs

User Experience

- Provide the power of a desktop application on a web platform
- Modern, intuitive interface to provide an IDE
- Flexible pane system for customized workspace layouts
- Integrated help system with examples and documentation
- Syntax highlighting for assembly code
- Consistent experience across web and desktop platforms

6. Development Roadmap

The following are key areas for continued development:

6.1. Educational Enhancements

Memory visualization improvements

- Memory region highlighting (stack, heap, code, data)
- Memory access history tracking
- Value change indicators

Advanced debugging tools
- Conditional breakpoints
- Memory watchpoints
- Execution path recording and playback

Interactive learning features
- Guided tutorials on assembly concepts
- Challenge exercises with automatic verification
- Interactive datapath visualization

7. Benefits to the University of Auckland

Enhanced Learning Outcomes
- Improved student comprehension of computer architecture concepts
- Reduced learning curve through integrated tooling
- Consistent experience aligned with textbook material
- Accessible anywhere through web platform
- Reduced technical support requirements

Administrative Advantages
- Single platform for all course requirements
- Reduced need for multiple software tools
- Cross-platform support for all student devices

7.1. Long-term Value

Future-Proof Investment
- Modular architecture allows easy updates as curriculum evolves
- Framework extensible to other educational architectures
- Demonstrates university's commitment to quality educational tools
- Enhances appeal of the computer science program to prospective students
- Potential for future COMPSCI tooling based on this foundation

8. Conclusion

Our LC-3 emulator represents a significant advancement over existing solutions, offering a comprehensive, textbook-accurate implementation with modern features and cross-platform support. By funding its continued development, the University of Auckland can provide students with a superior educational tool that enhances learning outcomes while reducing technical barriers.

The platform unifies the best features from all existing alternatives while adding innovative capabilities that directly support the educational goals of COMPSCI 210. This investment will yield returns through improved student learning outcomes, reduced technical support needs, and enhanced reputation for the Computer Science department.