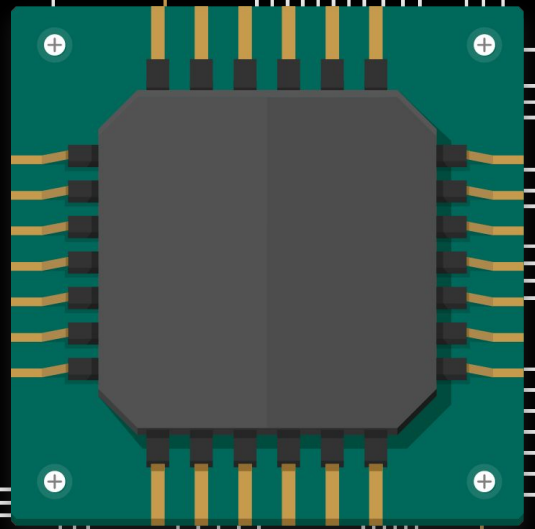
Breadboard Computer Project

7 August 2024

Members: Dilan Gandhi and Rithvik Madiraju

Project Advisors: Dr. Richard Howard and Dr. Richard Martin



Team Introduction

Dilan Gandhi ‘25 - Rising Senior at Manalapan High School

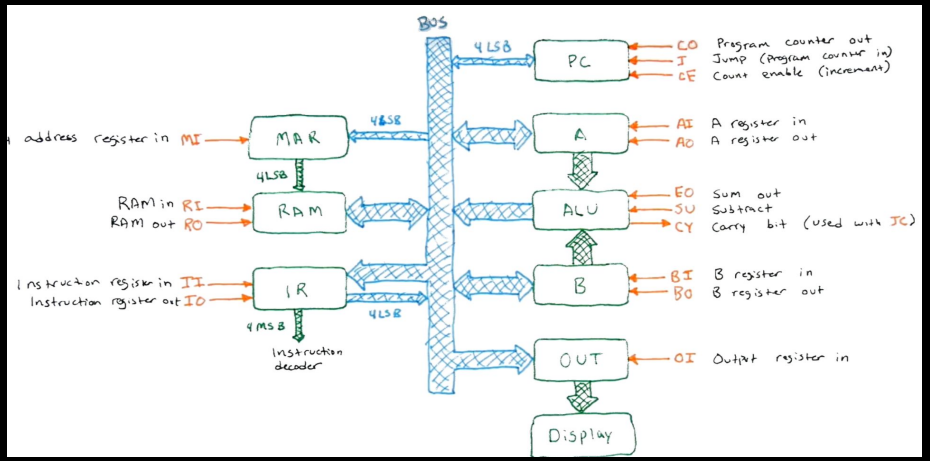
- Manalapan, New Jersey

Rithvik Madiraju ‘26 - Rising Junior at Francis Parker High School

- San Diego, California

Project Objective

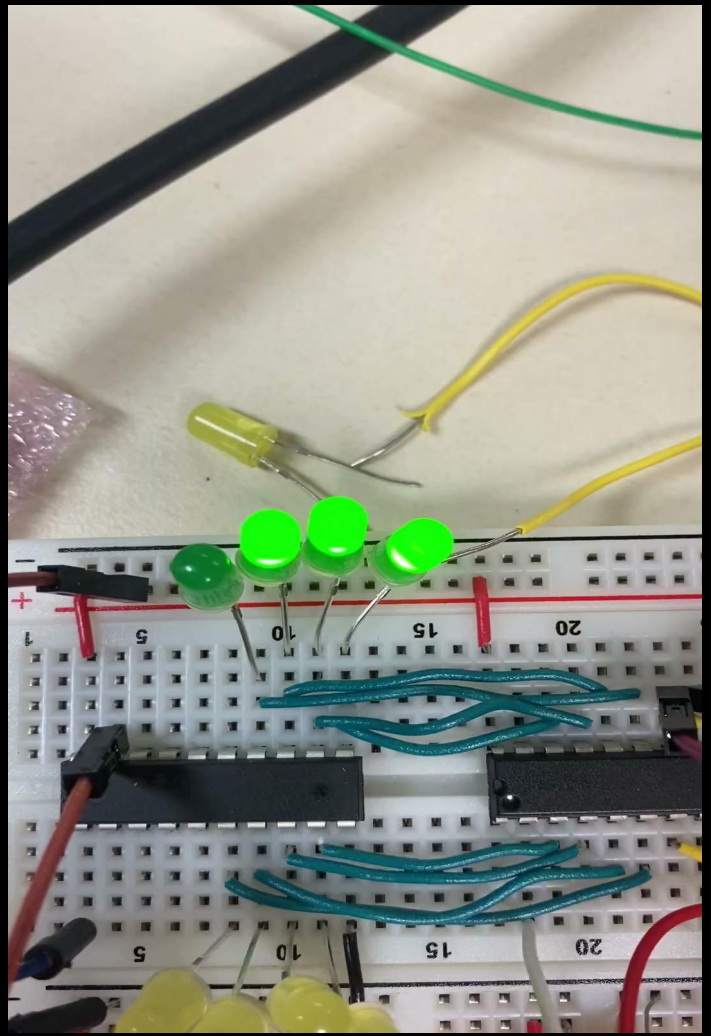
- Building a 8-bit computer from scratch on breadboards
- Learn and teach how to build a computer using different components
- To document the step-by-step process
  - Using schematics, notes, and images
  - Aim to help others replicate the creation



Terminology

Binary

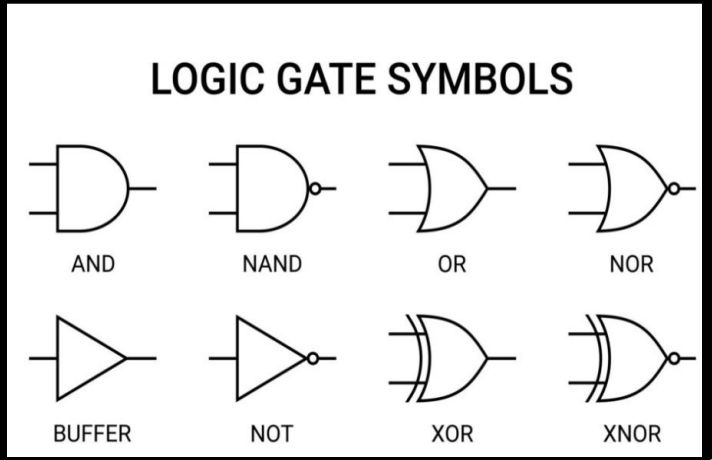
- Number system that uses only two digits, 0 and 1, to represent values



