Experiment - 1

AIM: Familiarization of different keys of 8085 microprocessor kit and its memory map.

APPARATUS REQUIRED: Microprocessor kit alongwith its supply and user manual.

THEORY: Microprocessor 8085 trainer kit is a single board computer using 8085 CPU designed for training and developing applications. 8085 microprocessor kits of different manufacturer's may have different features and specifications.

CPU: The kit has a high performance 8085 CPU operating at 3.072 MHz clock frequency.

DISPLAY: The kit has seven-segment display digits (6). Four digits (Left most) are used for displaying the address of a location or name of a register. Rest of the two digits are used for displaying the contents of the memory location / register.

MEMORY: 8085 microprocessor has 16-bit address bus and 8-bit data bus. Thus a maximum of 64 k bytes of memory (RAM/ROM) can be provided with a 8085 kit.

I/O Mapping: The I/O chips used in the kit are 8279 (keyboard and display Controller), 8255 (Programmable Peripheral Interface - PPI), 8253

(Programmable Internal Timer), and 8259 (Programmable Interrupt Controller). These devices are interfaced with the 8085 CPU in I/O Mapped I/O scheme and have 8-bit address. I/O mapping of these devices are done

KEYBOARD : The kit has 28 keys keyboard. These are 16 hexadecimal keys (0-F), ten function keys, RESET key and SHIFT key. The function of these keys is as under:-

KEY	FUNCTION
RESET	- Reset the system.
VCT INT	- Hardware interrupt via keyboard, RST 7.5.
SHIFT	- Provides a second level command to keys.
GO	- To execute the program.
SI	- To execute the program in single step mode.
EXREG	- Examine Register; allows user to examine and modify the contents of different registers.
EXMEM	- Examine Memory; allows user to examine any memory location and modify any RAM location.
PRE	- Previous is used as an intermediate terminator in case of Examine Memory.
NEXT	- Increment is used as a intermediate terminator in case of Examine Memory, Examine Register etc.

"."	- Terminator is used to terminate the command and write the data in data field at the location displayed in address field.
DEL	- Delete the part of program or data, with relocation by one or more bytes.
INS	- Inserts the part of the program or data, with relocation by one or more bytes.
B.M.	- Allows user to move a block of memory to any RAM area.
FILL	- Allows user to fill RAM area with a constant.
REL	- Relocates a program written for some memory area and to be transferred to other memory area.
INS DATA	- Inserts one or more data bytes in the user's program/data area.
DEL DATA	- Deletes one or more data bytes from the user's program/data area.
STRING	- finds out the string of data lying at a particular address or addresses.
MEMC	- Memory Compare : Compare two blocks of memory for equality.
0-F	- Hexadecimal keys.

All commands are followed by a set of numeric parameters separated by PREV, NEXT & "." (Execute) to work as delimiters.

Experiment - 2

AIM : Steps to enter, modify data/program and to execute a program on 8085 kit.

APPARATUS REQUIRED : 8085 kit and power supply.

THEORY : In all assembly language programming (ALP) exercises, following procedural steps are followed :

Step 1 : Analyzing the problem : The first steps in writing a program is to analyze the problems carefully for which we want to write the program.

Step 2 : Representing program operations : The sequence of operations used to solve a programming problem is called algorithm. Write algorithm and/or draw flow chart for the program.

Step 3 : Writing ALP : Find out the instructions required for each part of the program and finally write the ALP program.

Step 4 : Machine Codes : With the help of instruction table, assembly language program is manually converted into machine codes.

Step 5 : Loading / Entering the Program : The hex codes of the program are to be entered into RAM of the kit. Steps to enter program in the kit are :

- Press RESET key.
- Press EXMEM key.
- Enter Program Starting Address. Entered address will be displayed in the address field of the display.
- Press NEXT key. The data field will shows the memory contents.
- Enters the Hex code of the displayed memory location. Entered hex code will be displayed in the data display field.
- Press NEXT key. Address field will show the next memory address.
- Repeat above two steps until last hex code is filled.
- Press . (Dot) key.

Step 6 : Execute the Program : After the entering the program in 8085 kit memory, it can be executed by following steps :

- Press RESET key.
- Press GO key
- Enter Program starting Address
- Press . (DOT) key.

