Microprocessor and Microcontroller Lab

(EE-311-G)

LABORATORY MANUAL

V - SEMESTER

Prepared by
Sh. Dayal Sati (A.P.)

BRCM College of Engineering and Technology

Department of Electrical Engineering
Bahal, Bhiwani - 127028

Microprocessor and Microcontroller Lab

Theory : 25
Class Work : 25
Total : 50

Course Code	LC-EE-311G		
Category	Program Core Course		
Course title	Microprocessor and Microcontroller Lab		
Scheme	L	T	P
	-	-	02

Notes:

(i) At least 10 experiments are to be performed by students in the semester.

(ii) At least 7 experiments should be performed from the list, remaining three experiments may either be performed from the above list or designed and set by the concerned institution as per the scope of the syllabus.

(iii) Group of students for practical should be 15 to 20 in number.

List of Experiments:

1. Write a program using 8085 and verify for :

a. Addition of two 8-bit numbers.

b. Addition of two 8-bit numbers (with carry).

2. Write a program using 8085 and verify for :

a. 8-bit subtraction (display borrow)

b. 16-bit subtraction (display borrow)

3. Write a program using 8085 for multiplication of two 8- bit numbers by repeated addition method. Check for minimum number of additions and test for typical data.

4. Write a program using 8085 for multiplication of two 8- bit numbers by bit rotation method and verify.

5. Write a program using 8086 for finding the square root of a given number and Verify.

6. Write a program using 8086 for copying 12 bytes of data from source to destination and verify.

7. Write a program using 8086 and verify for:

a. Finding the largest number from an array.

b. Finding the smallest number from an array.

8. Write a program using 8086 for arranging an array of numbers in descendingorder and verify.

9. Write a program using 8086 for arranging an array of numbers in ascending order and verify.

10. Write a program to interface a two digit number using seven-segment LEDs. Use 8085/8086 microprocessor and 8255 PPI.

11. Write a program to control the operation of stepper motor using 8085/8086microprocessor and 8255 PPI.

12. To study implementation & interfacing of Display devices Like LCD, LED Bargraph& seven segment display with Microcontroller 8051/AT89C51

13. To study implementation & interfacing of Different motors like stepper motor,DC motor & servo Motors.

14. Write an ALP for temperature & pressure measurement

Write a program to interface a graphical LCD with 89C51

LIST OF EXEPRIMENTS

S.No	Name of Experiments	Page No
1.	Introduction of microprocessor 8085 trainer kit – 85AD	
2.	Write an ALP to perform the addition of two 8 bit numbers.	
3.	Write an ALP to perform the subtraction of two 8 bit numbers.	
4.	Write an ALP to perform the addition with carry of two 8 bit numbers.	
5.	Write an ALP to perform the subtraction with barrow of two 8 bit numbers.	
6.	Write an ALP to perform the addition of two BCD numbers.	
7.	Write an ALP to perform the subtraction of two BCD numbers.	
8.	Write an ALP to perform the multiplication of two 8 bit numbers by repeated addition method.	
9.	Write an ALP to perform the multiplication of two 8 bit numbers by bit Rotation method.	
10.	Write an ALP to perform the division of two 8 bit numbers by repeated addition method.	
11.	Write an ALP to perform the division of two 8 bit numbers by bit rotation method.	
12.	Write an ALP to find the square of given numbers in array.	
13.	Write an ALP to find largest number in an array.	
14.	Study of 8086 microprocessor kit	
15.	Write an ALP to perform addition of two 16 bit numbers.	

EXPERIMENT NO.1

AIM:

To study about introduction of microprocessor 8085 trainer kit - 85AD.

APPARATUS REQUIRED:

8085 - Microprocessor kit.

THEORY:

The system has got 8085 as the Central Processing Unit. The clock frequency for the system is 3.0MHz and is generated from a crystal of 6.14MHz. 8085 has got 8 bit data lines and 16 bit address lines. The lower 8 address lines and 8 bit data lines are multiplexed. Since the lower 8 address bits appear on the bus during the first clock cycle of a machine cycle and the 8 bit data appears on the bus during the 2nd and 3rd clock cycle, it becomes necessary to latch the lower 8 address bits during the first clock cycle so that the 16 bit address remains available in subsequent cycles. This is achieved using a latch 74LS373.

The training kit which we are going to use in this lab is STUDENT-85AD which communicates with the outside world through a general purpose IBM PC Compatible ASCII keyboard and 16x2 Liquid Crystal Display (LCD). The kit also has the capability of interfacing with CRT terminal through the interface provided on the board.

The on board resident system monitor software is very powerful and provides various software utilities. The kit provides support for powerful software commands like INSERT, DELETE, BLOCK MOVE, RELOCATE, STRING, FILL and MEMORY COMPARE etc. The kit is configured around the internationally adopted standard STD bus which is most popular bus for process control and real time applications. All the address, data and control lines are available at the edge connector. The kit is fully expandable for any kind of application.

MEMORY:

8085 kit provides 8/32K bytes of RAM using 6264/62256 chip and 8K bytes of EPROM for monitor. There is one memory space provided on kit. This one space can be defined any address slots from 8000 - DFFF depending upon the size of the memory chip to be used. Total onboard memory can be extended to 64K bytes.

I/O DEVICES

The various I/O chips used in STUDENT-85AD microprocessor kit are 8255, 8253 & 8155. The functional role of all these chips is given below:

8255(Programmable Peripheral Interface)

8255 is a programmable peripheral interface (PPI) designed to use with 8085 Microprocessor. This basically acts as a general purpose I/O device to interface peripheral equipments to the system bus. It is not necessary to have an external logic to interface with peripheral devices since the functional configuration of 8255 is programmed by the system software. It has got three Input/Output ports of 8 lines each (PORT-A, PORT-B & PORT-C). Port C can be divided into two ports of 4 lines each named as Port C upper and Port C lower. Any Input output combination of Port A, Port B, Port C upper and lower) can be defined using the appropriate software commands. The kit provides 24 Input/output ports using 8255 chips.

8253(Programmable Internal Timer)

This chip is a programmable interval Timer/Counter and can be used for the generation of accurate time delays under software control. Various other functions that can be implemented with this chip are programmable rate generator, Even Counter, Binary rate Multiplier, Real Time Clock etc. This chip has got three in dependent 16 bit counters each having a count rate of up to 2 KHz. The first Timer/Counter (i.e. Counter 0) is being used for Single Step operation. However, its connection are also brought at connector space C4. For single step operation CLKO signal of Counter 0 is getting a clock frequency of 1.535 MHz. The counter 1 is. used to generate clock for 8251. Counter 1 & Counter 2 are free for the user. Clock for the CLK1, CLK2 is to be given externally.

8155 (Programmable I/O Port & Timer Interface) –Optional

8155 is a programmable I/O ports and timer interface designed to use with 8085 Microprocessor. The 8155 includes 256 bytes of R/W memory, three I/O ports and a Timer. This basically acts as a general purpose I/O device to interface peripheral equipments to the system bus. It is not necessary to have an external logic to interface with peripheral devices since the functional configuration of 8155 is programmed by the system software. It has got two 8-bit parallel I/O port (Port-A, Port-B) and one 6-bit (Port-C). Ports A & B also can be programmed in the handshake mode, each port using three signals as handshake signals from Port-C. The timer is a 14 bit down counter and has four modes.

List of ASCII Keyboard Commands

Sr. No.	*Command*	*Description*	*Command Syntax*
1.	M	Examine/ Modify Memory	[M]<ADDRESS>[DATA] [DATA]….[,]
2.	E	Enter a memory block	[E]<ADDRESS>[DATA] [DATA]….[$]
3.	R	Examine/ Modify register	[R] <REG. IDENTIFIER> [$]
4.	S	Single Step	[S] <Starting Address>[,]
5.	G	Go	[G] < Starting Address>[$]
6.	B	Block Move	[B]< Starting Address of source>[,]< End Address of source >[,]< Starting Address of destination>[$]
7.	I	Insert	[I]< Starting Address of the program>[,]< End Address of the program >[,]<Address from where the byte or bytes are to be entered>[,]<No. of bytes>[,][DATA][$]

8.	D	Delete	[D]<Starting Address of the program>[,]<Ending address of the program>[,]<starting address from where the bytes are to be deleted>[,]<Ending address till where the bytes are to be deleted>[$]
9.	N	Insert Data	[N]< Starting Address of the program/ data area >[,] <Ending Address of the program>[,]< Starting Address at which the bytes are to be entered>[,]<No. of bytes>[,][DATA][.][$]
10.	O	Delete Data	[O]< Starting Address of the program or data area>[,]< End Address of the program/ data area >[,]< Starting Address from where the deletion should start>[,]<End address till where bytes are to be deleted>[$]
11.	F	Fill	[F]< Starting Address of program/ data area>[,]<End Address>[,]<Constant to be filled>[$]
12.	H	Relocate	[H]< Starting Address of the program>[,]<End Address of the program>[,]<Destination Address>[$]
13.	J	Memory Compare	[J]< Starting Address of the first block>[,]<End Address of thefirst block>[,]< Starting Address of second block>[$]
14.	K	String	[K]< Starting Address of the program>[,]<End address of the program>[,]<Address of the location at which first byte of the string lies>[,]<Address of the location at which last byte of the string lies>[$]

Table: 1 Details of the commands used in STUDENT-85AD

Sr. No.	*Register Identifier*	*Register Name*
1.	A	Register A or accumulator
2.	B	Register B
3.	C	Register C
4.	D	Register D
5.	E	Register E
6.	F	Register F
7.	I	Interrupt Mask Register
8.	H	Register H
9.	L	Register L
10.	S	Stack Pointer MSB
11.	P	Program Counter MSB

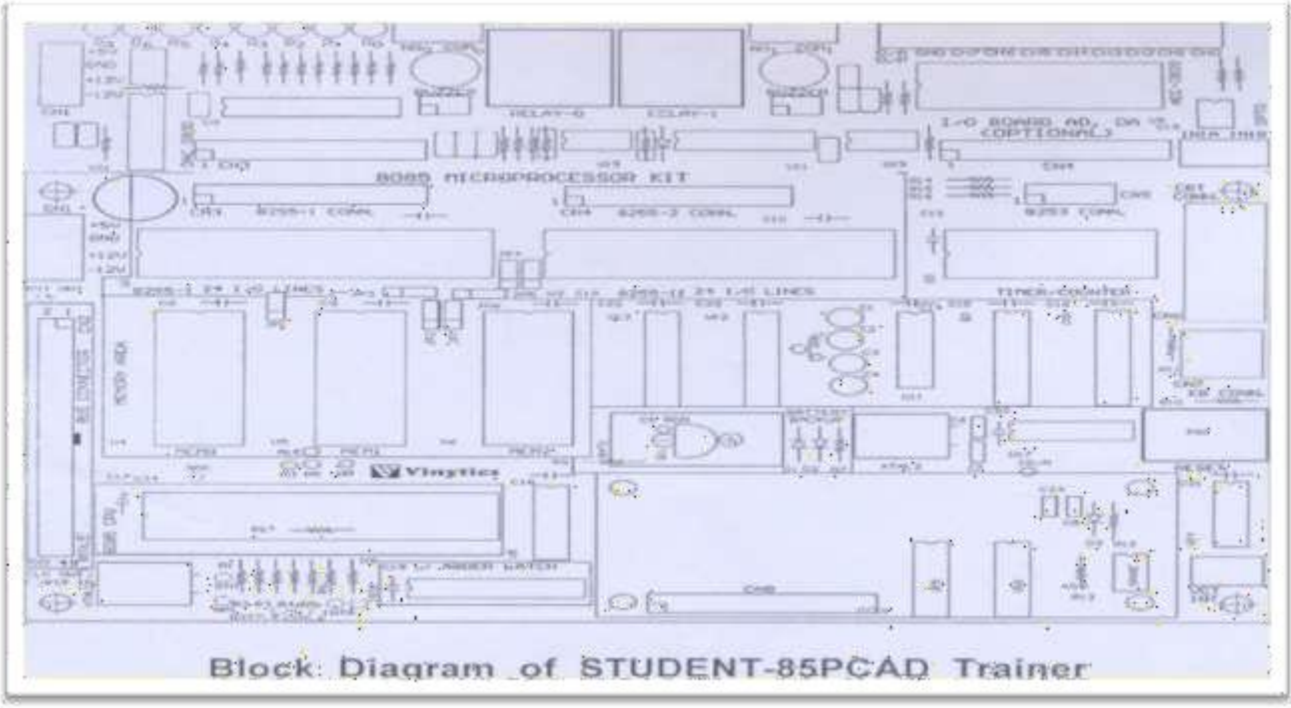
Table: 2 Details of the Register Identities used in STUDENT-85AD

<u>Modes of Operation:</u>

In STUDENT-85AD there is an onboard facility of assembler/ disassemble. On pressing ‘1’ key, STUDENT-85AD comes in assembler/ disassemble mode depending upon the next key you press i.e.:

‘A’- Assembler mode

‘C’- Disassembler mode



Block Diagram of STUDENT-85PCAD Trainer

Assembler Mode:

On pressing the key 'A' STUDENT-85AD comes into the assembler mode. As soon as 'A' is pressed, kit asks RAM address. This will be the starting address of the program to be entered. After entering the starting address, press <CR> key, it displays the entered starting address in the upper line of the LCD screen. Now it waits for the mnemonics entry.

One can enter all the valid mnemonics of 8085 and the Pseudo commands. If the entered alphabets do not form a valid mnemonics or a Pseudo command, the carriage goes to same line and prints the address of the previous line. Hence the entry of the wrong mnemonic is indicated by giving the same line to the user.

Entry of a space completes one field of entry: and processing of that field is done immediately by the command. By field, we mean, mnemonic as one field, operand or label as another field.

Using this mode one can write or feed the program using assembly language mnemonics.

