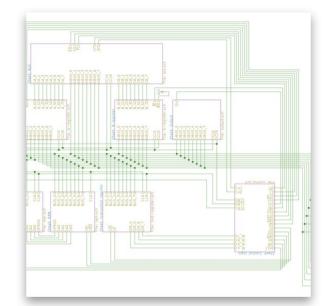
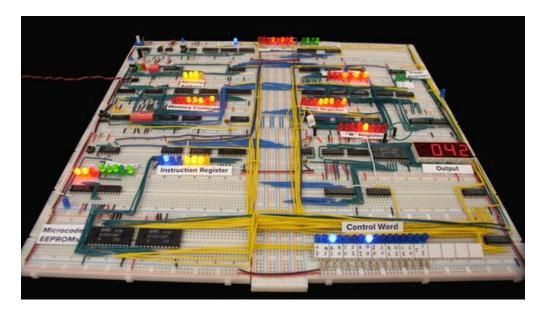


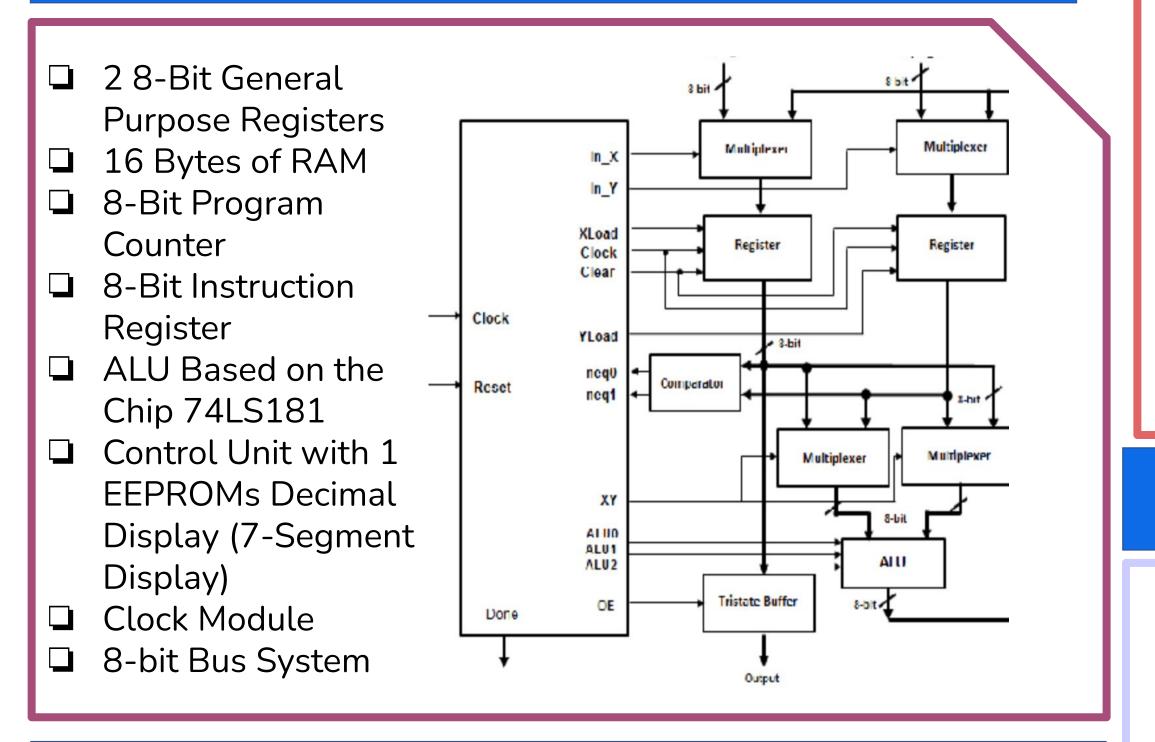
Objectives

- **GOAL**: Built an 8-bit computer from scratch on a breadboard for fun.
- Learn and teach how to build a computer using different components
- Documented the step-by-step process using schematics, notes, and images so that others can replicate the creation