Step 7 : Checking the Result : After the execution of the program, the next step is to see the result. The result may be in a memory location or in a CPU register. To view memory contents, following steps are executed :

- Press EXMEM key.
- Enter Address of the memory location where

result lies.

- Press NEXT key. Contents of the memory location are shown on the data field of the display.
- Press NEXT key / PRE key to see contents of NEXT / Previous Memory location.
- Press . (DOT) key to end examine - memory command.

PROCEDURE :

1. Connect the power supply to the 8085 Kit and switch on the Power supply.
2. Enter the program.
3. Execute the program.
4. Check the result.

PROGRAM OUTPUT :

After execution of the program, the registers A and C contents are 08 H.

Experiment - 3

AIM: Writing and execution of ALP for addition and subtraction of two 8-bit numbers.

APPARATUS REQUIRED : 8085 kit and power supply.

THEORY : Problem is analyzed and assembly language program for the problems is written using the instructions MOV, LXI, INX, ADD, SUB, STA and HLT.

PROGRAM for Addition of two 8-bit Numbers (Result 8 bit) :

The first number 48H is in memory location 2401.
The second number 56H is in memory location 2402.
The result is to be stored in memory location 2403.

The two numbers which are to be added are given in memory location 2401 and 2402. The instruction LXI H, 2401 will place the address of 1st number (2401) in the H-L register pair. Its machine code is 21, 01, 24, 21 is the machine code of instruction LXI H. The instruction MOV A, M will place the contents of memory addressed by H-L register in the accumulator i.e. the 1st number 48H is placed in the accumulator. The machine code for the instruction MOV A, M is 7E. Now since the H-L register pair contains the address 2401, the instruction INX H will add 1 to the contents of H-L register pair i.e. H-L pair now will contain 2402. The machine code for STA is 32. After the execution of a program, the contents of memory location 2403 can be examined which will contain the result of addition 9EH.

Program for subtraction of Two - 8 bit Numbers:
1st number 48H in memory location 2401.
2nd number 33H in memory location 2402.
result is to be stored in memory location 2403.
The first instruction LXI H, 2401 H will load the address (2401) of the first number in the register pair H-L. Its machine code is 21,01,24. The instruction MOV A,M will move the first number in the accumulator. Its machine code is 7E. The instruction INX H (Machine code 23) will increase the contents of H-L register pair by 1 i.e. H-L register pair now contains 2402. The instruction SUB M (Machine code 96) will subtract the contents of that memory location whose address is in the register pair H-L from the contents of accumulator. So the 2nd number is subtracted from the first number and the result is placed in accumulator. The instruction INX H (Machine code 23) will increase the contents of H-L pair by 1. After this instruction the H-L register pair will contain the address 2403. The next instruction MOV M, A (Machine code 77) will move the contents of the accumulator to the memory location whose address is in the H-L register pair. So, the contents of the accumulator are copied in the memory location 2403. The content of the memory location 2403 can be examined

which will contain 15 H.

RESULT :

$$\begin{array}{r} 48\,H \\ -\,33\,H \\ \hline 15\,H \\ \hline \end{array}$$

PROCEDURE :

1. Connect the power supply to the 8085 and switch on the power supply.
2. Enter the program and the input data.
3. Execute the program.
4. Check the result.

Experiment - 4

AIM : Writing and execution of ALP for multiplication and division of two 8-bit numbers.

APPARATUS REQUIRED : 8085 kit and power supply.

THEORY : Multiplication of two numbers can also be done by repetitive addition method. Division can be performed by the method of repeated subtraction method.

ALP for MULTIPLICATION :

1st number 82 H is placed in memory 2401 H, 2nd number 14 H in memory location 2402 H. The product is of 16 bits and is to be stored in memory locations 2403 H and 2404 H.

