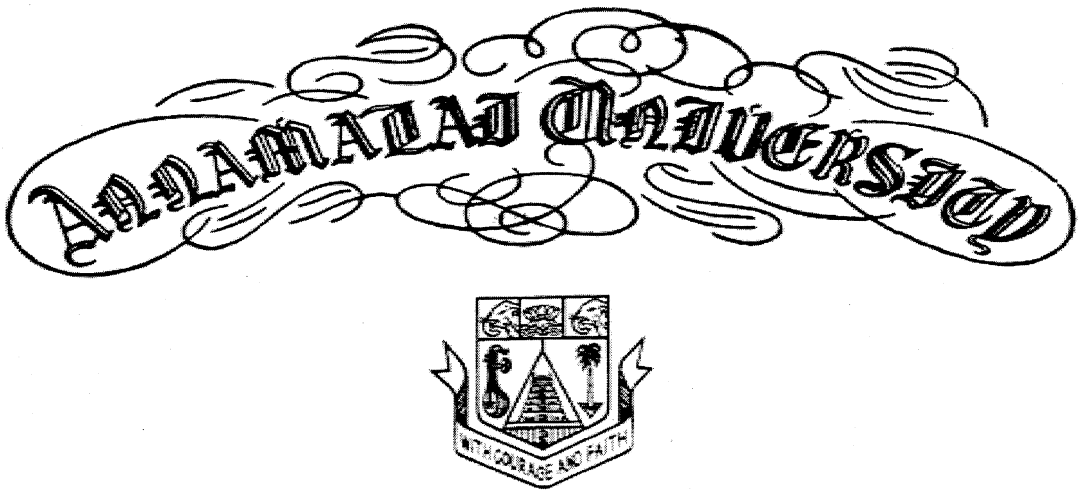


FACULTY OF ENGINEERING TECHNOLOGY

DEPARTMENT OF INSTRUMENTATION ENGINEERING

B.E. (ELECTRONICS AND INSTRUMENTATION ENGINEERING)

VI - SEMESTER

MICROPROCESSOR LAB

Certified that this is the Bonafide Record of work done by Mr./Ms. _____

*Reg. No.*_____ *of VI-Semester B.E. Electronics & Instrumentation Engineering class in the Microprocessor Laboratory during the year 2013-2014.*

Annamalainagar **External Examiner** **Internal Examiner** **Staff-in-Charge**

Date:.....2014

CONTENTS

Name: **Batch:**

Exp. No.	Date	Name of the Experiment	Page No.	Class work (10)	Result (10)	Total (10)	Staff Sign

CONTENTS

Name: **Batch:**

Exp. No.	Date	Name of the Experiment	Page No.	Class work (10)	Result (10)	Total (10)	Staff Sign

LIST OF EXPERIMENTS

1. a) Multiplication and division by repeated addition and subtraction.
   b) Multi byte decimal addition and subtraction.
2. Code conversion.
3. a) Finding Largest / Smallest number in an array of 'N' numbers.
   b) Sorting an array of 'N' numbers in Asending / Desending order.
4. a) Block movement of data.
   b) Interrupt demo using RST 5.5.
5. Switches and LEDs interface.
6. Interfacing ADC 0809 with 8085 microprocessor.
7. Interfacing two channel DAC 0800 with 8085 microprocessor.
8. Interfacing 8255 PPI with 8085 microprocessor.
9. Interfacing 8253 PIT with 8085 microprocessor.
10. Interfacing 8259 PIC with 8085 microprocessor.
11. Kit to Kit data transfer using USART 8251.
12. 8279 keyboard / display interface.
13. Liquid Crystal Alphanumeric Display Interface.
14. Frequency measurement using SID line of 8085 microprocessor.
15. Stepper motor Interface.

APPENDIX A : MONITOR SYSTEM CALLS, DATA FORMAT FOR OF SEVEN SEGMENT DISPLAY

APPENDIX B : DETAILS ABOUT 8255 PROGRAMMABLE PERIPHERAL INTERFACE

APPENDIX C : DETAILS ABOUT 8253 PROGRAMMABLE INTERVAL TIMER

APPENDIX D : SUMMARY OF 8085 INSTRUCTION AND MACHINE CODES

Exp. No. :
Date :

A) MULTIPLICATION AND DIVISION BY REPEATED ADDITION AND SUBTRACTION

AIM

To multiply and divide two 8bit numbers by repeated addition and subtraction in 8085 microprocessor trainer kit.

MULTIPLICATION OF TWO 8-BIT NUMBERS BY REPEATED ADDITION:

This program multiplies two 8-bit numbers which are stored in the location 4300H & 4301H the result is stored in the location 4302H.

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
4200	21		LXI H , 4300H	Pointer where the multiplicand is stored.
4201	00			
4202	43			
4203	7E		MOV A,M	Move Multiplicand to accumulator
4204	FE		CPI 00	Compare multiplicand with 00.
4205	00			
4206	CA		JZ LAST	Jump on no zero to last
4207	18			
4208	42			
4209	47		MOV B,A	Save multiplicand is B Register.
420A	23		INX H	Increment the memory pointer.
420B	7E		MOV A,M	Move multiplier to accumulator.
420C	FE		CPI 00	Compare multiplier with zero.
420D	00			
420E	CA		JZ LAST	Jump on zero to LAST.
420F	18			
4210	42			
4211	4F		MOV C, A	Save multiplier in C register.
4212	AF		XRA A	Clear the accumulator.
4213	80	RPT:	ADD B	Add [B] with [A] by C times.
4214	0D		DCR C	Decrement [C] by 1.
4215	C2		JNZ RPT	Jump on no Zero to RPT.
4216	13			
4217	42			
4218	32	LAST	STA 4302	Store multiplied o/p to memory.
4219	02			
421A	43			
421B	76		HLT	Halt the program execution.

SAMPLE DATA

Input

4300H : 02 (Multiplicand)
4301H : 05 (multiplier)

Output

4302H : 0A

DIVIDING AN 8-BIT NUMBER BY ANOTHER 8-BIT NUMBER

This program divides an 8-bit number by another 8-bit number which are stored in the location 4300H & 4301H and the results are in the locations 4302H & 4303H.

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
4200	0E		MVI C , 00	Initialize the counter for quotient.
4201	00			
4202	3A		LDA 4301H	Load the divisor to accumulator.
4203	01			
4204	43			
4205	FE		CPI 00	Compare divisor with zero.
4206	00			
4207	CA		JZ LAST	Jump on zero to LAST.
4208	17			
4209	42			
420A	47		MOV B,A	Move divisor to accumulator.
420B	3A		LDA 4300H	Load the dividend to accumulator.
420C	00			
420D	43			
420E	B8	RPT:	CMP B	Compare [B]with [A].
420F	DA		JC LAST	Jump on carry to LAST.
4210	17			
4211	42			
4212	90		SUB B	Subtract [B] from [A].
4213	0C		INR C	Increment the quotient by 1.
4214	C3		JMP RPT	Jump to RPT.
4215	0E			
4216	42			
4217	21	LAST	LXI H, 4302H	Initialize HL pair to store the result.
4218	02			
4219	43			
421A	71		MOV M, C	Store the quotient in the memory.
421B	23		INX H	Increment memory pointer.
421C	77		MOV M, A	Store the remainder in the memory.
421D	76		HLT	Halt the program execution.

SAMPLE DATA

Input
4300H: 0A (Dividend)
4301H: 03(Divisor)

Output
4302H: 03(Quotient)
4303H: 01(Remainder).

RESULT

The program was executed and the result was verified.

EXERCISE

1. How will you multiply two 16 bit numbers by repeated addition?
2. Discuss DAD instruction with example?
3. What is the difference between Sub and CMP instruction?
4. Discuss XRA instruction with example?

B) MULTIBYTE DECIMAL ADDITION AND SUBTRACTION

AIM

To add and subtract two multibyte BCD numbers in 8085 microprocessor trainer kit.

ADDITION OF TWO MULTIBYTE BCD NUMBERS

This program adds two multibyte numbers. The number of bytes in the operand is stored in the location 4200H. The first operand is stored from the location 4201H and second operand is stored from the location 4301H and the result is stored from the location 4201H.

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
4100	21		LXI H, 4200H	Address of byte count in H-L pair
4101	00			
4102	42			
4103	4E		MOV C,M	Byte count in Register C
4104	23		INX H	Address of 1st byte of 1st number,
4105	11		LXI D,4301H	Address of 1st byte of 2nd number
4106	01			
4107	43			
4108	AF		XRA A	Clear the carry flags and Acc.
4109	1A	RPT:	LDAX D	get byte of 2nd number in Acc
410A	8E		ADC M	Byte of 2nd number + Byte of 1st number + Carry
410B	27		DAA	Decimal adjust accumulator.
410C	77		MOV M,A	Store the result in memory addressed by H-L pair
410D	13		INX D	Increment DE pair to get next byte.
410E	23		INX H	Increment HL pair to get next byte.
410F	0D		DCR C	Decrement the counter.
4110	C2		JNZ RPT	Jump on no zero to RPT.
4111	09			
4112	41			
4113	76		HLT	Halt the program execution.

SAMPLE DATA

Input

4200H : 04 (No. of bytes in the multibyte numbers)

First operand : 01 02 12 61

4201H : 61
4202H : 12
4203H : 02
4204H : 01

second operand : 03 06 02 48

4301H : 48
4302H : 02
4303H : 06
4304H : 03

OUTPUT

4201H : 09
4202H : 15
4203H : 08
4204H : 04

SUBTRACTION OF TWO MULTIBYTE BCD NUMBERS

This program subtracts two multibyte numbers using 10's complement.The number of bytes in the two operands is stored in the location 4200H. The first operand is stored from the location 4201H and second operand is stored from the location 4301 and the result is stored from the location 4201H.

SAMPLE DATA

INPUT

4300H : 04 (number of bytes in the multibyte number)

First operand : 03 04 05 06
4201H : 06
4202H : 05
4203H : 04
4204H : 03

second operand : 01 02 01 02
4301H : 02
4302H : 01
4303H : 02
4304H : 01

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
4100	11		LXI D, 4201	Initiate DE pair.
4101	01			
4102	42			
4103	21		LXI H, 4300	Initiate HL pair.
4104	00			
4105	43			
4106	4E		MOV C,M	Counter for number of bytes in the BCD number.
4107	23		INX H	Increment HL pair.
4108	3E	RPT:	MVI A,99	Load 99 in accumulator.
4109	99			
410A	96		SUB M	Find 9's complement of the number.
410B	3C		INR A	Find 10's complement of the number.
410C	EB		XCHG	Exchange [DE] with [HL].
410D	86		ADD M	Perform subtraction using addition.
410E	27		DAA	Decimal adjust accumulator.
410F	77		MOV M,A	Store the result in to memory.
4110	23		INX H	Increment HL pair to get next byte.
4111	13		INX D	Increment DE pair to get next byte.
4112	EB		XCHG	Exchange [DE] with [HL].
4113	OD		DCR C	Decrement the counter.
4114	C2		JNZ RPT	Jump on no zero to RPT.
4115	08			
4116	41			
4117	76		HLT	Halt the program execution.

OUTPUT

4201H : 04

4202H : 04

4203H : 02

4204H : 02

RESULT

Addition and subtraction of two multibyte numbers have been performed using 8085 microprocessor.

EXERCISE

1. How will you determine the total number of address lines from the memory?
2. What is the necessity of using DAA instruction?
3. Explain briefly where we are using XCHG instruction and why?
4. Discuss the difference between the conditional looping and unconditional looping instructions?

CODE CONVERSION

A) BINARY TO ASCII (Hex) AND VICE VERSA

AIM

To convert binary to ASCII and vice versa in 8085 assembly language using subroutine.

BINARY TO ASCII (Hex)

This program converts the binary value stored in the location 4300H to the corresponding ASCII values that are stored in the locations 4301H & 4302H.

MAIN PROGRAM

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
4200	21 00 43		LXI H, 4300H	Pointer where the binay number is stored
4203	7E		MOV A,M	Move binary number to accumulator
4204	47		MOV B,A	Save binary number
4205	0F		RRC	Shift the higher order nibble to the position of the lower order nibble
4206	0F		RRC	Shift the higher order nibble to the position of the lower order nibble
4207	0F		RRC	Shift the higher order nibble to the position of the lower order nibble
4208	0F		RRC	Shift the higher order nibble to the position of the lower order nibble
4209	CD 50 42		CALL BTOA	Call the subroutine for conversion
420C	23		INX H	Increment the memory pointer
420D	77		MOV M,A	Move converted value to memory
420E	78		MOV A,B	Move the binary no. again
420F	CD 50 42		CALL BTOA	Call the subroutine for conversion
4212	23		INX H	Increment the memory pointer
4213	77		MOV M,A	Move converted value to memory
4214	76		HLT	Halt the program execution

SAMPLE DATA

Input

4300H: 5A

Output

4301H: 35
4302H: 41

SUBROUTINE

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
4250	E6 0F	BTOA:	ANI 0F	Mask the higher nibble
4252	FE 0A		CPI 0A	ComparE [A] with 0A
4254	DA 59 42		JC L1	Jump to L1 on carry
4257	C6 07		ADI 07	Add 07 with Accumulator
4259	C6 30	L1:	ADI 30	Add 30 with Accumulator
425B	C9		RET	Return back to the main program

ASCII (Hex) TO BINARY

This program converts ASCII(Hex) value stored in the location 4300H to the corresponding binary value that is stored in the location 4301H.

MAIN PROGRAM

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
4200	21 00 43		LXI H,4300H	Pointer where the ASCII number is stored
4203	7E		MOV A,M	Move ASCII number to Accumulator
4204	CD 50 42		CALL ATOB	Call the subroutine for conversion
4207	23		INX H	Increment the memory pointer
4208	77		MOV M,A	Move converted value to memory
4209	76		HLT	Halt the program execution

SUBROUTINE

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
4250	D6 30	ATOB:	SUI 30	Subtract 30 from [A]
4252	FE 0A		CPI 0A	Compare [A] with 0A
4254	D8		RC	Return back to main program on carry
4255	D6 07		SUI 07	Subtract 07 from [A]
4257	C9		RET	Return back to the main program

SAMPLE DATA

Input	Output
4300H: 31	4301H: 07

B) DECIMAL TO HEXA DECIMAL AND VICE VERSA

AIM

To convert decimal to hexadecimal and vice versa using 8085 assembly language.

DECIMAL TO HEXADECIMAL

This program converts the decimal number stored in the location 4300H to the corresponding hexadecimal number that is stored in the location 4301H.

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
4200	21 00 43		LXI H,4300H	Pointer where the decimal number is stored
4203	7E		MOV A,M	Move decimal number to Accumulator
4204	47		MOV B,A	Save the decimal number
4205	E6 0F		ANI 0F	Mask the higer nibble
4207	4F		MOV C,A	Move the lower nibble to [C]
4208	78		MOV A,B	Move the decimal number again to Accumulator
4209	0F		RRC	Shift the higher order nibble to the positiion of the lower order nibble
420A	0F		RRC	Shifting the higher order nibble to the position of the lower order nibble
420B	0F		RRC	Shift the higher order nibble to the position of the lower order nibble
420C	0F		RRC	Shift the higher order nibble to the position of the lower order nibble
420D	E6 0F		ANI 0F	Mask the higher nibble
420F	47		MOV B,A	Move the higher nibble to [B]
4210	AF		XRA A	Clear the Accumulator
4211	B8		CMP B	Compare [A] with [B]
4212	CA 1B 42		JZ LAST	Jump on zero to LAST
4215	C6 0A	BACK:	ADI 0A	Add 0A with [A] ([B] times)
4217	05		DCR B	Decrement the counter by 1
4218	C2 15 42		JNZ BACK	Jump on no zero to BACK
421B	81	LAST:	ADD C	Add [C] with [A]
421C	23		INX H	Increment the memory pointer
421D	77		MOV M,A	Move the hexadecimal number to memory
421E	76		HLT	Halt the program execution

SAMPLE DATA

Input	Output
4300H : 99_{10}	4301H : 63H

HEXADECIMAL TO DECIMAL

This program converts the hexadecimal number stored in the location 4300H to the corresponding decimal and the individual digits of the decimal number is stored in the locations 4301H, 4302H & 4303H.

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
4200	0E 00		MVI C,00	Initialize counter for number of 100_{10}s
4202	41		MOV B,C	Initialize counter for number of 10_{10}s
4203	21 00 43		LXI H,4300H	Pointer where the HEX number is stored
4206	7E		MOV A,M	Move HEX number to Accumulator
4207	FE 64	B0:	CPI 64	Compare [A] with 64(100_{10})
4209	DA 12 42		JC L1	Jump on carry to L1
420C	D6 64		SUI 64	Subtract 64(100_{10}) from [A]
420E	0C		INR C	Increment [C] by 1
420F	C3 07 42		JMP B0	Jump to B0
4212	FE 0A	L1:	CPI 0A	Compare [A] with 0A(10_{10})
4214	DA 1D 42		JC L2	Jump on carry to L2
4217	D6 0A		SUI 0A	Subtract 0A(10_{10}) from [A]
4219	04		INR B	Increment [B] by 1
421A	C3 12 42		JMP L1	Jump to L1
421D	23	L2:	INX H	Increment the memory pointer
421E	71		MOV M,C	Move the number of 100_{10}s to memory
421F	23		INX H	Increment the memory pointer
4220	70		MOV M,B	Move the number of 10_{10}s to memory
4221	23		INX H	Increment the memory pointer
4222	77		MOV M,A	Move the number of 1_{10}s to memory
4223	76		HLT	Halt the program execution

SAMPLE DATA

Input

4300H : FFH

Output

4301H : 02
4302H : 05
4303H : 05

RESULT :

Binary number has been converted to ASCII (Hex) and vice versa and Decimal number has been converted to Hexadecimal number and vice versa using 8085-microprocessor trainer kit.

Exp. No. :
Date :

(A) FINDING LARGEST/SMALLEST NUMBER IN AN ARRAY OF 'N' NUMBERS
(B) SORTING AN ARRAY OF 'N' NUMBERS IN ASCENDING / DESCENDING ORDER

AIM

To find the largest and sm allest number of an array and to sort the given array in ascending / descending order.

PROGRAM TO FIND THE LARGEST NUMBER IN THE GIVEN ARRAY.

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENTS
4250		3E 00	MVI A,00H	Clear the accumulator
4252		06 04	MVI B,04H	Initialize the counter
4254		21 75 42	LXI H,4275	Address of the first number in HL pair
4257	LP2	BE	CMP M	Compare next number with previous maximum number
4258		D2 5C 42	JNC LP1	Jump on no carry to LP1
425B		7E	MOV A,M	Get larger number, in accumulator
425C	LP1	23	INX H	Address of next number
425D		05	DCR B	Decrement the count
425E		C2 57 42	JNZ LP2	Jump if non zero to LP2
4261		32 00 43	STA 4300	Store the result
4264		76	HLT	Stop the program execution

SAMPLE DATA

Input				Output
4275	4276	4277	4278	4300 : 5B
04	3C	5B	2E	

PROGRAM TO FIND THE SMALLEST NUMBER IN THE GIVEN ARRAY.

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENTS
4250		3E FF	MVI A,FFH	Move FF to accumulator
4252		06 04	MVI B,04H	Initialize the counter
4254		21 75 42	LXI H,4275	Address of the first number in HL pair
4257	LP2	BE	CMP M	Compare next number with previous small number
4258		DA 5C 42	JC LP1	
425B		7E	MOV A,M	Get smaller number, in accumulator
425C	LP1	23	INX H	Address of next number
425D		05	DCR B	Decrement the count
425E		C2 57 42	JNZ LP2	Jump if non zero
4261		32 00 43	STA 4300	Store the result
4264		76	HLT	Stop the program execution

SAMPLE DATA

Input				Output
4275	4276	4277	4278	4300 : 04
04	3C	5B	2E	

PROGRAM TO SORT THE GIVEN ARRAY IN ASCENDING ORDER.

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENTS
4200		0E 04	MVI C,04H	Counter to check whether all numbers have been arranged in ascending order
4202		11 00 43	LXI D,4300H	Memory location to store result
4205	LP3	06 04	MVI B,04H	Number of data in the array
4207		21 50 43	LXI H,4350H	Address of first number
420A		3E FF	MVI A, FF	Move FF to Accumulator
420C	LP2	BE	CMP M	Compare next number with previous smallest number. Is previous smallest number < next number
420D		DA 14 42	JC LP1	Yes, smaller number in accumulator
4210		7E	MOV A,M	No, get smaller number in accumulator
4211		22 00 44	SHLD 4400H	Address of the smallest number
4214	LP1	23	INX H	Address of next number
4215		05	DCR B	Decrement count
4216		C2 0C 42	JNZ LP2	Jump on no zero to LP2
4219		12	STAX D	Store the smallest number
421A		13	INX D	Address to store the next smallest number.
421B		2A 00 44	LHLD 4400H	Address of the smallest number in HL pair
421E		36 FF	MVI M,FF	Replace the smallest no by FF
4220		0D	DCR C	Have all numbers been arranged in ascending order
4221		C2 05 42	JNZ LP3	No repeat process
4224		76	HLT	Stop the program execution

SAMPLE DATA

Input

4350	4351	4352	4353
5B	2A	07	7D

Output

4300	4301	4302	4303
07	2A	5B	7D

PROGRAM TO SORT THE GIVEN ARRAY IN DESCENDING ORDER.