Format:

[1]<ENTER> [A] <ENTER>

The display will show

```
RAM ADRR:
```

Type the desired RAM ADDRESS e.g. 2000, the display looks like figure below

```
RAM ADRR:  2000
```

(Note: RAM Address is always a16-bit hexadecimal number.)

Press <ENTER> and the display will show the RAM Address blank against it.

```
2000:
```

One can type the command mnemonics in blank space. After that on pressing ENTER or SPACE key, the RAM Address will be automatically incremented. In case of invalid mnemonics the monitor software will erase the mnemonic and will remain on the same RAM Address until the command typed is a valid mnemonic. Then the display will appear like figure below:

```
2000: MOV B, A
```

In this manner one can type all a whole program which automatically gets stored in the RAM of the kit and RUN the program using 'GO' command thereafter.

DISASSEMBLER MODE:

Disassembler mode can be selected by pressing the '1' and then 'C'. This command disassembles the program as specified by the STARTING address and END address. In case one wants to proceed further, press <CR> key, otherwise <Esc> key will exit from the disassemble mode.

MEMORY MAPPING:

STUDENT-85AD kit provides 8/32 KB of RAM and 16 KB of EPROM. The total onboard memory can be expanded to 64 KB. For the system operation the monitor should start from address 0000H. A minimum of 8KB RAM should be there on the board with starting address as 2000H.

INPUT/ OUTPUT MAPPING:

Device	Active range Port Addresses	Port Numbers	Selected Device
8255-I	00-07		PPI
		00 and 04	Port-A
		01 and 05	Port-B
		02 and 06	Port-C
		03 and 07	Control Word
8255-II	08-0B		PPI
		08	Port-A
		09	Port-B
		0A	Port-C
		0B	Control Word
8253	10-17		PIT
		10 and 14	Counter 0
		11 and 15	Counter 1
		12 and 16	Counter 2
		13 and 17	Control Word
LCD	38-3F		

Developing/ Debugging Software:

STUDENT-85AD kit provides software features like Relocate, String, Insert, Delete, Assembler, Disassembler, Programming etc. which find extensive application in developing/ debugging software. The various steps involved in developing software are:

1. Define the problem in the form of a flow chart.

2. Write the program in Assembly Language of 8085.

3. Assemble the program through Assembler command.

4. Enter the program in RAM area and RUN it.

It is likely that the program may not run in one shot because some mistakes can be there in it. The process of finding these mistakes and removing them is called the debugging of the program.

One way of entering the program is in HEX code of the mnemonics and the other way is through assembler. In assembler mode you can write the program in mnemonics form and inspect the disassembly form with its HEX code.

One way of finding the mistakes in the program is to run the program in single instruction mode and after each step compare what the program is doing and what is it supposed to do. In the process of this one might have to examine the contents of the memory locations or the content of internal registers after the execution of each instruction.

During this process of debugging, at some time user might just like to examine the status of the program at a particular point. If this point is near the beginning of the program, one can reach this point by single instruction facility. But if the point is quite far from the beginning of the program, it is time saving to make use of BREAK POINT facility. For this introduce a RST5 instruction (EF) at the point to be examined and run the program at full speed using ‘GO’ command. When during the execution of the program, this instruction is encountered; the control of the processor is transferred to the monitor. The monitor saves the user registers and displays a sign ‘STUDENT-85AD’ on the LCD screen. No one can examine the status of any memory location or any internal register. One can change the content of memory location or register if necessary.

Sometimes while debugging user may find that certain instructions are to be added to the program or to be deleted from the program. The program written for one memory area can be made operative for some other area using the RELOCATE command.

Sometimes it is required to execute the program in single cycle mode. For this one can make use of single cycle facility (optional) on the board of STUDENT-85AD kit.

EXPERIMENT NO. 2

Aim

Write a well-documented program using 8085 for addition of two 8-bit numbers.

Apparatus

8085 microprocessor kit (STUDENT-85AD), ASCII (PS-2) keyboard.

Theory :-

2501H is the address of memory location for the 1st number. 2nd number is stored at next memory location. By the virtue of the program given below we first move the 1st number into accumulator register then add the 2nd number to this and use the next memory location i.e. 2503h to store the result. The program given below is self explanatory.

Program:-

MEMORY	MACHINE	MNEMONICS	OPERANDS	COMMENTS
2000	21,01,25	LXI	H,2501	Get address of 1st no. in HL pair
2003	7E	MOV	A,M	Move 1st no. in accumulator
2004	23	INX	H	HL points the address 2502H
2005	86	ADD	M	Add the 2nd no.
2006	23	INX H		HL points 2503H
2007	77	MOV	M,A	Store result in 2503H.
2008	CF	RST 1		Terminate

Procedure:-

(1) Connect the kit to the power supply.

(2) Connect the PS-2 keyboard to the PS-2 (Female) connector CN-7 of the kit.

(3) Switch ON the kit.

(4) LCD displays “STUDENT-85”, if not so check power connections.

(5) If backlight is glowing and display shows something else press the RESET key provided on the kit. If still the persists contact Lab staff.

(6) To Feed the program and data in the RAM, for machine language follow the steps 7 to 9 and for assembly language programming steps 10 to 14.

(7) Press [M]<Starting Address >[ENTER]. Starting address here is 2000H, so syntax will be [M]<2000>[ENTER].

(8) The LCD display will show the address and its content. Modify the content according to the program using keyboard then press [ENTER] or [SPACEBAR]. The content will be saved and display will move the next address. Modify if required.

(9) Use UP/DOWN arrow keys move between the memories locations.

(10) Press [1][A] to select assembler mode.

(11) The display will show ‘RAM ADR: [BLANK]’, asking for the starting address which is 2000H in the above program.

(12) Type [2000] and press [ENTER].

(13) Display will show ‘2000: [BLANK]’. Type the mnemonics of the command operand etc. and press [ENTER].

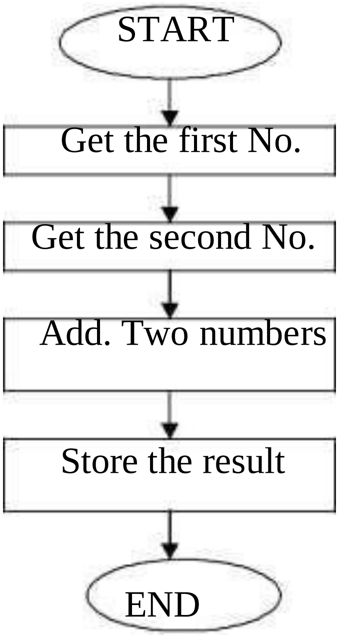
(14) The display will show the next location. Type desired mnemonics and operand. Keep on doing this until the end of the program.

(15) Feed the data using the [M] command.

(16) To RUN/ EXECUTE the program Press [G]<Starting Address>[$].

(17) Go to memory locations used for result storage using [M] command and verify the results.

Figure: Flow-Chart of the Process



Input Data: 2501- 67h
2502- 22h

Output Data: 2503- 89h

Precautions:-

(1) Make sure proper handling of equipments/ kits.

(2) Make sure that all the machine codes/ mnemonics are as per the program.

Result:-

Addition of 67H and 22H is 89H. Hence experiment for the addition of two 8-bit numbers has been successfully performed.

EXPERIMENT NO. 3

Aim:-

Write a program using 8085 for subtraction of two 8-bit numbers.

Apparatus:-

8085 microprocessor kit (STUDENT-85AD), ASCII (PS-2) keyboard.

Theory:-

2501h is the address of memory location for the 1st number. 2nd number is stored at next memory location. By the virtue of the program given below we first move the 1st number into accumulator register then subtract the 2nd number from this and use the next memory location i.e. 2503h to store the result. The program given below is self explanatory.

Program:-

Memory address	Opcode	Mnemonics	Operands	Comments
2000	21,01,25	LXI	H, 2501	Get address of 1st no. in HL pair
2003	7E	MOV	A, M	Move 1st no. in accumulator
2004	23	INX	H	HL points 2502H.
2005	96	SUB	M	Subtract 2nd no. from 1st no.
2006	23	INX	H	HL points 2503H.
2007	77	MOV	M, A	Move contents of acc. to memory
2008	CF	RST 1		Stop

Procedure:-

(1) Connect the kit to the power supply.

(2) Connect the PS-2 keyboard to the PS-2 (Female) connector CN-7 of the kit.

(3) Switch ON the kit.

(4) LCD displays “STUDENT-85”, if not so check power connections.

(5) If backlight is glowing and display shows something else press the RESET key provided on the kit. If still the persists contact Lab staff.

(6) To feed the program and data in the RAM, for machine language follow the steps 7 to 9 and for assembly language programming steps 10 to 14.

(7) Press [M]<Starting Address >[ENTER]. Starting address here is 2000H, so syntax will be [M]<2000>[ENTER].

(8) The LCD display will show the address and its content. Modify the content according to the program using keyboard then press [ENTER] or [SPACEBAR]. The content will be saved and display will move the next address. Modify if required.

(9) Use UP/DOWN arrow keys move between the memory locations.

(10) Press [1][A] to select assembler mode.

(11) The display will show ‘RAM ADR: [BLANK]’, asking for the starting address which is 2000H in the above program.

(12) Type [2000] and press [ENTER].

(13) Display will show ‘2000: [BLANK]’. Type the mnemonics of the command operand etc. and press [ENTER].

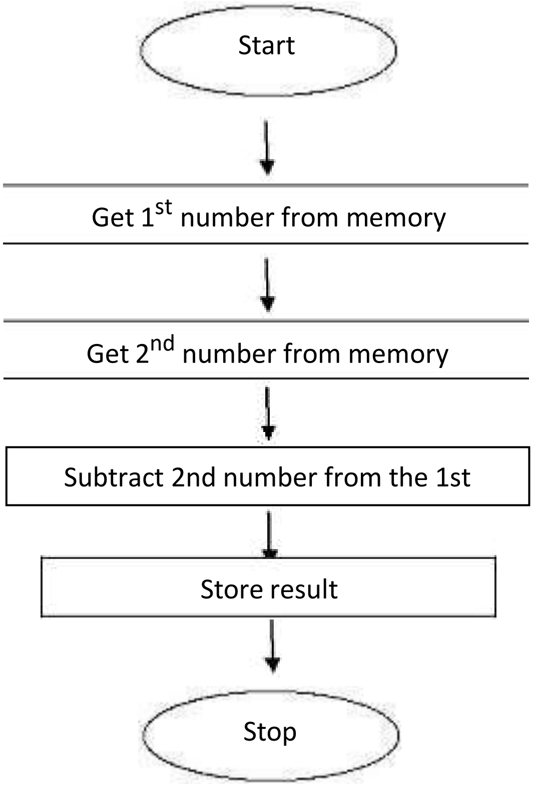
(14) The display will show the next location. Type desired mnemonics and operand. Keep on doing this until the end of the program.

(15) Feed the data using the [M] command.

(16) To RUN/ EXECUTE the program Press [G]<Starting Address>[$].

(17) Go to memory locations used for result storage using [M] command and verify the results.

Flow-Chart of the process :-


Start
Get 1st number from memory
Get 2nd number from memory
Subtract 2nd number from the 1st
Store result
Stop

Input Data: **Case-I** 2501h- 67h, 2502h- 22h

Case-II 2501h- 22h, 2502h- 67h

Output Data: **Case-I** 2503h- 45h

Case-II 2503h- BBh

Precautions:-

(1) Make sure proper handling of equipments/ kits.

(2) Make sure that all the machine codes/ mnemonics are as per the program.

Result:-

Subtraction of 22h from 67h is 45h. If we are subtracting a bigger number from a smaller number i.e. 22h-67h, then result is two's complement of 45h i.e. BBh. Hence experiment for the subtraction of two 8-bit numbers has been successfully performed.

EXPERIMENT NO. 4

Aim:-

Write a well-documented program using 8085 for addition of two 8-bit numbers with carry.

Apparatus:-

8085 microprocessor kit (STUDENT-85AD), ASCII (PS-2) keyboard.

Theory:-

2501h is the address of memory location for the 1st number. 2nd number is stored at next memory location. By the virtue of the program given below we first move the 1st number into accumulator register then add the 2nd number to this and if a carry is there we use C register to store carry. We use next two memory locations i.e. 2503h & 2504h to store the result and carry respectively. The program given below is self explanatory.

Program:-

Memory Address	Machine Code	Labels	Mnemonics	Operands	Comments
2000	21,01,25		LXI	H,2501	Get address of 1st no. in HL pair
2003			MVI	C,00	MSB of sum, Initial value=00
2005	7E		MOV	A,M	Move 1st no. in accumulator
2006	23		INX	H	HL points the address 2502H
2007	86		ADD	M	Add the 2nd no.
2008	D2,0D,25		JNC	AHEAD	If no carry jump to AHEAD
200B	0C		INR	C	If carry increment C
200C	32,03,25	AHEAD	STA	2503h	Store result at 2503h.
200F	79		MOV	A,C	Get carry in acc.
2010	32,04,25		STA	2504h	Store carry at 2504h
2013	76		HLT		Stop

Procedure:-

(1) Connect the kit to the power supply.

(2) Connect the PS-2 keyboard to the PS-2 (Female) connector CN-7 of the kit.

(3) Switch ON the kit.

(4) LCD displays “STUDENT-85”, if not so check power connections.

(5) If backlight is glowing and display shows something else press the RESET key provided on the kit. If still the persists contact Lab staff.

(6) To Feed the program and data in the RAM, for machine language follow the steps 7 to 9 and for assembly language programming steps 10 to 14.

(7) Press [M]<Starting Address >[ENTER]. Starting address here is 2000H, so syntax will be

[M]<2000>[ENTER].

(8) The LCD display will show the address and its content. Modify the content according to the program using keyboard then press [ENTER] or [SPACEBAR]. The content will be saved and display will move the next address. Modify if required.

(9) Use UP/DOWN arrow keys move between the memory locations.

