

## **AIM**

To perform basic arithmetic operations through an assembly language program in 8085

## **Apparatus Required**

Microprocessor 8085 Kit

## **Theory**

### **Algorithm :**

1. Load the first number from memory location C050 to the accumulator.
2. Move the content of the accumulator to register H.
3. Load the second number from memory location C051 to the accumulator.
4. Then add the content of register H and accumulator using “ADD” instruction and storing the result at C120
5. The carry generated is recovered using the “ADC” command and is stored at memory location C121
6. Stop

### **Program :**

<b>Memory location</b>	<b>Label</b>	<b>Hexcode</b>	<b>Mnemonics</b>	<b>Comments</b>
C000		3A	LDA C050H	$A \leftarrow [C050]$
C001		27		
C002		C0		
C003		67	MOV H,A	$H \leftarrow A$
C004		3A	LDA C051H	$A \leftarrow [C051]$
C005		51		
C006		C0		
C007		84	ADD H	$A \leftarrow A+H$
C008		6F	MOVL,A	$L \leftarrow A$
C009		3E	MVI A,00	$A \leftarrow 00$
C00A		00		
C00B		8F	ADC A	$A \leftarrow A+A+\text{carry}$
C00C		67	MOV H,A	$H \leftarrow A$
C00D		22	SHLD C120H	$H \rightarrow C121, L \rightarrow C120$
C00E		20		
C00F		C1		
C010		CF	HALT	Stop

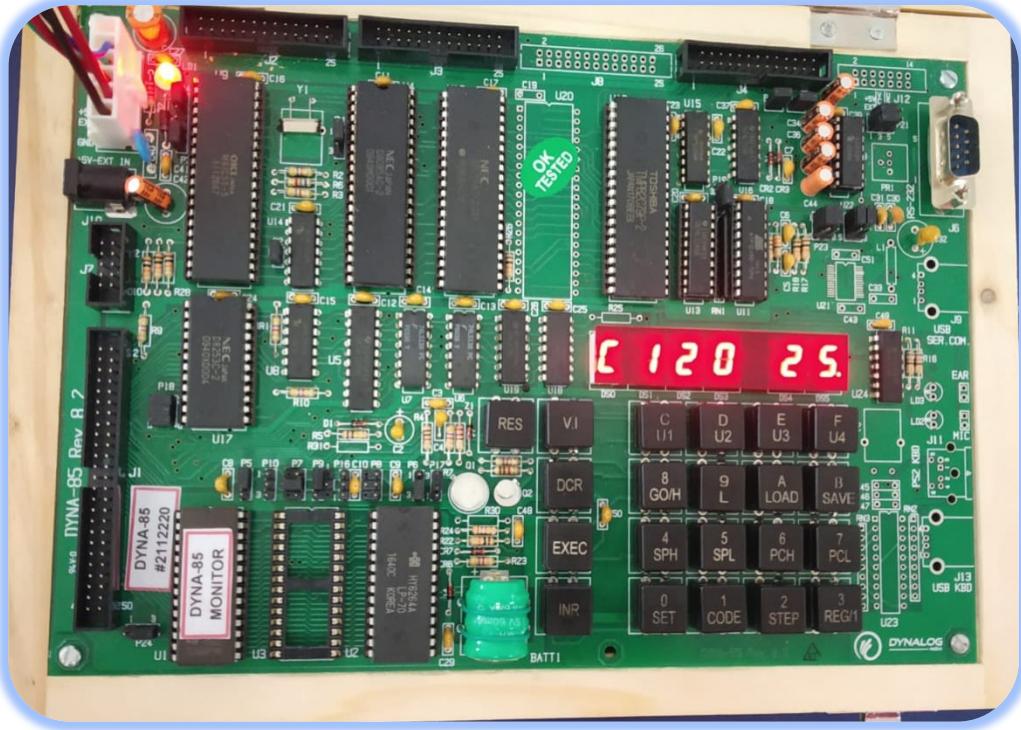
## **Procedure**

- 1) Open the Microprocessor kit
- 2) Write Hex code on Microprocessor kit from C000.

- 3) Store the value in respective memory location.
- 4) Execute the respective Hex code.
- 5) Check the result and note the result.

## Results and Observations

- C050→12 H
- C051→13 H
- C120:



- C121→00H

## Conclusion

We Conclude that we can perform basic arithmetic operation through an assembly language program in 8085

## AIM

To simulate COPY and PASTE operation through 8085 assembly program

## Apparatus Required

Microprocessor 8085 Kit

## Theory

### Algorithm :

1. Store 05 to the register L from C000.
2. Load the C500 to register BC.
3. Load the C600 to register DE.
4. Load the content from the memory location given by BC register to Accumulator.
5. Move the content of Accumulator to the address given by register DE.
6. Increment BC register.
7. Increment DE register.
8. Decrement L register.
9. Jump to address C008 till L is 0.
10. Stop

### Program :

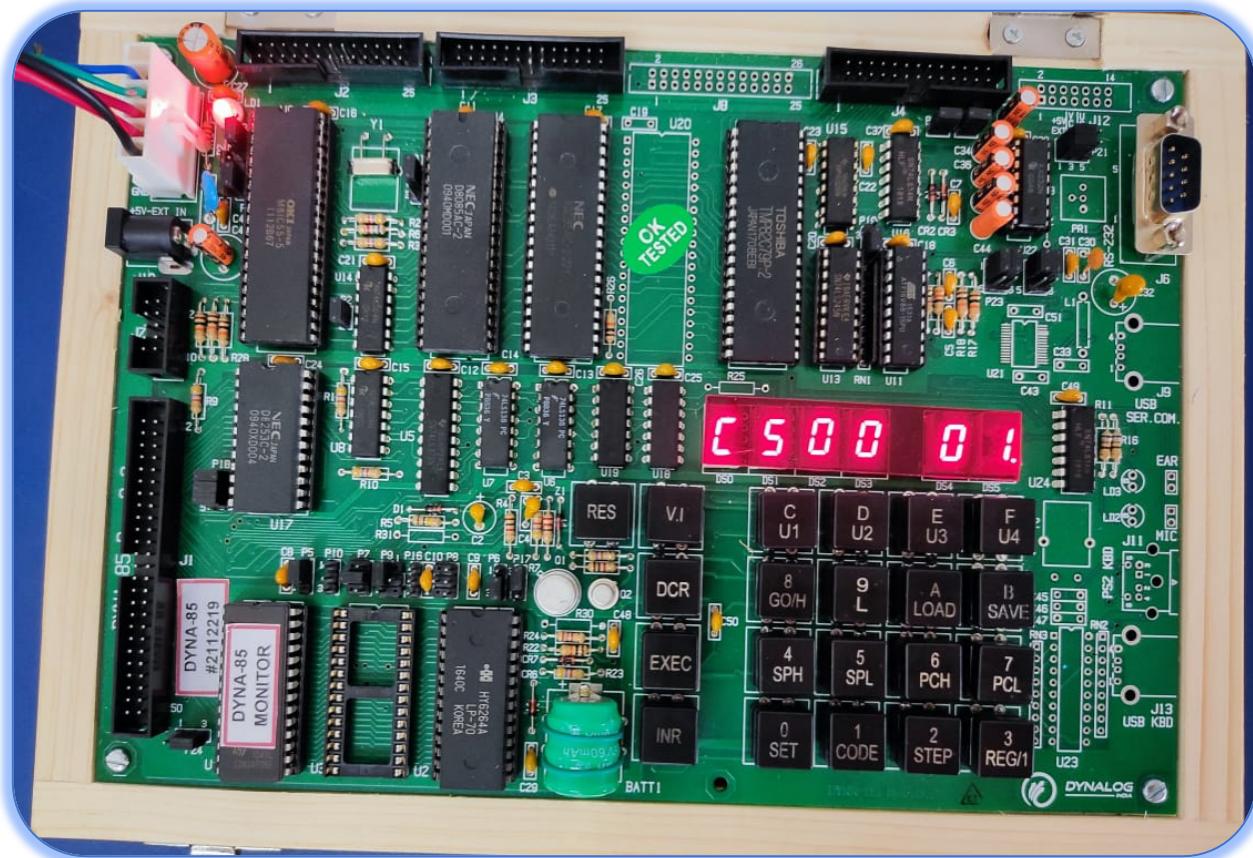
Memory location	Label	Hexcode	Mnemonics	Comments
C000		2E	MVI L, 05	L <--05
C001		05		
C002		01	LXI B,C500H	B<--C5,C<--00
C003		00		
C004		C5	LXI D,C600H	D<--C6,E<--00
C005		11		
C006		00	LDAX B	A<--[BC]
C007		C6		
C008	again	0A	STAX D	[DC] <--A
C009		12	INX B	C<--C+1
C00A		03	INX D	E<--E+1
C00B		13	DCR L	L<--L-1
C00D		C2	JNX C008	Jumps back to C008 till L is not equal to 00H
C00E		08		
C00F		C0		
C010		CF	HALT	Stop