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENTS
4200		0E 04	MVI C,04H	Counter to check whether all numbers have been arranged in descending order
4202		11 00 43	LXI D,4300H	Memory location to store result
4205	LP3	06 04	MVI B,04H	Number of data in the array
4207		21 50 43	LXI H,4350H	Address of first number
420A		3E 00	MVI A, 00H	Move 00 to Accumulator
420C	LP2	BE	CMP M	Compare next number with previous largest number. Is previous largest number < next number
420D		D2 14 42	JNC LP1	No, larger number in accumulator
4210		7E	MOV A,M	Yes, get larger number in accumulator
4211		22 00 44	SHLD 4400H	Address of the largest number
4214	LP1	23	INX H	Address of next number
4215		05	DCR B	Decrement count
4216		C2 0C 42	JNZ LP2	Jump on no zero to LP2
4219		12	STAX D	Store the largest number
421A		13	INX D	Address to store the next Largest number.
421B		2A 00 44	LHLD 4400H	Address of the largest number in HL pair
421E		36 00	MVI M,00H	Replace the largest number by 00
4220		0D	DCR C	Have all numbers been arranged in descending order
4221		C2 05 42	JNZ LP3	No repeat process
4224		76	HLT	Stop the program execution

SAMPLE DATA

Input

4350	4351	4352	4353
5B	2A	07	7D

Output

4300	4301	4302	4303
7D	5B	2A	07

RESULT

Programs were written to find the largest, smallest number in a data array and for sorting the array in ascending, decending order. Programs were executed using 8085 Microprocessor.

QUESTIONS

1. List the registers available in the 8085 microprocessor?
2. What is the function of stack?
3. How the compare statement is executed?

(A) BLOCK MOVEMENT OF DATA

AIM

To move a set of data from one memory block to another memory block.

PROGRAM

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
4100	2100 42		LXI H,4200H	1st memory block address
4103	11 00 43		LXI D,4300H	2nd memory block address
4106	06 08		MVI B,05H	No.of memory locations in a block
4108	7E	LOOP	MOV A,M	Move the data from 1st memory block to accumulator.
4109	12		STAX D	Store the data in the 2nd memory block
410A	23		INX H	Increment the source memory pointer (HL)
410B	13		INX D	Increment the destination memory pointer (DE)
410C	05		DCR B	Decrement the counter
410D	C2 08 41		JNZ LOOP	Jump to LOOP if entire block of data is not transferred
4110	76		HLT	Stop.

SAMPLE DATA

Input

4200	4201	4202	4203	4204
5B	2A	07	7D	6E

Output

4300	4301	4302	4303	4304
5B	2A	07	7D	6E

RESULT

The program was executed and the result was verified.

EXERCISE

1. The memory location 2050H holds the data byte F7H.write instructions to transfer the data bytes to accumulator using three different instructions: MOV, LDAX and LDA.
2. When are Zero (Z) and Carry (C) flags set?

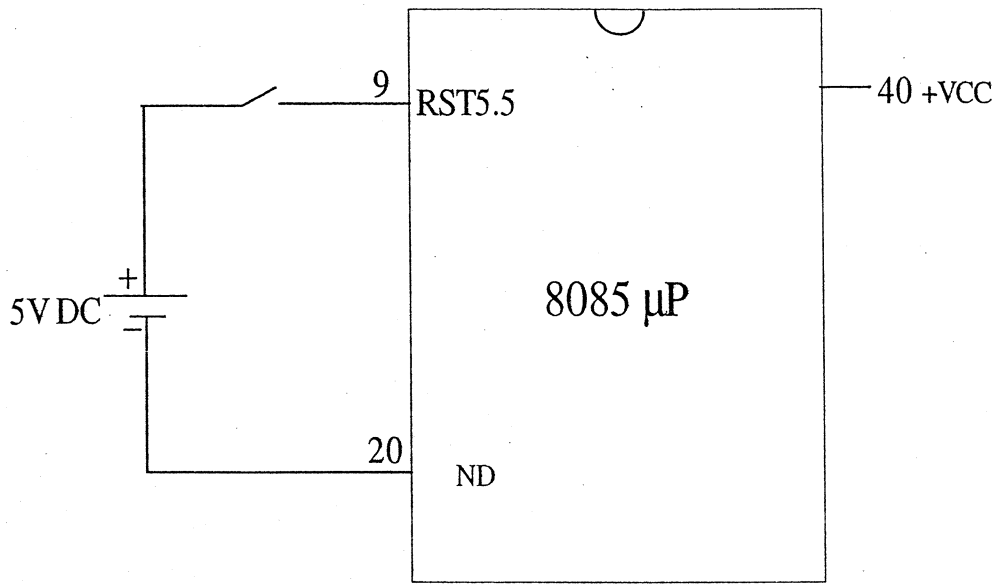
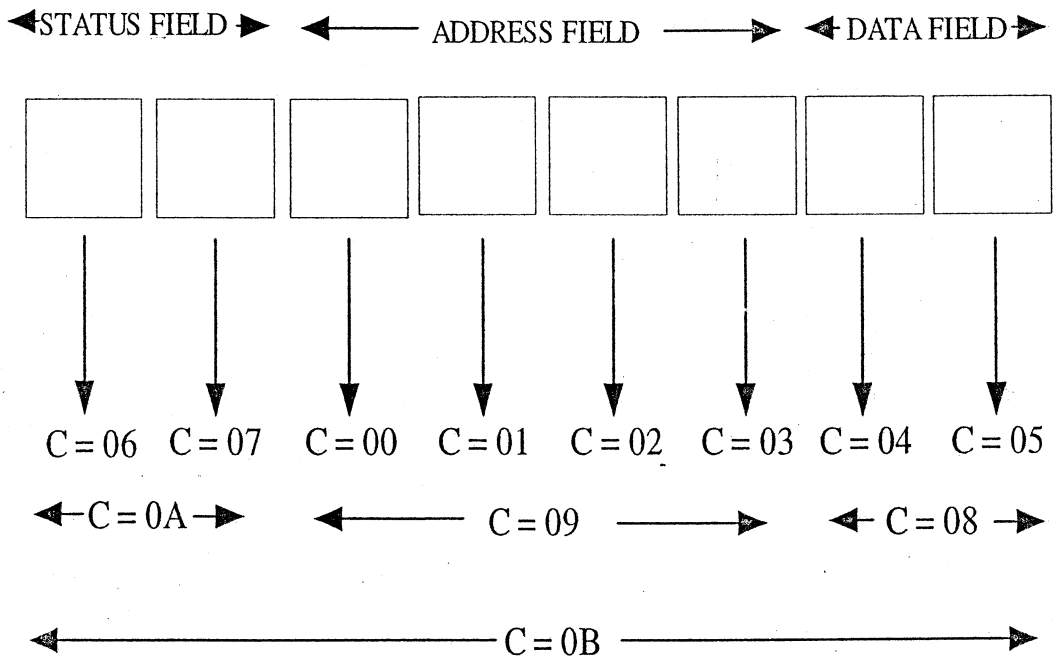


Figure : 4.1



The various fields in which the data will be displayed when the content of C varies from 00 to 0B

Figure : 4.2

4(B) INTERRUPT DEMO USING RST 5.5

AIM:

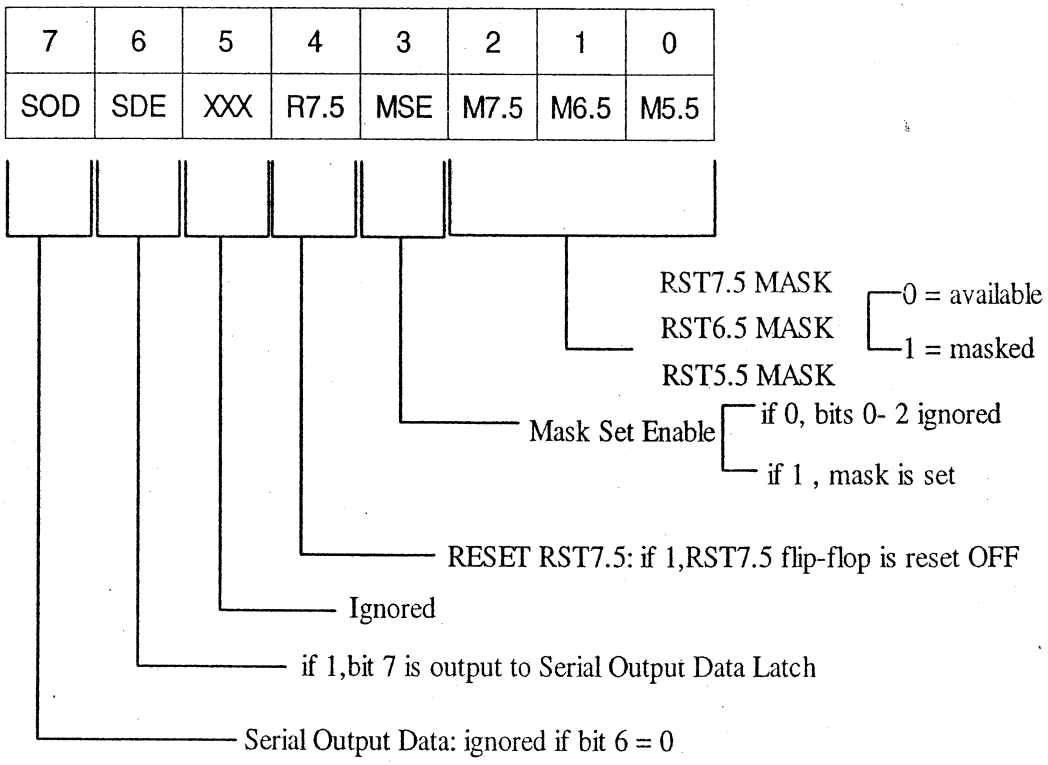
To display 0 and 1 alternatively until RST 5.5 interrupt occurs, and on receiving interrupt, display HELLO for 1 second and return back to display 0 and 1 alternatively.

MAIN PROGRAM

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
4100	3E 1E		MVI A, 1E	Loading data into accumulator to enable RST 5.5 interrupt
4102	30		SIM	Set interrupt mask
4103	FB		EI	Enable interrupt
4104	3E 07	BEGIN	MVI A, 07	Steps for inbuilt monitor call routine ASCII value of 0 is placed in location 4200 for display
4106	0E 40		MVI C, 40	
4108	21 00 42		LXI H, 4200	
410B	CD 05 00		CALL 0005	
410E	CD 00 45		CALL 4500	Call delay subroutine
4111	3E 07		MVI A, 07	Steps for inbuilt monitor call routine ASCII value of 1 is placed in location 4250 for display
4113	0E 40		MVI C, 40	
4115	21 50 42		LXI H, 4250	
4118	CD 05 00		CALL 0005	
411B	CD 00 45		CALL 4500	Call delay subroutine
411E	C3 04 41		JMP BEGIN	Jump to address 4103

INTERRUPT SERVICE ROUTINE

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
4300	3E 07		MVI A, 07	Steps for inbuilt monitor call. ASCII values for blank are placed in location 4450 for display
4302	0E 40		MVI C, 40	
4304	21 50 44		LXI H, 4450	
4307	CD 05 00		CALL 0005	
430A	CD 00 45		CALL 4500	Call delay subroutine
430D	3E 07		MVI A, 07	Steps for inbuilt monitor call. ASCII value of HELLO is placed in location 4400 for display
430F	0E 40		MVI C, 40	
4311	21 00 44		LXI H, 4400	
4314	CD 05 00		CALL 0005	
4317	CD 00 45		CALL 4500	Call delay subroutine
431A	CD 00 45		CALL 4500	Call delay subroutine
431D	FB		EI	Enable Interrupt
431E	C9		RET	Return to main program



Interpretation of the Accumulator Bit Pattern for the SIM Instruction

DELAY SUBROUTINE

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
4500	21 FF FF		LXI H, FFFF	Load HL pair with maximum count
4503	2B		DCX H	Decrement HL pair
4504	7C		MOV A, H	
4505	B5		ORA L	
4506	C2 03 45		JNZ 4503	Jump on non zero to 4503
4509	C9		RET	Return

LOOK UP TABLE FOR 0 & 1

ADDRESS	OPCODE	COMMENTS		ADDRESS	OP	COMMENTS
4200	30	ASCII value of '0'		4250	31	ASCII value of '1'
4201	20	ASCII value of blank		4251	20	ASCII value of blank
4202	20	ASCII value of blank		4252	20	ASCII value of blank
4203	20	ASCII value of blank		4253	20	ASCII value of blank
4204	20	ASCII value of blank		4254	20	ASCII value of blank
4205	20	ASCII value of blank		4255	20	ASCII value of blank
4206	20	ASCII value of blank		4256	20	ASCII value of blank
4207	20	ASCII value of blank		4257	20	ASCII value of blank
4208	20	ASCII value of blank		4258	20	ASCII value of blank
4209	20	ASCII value of blank		4259	20	ASCII value of blank
420A	24	End of string		425A	24	End of string

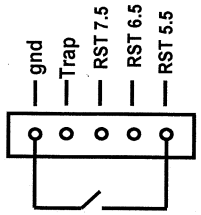
LOOK UP TABLE FOR HELLO & BLANK

ADDRESS	OPCODE	COMMENTS		ADDRESS	OP	COMMENTS
4400	48	ASCII value of 'H'		4450	20	ASCII value of blank
4401	45	ASCII value of 'E'		4451	20	ASCII value of blank
4402	4C	ASCII value of 'L'		4452	20	ASCII value of blank
4403	4C	ASCII value of 'L'		4453	20	ASCII value of blank
4404	4F	ASCII value of 'O'		4454	20	ASCII value of blank
4405	20	ASCII value of blank		4455	20	ASCII value of blank
4406	20	ASCII value of blank		4456	20	ASCII value of blank
4407	20	ASCII value of blank		4457	20	ASCII value of blank
4408	20	ASCII value of blank		4458	20	ASCII value of blank
4409	20	ASCII value of blank		4459	20	ASCII value of blank
440A	24	End of string		445A	24	End of string

RST 5.5 VECTOR ADDRESS

ADDRESS	OPCODE	COMMENTS
002C	C3 70 40	Jump to location 4070
4070	C3 00 43	Jump to location 4300 (Main program)

P11 CONNECTOR



The connection diagram for the experiment is shown in the figure 4.1. Figure 4.2. shows the various fields in which the data will be displayed when the content for C varies from 00 to 0B.

Note : (For the details of monitor system calls and Data format for 7 Segment Display refer Data sheets given in Appendix -A).

RESULT

The program was executed and the result was verified.

EXERCISE

1. What are the two major classes of interrupts? Discuss in detail.
2. Display your name in the address field before giving any interrupt on the RST 6.5 line and display your register number on receiving the interrupt.

SWITCHES AND LED INTERFACE

AIM :

To interface a set of 8 SPDT switches to 8 LED'S through 8255 chip and set up an upper counter generation .

THEORY:

The IC 8255 (Appendix-B) is known as programmable peripheral interface (PPI). 8255 contains three I/O ports each one of word length 8 bits. The input and output action of the I/O ports are controlled by the control word which is stored in the control register. Eight SPDT switches are connected to port A of 8255 and eight LED's are connected to port B of the 8255.

1. INTERFACING A SET OF 8 SPDT SWITCHES TO 8 LEDS

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENTS
4100		3E 90	MVI A,90H	Load Acc with control word
4102		D3 C6	OUT C6H	Out to the control register
4104	RPT	DB C0	IN C0H	Read switch status
4106		D3 C2	OUT C2H	Display it on LEDs
4108		C3 04 41	JMP RPT	Load to start and repeat it

2. RIPPLE COUNTER-1

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENTS
4100		3E 90	MVI A,90H	Load Acc with control word
4102		D3 C6	OUT C6H	Out to the control register
4104	LP1	3E 01	MVI A,01H	Get the first count
4106	LP2	D3 C2	OUT C2H	Display it on LEDs
4108		CD 12 42	CALL DELAY	Call the delay routine
410B		CD 12 42	CALL DELAY	
410E		87	ADD A	Get next count
410F		C2 06 41	JNZ LP2	Is this last count
4112		C3 04 41	JMP LP1	Continue the loop

3. RIPPLE COUNTER-2 : TO GLOW ALTERNATE LEDS

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENTS
4100		3E 90	MVI A,90H	Load Acc with control word
4102		D3 C6	OUT C6H	Out to the control register
4104	LP1	3E 01	MVI A,01H	Get the first count
4106	LP2	D3 C2	OUT C2H	Display it on LEDs
4108		CD 12 42	CALL DELAY	Call the delay routine
410B		CD 12 42	CALL DELAY	
410E		87	ADD A	Get next count
410F		87	ADD A	Add the previous count
4110		C2 06 41	JNZ LP2	Is this last count
4113		C3 04 41	JMP LP1	Continue the loop

4. BINARY COUNTER

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENTS
4100		3E 90	MVI A,90H	Load Acc with control word
4102		D3 C6	OUT C6H	Out to the control register
4104	LP1	3E 01	MVI A,01H	Get the first count
4106	LP2	D3 C2	OUT C2H	Display it on LEDs
4108		CD 12 42	CALL DELAY	Call the delay routine
410B		CD 12 42	CALL DELAY	Call the delay routine
410E		3C	INR A	Increment the count
410F		C2 06 41	JNZ LP2	Is this last count
4112		C3 04 41	JMP LP1	Continue the loop

5. BCD COUNTER

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENTS
4100		3E 90	MVI A,90H	Load Acc with control word
4102		D3 C6	OUT C6H	Out to the control register
4104	LP1	3E 01	MVI A,01H	Get the first count
4106	LP2	D3 C2	OUT C2H	Display it on LEDs
4108		CD 12 42	CALL DELAY	Call the delay routine
410B		CD 12 42	CALL DELAY	Call the delay routine
410E		C6 01	ADI 01	Get next count
4110		27	DAA	Decimal Adjust Accumulator
4111		C2 06 41	JNZ LP2	Is this last count
4114		C3 04 41	JMP LP1	Continue the loop

DELAY SUBROUTINE (For All Programs)

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENTS
4212		06 AC	MVI B, AC H	B is loaded with data AC
4214	LP2	0E FF	MVI C,FFH	C is loaded with FF
4216	LP1	0D	DCR C	Decrement register C
4217		C2 16 42	JNZ LP1	Inner loop
421A		05	DCR B	Decrement register B
421B		C2 14 42	JNZ LP2	Outer loop
421E		C9	RET	Go to main program

THE HARDWARE DESCRIPTION :

The CHIP SELECT (CS) for the IC 8255 is generated through the 4-bit comparator IC 74LS85. The IC 74LS85 has two 4 bit inputs (called A0 – A3 & B0 – B3). The 4 bit inputs A0 – A3 are connected to CPU address bus (A4 – A7). The other 4 bit inputs B0 – B3 are connected through jumpers to either ground or +5 v (high level). When the CPU address A4 = 0, A5 = 0, A6 = 1, A7 = 1, A = B output of IC 74LS85 goes high. After giving CS, by changing the levels of A0 and A1 pins of 8255, select the individual port. The CPU addresses and IC 8255 addresses are connected as follows.

