SEVEN SEGMENT DISPLAY INTERFACE USING ALS KIT

BY
SUBATHRA S

SEVEN SEGMENT DISPLAY INTERFACE USING ALS KIT

OBJECTIVE

To write an assembly language program to Interface a Seven Segment Display with the 8085 Microprocessor kit to perform various modes of display such as Still Display, Blinking Display and Moving Display

APPARATUS REQUIRED

- 8085 Microprocessor trainer kit.
- Display circuitry.
- Power supply.
- Flat Ribbon Cable.

DESCRIPITION

- Hexadecimal data corresponding to the Segment which has to glow latched using 8bit latch 74LS273.
- A logic`0` will turn `ON` the Segment specified and logic`1` will turn `OFF` the segment.
- BCD data for turning `ON` the display is latched through a 4bit latch 74LS175.
- The control bits A, B and C are decoded using 3 to 8 decoder 74LS138.

STILL DISPLAY

ALGORITHM

1.Initialize the port B and C as output ports by sending 80_H to the control word register.

2.Initialize HL register pair.

3.Move 00_H to accumulator and output it through port A.

4.Move 04_H to C register and 08_H to B register.

5.Move memory content to accumulator and output it through port C complement accumulator content and output it through port C. Decrement B register content by 1. It zero flag is set, go to step 5.

6.Increment HL register pair, decrement C register by 1. If zero flag is reset, go to step 4.

7.Stop the execution.

ASSEMBLY LANGUAGE PROGRAM

ADDRESS	LABEL	MNEMONICS	OPCODE/ OPERAND	COMMENT
C100		**MVI A,80 $_H$**	3E 80	Initialize all the ports as o/p ports.
C102		**OUT CWR**	D3 DB	Write contol word in CWR.
C104		**LXI H,C150 $_H$**	21 50 C1	Load the input sequence to

				be displayed.
C107		**MVI A,00 $_H$**	3E 00	Clear the accumulator and output it.
C109		**OUT PORTC**	D3 DA	
C10B		**MVI C,04 $_H$**	0E 04	Initialize number of characters (letters).
C10D	**YY**	**MVI B,08 $_H$**	06 08	Initialize the number of segments in the seven segment display for single character.
C10F	**XX**	**MOV A,M**	7E	Move the loaded memory content i.e. the recent data to accumulator .
C110		**OUT PORTB**	D3 D9	Output it through Port B.
C112		**RRC**	0F	Rotate the accumulator content right.
C113		**MOV M,A**	77	Move the accumulator content to memory.
C114		**MVI A,10 $_H$**	3E 10	Initialize accumulator and output it .
C116		**OUT PORTC**	D3 DA	
C118		**MVI A,00 $_H$**	3E 00	Clear the accumulator and output it.
C11A		**OUT PORTC**	D3 DA	
C11C		**DCR B**	05	Decrement the segment.
C11D		**JNZ XX**	C2 0F C1	Until all the segments are correspondingly enabled/disabled for a single character, continue looping.
C120		**INX H**	23	Get the next input character.
C121		**DCR C**	0D	Decrement the character count.
C122		**JNZ YY**	C2 0D C1	Until all the characters are inputted & displayed, continue looping.
C125		**HLT**	76	Stop the execution.

EXECUTION

LETTER	a	b	c	d	e	f	g	h	MEMORY LOCATION	HEX CODE
E	0	1	1	0	0	0	0	1	C150 $_H$	61 $_H$
S	0	1	1	0	0	0	1	1	C151 $_H$	63 $_H$
C	0	1	1	0	0	0	0	1	C152 $_H$	61 $_H$

S	0	1	0	0	1	0	0	1	C153 $_H$	49 $_H$

OUTPUT

S	C	S	E

MOVING DISPLAY

ALGORITHM

1.Initialize stack pointer.
2.Initialize the control word for 8255.
3.Initialise HL register pair.
4.Move 05 to C register and 08 to B register.
5.Output accumulator content through port B.
6.Rotate accumulator right through carry.
7.Move accumulator content to memory.
8.Clear accumulator content and output it through port C.
9.Complement accumulator content and decrement B register content.
10.Jump if no zero to step 4.
11.Call delay subroutine and decrement C register content.
12.Jump if no zero to step 2 and jump to step 3.

