EE 381: EE LABORATORIES (DIGITAL CIRCUITS AND MICROPROCESSORS)
2025-2026/II
EXPERIMENT 3: FAMILIARIZATION OF 8085 MICROPROCESSOR KIT
(BASIC OPERATIONS)

Introduction

In this experiment we will use the MPS 85-3 8085 Microprocessor kit to write and run simple programs using the Hex Key Pad and Serial modes. You will also use a commercial 8085 cross assembler to generate hex codes for your assembly language programs. You will use the RST7.5 interrupt also.

PART 1 : 8085 PROGRAMMING USING HEX KEY PAD

(Switch 4 of DIP switch in OFF mode)

A) FAMILIARIZATION

Familiarize yourself with the following Monitor Commands of the 8085 kit.

EXAMINE/MODIFY MEMORY
EXAMINE/MODIFY REGISTER
SINGLE STEP
GO

The above Monitor commands are required for operations using the Hex Keypad. If any wrong key is pressed, the LED Display will show “Err”. You will need to press the RESET key to get back to normal mode.

B) PROGRAMMING

Program 1

NORMAL EXECUTION

(a) Write a program to find the largest of the 10 numbers stored in memory locations 8040H to 8049H. Your output of the program (largest number) should be stored in 804AH.

(b) You will do the coding part (i.e. 8085 mnemonics to Hex code) for this program manually. Enter your assembly language program in a tabular form. Instruction set and opcodes are provided in one of the attached sheets. The table should have the following columns:

Memory address	Hex Opcode	Assembly Language Instructions		Comments/Remarks
		Label(if any)	8085 Mnemonics with operands	

The Origin of your program should be 8000H. Enter some data in locations 8040H to 8049H. Ensure that the initial data in 804AH is 00H.

(c) Execute your program using GO command from the Hex keypad.

(d) Check whether the largest number has been stored in 804AH.

SINGLE STEPPING MODE

(a) Once again write 00H in location 804FH.

(b) Use Single Step mode and check your program for two or three loops of the program to make yourselves familiar with this mode. Note that you can examine/modify registers as well as memory locations in between the single stepping operations.

PART 2 : MPS 85-3 KIT PROGRAMMING USING PC (Serial Mode)

(*Switch 4 of DIP switch in ON mode)*

A) FAMILIARIZATION

Familiarize yourself with the following Monitor Commands of the 8085 kit in the Serial mode.

D	: display memory command;	G	: command to execute a program
S	: Substitute memory command;	X	: Examine/modify register command

The above Monitor commands are frequently required for serial operations of the kit.
The relevant files are available in the folder DCMP in the desktop.
You will have to run Oracle VirtualBox to create a temporary (virtual) Windows. You will have to edit and assemble the assembly language file using the assembler within this virtual Windows. You will have to exit the virtual Windows and come back to the main Windows for downloading the hex file. The utility program Term853 has to be executed to download the .HEX file to the microprocessor kit.

B) PROGRAMMING

Program 1

(a) Run Oracle VirtualBox program available in the folder DCMP. Click on the Start tab. A virtual Windows will load.
(b) Write your program for finding the largest number as a .TXT file (use NOTEPAD) in the DCMP/ASM folder.
(c) Assemble your file. See the procedure mentioned under the "Assembling" section in the attached sheets. The cross assembler will generate .LST and .OBJ files.
(d) The .LST file gives the original assembly language program, hex codes, and label addresses. Open this file using Notepad and see the opcodes and addresses generated for each line. Check whether the opcodes in this file are the same as you wrote down manually under Part 1.
(e) Viewing the .HEX file: Use the Hex Editor XVI32.EXE in the DCMP\HEXEDIT folder to view all the hex codes in your .OBJ file. Note the extra characters due to the Intel Hex format. Also notice 0D and 0A characters (CR and LF) at the end of every line.
(f) Rename .OBJ file to.HEX and store it in the USB drive to be used later.
(g) Close the temporary (virtual) Windows and return back to the main Windows. Download the .Hex file to the Microprocessor Kit using Term853 software. Execute it and check that you are getting the result as in Part 1.

Modification to Program 1

(a) Make a small modification to the above program such that the result is displayed on the PC screen. You can use the Serial Monitor routine NMOUT for this purpose.
(b) Assemble the modified program, download and execute the same. Check whether the result is displayed on the screen.