8255 Address Pins		CPU Address Bus
A0	←———	A1
A1	←———	A2

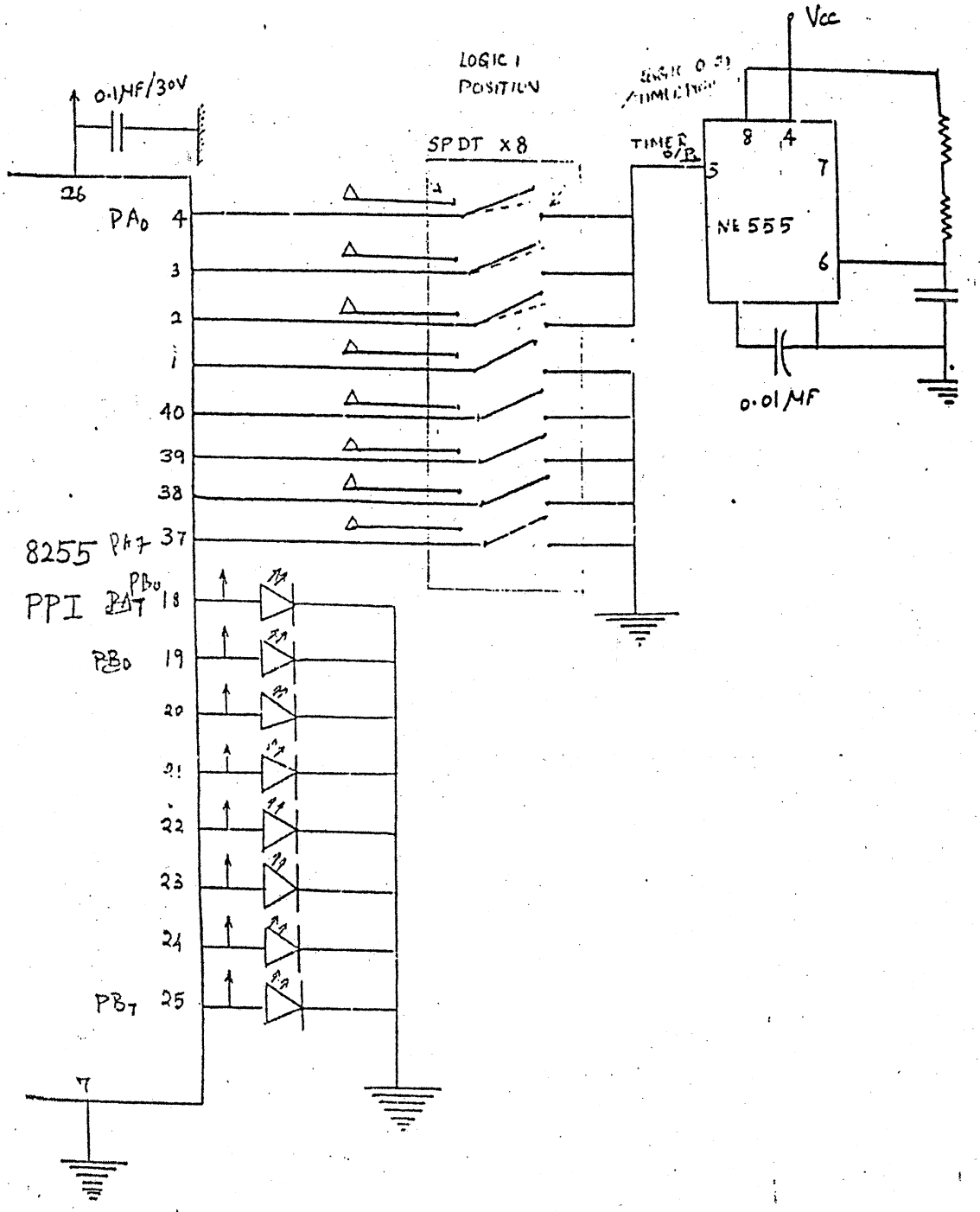
The following table gives the detail of selecting the port by A0 and A1 of IC 8255 programmable peripheral interface.

A0	A1	port selected
0	0	port A
0	1	port B
1	0	port C
1	1	control register

A3 A2 A1 A0 (OF CPU ADDRESS BUS)		Lower Nibble
0 0 0 0	port A	0 or 1
0 0 1 0	port B	2 or 3
0 1 0 0	port C	4 or 5
0 1 1 0	Control Reg	6 or 7

The complete addresses for the 8255 ports would be:

8255 port			I/D address
Switches	—— port A	—-→	C0 or C1
LEDs	—— port B	—-→	C2 or C3
For the bread} board area}	—— port C	—-→	C4 or C5
	Control Reg	—-→	C6 or C7


Vcc
LOGIC 1
POSITION
SPDT X8
0.1MF/30V
26
PA0 4
3
2
1
40
39
38
8255 PA7 37
PPI PA1 PB0 18
PB0 19
20
21
22
23
24
PB7 25
7
8 4
5 7
NE 555
6
0.01MF

A7	A6	A5	A4	A3	A2	A1	Ao	Port address
1	1	0	0	X(0)	0	0	X(0)	Co Port A
1	1	0	0	X(0)	0	1	X(0)	C2 Port B
1	1	0	0	X(0)	1	0	X(0)	C4 Port C
1	1	0	0	X(0)	1	1	X(0)	C6 Control Register

RIPPLE COUNTER-1:(PATTERN)	ALTERNATE LED DISPLAY:
0000 0001 - 01H 0000 0010 - 02H 0000 0100 - 04H 0000 1000 - 08H 0001 0000 - 10H 0010 0000 - 20H 0100 0000 - 40H 1000 0000 - 80H	0000 0001 - 01 0000 0100 - 04 0001 0000 - 10 0100 0000 - 40
BINARY COUNTER:	**BCD COUNTER PATTERN:**
0000 0001 - 01H 0000 0010 - 02H 0000 0011 - 03H 0000 0100 - 04H 0000 0101 - 05H 0000 0110 - 06H 0000 0111 - 07H 0000 1000 - 08H 0000 1001 - 09H 0000 1010 - 0AH(10) 0000 1011 - 0BH(11) 0000 1100 - 0CH(12) 0000 1101 - 0DH(13) 0000 1110 - 0EH(14) 0000 1111 - 0FH(15) .. .. .. 1111 1111 - FFH(255)	0000 0001 - 1 0000 0010 - 2 0000 0011 - 3 0000 0100 - 4 0000 0101 - 5 0000 0110 - 6 0000 0111 - 7 0000 1000 - 8 0000 1001 - 9 0001 0000 - 10 0001 0001 - 11 0001 0010 - 12 0001 0011 - 13 0001 0100 - 14 0001 0101 - 15 ..... 1001 1001 - 99

Note : (For the details of 8255 Internal Block Diagram, Pin Configuration and the Control Word Format refer Data sheet given in Appendix-B)

RESULT :

Switches and LEDs were interfaced with 8085 micro processor using 8255 and various programs were executed and verified.

Exercise :

1. Explain how a level limit switch can be interfaced with microprocessor?
2. Explain the term "Handshaking Signal"

Exp. No. :
Date :

INTERFACING ADC 0809 WITH 8085 MICROPROCESSOR

AIM :

To interface ADC 0809 to 8085 microprocessor kit.

THEORY :

ADC 0809 is a monolithic CMOS device, with an 8-bit analog-to-digital converter, 8 channel multiplexer and microprocessor compatible control logic.

The main features of ADC 0809 are,

1) 8 bit resolution
2) 100 µS conversion time
3) 8 channel multiplexer with latched control logic
4) No need for external zero or full scale adjustments
5) Low power consumption 15mW
6) Latched tristate output

A particular input channel is selected by using the addressdecoding.

Selected Analog Channel	Address Line		
	ADD C	ADD B	ADD A
IN0	0	0	0
IN1	0	0	1
IN2	0	1	0
IN3	0	1	1
IN4	1	0	0
IN5	1	0	1
IN6	1	1	0
IN7	1	1	1

I/O DECODING

A 3 to 8 decoded 74LS138 (U2) is employed to generate I/O decoding logic.

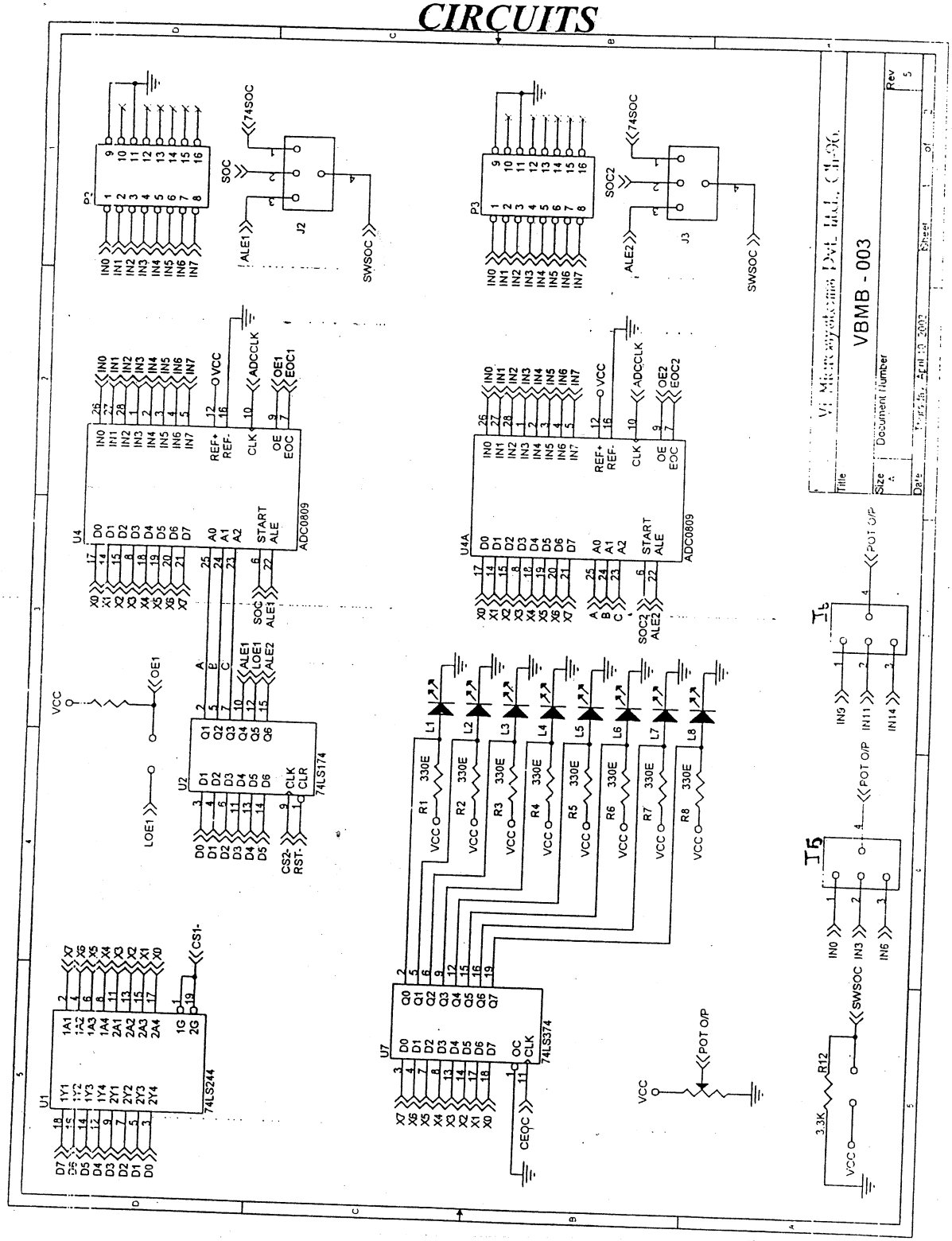
Thus the buffer 74LS244 which transfers the converted data outputs to data bus is selected when

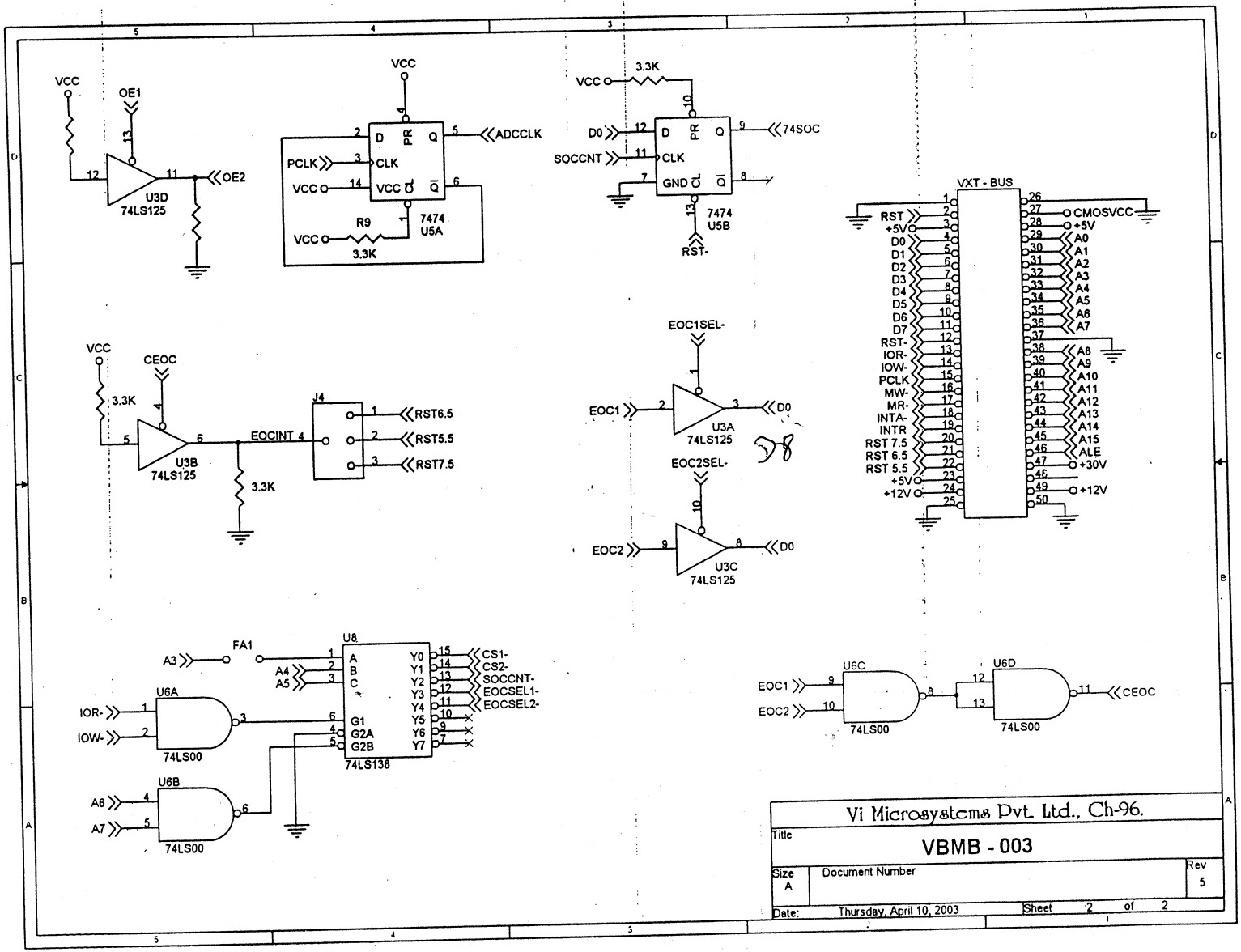
A7	A6	A5	A4	A3	A2	A1	A0	
1	1	0	0	0	X	X	X	= C0H

The I/O address for the latch 74LS174 which latches the data bus to ADD A, ADD B, ADD C and ALE 1 and ALE 2 is

A7	A6	A5	A4	A3	A2	A1	A0	
1	1	0	0	1	X	X	X	= C8H

CIRCUITS


VBMB - 003
Rev
Title
Size
Document Number
Date
Sheet
of
U1
74LS244
U2
74LS174
U4
ADC0809
U4A
ADC0809
U7
74LS374
J2
J3
J5
J6
P2
P3
R1
R2
R3
R4
R5
R6
R7
R8
330E
R12
3.3K
L1
L2
L3
L4
L5
L6
L7
L8
VCC
74SOC
SOC
SOC2
ALE1
ALE2
SWSOC
ADCCLK
POT O/P
CS1-
CS2-
RST-
LOE1
OE1
OE2
CEOC


VCC
OE1
U3D
74LS125
OE2
PCLK
CLK
ADCCLK
R9
3.3K
7474
U5A
D0
SOCCNT
74SOC
U5B
RST-
VXT - BUS
RST
+5V
D0
D1
D2
D3
D4
D5
D6
D7
IOR-
IOW-
PCLK
MW-
MR-
INTA-
INTR
RST 7.5
RST 6.5
RST 5.5
ALE
+30V
+12V
CMOSVCC
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
CEOC
EOCINT
J4
RST6.5
RST5.5
RST7.5
U3B
EOC1SEL-
EOC1
U3A
EOC2SEL-
EOC2
U3C
FA1
U8
A3
A4
A5
G1
G2A
G2B
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
CS1-
CS2-
SOCCNT-
EOCSEL1-
EOCSEL2-
74LS138
U6A
U6B
74LS00
U6C
U6D
CEOC
Vi Microsystems Pvt. Ltd., Ch-96.
Title
VBMB - 003
Size
A
Document Number
Rev
5
Date:
Thursday, April 10, 2003
Sheet
2
of
2

The start of conversion pulse can be given by means of software also. The flip flop 74LS74 which transfers the D0 line status to the SOC pin of ADC 0809 is selected when

A7	A6	A5	A4	A3	A2	A1	A0	
1	1	0	1	0	X	X	X	= D0H

The EOC output of ADC 1 and ADC 2 is transferred to D0 line by means of two tristate buffers.

The EOC 1 is selected when

A7	A6	A5	A4	A3	A2	A1	A0	
1	1	0	1	1	X	X	X	= D8H

The EOC 2 is selected when

A7	A6	A5	A4	A3	A2	A1	A0	
1	1	1	0	0	X	X	X	= E0 xH

PROGRAM

ADDRESS	OPCODE	LABEL	MNEMONICS	COMMENTS
4100	3E 10		MVI A,10	Select channel 0 and make ALE low
4102	D3 C8		OUT 0C8H	
4104	3E 18		MVI A,18	Make ALE high
4106	D3 C8		OUT 0C8H	
4108	3E 01		MVI A,01	SOC signal high
410A	D3 D0		OUT 0D0H	
410C	AF		XRA A	Delay
410D	AF		XRA A	
410E	AF		XRA A	
410F	3E 00		MVI A,00	SOC signal low
4111	D3 D0		OUT 0D0H	
4113	DB D8	LOP:	IN 0D8H	Check for EOC
4115	E6 01		ANI 01	
4117	FE 01		CPI 01	
4119	C2 13 41		JNZ LOP	
411C	DB C0		IN 0C0H	Read data from ADC
411E	32 50 41		STA 4150H	
4121	76		HLT	

PROCEDURE

Execute the program and compare the data displayed at the LED's with that stored at location 4150H.

TABULAR COLUMN

ANALOG INPUT (V)	DIGITAL OUTPUT IN HEX (AVAILABLE IN 4200)	
	THEORITICAL(Hex)	PRACTICAL(Hex)
1	33	
2	66	
3	99	
4	CC	
5	FF	

RESULT

Thus the ADC 0809 was interfaced to 8085 Microprocessor.

Exp. No. :
Date :

INTERFACING TWO CHANNEL DAC 0800 WITH 8085 MICROPROCESSOR

AIM

To interface a two-channel 8 bit DAC with 8085 microprocessor kit.

THEORY

The data bus is latched to the DAC by 74LS273 (U1 & U4). These latches are in turn clocked by the 3 to 8 decoder 74LS138. The address line A3, A4 & A5 are tied to Pin 1, Pin 2 and Pin 3 of 74LS138 respectively. The address line A6 and A7 are NANDed together and the NAND gate output is connected to Pin 5. Similarly IOW and IOR signals are NANDed and the NAND gate output is connected to Pin 6 of 74LS138 (Refer Hardware Diagram) Pin 4 is grounded.

DAC-1 is selected when

A7	A6	A5	A4	A3	A2	A1	A0	
1	1	0	0	0	X	X	X	= C0 (Hex)

DAC – 2 is selected when

A7	A6	A5	A4	A3	A2	A1	A0	
1	1	0	0	1	X	X	X	= C8 (Hex)

DAC 0800 is a monolithic high speed current output digital to analog converter. Its unique features are

1. Typical settling time of 100ns.
2. Complementary current output.
3. Differential output voltage of 20 Vpp by simply changing load resister.
4. Two quadrant wide range multiplying capacity.

The current output of DAC 0800 is converted to voltage by op-Amp 741. The feedback resistor selected is 2.2k, so that the output varies between –5V to +5V.

The DAC outputs are available at the 5 pin connector (P2). DAC1 & DAC2 outputs are available at Pin 5 and Pin 4 respectively. Pin 1 is grounded.