ASSEMBLY LANGUAGE PROGRAM

ADDRESS	LABEL	MNEMONICS	OPCODE/ OPERAND	COMMENT
C000		**LXI SP,C100 $_H$**	31 60 C1	Initialize the stack pointer.
C003		**MVI A,80 $_H$**	3E 80	Initialize all the ports as o/p ports.
C005		**OUT CWR**	D3 DB	
C007	**ZZ**	**LXI H,C150 $_H$**	21 50 C1	Load the input sequence to be displayed.
C00A		**MVI A,00**	3E 00	Clear the accumulator and output it.
C00C		**OUT PORTC**	D3 DA	
C00E		**MVI C,08 $_H$**	0E 08	Initialize number of characters (letters).
C010	**YY**	**MVI B,08 $_H$**	06 08	Initialize the number of segments in the seven segment display for single character.
C012	**XX**	**MOV A,M**	7E	Move the loaded memory content i.e. the recent data to accumulator .
C013		**OUT PORTB**	D3 D9	Output it through Port B.

C015		**RRC**	0F	Rotate the accumulator content right.
C016		**MOV M,A**	77	Move the accumulator content to memory.
C017		**MVI A,01**	3E 01	Initialize accumulator and output it .
C019		**OUT PORTC**	D3 DA	
C01B		**MVI A,00 $_H$**	3E 00	Clear the accumulator and output it.
C01D		**OUT PORTC**	D3 DA	
C01F		**DCR B**	05	Decrement the segment.
C020		**JNZ XX**	C2 12 C0	Until all the segments are correspondingly enabled/disabled for a single character, continue looping.
C023		**INX H**	23	Get the next input character.
C024		**CALL DELAY**	CD 2E C0	Call delay subprogram in order to make the display visible.
C027		**DCR C**	0D	Decrement the character count.
C028		**JNZ YY**	C2 10 C0	Until all the characters are inputted & displayed, continue looping.
C02B		**JMP ZZ**	C3 07 C0	Repeat the process.
C02E	**DELAY**	**PUSH B**	C5	Delay subprogram is used to create a time delay in order to make the display visible.
C02F		**MVI B,05 $_H$**	06 05	
C031	**DEL2**	**LXI D,FFFF $_H$**	11 FF FF	
C034	**DEL1**	**DCX D**	1B	
C035		**MOV A,E**	7B	
C036		**ORA D**	B2	
C037		**JNZ DEL1**	C2 34 C0	
C03A		**DCR B**	05	
C03B		**JNZ DEL2**	C2 31 C0	
C03E		**POP B**	C1	
C03F		**RET**	C9	

EXECUTION

LETTER	a	b	c	d	e	f	g	h	MEMORY LOCATION	HEX CODE
C	0	1	1	0	0	0	1	1	C150 $_H$	63 $_H$
E	0	1	1	0	0	0	0	1	C151 $_H$	61 $_H$
P	0	0	1	1	0	0	0	1	C152 $_H$	31 $_H$

A	0	0	0	1	0	0	0	1	C153 $_{H}$	11 $_{H}$
	1	1	1	1	1	1	1	1	C154 $_{H}$	FF $_{H}$
	1	1	1	1	1	1	1	1	C155 $_{H}$	FF $_{H}$
	1	1	1	1	1	1	1	1	C156 $_{H}$	FF $_{H}$
	1	1	1	1	1	1	1	1	C157 $_{H}$	FF $_{H}$

OUTPUT

A	P	E	C
	A	P	E
		A	P
			A
C			
E	C		
P	E	C	
A	P	E	C

BLINKING DISPLAY

ALGORITHM

1.Initialize stack pointer.
2.Intialise the control word for 8255.
3.Initialise HL register pair,
4.Move 05 to C register and 08 to B register.
5.move memory content to accumulator and Output accumulator content through port B.
6.clear accumulator content and output it.
7.Complement accumulator content and output it through port C .
8.Decrement B register content and jump if no zero to step 6.
9.Increment HL pair and call delay subroutine.
10.Decrement C register content and jump if no zero to step 5.
11.Jump to step 3.

