Alternative Implementations

This directory contains documentation for alternative implementations of the Maiko emulator.

Implementations

[Common Lisp Implementation](lisp-implementation.md)

Complete implementation of the Maiko emulator in Common Lisp (SBCL).

- **Status**: ✅ Complete (77/78 tasks, 98.7%)
- **Location**: `alternatives/lisp/`
- **Build System**: ASDF
- **Display Backend**: SDL3
- **Opcodes**: 189 of 256 implemented (73.8%)
- **Source Files**: 24 Lisp files
- **Test Files**: 11 test files

Key Features:

- Complete VM core with dispatch loop
- Memory management (storage, GC, virtual memory)
- Display subsystem (SDL3 backend)
- I/O subsystem (keyboard, mouse, filesystem)
- Sysout file loading with endianness handling
- Comprehensive error handling
- Platform-specific support (endianness, pathnames)

[C Implementation (Reference)](c-emulator-memory-loading-analysis.typ)

Reference implementation of the Maiko emulator in C.

- **Status**: ✅ Complete (Reference Implementation)
- **Location**: `maiko/src/`
- **Build System**: CMake / Make
- **Display Backend**: X11 / SDL2
- **Opcodes**: All 256 implemented
- **Source Files**: Complete C implementation

Key Documentation:

- [Memory Loading Analysis](c-emulator-memory-loading-analysis.typ) - FPtoVP table, page loading, byte-swapping
- [PC Calculation Logic](c-emulator-pc-calculation.typ) - Program counter calculation from frame data
- [PC Advancement Fix](c-emulator-pc-advancement-fix.typ) - Critical bug fix for PC synchronization (2026-01-12)
- [Byte-Swapping Logic](c-emulator-byte-swapping.typ) - FPtoVP and page content byte-swapping
- [Execution Byte Mismatch](c-emulator-execution-byte-mismatch.typ) - Investigation of execution vs loading bytes
- [Unified Logging Format](unified-logging-format.typ) - Execution log format for C/Zig comparison
- [Execution Comparison Results](execution-comparison-results.typ) - C vs Zig execution comparison results

Opcode Tracing (2025-01-27):

- [GVAR Opcode Tracing](c-emulator-address-xor-tracing.typ) - XOR addressing mechanism for BIGATOMS mode
- [UNBIND Opcode Tracing](c-emulator-unbind-tracing.typ) - Stack unwinding and variable unbinding
- [GETBASEPTR_N Opcode Tracing](c-emulator-getbaseptr-tracing.typ) - Base pointer access and memory reading
- [COPY Opcode Tracing](c-emulator-copy-tracing.typ) - Stack duplication mechanism
- [TJUMP1 Opcode Tracing](c-emulator-tjump1-tracing.typ) - Conditional jump mechanism (true jump)
- [CONST_1 Opcode Tracing](c-emulator-const1-tracing.typ) - Constant push mechanism (small positive integer 1)

Verified Logic (2025-01-27):

- ✅ FPtoVP mapping: File page 5178 ⟶ Virtual page 6204
- ✅ Address conversion: DLword offset ⟶ byte offset (multiply by 2)
- ✅ PC calculation: `PC = Lisp_world + (FX_FNHEADER * 2) + CURRENTFX->pc`
- ✅ Byte-swapping: `ntohl()` for FPtoVP, 32-bit swap for page content
- ⚠️ Execution byte mismatch: Needs investigation (doesn't affect core logic)

[Zig Implementation](zig-implementation.typ)
Implementation of the Maiko emulator in Zig programming language.

- **Status**: ⚠️ Execution Debugging Required - Critical discrepancies found
- **Location**: `zaiko/`
- **Build System**: Zig build system (`build.zig`)
- **Display Backend**: SDL2 (linked, integration complete)
- **Opcodes**: 100 of 256 implemented
- **Source Files**: Multiple Zig modules
- **Test Files**: Test suite structure

Key Features:

- Complete framework structure
- VM core framework (dispatch loop structure)
- Memory management framework (GC, storage, virtual memory)
- Display subsystem framework (SDL2 backend)
- I/O subsystem framework (keyboard, mouse, filesystem)
- Comprehensive opcode enumeration
- SDL2 display integration complete

Critical Issues (2025-12-22):

- ⚠️ Memory loading failure - wrong content at PC 0x307898
- ⚠️ FuncObj offset calculation wrong (+4687768 vs +104)
- ⚠️ PC progression wrong (always increments vs can stay same)
- ⚠️ Frame header reading wrong (0x780030 vs 0x307864)
- ⚠️ TOS values wrong (all zeros vs actual values)
- ⚠️ Execution stops early (30 lines vs 1000+)

See: Execution Debugging for detailed investigation and fixes

Implementation Status

Related Documentation

- Rewrite Specifications - Language-agnostic specifications
- Component Documentation - System architecture
- API Reference - Function signatures