The instruction LDA 2401 H, loads first number in register A. MOV B,A instruction copies that number in register B and instruction LDA 2402 H stores 2nd number in A. Instruction MVI D,00H is used to clear the contents of D register as DE is to be used to store the number which is added to product (HL register) repeatedly. Next six instructions are used to store larger number in E register and smaller number in A register as counter. LXI H,0000H is to store initial product 0000 it in register pair HL. ORA A is used to check whether contents of A (smaller number) is zero or not. If zero then multiplication is zero over. Instruction DAD D adds DE pair contents to HL register pair. Register A is decremented by 1. Instruction JNZ loop will transfer the control back to DAD D instruction till number 82 H is not added to HL register 14 H times. After the execution of the program, the contents of memory location 2403 H and 2404 H will be 28 H and 0A AH respectively as 82H (130D) × 14H (20D) = 0A 28H (2600 D).

ALP for Division:

Instruction LDA 2501 H stores divisor in register A. ORA A instruction OR the A with itself. Contents of A remains same, but flags are affected. Zero flag sets if A = 0. If zero flag not set, then program jumps to label Ahead otherwise quotient C = FF H and program control jumps to label over. Divisor in A is copied to register B. Divident is copied in A by instruction LDA 2500 H. Register C is initialized to 00H as initial quotient. Instruction CMP B, compares A (divident) with B (divisor). carry sets if A<B and in that case next instruction JC OVER transfers program control to the instruction with label OVER. Otherwise, divisor (B) is subtracted from divident (A) and C (quotient) is incremented by 1. Program control is transferred back to instruction with Label Loop by the instruction JMP Loop. Now remainder is left in ~~A~~ and quotient is in C. Both are stored at memory

locations 2502 H and 2503 H will be 0AH and 06 H respectively as 2H (130D) ÷ 14H (20D) = 06 H (06D) as quotient and 0AH (10D) as remainder.

PROCEDURE :

1. Connect the power supply to the 8085 kit and switch on the power supply.
2. Enter the program and the input data.
3. Execute the program.
4. Check the result.

Experiment - 5

AIM: Writing and execution of ALP for arranging numbers in ascending / descending order.

APPARATUS REQUIRED:
8085 kit and power supply.

THEORY:
There are different methods to sort a series of numbers. Method to arrange in descending order is similar to arranging the numbers in ascending order. In ascending order, the contents of accumulator and memory location are interchanged when accumulator contents are greater than contents of memory location. But in descending order case, accumulator and memory location contents are interchanged when accumulator contents are less than contents of memory location.

ALP for Arranging numbers in Ascending order:
Instruction LXI SP, 2500 H is used to initialize the stock pointer so that stack area can be used to store data temperately. LXI H, 2050 H and MOV C, M instruction are used to load register with length of data series. INX H increment register pair HL by one so that it now points to first data memory location. DCR C instruction decrements contents of C. First data is copied into register A by

instruction MOV A,M. Instruction MOV B,C copies data of C into register B. Contents of HL register are saved on stack by instruction PUSH H so that after end of each inner loop round, sorted data can be stored at its position and HL contents are fit for next round. INX H and CMP M Instructions compare next memory location data with accumulator and JC Ahead causes jump to instruction with label Ahead in case there is carry (A < Memory contents). Instruction MOV D,M; MOV M,A and MOV A,D are executed when there is no carry. These instructions interchanges the contents of A and memory location so that A has smaller number.

Register B is decremented by 1 and if result not zero then comparison with next memory data is repeated. If zero then contents of HL register saved on stack are taken back and accumulator data is saved at the memory location provided by HL with the help of instructions POP H and MOV M,A. Instruction INX H increments HL by one for next round of comparisons which starts from 2nd number in the series of previous round. Instruction DCR C and JNZ LOOP2 are used to repeat the process till all number are arranged. So sorted numbers will be there at memory locations starting from 2051 H after the execution of the program.

ALP for Arranging numbers in discending order:
Suppose length of the series (N) is stored at location 2050 H and the numbers are stored at memory locations starting from 2051 H.

PROCEDURE -
1. Connect the power supply to the 8085 kit and switch on the power supply.
2. Enter the program and the input data.
3. Execute the program.
4. Check the result.

Experiment - 6

AIM: Writing and execution of ALP for 0 to 9 BCD counters (up/down) counter according to choice stored in memory).

APPARATUS REQUIRED:
8085 kit and power supply.