(10) Press [1][A] to select assembler mode.

(11) The display will show 'RAM ADR: [BLANK]', asking for the starting address which is 2000H in the above program.

(12) Type [2000] and press [ENTER].

(13) Display will show '2000: [BLANK]'. Type the mnemonics of the command operand etc. and press [ENTER].

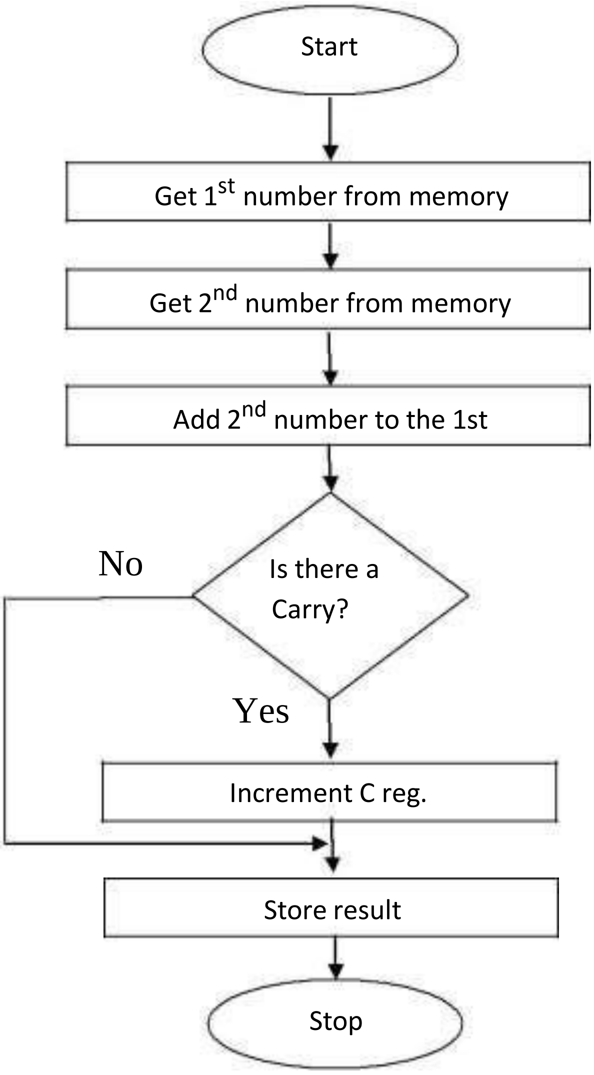
(14) The display will show the next location. Type desired mnemonics and operand. Keep on doing this until the end of the program.

(15) Feed the data using the [M] command.

(16) To RUN/ EXECUTE the program Press [G]<Starting Address>[$].

(17) Go to memory locations used for result storage using [M] command and verify the results.

Flow-Chart of the process :-


Start
Get 1st number from memory
Get 2nd number from memory
Add 2nd number to the 1st
Is there a Carry?
No
Yes
Increment C reg.
Store result
Stop

Input Data:	2501- 67h
	2502- 22h
Output Data:	2503- 89h

Precautions:-

(1) Make sure proper handling of equipments/ kits.

(2) Make sure that all the machine codes/ mnemonics are as per the program.

Result:-

Addition of 67h and 22h is 89h. Hence experiment for the addition of two 8-bit numbers has been successfully performed.

EXPERIMENT NO. 5

Aim:

Write a well-documented program using 8085 for subtraction of two 8-bit numbers with carry.

Apparatus:

8085 microprocessor kit (STUDENT-85AD), ASCII (PS-2) keyboard.

Theory:-

2501h is the address of memory location for the 1st number. 2nd number is stored at next memory location. By the virtue of the program given below we first move the 1st number into accumulator register then subtract the 2nd number from this and if a borrow is there we use C register to store it. We use next two memory locations i.e. 2503h & 2504h to store the result and borrow respectively. The program given below is self explanatory.

Program:-

Memory Address	Machine Code	Labels	Mnemonics	Operands	Comments
2000	21,01,25		LXI	H,2501	Get address of 1st no. in HL pair.
2003	0E,00		MVI	C,00	MSB of sum, Initial value=00
2005	7E		MOV	A,M	Move 1st no. in accumulator
2006	23		INX	H	HL points the address 2502H
2007	86		SUB	M	Subtract 2nd no. from acc.
2008	D2,0D,25		JNC	AHEAD	If no borrow jump to AHEAD
200B	0C		INR	C	If borrow increment C
200C	32,03,25	AHEAD	STA	2503h	Store result at 2503h.
200F	79		MOV	A,C	Get carry in acc.
2010	32,04,25		STA	2504h	Store carry at 2504h
2013	76		HLT		Stop

Procedure:-

(1) Connect the kit to the power supply.

(2) Connect the PS-2 keyboard to the PS-2 (Female) connector CN-7 of the kit.

(3) Switch ON the kit.

(4) LCD displays “STUDENT-85”, if not so check power connections.

(5) If backlight is glowing and display shows something else press the RESET key provided on

the kit. If still the persists contact Lab staff.

(6) To feed the program and data in the RAM, for machine language follow the steps 7 to 9 and for assembly language programming steps 10 to 14.

(7) Press [M]<Starting Address >[ENTER]. Starting address here is 2000H, so syntax will be [M]<2000>[ENTER].

(8) The LCD display will show the address and its content. Modify the content according to the program using keyboard then press [ENTER] or [SPACEBAR]. The content will be saved and display will move the next address. Modify if required.

(9) Use UP/DOWN arrow keys move between the memory locations.

(10) Press [1][A] to select assembler mode.

(11) The display will show 'RAM ADR: [BLANK]', asking for the starting address which is 2000H in the above program.

(12) Type [2000] and press [ENTER].

(13) Display will show '2000: [BLANK]'. Type the mnemonics of the command operand etc. and press [ENTER].

(14) The display will show the next location. Type desired mnemonics and operand. Keep on doing this until the end of the program.

(15) Feed the data using the [M] command.

(16) To RUN/ EXECUTE the program Press [G]<Starting Address>[$].

(17) Go to memory locations used for result storage using [M] command and verify the results.

Input Data: 2501- 67h

2502- 22h

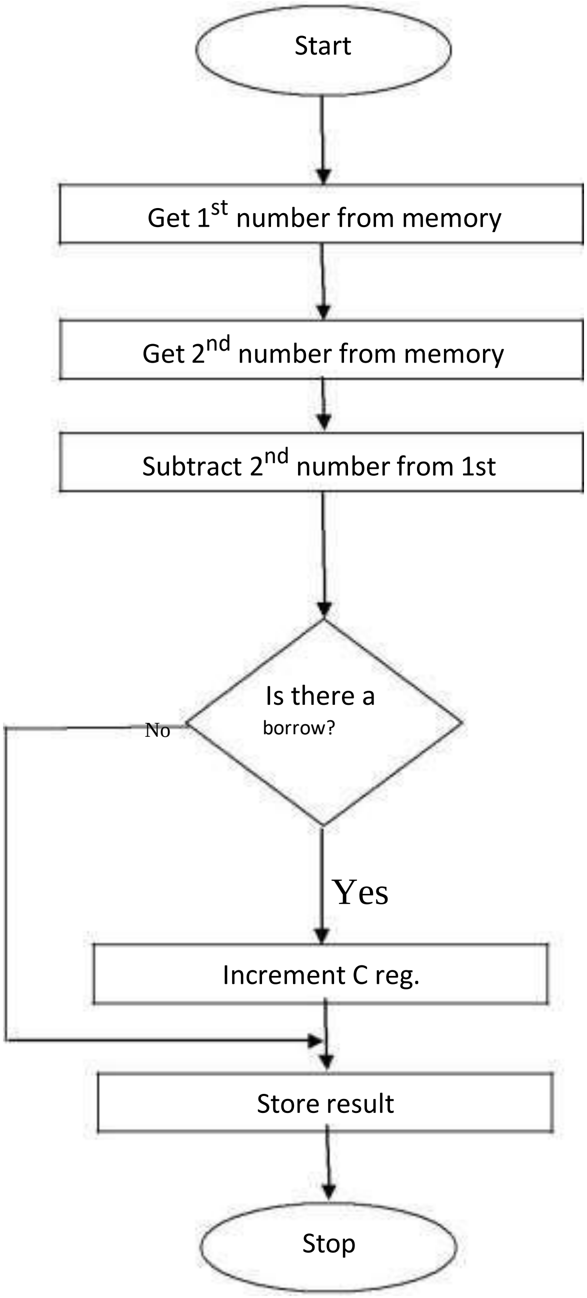
Output Data: 2503- 89h

Precautions:-

(1) Make sure proper handling of equipments/ kits.

(2) Make sure that all the machine codes/ mnemonics are as per the program.

Flow-Chart of the process :-


Start
Get 1st number from memory
Get 2nd number from memory
Subtract 2nd number from 1st
Is there a borrow?
No
Yes
Increment C reg.
Store result
Stop

Result:-

Addition of 67h and 22h is 89h. Hence experiment for the addition of two 8-bit numbers has been successfully performed.

EXPERIMENT NO. 6

Aim:-

Write a program using 8085 for addition of two BCD (8-bit) numbers.

Apparatus:-

8085 microprocessor kit (STUDENT-85AD), PS-2 keyboard.

Theory:

2501h is the address of memory location for the 1st number. 2nd number is stored at next memory location. By the virtue of the program given below we first move the 1st number into accumulator register then add the 2nd number to this and use the DAA (Decimal Adjust Accumulator) instruction to convert result to BCD. The result is then stored into the memory location 2503h. The comments in program give explanation of what happens after the execution of that particular instruction.

Procedure:-

(1) Connect the kit to the power supply.

(2) Connect the PS-2 keyboard to the PS-2 (Female) connector CN-7 of the kit.

(3) Switch ON the kit.

(4) LCD displays "STUDENT-85", if not so check power connections.

(5) If backlight is glowing and display shows something else press the RESET key provided on the kit. If still the persists contact Lab staff.

(6) To feed the program and data in the RAM, for machine language follow the steps 7 to 9 and for assembly language programming steps 10 to 14.

(7) Press [M]<Starting Address >[ENTER]. Starting address here is 2000H, so syntax will be [M]<2000>[ENTER].

(8) The LCD display will show the address and its content. Modify the content according to the program using keyboard then press [ENTER] or [SPACEBAR]. The content will be saved and display will move the next address. Modify if required.

(9) Use UP/DOWN arrow keys move between the memory locations.

(10) Press [1][A] to select assembler mode.

(11) The display will show 'RAM ADR: [BLANK]', asking for the starting address which is 2000H in the above program.

(12) Type [2000] and press [ENTER].

(13) Display will show '2000: [BLANK]'. Type the mnemonics of the command operand etc. and press [ENTER].

(14) The display will show the next location. Type desired mnemonics and operand. Keep on doing this until the end of the program.

(15) Feed the data using the [M] command.

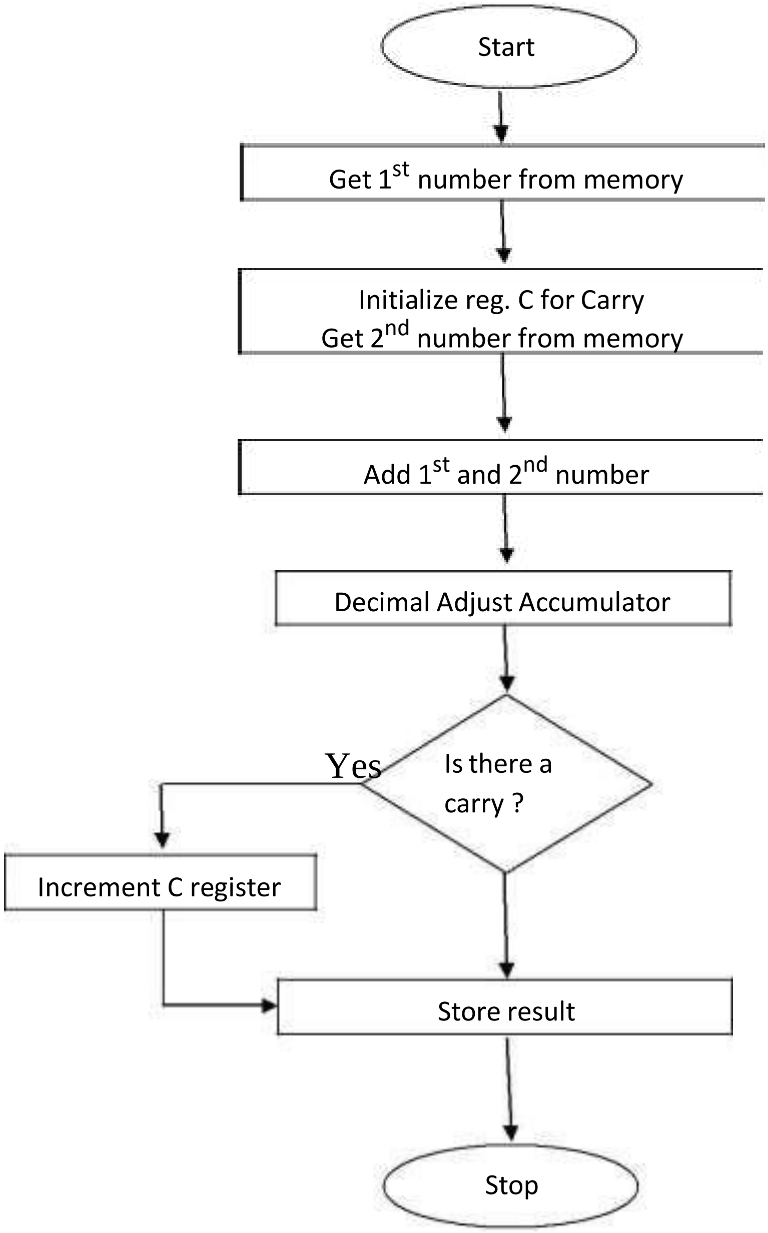
(16) To RUN/ EXECUTE the program Press [G]<Starting Address>[$].