Program 2 (Use of RST7.5 Interrupt)

(a) RST7.5 interrupt of the 8085 Microprocessor is provided on the kit in the Hex Keypad – KBINT. Write an RST7.5 ISR which will increment the accumulator each time KBINT is pressed (initial value of A=00H) and display the current value of A on the PC Screen. You will need to use SIM and EI instructions in a main program which runs in an infinite loop. Note that when RST7.5 interrupt comes the monitor will go to the RAM location 8FBFH location. Again, you will need to store a JUMP instruction in 8FBFH location to your ISR subroutine (it is better to do this step by hand assembly). Use the A85.EXE Cross-assembler for the Main program and the ISR.
(b) Observe the display on the PC screen. Is it incrementing properly? Observe the increments. Why this abnormal behavior?

Modification to Program 2

(a) Modify the above program so as to rectify the abnormal behavior observed above. Incorporate some delay in the ISR and then reset the RST7.5 flip-flop inside the ISR, before returning to the main program.
(b) Execute the program again and observe the PC screen. Do modifications as necessary.

SUMMARY OF KEYBOARD MONITOR COMMANDS

COMMAND	FUNCTION/FORMAT
EXAM MEM	Displays/Modifies the content of the memory location. EXAM MEM <address> NEXT [<data>] NEXT/PREV EXEC
EXAM REG	Displays/Modifies the content of the REGISTER. EXAM REG <reg key> [<data>] NEXT* EXEC
SINGLE STEP	Execute a single user program instruction. SINGLE STEP <Start address> NEXT [<Start address>] NEXT* EXEC
GO	Transfers control from monitor to user program GO <address> EXEC

SUMMARY OF SERIAL MONITOR COMMANDS

COMMAND	FUNCTION/FORMAT
C (Compare Memory)	Compare a block of memory with destination block C <Start address>,<End address>,<destination address> <CR>
D (Display Memory)	Display memory contents in line formatted output D<Start address>,<End address> <CR>
G (GO)	Transfers the processor control from the Monitor to user program with optional breakpoints. G [<Start address>], [<breakpoint address 1>,] [<breakpoint address 2>,] [<breakpoint address 3>,] <CR>
M (Move Memory)	Moves a block of memory contents M <Start address>,<End address>,<destination address> <CR>
S (Substitute Memory)	Displays/Modifies memory locations S <address>, /- [[<new address>]]* <CR>
X (Examine/Modify Registers)	Displays/Modifies the processor registers X [<reg>] [[<new address>],] <CR>

Useful Keyboard Monitor Routines accessible to user(in HEX KEY PAD mode)		
Call Address	Mnemonic	Functions
0440H	UPDAD	Updates address field of the display. The contents of the locations 8FEFH&8FF0H are displayed in the address field. All CPU registers and flags re affected. If Reg. B=1, dot at the right edge of the field; If B=0, no dot.
044CH	UPDDT	Updates data field of the display. The contents of the locations , 8FF1H are displayed in the data field. All CPU registers and flags re affected. If Reg. B=1, dot at the right edge of the field; If B=0, no dot.
Useful Serial Monitor Routines accessible to user(in SERIAL mode)		
0C41H	NMOUT	Outputs one byte as two hex digits to the serial I/O device. Input: A=Byte to be Output. Reg A,B,C and flags are affected.
0B5BH	DISPM	Displays a string of characters. The string should be terminated by character Zero which is not output. Input: HL=String start address of the string characters. Reg A,C, H,L and flags are affected

STEPS FOR DEVELOPING, ASSEMBLING AND EXECUTING YOUR PROGRAM IN SERIAL MODE

WRITING/EDITING

Type your assembly language file (.TXT file) using NOTEPAD. You need the following assembler directives.

Assembly Directives	Purpose	Examples
.org <addr>	To define the ORIGIN of your program	.org h'8000, .ORG H'8200
.equ	To define address and data constants	.equ START,h'8040, .EQU output,H'8501
.END	To mark the end of your program	.END, .end

Assembler Line Format, Rules regarding Labels, and Constants

Assembler Line Format	<label>: Mnemonic ;<comments> • If no label, precede Mnemonic with a Tab • Comments are optional	Eg. YYY: Mov a,b ; data into A register Eg. MVI A,h'06
Address Labels	Alphanumerics only, no spaces in between, must end with a colon	Eg. XX: , Loop:
Constants	Must have a prefix such as b or h. If no prefix, number is taken as decimal	b'0101 ;binary number 101 = decimal 5 77 ;decimal number 77 = octal 115 h'ff ;hexidecimal ff = decimal 255

See Assembly program example given at the end.

ASSEMBLING

- Run Oracle VirtualBox in DCMP folder of the Main Windows. Click on Start tab. Wait for the virtual Windows to load. Go to the folder DCMP/ASM in the desktop. Edit a .txt file using Notepad for entering the assembly language program.
- Run the assembler A85.EXE to assemble the program in the .txt file.
- The .Lst and .Obj files will be generated by the assembler.
- Rename .Obj file to .Hex file
- Store .Hex file in a USB drive to be used later for downloading to the Microprocessor Kit.
- Close the virtual Windows to return back to the Main Windows.