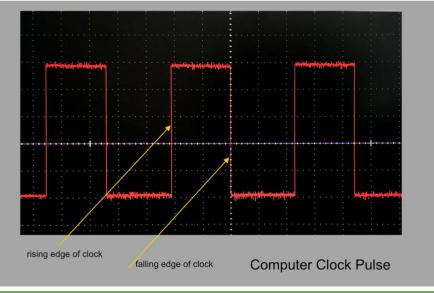


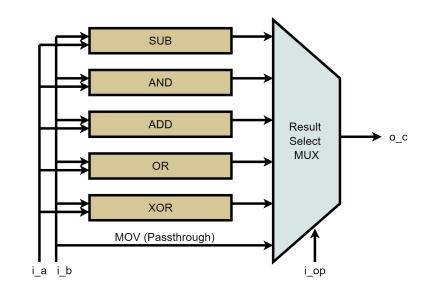
Architecture and Components



Challenges

- Difficulties in pinpointing faults in complex wiring like the ALU
- Encountered many connectivity issues, especially in the clock module
- Used Oscilloscope to check voltages and signals to ensure proper connectivity and to debug any issues

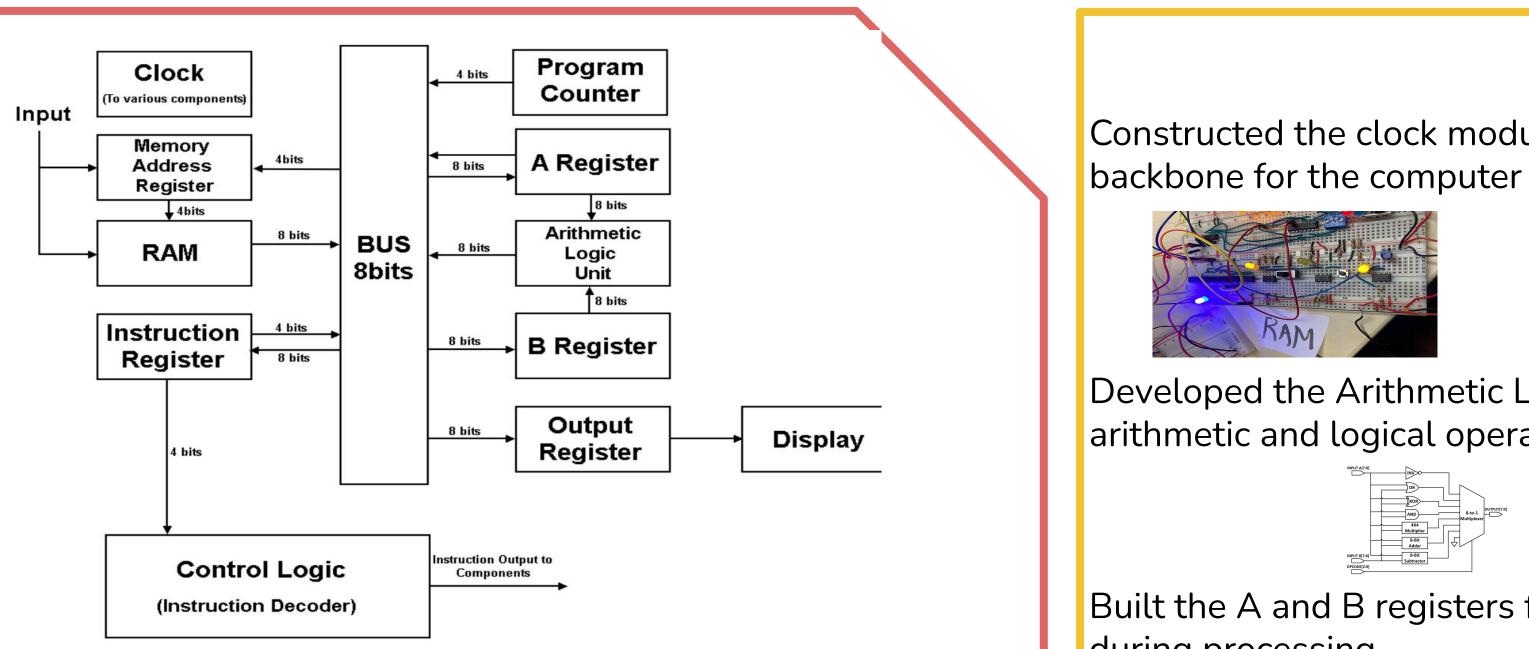




Breadboard Computer Dilan Gandhi^{HS}, Rithvik Madiraju^{HS}

Advisors: Dr. Richard Howard and Dr. Richard Martin

Methodologies



Subcompents

- 310 wires and 500 connections total

- Clock: Test modules to ensure proper functionality.
- ALU
- RAM
- Registers

Intergration

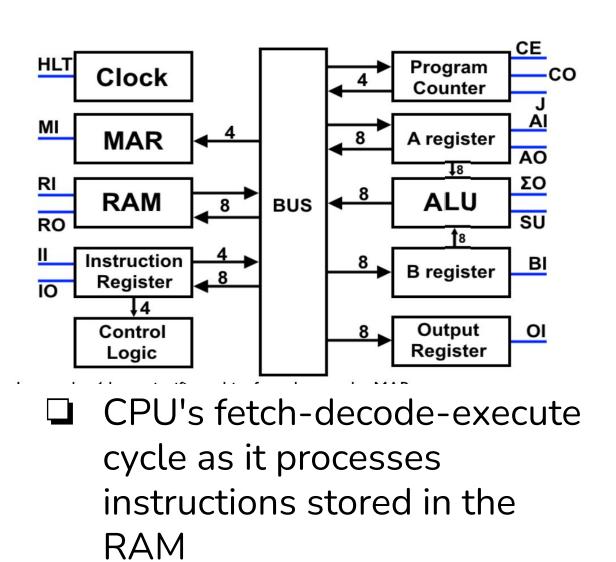
- All subcompents completed; intergration left to do
- Power and ground connections are very important _
- Various chips, including flip-flops and 555 timers

Completion

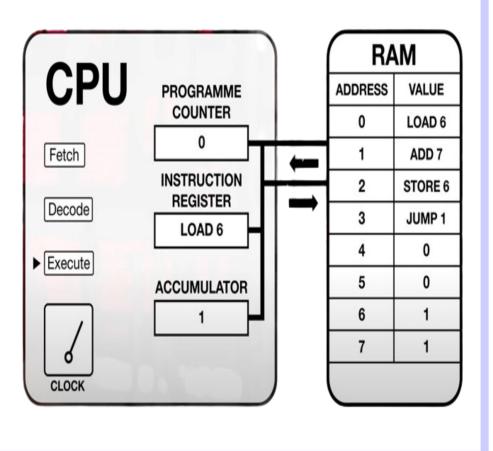
- EEPROMs to store and load programs into memory