THEORY: Zero to nine BCD up counter counts from 0 to 9 with some delay between the two steps and then process repeats. The down counter counts from 9 to 0 with delay in steps and then repeats. For delay between two steps, two-three register delay can be used and the delay time depends upon the delay program instructions and delay count used.

PROGRAM: PPI 8255 is connected to 8085 microprocessor as an I/O device with Port A, Port B, Port C and Control Register Addresses 00H, 01H, 02H and 03 H respectively. Let counter output is to be sent by port A of the 8255. choice for up/down counter is stored at memory location 2050 H. The choice is 00H for up-counter and a non-zero data for down counter. Delay counter is stored at memory location 2051 H.
Instruction LXI SP, 2500 H initialize stack pointer. Instructions MVI A, 80H and OUT 03H are used to initialize control register of 8255

so that Port A functions as output port in I/O mode. Instructions LDA 2050 H transfers choice to register A. Instruction ORA A OR's the contents of A with A. Content of A remains unchanged but flags are affected to reflect result. If contents are not zero, means choice is for down counter and instruction JNZ DOWN causes jump to instruction with label DOWN. Otherwise choice is for up counter. MVI A, 00H places initial count of up counter (00H) in register A.

OUT 80H causes transfer of A contents to Port A. CALL Delay calls Delay subroutine to provide time gap. INR A instruction increases count by 1. Now instruction CPI 0AH compares contents of register A with immediate data 0AH (10D), as on increment, data in A become 10D from 9D. If not zero i.e. A ≠ 0A H, jumps back to LOOP 1 otherwise to LOOP 2. Rest of the program similarly work as down counter.

Suppose 'd' data is ~~used~~ stored at location 2051 H for delay count. Total number of T states required to execute the delay subroutine are

No. of T states: $12+12+13+(7+(7+(4+4+4+4+10)\times 255-3+4+10\times 231-3+4+10)\times d+10+10+10 = 62 + 1535706\,d = 1535706\,d$

(C-3 is taken as in conditional jump instructions, 7T states are used when condition is false and 10T states when condition is True).

With a crystal frequency 6.144 MHz (clock frequency is half of crystal frequency) the delay subroutine will produce a delay of 0.5 d sec. The required delay between the two counts can be adjusted by storing suitable data at memory location 2051 H.

PROCEDURE :

1. Connect the power supply to the 8085 kit and switch on the power supply.
2. Enter the program and the input data.
3. Execute the program.
4. Check the result.

Experiment - 7

AIM: Interfacing exercise on 8255 like LED Display Control.

APPARATUS REQUIRED:
8085 kit, power supply.

THEORY: 8255 is a programmable peripheral interface device used for parallel data transfer. It can be programmed to transfer data under various conditions, from simple I/O to interrupt I/O. It is flexible, versatile and economical. It is an important general purpose I/O device that can be used with almost all microprocessor.

The 8255 is available as 40 PIN DIP package. It has 24 I/O pins. It has three 8 bit ports, Port A, Port B and Port C. The 8 bits of port C can be used as individual bits or can be grouped in two 4 bit ports, C_U (upper) and C_L (lower) ports.

Control word is written in the control register of 8255. The second word written in the control register specify the I/O function for each port. The control register can be accessed to write a control word when address lines A_1 and A_0 are at logic 1.

LED & Switch interfacing with 8255:
On board 8255 ports pin are available at

the FRC connector and ZIF sockets. External components/devices can be connected with the on board 8255 through these connectors. For example, four LED and one switch can be connected with 8255 PPI through ZIF socket. Here Port A is to function as input port and port B as output port. Hence the control word for this is 90H.

PROGRAM :

The program is to control ON-OFF of the LED's through switch SWI.

First two instructions will initialize the 8255 with port A as input port and port B as output port. Instructions IN 00H will read the switch position (A_7 bit) and to check this bit. Accumulator contents are ANDed with immediate data 80H. Result will be zero if switch is OFF and non-zero if switch is ON. If the result is zero, instruction JZ OFF will cause jump to label OFF and 00H will be sent to port B otherwise 0FH data will be sent to port B. The program will again jump to instruction with label REP to repeat this process.

PROCEDURE :

1. Connect the power supply, LED's and switch as to the 8085 kit and switch on the power supply.
2. Enter the program.
3. Execute the program.
4. Change switch program and observe the LED's.