DOWNLOADING

- In the Main Windows, go to the DCMP folder in the desktop.
- Run the program Term853 for establishing the serial communication with the Microprocessor Kit.
- Press RESET key on the Microprocessor Kit to ensure the serial communication between the Kit and the PC.
- Click on the Download tab on the command bar of Term853 windows.
- Browse your .Hex file from the USB and click Ok. The download will happen automatically.

EXCECUTING

- To execute the code which was downloaded to the kit, type G
  "GXXX=XX-"will appear on the screen.
- Type the starting address of the program and press <Enter>. The program will be executed.

Example showing the various file formats in the assembly process

Myfile.txt	Myfile.lst	Myfile.obj
.ORG H'8000 .EQU INPUT,H'8500 .EQU OUTPUT,H'8501 LXI H,INPUT MOV A,M CMA INR A STA OUTPUT RST 5 .END	000001 8000 .ORG H'8000 000002 8500 .EQU INPUT,H'8500 000003 8501 .EQU OUTPUT,H'8501 000004 8000 210085 LXI H,INPUT 000005 8003 7E MOV A,M 000006 8004 2F CMA 000007 8005 3C INR A 000008 8006 320185 STA OUTPUT 000009 8009 EF RST 5 000010 800A .END ▯ INPUT =8500 OUTPUT =8501	:0A8000002100857E2F3C320185EF60 :00000001FF

ASCII character set

dec	oct	hex	char	dec	oct	hex	char	dec	oct	hex	char	dec	oct	hex	char
0	000	00	^@ null	32	040	20	sp	64	100	40	@	96	140	60	`
1	001	01	^A soh	33	041	21	!	65	101	41	A	97	141	61	a
2	002	02	^B stx	34	042	22	"	66	102	42	B	98	142	62	b
3	003	03	^C etx	35	043	23	#	67	103	43	C	99	143	63	c
4	004	04	^D eot	36	044	24	$	68	104	44	D	100	144	64	d
5	005	05	^E enq	37	045	25	%	69	105	45	E	101	145	65	e
6	006	06	^F ack	38	046	26	&	70	106	46	F	102	146	66	f
7	007	07	^G bel	39	047	27	'	71	107	47	G	103	147	67	g
8	010	08	^H bs	40	050	28	(	72	110	48	H	104	150	68	h
9	011	09	^I ht	41	051	29	)	73	111	49	I	105	151	69	i
10	012	0A	^J lf	42	052	2A	*	74	112	4A	J	106	152	6A	j
11	013	0B	^K vt	43	053	2B	+	75	113	4B	K	107	153	6B	k
12	014	0C	^L ff	44	054	2C	,	76	114	4C	L	108	154	6C	l
13	015	0D	^M cr	45	055	2D	-	77	115	4D	M	109	155	6D	m
14	016	0E	^N so	46	056	2E	.	78	116	4E	N	110	156	6E	n
15	017	0F	^O si	47	057	2F	/	79	117	4F	O	111	157	6F	o
16	020	10	^P dle	48	060	30	0	80	120	50	P	112	160	70	p
17	021	11	^Q dc1	49	061	31	1	81	121	51	Q	113	161	71	q
18	022	12	^R dc2	50	062	32	2	82	122	52	R	114	162	72	r
19	023	13	^S dc3	51	063	33	3	83	123	53	S	115	163	73	s
20	024	14	^T dc4	52	064	34	4	84	124	54	T	116	164	74	t
21	025	15	^U nak	53	065	35	5	85	125	55	U	117	165	75	u
22	026	16	^V syn	54	066	36	6	86	126	56	V	118	166	76	v
23	027	17	^W etb	55	067	37	7	87	127	57	W	119	167	77	w
24	030	18	^X can	56	070	38	8	88	130	58	X	120	170	78	x
25	031	19	^Y em	57	071	39	9	89	131	59	Y	121	171	79	y
26	032	1A	^Z sub	58	072	3A	:	90	132	5A	Z	122	172	7A	z
27	033	1B	^[esc	59	073	3B	;	91	133	5B	[	123	173	7B	{
28	034	1C	^\ fs	60	074	3C	<	92	134	5C	\	124	174	7C	|
29	035	1D	^] gs	61	075	3D	=	93	135	5D	]	125	175	7D	}
30	036	1E	^^ rs	62	076	3E	>	94	136	5E	^	126	176	7E	~
31	037	1F	^_ us	63	077	3F	?	95	137	5F	_	127	176	7F	del

^ denotes control key simultaneous with character key.