1. SQUAREWAVE

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENTS
4100		3E	MVI A, 00	Data for –5V
4101		00		
4102		D3	OUT C8	DAC 2 Address
4103		C8		
4104		CD	CALL 4111	call delay
4105		11		
4106		41		
4107		3E	MVI A, FF	Data for +5V
4108		FF		
4109		D3	OUT C8	DAC 2 Address
410A		C8		
410B		CD	CALL 4111	call delay
410C		11		
410D		41		
410E		C3	JMP 4100	Repeat the same
410F		00		
4110		41		
4111		06	MVI B, 05	Outer loop of delay
4112		05		
4113		0E	MVI C, FF	Inner loop for delay
4114		FF		
4115		0D	DCR C	Decrement the value of inner loop
4116		C2	JNZ 4115	Jump on no zero to decrement the inner loop
4117		15		
4118		41		
4119		05	DCR B	Decrement the value of Outer loop
411A		C2	JNZ 4113	Jump on no zero to decrement the outer loop
411B		13		
411C		41		
411D		C9	RET	Return to main program

Theoretical calculation of frequency

MNEMONICS	NO. OF T-STATES	NO. OF TIMES OCCURRED	TOTAL NO. OF T-STATES
MVI B, 05	7	1	7
MVI C, FF	7	5	35
DCR C	4	5 x FF	5100
JNZ	10	5 x FE	12700
NO JUMP	7	5 x 1	35
DCR B	4	5	20
JNZ	10	4	40
NO JUMP	7	1	7
RET	10	1	10
		Total No of T - States	17954

Frequence Calculation

t = 17954 / 3 x 10^6 = 5.98 x 10^{-3} sec

t = 2 x t = 0.01196

f = 1/t = 83.612 Hz

II. SAW TOOTH WAVE

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4100	3E	MVI A, 00	Data for -5V
4101	00		
4102	D3	OUT C0	DAC 1
4103	C0		
4104	3C	INR A	Increment accumulator
4105	C2	JNZ 4102H	Jump on no zero to DAC 1
4106	02		
4107	41		
4108	C3	JMP 4100H	Repeat the same
4109	00		
410A	41		

Theoretical calculation of frequency

MNEMONICS	NO. OF T-STATES	NO. OF TIMES OCCURRED	TOTAL NO. OF T-STATES
MVI A, 00	7	1	7
OUT C0	10	FF	2550
INR A	4	FF	1020
JNZ L1	10	FE	2540
NO JUMP	7	1	7
JMP	10	1	10
			6134

$t = 6134 / 3 \times 10^{6} = 2.044 \times 10^{-3}$ sec

$f = 1/t = 489.07$ Hz

III. TRIANGULAR WAVE FORM

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENTS
4100		3E	MVI A, 00	Data for -5V
4101		00		
4102		D3	OUT C8	DAC 2
4103		C8		
4104		3C	INR A	Increment accumulator
4105		C2	JNZ 4102	Jump on no zero to DAC - 2
4106		02		
4107		41		
4108		3E	MVI A, FF	Data for +5V
4109		FF		
410A		D3	OUT C8	
410B		C8		
410C		3D	DCR A	Decrement accumulator
410D		C2	JNZ 410A	Jump on no zero to DAC – 2
410E		0A		
410F		41		
4110		C3	JMP 4100	Repeat the same
4111		00		
4112		41		

Theoretical calculation of frequency

MNEMONICS	NO. OF T-STATES	NO. OF TIMES OCCURRED	TOTAL NO. OF T-STATES
MVI A, 00	7	1	7
OUT C8	10	FF	2550
INR A	4	FF	1020
JNZ L1	10	FE	2540
NO JMP	7	1	7
MVI A, FF	7	1	7
OUT C8	10	FF	2550
DCR A	4	FF	1020
JNZ L2	10	FE	2540
NO JMP	7	1	7
JMP	10	1	10

Total No of T - States 12448

$t = 12448 / 3 \times 10^6 = 4.086 \times 10^{-3}$ sec(Frequency)

$f = 1/t = 244.78$ Hz

IV. SINE WAVE GENERATION

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENTS
4100		21	LXI H, 4110	Pointer where the sine wave data is stored.
4101		10		
4102		41		
4103		0E	MVI C, 59	Counter for sine wave data.
4104		59		
4105		7E	MOV A, M	Move data to accumulator.
4106		D3	OUT C0	
4107		C0		
4108		23	INX H	Increment the pointer.
4109		0D	DCR C	Decrement the pointer.
410A		C2	JNZ 4105	Jump on no zero to 4105
410B		05		
410C		41		
410D		C3	JMP 4100	Repeat the same
410E		00		
410F		41		

LOOK UP TABLE FOR THE SINE WAVE

(* sine wave sampling data)

4110	10	11	19	22
4114	2A	32	39	40
4118	46	4D	52	57
411C	5A	60	64	6A
4120	72	77	81	86
4124	90	95	9A	A0
4128	A4	A9	AF	B4
412C	B9	C0	C5	CA
412D	CF	D4	D9	DF
4134	E4	E7	E7	E8
4138	E8	E8	E7	E7
413C	E6	E6	E5	E0
4140	DB	D5	D0	CB
4144	C6	C1	BB	B6
4148	B1	AA	A5	A1
4150	88	82	7A	75
4154	6E	67	5E	56
4158	4D	47	40	3A
415C	33	2E	29	26
4160	21	1B	17	14
4164	11	10	10	10
4168	09			
*414C	9B	97	92	8D

Theoretical calculation of frequency

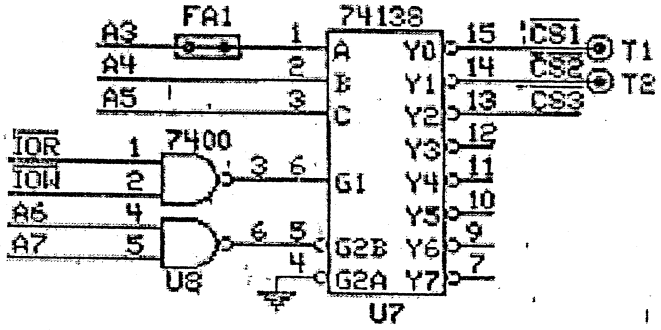
MNEMONICS	NO. OF T-STATES	NO. OF TIMES OCCURRED	TOTAL NO. OF T-STATES
LXI H, 4110	10	1	10
MVI C, 59	7	1	7
MOV A, M	7	90	630
OUT C0	10	90	900
INX H	6	90	540
DCR C	4	90	360
JNZ LOOP	10	89	890
NO JMP	7	1	7
JMP	10	1	10
			3354

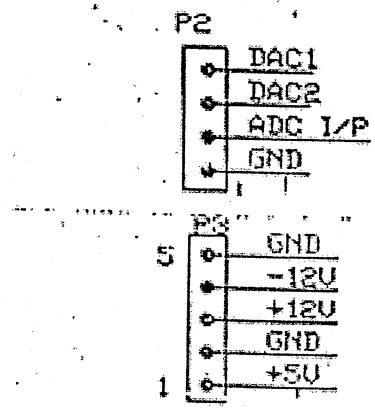
$t = 3354 / 3 \times 10^{6} = 1.118 \times 10^{-3}$ sec

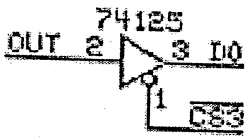
$f = 1/t = 898.8$ Hz

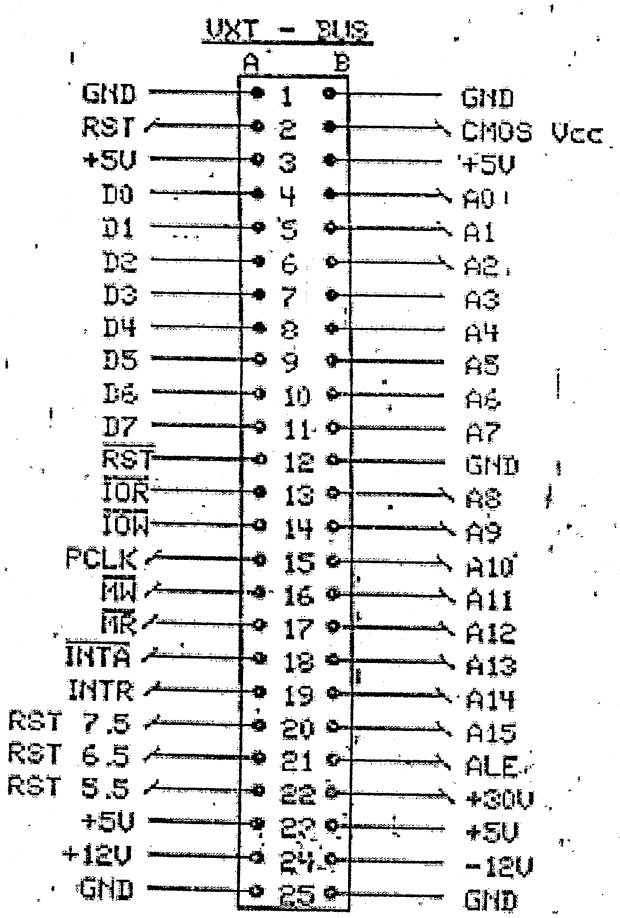
V. STEP WAVE

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4100	3E	MVI A, 00	Data for -5V
4101	00		
4102	D3	OUT C8	DAC – 2
4103	C8		
4104	CD	CALL 411F	Call Delay routine
4105	1F		
4106	41		
4107	3E	MVI A, 80	Data for 0V
4108	80		
4109	D3	OUT C8	DAC - 2
410A	C8		
410B	CD	CALL 411F	Call Delay routine
410C	1F		
410D	41		
410E	3E	MVI A, FF	Data for +5V
410F	FF		
4110	D3	OUT C8	DAC - 2
4111	C8		
4112	CD	CALL 411F	Call Delay routine
4113	1F		
4114	41		
4115	3E	MVI A, 80	Data for 0V
4116	80		
4117	D3	OUT C8	DAC - 2
4118	C8		
4119	CD	CALL 411F	Call Delay routine
411A	1F		
411B	41		
411C	C3	JMP 4100	Repeat the same
411D	00		
411E	41		
411F	06	MVI B, 05	Outer loop of delay
4120	05		
4121	0E	MVI C, FF	Inner loop of delay
4122	FF		
4123	0D	DCR C	Decrement the inner loop value
4124	C2	JNZ 4123	Jump on no zero to decrement the inner loop
4125	23		
4126	41		
4127	05	DCR B	Decrement the outer loop value
4128	C2	JNZ 4121	Jump on no zero to decrement the outer loop
4129	21		
412A	41		
412B	C9	RET	Return to main program


A3
A4
A5
FA1
74138
A
B
C
G1
G2B
G2A
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
CS1
CS2
CS3
T1
T2
IOR
IOW
A6
A7
7400
U8
U7


P2
DAC1
DAC2
ADC I/P
GND
P3
GND
-12V
+12V
GND
+5V


74125
OUT
D0
CS3


UXT - BUS
A
B
GND
RST
+5V
D0
D1
D2
D3
D4
D5
D6
D7
RST
IOR
IOW
PCLK
MW
MR
INTA
INTR
RST 7.5
RST 6.5
RST 5.5
+5V
+12V
GND
CMOS Vcc
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
ALE
+30V
-12V

Theoretical calculation of frequency

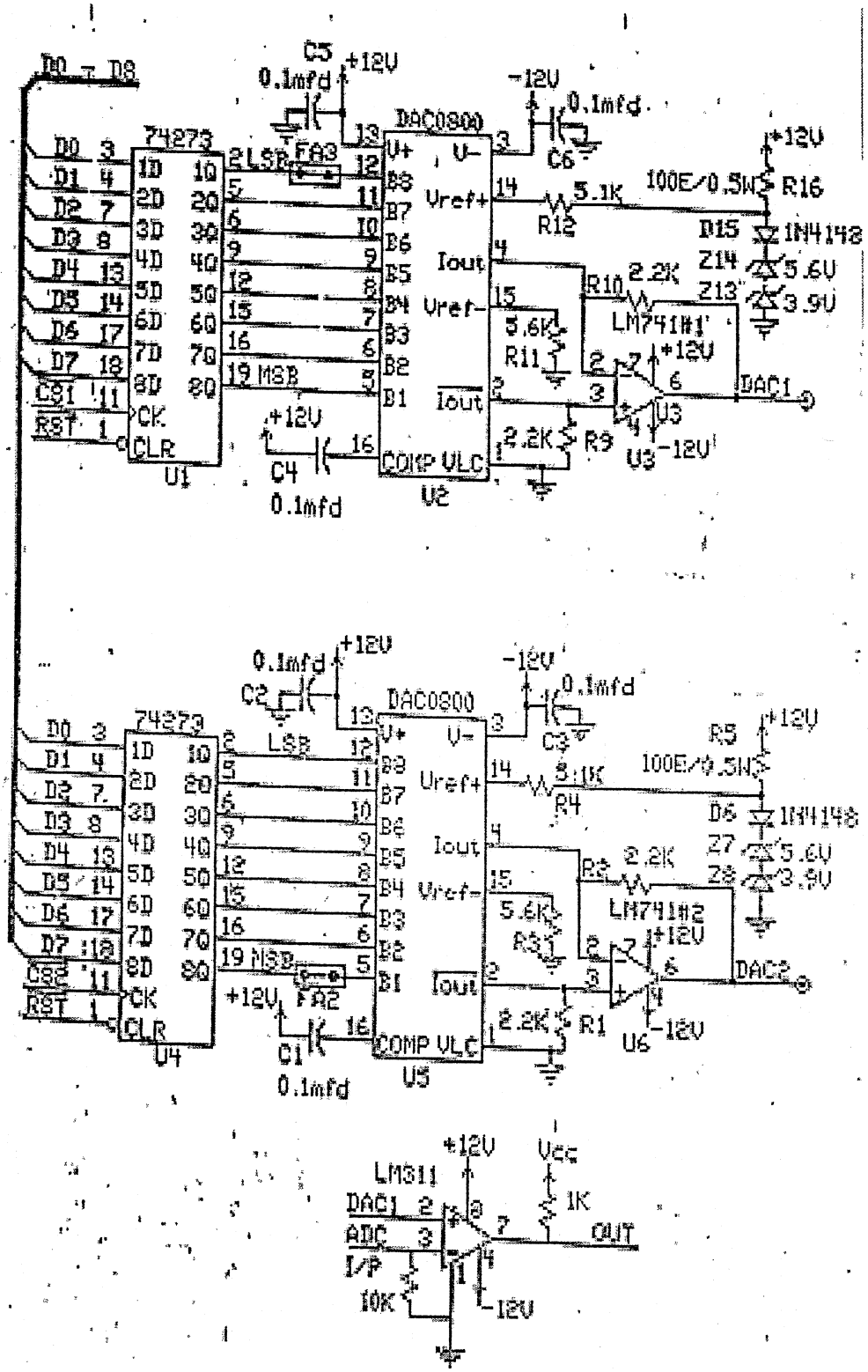
MNEMONICS	NO. OF T-STATES	NO. OF TIMES OCCURRED	TOTAL NO. OF T-STATES
MVI A, C0	7	4	28
OUT C8	10	4	40
CALL D	18	4	72
MVI B, 05	7	5 X 4	140
DCR C	4	5 X FF X 4	20400
JNZ L1	10	1 X 254 X 4	50800
DCR B	4	5 X 4	80
JNZ L2	10	4 X 4	100
RET	10	1 X 4	40
			71760

$t = 71760 / 3 \times 10^{-6} = 0.02372$ sec

$f = 1/t = 41.86$ Hz

VI. TRAPEZOIDAL WAVEFORM

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4100	3E	MVI A, 00	Data for -5V
4101	00		
4102	D3	OUT C8	DAC – 2
4103	C8		
4104	CD	CALL 4200	Call delay routine
4105	00		
4106	42		
4107	3C	INR A	Increment the accumulator
4108	D3	OUT C8	DAC – 2
4109	C8		
410A	C2	JNZ 4107	Jump on no zero to 4107
410B	07		
410C	41		
410D	3E	MVI A, FF	Data for +5V
410E	FF		
410F	D3	OUT C8	DAC – 2
4110	C8		
4111	CD	CALL 4200	Call Delay loop
4112	00		
4113	42		
4114	3D	DCR A	Decrement accumulator
4115	D3	OUT C8	DAC – 2
4116	C8		
4117	C2	JNZ 4114	Jump on no zero to 4114
4118	14		
4119	41		
411A	C3	JMP 4100	Repeat the same
411B	00		
411C	41		


D0 - D8
74273
U1
DAC0800
U2
C5
0.1mfd
+12V
-12V
C6
C4
0.1mfd
LSB
MSB
FA3
5.1K
R12
100E/0.5W
R16
D15
1N4148
Z14
5.6V
Z13
3.9V
2.2K
R10
5.6K
R11
LM741#1
U3
DAC1
2.2K
R9
CS1
RST
CK
CLR
Iout
Vref+
Vref-
COMP VLC
U4
U5
C2
C1
0.1mfd
C3
R4
R5
D6
Z7
Z8
R2
R3
R1
LM741#2
U6
DAC2
FA2
LM311
Vcc
1K
OUT
ADC
I/P
10K

Theoretical calculation of frequency

MNEMONICS	NO. OF T-STATES	NO. OF TIMES OCCURRED	TOTAL NO. OF T-STATES
MVI A, 00	7	1	7
OUT C8	10	255	2550
INR A	4	255	1020
JNZ 4107	10	254	2540
NO JMP	7	1	7
MVI A, FF	7	1	7
OUT C8	10	1	10
CALL DELAY	18	1	18
MVI B, 05	7	1	7
MVI C, FF	7	5	35
DCR C	4	5 X 255	5100
JNZ 4114	10	5 X 254	12700
NO JMP	7	5 X 1	35
DCR B	4	5	20
JNZ 4114	10	4	40
NO JMP	7	1	7
RET	10	1	10
OUT C8	10	255	2550
DCR A	4	255	1020
JNZ 4100	10	254	2540
NO JMP	7	1	7
JMP	10	1	10
			30240

$t = 30240 / 3 \times 10^6 = 0.0108$ sec

$f = 1/t = 99.2$ Hz

DELAY LOOP

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4200	06	MVI B, 05	Outer loop of delay
4201	05		
4202	0E	MVI C, FF	Inner loop of delay
4203	FF		
4204	0D	DCR C	Decrement the inner loop value
4205	C2	JNZ 4204	Jump on no zero to decrement the inner loop
4206	04		
4207	42		
4208	05	DCR B	Decrement the outer loop value
4209	C2	JNZ 4202	Jump on no zero to decrement the outer loop
420A	02		
420B	42		
420C	C9	RET	Return to main program

RESULT

Thus the square wave, saw tooth wave, Triangular wave, Trapezoidal wave, step wave, sine wave are generated and frequency of the waves are verified with the theoretical value.

EXERCISE

1) Differentiate between Unipolar and Bipolar Digital to Analog Converter?

2) Define Voltage Switching and Current Switching in DAC? What type of Switching is used in DAC 0800?

Exp. No. :
Date :

INTERFACING 8255 PPI WITH 8085 MICROPROCESSOR

AIM:

To study and verify the operations of the 8255A in Mode 1 and Mode 2.

THEORY:

The 8255A (Appendix-B) is a widely used, programmable, parallel I/O device. It can be programmed to transfer data under various conditions, from simple I/O to interrupt I/O.

All the functions of the 8255A can be classified according to two modes: the Bit/Reset (BSR) mode and the I/O mode. The BSR mode is used to set or reset the bits in port C. The I/O mode is further divided into three modes: Mode 0, Mode 1, and Mode 2. In Mode 0, all ports function as simple I/O ports. Mode 1 is a handshake mode whereby ports A and/or B use bits from port C as handshake signals. In the handshake mode two types of I/O data transfer can be implemented: status check and interrupt. In Mode 2, port A can be setup for bidirectional data transfer using handshake signals from port C, and port B can be set up either in Mode 0 or Mode 1.

CONTROL REGISTER

The 8255A has a register called the **control register**. The contents of this register called the **control word**, specify an I/O function for each port. The control word format of the 8255A is shown in figure 2. This register can be accessed to write a control word when A_0 and A_1 are at logic 1.The register is not accessible for a Read operation.

Bit D7 of the control register specifies either the I/O function or the Bit Set/Reset function. If bit D7=01, bits D6-D7 determine I/O functions in various modes, as shown in figure 2. If bit D7=0, port C operates in the Bit Set/Reset (BSR) mode. The BSR control does not affect the functions of ports A and B.

To communicate with peripherals through the 8255A, the following steps are necessary:

- Determine the addresses of ports A, B, and C and of the control register according to the Chip Select logic and address lines A_0 and A_1.
- Write a control word in the control register.
- Write I/O instructions to communicate with peripherals through ports A, B and C.

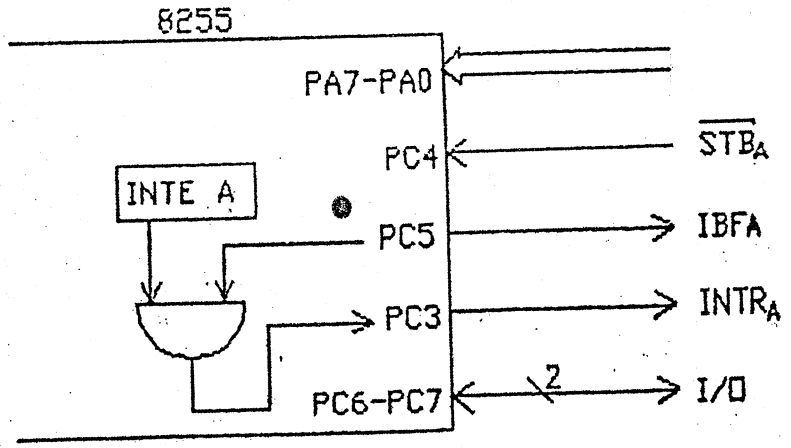
MODE 1 (PORT A):

CONTROL WORD

1	0	1	1	I/O	X	X	X

PC6-PC7 — 1 - Input

0 - Output



EXPERIMENTS IN MODE-1:

EXPERIMENT 1:

To initialize port A as an input port in Mode-1, to enable RST 5.5 interrupt of 8085 CPU and to input the data to the CPU in conjunction with the hand shaking signals.