EXPERIMENT-8

AIM : Interfacing exercise on 8253 PIT.

Apparatus Required : 8085 kit and power supply

Theory : The 8253 is a programmable interval timer and can be programmed to generate accurate time delays under software control. The use of this chip avoids the timing loops in the system software and provides multiple time delays. This chip also provides counter/timer function apart from generation of delays.

The 8253 is available in 24 Pin DIP package. It is a programmable interval timer. It has three independent 16 bit counters. It can operate up to 2 MHz. It generates accurate time delays and can be used for applications such as a real time clock, an event counter, a square wave generator, hardware/software triggered strobe signal etc. The 8253 operates in the following six modes.

(*i*)	Mode 0	Interrupt on terminal count
(*ii*)	Mode 1	Programmable one shot
(*iii*)	Mode 2	Rate Generator.
(*iv*)	Mode 3	Square Wave Generator.
(*v*)	Mode 4	Software Triggered strobe
(*vi*)	Mode 5	Hardware triggered strobe.

8253 Control Word

SC_1	SC_0	RL_1	RL_0	M_2	M_1	M_0	BCD
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

BC (D_0)

0	Binary counter 16 bits
1	Binary coded decimal counter

RL-Read/Load (D_5–D_4)

RL_1	RL_0	Read/Load
0	0	Counter latching operation
0	1	Read/load least significant Byte only.
1	0	Read/Load most significant byte only.
1	1	Read/Load LSB first and then MSB

M-Mode Select (D_3–D_1)

M_2	M_1	M_0	Mode Select
0	0	0	Mode 0
0	0	1	Mode 1
×	1	0	Mode 2
×	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

SC-Select counter (D_7–D_6)

SC_1	SC_0	Select counter
0	0	Select counter 0
0	1	Select counter 1
1	0	Select counter 2
1	1	Illegal

The functions of the counters of 8253 can be programmed by the system software. The control word decides the selection of the counter, mode of operation, loading sequence of the count and the selection of the binary or BCD counting. When the control word is written into the control word register, the desired counter is selected, mode of operation is set and the loading sequence of the count and the selection of binary or BCD counting are defined.

Program : Connect 2 MHz clock at CLK2 input through D-flip flop and +5v at Gate 2 input of the PIT 8253 .Then for programming the PIT 8253 to produce a square wave of frequency 20 Hz at the output OUT 2 of PIT 8253 counter 2 :

Counter Count: The required square wave frequency = 20Hz. Hence, Time Period $= \frac{1}{20}$ sec = 50 msec.

The D-flip flop divides the input clock frequency by 2. Thus the clock frequency input to counter 2 of 8253 is 1MHz and the time period is 1 μsec. Thus, the counter count is

$$\text{Count} = \frac{50\,m\text{sec}}{1\,\mu\text{sec}}$$

$$= 50000\text{ D} = \text{C350H}$$

Control Word: For square generation, the counter 2 should operate in Mode 3. The counter starts counting as soon as count is loaded into the counter.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
1	0	1	1	0	1	1	0	B6H
Select Counter 2		Load 16 bit Count LSB first then MSB		Mode 3			Count in Binary	

Address	Hex Code	Label	OPCODE	Comments
2000	3E, B6		MVI A, B6H	Load control Word
2002	D3,13		OUT 13 H	in the control Register
2004	3E,50		MVI A, 50H	Load count LSB
2006	D3 ,12		OUT 12 H	in counter 2
2008	3E , C3		MVI A, C3H	Load count MSB
200 A	D3, 12		OUT 12 H	in counter 2
200 C	C3, 0C, 20	REP	JMP REP	Wait
200 F	76		HLT	Stop

The first two instruction of the program will initialize counter 2 of the 8253 in Mode 3 binary counter. Next two instruction will load the count LSB and then next two instruction count MSB byte in the counter register of counter 2. As soon as the count is loaded into coonter2, it starts its operations and we will get square wave at the OUT2 output of the PIT 8253.

PROCEDURE :

1. Do the connections as required for the CLK2 and GATE2 input of the PIT 8253.
2. Connect the supply to the kit.
3. Enter the program.
4. Execute the program .
5. Observe the square wave output at OUT2 output of the PIT.

