

Assembly Language Program in 8085 Microprocessor

By:- Assignwise

AIM: Write an assembly language Program to add and subtract two 8 bit numbers stored at address 2050 in 8085 Microprocessor. The starting address of the program is taken as 2000.

Trainer Kit Required: 8085 Microprocessor system.

**Task: Verify the given below code using 8085 Microprocessor.
(According to Comment given below)**

Addition

Address	Hex code	Mnemonics	COMMENT
2000	3E H, 08 H	MV1 A, 08H	Load the accumulator with 08H hex number
2002	06 H, 06 H	MV1 B, 06H	Load the register B with 06H hex number
2004	80H	ADD B	Add the content of B with A
2005	32H,50 H,20 H	STA 2050	Store the content of A on 2050H address.
2008	76 H	HLT	End

Subtraction

Address	Hex code	Mnemonics	COMMENT
2000	3E H, 08 H	MV1 A, 08H	Load the accumulator with 08H hex number
2002	06 H, 06 H	MV1 B, 06H	Load the register B with 06H hex number
2004	80H	SUB B	Subtract the content of B with A
2005	32H,50 H,20 H	STA 2050	Store the content of A on 2050H address.
2008	76 H	HLT	End

Introduction

In the field of computer engineering, the 8085 microprocessor is a commonly used and well-liked microprocessor. Intel released the first 8-bit microprocessor in 1976. Early personal computers and household appliances frequently employed the 8085 Microprocessor, which was created for compact computer systems and embedded applications. From straightforward arithmetic activities to more difficult ones like data processing and control operations, the 8085 Microprocessor is a flexible microprocessor that can be utilized for a variety of tasks. Low-level programming languages like assembly language are used to create applications that the microprocessor can run right away. Assembly language can be used to program the 8085 Microprocessor to carry out a variety of activities.

Writing an assembly language program to add and subtract two 8-bit values stored at address 2050 on the 8085 microprocessor is required for this project. The program's initial address is assumed to be 2000. One component of the program is for addition, and the other is for subtraction. The addition program begins by loading the accumulator with an 08H hex value, loading the register B with a 06H hex value, adding the contents of B and A, storing the results on address 2050H, and then terminating. The subtracting program begins by loading the accumulator with an 08H hex value, loading the register B with a 06H hex value, subtracting the contents of B and A, storing the results on address 2050H, and then exiting. We must first compile the code before running it on the 8085 microprocessor in order to validate the provided code. An assembler, which transforms assembly language code into machine code that the CPU can execute, can be used to put the code together. The compiled code can then be fed into the CPU and run. After completing the necessary actions, the microprocessor will save the outcome at the designated address.

Finally, this project offers a chance to gain knowledge about assembly language programming and the 8085 microprocessor. It explains how to construct a program in assembly language that adds and subtracts two 8-bit numbers and how to check the code using an 8085 microprocessor. We can obtain a greater understanding of microprocessor operation and assembly language programming by completing this project.

Task 1

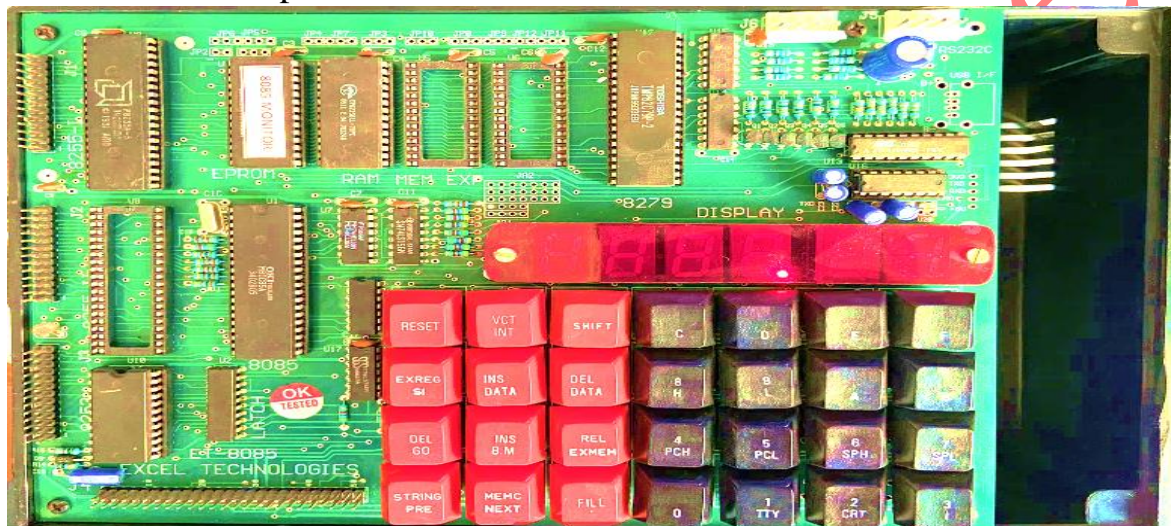
Addition

To add the binary number through the 8085 microprocessor.

The steps are as follows:

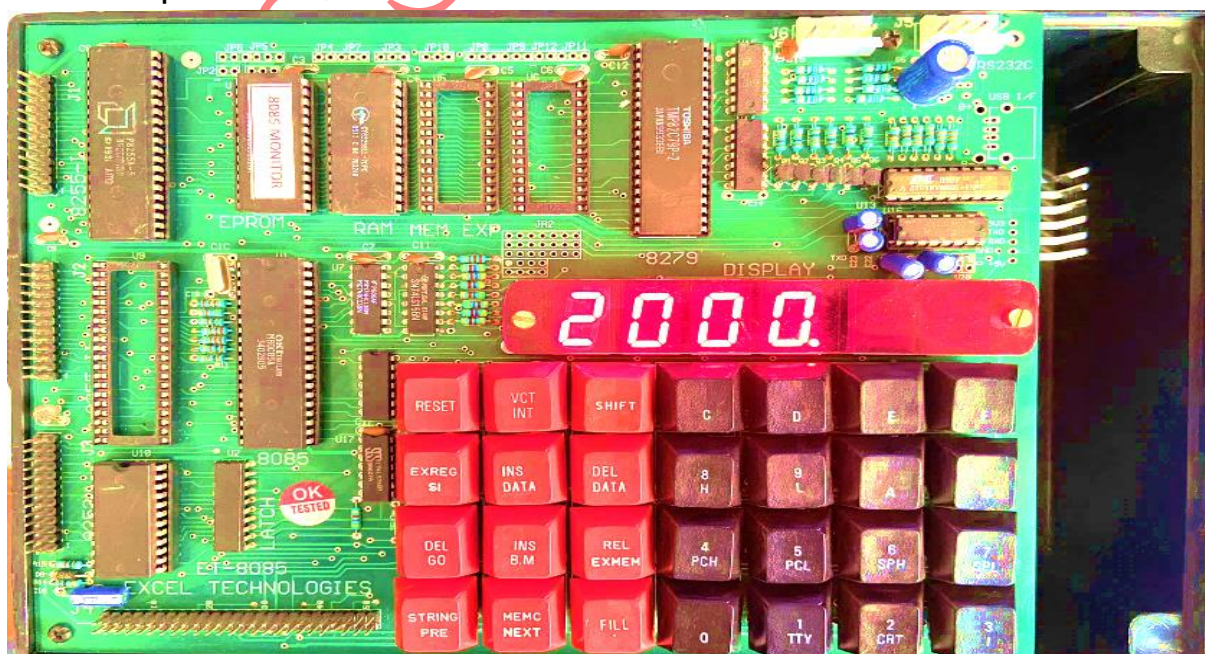
Step 1

First, power on the 8085 microprocessor. then, press RESET and REL EXMEM on the 8085 microprocessor.