MAIN PROGRAM:

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4100	3E B6	MVI A, B6	Initialize port A as input port in Mode 1
4102	D3 C6	OUT C6	
4104	3E 09	MVI A, 09	Set the PC_4 bit for $INTE_A$
4106	D3 C6	OUT C6	
4108	F3	DI	Reset the interrupt enable flip flop
4109	3E 08	MVI A, 08	Enable RST 5.5
410B	30	SIM	
410C	FB	EI	Set the interrupt enable flip flop
410D	76	HLT	

ISR(Interrupt Service Routine):

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4200	DB C0	IN C0	Get the data through Port A
4202	32 00 45	STA 4500	Store the result at the location 4500
4205	76	HLT.	

EXPERIMENT-2:

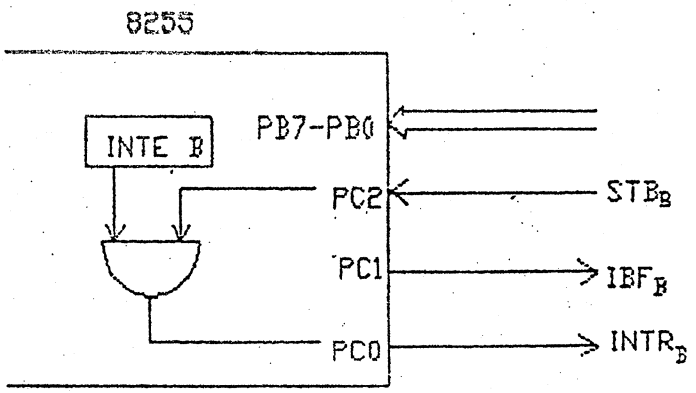
To initialize port A as input port in Mode-1, to enable RST 5.5 and to display the character 'A' on the kit on interrupt.

MAIN PROGRAM:

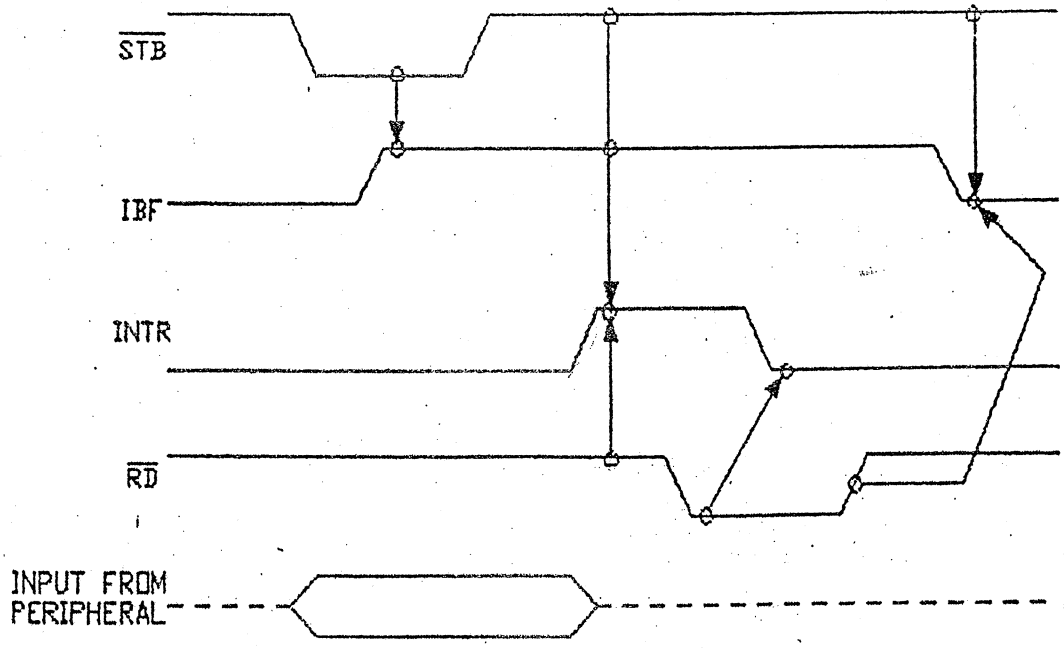
ADDRESS	OPCODE	MNEMONICS	COMMENTS
4100	3E B0	MVI A, B0	Initialize port A as input port in Mode 1
4102	D3 C6	OUT C6	
4104	3E 09	MVI A, 09	Set the PC_4 bit for $INTE_A$
4106	D3 C6	OUT C6	
4108	F3	DI	Reset the interrupt enable flip flop
4109	3E 08	MVI A, 08	Enable RST 5.5
410B	30	SIM	
410C	FB	EI	Set the interrupt enable flip flop
410D	76	HLT	

MODE 1 (PORT B):

CONTROL WORD	1	X	X	X	X	1	1	X



MODE 1 (STROBED INPUT):



ISR (Interrupt Service Routine)

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4200	3E 07	MVI A, 07	
4202	0E 28	MVI C, 28	
4205	21 00 43	LXI H, 4300	
4207	CD 05 00	CALL 0005	Call monitor program
420A	76	HLT	
DATA 4300 – 41;4301 - 24			

EXPERIMENT-3:

To initialize port A as input port in Mode-1, port B as output port in Mode-0, to get data from port A and to output the same through port B.

MAIN PROGRAM:

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4100	3E B0	MVI A, B0	Initialize port A as input port in Mode 1
4102	D3 C6	OUT C6	
4104	3E 09	MVI A, 09	Set the PC_4 bit for $INTE_A$
4106	D3 C6	OUT C6	
4108	F3	DI	Reset the interrupt enable flip flop
4109	3E 08	MVI A, 08	Enable RST 5.5
410B	30	SIM	
410C	FB	EI	Set the interrupt enable flip flop
410D	76	HLT	

ISR(Interrupt Service Routine):

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4200	DB C0	IN C0	Get the data through Port A
4202	D3 C2	OUT C2	Out the data through Port B
4204	CF	RST 1	

EXPERIMENT-4:

To initialize port A as an input port in Mode-0 and port B as an output port in Mode-1.

MAIN PROGRAM:

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4100	3E 94	MVI A, 94	Initialize port A as input port in Mode 0 and Port B as output port in Mode 1
4102	D3 C6	OUT C6	
4104	3E 05	MVI A, 05	Set the PC_2 bit for $INTE_B$
4106	D3 C6	OUT C6	
4108	3E 01	MVI A, 01	
410A	D3 C2	OUT C2	
410C	F3	DI	Reset the interrupt enable flip flop
410D	3E 08	MVI A, 08	Enable RST 5.5
410F	30	SIM	
4110	FB	EI	Set the interrupt enable flip flop
4111	76	HLT	

ISR(Interrupt Service Routine):

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4200	DB C0	IN C0	Get the data through Port A
4202	D3 C2	OUT C2	Out the data through Port B
4205	76	HLT.	

EXPERIMENTS IN MODE-2:

EXPERIMENT-1:

To initialize port A as input port in Mode-2.

MAIN PROGRAM:

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4100	3E C0	MVI A, C0	Initialize port A as input port in Mode 1
4102	D3 C6	OUT C6	
4104	3E 09	MVI A, 09	Set the PC_4 bit for $INTE_A$
4106	D3 C6	OUT C6	
4108	F3	DI	Reset the interrupt enable flip flop
4109	3E 08	MVI A, 08	Enable RST 5.5
410B	30	SIM	
410C	FB	EI	Set the interrupt enable flip flop
410D	76	HL	

ISR(Interrupt Service Routine):

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4200	DB C0	IN C0	Get the data through Port A
4202	32 00 45	STA 4500	Store the result at the location 4500
4205	76	HLT.	

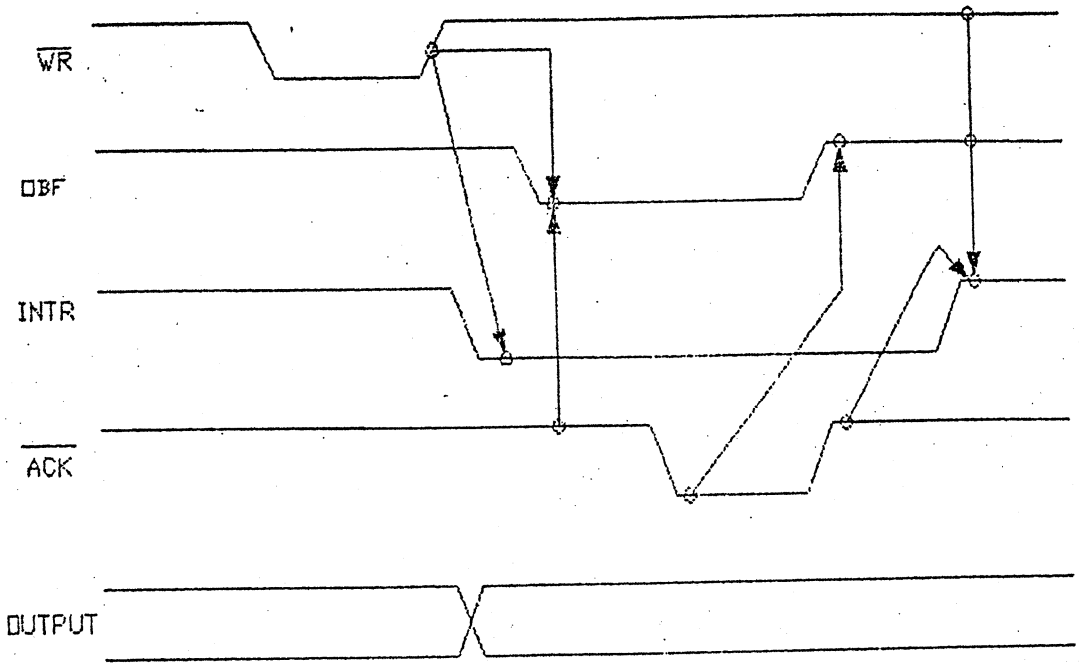
EXPERIMENT-2:

To initialize port A as input port in Mode-2 and port B as output port in Mode-0, to get data from port A and to output the same through port B.

MAIN PROGRAM:

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4100	3E C0	MVI A, C0	Initialize port A as input port in Mode 2 and Port B as output port in Mode 0
4102	D3 C6	OUT C6	
4104	3E 09	MVI A, 09	Set the PC_4 bit for INTE A
4106	D3 C6	OUT C6	
4108	F3	DI	Reset the interrupt enable flip flop
4109	3E 08	MVI A, 08	Enable RST 5.5
410B	30	SIM	
410C	FB	EI	Set the interrupt enable flip flop
410D	76	HLT	

MODE 1 (STROBED OUTPUT):



ISR(Interrupt Service Routine):

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4200	DB C0	IN C0	Get the data through Port A
4202	D3 C2	OUT C2	Out the data through Port B
4205	76	HLT.	

Note : (For the details of 8255 Internal Block Diagram, Pin Configuration and the Control Word Format refer Data sheet given in Appendix-B)

RESULT:

Thus the operations of the 8255A in Mode 1 and Mode 2 have been studied and verified experimentally.

EXERCISE:

1. List the operating modes of the 8255A Programmable Peripheral Interface.
2. Specify the handshake signals and their functions if port A of the 8255A is set up as an output port in Mode 1.
3. List the necessary conditions to generate INTR when port A is set up as an output port in Mode 1.
4. Specify the functions of port C in Mode 2.

Exp. No. :
Date :

INTERFACING 8253 PIT WITH 8085 MICROPROCESSOR

AIM:

To study the different modes of 8253 timer interface.

FEATURES OF 8253 TIMER:

1. Three independent 16 bit counters
2. Programmable counter modes.
3. Input clock from DC to 2MHZ.
4. Count binary or BCD.

The control signals with which the 8253 interfaces with the CPU are C5 *, RD *, WR *, A1 and A2. The basic operation of 8253 can be classified as shown in **table1.**

HARDWARE IMPLEMENTATION FOR TIMER SECTION:

In the circuit diagram clock '0' is connected to either PCLK (1.5 MHz) clock which is available at the VXT BUS or to the debounce circuit. Using the debounce circuit, generate a pulse and clock a timer. Similarly clock1 may be connected to PCLK and clock2 may be connected to either PCLK or out0. In a Microprocessor based application, interrupting the processor after a time delay is sometimes necessary. This technique can be employed using a Timer. Provision is made to connect the output of channels to RST 6.5 or RST 5.5. Thus, the output of channels may be used to interrupt the CPU. The output of channel-2 is also made to glow an LED provided on board. In this interface board the gate input of 8253 are pulled high by 3.3 K resistors.

I/O DECODING:

The address lines A3, A4 and A5 are connected to Pin 1, 2 and 3 of 74LS138. Address lines A6 and A7 are NANDed and NAND gate output is connected to Pin 6 of 74LS138. $\overline{IOW}$ and $\overline{IOR}$ are NANDed together and the NAND gate output is connected to Pin5. Pin4 is grounded. Thus 8251 is selected with address.

A7	A6	A5	A4	A3	A2	A1	A0
1	1	0	0	0	X	X	X

Since the address line A1 is connected to control/ data register pin (pin12) of 8251, the control register of 8251 is selected when

A7	A6	A5	A4	A3	A2	A1	A0
1	1	0	0	0	0	1	X

=C2(HEX)

and the data register is selected with the address.

A7	A6	A5	A4	A3	A2	A1	A0
1	1	0	0	0	0	0	X

= C0 (HEX)

The address lines A1 and A2 are connected to the pin A0 (pin 19) and A1 (pin20) of 8253 respectively. Thus the I/O addresses for the control register, channel – 0, channel – 1, channel – 2 are as shown in table2.

DIFFERENT MODES OF 8253:

(I) MODE - 0: INTERRUPT ON TERMINAL COUNT

(a) CONTROL WORD FORMAT

D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	1	0	0	0	0

Control word is 30 H

D7, D6 : channel 0
D5, D4 : LSB first, MSB next
D3, D2, D1 : mode 0
D0 : binary

(b) JUMPER DIAGRAM

(c) CRO CONNECTION

CRO is connected between pin 3 and pin 10 of timer port P3 and the level change has to be seen.

(d) GATE POSITION

Gate is kept high throughout.

(e) THEORY

The output will be initially low after mode set operation. After loading the counter, the output will remain low while counting and on terminal count the output will become high, until reloaded again.

Set the channel - 0 in mode 0. Connect the clk 0 to the debounce circuit and execute the following program.

Table 9.1 Counters and Control Register Selection

CS*	RD*	WR*	A1	A0	Function
0	1	0	0	0	Load Counter 0
0	1	0	0	1	Load Counter 1
0	1	0	1	0	Load Counter 2
0	1	0	1	1	Write Control word
0	0	1	0	0	Read Counter 0
0	0	1	0	1	Read Counter 1
0	0	1	1	0	Read Counter 2
0	0	1	1	1	No operation 3 - state
1	X	X	X	X	Disabled 3 - state
0	1	1	X	x	No operation 3 - state

Table 1

	A7	A6	A5	A4	A3	A2	A1	A0	HEX
Control Register	1	1	0	0	1	1	1	0	CE
Channel 0	1	1	0	0	1	0	0	0	C8
Chanel 1	1	1	0	0	1	0	1	0	CA
Channel 2	1	1	0	0	1	1	0	0	CC

ADDRESS	OPCODE		MNEMONICS	COMMENTS
4100	3E	30	Start: MVI A, 30 H	channel 0 in mode 0
4102	D3	CE	OUT CE H	
4104	3E	05	MVIA,05 H	LSB of count
4106	D3	C8	OUT C8 H	
4108	3E	00	MVI A, OO H	MSB of count
410A	D3	C8	OUT C8 H	
410C		76	HLT	

Using a CRO observe that the output of the channel -0 is initially zero, after giving six clock pulses, the output goes high.

(II) MODE 1: PROGRAMMABLE ONE SHOT

After loading the counter, the output will remain low, following the raising edge of the gate input. The output will go high on the terminal count. It is re-triggerable, hence the output will remain low for the full count after any rising edge of the gate input.

Choose channel - 0, execute the following program. Trigger channel - 0 externally. Give clock pulses through the debounce circuit. Using a CRO verify the output which is initially low, after triggering goes high after 6 clock pulses.

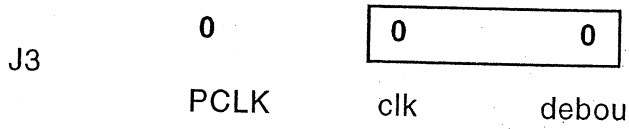
(a) CONTROL WORD FORMAT

D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	1	0	0	1	0

Control word is 32 H

D7, D6 : channel 0
D5, D4 : LSB first, MSB next
D3, D2, D1 : mode 1
D0 : binary

(b) JUMPER DIAGRAM



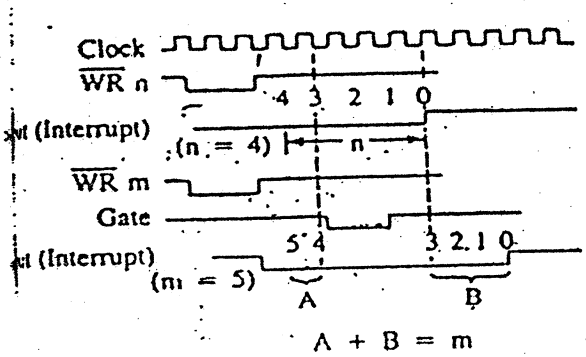
(c) CRO CONNECTION

CRO is connected between pin 3 and pin 10 of timer port P3 and the level change has to be seen.

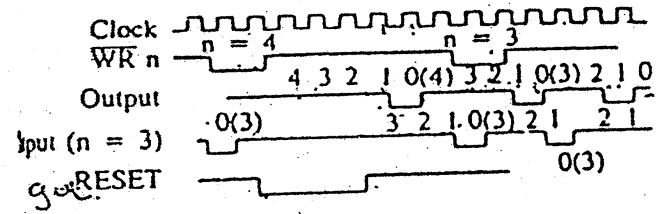
(d) GATE POSITION

To make rising edge of the gate input, pin2(gate 0) to pin10(ground) and then release this connection .

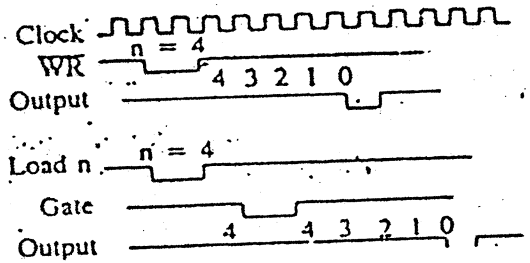
Interupt on Terminal Count



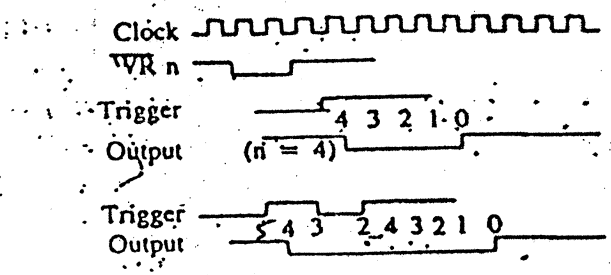
Mode 2: Rate Generator Clock



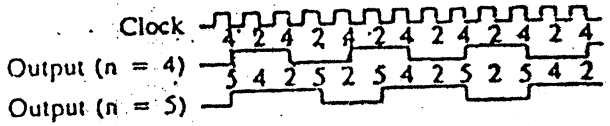
Mode 4: Software Triggered Strobe



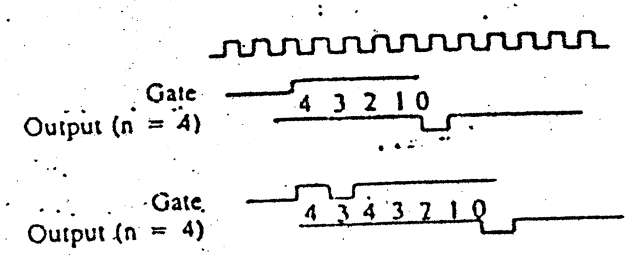
Mode 1: Programmable One-Shot



Mode 3: Square Wave Generator



Mode 5: Hardware Triggered Strobe



ADDRESS	OPCODE	MNEMONICS	COMMENTS
4100	3E 32	MVI A , 32 H	Channel 0 in mode 1
4102	D3 CE	OUT CE H	
4104	3E 05	MVI A , 05 H	LSB of count
4106	D3 C8	OUT C8 H	
4108	3E 00	MVI A , 00 H	MSB Of count
410A	D3 C8	OUT C8 H	
410C	76	HLT	

III MODE 2: RATE GENERATOR

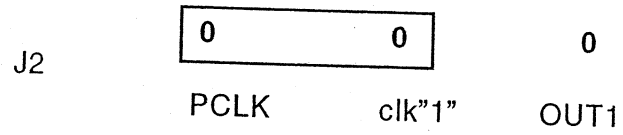
(a) CONTROL WORD FORMAT

D7	D6	D5	D4	D3	D2	D1	D0
0	1	1	1	0	1	0	0

Control word is 74 H

D7, D6 : channel 1
D5, D4 : LSB first, MSB next
D3, D2, D1 : mode 2
D0 : binary

(b) JUMPER DIAGRAM



(c) CRO CONNECTION

CRO is connected between pin 3 and pin 10 of Timer Port P3 and the waveform is traced.