ASSEMBLY LANGUAGE PROGRAM

ADDRESS	LABEL	MNEMONICS	OPCODE/ OPERAND	COMMENT
C100		**MVI A,80 $_H$**	3E 80	Initialize all the ports as o/p ports.
C102		**OUT CWR**	D3 DB	
C104	**REP**	**LXI H,C150 $_H$**	21 50 C1	Load the input sequence to be displayed.
C107		**MVI E,03 $_H$**	1E 03	Initialize alternative sequence for blinking.
C109	**LP3**	**MVI C,04 $_H$**	0E 04	Initialize number of characters (letters).
C10B	**LP2**	**MVI B,08 $_H$**	06 08	Initialize the number of segments in the seven segment display for single character.
C10D	**LP1**	**MVI A,00 $_H$**	3E 00	Clear the accumulator and output it.
C10F		**OUT PORTC**	D3 DA	
C111		**MOV A,M**	7E	Move the loaded memory content i.e. the recent data to accumulator .
C112		**OUT PORTB**	D3 D9	Output it through Port B.
C114		**RRC**	0F	Rotate the accumulator content right.
C115		**MOV M,A**	77	Move the accumulator content to memory.
C116		**MVI A,01 $_H$**	3E 01	Initialize accumulator and output it .
C118		**OUT PORTC**	D3 DA	
C11A		**DCR B**	05	Decrement the segment.
C11B		**JNZ LP1**	C2 0D C1	Until all the segments are correspondingly enabled/disabled for a single character, continue looping.
C11E		**INX H**	23	Get the next input character.
C11F		**DCR C**	0D	Decrement the character count.
C120		**JNZ LP2**	C2 0B C1	Until all the characters are inputted & displayed, continue looping.
C123		**CALL DELAY**	C2 2D C1	Call delay subprogram.

C126		**DCR E**	1D	Decrement the alternative sequence display.
C127		**JNZ LP3**	C2 09 C1	Loop until not zero.
C12A		**JMP REP**	C3 04 C1	Repeat the process.
C12D	**DELAY**	**PUSH D**	05	Delay subprogram.
C12E		**LXI D,FFFF$_H$**	11 FF FF	
C131	**DEL1**	**DCX D**	1B	
C132		**MOV A,F**	7B	
C133		**ORA D**	B2	
C134		**JNZ DEL1**	C2 31 C1	
C137		**POP D**	D1	
C138		**RET**	C9	

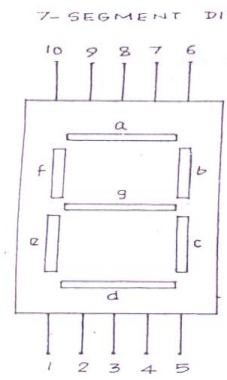
EXECUTION

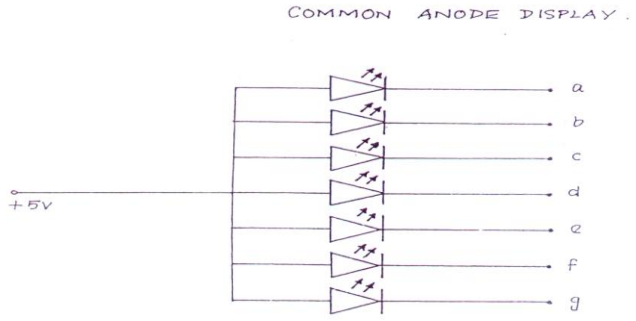
LETTER	a	b	c	d	e	f	g	h	MEMORY LOCATION	HEX CODE
C	0	1	1	0	0	0	1	1	C150 $_H$	63 $_H$
E	0	1	1	0	0	0	0	1	C151 $_H$	61 $_H$
P	0	0	1	1	0	0	0	1	C152 $_H$	31 $_H$
A	0	0	0	1	0	0	0	1	C153 $_H$	11 $_H$
E	0	1	1	0	0	0	0	1	C154 $_H$	61 $_H$
S	0	1	1	0	0	0	1	1	C155 $_H$	63 $_H$
C	0	1	1	0	0	0	0	1	C156 $_H$	61 $_H$
S	0	1	0	0	1	0	0	1	C157 $_H$	49 $_H$
D	1	0	0	0	0	1	0	1	C158 $_H$	85 $_H$
O	0	0	0	0	0	0	1	1	C159 $_H$	03 $_H$
O	0	0	0	0	0	0	1	1	C15A $_H$	03 $_H$
G	0	1	0	0	0	0	1	1	C15B $_H$	43 $_H$