Step 2

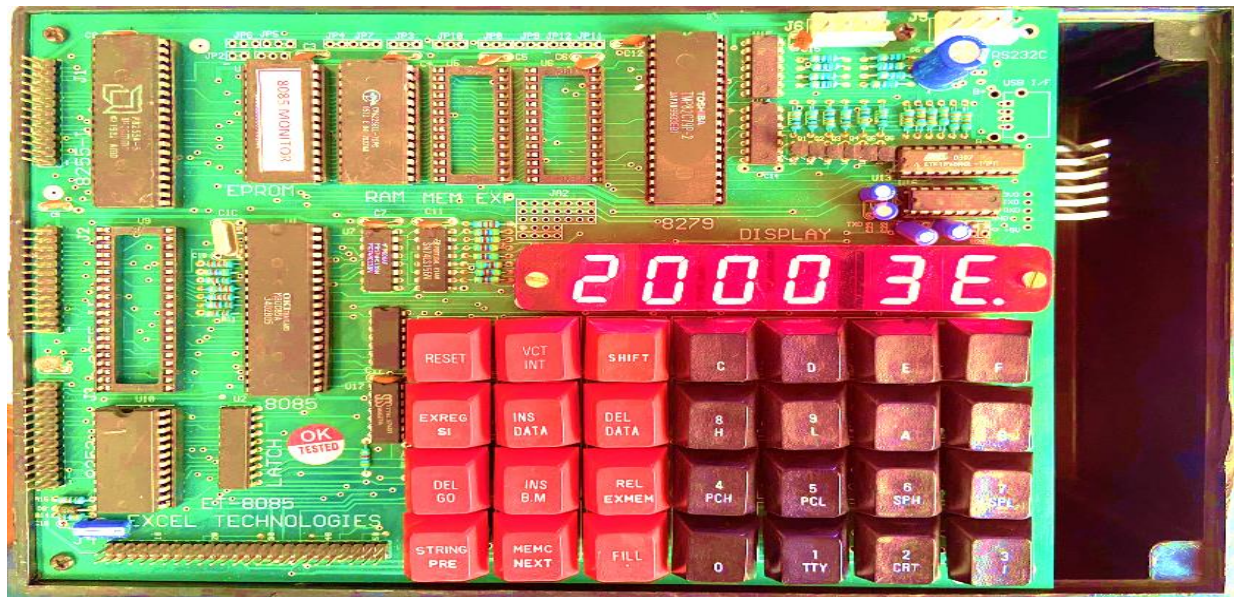
After the 8085 microprocessor has started, enter the first address into the microprocessor. i.e. 2000



Step 3

Press MEMC NEXT once you've entered the address.

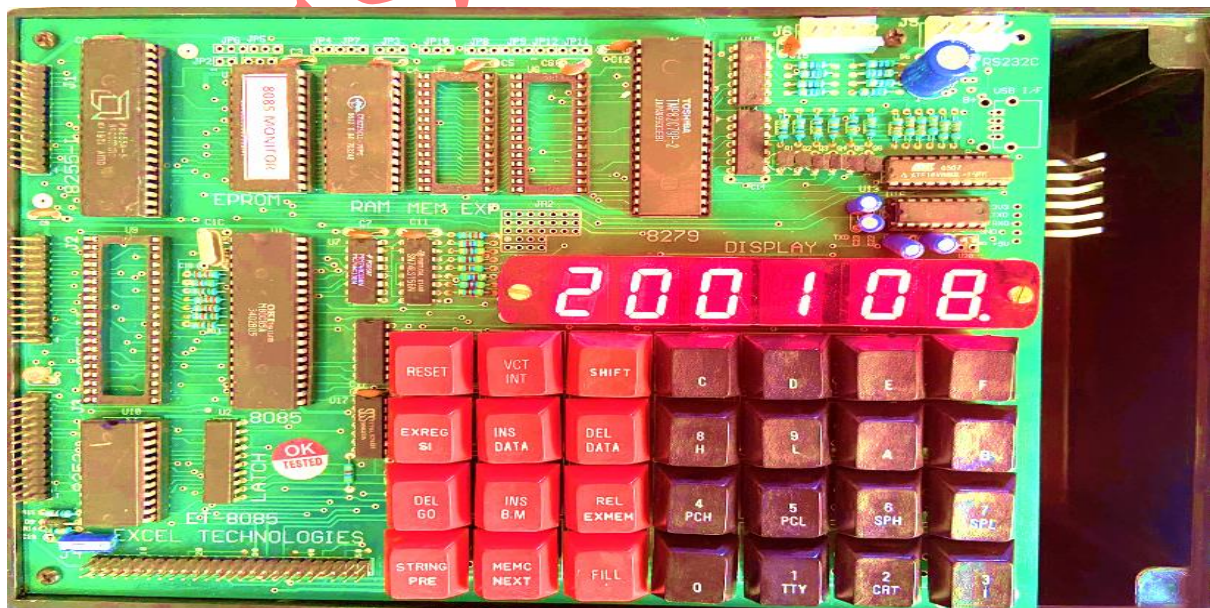
Making addresses more valuable. i.e 3E



Step 4

Enter another address 2001 into the 8085 microprocessor by pressing MEMC NEXT. Next, click MEMC NEXT.