(d) GATE POSITION:

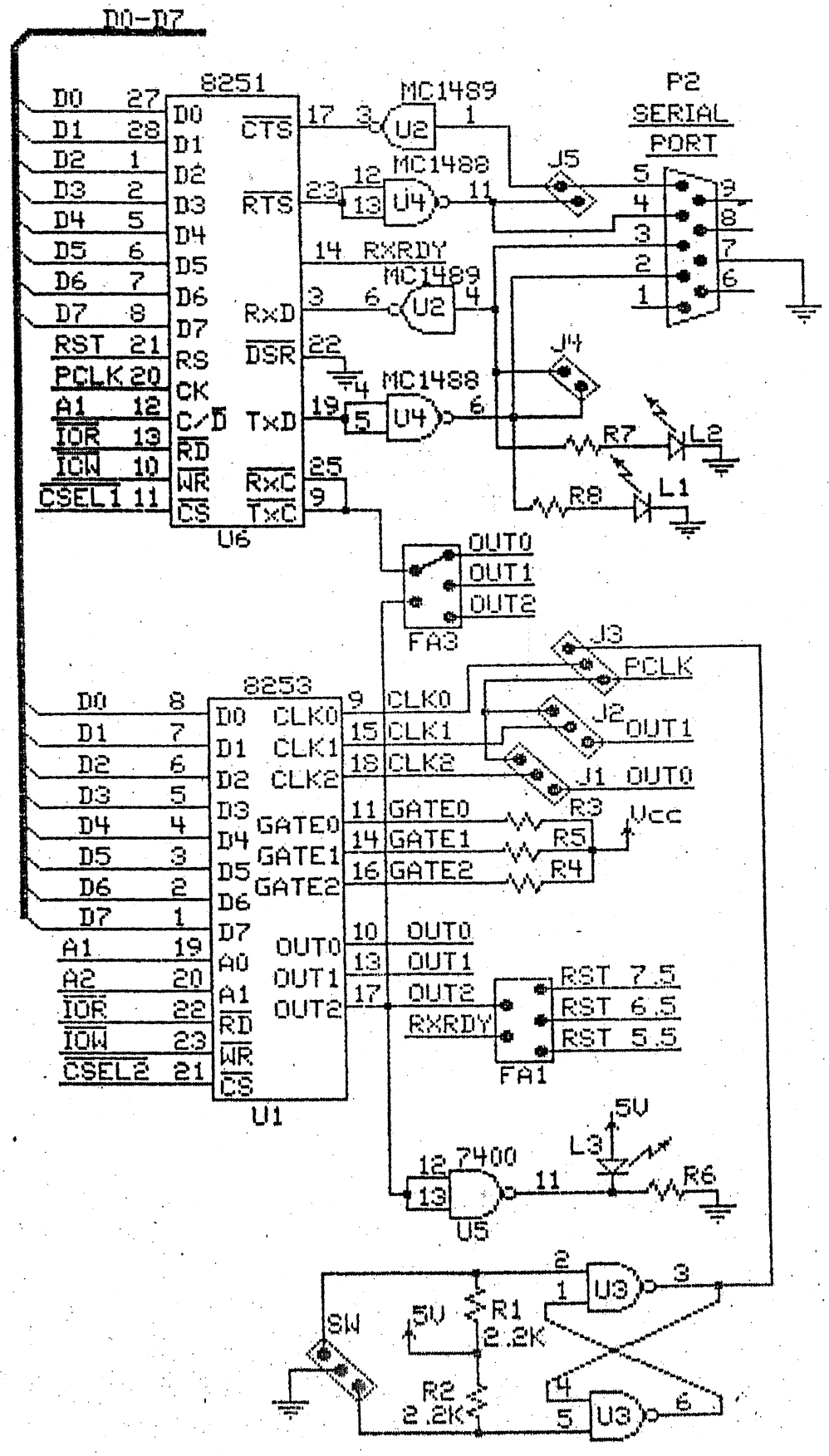
Gate is kept high. If the gate input is made low, the counter is disabled and the output stays high .

(e) THEORY

It is a simple divide by N counter. The output will be low for one period of the input clock . The period from one o/p pulse to the next equals the number of input counts in the count register. If the register is reloaded between o/p pulses, the present period will not be affected but the subsequent period will reflect the new value.

Using this mode, divide the clock present at channel 1 by 10. Connect clk 1 to PCLK.

In the CRO observe simultaneously the input clock to channel1 and clock output.


D0-D7
8251
D0 27
D1 28
D2 1
D3 2
D4 5
D5 6
D6 7
D7 8
RST 21
PCLK 20
A1 12
IOR 13
IOW 10
CSEL1 11
D0
D1
D2
D3
D4
D5
D6
D7
RS
CK
C/D
RD
WR
CS
CTS
RTS
RxD
DSR
TxD
RxC
TxC
17
23
3
22
19
25
9
U6
MC1489
U2
MC1488
U4
14 RXRDY
P2
SERIAL
PORT
J5
J4
R7
L2
R8
L1
OUT0
OUT1
OUT2
FA3
8253
D0 8
D1 7
D2 6
D3 5
D4 4
D5 3
D6 2
D7 1
A1 19
A2 20
IOR 22
IOW 23
CSEL2 21
CLK0
CLK1
CLK2
GATE0
GATE1
GATE2
OUT0
OUT1
OUT2
RD
WR
CS
A0
A1
9
15
18
11
14
16
10
13
17
U1
J3
PCLK
J2
OUT1
J1
OUT0
R3
R5
R4
Vcc
RST 7.5
RST 6.5
RST 5.5
RXRDY
FA1
5V
L3
R6
7400
U5
SW
R1
2.2K
R2
2.2K
U3

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4100	3E 74	MVI A,74 H	Channel 1 in mode 2
4102	D3 CE	OUT CE H	
4104	3E 0A	MVI A,0A H	LSB of count
4106	D3 CA	OUT CA H	
4108	3E 00	MVI A, OO H	MSB of count
410A	D3 CA	OUT CA H	
410C	76	HLT	

IV. MODE 3: SQUARE WAVE GENERATOR

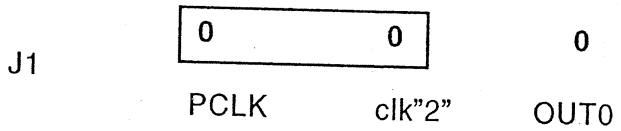
(a) CONTROL WORD FORMAT

D7	D6	D5	D4	D3	D2	D1	D0
1	0	1	1	0	1	1	0

Control word is B6 H

D7, D6 : channel 2
D5, D4 : LSB first, MSB next
D3, D2, D1 : mode 3
D0 : binary

(b) JUMPER DIAGRAM



(c) CRO CONNECTION

CRO is connected between pin3 and pin10 of Timer Port P3 and the waveform is traced.

(d) GATE POSITION

In mode 3, gate is kept high.

(e) THEORY

It is similar to mode 2 except that the o/p will remain high until one half of count and go low for the other half for even number of count. If the count is odd, the output will be go high for (count+1)/2 and low for (count-1) /2. This mode is used to generate baud rate for 8251A. Utilize this mode; a square wave of frequency 150KHZ at channel 2 is generated.

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4100	3E B6	MVI A, B6 H	Channel 2 in mode 3
4102	D3 CE	OUT CE H	
4104	3E 0A	MVI A,0A H	LSB of count
4106	D3 CC	OUT CC H	
4108	3E 00	MVI A, OO H	MSB of count
410A	D3 CC	OUT CC H	
410C	76		HLT

Set the jumper so that clock2 of 8253 is connected to Pclk of frequency 1.5MHz. This program divides this Pclk by 10 and thus the o/p at channel 2 is 150KHz.

Vary the frequency by varying the count. Here the maximum count is FFFF. So at the most, the square wave will remain high for 7FFF counts and remain low for 7FFF counts. Thus with the input clock frequency of 1.5MHz which corresponds to a period of 0.67 microsecond, the resulting square wave has an on time of 0.0281 sec and an OFF time of 0.0281 sec.

To increase the time period of the square wave set the jumper so that clk2 of 8253 is connected to out 0. Using the above mentioned program, output is a square wave of frequency 150 KHz at channel 0. Now this is the clk to channel 2 with 150KHz as the input frequency, calculate the maximum period of square wave.

V. MODE 4: SOFTWARE TRIGGERED STROBE:

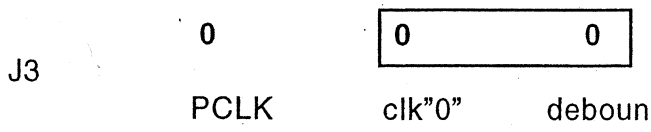
(a) CONTROL WORD FORMAT

D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	1	1	0	0	0

Control word is 38 H

D7, D6 : channel 0
D5, D4 : LSB first, MSB next
D3, D2, D1 : mode 4
D0 : binary

(b) JUMPER DIAGRAM



(c) CRO CONNECTION

CRO is connected between pin3 and pin10 of Timer Port P3 and the waveform is traced.

(d) GATE POSITION

In mode 4, gate is kept high.

(e) THEORY

The output is high after mode is set and also during counting. On terminal count, the output will go low for one clock period and becomes high again. This mode can be used for interrupt generation. In this interface board, the output of channel – 2 may be connected to RST 7.5 or RST 6.5 or RST 5.5.

Connect the OUT 2 of 8253 to RST 5.5 (RST 5.5 is the interrupt meant for user in our kits). Moreover connect the output of channel – 0 to clk 2. Execute the following program and observe that the CPU is interrupted after a time delay of 0.1 seconds.

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4100	3E 38	MVI A, 38 H	
4102	D3 CE	OUT CE H	
4104	3E 05	MVI A,05 H	LSB of count
4106	D3 C8	OUT C8 H	
4108	3E 00	MVI A, OO H	MSB of count
410A	D3 C8	OUT C8 H	
410C	76	HLT	

MODE 5: HARDWARE TRIGGERED STROBE

(a) CONTROL WORD FORMAT

D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	1	1	0	1	0

Control word is 34 H

D7, D6 : channel 0
D5, D4 : LSB first, MSB next
D3, D2, D1 : mode 5
D0 : binary

(b) JUMPER DIAGRAM

(c) CRO CONNECTION

CRO is connected between pin3and pin10 of the Timer Port P3 connecter and level change has to be seen.

(d) GATE POSITION

To make rising edge of gate 0 connect pin2 to ground then release the connection, therefore gate is kept on triggered.

(e) THEORY

Counter starts counting after rising edge of trigger Input and Output goes low for a clock period on terminal count. The counter is retriggerable. Execute the previous program by modifying the control word and by giving the external trigger input.

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4100	3E 3A	MVI A, 3A H	
4102	D3 CE	OUT CE H	
4104	3E 05	MVI A,05 H	LSB of count
4106	D3 C8	OUT C8 H	
4108	3E 00	MVI A, OO H	MSB of count
410A	D3 C8	OUT C8 H	
410C	76	HLT	

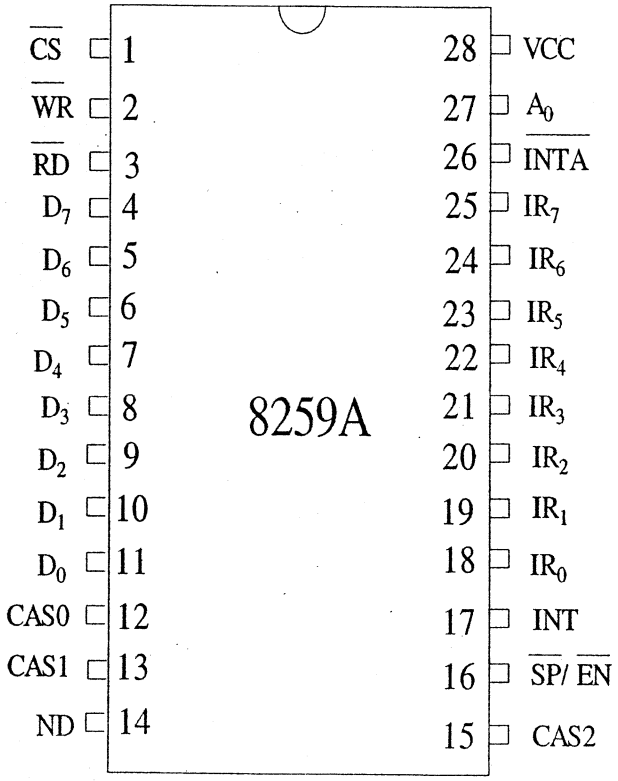
RESULT:

The different modes of 8253 timer operation were studied.

EXERCISE :

1. List the different modes of operation of 8253 timer .
2. Describe the working of interrupt on terminal count mode.
3. Explain the mode III operation of 8253.
4. What is the maximum delay that can be used with 8253 with a 2MHZ clock?

PIN CONFIGURATION



PIN NAMES

D_0	Databus(Bidirectional)
$\overline{RD}$	Read-input
$\overline{WR}$	Write input
A_0	Command select address
$\overline{CS}$	Chip select
CAS2 - CAS0	Cascade lines
$\overline{SP}/\overline{EN}$	slave program/ Enable buffer
INT	Interrupt output
$\overline{INTA}$	Interrupt Acknowledge Input
IR_0 - IR_7	Interrupt Request Input

Exp. No. :
Date :

(A) INTERFACEING 8259 PIC WITH 8085 MICROPROCESSOR

AIM

To interface the 8259 programmable interrupt controller (PIC) with 8085 microprocessor kit.

THEORY

The PIC functions as an overall manager in an interrupt driven system environment. It accepts request from the peripheral equipment, determines which of the incoming requests is of highest importance, ascertains whether the incoming request has a higher priority value than the level currently being serviced and issues an interrupt to the CPU based on this determination.

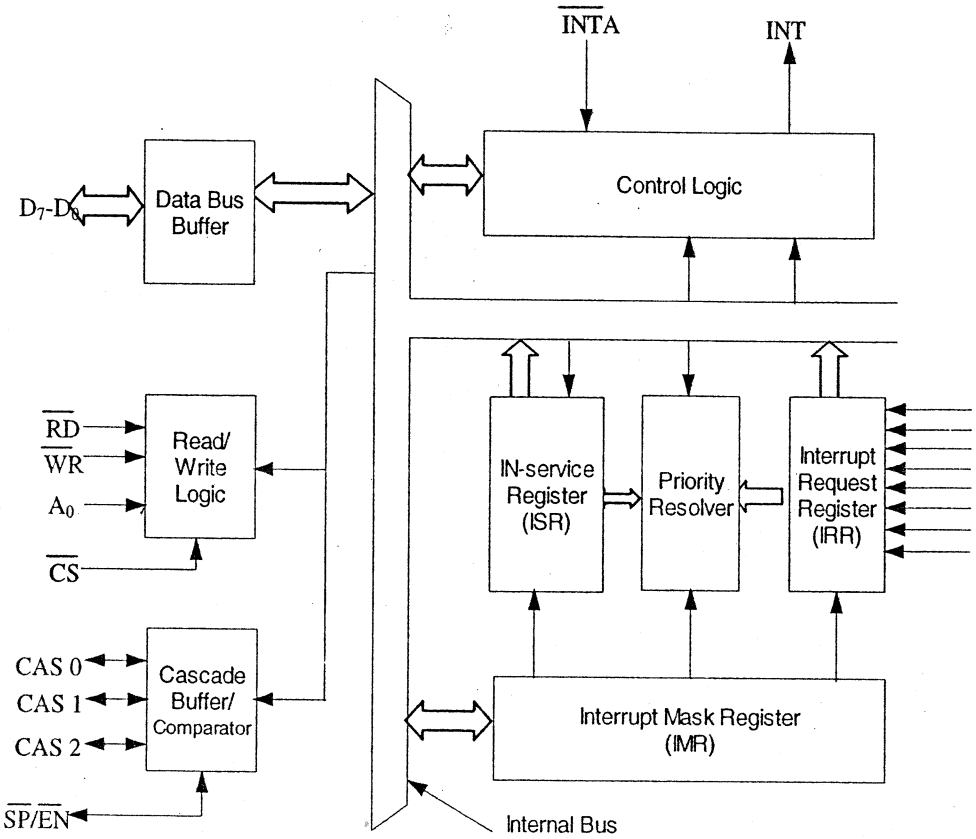
The special features of 8259 are

1. 8 level priority controller
2. Expandable to 64 levels
3. Programmable interrupt modes
4. Individual request mask capability
5. Software programmable capability of all modes of interrupts servicing.

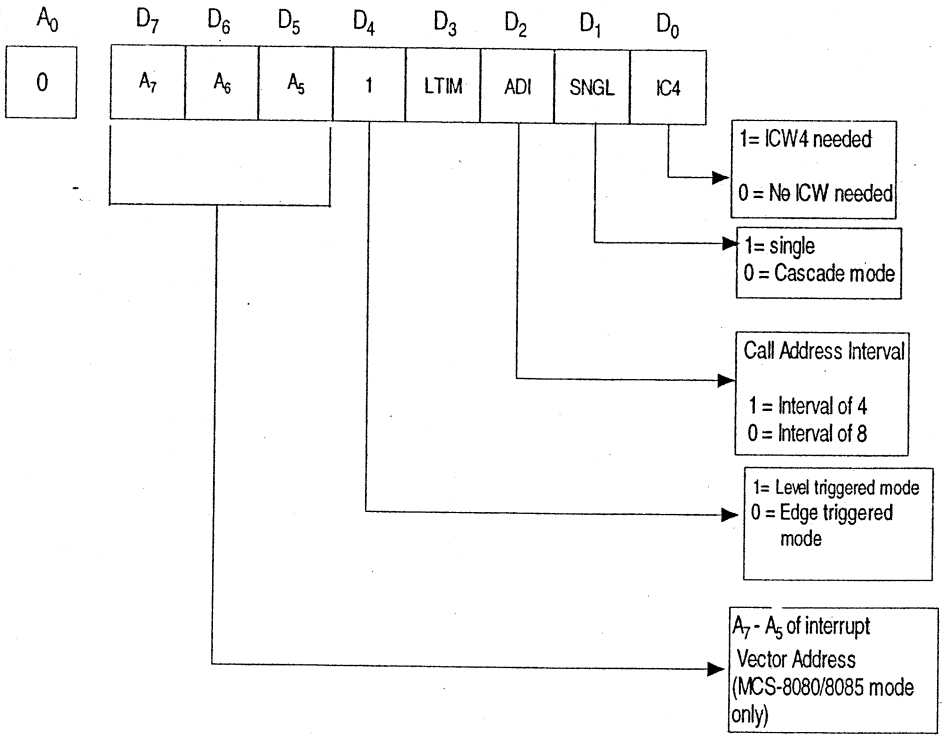
8259 has multichannel priority interrupts. It has several modes, permitting optimization for a variety of system requirements.

