Microprocessor based System Design Laboratory (Gy)

Department of Computer Science and Technology, IIEST Shibpur

Date: September 20, 2024

-

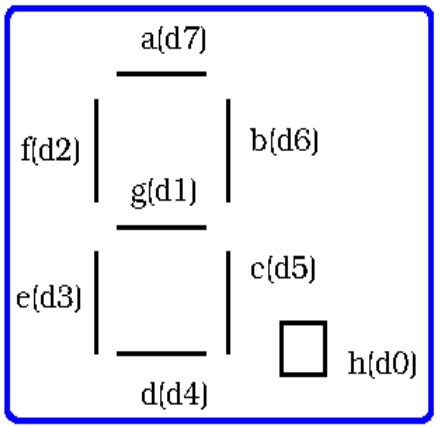
Experiment No. 8: Using 8279 of the SDK (Assembly Language Programming)

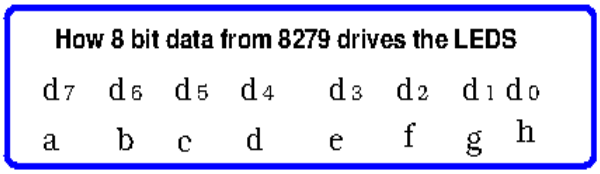
Objective: To write 8085 Assembly Language Programs operating on the 8279 Keyboard-Display Controller of the SDK**.**

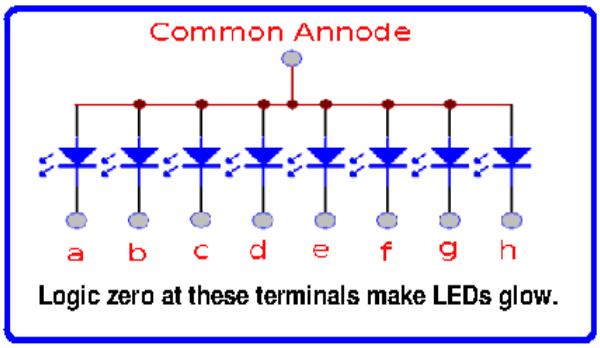
Assignments:

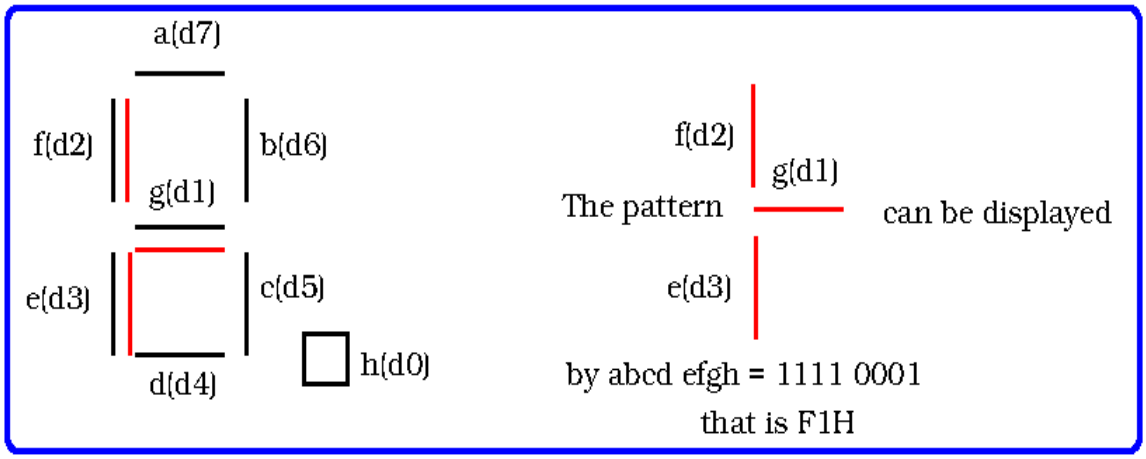
1. Let the **initials** of the names of the group members, concatenated together, give a **string of length 6** (say, 2 letters for each student of a group of 3). Choose **patterns** for 7-segment LED modules, which "look like" the letters of the **string of length 6.** Write a program to display these patterns (six patterns for six letters) in the display section (six 7-segment LED modules) of the SDK, by directly interacting with the **8279 keyboard-display controller** of the kit through IO Port Nos **18H (data port)** and **19H(control port)**. Refer to the following figure and sample program for your reference.

   Modify your program so that the displayed "string" **blinks**.
   Upload both the programs in Google Classroom.

