Raspberry Pi Based Mainframe Simulator

Brent Seidel
Phoenix, AZ

September 16, 2021

Contents

Chapter 1

Introduction

A rather simplistic view is that this provides a way to get a Raspberry Pi to blink some LEDs and read some switches.

Chapter 2

Hardware

With some modifications, this hardware could be repurposed for a number of other applications that require lots of discrete I/O, such as burglar alarms or sprinkler controls.

2.1 Electronics

2.1.1 Processor

I had a Raspberry Pi 3 board laying around so I figured that I would use it.

2.1.2 Discrete I/O

Each of these boards has one or two MCP23017 16 bit I/O expander chips with I2C interface. Three pins are provided to configure the address, so the system can have up to 8 of these chips or 128 discrete I/O pins.

2.2 3D Printed Parts

2.2.1 Address/Data Panels

The address/data panels provide 8 switches and LEDs numbered consecutively from right to left. The starting number is selectable in the model. Each panel has two cables (one for LEDs and one for switches) that attach to connectors on the discrete I/O boards. Enough of these need to be provided to account for all the address and data lines in the simulated processor. A system would typically have two or four of these panels for 16 or 32 bit systems. Note that there are many cases where the number of address lines does not match the number of data lines. In these cases panels would need to be provided for whichever number is greater.

2.2.2 Control Panel

At this point, without an actual CPU simulation, this is mostly a way to organize what needs to be in a relatively generic control panel. The following functions are needed:

- Read from a memory location.
- Write to a memory location.
- Stop/pause processor execution.
- Start processor execution at a specified address.
- Continue processor execution.
- Reset the processor.

To Specify an Address

Note that if the word size is greater than 8 bits, some of the lower order address bits may be ignored.

- Toggle the desired address using the address/data switches.
- Move the "Addr"/"Data" switch to the "Addr" position.
- Move the "Dep" (deposit) switch to the "Dep" position.
- Once the LED for the "Dep" switch illuminates, move the switch to the unlabeled position.

To Read Memory

- Specify the starting address as described above.
- Move the "Addr"/"Data" switch to the "Data" position.
- Move the "Exam" switch to the "Exam" position.
- The contents of memory at the specified address will be displayed in the Address/Data LEDs.
- The "Addr"/"Data" switch can be used to toggle between display of the address and the data.
- Move the "Exam" switch to the unlabeled position. This increments the internal address pointer so that by toggling this switch a block of memory can be modified.

To Write Memory

- Specify the starting address as described above.
- Move the "Addr"/"Data" switch to the "Data" position.
- Toggle the desired data using the address/data switches.
- Move the "Dep" switch to the "Dep" position.
- The contents of memory at the specified address will be updated and displayed in the Address/Data LEDs.

- The “Addr”/“Data” switch can be used to toggle between display of the address and the data.
- Move the “Dep” switch to the unlabeled position. This increments the internal address pointer so that by toggling this switch a block of memory can be examined.

2.3 Other Hardware

Chapter 3

Software

3.1 Overview

3.2 Simulation

3.3 Web Server