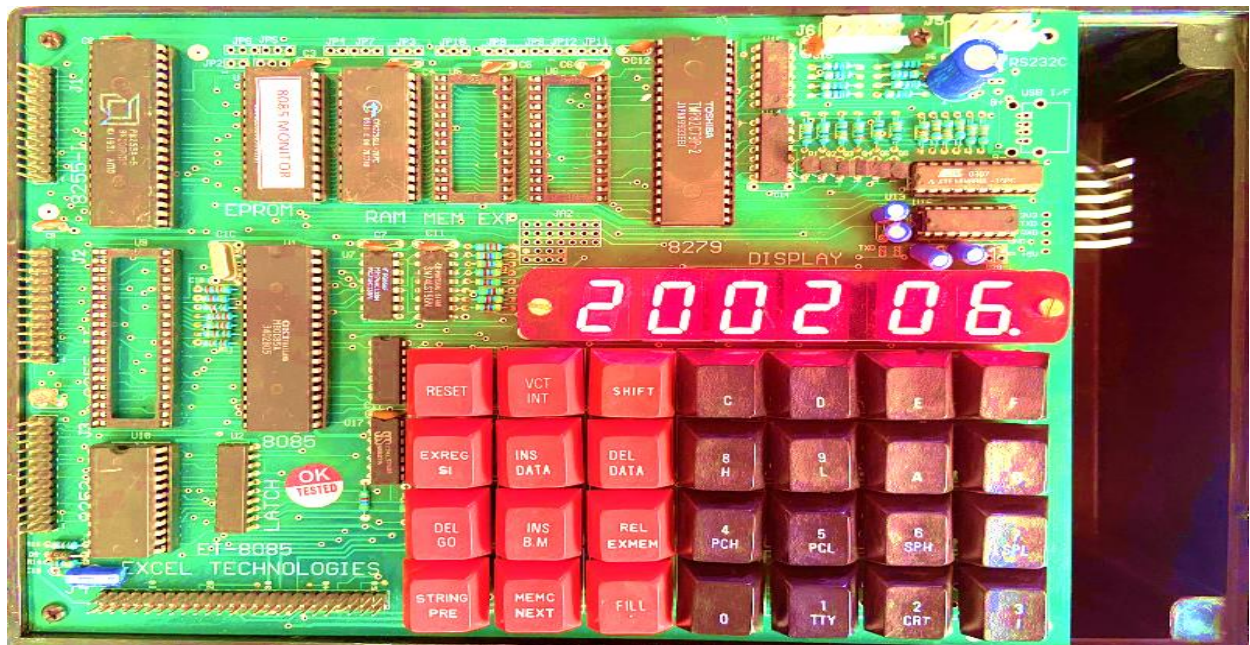
Likewise, enter 08 in address 2001.



Step 5

To input the address 2002, hit MEMC NEXT one again.

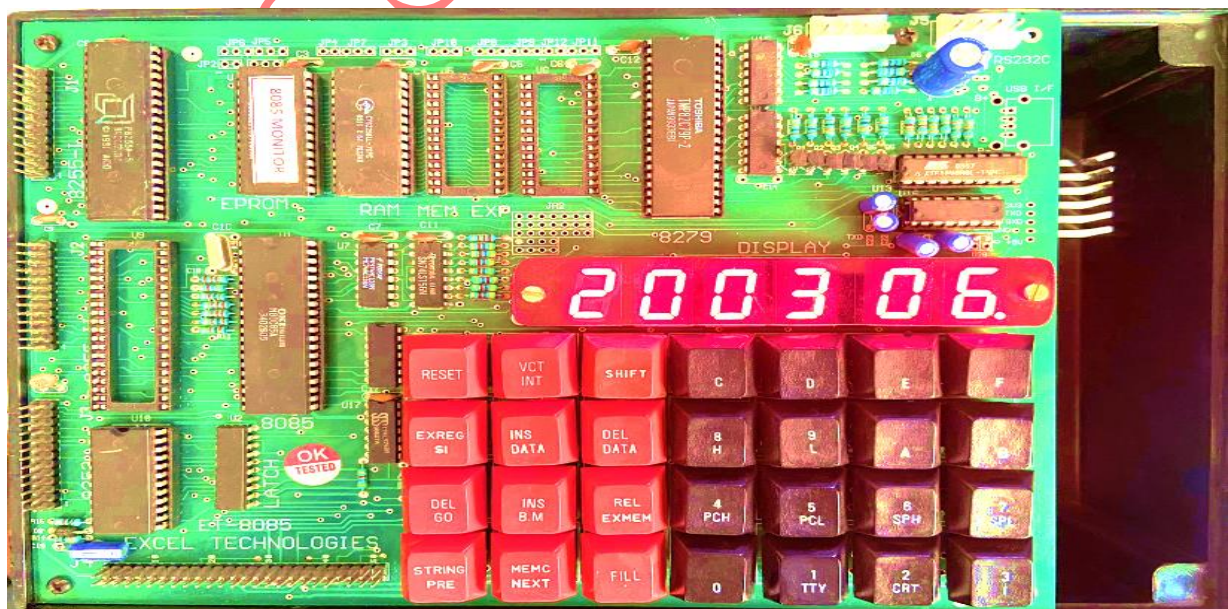
After that, press MEMC Next and type 06 into the advertisement for 2002.



Step 6

Once more, hit MEMC NEXT, then type in 2003.

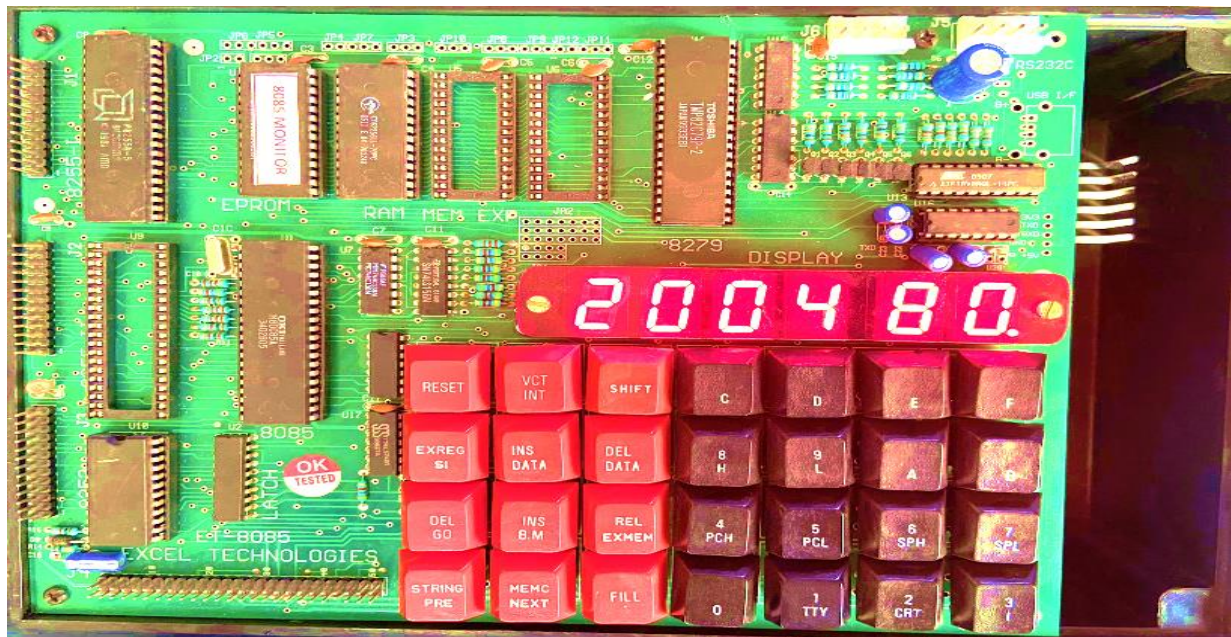
Press MEMC NEXT after that, then type 06 into the advertisement for 2003.



