65c02 Homebrew Computer

Joseph R. Freeston

June 10, 2025

1 Introduction

This document aims to document the verification of the boolean operations performed by the logic in my 65c02 homebrew computer design prior to laying out the PCB. In addition, this document will document the propagation delay calculations for timing verification.

For each major component, there will be a logical verification subsection, as well as a timing verification subsection.

2 Fault Detection and User/Kernel Modes

The access control hardware has two states: user mode and kernel mode. Any interrupt should initiate a switch to kernel mode by the time the interrupt service routine (ISR) begins execution. User mode must be manually initiated, and entered as execution returns from an ISR or begins execution of user-space and/or application code.

No faults should occur when operating under kernel mode. User mode shall be restricted by prohibiting the following actions:

- Executing a STP instruction
- Attempting access of any sort to high-memory

2.1 Logic Verification

Execution mode handling is done via a J-K flip-flop that stores a 1 if execution is happening in user mode. All other logic is based around this.

Variable	**Type**	**Description**
`XLEVEL_USER`	5V CMOS	In user mode.
`XLEVEL_KERNEL`	5V CMOS	In kernel mode.
`HIMEM`	5V CMOS	Accessing (for read or write) memory in the top 16k of address space.
`FAULTCLK`	5V CMOS	A phase-shifted `PHI2`
`UMINH`	5V CMOS	User-mode inhibit

Table 1: Variable Definitions for Execution Modes

The unsimplified logic:

3 High-memory Address Decode

4 Delayed Clocks

5 General-purpose Memory Bank Selection

6 SRAM Interface

7 DMA Interface

8 Character i/o Interface

9 Interrupts