Terminology

Logic Gates

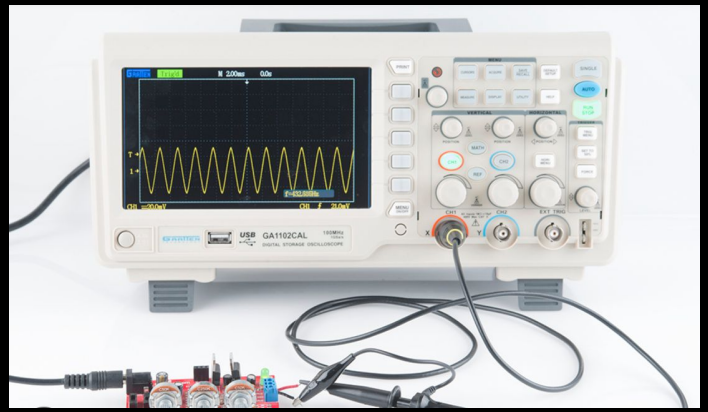
- a device performs a true or false operation binary inputs and then outputs a single a binary



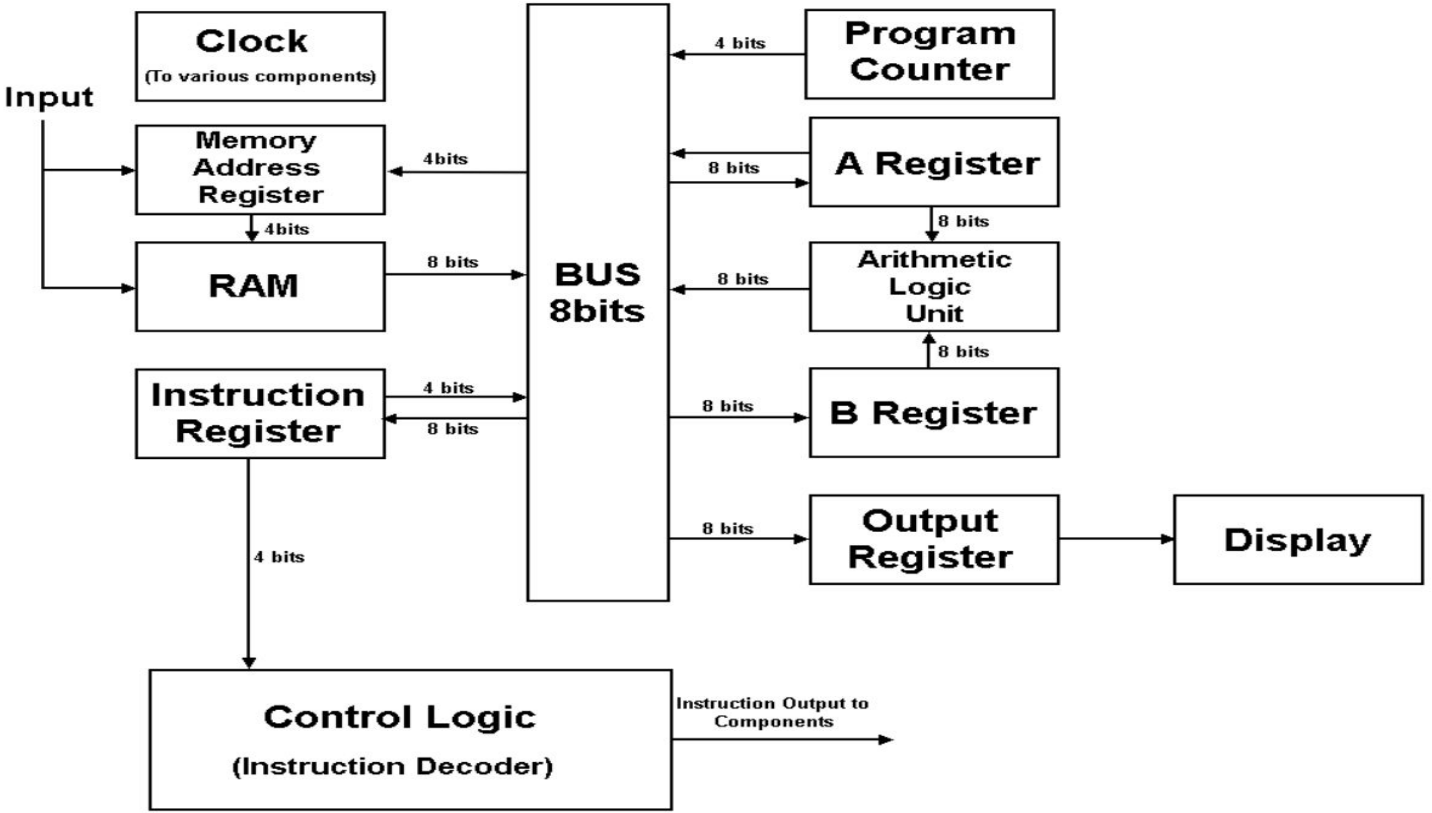
Terminology

Oscilloscope

- a voltage measuring device that shows the waveform of the voltage signal on its screen

