Functional Specification

Year: 2025 Semester: Spring Team: 15 Project: αCassiopeiae 8800

Creation Date: January 22, 2025 Last Modified: January 25, 2025

Member1: Brian Lu Email: lu964@purdue.edu

Member2: Seth Deegan Email: sdeegan@purdue.edu

Member3: Caleb Shinkle Email: cshinkle@purdue.edu

Member4: Nathan Huang Email: huan1811@purdue.edu

1.0 Functional Description

The αCassiopeiae microcomputer acts as both a personal computer with a general microprocessor and provides expansion capabilities for different input and outputs. This device in its minimally complete form will allow the user to:

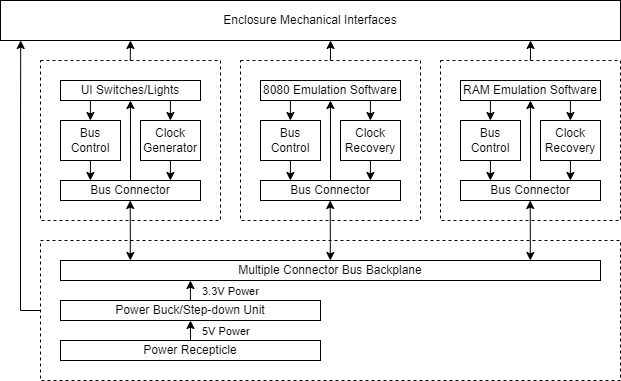
1. Utilize the front panel to write, read, and execute Intel 8080 instructions in memory.
2. Add expansion cards to allow the processor to read and write data over UART, to a disk, etc.

The front panel will contain 8 status lights to indicate the CPU’s operation, 16 address bit lights to indicate the address of the memory currently selected, 8 data bit lights to show the value at that point in memory, and 4 other CPU indicator lights. There are 16 switches for manipulating the data and address lines, and 6 switches for turning on, resetting, running, etc. the computer.

There will be at least 3 card slots using a DDR4 dev that users can slot expansion cards into.

Users can utilize the front panel to write programs for the computer but will typically use it to write a bootstrapper for an operating system the CPU can run. The lights can also be used as a debugging and step-through tool for programs.

A top-level diagram for the miniaturized Altair can be found in figure 1, below.

**

*Figure 1: Top Level Diagram of all systems*

2.0 Theory of Operation

The device is a multi-board implementation of the Altair 8800. This design features a custom 8080-bus featuring data, addressing, bus control, and front panel signals at 3.3V logic levels. A “Front Panel” board provides a reverse-mounted LED display of either user-controlled or bus-controlled signals (depending on the state of slide switches on the board), interfaces with the bus, and provides clock control. “CPU” and “RAM” boards are connected to this common bus and are driven by the clock signal to execute instructions, load memory, store memory, and interface with other peripherals (potentially) on the bus.

For example, as a stretch goal, a “UART” board may also interface with the bus using the same signals. This board would be mapped (possibly user configurable) to the I/O address space of the 8080. In this manner, it would work similarly to other UART boards of the 8800’s era.

Because many modern microprocessors do not have native parallel bus support, and even less so custom parallel bus interface support, our design relies on the ability for Programmable I/O interfaces. Raspberry Pi RPxxxx series of microcontrollers feature custom Programmable I/O (PIO) units that can be programmed to both drive and receive clocks on both serial and parallel interfaces at high speeds, all of which can be user defined. This project seeks to leverage PIO functionality to implement this bus interface for all boards: front panel, CPU, RAM, UART, etc.

A diagram of a machine

Description automatically generated

The PIO interface allows for the rapid transmission and control of all required parallel bus signals, including the 16 addressing bits, 8 data bits, bus control, and clock signals. Depending on the specific board, the TX/RX directions are different.

The specific choice of the RP2350B microprocessor and package is justified by the fact that it is the RPxxxx chip with the highest number of exposed GPIO pins, 48 (forty-eight), with all of them being connected to each of the 3 PIO blocks (pictured above).

While the front panel and backplane boards will have unique designs, it is the intention to have as much board design commonality as possible on the edge board connected PCBs. Therefore, what differentiates the CPU, RAM, and UART boards is mostly a matter of loaded software and populated components.

* The software loaded on the “CPU” card receives the clock signal to advance its execution and coordinates the bus control signals to determine whether it controls the address/data lines for read/write.
* The software loaded on the “RAM” card receives clock and bus control signals to both read and write memory, as dictated by the CPU/Front Panel.
* The software on the UART listens to I/O port signals to determine what data to read and write to/from the CPU.

A high performance 8080 emulator must be present on the CPU board software in order to keep up with the target clock rate. Tentatively, this emulator will be written in the Rust programming language. Because of the usage of Rust, the rp-rs framework for programming RPxxxx series boards is a tentative choice for the full selection of software running on these boards.

3.0 Expected Usage Case

* Stationary setting, indoors with no environmental hazards such as dust, electromagnetic interference, and humidity.
* One user at a time, users are likely hobbyists/computer enthusiasts who are knowledgeable about old school computers:
  + Adults with a formidable background in computers
  + Vintage computer collectors
  + Computer hobbyist clubs
* Likely individuals older than 40-50, because of the Altair 8800’s 1974 release date.
* Classroom setting for learning purposes.
* Can be used as a dev board for the RP2350.