EXPERIMENT-9

AIM : Interfacing exercise on 8279 programmable KB/display interface to display the hex code of key pressed on display.

APPARATUS REQUIRED : 8085 kit and power supply

THEORY : The 8279 programmable keyboard/Display interface is a hardware approach to interface a matrix keyboard and a multiplexed display with microprocessor. A considerable amount of time of the microprocessor is saved by using 8279 in checking the keyboard and refreshing the display. In a software approach to interface microprocessor with keyboard and multiplexed display, microprocessor remains busy for a lot of time in checking the keyboard and multiplexed display.

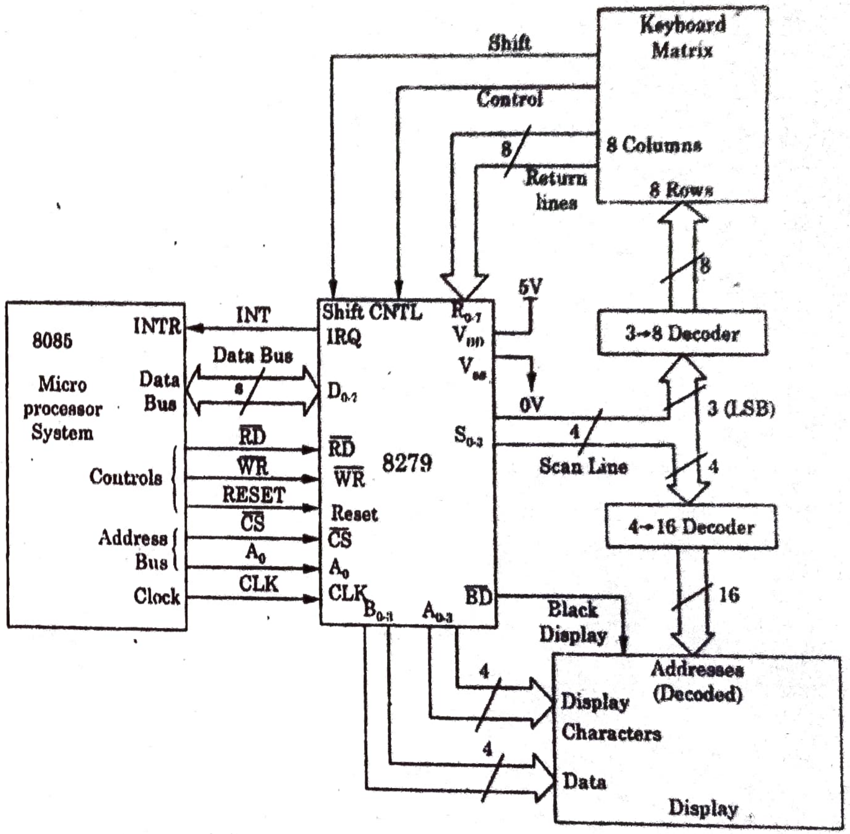
The 8279 can operate without involving the microprocessor, except for relatively short time when actual data transfer between I/O devices and CPU takes place. It relieves the CPU from the normal house keeping job with I/O devices. 8279 can perform two important functions.

(*i*) It scans a keyboard, detects any key press, and transmits to the CPU information.

(*ii*) It puts out data received from the CPU for use of display devices.

The 8279 is a 40 Pin device and has two major sections–keyboard and display. The keyboard section can be connected to a 64 key matrix in a number of modes. The display section can provide a 8 or 16 seven segment LED displays. It can be also used in other types of displays *i.e.* incandescent and other popular display technologies. It has 16×8 display RAM, which can be used to read/write information for display purposes. The display can be set up in either right entry or left entry format.

A matrix keyboard (8×8: maximum) and a display (sixteen seven-segment: maximum) can be interfaced with 8085 microprocessor through 8279 as shown in figure.



Control words : (*i*) Control word for Mode setting : Control word to set the keyboard/display mode.