(17) Go to memory locations used for result storage using [M] command and verify the results.

Program;-

Memory address	Op-code	Labels	Mnemonics	Operands	Comments
2000	21,01,25		LXI	H, 2501	Get address of 1st no. in H-L Pair
2003	0E,00		MVI	C,00h	Initialize C reg. to store MSB of result.
2005	7E		MOV	A, M	Move 1st no. in Accumulator
2006	23		INX	H	H-L points to 2502h.
2007	86		ADD	M	Add 2nd no. to 1st no.
2008	27		DAA		Decimal Adjust Accumulator
2009	D2,0D,20		JNC	AHEAD	If no carry go to AHEAD
200C	0C		INR	C	If carry increment C reg.
200D	32,03,25	AHEAD	STA	2503h	LSB of the sum in 2503h.
2010	79		MOV	A,C	MSB of sum in acc.
2011	32,04,25		STA	2504h	Store MSB at 2504h
2014	76		HLT		Stop

Flow-Chart of the process :



Start
Get 1st number from memory
Initialize reg. C for Carry
Get 2nd number from memory
Add 1st and 2nd number
Decimal Adjust Accumulator
Yes
Is there a carry ?
Increment C register
Store result
Stop

Input Data: 2501- 67h, 2502- 22h

Output Data: 2503- 89h

Precautions:-

(1) Make sure proper handling of equipments/ kits.

(2) Make sure that all the machine codes/ mnemonics are as per the program.

Result:-

Addition of 22h from 67h is 89h. Hence experiment for the addition of two 8-bit BCD numbers has been successfully performed.

EXPERIMENT NO. 7

Aim:-

Write a program using 8085 for subtraction of two BCD (8-bit) numbers.

Apparatus:-

8085 microprocessor kit (STUDENT-85AD), PS-2 keyboard.

Theory:-

2501h is the address of memory location for the 1st number. 2nd number is stored at next memory location. By the virtue of the program given below we first move the 1st number into accumulator register then subtract the 2nd number from this and use the DAA (Decimal Adjust Accumulator) instruction to convert result to BCD. The result is then stored into the memory location 2503h. The comments in program give explanation of what happens after the execution of that particular instruction.

Program;-

Memory address	Op-code	Mnemonics	Operands	Comments
2000	21,02,25	LXI	H, 2502	Get address of 2nd no. in H-L pair
2003	3E,99	MVI	A,99h	Place 99h in acc.
2005	96	SUB	M	9's complement of 2nd number
2006	3C	INR	A	10's complement of 2nd number
2007	2B	DCX	H	Decrement H-L pair
2008	86	ADD	M	Add 1st number to 10's complement of 2nd number
2009	27	DAA		Decimal Adjust Accumulator
200A	32,03,25	STA	2503h	Store result at 2503h
200D	CF	RST 1		Stop

Procedure:-

(1) Connect the kit to the power supply.

(2) Connect the PS-2 keyboard to the PS-2 (Female) connector CN-7 of the kit.

(3) Switch ON the kit.

(4) LCD displays "STUDENT-85", if not so check power connections.

(5) If backlight is glowing and display shows something else press the RESET key provided on the kit. If still the persists contact Lab staff.

(6) To Feed the program and data in the RAM, for machine language follow the steps 7 to 9 and for assembly language programming steps 10 to 14.

(7) Press [M]<Starting Address >[ENTER]. Starting address here is 2000H, so syntax will be

[M]<2000>[ENTER].

(8) The LCD display will show the address and its content. Modify the content according to the program using keyboard then press [ENTER] or [SPACEBAR]. The content will be saved and display will move the next address. Modify if required.

(9) Use UP/DOWN arrow keys move between the memory locations.

(10) Press [1][A] to select assembler mode.

(11) The display will show ‘RAM ADR: [BLANK]’, asking for the starting address which is 2000H in the above program.

(12) Type [2000] and press [ENTER].

(13) Display will show ‘2000: [BLANK]’. Type the mnemonics of the command operand etc. and press [ENTER].

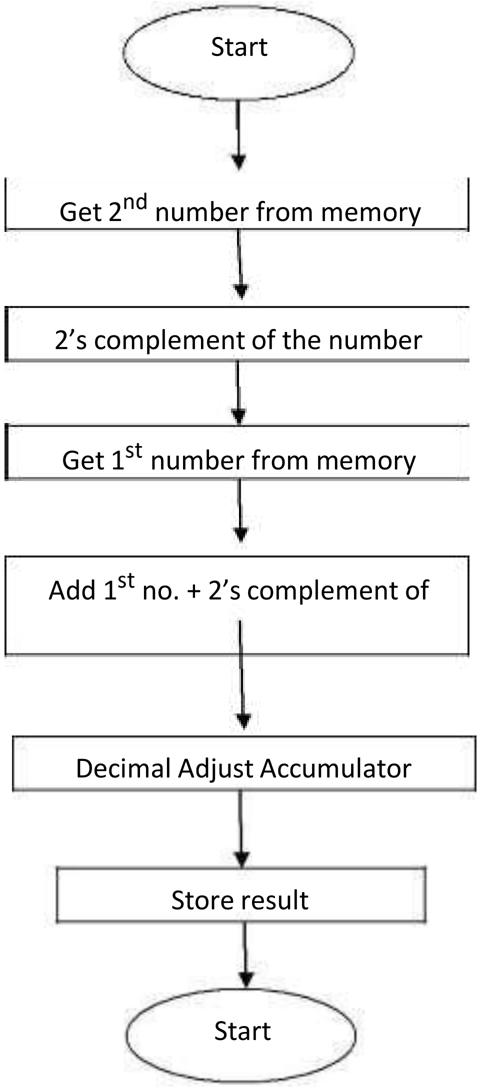
(14) The display will show the next location. Type desired mnemonics and operand. Keep on doing this until the end of the program.

(15) Feed the data using the [M] command.

(16) To RUN/ EXECUTE the program Press [G]<Starting Address>[$].

(17) Go to memory locations used for result storage using [M] command and verify the results.

Flow-Chart of the process :-


Start
Get 2nd number from memory
2’s complement of the number
Get 1st number from memory
Add 1st no. + 2’s complement of
Decimal Adjust Accumulator
Store result
Start

Input Data: 2501- 67h (0110 0111), 2502- 22h (0010 0010)

Output Data: 2503- 45h (0100 0101)

Precautions:-

(1) Make sure proper handling of equipments/ kits.

(2) Make sure that all the machine codes/ mnemonics are as per the program.

Result:-

Subtraction of 22h from 67h is 45h. Hence experiment for the subtraction of two 8-bit BCD numbers has been successfully performed.

EXPERIMENT NO. 8

Aim:-

Write a program using 8085 for multiplication of two 8-bit numbers by repeated addition method.

Apparatus:-

8085-micro-processor kit (STUDENT-85AD), ASCII (PS-2) keyboard.

Theory:

Repeated Addition method of multiplication is simplest method of multiplying two numbers. For use in microprocessors, we first store multiplicand and multiplier in any of its general purpose registers. We use a register pair for the storage of result. Then we keep on adding the multiplicand to result and decrement the multiplier every time we add. This cycle continues until multiplier becomes zero. Then the result is moved to any desired memory location. A well documented program for this purpose is given below.

Program:-

Memory Address	Machine Code	Labels	Mnemonics	Operands	Comments
2000	0E,10		MVI	C,08h	Move the multiplicand in reg. C
2002	1E,20		MVI	E,07h	Move the multiplier in reg. E
2004	06,00		MVI	B,00	Load 00h in B reg.
2006	21,00,00		LXI	H,0000	Initial Product=0000
2009	09	UP1:	DAD	B	HL+BC=>HL
200A	1D		DCR	E	Decrement reg. E
200B	C2,09,20		JNZ	UP1	Jump if not zero to location up1
200E	22,01,25		SHLD	2501h	Store result at 2501h
2012	76		HLT		Stop

Procedure:-

(1) Connect the kit to the power supply.

(2) Connect the PS-2 keyboard to the PS-2 (Female) connector CN-7 of the kit.

(3) Switch ON the kit.

(4) LCD displays “STUDENT-85”, if not so check power connections.

(5) If backlight is glowing and display shows something else press the RESET key provided on the kit. If still the persists contact Lab staff.

(6) To Feed the program and data in the RAM, for machine language follow the steps 7 to 9 and for assembly language programming steps 10 to 14.

(7) Press [M]<Starting Address >[ENTER]. Starting address here is 2000H, so syntax will be [M]<2000>[ENTER].

(8) The LCD display will show the address and its content. Modify the content according to the program using keyboard then press [ENTER] or [SPACEBAR]. The content will be saved and display will move the next address. Modify if required.

(9) Use UP/DOWN arrow keys move between the memory locations.

(10) Press [1][A] to select assembler mode.

(11) The display will show 'RAM ADR: [BLANK]', asking for the starting address which is 2000H in the above program.

(12) Type [2000] and press [ENTER].

(13) Display will show '2000: [BLANK]'. Type the mnemonics of the command operand etc. and press [ENTER].

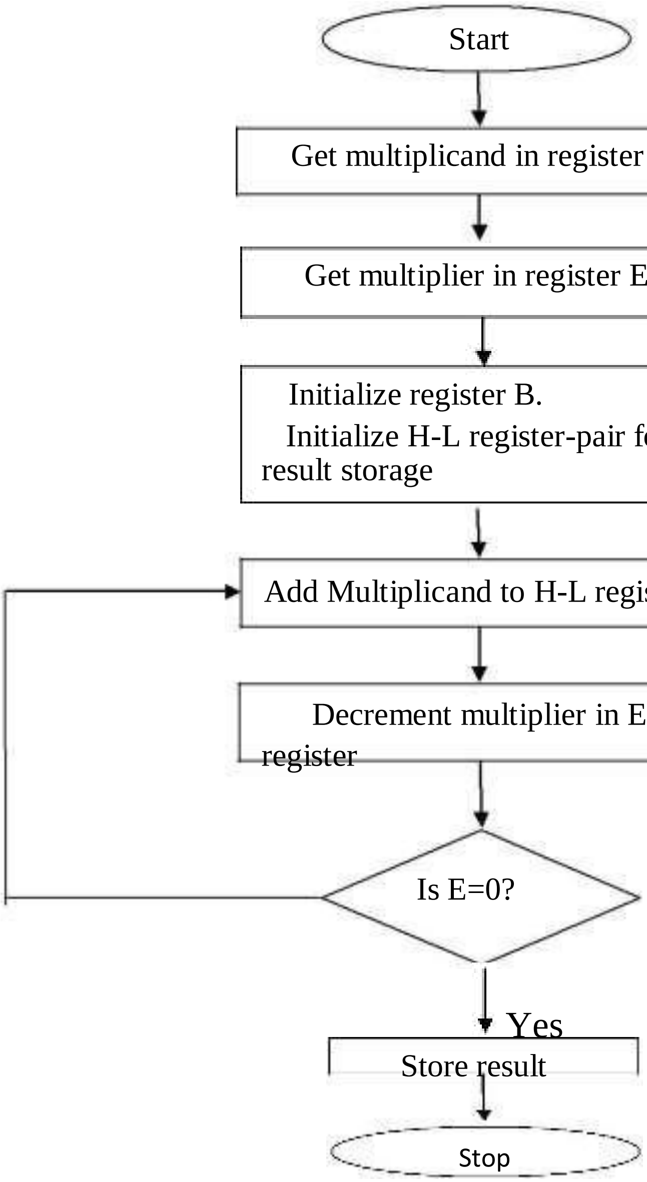
(14) The display will show the next location. Type desired mnemonics and operand. Keep on doing this until the end of the program.

(15) Feed the data using the [M] command.

(16) To RUN/ EXECUTE the program Press [G]<Starting Address>[$].

(17) Go to memory locations used for result storage using [M] command and verify the results.

Flow-Chart of the process:-


Start
Get multiplicand in register C
Get multiplier in register E
Initialize register B.
Initialize H-L register-pair for result storage
Add Multiplicand to H-L register
Decrement multiplier in E register
Is E=0?
Yes
Store result
Stop

Input Data: C reg.- 08h (0000 1000), E reg.- 07h (0000 0111)

Output Data: 2501- 38h (0011 1000)

Precautions:-

(1) Make sure proper handling of equipments/ kits.

(2) Make sure that all the machine codes/ mnemonics are as per the program.

Result:-

Multiplication of 08h and 07h is 38h. Hence experiment for the multiplication of two 8-bit numbers by repeated addition method has been successfully performed.

EXPERIMENT NO.9

Aim:-

Write a program using 8085 for multiplication of two 8-bit numbers by bit-rotation method.

Apparatus:-

8085- micro-processor kit (STUDENT-85AD), ASCII (PS-2) keyboard.

Theory:-

In binary multiplication we see that when a multiplicand is multiplied by 1 the product is same as the multiplicand. When a multiplicand is multiplied by zero, the product is zero. The procedure for multiplication is that multiplicand is rotated to left by one bit if the MSB of multiplier is 1. Partial product is stored. We keep on rotating the multiplier, decrementing cycle-counter and add the multiplicand to the partial product until the cycle-counter which is set to 8 becomes zero. When cycle-counter is 0 that means we have completed the multiplication and we store the result as our final result. The method of multiplication applied here is known as BIT ROTATION METHOD.