- Will function as 8 bit computer and Calculator

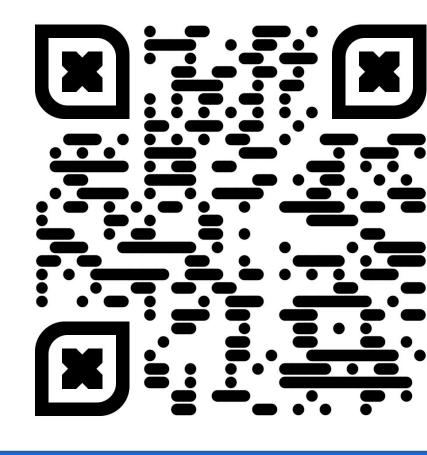
Structure of CPU

- Enable or disable the EEPROMs with the necessary control signals
- EEPOMS generate active high control signals for CPU
- The control unit uses the instruction register to fetch the current instruction from memory



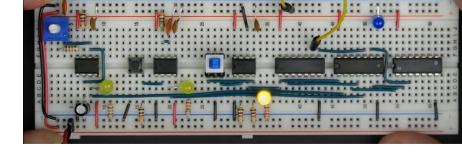
Gignals determine the specific operation the ALU needs to perform





Results

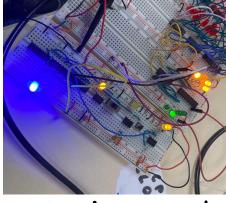
Constructed the clock module to serve as the timing



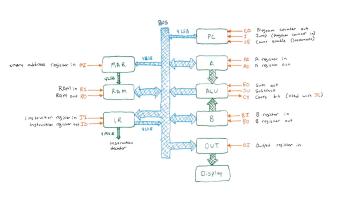
Developed the Arithmetic Logic Unit (ALU) to handle all arithmetic and logical operations.

Built the A and B registers for temporary data storage during processing.

Created the program counter to manage the sequencing of instruction execution.

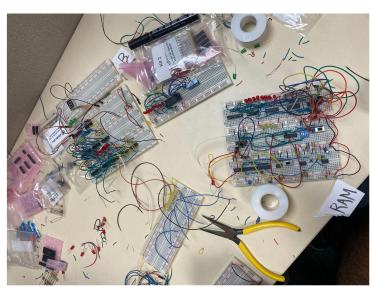


Assembled the RAM module for storing and retrieving data and instructions during operation.