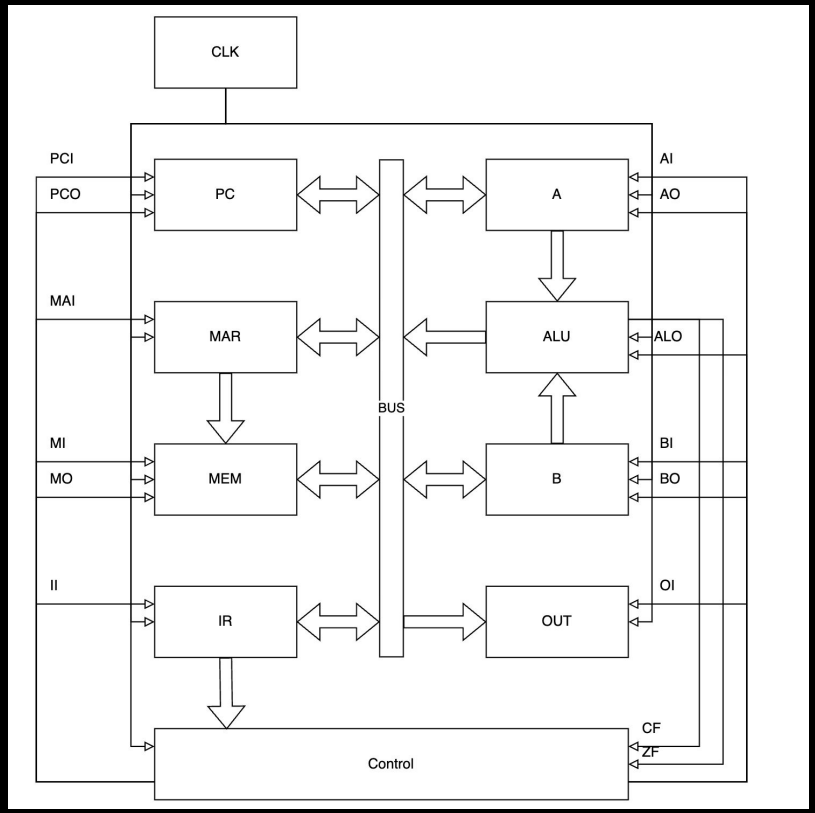


Architecture



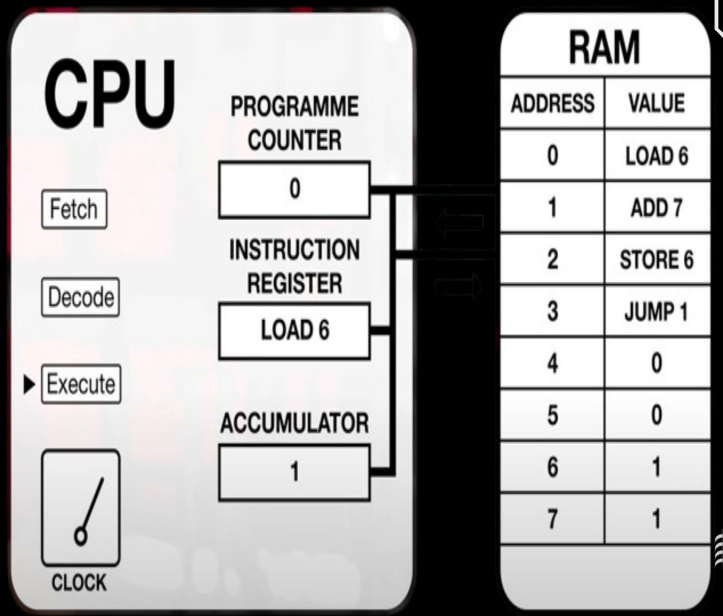
Components

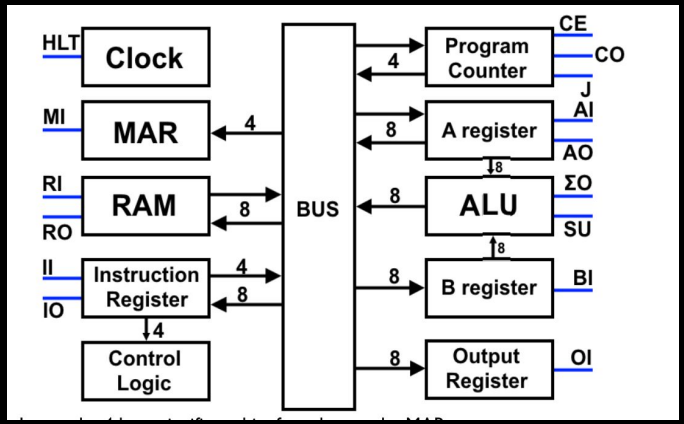
- 2 8-Bit General Purpose Registers
- 16 Bytes of RAM
- 8-Bit Program Counter
- 8-Bit Instruction Register
- ALU Based on the Chip 74LS181
- Control Unit with 1 EEPROMs Decimal Display (7-Segment Display)
- Clock Module
- 8-bit Bus System



Structure of CPU

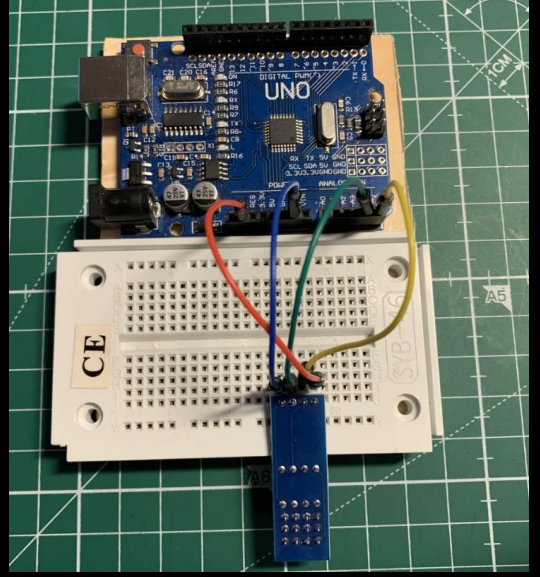
- **CPU fetch-decode-execute cycle**
  - Processes instructions stored in the RAM
- Control Signals determine the specific operation the ALU needs to perform





Software (EEPROM)

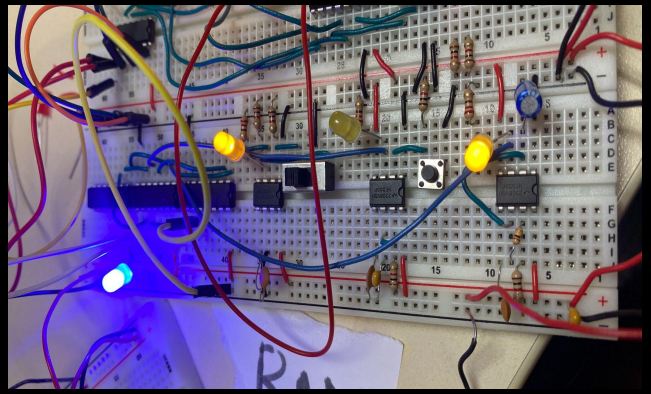
- Used for storing programs and data of the computer's instructions
- **Arduino Integration**
  - Writing data to the EEPROM.
- **Practical Application**
  - Simple program to blink an LED.