## Procedure

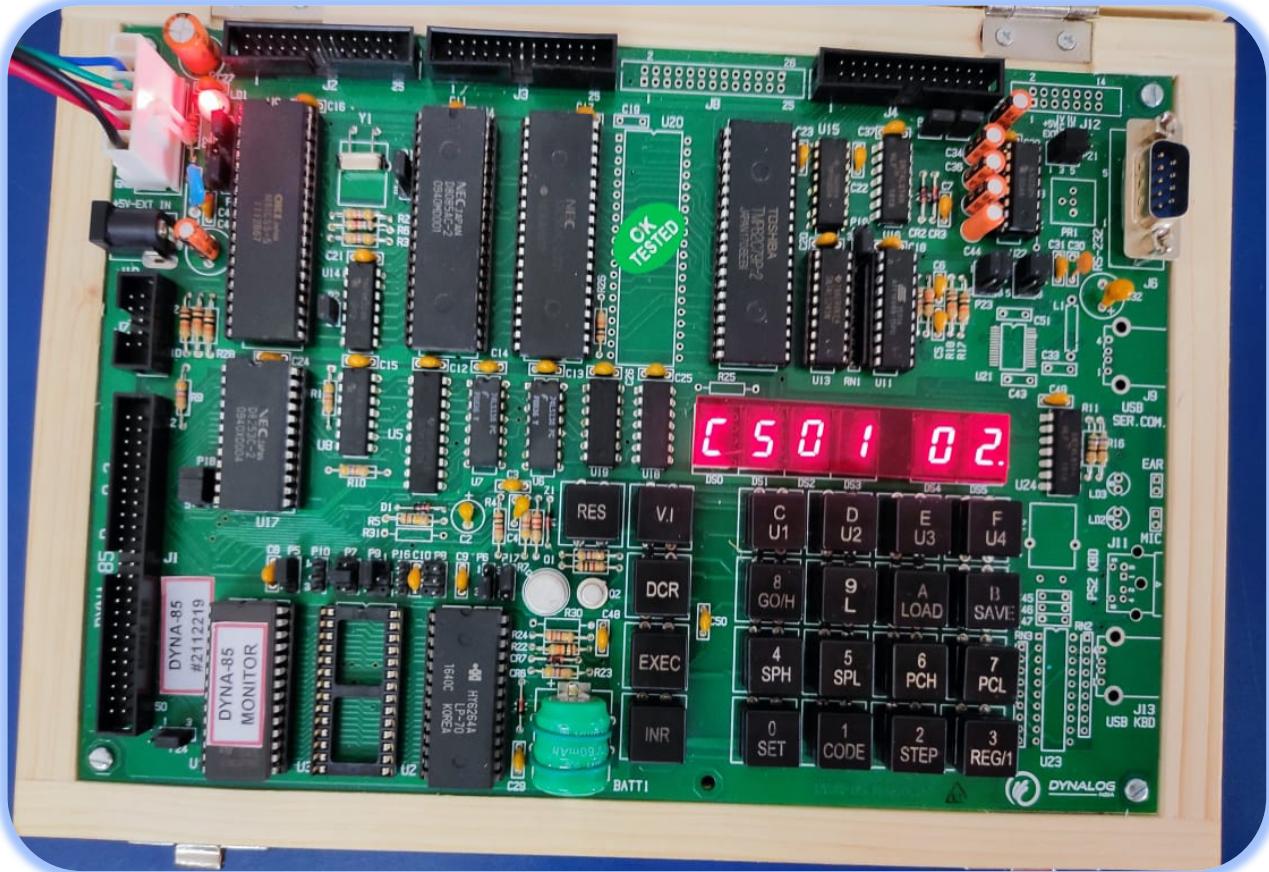
- 1) Open the Microprocessor kit
- 2) Write Hex code on the Microprocessor kit from C000.
- 3) Store the value in a respective memory location.
- 4) Execute the respective Hex code.
- 5) Check the result and note the result.

## Results and Observations

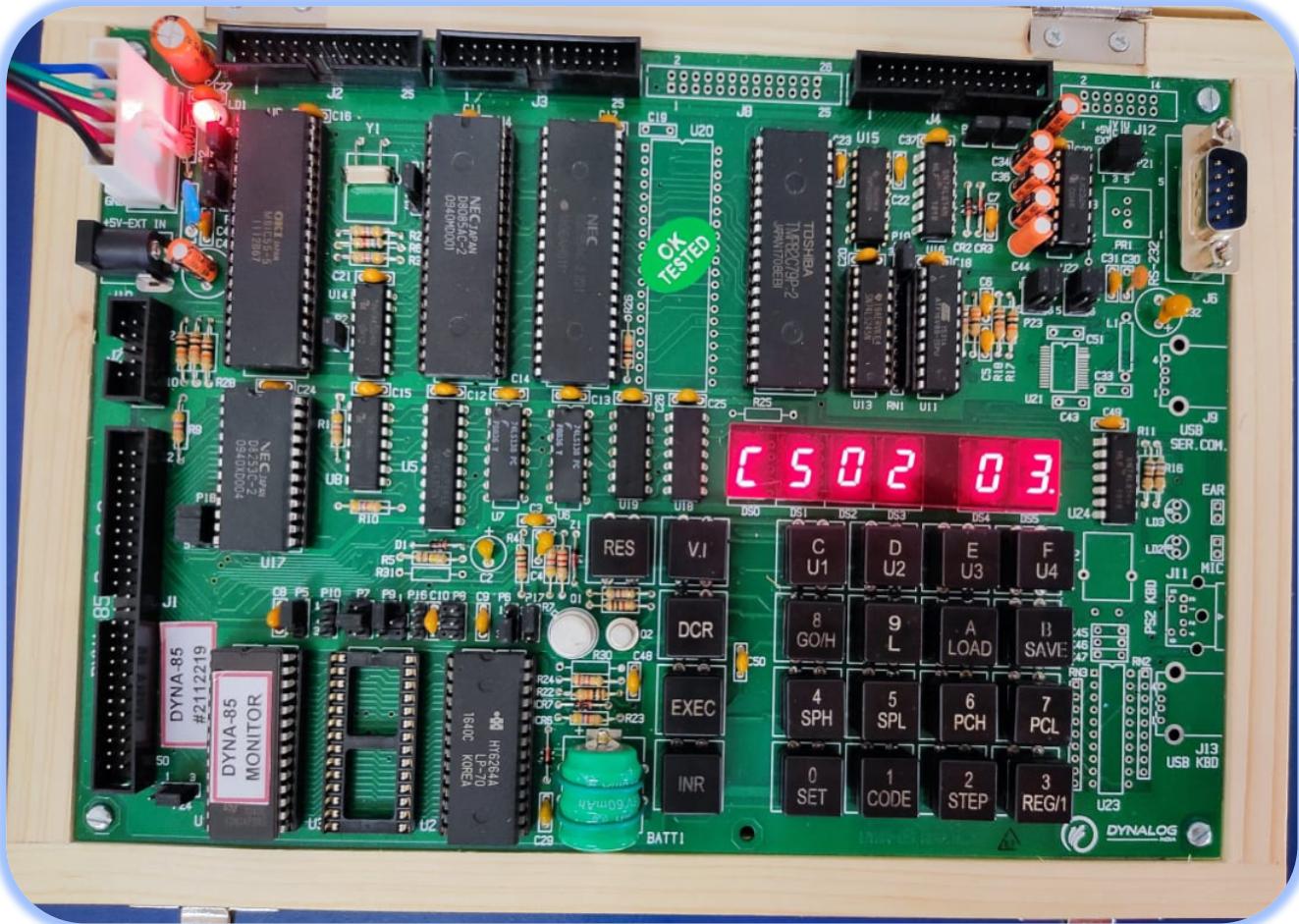
- Inputs
  - C500:



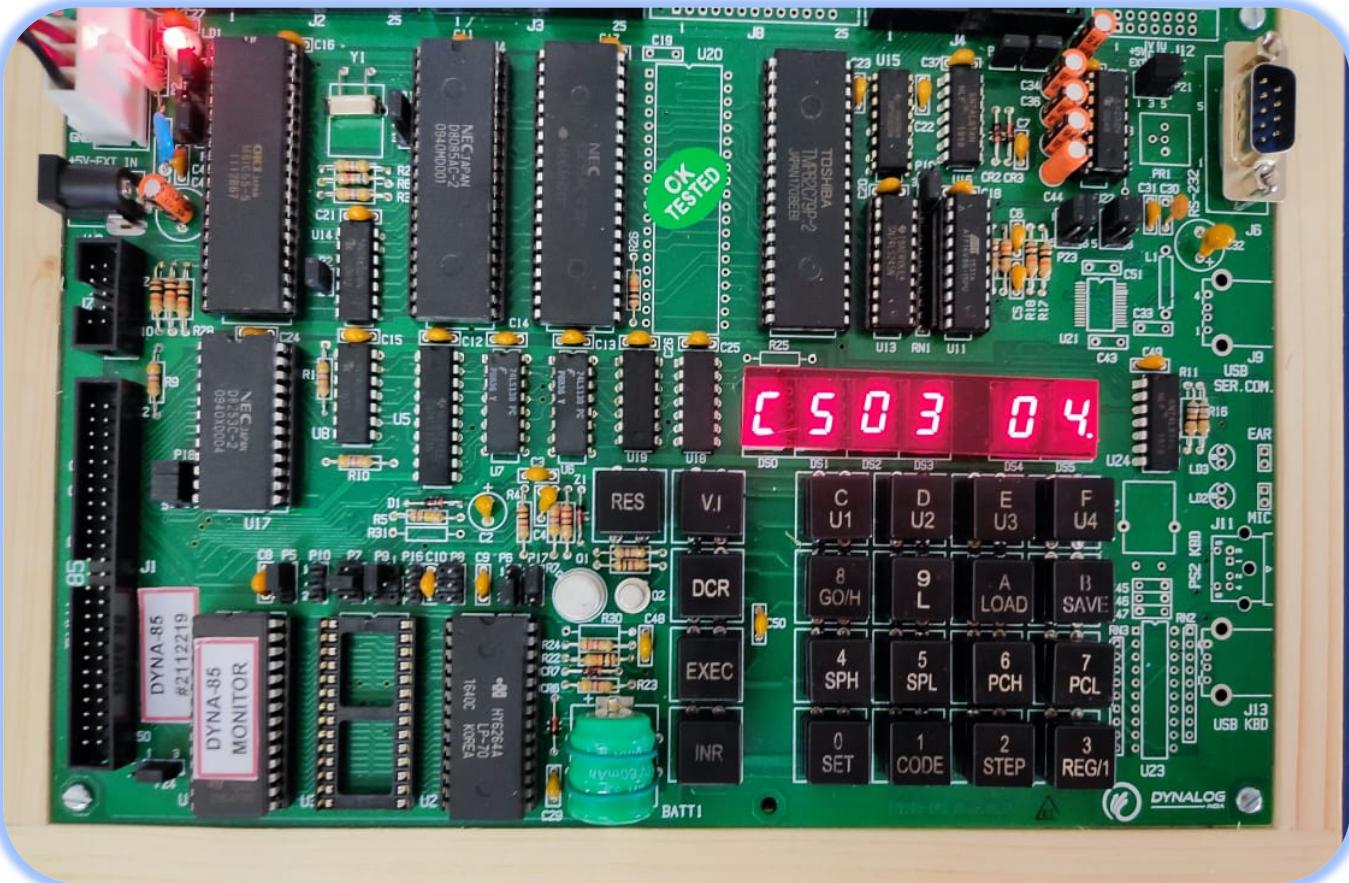
○ C501:



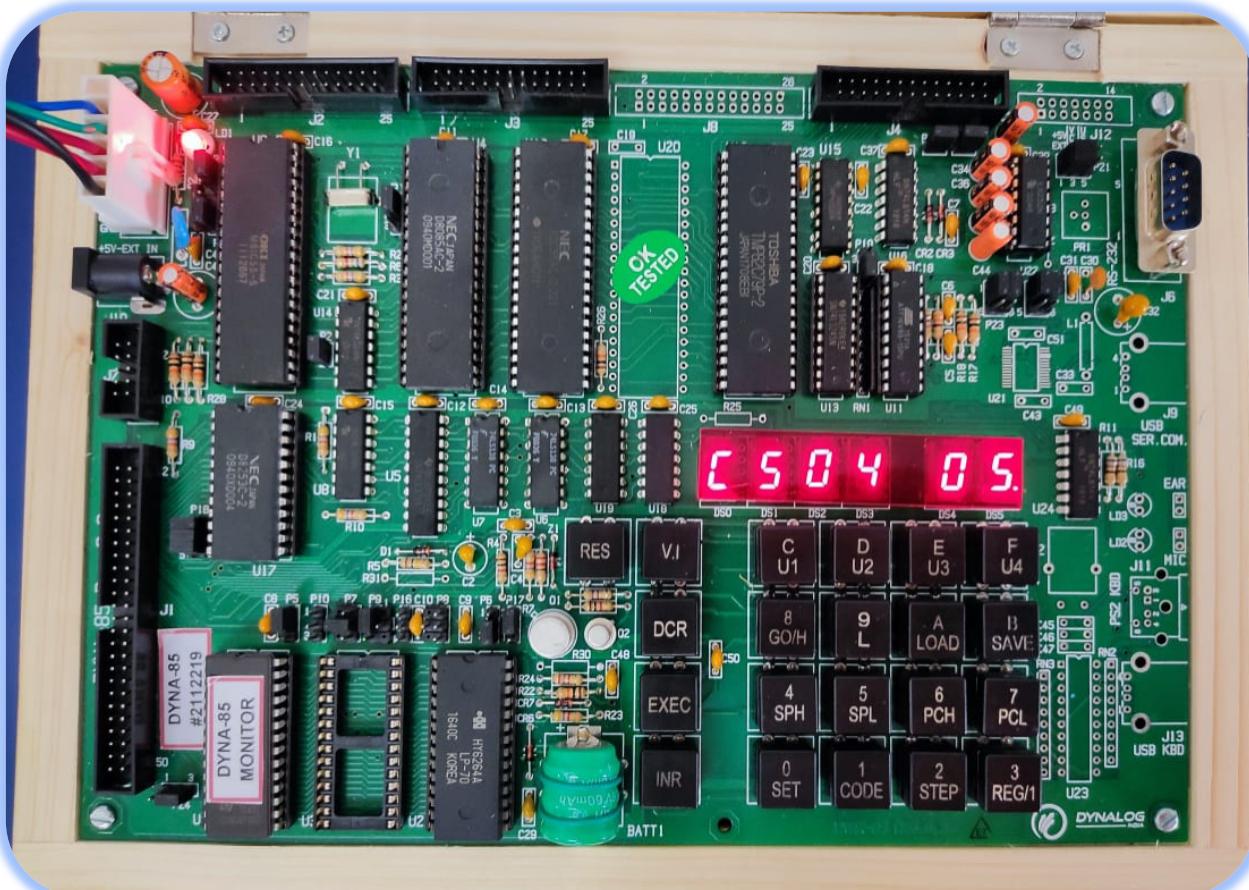
○ C502:



○ C503:



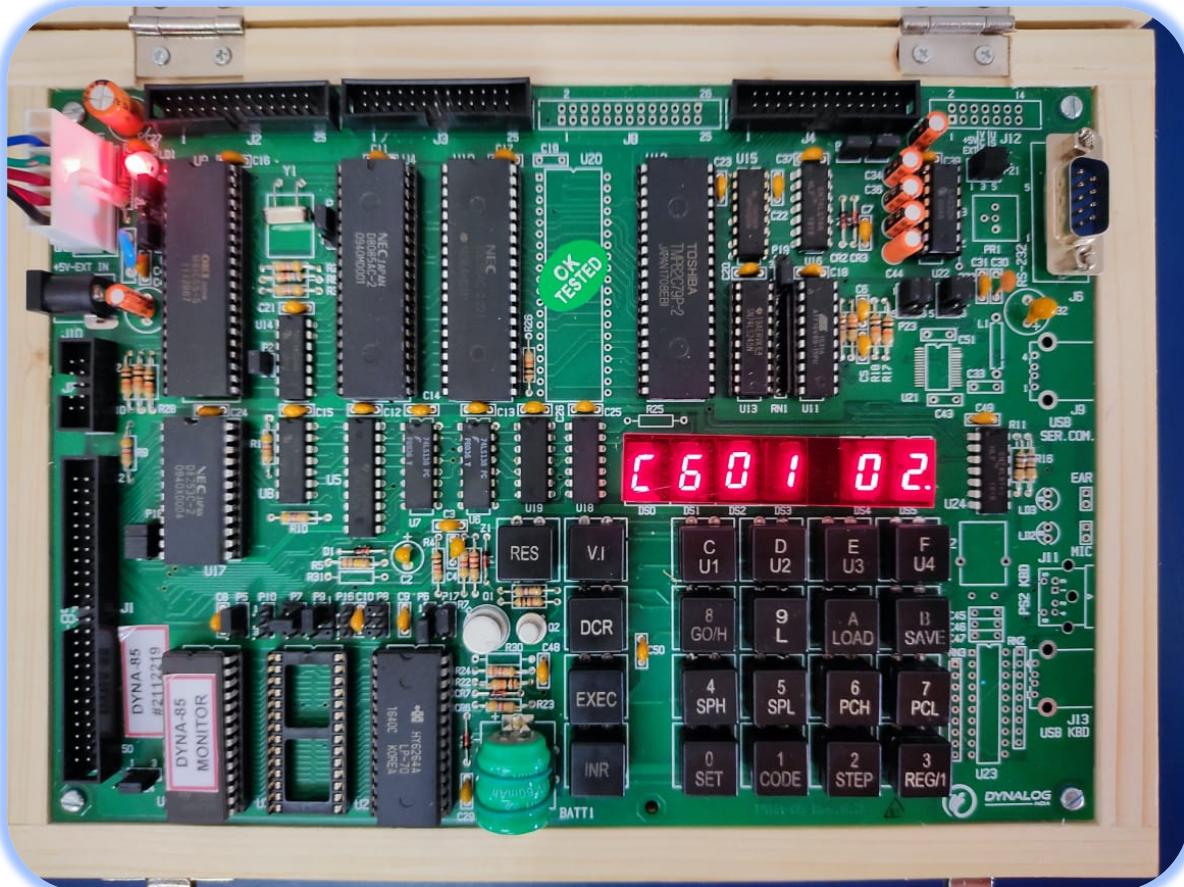
○ C504:



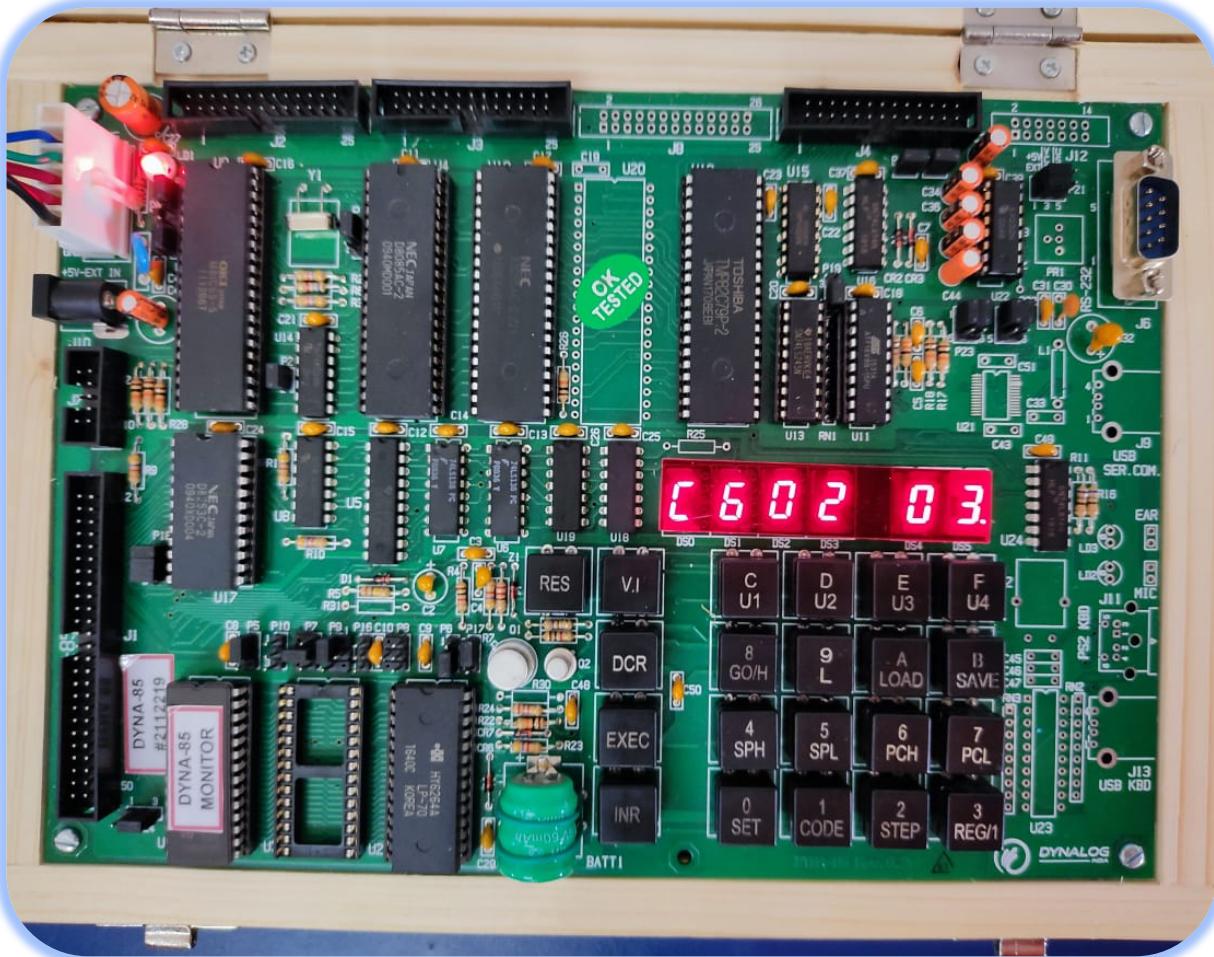
- Outputs:
  - C600:



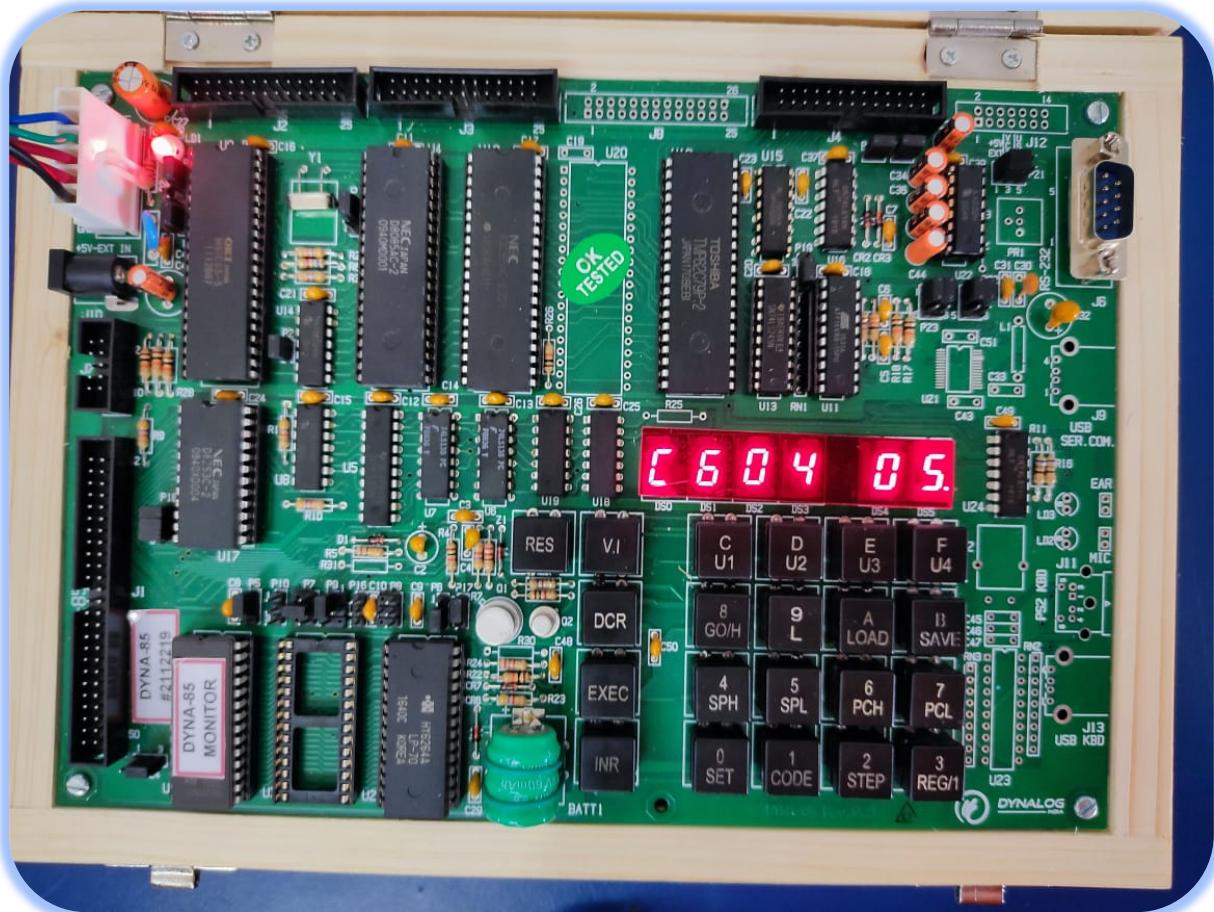
- C061:



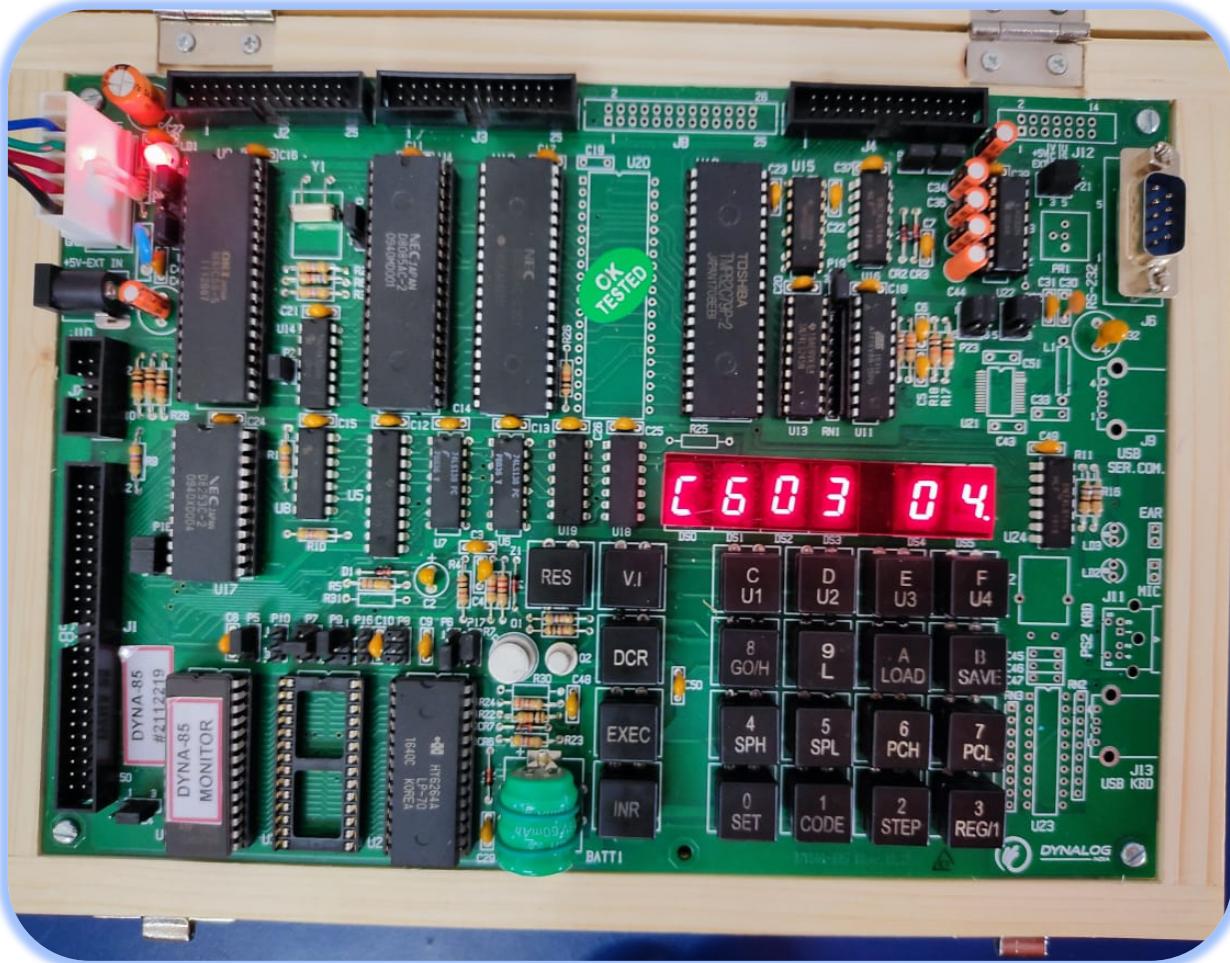
○ C062:



○ C063:



- C064:



## Conclusion

We conclude that we can simulate copy and paste operation through 8085 assembly program

## AIM

To write a program to add N elements stored in an array of 8 bit numbers (8085)

## Apparatus Required

Microprocessor 8085 Kit

## Theory

### Algorithm :

1. Load the content of C100H to the accumulator
2. Move the content of the accumulator to register B
3. Load the address D000H to the HL pair.
4. Store 00H in an accumulator.
5. Store 00H in the accumulator.
6. Add accumulator and content of HL pair and store in the accumulator.
7. Increment register L
8. Jump to step number 10 if carry is not generated.
9. Increment register C.
10. Decrement register B.
11. Jump back to step number 6 till B is not 0.
12. Store the content of the accumulator to memory location D100H
13. Move the content of register C to the accumulator.
14. Store the content of the accumulator to memory location D101H
15. Stop