Step 7

Once more, hit MEMC NEXT, then type in 2004.

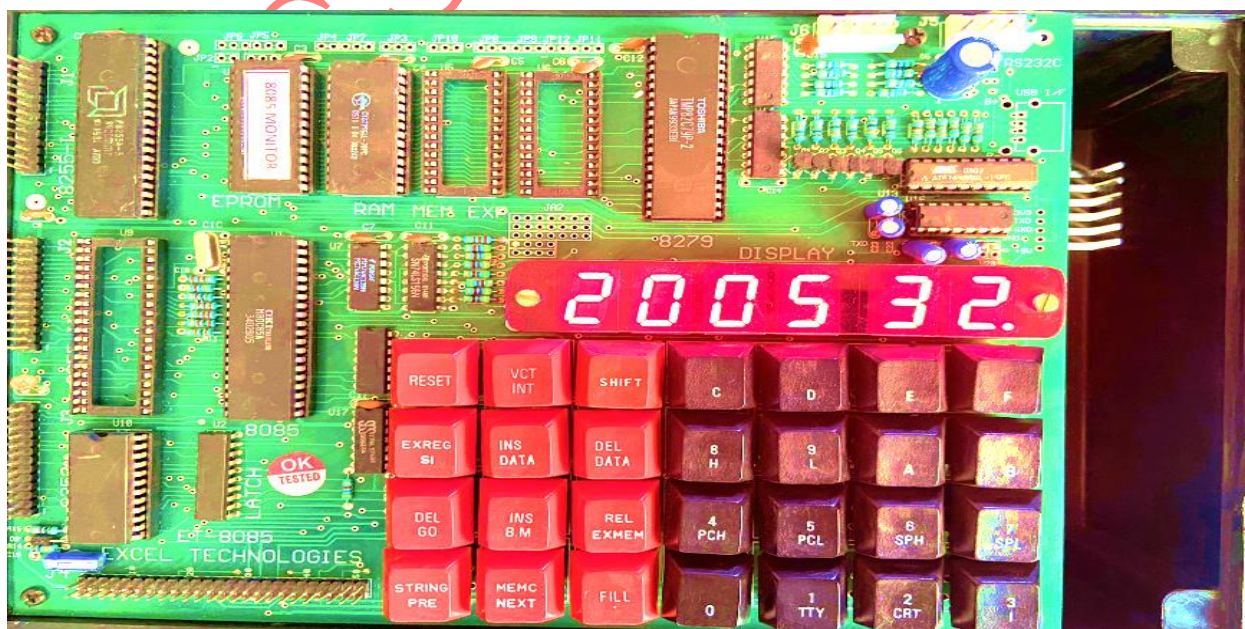
Then, in 2004 press MEMC NEXT and enter the value 80 in the advertisement.



Step 8

Once more, hit MEMC NEXT, then type in 2005.

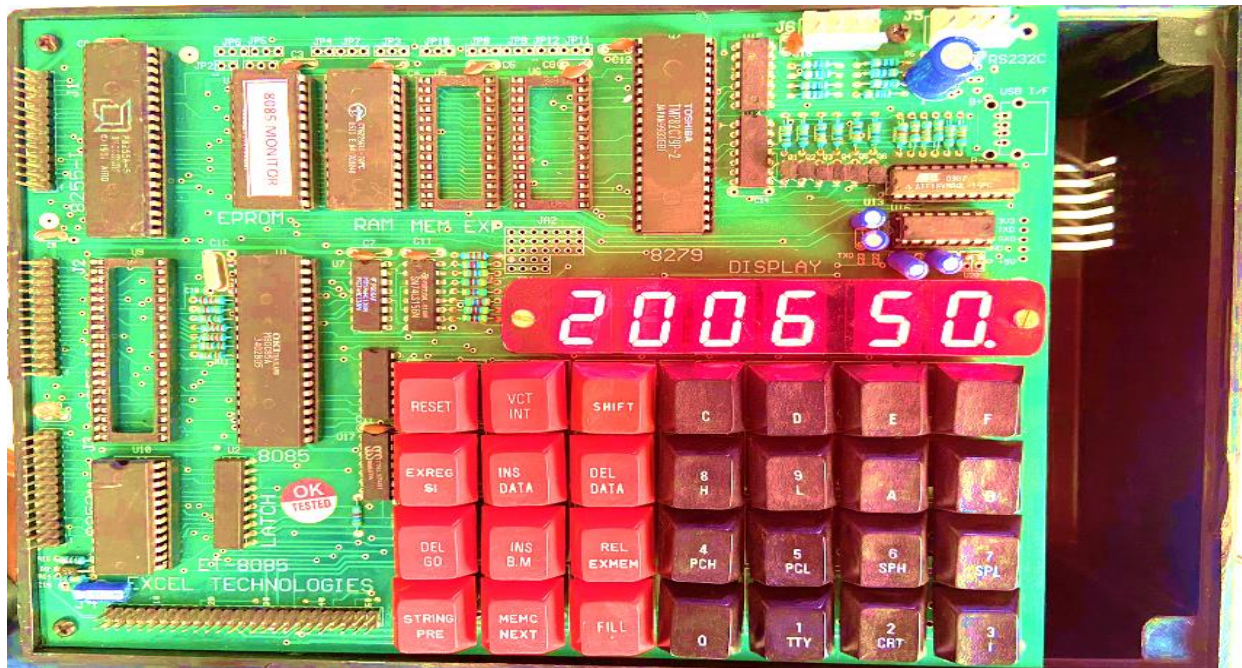
Then, press MEMC NEXT, and enter the value 32 in the advertisement.
2005



Step 9

Once more, hit MEMC NEXT, then key in 2006.