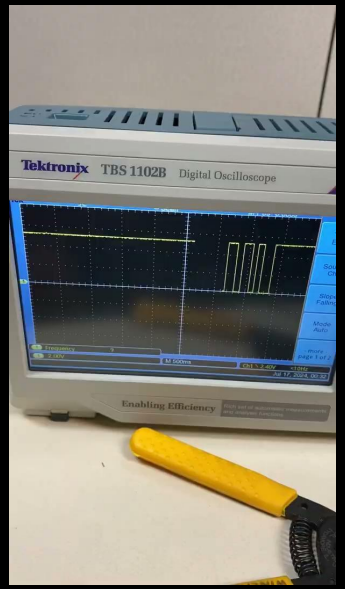


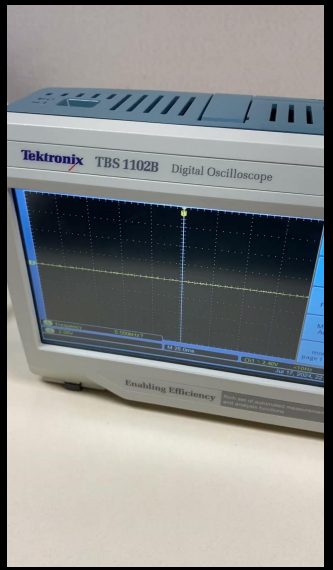

```
#include <EEPROM.h>

int address = 0;


void setup() {
  // put your setup code here,
  Serial.begin(9600);
  Serial.println("bois");
```

Clock Module

- Signal that keeps the control unit moving,
- **3 Logic Gates**
  - NOT Gate: Inverts input signal
  - NAND Gate: Controls clock signal
  - AND Gate: Combines signals
- **Testing**
  - RAM, ALU, and Program Counter

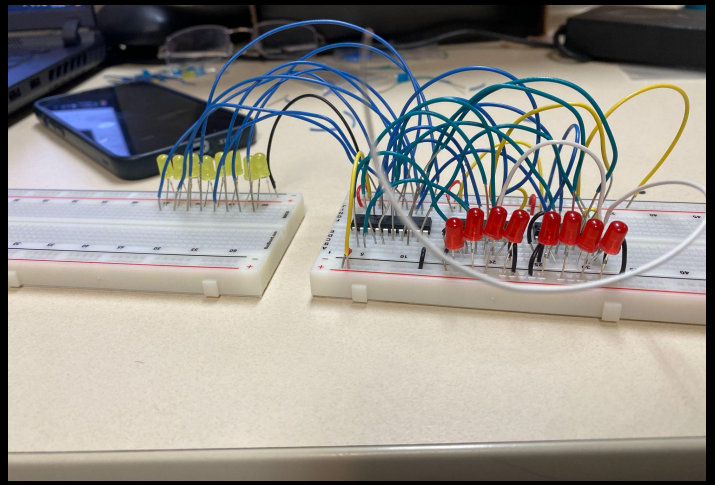


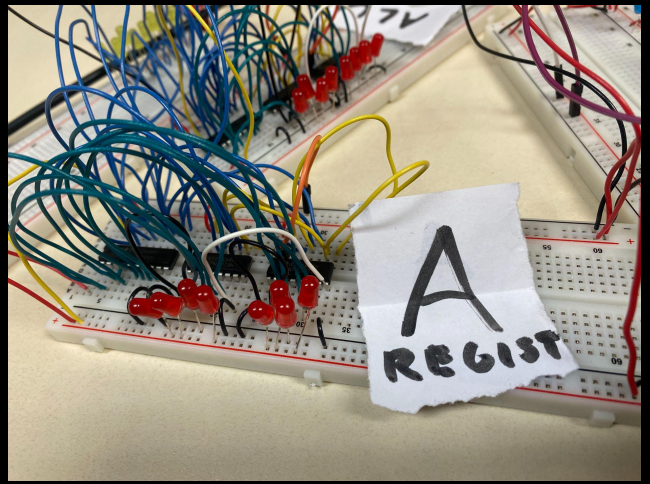




A-Register

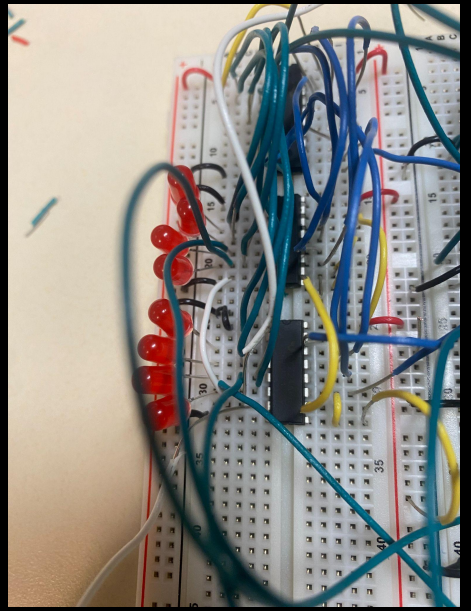
- **Accumulator**
  - Arithmetic operations
  - Stores result of operation
- **Transceiver Chip**
- **LEDs**
  - Test functionality
  - Show output of algebraic operations

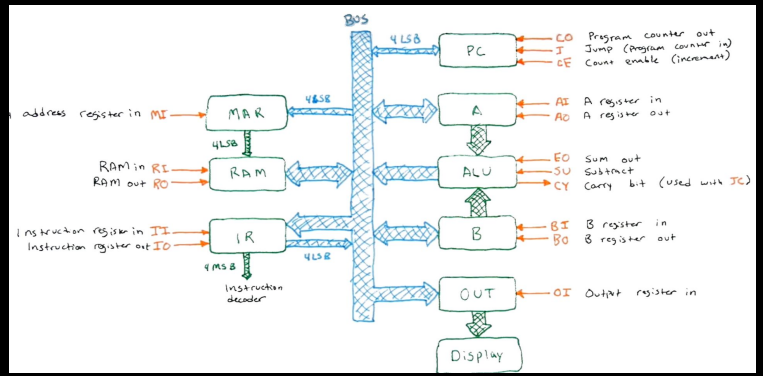




B-Register

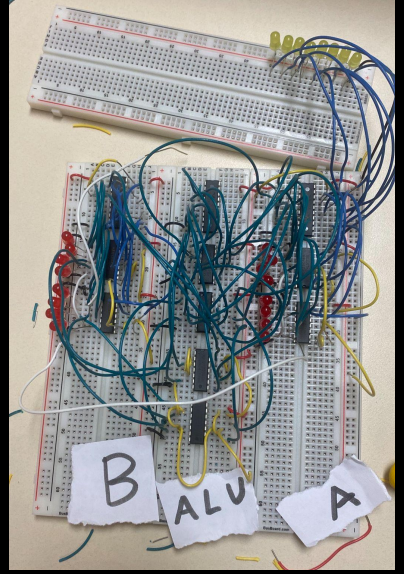
- **Function**
  - temporary storage location for data
  - hold data during transfer
- **4-bit registers**
  - Stores 4 bits of data
- **Bidirectional Bus Transceivers**
  - data flow between the B and A register