Program:-

Memory Address	Machine Code	Labels	Mnemonics	Operands	Comments
2000	2A,01,25		LHLD	7501 H	Get Multiplicand in H-L pair.
2003	EB		XCHG		Exchange HL pair with DE pair
2004	3A,03,25		LDA	7503 H	Get 2nd no. in acc.
2007	21,00,00		LXI	H,0000	Initial product in HL=00
200A	0E,08		MVI	C,08H	Count=08 in reg. C
200C	29	LOOP	DAD	H	Shift partial product left by 1 bit
200D	17		RAL		Rotate multiplier by 1bit.
200E	D2,12,20		JNC	AHEAD	If no carry, go to AHEAD
2011	19		DAD	D	Product=Product +Multiplicand
2012	0D	AHEAD	DCR	C	Decrement Count
2013	C2,0C,20		JNZ	LOOP	If C≠0, go to LOOP
2016	22,04,25		SHLD	7504	Store result
2019	76		HLT		Stop

Procedure:-

(1) Connect the kit to the power supply.

(2) Connect the PS-2 keyboard to the PS-2 (Female) connector CN-7 of the kit.

(3) Switch ON the kit.

(4) LCD displays “STUDENT-85”, if not so check power connections.

(5) If backlight is glowing and display shows something else press the RESET key provided on the kit. If still the persists contact Lab staff.

(6) To feed the program and data in the RAM, for machine language follow the steps 7 to 9 and for assembly language programming steps 10 to 14.

(7) Press [M]<Starting Address >[ENTER]. Starting address here is 2000H, so syntax will be [M]<2000>[ENTER].

(8) The LCD display will show the address and its content. Modify the content according to the program using keyboard then press [ENTER] or [SPACEBAR]. The content will be saved and display will move the next address. Modify if required.

(9) Use UP/DOWN arrow keys move between the memory locations.

(10) Press [1][A] to select assembler mode.

(11) The display will show ‘RAM ADR: [BLANK]’, asking for the starting address which is 2000H in the above program.

(12) Type [2000] and press [ENTER].

(13) Display will show ‘2000: [BLANK]’. Type the mnemonics of the command operand etc. and press [ENTER].

(14) The display will show the next location. Type desired mnemonics and operand. Keep on doing this until the end of the program.

(15) Feed the data using the [M] command.

(16) To RUN/ EXECUTE the program Press [G]<Starting Address>[$].

(17) Go to memory locations used for result storage using [M] command and verify the results.

Input Data: C reg.- 08h (0000 1000), E reg.- 07h (0000 0111)

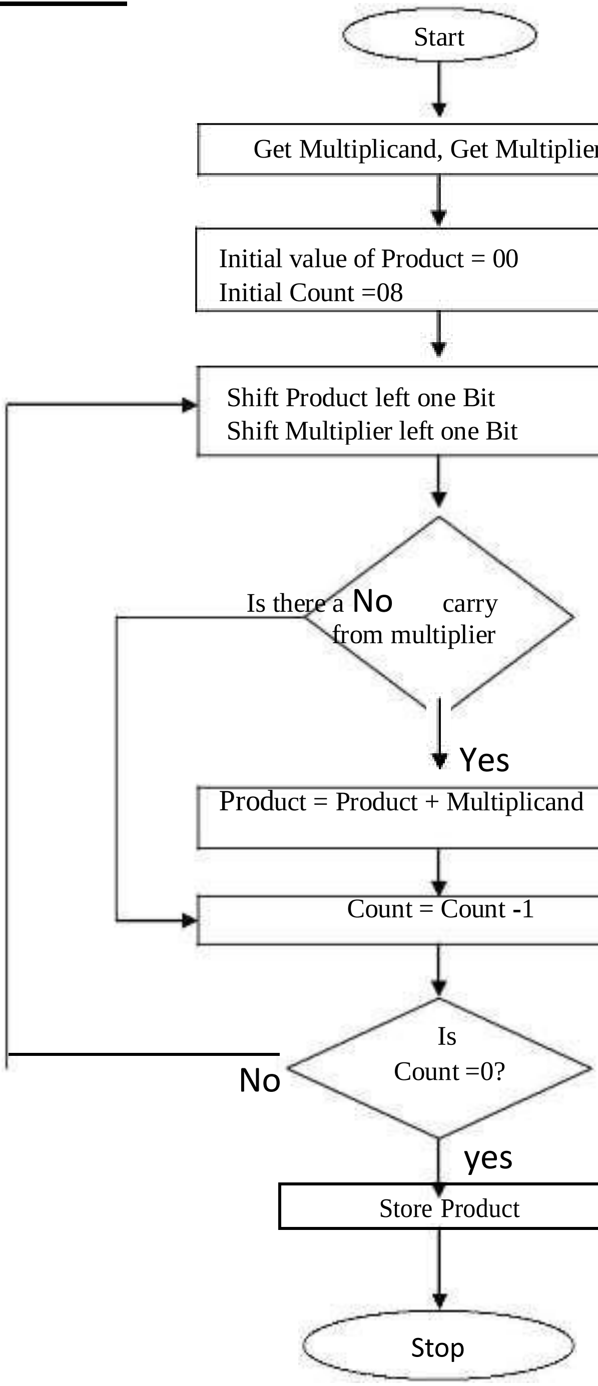
Output Data: 2501- 38h (0011 1000)

Precautions:-

(1) Make sure proper handling of equipments/ kits.

(2) Make sure that all the machine codes/ mnemonics are as per the program.

Flow-Chart of Process:-


Start
Get Multiplicand, Get Multiplier
Initial value of Product = 00
Initial Count =08
Shift Product left one Bit
Shift Multiplier left one Bit
Is there a No carry from multiplier
Yes
Product = Product + Multiplicand
Count = Count -1
Is Count =0?
No
yes
Store Product
Stop

Result:-

Multiplication of 08h and 07h is 38h. Hence experiment for the multiplication of two 8-bit numbers by repeated addition method has been successfully performed.

EXPERIMENT NO. 10

Aim:-

Write a program using 8085 for division of a number by another number (8-bit) using repeated subtraction method.

Apparatus:-

8085-micro-processor kit (STUDENT-85AD), ASCII (PS-2) keyboard.

Theory:-

Repeated subtraction method of division is the simplest method of dividing two numbers. For use in microprocessors, we can store dividend and divisor in any of its general purpose registers. If the dividend is bigger than 8-bit we can use a register pair for its storage. Then we keep on subtracting the divisor form dividend or remainder of subtraction and increment the quotient every time we subtract. This cycle continues until remainder becomes zero or less than divisor. Then the result is moved to any desired memory location. A well documented program for this purpose is given below.

Program:-

Memory Address	Machine Code	Labels	Mnemonics	Operands	Comments
2000	21,00,21		LXI	H,2100h	Address of divisor in H-L pair
2003	46		MOV	B,M	Divisor in reg. B
2004	23		INX	H	Increment H-L pair
2005	7E		MOV	A,M	Dividend in Acc.
2006	23		INX	H	Increment H-L pair
2007	0E,00		MVI	C,00h	Initialize C reg. for quotient storage
2009	B8	LP1	CMP	B	Compare acc. With B reg.
200A	DA,13,20		JC	LOOP	If carry jump to LOOP
200D	90		SUB	B	Subtract divisor from dividend
200E	0C		INR	C	Increment C reg.
200F	C3,09,20		JMP	LP1	Jump back and repeat from to LP1
2012	77	LOOP	MOV	M,A	Store remainder at 2102h
2013	23		INX	H	Increment H-L pair
2014	71		MOV	M,C	Store result at 2103h
2015	76		HLT		Stop

Procedure:-

(1) Connect the kit to the power supply.

(2) Connect the PS-2 keyboard to the PS-2 (Female) connector CN-7 of the kit.

(3) Switch ON the kit.

(4) LCD displays “STUDENT-85”, if not so check power connections.

(5) If backlight is glowing and display shows something else press the RESET key provided on the kit. If still the persists contact Lab staff.

(6) To Feed the program and data in the RAM, for machine language follow the steps 7 to 9 and for assembly language programming steps 10 to 14.

(7) Press [M]<Starting Address >[ENTER]. Starting address here is 2000H, so syntax will be [M]<2000>[ENTER].

(8) The LCD display will show the address and its content. Modify the content according to the program using keyboard then press [ENTER] or [SPACEBAR]. The content will be saved and display will move the next address. Modify if required.

(9) Use UP/DOWN arrow keys move between the memory locations.

(10) Press [1][A] to select assembler mode.

(11) The display will show ‘RAM ADR: [BLANK]’, asking for the starting address which is 2000H in the above program.

(12) Type [2000] and press [ENTER].

(13) Display will show ‘2000: [BLANK]’. Type the mnemonics of the command operand etc. and press [ENTER].

(14) The display will show the next location. Type desired mnemonics and operand. Keep on doing this until the end of the program.

(15) Feed the data using the [M] command.

(16) To RUN/ EXECUTE the program Press [G]<Starting Address>[$].

(17) Go to memory locations used for result storage using [M] command and verify the results.

Input Data: 2100h - 38h (0011 1000), 2101h- 07h (0000 0111)

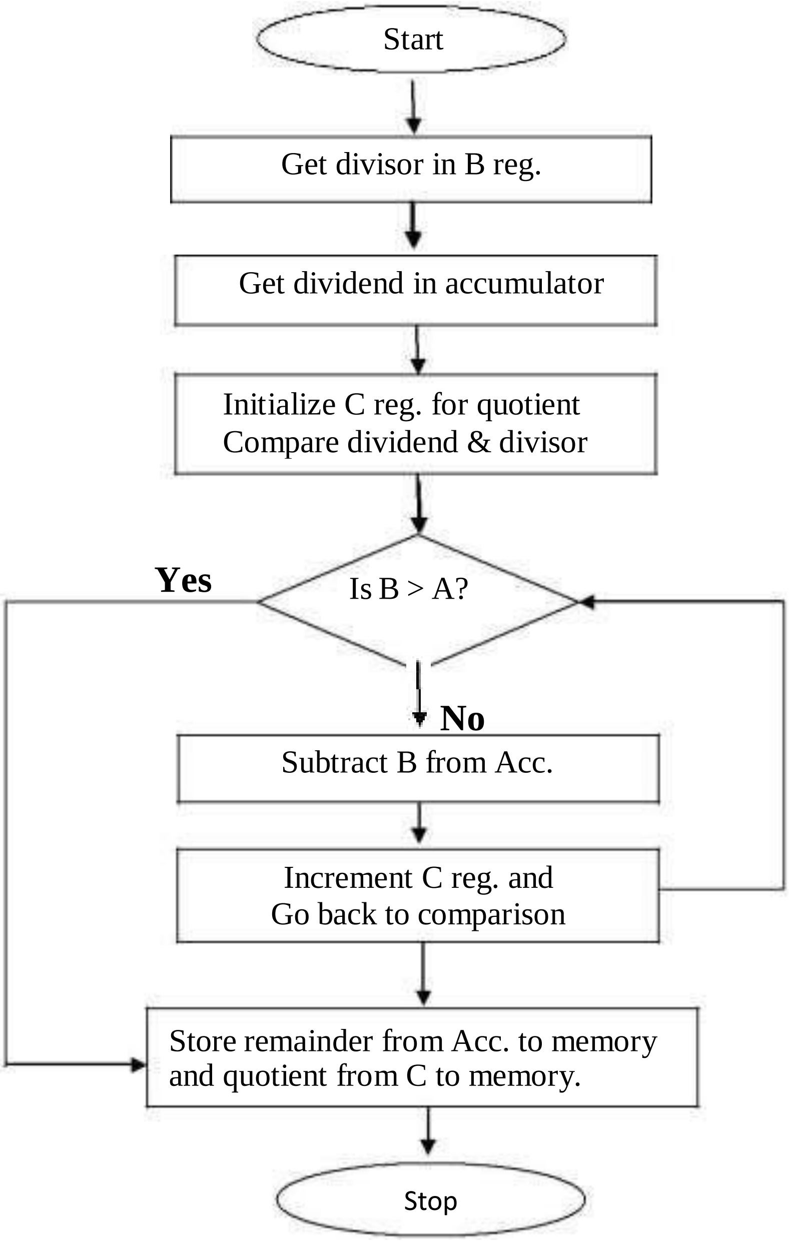
Output Data: 2102h - 00h (0000 0000), 2103h- 08h (0000 1000)

Precautions:-

(1) Make sure proper handling of equipments/ kits.

(2) Make sure that all the machine codes/ mnemonics are as per the program.

Flow-Chart of the process:-


Start
Get divisor in B reg.
Get dividend in accumulator
Initialize C reg. for quotient
Compare dividend & divisor
Yes
Is B > A?
No
Subtract B from Acc.
Increment C reg. and
Go back to comparison
Store remainder from Acc. to memory
and quotient from C to memory.
Stop

Result:-

Multiplication of 08h and 07h is 38h. Hence experiment for the multiplication of two 8-bit numbers by repeated addition method has been successfully performed.

EXPERIMENT NO.11

Aim:

Write a well document program using 8085 for division of two 8-bit numbers by bit-rotation method.

Apparatus:

8085-micro-processor kit (STUDENT-85AD), ASCII (PS-2) Keyboard.

Theory:

The computer performs division by trail subtraction. The divisor is subtracted from 8 MSBs of dividend. If there is borrow, the bit of quotient is set to 1; otherwise 0.to line up the dividend and quotient properly the dividend is shifted left by one bit before each trail of subtraction. The dividend and quotient share a 16-bit register. Due to shifting of dividend one bit register falls vacant in each step. The quotient is stored in vacant bit positions. The program and flow-chart are given below.

Program:

Memory Address	Machine Code	Labels	Mnemonics	Operands	Comments
2000	2A, 01,25		LHLD	2501 H	Enter the 16 bit address in HL pair
2003	3A, 03,25		LDA	2503 H	Get divisor from 2503
2006	47		MOV	B, A	Divisor in register B
2007	0E, 08		MVI	C, 08	Count = 08 in register C.
2009	29	LOOP	DAD	H	Shift dividend and quotient left by one bit.
200A	7C		MOV	A, H	Most significant bits of dividend in acc.
200B	90		SUB	B	Subtract divisor from MSB of dividend.
200C	DA,11,24		JC	AHEAD	Is MSB of dividend>divisor? No, go to AHEAD.
200F	67		MOV	H, A	MSB of dividend in reg. H
2010	2C		INR	L	Yes, add 1 to quotient.
2011	0D	AHEAD	DCR	C	Decrement count.
2012	C2,09,24		JNZ	LOOP	Is count=0? No, jump to loop.
2015	22,04,25		SHLD	2504 H	Store quotient in 2504 and remainder in 2505 H.
2018	76		HLT		Stop.