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4100	F3	DI	Disable Interrupt
4101	3E 13	MVI A, 13	Load ICW1 into register A
4103	D3 C0	OUT C0	Move ICW1 into 8259
4105	3E 50	MVI A, 50	Load ICW2 into register A
4107	D3 C2	OUT C2	Move ICW2 into 8259
4109	3E 02	MVI A, 02	Load ICW4 into register A
410B	D3 C2	OUT C2	Move ICW4 to 8259
410D	3E 00	MVI A, 00	Load OCW1 into register A
410F	D3 C2	OUT C2	Move OCW1 into 8259
4111	FB	EI	Enable Interrupt
4112	00	NOP	
4113	00	NOP	Wait to accept interrupt
4114	00	NOP	
4115	00	NOP	
4116	C3 12 41	JMP 4112	

BLOCK DIAGRAM



ICW1



INTERRUPT SERVICE ROUTINE: ISR0

ADDRESS	OPCODE	MNEMONICS	COMMENTS
5000	21 00 43	LXI H, 4300H	Load the address of data to be displayed in HL
5003	CD 00 45	CALL DISPLAY	Call display subroutine
5006	C9	RET	

INTERRUPT SERVICE ROUTINE: ISR1

ADDRESS	OPCODE	MNEMONICS	COMMENTS
5008	21 01 43	LXI H, 4301H	Load the address of data to be displayed in HL
500B	CD 00 45	CALL DISPLAY	Call display subroutine
500E	C9	RET	

INTERRUPT SERVICE ROUTINE: ISR2

ADDRESS	OPCODE	MNEMONICS	COMMENTS
5010	21 02 43	LXI H, 4302H	Load the address of data to be displayed in HL
5013	CD 00 45	CALL DISPLAY	Call display subroutine
5016	C9	RET	

INTERRUPT SERVICE ROUTINE: ISR3

ADDRESS	OPCODE	MNEMONICS	COMMENTS
5018	21 03 43	LXI H, 4303H	Load the address of data to be displayed in HL
501B	CD 00 45	CALL DISPLAY	Call display subroutine
501E	C9	RET	

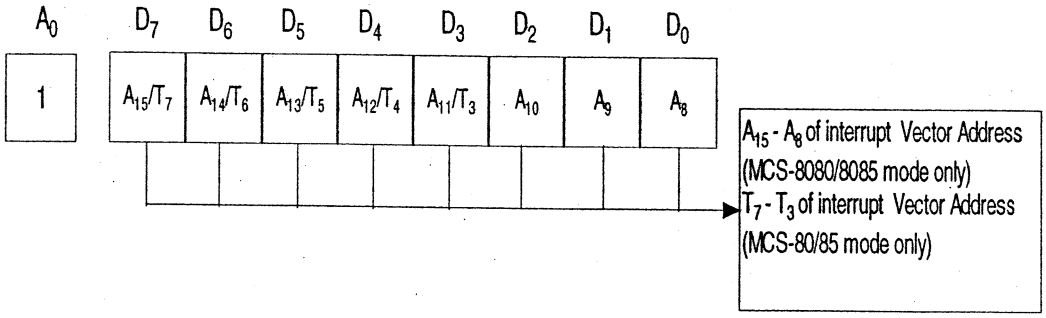
INTERRUPT SERVICE ROUTINE: ISR4

ADDRESS	OPCODE	MNEMONICS	COMMENTS
5020	21 04 43	LXI H, 4304H	Load the address of data to be displayed in HL
5023	CD 00 45	CALL DISPLAY	Call display subroutine
5026	C9	RET	

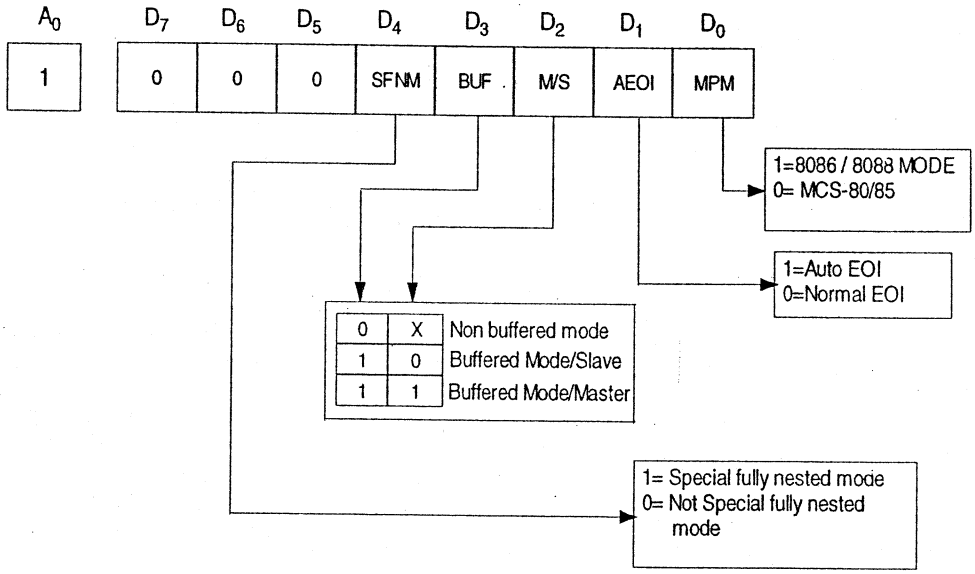
INTERRUPT SERVICE ROUTINE: ISR5

ADDRESS	OPCODE	MNEMONICS	COMMENTS
5028	21 05 43	LXI H, 4305H	Load the address of data to be displayed in HL
502B	CD 00 45	CALL DISPLAY	Call display subroutine
502E	C9	RET	

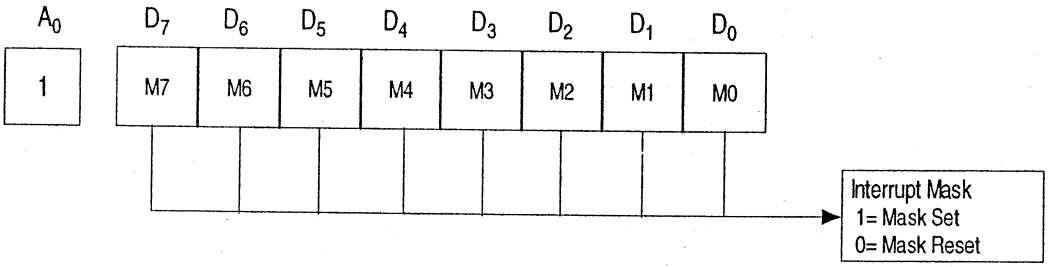
ICW2



ICW4



OCW1



INTERRUPT SERVICE ROUTINE: ISR6

ADDRESS	OPCODE	MNEMONICS	COMMENTS
5030	21 06 43	LXI H, 4306H	Load the address of data to be displayed in HL
5033	CD 00 45	CALL DISPLAY	Call display subroutine
5036	C9	RET	

INTERRUPT SERVICE ROUTINE: ISR7

ADDRESS	OPCODE	MNEMONICS	COMMENTS
5038	21 07 43	LXI H, 4307H	Load the address of data to be displayed in HL
503B	CD 00 45	CALL DISPLAY	Call display subroutine
503E	C9	RET	

DATA

4300 – 30,31,32,33

4304 – 34,35,36,37

Display subroutine (Monitor program-refer Appendix)

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4200	3E 01	MVI A,01	
4202	CD 05 00	CALL 0005	Call monitor program
4205	3E 02	MVI A,02	
4207	CD 05 00	CALL 0005	Call monitor program
420A	3E 03	MVI A,03	
420C	CD 05 00	CALL 0005	Call monitor program
420F	3E 06	MVI A,06	
4211	0E 40	MVI C,40	
4213	46	MOV B,M	Move content of HL reg to B reg
4215	CD 05 00	CALL 0005	
4218	FB	EI	Enable Interrupt
4219	C9	RET	Return to main program

RESULT

Thus the 8259 programmable interrupt controller was interfaced and was checked for ISR 0 to ISR 7.

EXERCISE

1. What are the features of 8259 interrupt controller?

2. Display " NEW" before interrupt and "OLD" after interrupt?

(B) 8259 SIMULATOR INTERFACE

AIM

To interface an 8259 simulator with 8085 microprocessor kit and to display "HELLO" for 0.5 sec on interrupts.

PROGRAM TO DISPLAY "GOOD" (Monitor Program : Appendix-A)

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4100	3E 03	MVI A, 03	Function 03 displays the data
4102	0E 0B	MVI C, 0B	8 digit data will be displayed in all the eight digits
4104	21 00 50	LXI H, 5000H	Address of the data
4107	CD 05 00	CALL 0005	Monitor Subroutine
410A	C3 00 41	JMP 4100	

INTERRUPT SERVICE ROUTINE TO DISPLAY "HELLO"

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4300	3E 03	MVI A, 03	Function 03 displays the data
4302	0E 0B	MVI C, 0B	8 digit data will be displayed in all the eight digits
4304	21 00 42	LXI H, 4200H	Load the address of data
4307	CD 05 00	CALL 0005	Monitor Subroutine
430A	CD 50 42	CALL 4250	Call delay
430D	C9	RET	

DATA: (HELLO)

4200 14 0E 15 15
4204 00 10 10 10

DATA: (GOOD)

5000 13 00 00 00
5004 10 10 10 10

Note : (For the details of Monitor System Calls and Data Format for Seven Segment Display refer Data Sheet given in Appendix - A)

DELAY SUBROUTINE

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4250	11 FF FF	LXI D, FFFFH	Load counter to DE register pair
4253	1B	DCX D	Decrement DE register pair
4254	7B	MOV A,E	Move the content of E register to A
4255	B2	ORA D	Logically OR the content of D register with A
4256	C2 53 42	JNZ 4253	Jump on No Zero to 4253 location
4259	C9	RET	

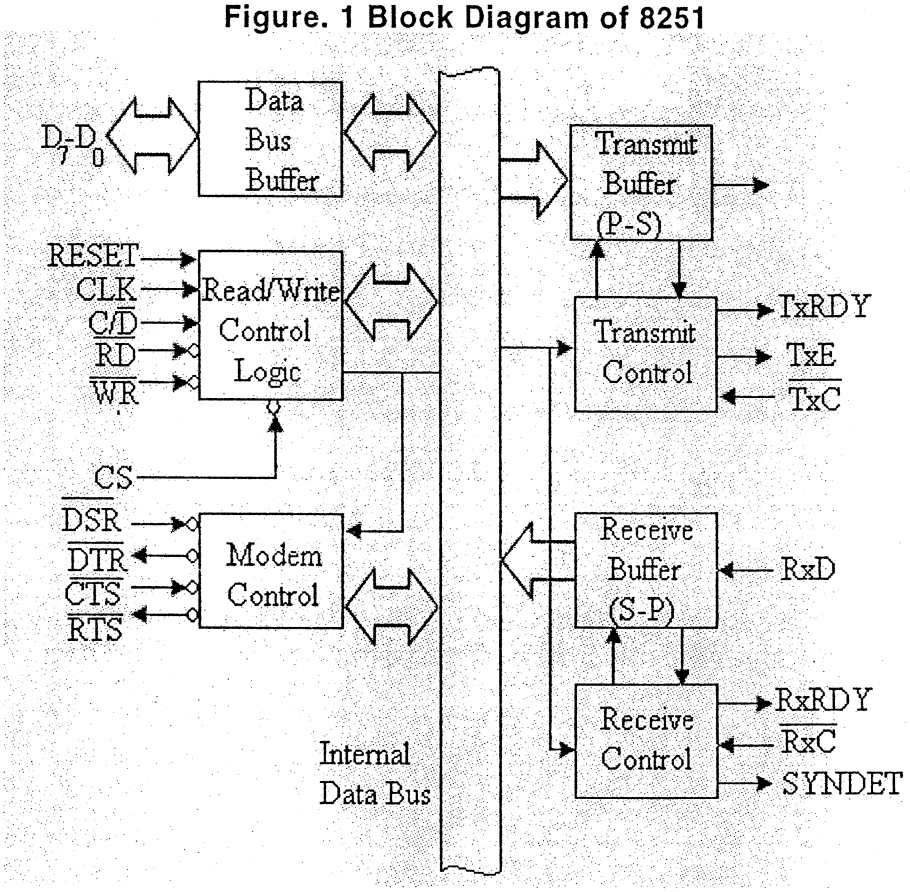
RESULT

The 8259-simulator was interfaced & "HELLO" was displayed for 0.5 sec after the interrupt.

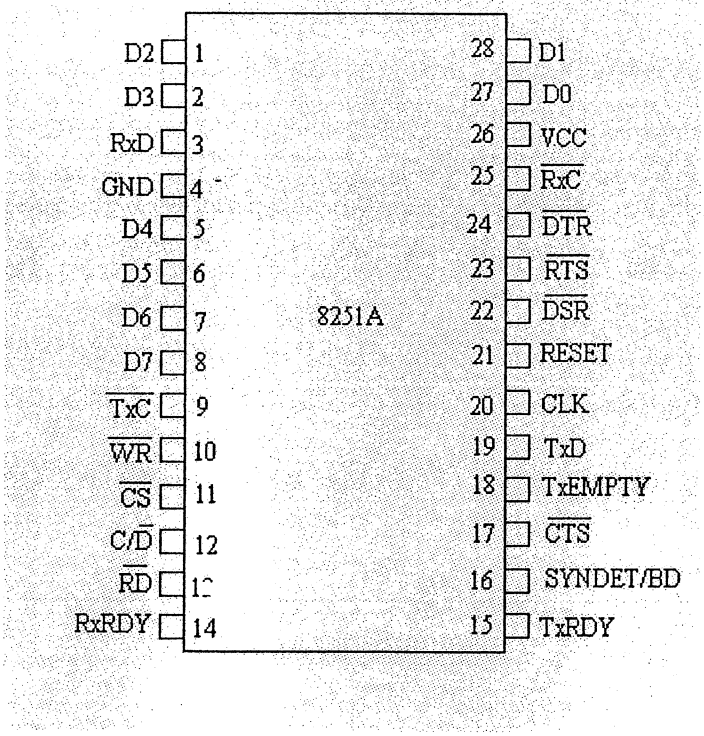
EXERCISE

1. Display the sum of the numbers stored in 4301H, 4302H before interrupt & difference of the numbers after interrupt.
2. Explain hardware interrupts.

Figure. 1 Block Diagram of 8251



Pin Configuration of 8251A



KIT TO KIT DATA TRANSFER USING USART 8251

AIM

To transfer a block of data from one 8085 kit to another 8085 kit using 8251 USART.

APPARATUS REQUIRED

8085 kit with transmitter	: 2 Nos
RS 232 Cable	: 2 meters.

THEORY

The 8251 is a programmable chip designed for synchronous and asynchronous serial data communication, packed in a 28-pin DIP. Figure.1 shows the block diagram of 8251.It includes five sections: Read/Write control Logic, Transmitter, Receiver, Data Bus Buffer and Modem Control.

The control logic interfaces the chip with microprocessor (MPU), determines the functions of the chip according to the control word register and monitors the data flow. The transmitter section converts a parallel word received from the MPU into serial bits and transmits them over the TxD line to a peripheral. The modem control is used to establish data communication through modems over telephone lines.

STATUS READ FORMAT

D7	D6	D5	D4	D3	D2	D1	D0
DSR	SYNDET	FE	OE	PE	TxEMPTY	RxRDY	TxRDY

TxRDY - TxRDY status bit has different meanings from the TxRDY output pin. The form is not conditioned by CTS & TxEN; the latter is conditioned by both CTS & TxEN (i.e.) TxRDY status bit = DB buffer empty TxRDY pin out = DB buffer empty. (CTS-O) (TxEN-1)

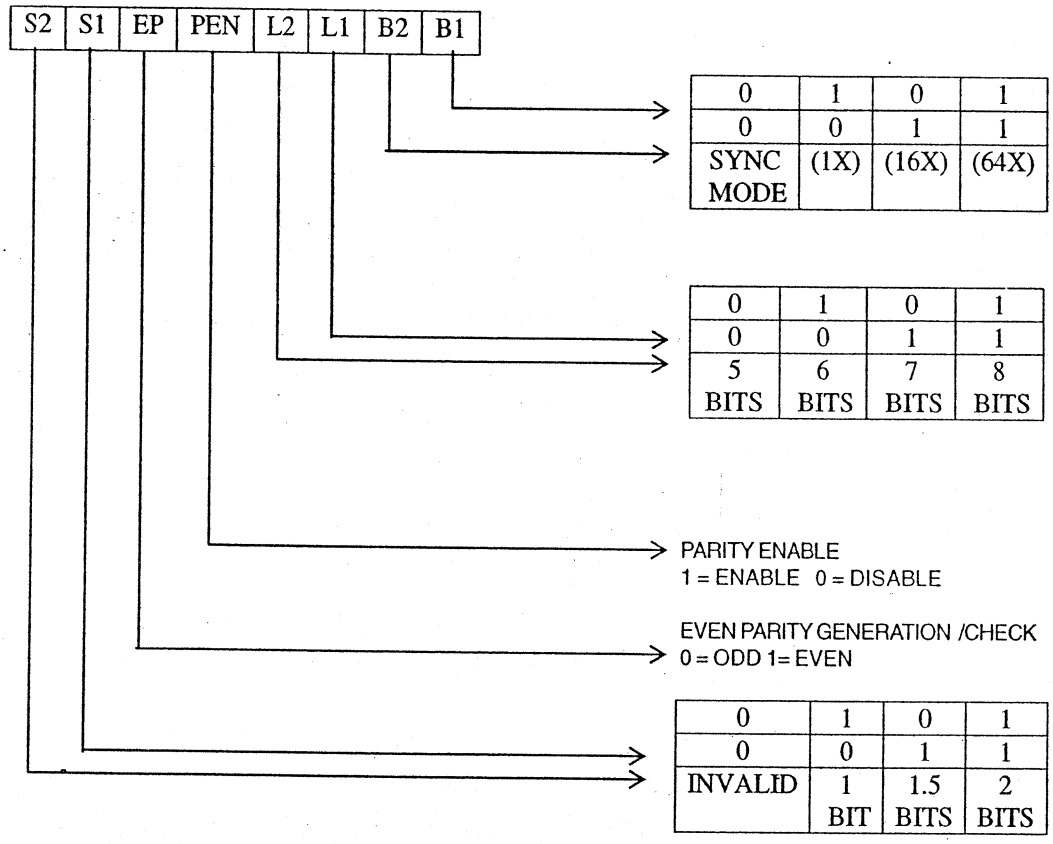
RxRDY - Receiver Ready. This bit indicates that the 8215A contains a character that is ready to be input to the CPU.

TxEMPTY - Transmitter Empty. When the 8251A has no character to transmit this bit will go high.

PE - Parity Error. The PE flag is set when a parity error is detected. It is reset by the ER bit of the command instruction. PE does not inhibit operation of the 8251A .

OE - Overrun Error. The OE flag is set when the CPU does not read a Character before the next one becomes available. OE is reset by the ER bit of the command instruction. OE does not inhibit the operation of 8251A however the previously over run character is lost.

MODE INSTRUCTION FORMAT ASYNCHRONOUS MODE OF 8251A



FE - Framing error (ASYNC only). The FE flag is set when a valid stop bit is not detected at end of every character. It is reset by the ER BIT of the command instruction. FE does not inhibit the operation of 8251A.

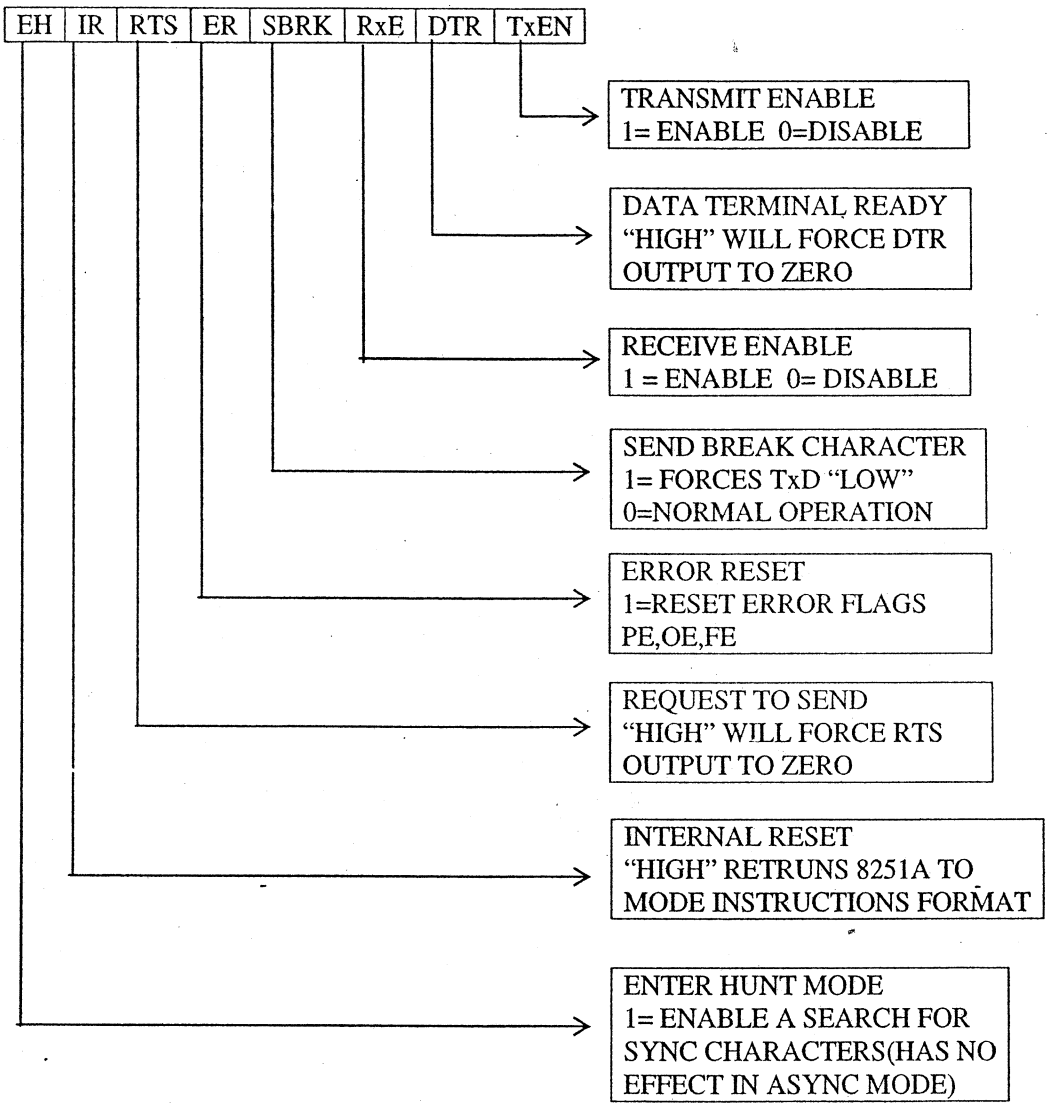
SYNDET - SYNC Detect. This pin is used in synchronous mode for syndet and is used in asynchronous mode for break detect.