OUTPUT

A	P	E	C
S	C	S	E
G	O	O	D
A	P	E	C

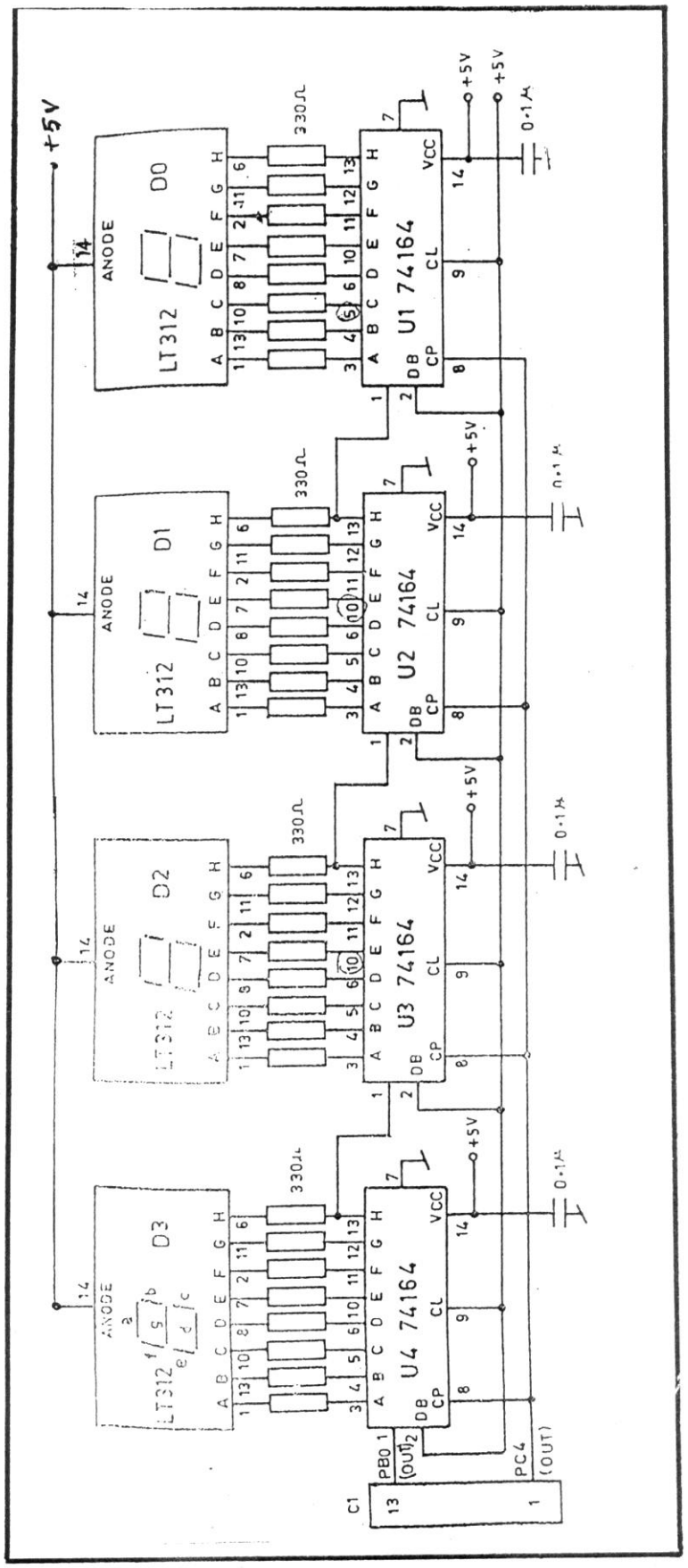
DISPLAY





CIRCUIT DIAGRAM

SEVEN SEGMENT DISPLAY INTERFACE



REFERENCE

1. Ramesh S.Gaonkar, Microprocessor Architecture, Programming, and Applications, Fourth Edition, Penram International Publishing (India), 2000.

2. S.Subathra, “Advanced Microprocessor Laboratory”, Record work, Adhiparashakthi Engineering College, Melmaruvathur, October 2002

3. S.Subathra, “Programming in 8085 Microprocessor and its applications – An Innovative Analysis”, Technical Report, Adhiparashakthi Engineering College, Melmaruvathur, March 2003

4. Micro-85 EB, User Manual, Version – 3.0, CAT #M85 EB-002, VI Microsystems Pvt. Ltd., Chennai.