Procedure:-

(1) Connect the kit to the power supply.

(2) Connect the PS-2 keyboard to the PS-2 (Female) connector CN-7 of the kit.

(3) Switch ON the kit.

(4) LCD displays “STUDENT-85”, if not so check power connections.

(5) If backlight is glowing and display shows something else press the RESET key provided on the kit. If still the persists contact Lab staff.

(6) To Feed the program and data in the RAM, for machine language follow the steps 7 to 9 and for assembly language programming steps 10 to 14.

(7) Press [M]<Starting Address >[ENTER]. Starting address here is 2000H, so syntax will be [M]<2000>[ENTER].

(8) The LCD display will show the address and its content. Modify the content according to the program using keyboard then press [ENTER] or [SPACEBAR]. The content will be saved and display will move the next address. Modify if required.

(9) Use UP/DOWN arrow keys move between the memory locations.

(10) Press [1][A] to select assembler mode.

(11) The display will show ‘RAM ADR: [BLANK]’, asking for the starting address which is 2000H in the above program.

(12) Type [2000] and press [ENTER].

(13) Display will show ‘2000: [BLANK]’. Type the mnemonics of the command operand etc. and press [ENTER].

(14) The display will show the next location. Type desired mnemonics and operand. Keep on doing this until the end of the program.

(15) Feed the data using the [M] command.

(16) To RUN/ EXECUTE the program Press [G]<Starting Address>[$].

(17) Go to memory locations used for result storage using [M] command and verify the results.

Input Data: 2100h - 38h (0011 1000), 2101h- 07h (0000 0111)

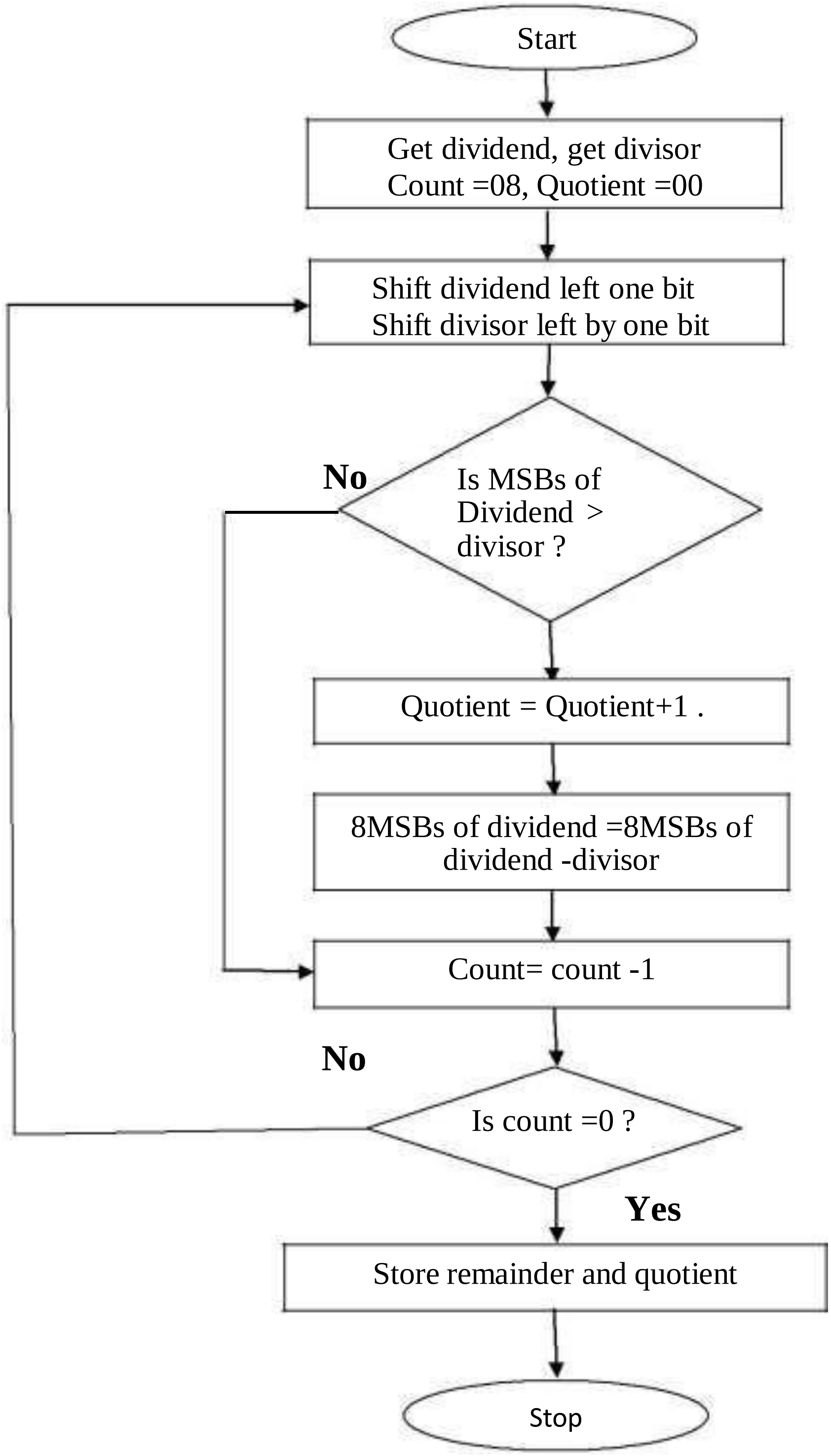
Output Data: 2102h - 00h (0000 0000), 2103h- 08h (0000 1000)

Precautions:-

(1) Make sure proper handling of equipments/ kits.

(2) Make sure that all the machine codes/ mnemonics are as per the program.

Flow-Chart of the process:-


Start
Get dividend, get divisor
Count =08, Quotient =00
Shift dividend left one bit
Shift divisor left by one bit
No
Is MSBs of
Dividend >
divisor ?
Quotient = Quotient+1 .
8MSBs of dividend =8MSBs of
dividend -divisor
Count= count -1
No
Is count =0 ?
Yes
Store remainder and quotient
Stop

Result:-

Division of 38h by 07h is 08h. Hence experiment for the division of an 8-bit numbers by repeated addition method has been successfully performed.

EXPERIMENT NO.12

Aim:

To write an ALP for finding the square of a number from look up table method using 8085.

Apparatus:

8085-micro-processor kit (STUDENT-85AD), ASCII (PS-2) keyboard.

Theory:

The squares of number are stored in certain memory locations in tabular form. This table is called look-up table. Here squares of numbers from 00 to 09 are stored at locations 2600 to 2609h. All these values are in decimal numbers. The program given below is self explanatory.

Program:

Memory	Machine	Mnemonics	Operands	Comments
2000	3A,00,25	LDA	2500 H	Get number in accumulator
2003	6F	MOV	L,A	Move From A into reg. L
2004	26,26	MVI	H,26H	Get 26 in reg. H
2006	7E	MOV	A,M	Square of data in accumulator
2007	32,01,25	STA	2501 H	Store square in 2501 H.
200A	76	HLT		Stop

Look-Up table

Memory Address	Data
2600h	00
2601h	01
2602h	04
2603h	09
2604h	16
2605h	25
2606h	36
2607h	49
2608h	64
2609h	81

Procedure:-

(1) Connect the kit to the power supply.

(2) Connect the PS-2 keyboard to the PS-2 (Female) connector CN-7 of the kit.

(3) Switch ON the kit. LCD displays "STUDENT-85", if not so check power connections. If backlight is glowing and display shows something else press the RESET key provided on the kit. If still the persists contact Lab staff.

(4) To Feed the program and data in the RAM, for machine language follow the steps 7 to 9 and for assembly language programming steps 10 to 14.

(5) Press [M]<Starting Address >[ENTER]. Starting address here is 2000H, so syntax will be [M]<2000>[ENTER].

(6) The LCD display will show the address and its content. Modify the content according to the program using keyboard then press [ENTER] or [SPACEBAR]. The content will be saved and display will move the next address. Modify if required.

(7) Use UP/DOWN arrow keys move between the memory locations.

(8) Press [1][A] to select assembler mode.

(9) The display will show 'RAM ADR: [BLANK]', asking for the starting address which is 2000H in the above program.

(10) Type [2000] and press [ENTER].

(11) Display will show '2000: [BLANK]'. Type the mnemonics of the command operand etc. and press [ENTER].

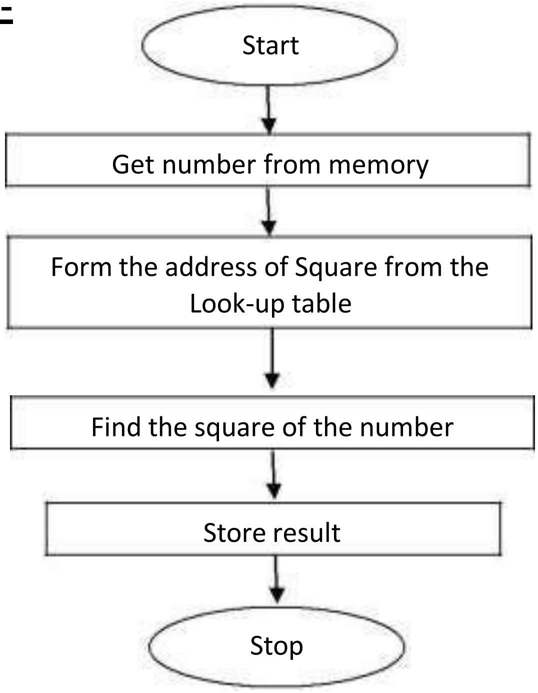
(12) The display will show the next location. Type desired mnemonics and operand. Keep on doing this until the end of the program.

(13) Feed the data using the [M] command.

(14) To RUN/ EXECUTE the program Press [G]<Starting Address>[$].

(15) Go to memory locations used for result storage using [M] command and verify the results.

Flow-Chart of the process:-


Start
Get number from memory
Form the address of Square from the Look-up table
Find the square of the number
Store result
Stop

Input Data: 2500h- 07d

Output Data: 2503h- 49d

Precautions:-

(1) Make sure proper handling of equipments/ kits.

(2) Make sure that all the machine codes/ mnemonics are as per the program.

Result:-

Square of the number 07d is 49d. Hence experiment for the calculation of square of given number using look-up table has been successfully performed.

EXPERIMENT NO. 13

Aim:

Write a program using 8085 for finding largest number in a data array.

Apparatus:

8085 microprocessor kit (STUDENT-85AD),ASCII (PS-2) Keyboard.

Program:

Memory Address	Machine Code	Mnemonics	Comments
7000	21,00,75	LXI H, 7500H	Address for count in H-L pair
7003	4E	MOV C, M	Count in register C
7004	23	INX H	Address of Ist number in HL Pair
7005	7E	MOV A,M	Ist no. in accumulator
7006	0D	DCR C	Decrement count
7007	23 Loop	INX H	Address of next number
7008	BE	CMP M	Is next number>previous no.
7009	D2,0D,70	JNC Ahead	If not carry,jump to ahead
700C	7E	MOV A,M	Get larger no. into acc.
700D	0D Ahead	DCR C	Decrement count
700E	C2,07,70	JNZ Loop	Jump if no zero to loop.
7011	32,50,74	STA 7450 H	Store result at 7450.
7014	CF	RST 1	Terminate.

Procedure:-

(1) Connect the kit to the power supply.

(2) Connect the PS-2 keyboard to the PS-2 (Female) connector CN-7 of the kit.

(3) Switch ON the kit.

(4) LCD displays “STUDENT-85”, if not so check power connections.

(5) If backlight is glowing and display shows something else press the RESET key provided on the kit. If still the persists contact Lab staff.

(6) To Feed the program and data in the RAM, for machine language follow the steps 7 to 9 and for assembly language programming steps 10 to 14.

(7) Press [M]<Starting Address >[ENTER]. Starting address here is 2000H, so syntax will be [M]<2000>[ENTER].

(8) The LCD display will show the address and its content. Modify the content according to the program using keyboard then press [ENTER] or [SPACEBAR]. The content will be saved and

display will move the next address. Modify if required.

(9) Use UP/DOWN arrow keys move between the memory locations.