Conclusion and Future Goals

- **Given Successfully built a working Breadboard Computer** from scratch
- Gained an understanding of CPU architecture
- Created a intrustuiton guide that could be followed to replicate a working computer



	82	00	70	START	ORG	ROM+\$0000 #STACK	BEGIN MONITOR	
							ialize ACIA	
				* INPUT		uiw - init	INIIIO ACIA	
				* OUTPL				
				* CALLS		5		
				* DESTR				
				· DESTR	015; 40	C A		
0013				RESETA	EQU	+00010011		
0011				CTLREG	EQU	+00010001		
c003	86	13		INITA	TOB &	#RESETA	RESET ACIA	
C005					STA A			
C008							SET 8 BITS AND 2 STOP	
COOA					STA A			
COOD	78	co	F1		JMD	STGNON	GO TO START OF MONITOR	
				 FUNCT INPUT 		ICH - Input	character	
						in acc A		
				* DESTR				
				* CALLS		IC A		
						Gets 1 ch	aracter from terminal	
			04				GET STATUS	
C013					ASR A		SHIFT RDRF FLAG INTO CARRY	
C014					BCC	INCH	RECIEVE NOT READY	
C016					LDA A	ACIA+1	GET CHAR	
C019					AND A	42.15	SHIFT RORF FLAG INTO CARRY RECIEVE NOT READY GET CHAR MASK PARITY ECHO 4 RTS	
COIB	7E	co	79		JMP	OUTCH	ECHO 4 RTS	
				******	******	********		
				* FUNCTION: INHEX - INPUT HEX DIGIT				
				· INPUT				
				* OUTPO	T: Digi	t in acc A		
				* CALLS				
				* DESTR				
				* Retur	ns to r	conitor if	not HEX input	
				INHEX	BSR	INCH	GET A CHAR	
COIR	8D	70			CMP A		ZERO	
C020	81	30			BMI			
C020 C022	81 2B	30 11			BMI	HEXERR		
C020 C022 C024	81 2B 81	30 11 39			BMI	HEXERR	NOT HEX	
C020 C022 C024 C026	81 2B 81 2F	30 11 39 0A			BMI CMP A	HEXERR #'9 HEXR75	NOT HEX NINE	
C020 C022 C024 C026 C028	81 2B 81 2F 81	30 11 39 0A 41			EMI CMP A BLE	HEXERR #'9 HEXR75	NOT HEX NINE GOOD HEX	
C020 C022 C024 C026 C028 C028	81 2B 81 2F 81 2B	30 11 39 0A 41 09			EMI CMP A BLE CMP A	HEXERR #'9 HEXRTS #'A HEXERR	NOT HEX NINE GOOD HEX	
C020 C022 C024 C026 C028 C028 C02A C02C	81 2B 81 2F 81 2B 81	30 11 39 0A 41 09 46			BMI CMP A BLE CMP A BMI CMP A	HEXERR #'9 HEXRTS #'A HEXERR	NOT HEX NINE GOOD HEX	
C020 C022 C024 C026 C028 C028 C028 C022 C022	81 2B 81 2F 81 2B 81 2B 81 2E	30 11 39 0A 41 09 46 05			BMI CMP A BLE CMP A BMI CMP A BGT	HEXERR #'9 HEXRTS #'A HEXERR #'P HEXERR	NOT HEX NINE GOOD HEX	
C018 C020 C022 C024 C026 C028 C028 C022 C028 C022 C028 C020 C028 C020 C028 C020 C028	81 2B 81 2F 81 2B 81 2E 80	30 11 39 0A 41 09 46 05 07			BMI CMP A BLE CMP A BMI CMP A BGT SUB A	HEXERR #'9 HEXRTS #'A HEXERR #'P HEXERR #7	NOT HEX NIME GOOD HEX NOT HEX FIX A-F	
C020 C022 C024 C026 C028 C028 C028 C022 C028 C022 C028 C020 C028 C020	81 2B 81 2F 81 2B 81 2E 80 84	30 11 39 0A 41 09 46 05 07 0F			BMI CMP A BLE CMP A BMI CMP A BGT SUB A	HEXERR #'9 HEXRTS #'A HEXERR #'P HEXERR #7	NOT HEX NINE GOOD HEX NOT HEX	

Future Goals

Expand the Breadboard Computer to be able to run programs in assembly code via an online IDE Add more peripherals such as displays, keyboards,