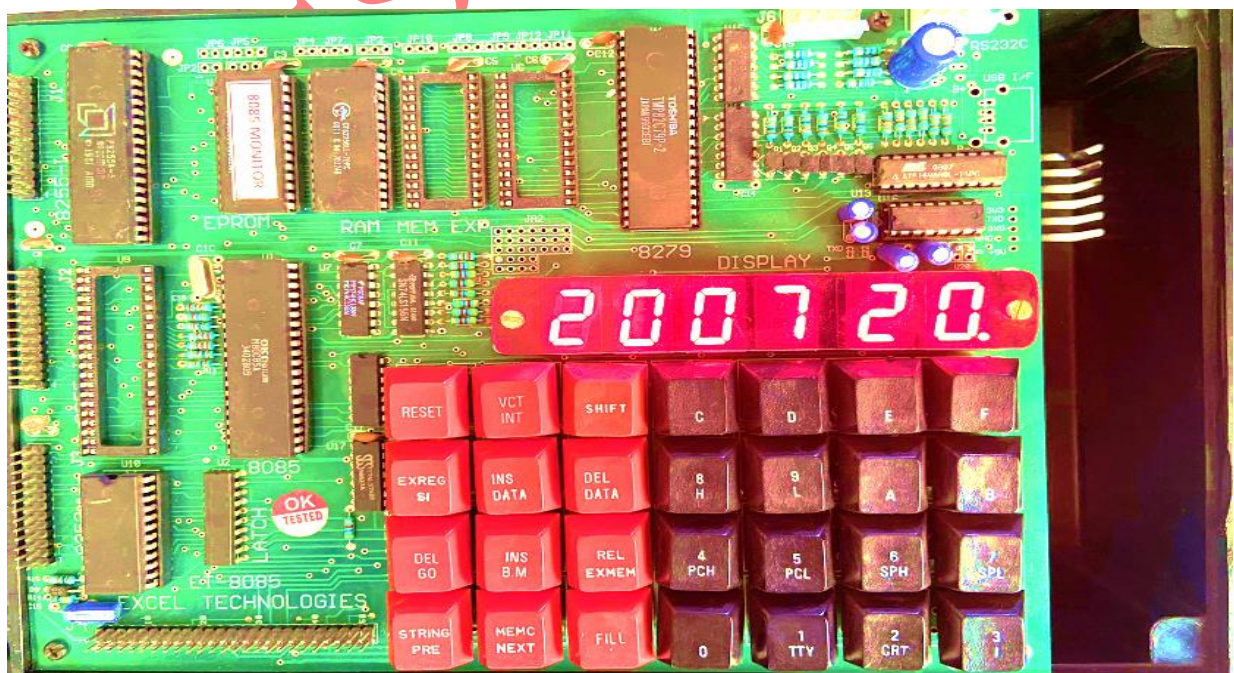
Next, click MEMC Next, and then type 50 into the ad field.



Step 10

Once more, hit MEMC NEXT, then type in 2007.

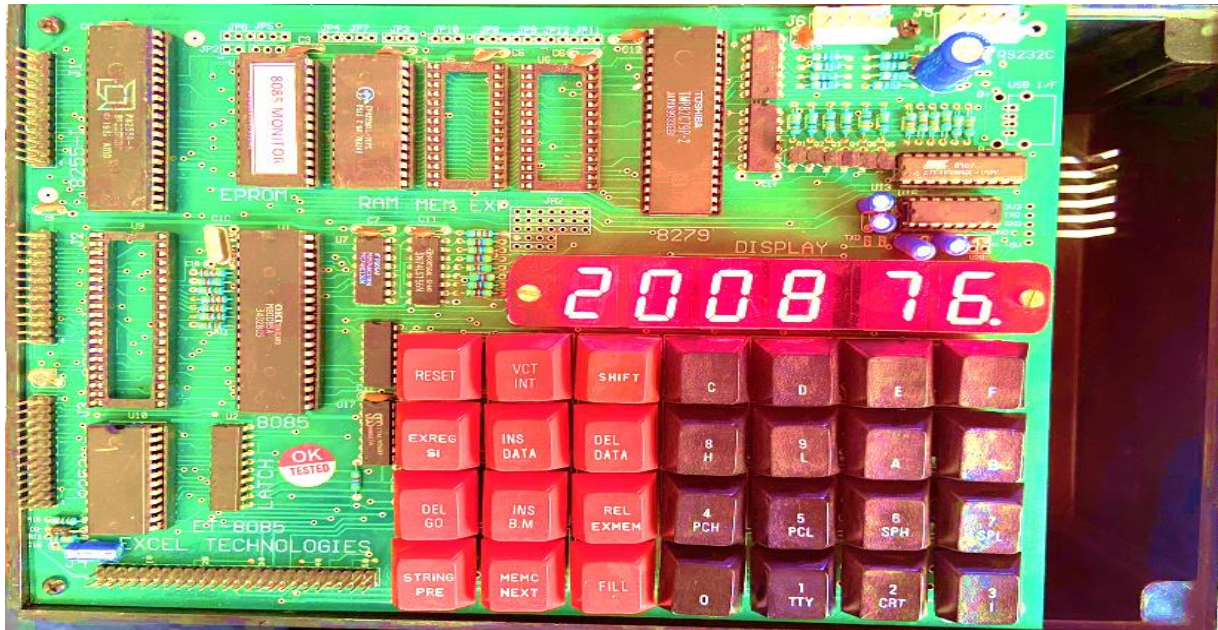
Input the value 20 into the advertisement after pressing MEMC NEXT.



Step 11

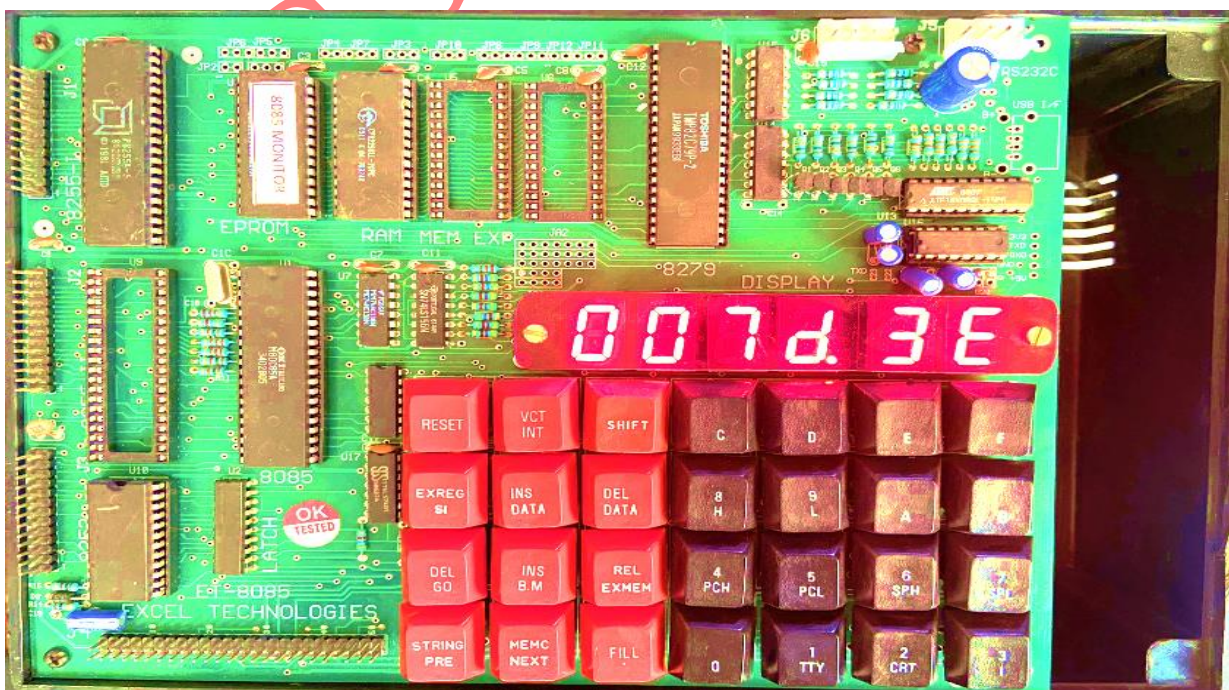
Once more, hit MEMC NEXT, then type in 2008.