4.0 Design Constraints

The design of this system has these major design constraints:

* The size of the entire device itself: The device should be as small as reasonably possible, portability, affordability, and “cuteness factor” can be maximized.
* The device should be able to support multiple boards. Part of the original hobbyist factor of the original 8800 was its expandability with additional cards using what became the S-100 bus. Being able to install and remove cards should be part of this device’s user experience.
* The device should appear to resemble in general shape and visual design of the original Altair 8800 as much as possible. Consequentially, the device’s front panel should follow the appearance of the 8800’s front panel, with lights and switches in approximately the same location, and a silkscreen design that emulates the original silkscreen; momentary actions can be emulated with tactile switches, while double throw actions can be emulated with slide switches.

4.1 Computational Constraints

The original Altair 8800 ran at a speed of 2 MHz. The RP2350 can run at a speed of 150 MHz. [1] While there isn’t an equivalent between modern Cortex cores and the Intel 8080, especially with modern IPCs, we do need to achieve a rough ratio of 75 instructions per 1 8080 instructions to emulate the 8080 at-speed. This will be able to run all the programs the original Altair ran, which is the goal of this project.

We need at least 64 kilobytes of memory so the emulated CPU can utilize its full address space. The RP2350 has 520 kb of on-chip SRAM itself, however we would like to maintain the functionality of the original Altair so we will have a separate memory card that will have an RP2350 itself on it to act as memory and meet this constraint. [1]

4.2 Electronics Constraints

The project plans to use The RP2xxx series of MCU’s PIO (Programmable I/O) functionality to implement parallel bus functionality (including 8080, bus control, and front panel signals). Most devices will listen to the clock signals, while one device (the front panel) will drive the clock. The parallel bus interface must be implemented in the PIO-specific state machine language, and run at a target speed of 2 MHz.

The display LEDs on the front panel, due to current draw, may require dedicated constant current power supplies, that are each switched based on their various signals (data, address, clock, etc.).

The layout of signals on each edge board connector, and the routing between connectors, should be done in such a way that there is little to no interference between signals. Depending on the choice of connector used to interface the front panel, this may also have to be done separately for the front panel connector choice.

4.3 Thermal/Power Constraints

Because nearly all the components on the device are solid state and have relatively low power consumption, we are targeting both 3.3V signaling and power across the device. The device should be able to receive power from a 5V source (notionally barrel plug or passive USB-C connector), step-down and regulate the voltage to the 3.3V levels necessary to power the RP2350 and other electronic components.

The old Altair required both +8V, +16V, and other voltage rails, running around 8 A of current. This led to around +192W power consumption, nearing 200W on an upper limit. Our compact Altair will run on a much smaller and lower voltage microcontrollers, each requiring only 500 mA at 3.3V. For a minimum setup of 3 microcontrollers to account for each board, the computational components of the board should consume about 4.95 Watts at a maximum. The front panel consisted of 36 LEDs, of which the ones we selected require 150 mA at 3.3V, with a combined maximum power consumption of 17.82 W. Combined, this is a much lower power consumption compared to the Altair, but is still something that should be taken into account when designing power supplies.

The operating temperature should match the temperature of the RP2350 chip, as that is likely the most volatile hardware placed in our project. This ranges to an “Operating temperature of -20° C to 70° C, but preferably around room temperature to ensure proper operation of the LEDs, and to fit the constraint of our Altair being a hobbyist PC.

4.4 Mechanical Constraints

The design should use a two-part aluminum profile extrusion enclosure. This enables use to use the Front Panel PCB structurally as the entire front panel of the device. The “ribs” on the bottom shell of the extruded enclosure can hold the backplane board. The spacing between the aluminum ribs dictates the thickness of the backplane board. The screw spacing of the profiles of the enclosure dictate the exact dimensions of our front panel board, and the sizing of the two rear panel, which are split such that the top shell can still be slid out the back for easy user expansion, once the top two front panel screws are removed.

A diagram of a shell

Description automatically generated

Add-on boards should be sized for the backplane connector choice, which is notionally DDR4 DIMM. These board should be short enough to have clearance below the top shell enclosure, and allow for additional clearance at the rear to allow for screw-on connectors to be attached to the rear plates.

The electrical connection between the front panel and the backplane may need to have some level of compliance, as there is no direct method of securing the backplane onto the enclosure on the axis normal to the front panel. Because of this, we are tentatively considering using a set of flat flex connectors (FFCs) to connect the backplane and front panel.

4.5 Economic Constraints

Because the intention of this project is to be competitive against other Altair 8800 clones, the project should try to be cost competitive against them (ideally less). Altair 8800 clones include the Altair 8800 mini which sells for €160.00 ($168) [2] and the Altair-Duino Pro Emulator Kit which sells for $475.95 [3].

5.0 Sources Cited:

# References

|  |  |
| --- | --- |
| [1] | Raspberry Pi Ltd, "RP 2350 Datasheet," 16 October 2024. [Online]. Available: https://datasheets.raspberrypi.com/rp2350/rp2350-datasheet.pdf. [Accessed 25 January 2025]. |
| [2] | "Altair 8800 Mini Shop," [Online]. Available: https://altairmini.com/store/product/altair-8800-mini. [Accessed 25 January 2025]. |
| [3] | "Altair-Duino Pro Emulator Kit," Adwater & Stir, [Online]. Available: https://adwaterandstir.com/product/altair-8800-emulator-kit/?srsltid=AfmBOoqMc88Nnfw\_rYbM99CBi4EJqSNDStYxRwJoKISQSL5VbLILhQVq. [Accessed 1 January 2025]. |
| [4] | "Altair 8800 Theory of Operations and Schematics Manual," MITS, Online, 1975. |

Raspberry Pi Foundation, *Raspberry Pi Pico 2 Datasheet*, 2021. [Online]. Available: <https://datasheets.raspberrypi.com/pico/pico-2-datasheet.pdf>. [Accessed: Jan. 25, 2025].