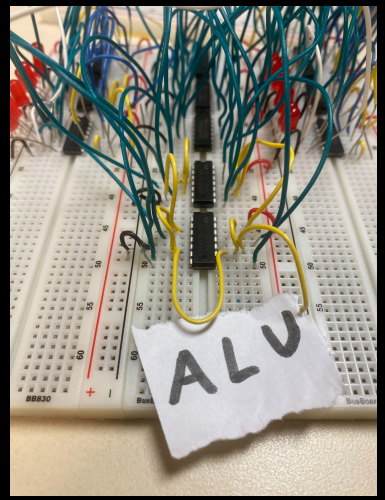




ALU (Arithmetic Logic Unit)

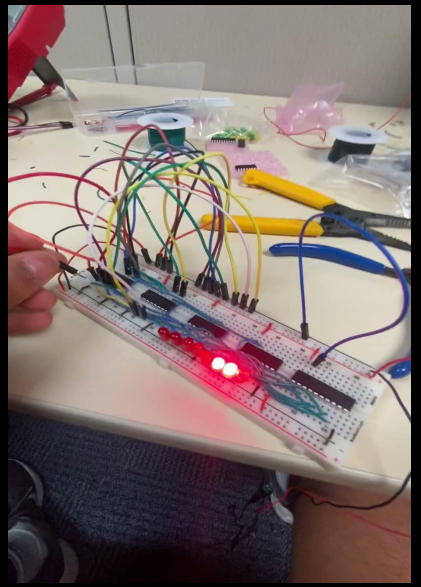
- **A and B registers connection**
  - Outputs the sum or difference of registers
- **XOR Logic (Exclusive OR)**
  - Helps determine the sum
- **4-bit Binary Adder**
  - Add two 4-bit binary numbers

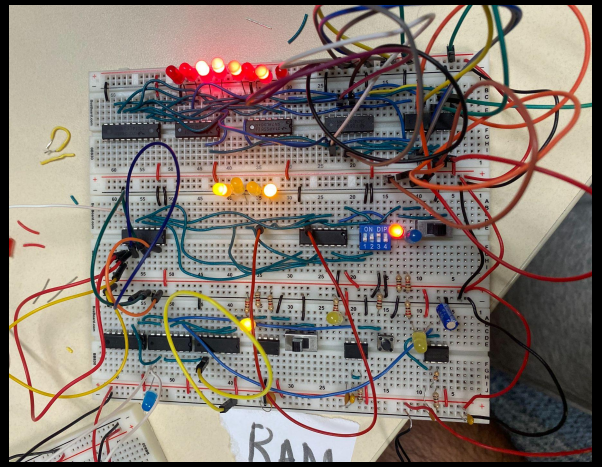




RAM (Random Access Memory)

- **4-bit addresses**
  - 16 bytes of RAM
- **Flip-Flop chip (data storage)**
- **Multiplexer (Quad 2-to-1)**
  - Channels multiple data lines into one.
- **BUS Transceiver Chip (buffer)**
  - Bidirectional data to flow between buses.



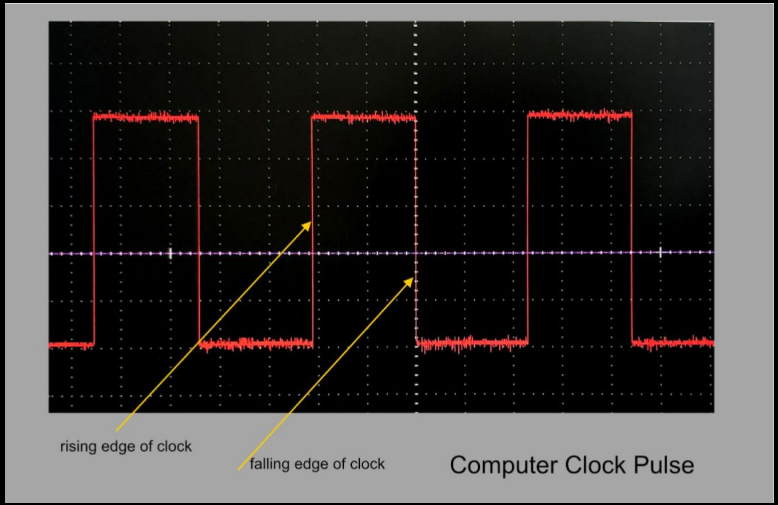


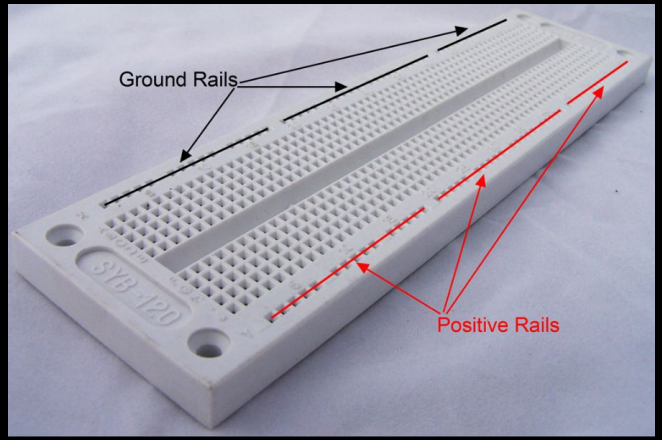
Program Counter

- Counts in binary to know computer execution
- **JK flip-flop**
  - Storage elements that counts binary numbers
- Connected to Clock Module to test input of counter