Then, enter the value 76 in the advertisement by pressing MEMC NEXT. 2008.



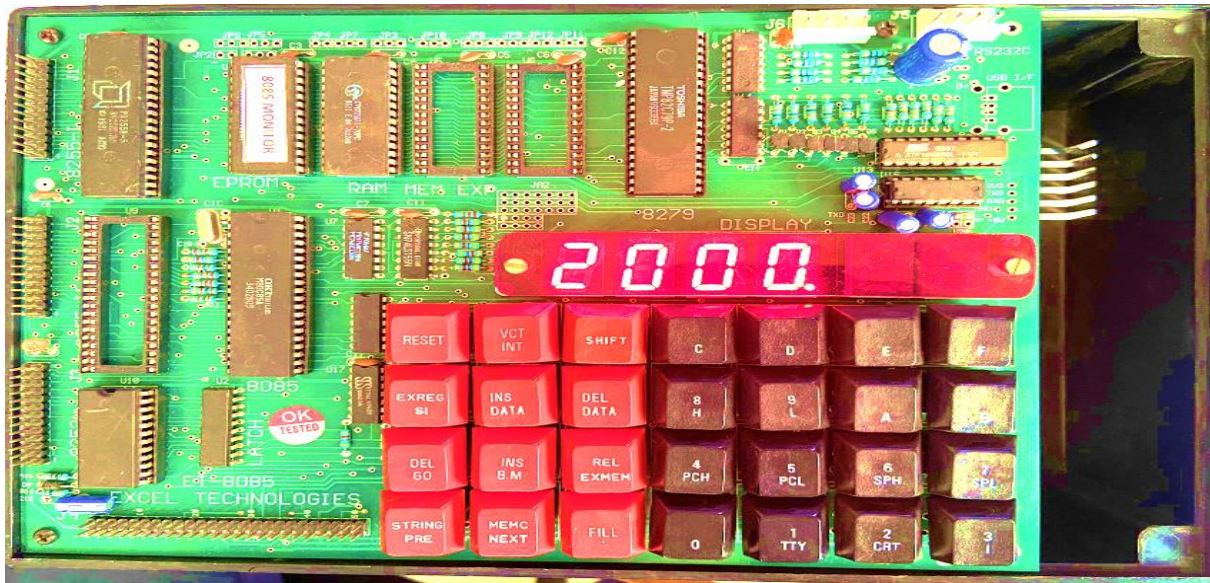
Step 12

After inputting the ad. 2008 value in this phase, I have to hit the buttons FILL and DEL GO.



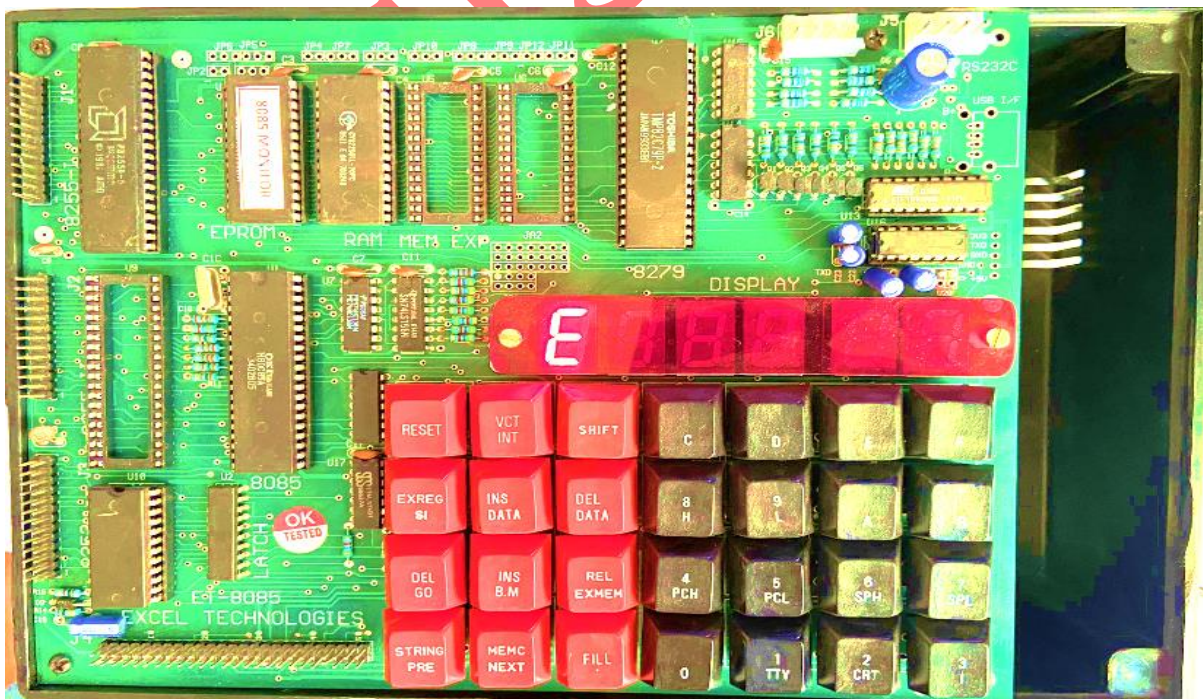
Step 13

I have entered the starting address, 2000, in the 8085 microprocessor after hitting the buttons FILL and DEL GO.