MSB							LSB	
0	0	0	D	D	K	K	K	
0	0	0	0	0	0	0	0	= 00H
Fixed code			8 8-bit character display – left entry		Encoded scan keyboard – 2 key lockout			

(ii) Control Word for Program Clock: All timing and multiplexing signals for the 8279 are generated by an internal prescaler. This prescaler divides the external clock (at pin 3) by a programmable integer.

MSB							LSB	
0	0	1	P	P	P	P	P	
0	0	1	1	1	1	1	1	= 3 F H
Fixed Code			To divide clock by 31					

Reset signal also sets the prescaler to 31 hence may be avoided.

(iii) Control Word to Read FIFO RAM: The control word to read scan code of pressed key which is stored in FIFO RAM is

0	1	0	AI	X	A	A	A	
0	1	0	0	0	0	0	0	= 40 H
Fixed Code			For Scan keyboard Mode					

The format of data read will be

MSB							LSB
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CNTL	SHIFT	Row			Column		

In Hex keyboard, the keys are placed such that when CNTL & SHIFT keys are not pressed, the read data is the hex code of the pressed key.

(iv) Control Word to Write Display RAM: CPU sets up the 8279 for a write to Display RAM by first writing this command. All subsequent data writes will be to the display RAM. The RAM address is auto incremented if AI = 1

1	0	0	AI	A	A	A	A	
1	0	0	1	0	1	0	0	= 94 H
Fixed Code			For Auto increment	To select fifth Display from left				

PROGRAM

Address	Op-code	Label	Mnemonics	Comment
2000	3E,00		MVI A, 00H	Initialize the 8279 by
2002	D3, 19		OUT 19 H	loading control word
2004	3E,40	REP	MVI A, 40 H	Initialize the FIFO RAM
2006	D3,19		OUT 19 H	for Read operation
2008	DB,18		IN 18 H	Read FIFO RAM (Hex code of the pressed key) to accumulator
200 A	47		MOV B,A	Copy Hex code in Register B
200B	3E, 94		MVI A, 94 H	Initialize display RAM to
200 D	D3,19		OUT 19 H	Write onto 5th display
200F	78		MOV A,B	Reload Hex code of pressed key in A
2010	D3,18		OUT 18H	Write Hex code in Display RAM
2012	CD,00,25		CALL DELAY	Call display subroutine
2015	C3, 04, 20		JMP REP	Repeat the operation
2018	76		HLT	Stop

First two instruction of the program will initialize the display as 8 8-bit character, left entry and key board as encoded scan, 2-key lockout. Next two instruction will initialize the FIFO RAM for read operation. Instruction IN 18H will read the FIFO RAM and stores the contents(Hex code of the pressed key) in accumulator. Instruction MVI A, 94 H and OUT 19 H will initialize RAM to write onto 5th display. Instruction OUT 18H will write the pressed key hex code onto display RAM and thus 5th display will show the hex code of the pressed key in coded form (For hex code display without coding, lock-up table according to display-common anode/cathode is to be used).

PROCEDURE :

1. Connect the power supply to the 8085 kit and switch on the power supply.
2. Enter the program.
3. Execute the program.
4. Check the result.

EXPERIMENT-10

AIM : Study of 8085 emulator for hardware testing.

APPARATUS USED : 8085 In- circuit Emulator, 8085-kit & supply.

THEORY : In-circuit emulators are used for designing a microprocessor based system (both hardware and software designing). In-circuit emulation is the execution of a prototype software program in prototype hardware under the control of a software development system. For this, the microprocessor is removed from the microprocessor based system under test or to be designed. A 40-pin cable from the In-circuit emulator is plugged into the socket previously occupied by the microprocessor. The emulator performs all the functions of the replaced microprocessor and in addition, it allows the system under test to share all its resources such as software, memory and I/O devices. It provides a window for looking into the dynamic real time operation of the prototype hardware.

To test subsystems (Such as memory, I/O etc), the minimum prototype hardware required is 40 pin microprocessor socket (without the microprocessor) and the power supply. All other resources can be borrowed from the emulator. As more and more hardware is added to the system, fewer resources from emulators will be required.

Emulator can be a stand-alone unit or a part of software development system. A small program can be directly entered into the emulator or can be transferred into emulator from software development system. Once a program is loaded, a user can interact with the emulator through its keyboard.