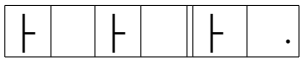






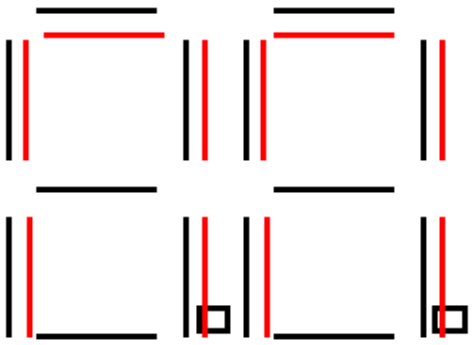


Following program displays the pattern ├ in the 1st,3rd, and 5th modules of the display section with a '.' in the last module. That is the display section would look like:



Label	Assembly Instruction	Memory Address (Hex)	Machine Language (Hex)	Comment
	CALL CLEAR	2100	CD	Monitor routine CLEAR (at address 02BE H clears all the six modules of the display. This can be achieved by making all the LEDs of the modules off (writing 00H).
		2101	47	
		2102	03	
/* Following is for initializing 8279 for display by writing the **control word** 9xH to the control port of 8279 (address 19H). X=0 for displaying pattern to the 1st module, X=1 for the 2nd and, so on. The last bit (**lsb**) of 9H (100**1** in binary) signifies **A**uto **I**ncrement (You may refer to 8279 for validation. That is, every write to the data port (at 18H) of 8279 will automatically set the next modules for the next write.*/				
	MVI A, 90H	2103	3E	
		2104	90	Start display from the 1st module with Auto-Increment **ON**
	OUT 19H	2105	D3	
		2106	19	
/* Every write to the data port (18H) of 8279 will now display the corresponding pattern stating from the 1st module */				
	MVI A, F1H	2107	3E	As explained in the above figure, F1H is the code for the pattern ├
		2108	F1	
	OUT 18H	2109	D3	Output 07H to the data port of 8279 – this will display the pattern ├ in the 1st module.
		210A	18	
	MVI A, FFH	210B	3E	The FFH (1111 1111 in binary should keep all the LEDs off.
		210C	FF	
	OUT 18H	210D	D3	Since the Auto-Increment bit was set in the control word 90, this will make the 2nd Module blank.
		210E	18	
	MVI A, F1H	210F	3E	F1H is the code for the pattern ├
		2110	F1	
	OUT 18H	2111	D3	This will display the pattern ├ in the 3rd module.
		2112	18	
	MVI A, FFH	2113	3E	FFH keep all the LEDs off.
		2114	FF	
	OUT 18H	2115	D3	This will make the 4th module blank.
		2116	18	
	MVI A, F1H	2117	3E	F1H is the code for the pattern ├
		2118	F1	
	OUT 18H	2119	D3	This will display the pattern ├ in the 5th module.
		211A	18	
	MVI A, FEH	211B	3E	FEH (1111 1110 in binary) will glow only the decimal point (h)
		211C	08	
	OUT 18H	211D	D3	This will display '.' in the 6th module.
		211E	18	
	HLT	211F	76	

2. If we use two 7-segment LED modules side-by-side, we may generate patterns which "look" closer to English letters as shown in the following figure.



Two 7-segment modules displaying 'm'

But, in that case, at one time, only three letters can be displayed in the 6 LED modules of the display section. Write a program that implements a "**Rolling Display**" to display the string of the above assignment (i.e., string of length 6, containing the initials of the student names). For example, if the string is "**ABCDEF**", then your program will display "**ABC**" -delay- "**BCD**" -delay- "**DEF**" -delay- "**EF** " -delay- "**F A**" -delay- " **AB**" -delay- "**ABC**" and so on in an infinite loop.

3. Comprehend and execute the following program and explain what you see in the display section upon key presses in your SDK.

Label	**Assembly Instruction**	**Memory Address (Hex)**	**Machine Language (Hex)**	**Comment**
/* Following is for initializing 8279 for reading from the keyboard by writing the **control word** 50H to the control port of 8279 (address 19H).*/				
	DI	2150	F3	
	MVI A, 50H	2151	3E	
		2152	50	Read from the keyboard
	OUT 19H	2153	D3	
		2154	19	
LOOP	IN 18H	2155	DB	
		2156	18	
	STA 27F6H	2157	32	Save the keycode at 27F6H for display by MODDT
		2158	F6	
		2159	27	
	CALL MODDT	215A	CD	
		215B	FA	
		215C	06	
	LXI D,0000H	215D	11	
		215E	00	
		215F	00	

	CALL DELAY	2160	CD	
		2161	BC	
		2162	03	
	JMP LOOP	2163	C3	
		2164	55	
		2165	21	