Step 14

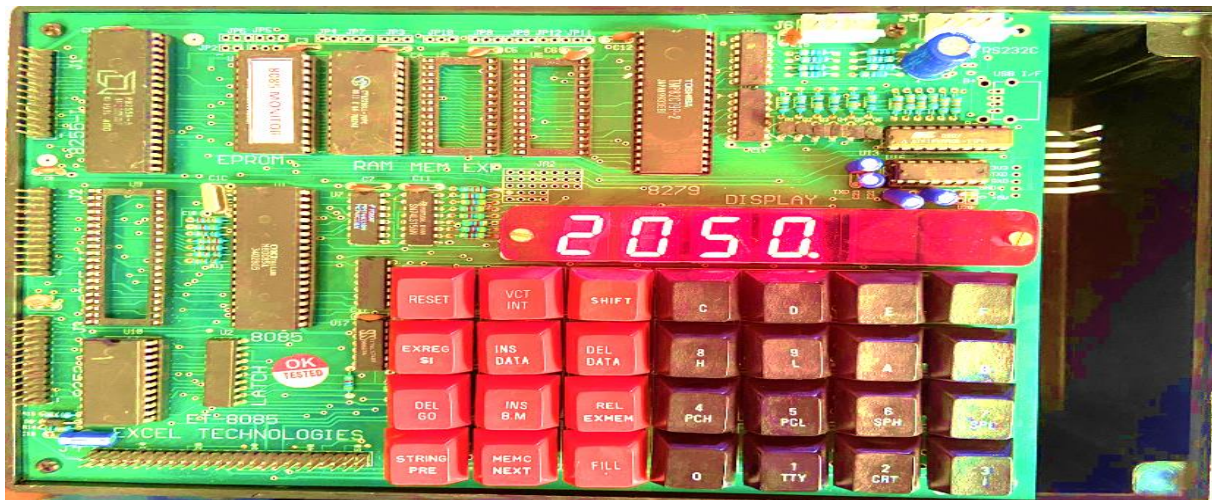
You must click the FILL button after entering the initial advertisement 2000. Then the procedure will run.



Step 15

I then entered the ad after using the RESET and REL EXMEM buttons.
2050

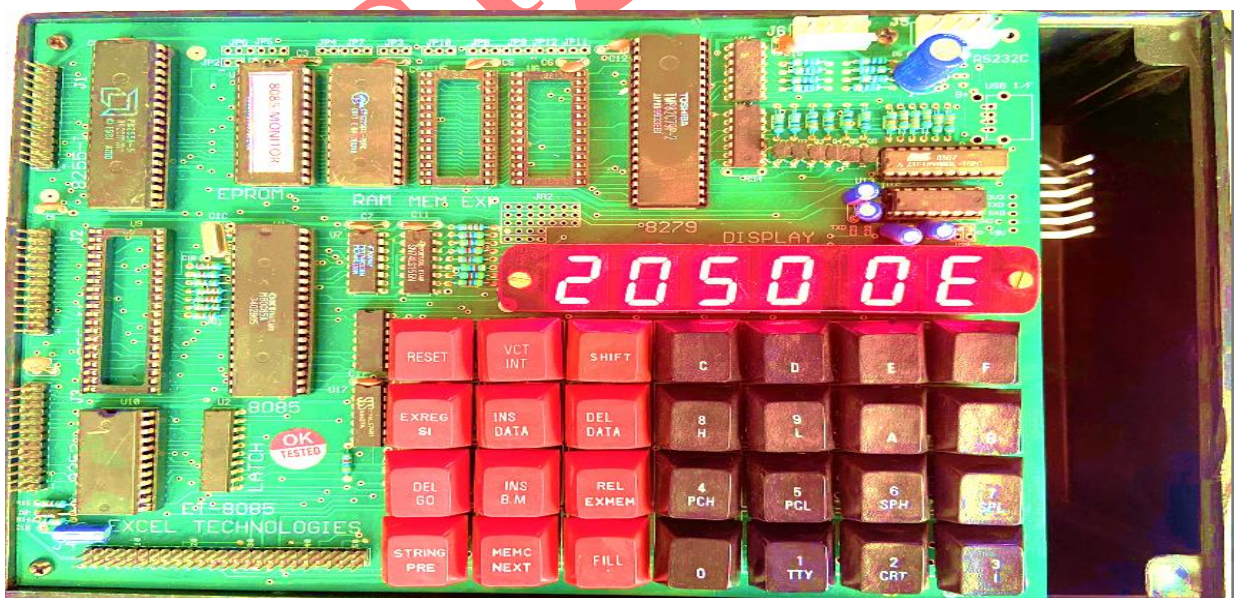
To store the finished product, this stage is completed.



Step 16

After entering the advertisement, 2050 to store the outcome. Then, in order to determine if the amount had been saved in the advertisement, I had pushed the MEMC NEXT button.

The final result is provided at this point, and this is also where my data is stored.



Finally, in this procedure, I added the two numbers, 8 and 6, and the result was 14; it is stored as indicated in the above image.