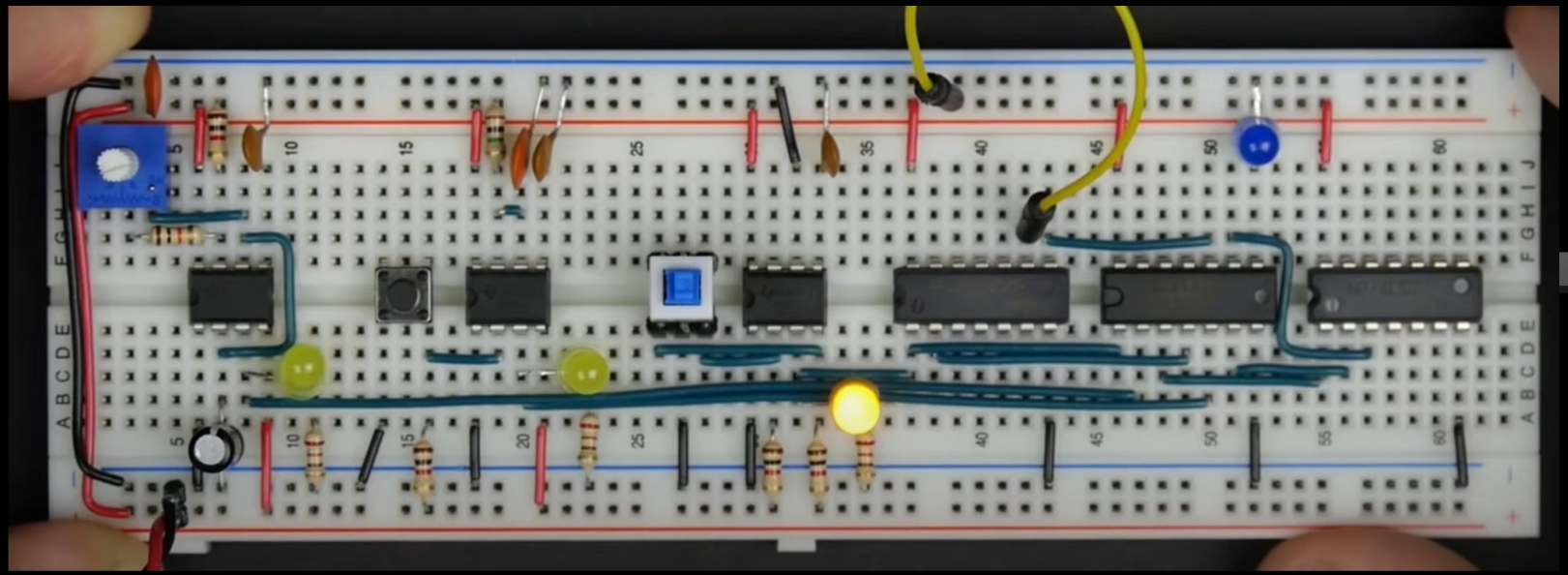
Challenges

- Steep learning curve, but rewarding
  - Understanding the architecture
  - Getting familiar with the components
  - Reading the data sheets
- Wiring issues needed significant amount of time for troubleshooting
  - Power supply backwards
  - Power strip not getting power
- Used an Oscilloscope to check voltages and signals

