Bryce Hawes Microcontroller Project - 2025

Effective Datasheet:

The computer follows Von Neumann's architecture and uses 2 registers, A and D. The ALU can perform 4 operations, NOT, AND, OR, and ADD. Further, it can switch the order of operations and replace one variable with zero. It works in a 5V computing environment.

The computer is made of the following parts:

Part	Datasheet
MM 74C157N (2:1 * 4 mux, will hold 4 bits, 2 inputs)	https://www.alldatasheet.com/datasheet-pdf/download/53707/FAIRCHILD/MM74C157N.html
M5M4416P-15 606119 (4 bit by 16348 address RAM)	https://www.alldatasheet.com/datasheet-pdf/view/168996/MITSUBISHI/M5M4416P-15.html
P8284A 9447FEB V2 (clock)	https://rocelec.widen.net/view/pdf/nbqx6y95rv/INTLS03360-1.pdf?t.download=true&u=5oefqw
hn4827128G-25 (8 bit by 16348 words electronically erasable PROM)	https://datasheet.octopart.com/HN4827128G-25-Hitachi-datasheet-129162366.pdf
2716CQD8130 (8 bit by 8k word uv eprom)	https://www.alldatasheet.com/datasheet-pdf/pdf/157888/INTEL/2716.html

You may notice a few restrictions:

Firstly, the components were, for the most part, very outdated and limited to 4 bits. The workaround to this was to take three apiece, and somewhat force them to be 12 bits, an arbitrary amount I figured would provide a nice challenge. This meant that I had, for example, 3 DRAM chips, one dedicated to the first four least significant bits, then another dedicated to the next four least significant bits, and finally the four most significant bits.

RAM

Each chip uses the same clock, address information, RAS, CAS, enable, and is written with their respective data in/out pins. This allows an effective 16348-word by 12-bit dynamic RAM.

Logic

While I easily found a logic unit, it was again restricted by 4 bit computing. I decided to expand and design my own 12 bit logic and arithmetic unit. It worked as a series of 12 bit selectors (each made from three 4-bit selectors), allowing the user to choose between NOT, AND, OR, and ADD. Further, another set of selectors allows the user to switch the two registers when performing the operation (useful for NOT), and switch the D register to a logical 0 (useful for AND(A,0)). The logic unit also uses some condition chips to determine whether the result of the operation is zero (EQ), negative (LT), or positive (GT).

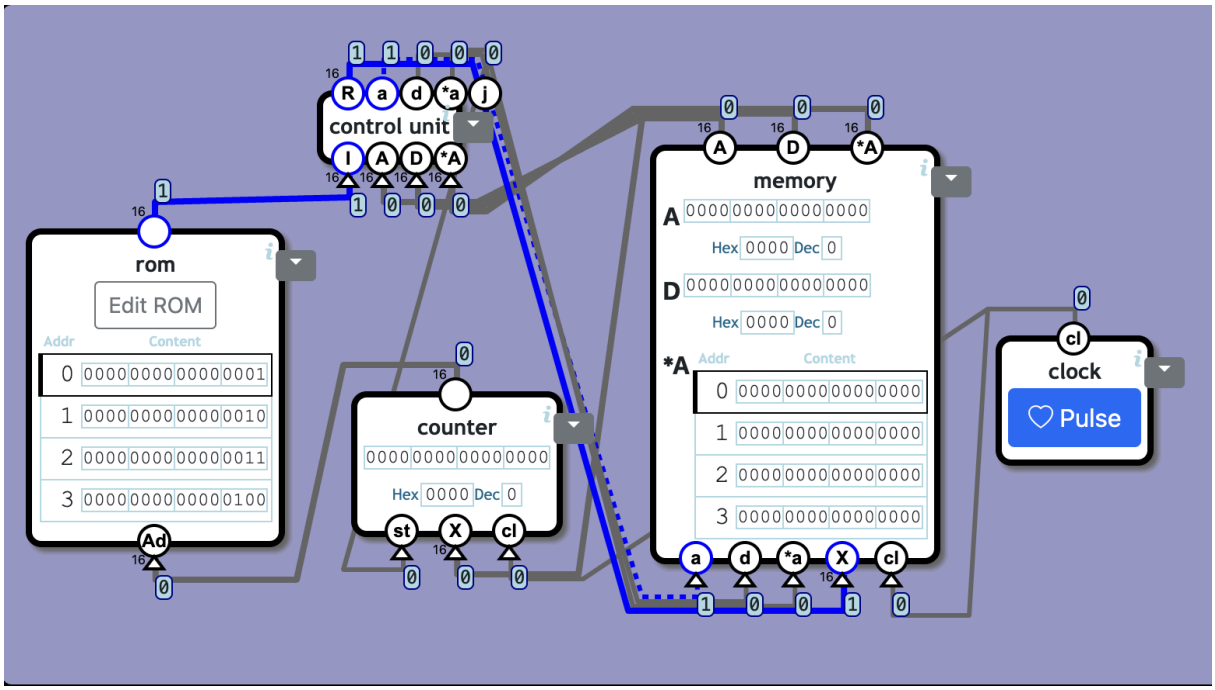
Bryce Hawes Microcontroller Project - 2025

Control

The control unit uses the logic and arithmetic unit to perform an operation if called by the instruction register Ci bit (11). If called, the current instruction, will be executed and a result will be passed, as well as a signal in the respective (a, d, *a) pins that direct where the result will be stored. If not called, the A register is selected as the result, and the exact binary number in the instruction register will be passed to the A register. Further, if the LT, GT, or EQ condition bits are 1, the j pin will be 1 if the operation result matches the respective condition, signifying the computer to jump to the address in the PROM held by A.

Architecture:

Credit: https://nandgame.com/



Example Code:

Goal: Put value of 7 at address 7.

```
//Set A to 7 (A = 7)
000000000101
//Set D to A (D = OR(A, 0))
101111010000
//Set value at address A to D (A* = OR(D, 0))
101011001000
```