### Program :

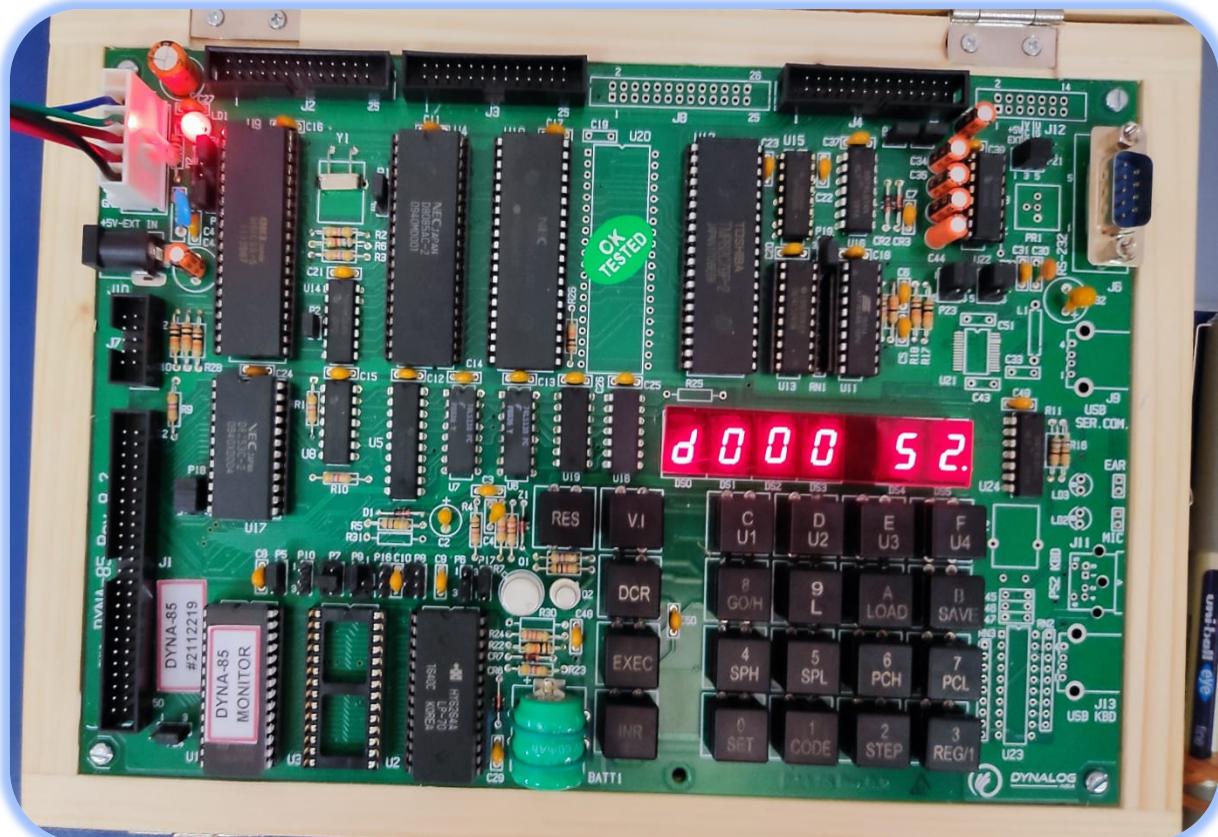
Memory location	Label	Hexcode	Mnemonics	Comments
C000		3A		
C001		00	LDA C100H	A <--- [C100H]
C002		C1		
C003		47	MOV B,A	B<---A
C004		21		
C005		00	LXI H,D000H	HL<---D000H
C006		D0		
C007		3E		
C008		00	MVI A,00	A<---00
C009		DE		
C00A		00	MVI C,00	C<---00
C00B	Again	86	ADD M	A<---A+[HL]
C00C		2C	INR L	L<---L+1
C00D		D2		
C00E		11	JNC C011H	Jumps to C011H if carry is not generated
C00F		C0		
C010		0C	INR C	C<---C+1
C011	Next	05	DCR B	B<---B-1
C012		C2		
C013		0B	JNZ C00BH	Jump back to C00BH till B is not equal to 0
C014		C0		
C015		32		
C016		00	STA D100H	[D100H]<---A
C017		D1		
C018		79	MOV A,C	A<---C
C019		32		
C01A		O1	STA D101H	[D101H]<---A
C01B		D1		
C01C		CF	HLT	Stop

## Procedure

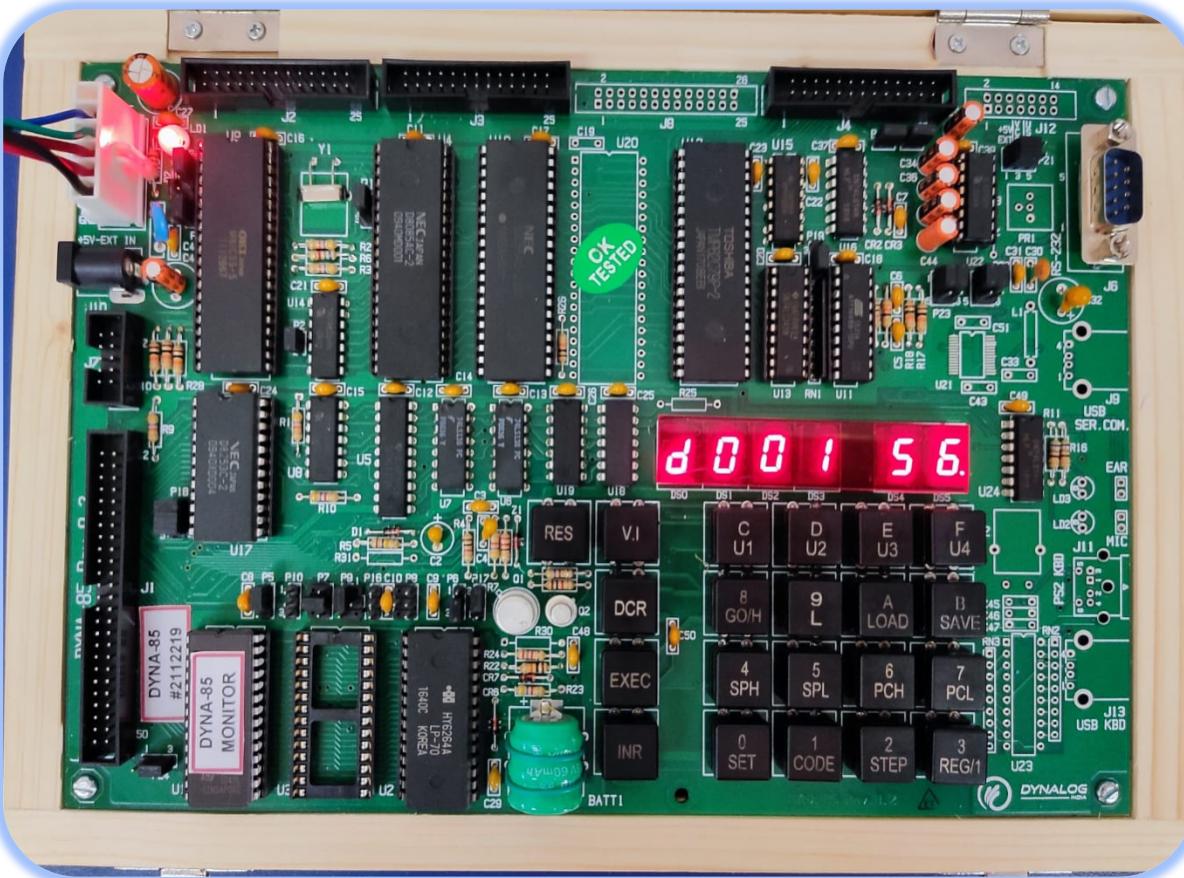
- 1) Open the Microprocessor kit
- 2) Write Hex code on Microprocessor kit from C000.
- 3) Store the value in respective memory location.
- 4) Execute the respective Hex code.
- 5) Check the result and note the result.