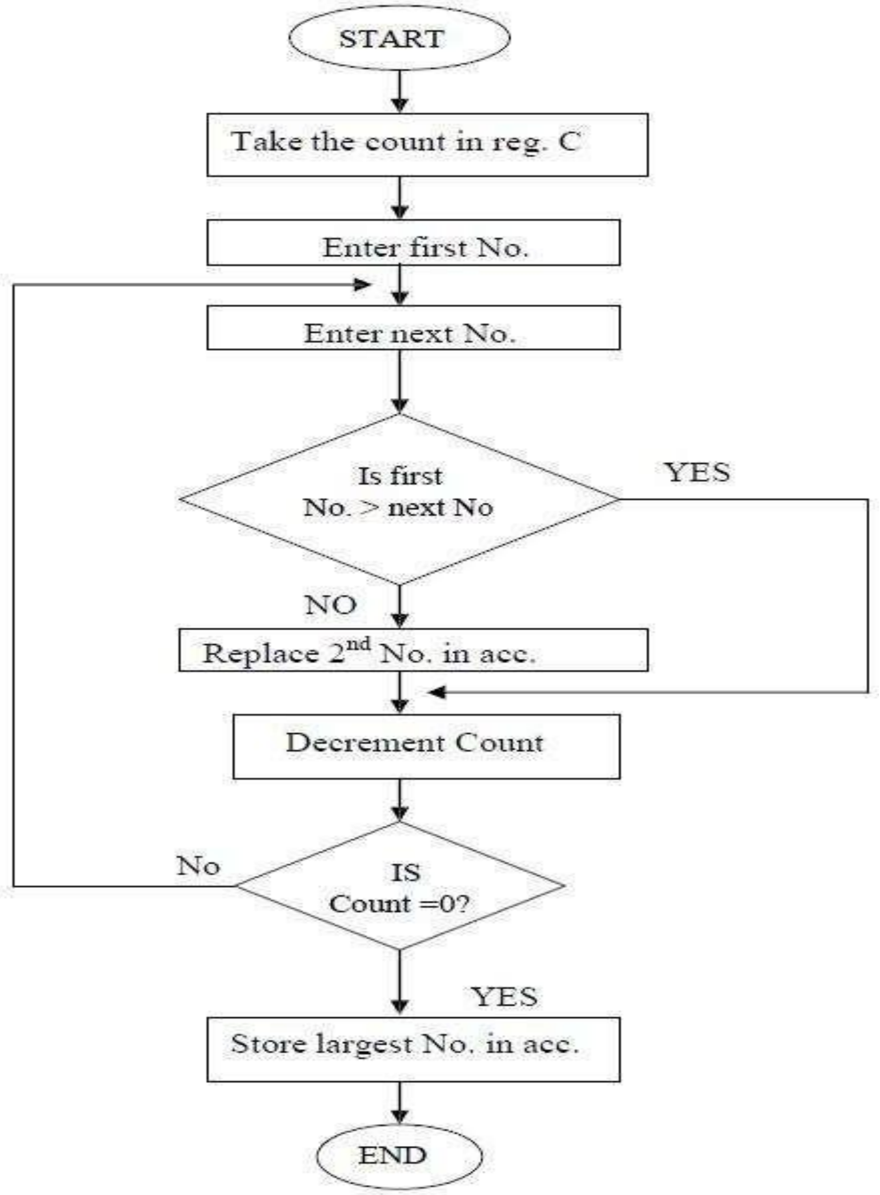
(10) Press [1][A] to select assembler mode.

(11) The display will show ‘RAM ADR: [BLANK]’, asking for the starting address which is 2000H in the above program.

(12) Type [2000] and press [ENTER].

(13) Display will show ‘2000: [BLANK]’. Type the mnemonics of the command operand etc. and press [ENTER].

Flow Chart:-



Input Data	7500-03 (Counter)
	7501-
	7502-
	7503-
Output Data:	7450-

Precautions:-

(1) Make sure proper handling of equipments/ kits.

(2) Make sure that all the machine codes/ mnemonics are as per the program.

Result:-

Thus the largest number in array stored in respective memory location and verified the data.

EXPERIMENT NO.14

AIM:

To study of 8086 Microprocessor Kit.

APPARATUS:

VMC-8609 8086 microprocessor trainer kit.

THEORY:

The 8086 is a 16-bit, N-channel, HMOS microprocessor. The term HMOS is used for"high-speed MOS". The 8086 uses 20 address lines and 16 data lines. It can directly address up to2^{20} = 1Mbytes of memory. The 16-bit data word is divided into a low-order byte and a high-order byte. The 20 address lines are time multiplexed lines. The 16 low-order address lines are time multiplexed with data, and the 4 high-order address lines are time multiplexed with status signals.

OPERATING MODES OF 8086

There are two modes of operation for Intel 8086, namely the minimum mode and the maximum mode. When only one 8086 CPU is to be used in a microcomputer system the 8086 is used in the minimum mode of operation. In this mode the CPU issues the control signals required by memory and I/O devices. In case of maximum mode of operation control signals are issued by Intel 8288 bus controller which is used with 8086 for this very purpose. When MN/MX is high the CPU operates in the minimum mode. When it is low the CPU operates in the maximum mode.

Pin Description for Minimum Mode

For the minimum mode of operation the pin MN/$\overline{\text{MX}}$ is connected to 5V d.c.supply. The description of the pins from 24 to 31 for the minimum mode is as follows:

$\overline{\text{INTA}}$(Output): Pin no. 24 Interrupt acknowledge. On receiving interrupt signal the processor issues an interrupt acknowledge signal. It is active LOW.

ALE(Output) : Pin no. 25 Address latch enable. It goes HIGH during T1. The microprocessor sends this signal to latch the address into the Intel 8282/8283 latch.

$\overline{\text{DEN}}$(Output) : Pin no. 26 Data enable. When Intel 8286/8287 octal bus transceiver is used this signal acts as an output enable signal. It is active LOW.

DT/R(Output) : Pin no. 27 Data Transmit/Receive. When Intel 8286/8287 octal bus transceiver is used this signal controls the direction of data flow through the transceiver. When it is High data are sent out. When it is LOW data are received.

M/$\overline{\text{IO}}$(Output) : Pin no. 28.Memory or I/O access. When it is HIGH the CPU wants to access memory. When it is LOW the CPU wants to access I/O device.

$\overline{\text{WR}}$ (Output): Pin no. 29. Write. When it is LOW the CPU performs memory or I/O write Operation.

HLDA (Output) : Pin no. 30.HOLD acknowledge. It is issued by the processor when it receives HOLD signal. It is active HIGH signal. When HOLD request is removed HLDA goes LOW.

HOLD (Output) : Pin no. 31.Hold. When another device in microcomputer system wants to use the address and data bus, it sends a HOLD request to CPU through this pin. It is an active HIGH signal.

Pin Description for Maximum Mode

For the maximum mode of operation the pin MN/$\overline{\text{MX}}$ is made LOW. It is grounded. The description of the pins from 24 to 31 is as follows:

QS1,QS0(Output): Pin no. 24,25 Instruction Queue status. Logic are given below:

QS1	QS0	
0	0	No operation
0	1	1st byte of opcode from queue
1	0	Empty the queue
1	1	Subsequent byte from queue

$\overline{\text{S0}}$,$\overline{\text{S1}}$,$\overline{\text{S2}}$(Output) : Pin nos. 26,27,28.status signals. These signals are connected to the bus controller Intel 8288.The bus controller generates memory and I/O access control signals. Table for status signals is:

$\overline{\text{S2}}$	$\overline{\text{S1}}$	$\overline{\text{S0}}$	
0	0	0	Interrupt acknowledge
0	0	1	Read data from I/O port
0	1	0	Write data into I/O port
0	1	1	Halt
1	0	0	Opcode fetch
1	0	1	Memory read
1	1	0	Memory write
1	1	1	Passive state.

$\overline{\text{LOCK}}$(Output) : Pin no. 29.It is an active LOW signal. When it is LOW all interrupts are masked and no HOLD request is granted. In a multiprocessor system all other processors are informed by this signal that they should not ask the CPU for relinquishing the bus control.

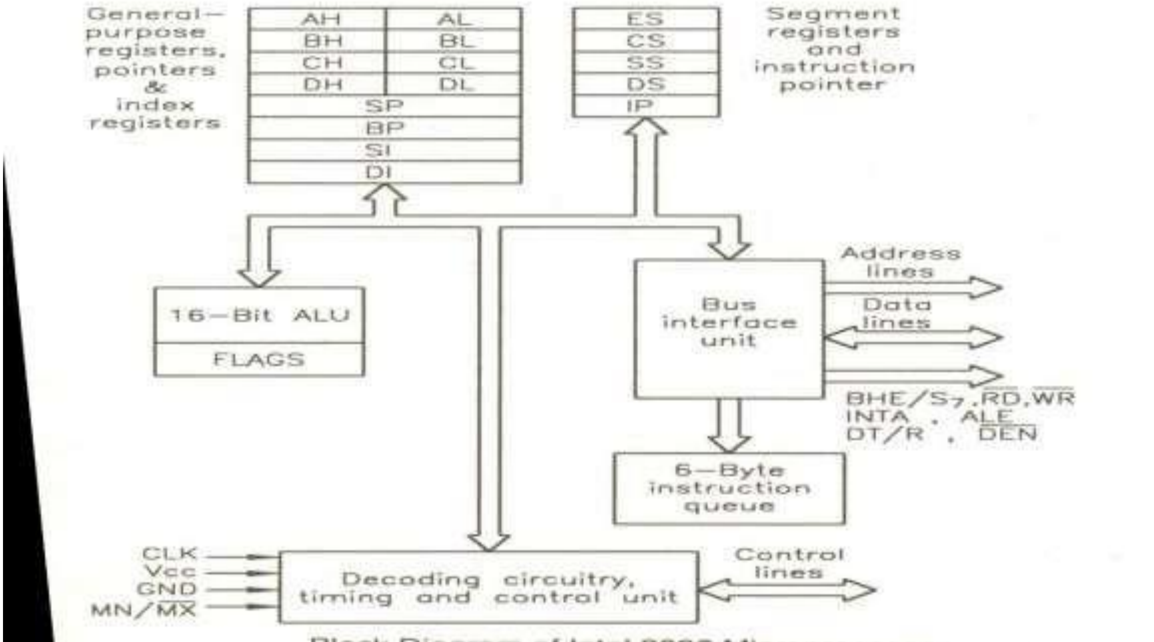
$\overline{\text{RQ}}$ / $\overline{\text{GT1}}$, $\overline{\text{RQ}}$ / $\overline{\text{GT0}}$ (Bidirectional) : Pin no. 30,31. Local bus Priority control. Other processors ask the CPU through these lines to release the local bus. **RQ / GT1** has higher priority than **RQ / GT0**

<u>FUNCTIONAL UNITS OF 8086</u> :

The 8086 contains two functional units: a bus interface unit (BIU) and an execution unit(EU). The general purpose registers, stack pointer, base pointer and index registers, ALU, flag register(FLAGS), instruction decoder and timing and control unit constitute execution unit(EU).

The segment registers, instruction pointer and 6-byte instruction queue are associated with the bus interface unit(BIU).

BLOCK DIAGRAM OF 8086:



Block Diagram of Intel 8086 Microprocessor

REGISTERS OF 8086 : The Intel 8086 contains the following registers:

a) General Purpose Register
b) Pointer and Index Registers
c) Segment Registers
d) Instruction Registers
e) Status Flags

Introduction to VMC-8609 (8086 Microprocessor Trainer Kit)

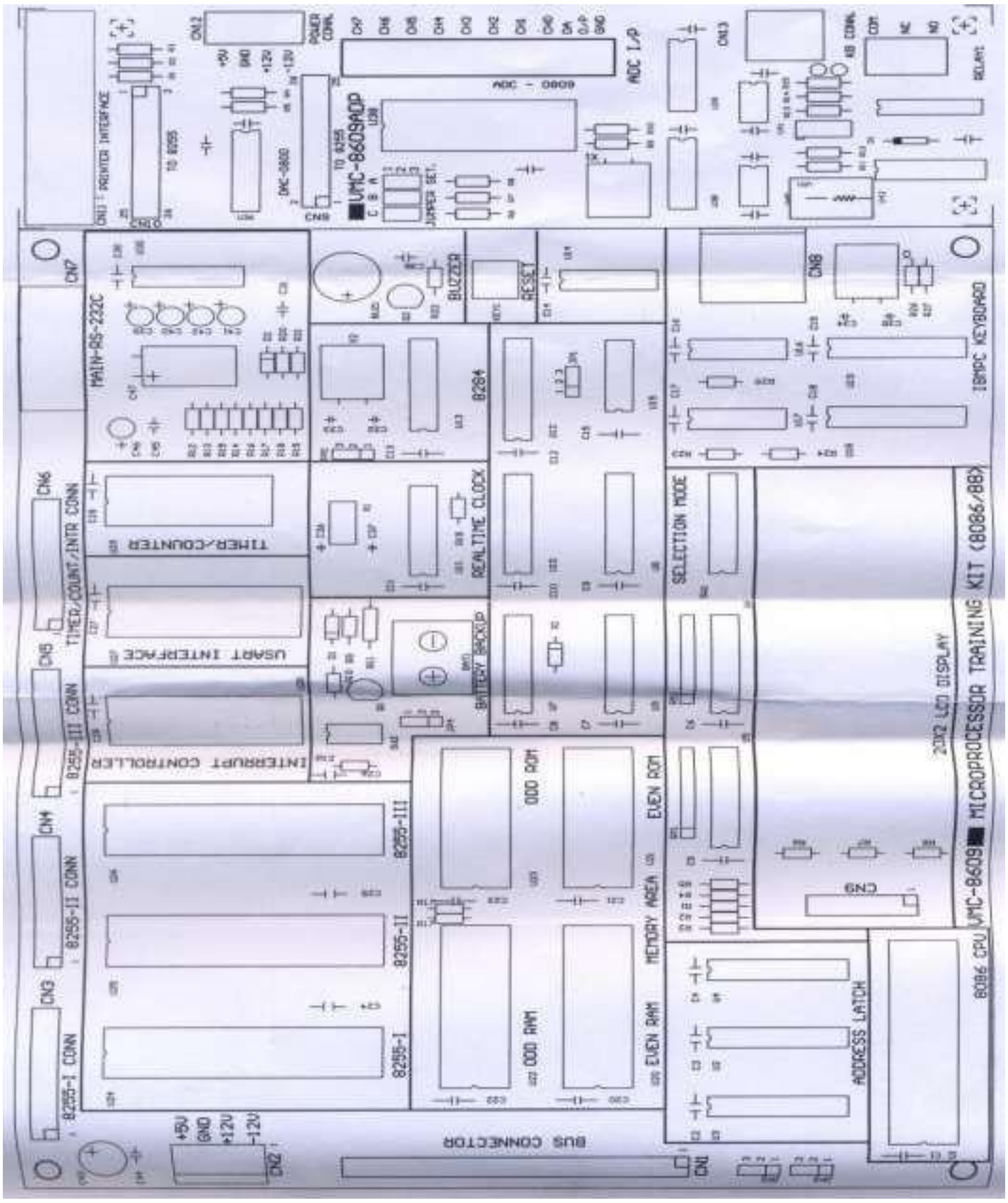
General Description:

VMC-8609 is a single board Microprocessor Training/ Development kit. Configured around the INTEL-8086 microprocessor. This kit can be used to train engineers, to control any industrial process and to develop software for 8086 system.