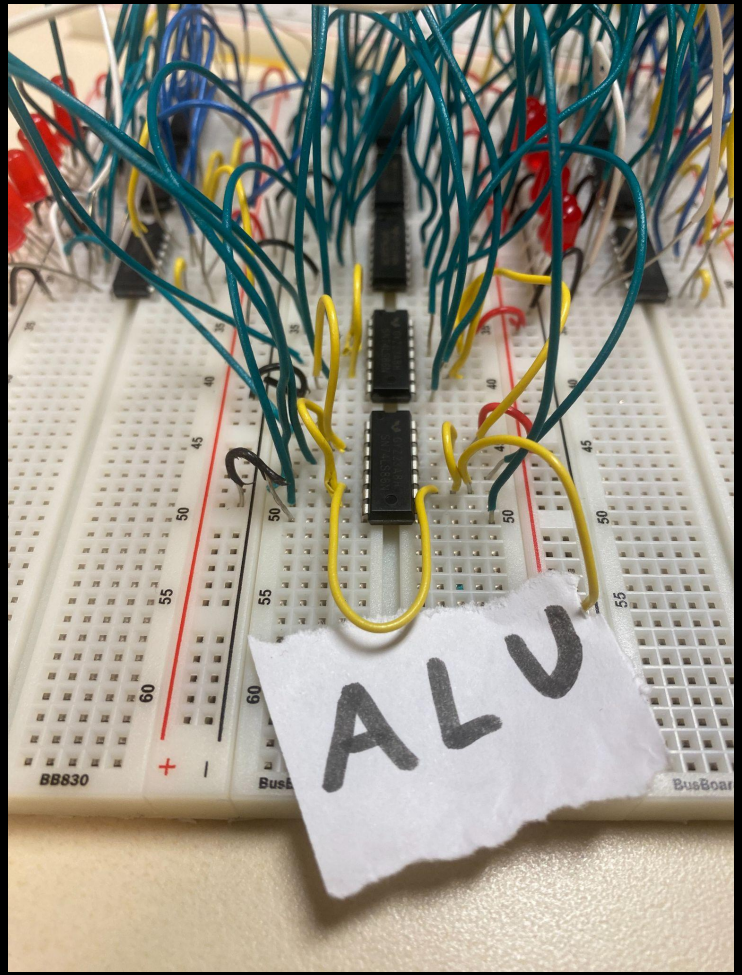




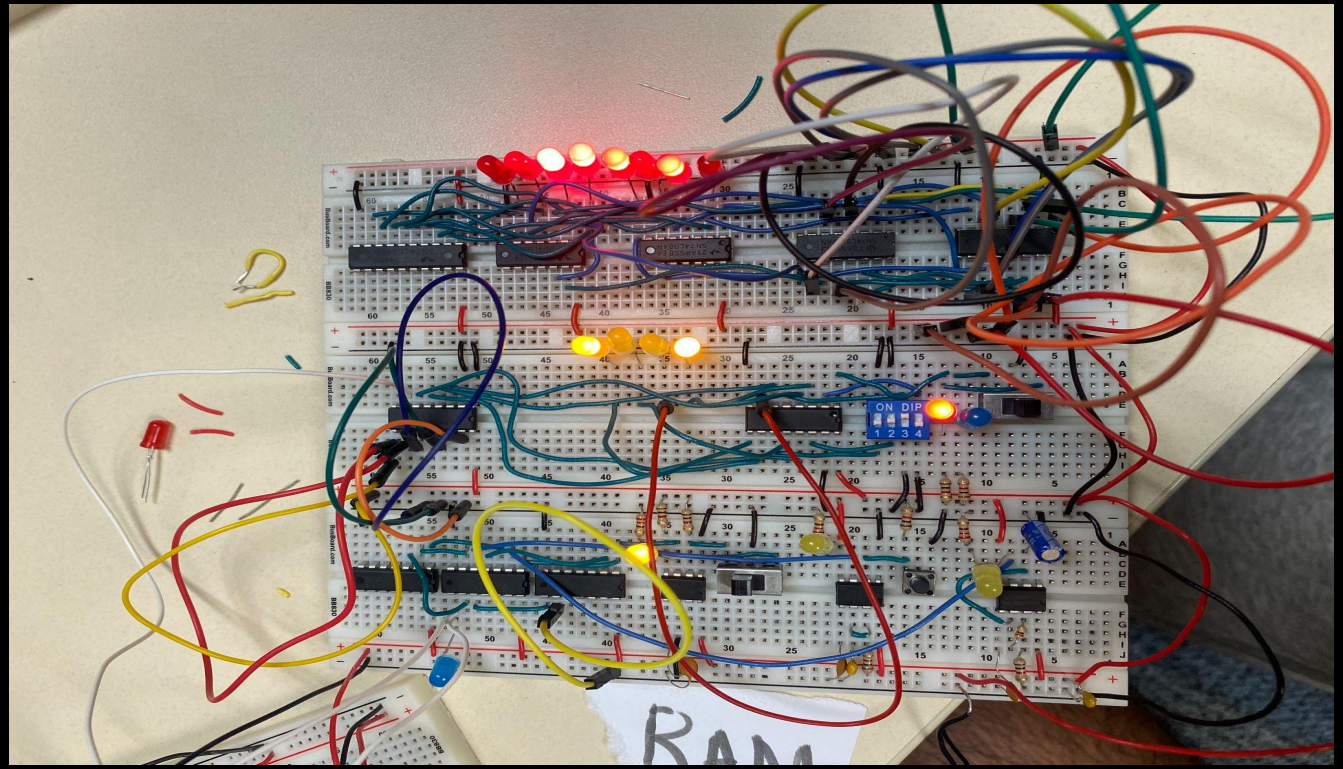
Results



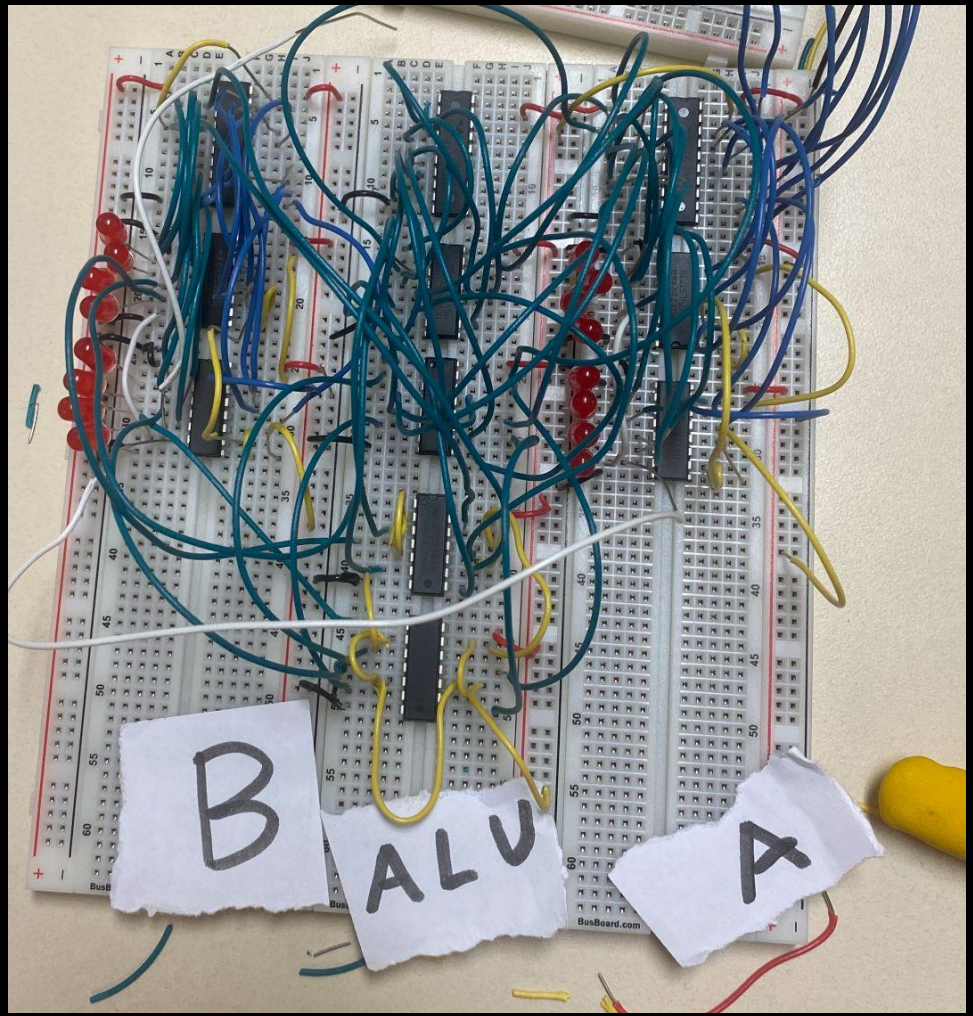
Results



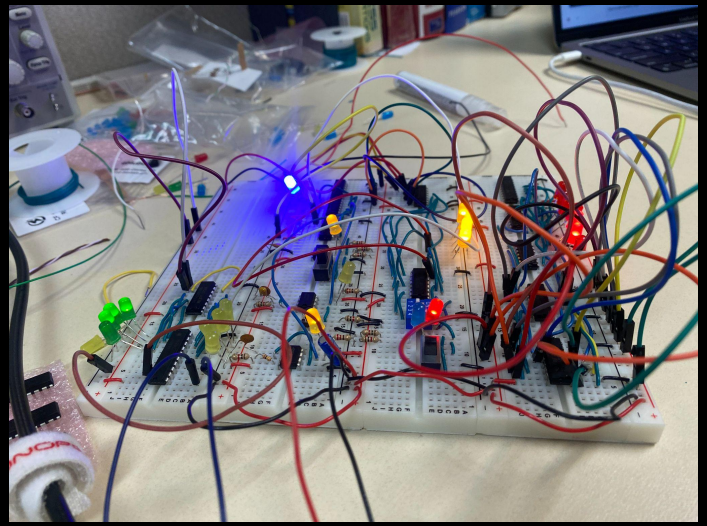
Results

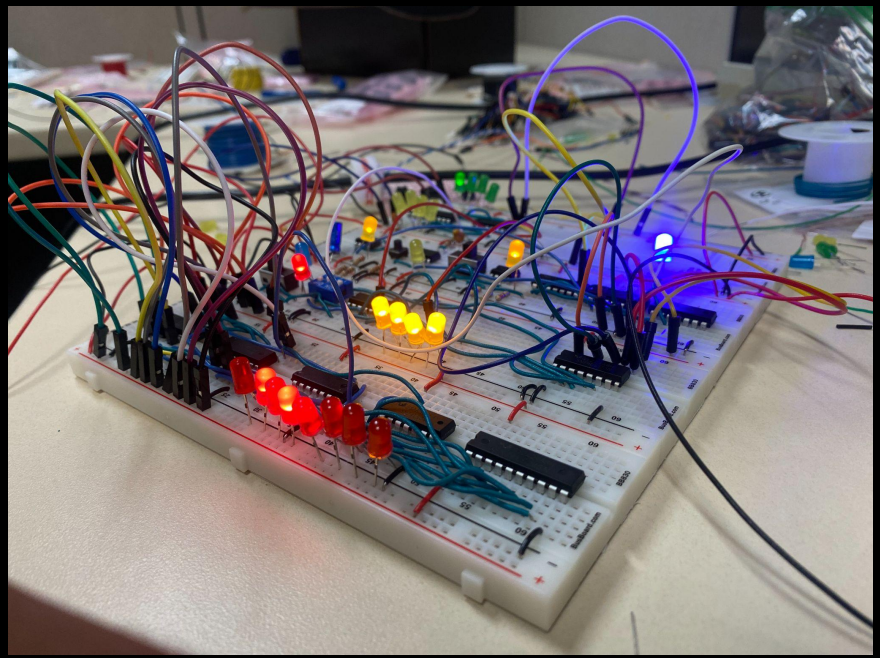


Results



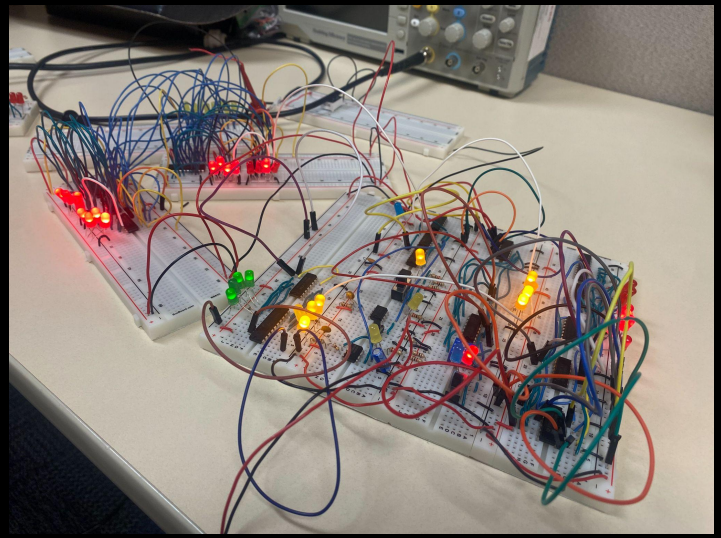
Results

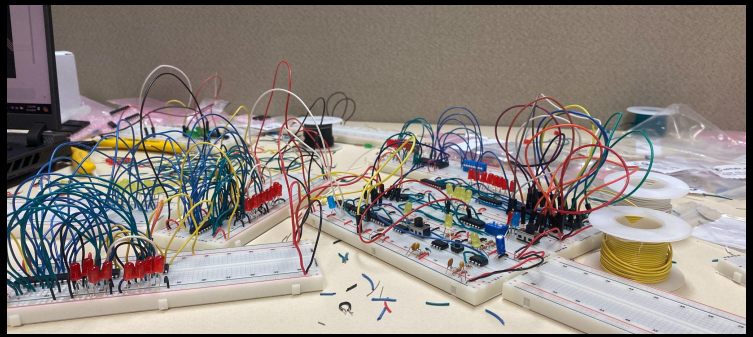




Conclusion

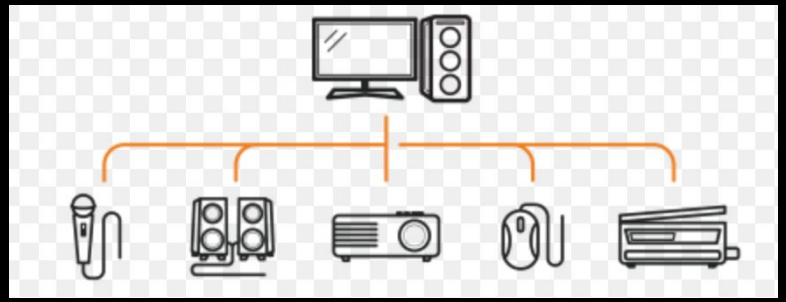
- Understand CPU architecture
- Able to learn and teach how to make a computer using all the parts
- Created a instruction guide that could be followed to replicate a working computer





Future Goals

- Finish putting together whole computer
- Expand the Breadboard Computer to be able to run programs in assembly code via an online IDE
- Add more peripherals such as displays, keyboards, or communication interfaces



```
def add_forty_two(n)

  pushq %rbp
  movq  %rsp, %rbp
  addl  $42, %edi
  movl  %edi, %eax
  popq  %rbp
  retq

end
```

Acknowledgements

Thank you to our advisors Professor Richard Martin and Professor Richard Howard for their guidance and support throughout this project.

Thank you to Jenny, Ivan, and WINLAB for this tremendous opportunity to work on this project!

Thank you!

Come stop by our demo to see our computer running!

Questions?