## Results and Observations

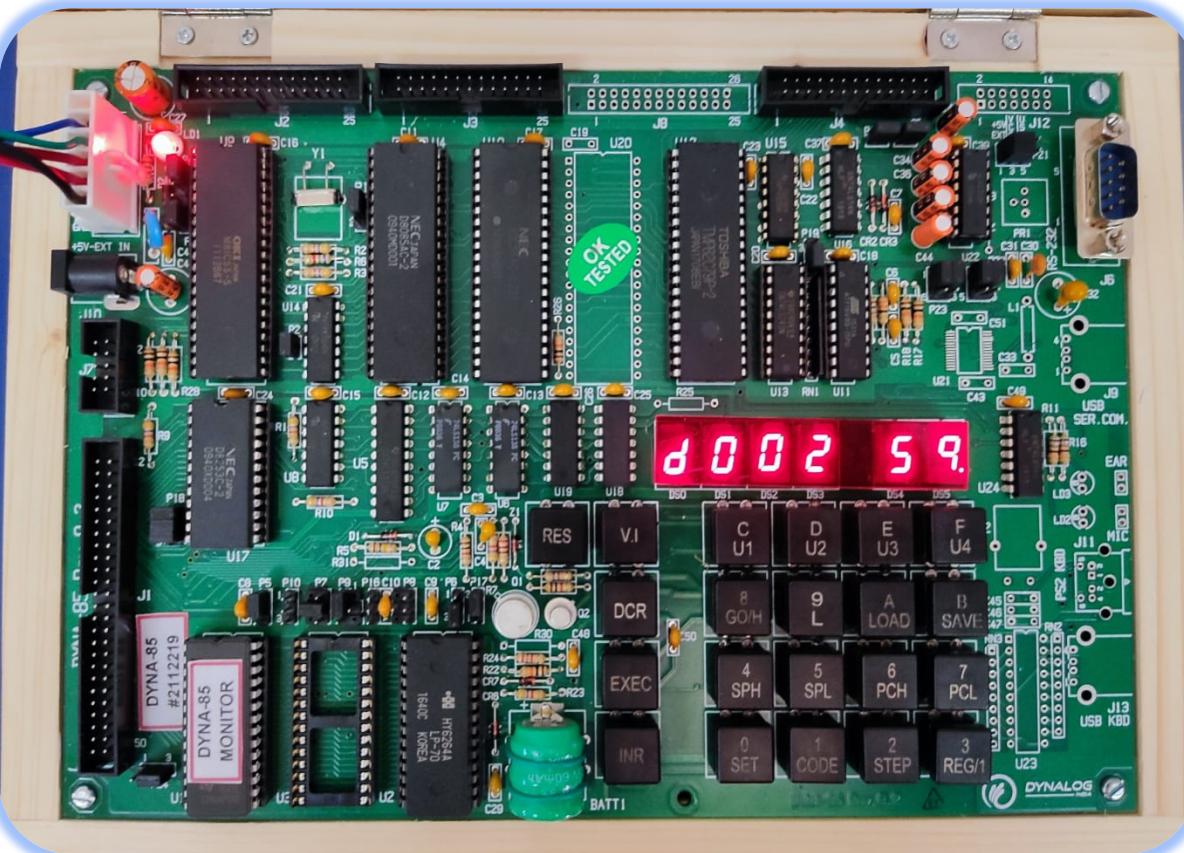
- Inputs:
  - D000H:



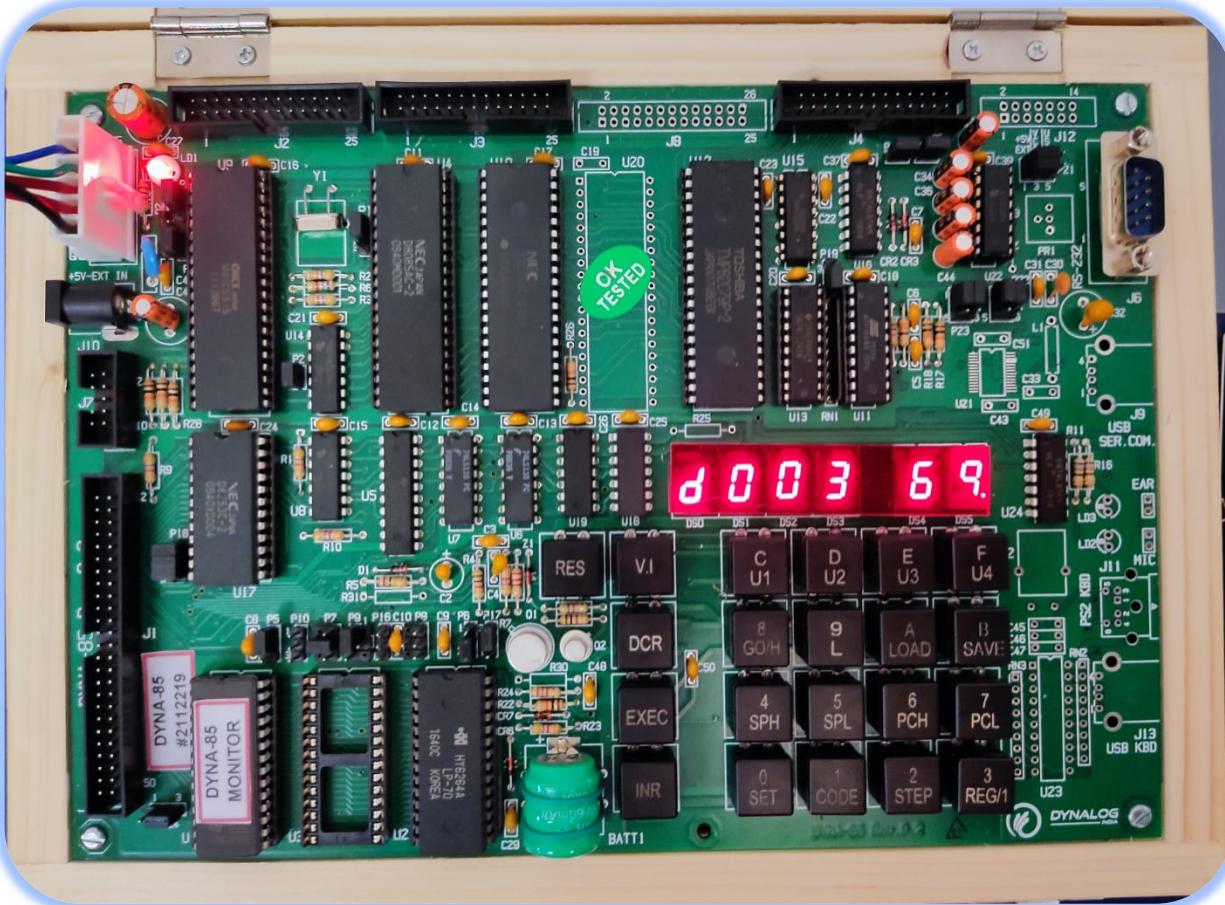
○ D001:



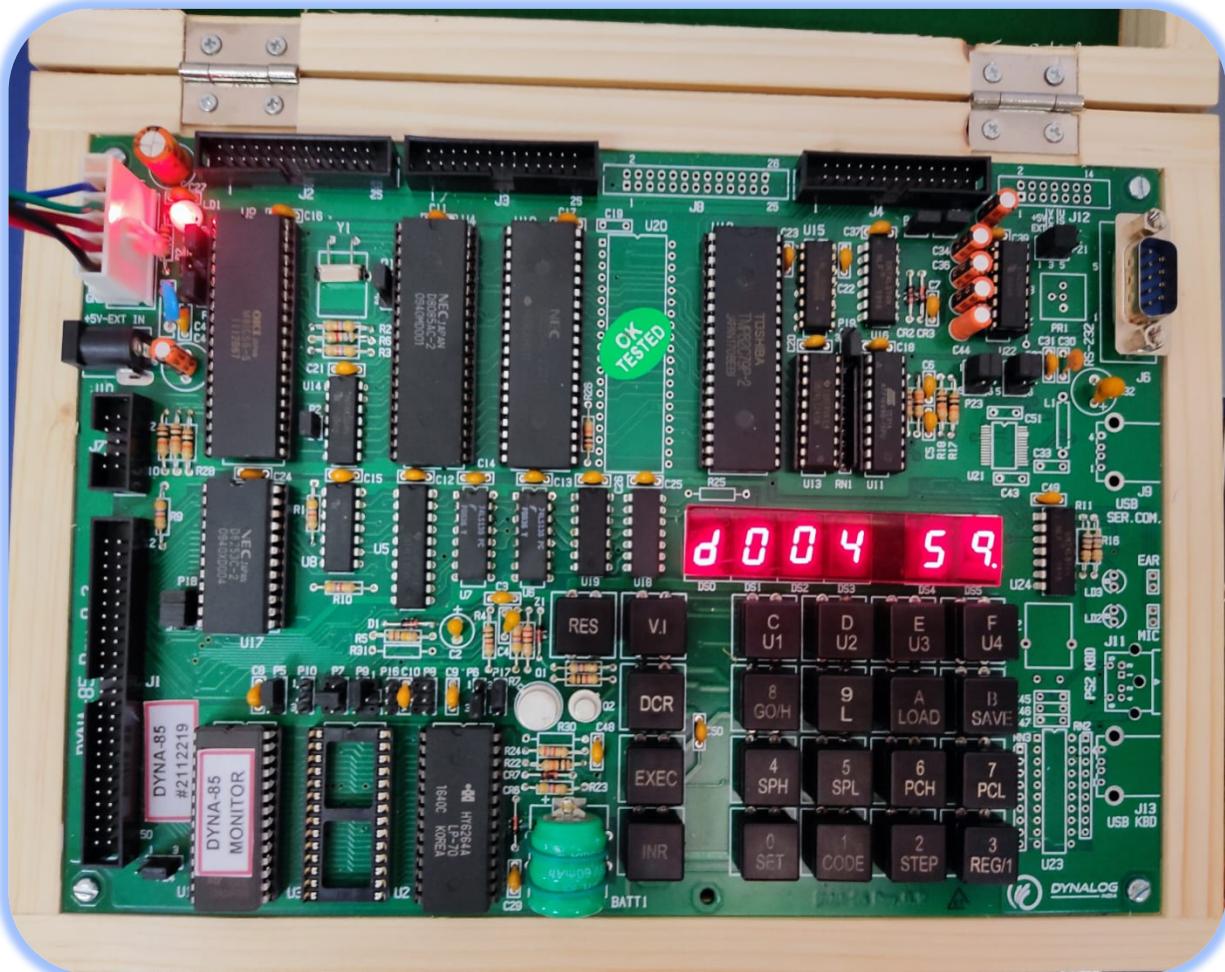
○ D002H:



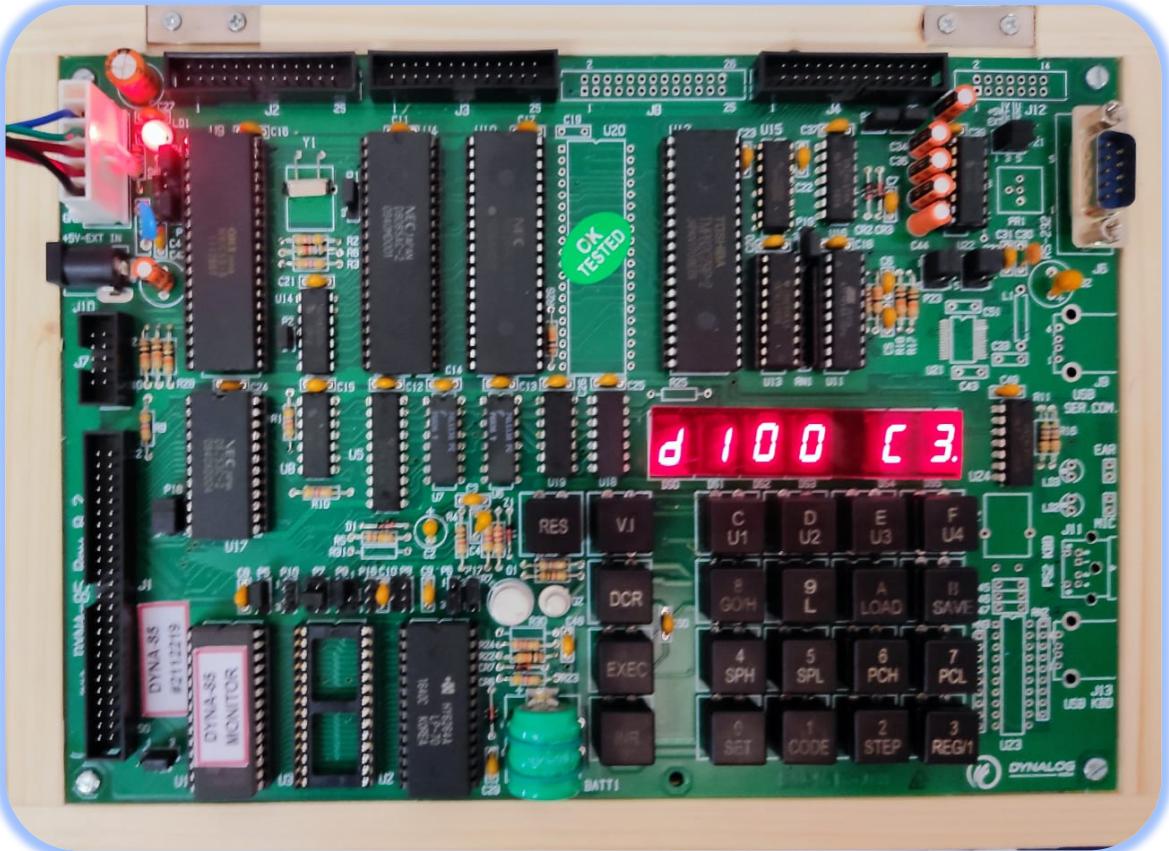
○ D003H:



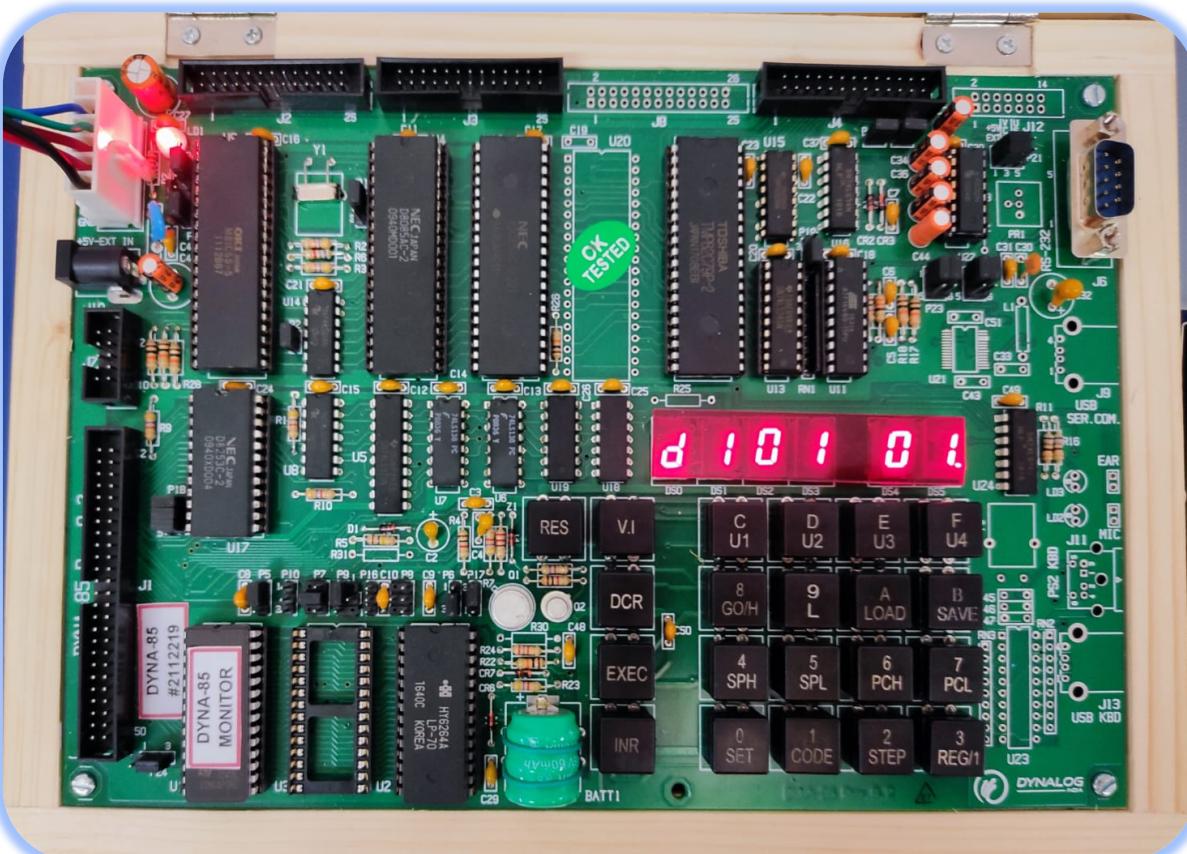
○ D004H:



- Outputs:
  - D100H(Sum):



- D101H(Carry):



## Conclusion

We can conclude that we can write a program to add N-elements stored in an array of 8-bit numbers (8085).