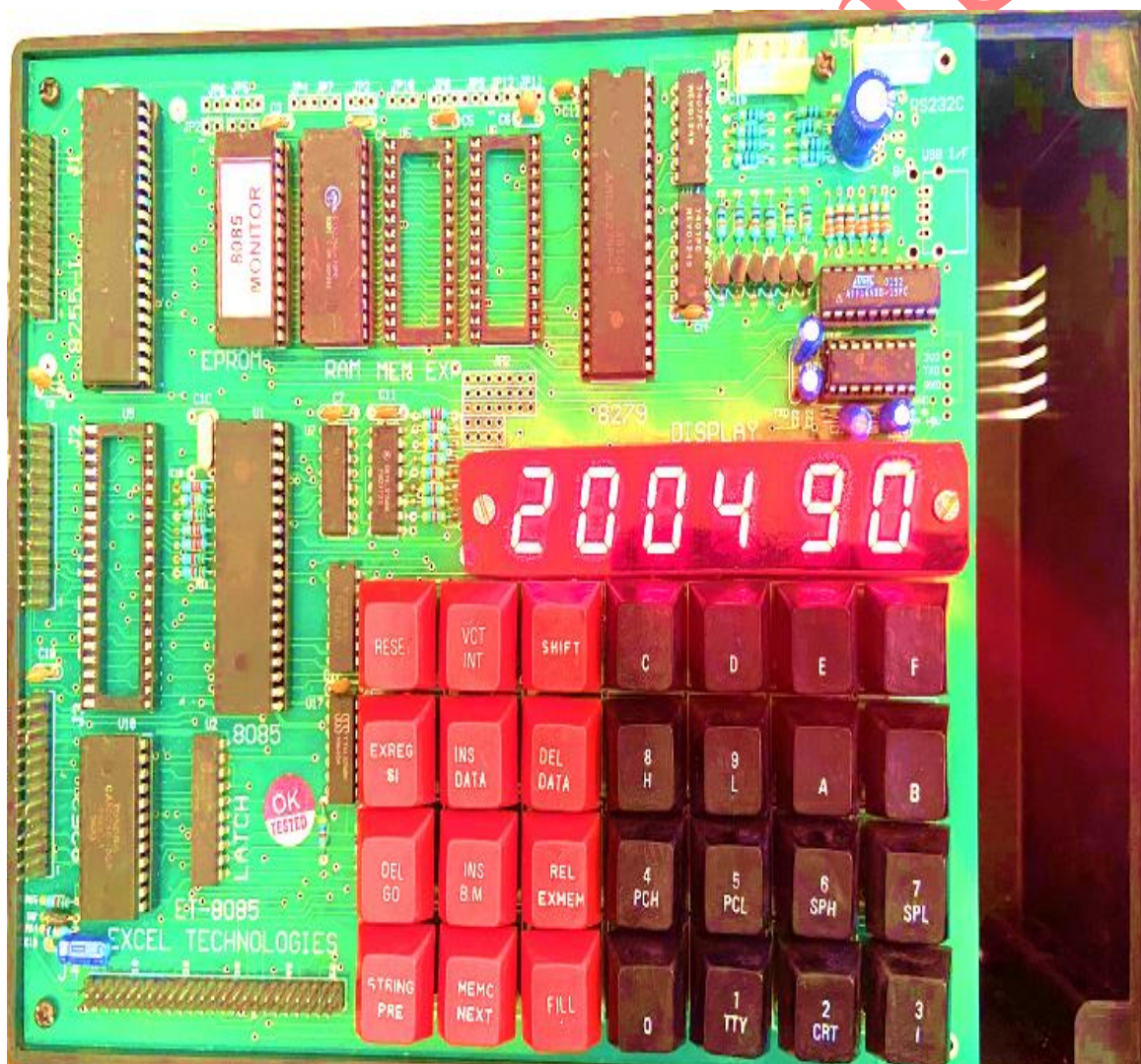
Task 2

Subtraction

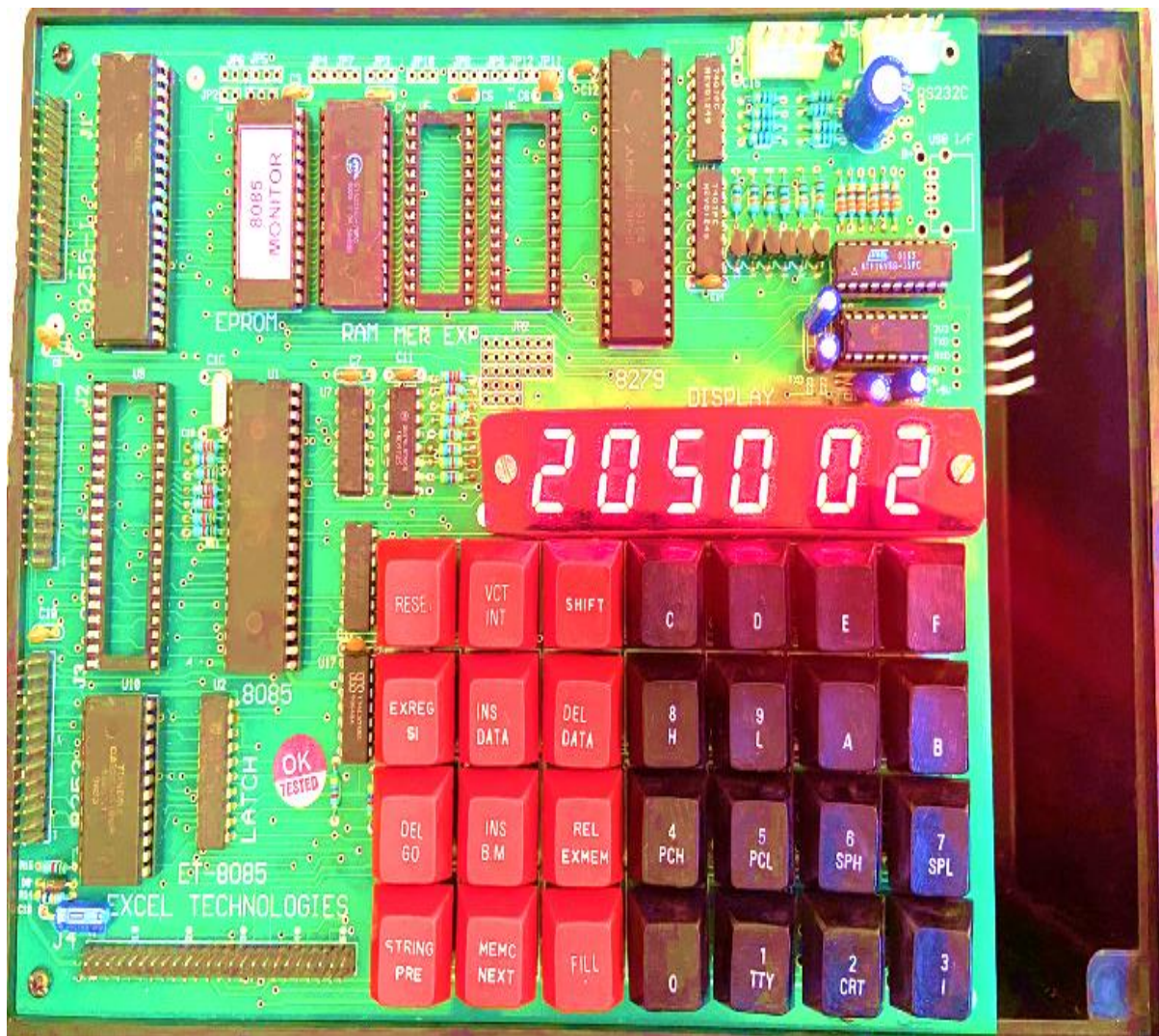
The steps are the identical for subtraction as they are for addition, but the value of a specific address and the process's outcome are altered in some way.

The modifications are depicted below:

In order to do the subtraction, I have entered the number 90 in address 2004.



The final execution of the program is depicted in this image, along with the address where each value is stored.



I've chosen two numbers for this subtraction process: 8 and 6. These numbers help to carry out the subtract and produce the desired result, which is 2.

Conclusion

In conclusion, the 8085 microprocessor assembly language program for adding and subtracting two 8-bit numbers serves as a useful illustration of how to program a microprocessor in assembly language. The program illustrates the fundamental concepts of entering input into registers, carrying out calculations, and saving the outcome in memory.

Students learning assembly language programming who also wish to understand microprocessors more thoroughly should start with this program. It offers a strong foundation for more difficult programming tasks, like managing peripherals and interacting with other hardware.

In general, the 8085 Microprocessor is a flexible and popular microprocessor that has contributed significantly to the advancement of computer technology. Students can get insightful knowledge into the inner workings of computer systems and acquire the abilities necessary to excel in embedded systems development, control systems engineering, and other related professions by learning how to program this microprocessor in assembly language.

References

1. University Slides (Online)
(April 28, 2023)
2. Youtube(Online)
https://youtu.be/punYQ03UU_k
<https://youtu.be/F-btuGnVrfl>
(April 28, 2023)

Assignmentwise

Marking schema

Marking Criteria	Allocated Marks	Marks Given
Question 1:		
Task-1: AIM	5 marks	
Task-2: Components required	5 marks	
Task-3: Code verification screenshot	10 marks	
Task-4: Screenshot of all primary steps while doing the specific task.	10 marks	
Task- 5: Procedure	5 marks	
Task- 6: VIVA	10 marks	
Task-7 Conclusion	5 marks	
Total Marks (50)		
Comments:		