The kit has been designed to operate in the Maximum or Minimum mode. Co-processors 8087 or 8089 can be added if required. The 8086 can also be replaced by 8088 microprocessor.

The kit communicates with the outside world through an IBM PC compatible keyboard and LCD display.

VMC-8609 is packed up with powerful monitor in 32KB of factory programmed EPROMs and 32KB of RAM for user. This memory can be expanded up to 256KB each. The system has 72 programmable I/O lines. The serial I/O communication is made possible through 8251.



Component Side Layout of VMC-8609 trainer kit

For control applications three 16-bit timer/counters are available through 8253. For real time applications, 8 level of interrupt are provided through 8259. VMC-8609 provides onboard battery backup for RAM. This saves the user program in case of power failure.

The onboard resident SYSTEM MONITOR SOFTWARE is very powerful. It provides various software commands like BLOCK MOVE, INSERT, DELETE, FILL etc. which are helpful in debugging/ developing software. An onboard Assembler/ Disassembler is also provided on VMC-

8609. The kit also supports MASM.VMC-8609 also has onboard buzzer for self testing of hardware and software. This kit is also provided with a Centronix Printer port to take out the prints of the program written in the RAM of kit. A Real-Time Clock is provided onboard for real time applications.

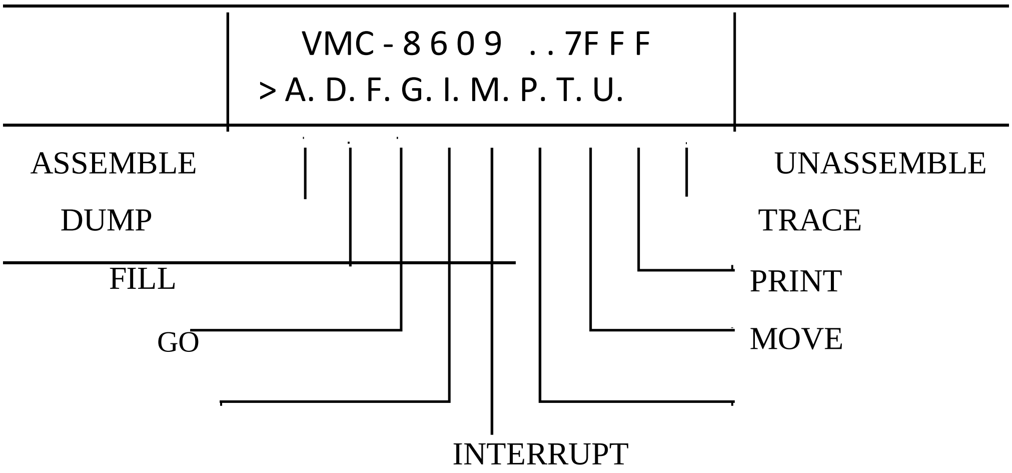
OPERATION:

The operation of the kit is very simple. The operation instructions will be displayed when the device is being switched ON or RESET.

After power on the system, it will display as follows:

```
      VMC-8609
  ENTER  RETURN  KEY . . .
```

After pressing [ENTER], the operating commands will be displayed:



COMMAND DESCRIPTION:

1) [A] ASSEMBLE : Press the key [A] and the LCD is shown as:

```
    VMC - 8 6 0 9 . . 7F F F
A
```

Now the user enters the segment address and effective address simultaneously as follows:

```
    VMC - 8 6 0 9 . . 7F F F
A 0 0 0 0 : 0 4 0 0
```

Now press [ENTER] key, the effective address will appear

```
0 4 0 0
```

From now onwards user can type the program in assembly language as shown below:

```
0400MOV BL,C0
```

Pressing [ENTER] from here will store the instruction at the effective address specified in the instruction and display will show next address. Then next instruction can be typed and stored in same manner. The sequence will continue till the end of the particular program.

2) [D] DISPLAY OR MODIFY THE RAM'S CONTENT:

Press [RESET] and [ENTER] to return to the command menu. Select [D] from this menu the display will show as follows

```
    VMC-8609 ..7FFF
D
```

Type the segment address and offset address of the desired location you wish to modify

```
    VMC-8609..7FFF
D1234:0300
```

(NOTE: Segment address will be taken as 0000H until and unless specified.)

Press [ENTER], the display will show the content of RAM as shown below:

Segment Base	Offset Address	Content of total 8 bytes of RAM

```
1234:0300B3C09A78
        F000F0BB
```

From here on content of any RAM location can be modified. UP/ DOWN and RIGHT/ LEFT arrow keys can be used to go to next or previous memory location.

3) [F] FILL CONSTANT DATA INTO RAM:

Press [RESET] and [ENTER] to return to the command menu. Select [F] from this menu the display will show as follows:

```
    VMC-8609 ..7FFF
F
```

Type the address and data in the manner shown below:

```
    VMC-8609 ..7FFF
F0000:04000450 57
```

Segment	Starting	Ending	Constant
Base	Address	Address	to be filled

Press [ENTER] and the value 57H will be stored at offset 0400H to 0450H.

4) [G] GO FOR RUN/ EXECUTION:

Press [RESET] and [ENTER] to return to the command menu. Select [G] from this menu the display will show as follows:

```
    VMC-8609 ..7FFF
G
```

Type the Segment Address and Offset Address of the starting point of the program. Press [ENTER] [F7] and [ENTER] again thereafter in order to RUN the program. The code will run and the end results will be stored at the memory locations specified in the program which can be verified using [D] command.

5) [I] INTERRUPT:

Press [RESET] and [ENTER] to return to the command menu. Select [I] from this menu the display will show as follows:

```
    VMC-8609 ..7FFF
INTP:0000.0000.0000
```

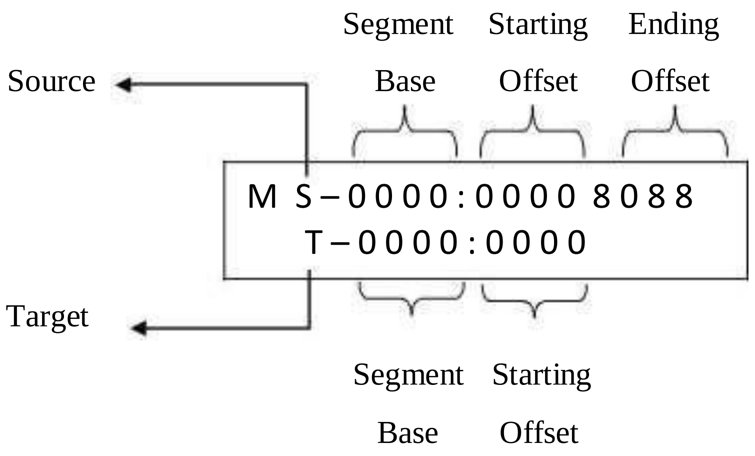
First Int.	Second	Third Int.
Offset	Int. Offset	Offset
Address	Address	Address

Three INTERRUPTs (Offset Addresses) can be set in for the program execution, the CPU will continuously make a single step subprogram for checking IP values. When the IP registers has the same value as the INTERRUPTs addresses, it will enter the INTERRUPT's subprogram.

6) [M] MOVING DATA:

The command MOVE is used to move data in the memory from a specified address to another address by inputting the starting address, the ending address and the desired target address. A return key is then used to execute the changes.

Press [RESET] and [ENTER] to return to the command menu. Select [M] from the menu, the display will show as follows:



```
M S–0000:0000 8088
   T–0000:0000
```

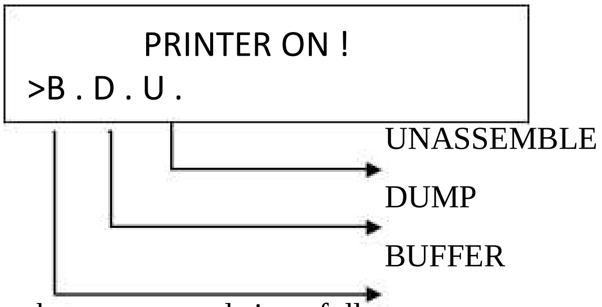
7) [P] PRINT:

This command allows the user to print the output. Printer to be connected to the I/O port of 8255.

Press [RESET] and [ENTER] to return to the command menu. Select [P] from the menu, the display will show as follows:

```
   PRESS  S / F  FOR  SPEED
```

If the printer is not connected at this time, press key [P] so the LCD would show 'PRINTER ERROR!' in the second line. Here S/F indicates 'SLOW' or 'FAST'. Now according to the printer, press 'S' or 'F' key, then following display will be shown on the screen.



```
        PRINTER ON !
>B . D . U .
```

Description of the above commands is as follows:

[D]…..command will allow the printer to print the machine code.

[U]…..command will allow the printer to print the assembly program.

[B]…..command sends data of RAM directly to the printer without going through any modification.

The starting address and ending address need to be entered first before using commands [B], [D] and [U]. Followed by an enter or ARROW UP key in order to print the output.

8) [T] TRACE PROGRAM (AN N-STEP DESIGNED COMMAND):

This command is used for program execution in any desired number of steps. TRACE will enter the interrupt subprogram every time the program is executed. N has a decimal range from 1-99 with 10 as the rounding off number and only operates if N is not 0; otherwise it will clear the function.

After feeding the program, Press [F7] key, then menu display will show as follows:

```
040E
>A.  D.  F.  G. I.  M. P.  T.  U.
```

Press the key [T], the screen displays as follows:

```
040E
T      00-STEP
```

Here user can enter any number of steps from 00 to 99 and start execution using [G] command. For example, 01 is entered in place of 00 in the above command the program will be interrupted after execution of every instruction and if 02 entered in place of 00 then the program will be interrupted after execution of every two instructions and so on.

9) [U] UNASSEMBLE:

The UNASSEMBLE command decodes the value of a group of memory location mnemonics and displays on the display.

Press [RESET] and [ENTER] to return to the command menu. Select [U] from the menu, the display will show as follows:

```
    VMC-8609..7FFF
U0400
```

Here 0400 is the address from where the assembly language code will be unassembled. Press [ENTER] and the display will show as:

Effective Address   Machine Code

```
0000:0400     B0 30
MOV AL,30
```

Assembly mnemonics

User can use UP/ DOWN arrow keys to go to previous or next address respectively.

MEMORY MAPPING:

Memory Locations	**Usage**
0000:0000 – 0000:FFFF	RAM AREA (Odd and Even RAM)
F000:0000 – F000:FFFF	ROM AREA (Odd and Even ROM)

RAM MEMORY

Memory Locations	**Usage**
0000:0000	Interrupt Vector Section (INT1, INT2, INT3) have the arranged Interrupt Section and Stack Segment.
0000:0390	Buffer
0000:039B	System Data
0000:93E0	Buffer (Only if needed)
0000:0400 – 0000:7FFF	User RAM Area

I/O MAPPING

Device Name	**Port Name**	**Port Address**	**Connector**
8255-I	Port-A	70	CN3
	Port-B	72	
	Port-C	74	
	CWR	76	
8255-II	Port-A	80	CN4
	Port-B	82	
	Port-C	84	
	CWR	86	
	Port-A	10	

8255-III	Port-B	12	CN5
	Port-C	14	
	CWR	16	
8253	Counter 0	00	CN6
	Counter 1	02	
	Counter 2	04	
	Counter 3	06	
Keyboard Latch	Input	20	
	Output	22	
8259	Data Word	30	CN6
	Command Word Register	32	
8251	Data Word Register	50	CN7
	Command Word Register	52	

RESULT:

Basic architecture of 8086 and operation of VMC-8609 Microprocessor trainer kit successfully studied.

EXPERIMENTNO. 15

AIM:

Write a program using 8086 for addition of two 16 bit numbers.

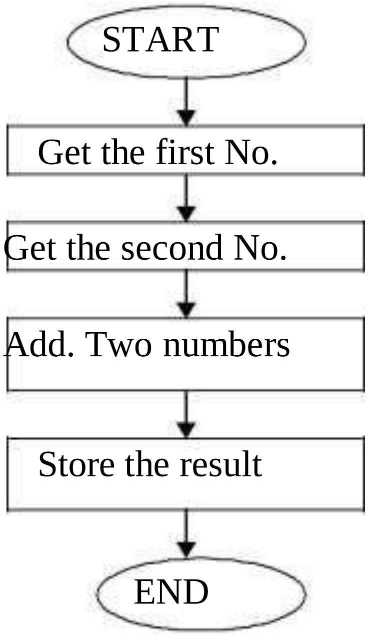
APPARATUS:

8086 microprocessor kit, 5V power supply, Keyboard.

Program:

Memory Address	Machine Code	Mnemonics	Operands	Comments
1000	B8,34,12	MOV	AX,1234	Load 1234 in AX
1003	BA,65,87	MOV	DX,8765	Load 8765 in DX
1006	03,C2	ADD	AX,DX	Add DX with AX
1008	8B,C8	MOV	CX,AX	Move answer to CX
1009	CD,A5	INT A5		Jump to command mode saving all registers.

CIRCUIT DIAGRAM / BLOCK DIAGRAM:-



PROCEDURE:-

1) Switch on the kit.
2) Go to assembler then type the mnemonics of our program in respective memory location.
3) Ensure the input weather placed in respective memory location.
4) Execute the program with starting address.
5) Verify the output data in respective memory location.

INPUT DATA

1000-1234(H)

1001-8765(H)

OUTPUT DATA

AX -9999(H)

PRECAUTIONS:-

Make sure that all the machine codes should be as per specified in the program.

RESULT:-

Thus the addition of two 16 bit numbers has performed and verified.