DSR - Data set Ready. Indicates that the DSR at zero level.

To transmit a character

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4100	3E 36	MVI A,36H	Timer 8253 control word
4102	D3 07	OUT 07	Timer control port
4104	3E 0A	MVI A,0AH	LSB of count to set the baud rate
4106	D3 04	OUT 04	Counter 0
4108	AF	XRA A	MSB of Count
4109	D3 04	OUT 04	Channel 0
410B	AF	XRA A	
410C	D3 09	OUT 09	8251 control word
410E	3E 40	MVI A,40H	Control word
4110	D3 09	OUT 09	Control register
4112	3E 4E	MVI A,4EH	Mode word
4114	D3 09	OUT 09	Control register
4116	3E 37	MVI A,37H	Command word
4118	D3 09	OUT 09	Control register
411A	06 08	MVI B,08H	Number datas
411C	21 00 45	LXI H,4500H	
411F	DB 09	IN 09	Receive the status word
4121	E6 04	ANI 04	Check for Tx empty
4123	CA 1F 41	JZ 411F	Keep on checking
4126	7E	MOV A,M	First byte transfer
4127	D3 08	OUT 08	Transfer it through data buffer register
4129	23	INX H	Next location
412A	05	DCR B	Decrement counter
412B	C2 1F 41	JNZ 411F	Keep on transferring, zero flag becomes zero
412E	CF	RST 1	
412F	76	HLT	END

COMMAND INSTRUCTION FORMAT OF 8251A

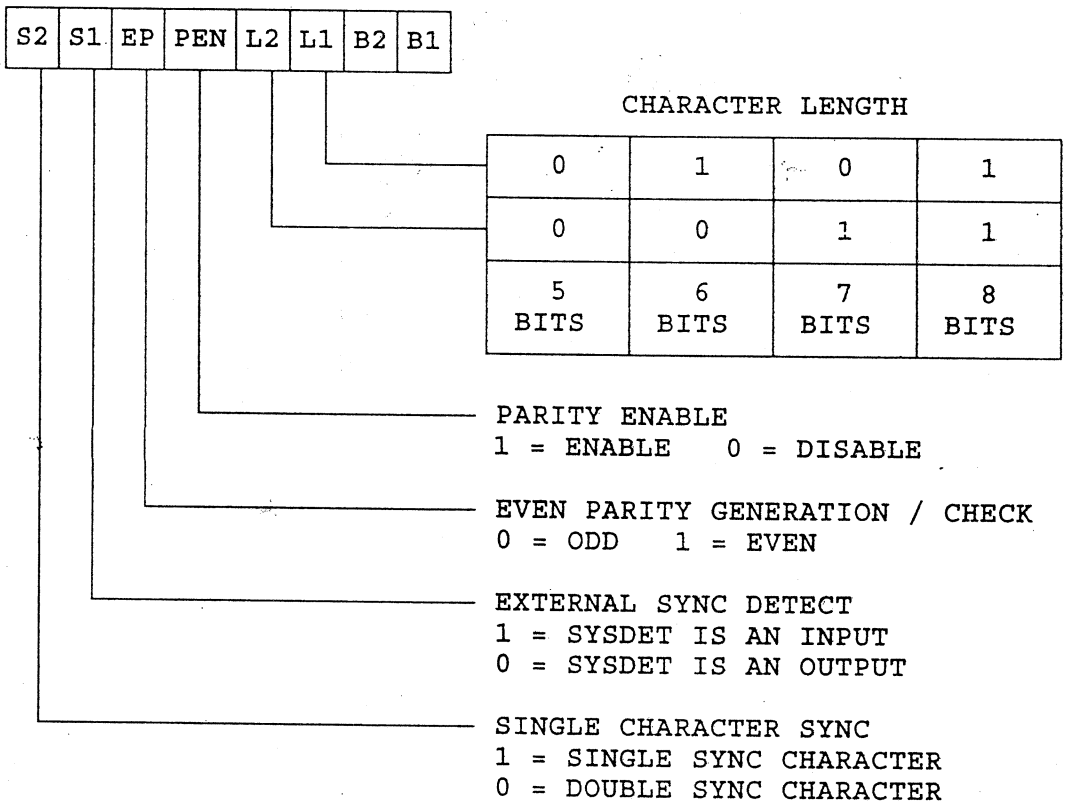


NOTE: Error reset must be performed whenever Rxenable and enter hunt or programmed

To receive a character

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4100	3E 36	MVI A,36H	Timer 8253 control word
4102	D3 07	OUT 07	Timer control port
4104	3E 0A	MVI A,0AH	LSB of count to set the baud rate
4106	D3 04	OUT 04	Counter 0
4108	AF	XRA A	MSB of Count
4109	D3 04	OUT 04	Channel 0
410B	AF	XRA A	
410C	D3 09	OUT 09	
410E	3E 40	MVI A,40H	
4110	D3 09	OUT 09	
4112	3E 4E	MVI A,4EH	8251 control word
4114	D3 09	OUT 09	Control register
4116	3E 37	MVI A,37H	Command register
4118	D3 09	OUT 09	Control register
411A	06 08	MVI B,08H	
411C	21 00 45	LXI H,4500H	
411F	DB 09	IN 09	Receive the status word
4121	E6 02	ANI 02	Check for Tx empty
4123	CA 1F 41	JZ 411F	Keep on checking
4126	DB 08	IN 08	
4128	77	MOV M,A	First byte received
4129	23	INX H	Next location
412A	05	DCR B	Decrement counter
412B	C2 1F 41	JNZ 411F	Keep on transferring, until zero flag becomes zero
412E	CF	RST 1	
412F	76	HLT	END

SYNCHRONOUS MODE 8251A



RESULT :

The data given in the transmitter is received in the receiver unit.

EXERCISE :

1. Differentiate between synchronous and asynchronous data communication.
2. What are the advantages and disadvantages of parallel and serial data communication?
3. What is meant by baud rate?
4. How to choose the baud rate with 8251?

(A) 8279 KEYBOARD / DISPLAY INTERFACE

AIM :

To Interface the 8279 (keyboard and display interface) with the 8085 microprocessor kit.

DESCRIPTION OF 8279 DEVICE :

The Intel 8279keyboard/display controller is a LSI device, designed to release the processor from performing the time consuming scan and refresh operations. The control and status registers share the odd address and the data buffer register uses the even address. The addressing is according to the following table:

CS	RD	WR	A0	TRANSFER DESCRIPTION
0	1	0	0	Data bus to data buffer register.
0	1	0	1	Data bus to control register.
0	1	0	1	Data buffer register to data bus.
0	0	1	1	Status register to data bus.

For keyboard control, the 8279 constantly scans each row of the keyboard by sending out the row addresses on SL2-SL0 and inputting the signals on the return lines RL0-RL7, which represent the column address.

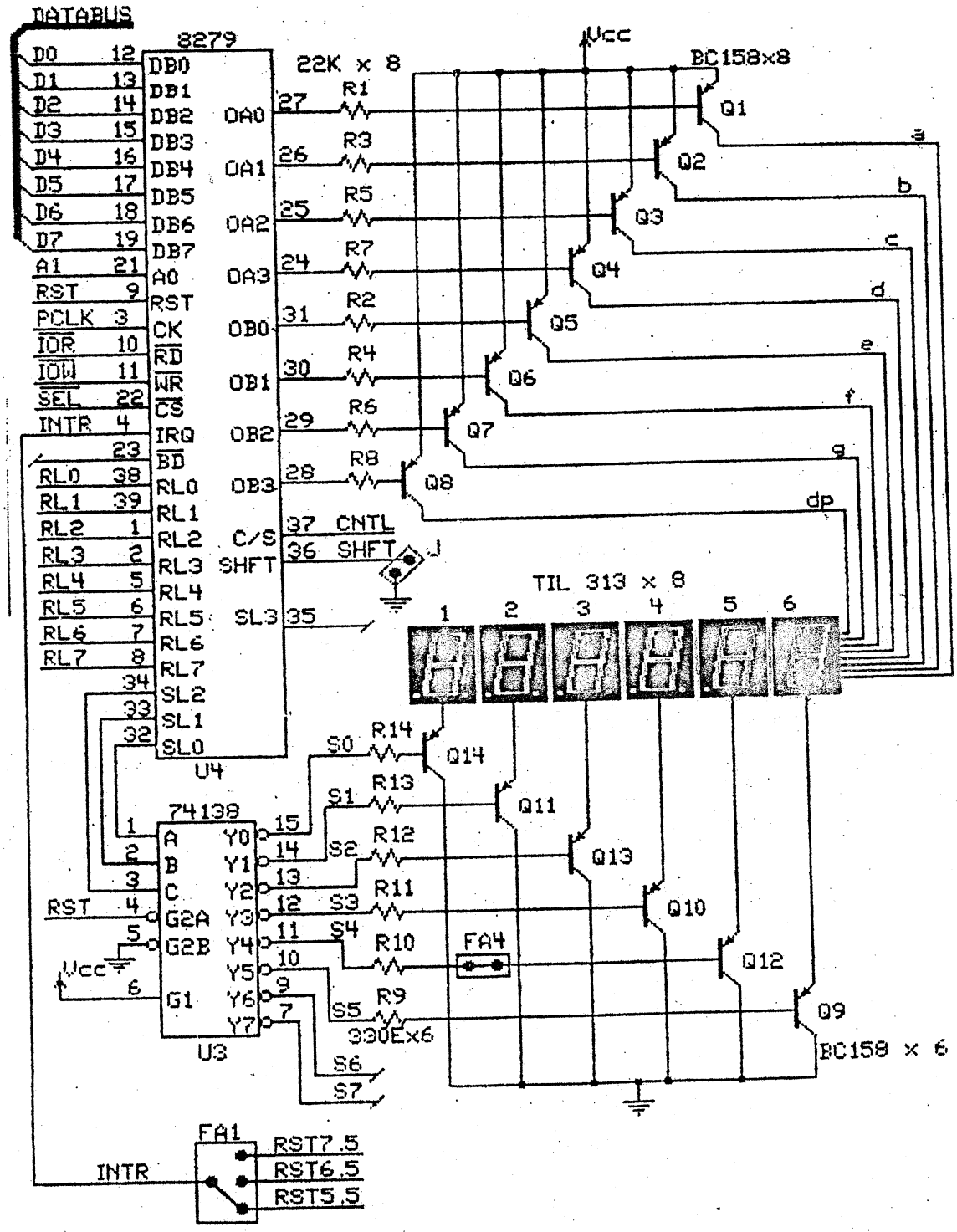
The SL3-SL0 lines are used for both keyboard scanning and display refreshing and will accommodate up to 16 display units. When a depressed key is detected, the key is automatically denounced by weighting for 10ms to check if the same key remains pressed. If a depressed key is detected on 8-bit code corresponding to the key position is assembled by combining the encoded column position, row position, shift status, and control status as shown below.

CNTL SHIFT R R R C C C

Encoded row Position (Scan line address)

Encoded column position (Return line address)

The key position is then entered into the 8x8 first in / first –out (FIFO) sensor memory. The control and timing registers are physically a collection of flags and registers that are accessed by commands which are sent to the 8279's odd address. The three MSBs of a command word determine its type and the remaining 5 bits depends on the type.


DATABUS
8279
22K x 8
Vcc
BC158x8
U4
74138
U3
TIL 313 x 8
FA4
FA1
INTR
RST7.5
RST6.5
RST5.5
330Ex6
BC158 x 6
CNTL
SHFT
RST
Vcc

BLOCK DIAGRAM OF 8279

The block diagram (Figure, 1) shows the major sections of the 8279 device.

1. Keyboard
2. Scan
3. Display and
4. MPU interface

The functions of these are described below.

KEYBOARD SECTION

This section has 8 lines RLO-RL7 that can be connected to 8 columns of a keyboard plus two additional lines: SHIFT and CNTL / STB. The status of shift and control key can be stored along with key code. The keys are automatically debounced and the keyboard operates in two kinds of modes (1). 2 key lock out, and (2), N-key rollover. In the 2 key lockout mode when two keys are pressed almost simultaneously, only the first key is recognized and their code is started in the internal buffer; it can also be set up so that no key is recognized until one key remains pressed.

The keyboard section also includes 8x8 FIFO RAM. The FIFO RAM consists of 8 registers that can store 8 keyboard entries. Each is then read in the order of entries. The status logic keeps track of the entries and provides an IRQ signal when FIFO is not empty.

SCAN SECTION

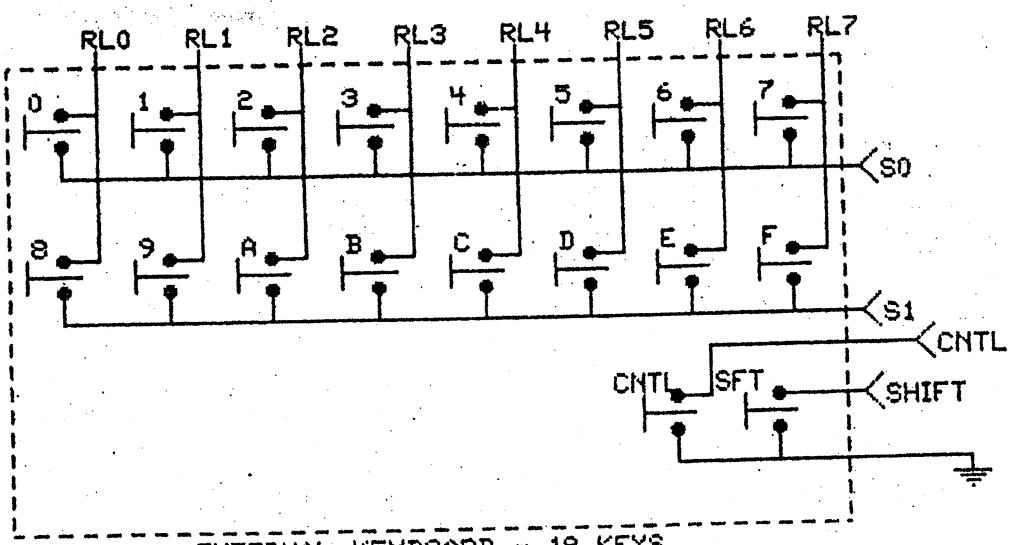
This section has a scan counter and 4 scan lines (SL2 SL0). These lines can be decoded using a 4x16 line decoder to generate 16 lines for scanning. The lines can be connected to the rows of a matrix keyboard or the digit drivers of a multiplexed display.

DISPLAY SECTION

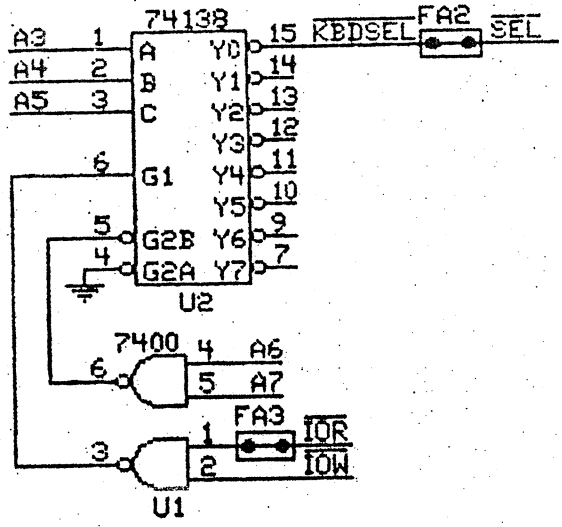
The display section has 8 output lines divided into 2 groups A0... A3 and B0..B3. These lines can be used either as a group of 8 lines are as 2 groups of 4, in conjunction with the scan lines for a multiplexed display. The display can be blanked by using the BD line. This section includes 16x8 Display RAM. The MPU can be read from or Write into any of their registers.

MPU INTERFACE SECTION

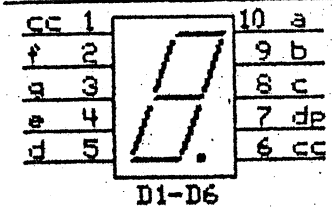
This section includes 8 bi-directional data lines DB0-DB7, one IRQ line and six lines for interfacing, including the buffer address line A0.When A0 is high, signals are interpreted as control signals or status. When A0 is low, signals are interpreted as data. The IRQ line goes high whenever data entries are stored in FIFO. The signal is used to interrupt the MPU to indicate the availability of data.



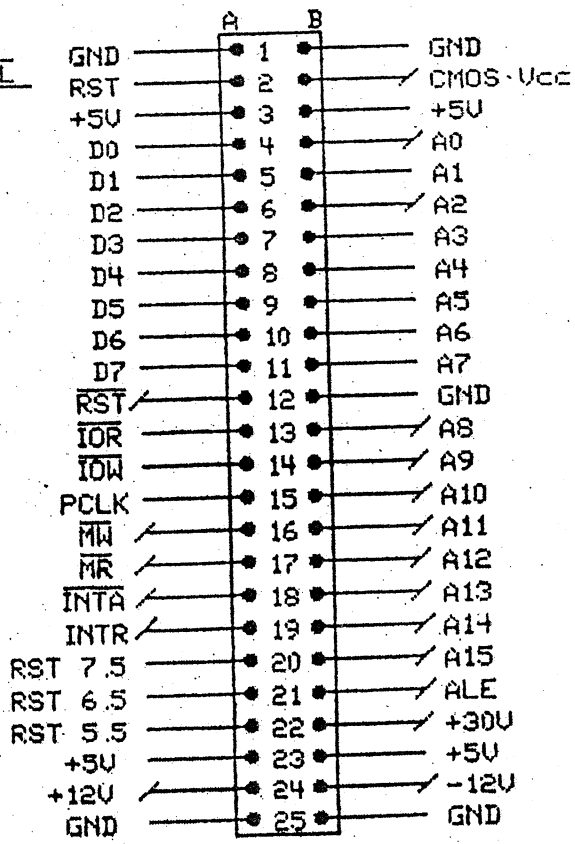
EXTERNAL KEYBOARD - 18 KEYS



PIN DETAILS OF DISPLAY



UXT - BUS



HARDWARE DETAILS OF 8279 INTERFACE CARD

In the hardware diagram, Ics 74LS00(U1) and 74LS138(U2) form the address decoding logic to generate the chip select signal for the 8279.

Address lines A6 and A7 are NANDed and NAD gate output is connected to pin 5 of 74LS138(U2). Similaerlly the IOW, IOR combination output is connected to pin6 of 74LS138. The inputs A,B &C of 74LS138 are connected to address lines A3,A4 &A5 respectively.

The 8272 is selected when the address is

A7	A6	A5	A4	A3	A2	A1	A0
1	1	0	0	0	X	X	X

Since the address line A1 is connected to A0 of 8279. when

A7	A6	A5	A4	A3	A2	A1	A0
1	1	0	0	0	0	1	0 = (C2) Hex

Control / Status register is selected. When

A7	A6	A5	A4	A3	A2	A1	A0
1	1	0	0	0	0	0	0 = (C0) Hex

Data register is selected. The keyboard section consist of 16 keys besides the separate keys for CNTL and SHIFT. Decoded scan display is employed in this interface board . A 3 to 8 decoder 74LS138(U3) is provided for that purpose. Display section consists of 6 seven-segment LED displays.

SOFTWARE EXAMPLES:

The commands are sent on the data bus with CS low and A0 high, which are loaded to the 8279 on the rising edge of IOR. The command word for setting keyboard and display mode is

0	0	0	D	D	K	K	K

DD - Display mode
00 - 8 bit character display-Left entry
01 - 16bit character display-Left entry
10 - 8 bit character display-Right entry
11 - 16 bit character display-Left entry

KKK - Keyboard mode
000 - Encoded scan keyboard mode – 2 key lockout.
001 - Decoded scan keyboard mode – 2 key lockout.
010 - Enclosed scan keyboard mode – N key rollover.
011 - Decoded scan keyboard mode – N key rollover.
100 - Encoded scan sensor matrix.
101 - Decoded scan encoded display scan.
111 - Strobed input decoded display scan.

CLEAR DISPLAY :

The command word for clear display is

1	1	0	CD	CD	CD	CF	CA

CD CD CD

0 X - A0 - A3 , B0 - B3 - 00 = (0000 0000)

1 0 - A0 - A3 , B0 - B3 - 20 = (0010 0000)

1 1 - A0 - A3 , B0 - B3 - - = (1111 1111)

Enable clear display when CD =1 (IfCD = 0, then the contents of RAM will be displayed.)

CF - If CF = 1, FIFO status is cleared, interrupt output line is reset.

CA - Clear all bit has the combined effect of CD and CF. Resynchronizes the internal timing chain. It uses CD for clearing display RAM and clears FIFO status.

WRITE DISPLAY RAM :

1	0	0	A1	A	A	A	A

The write display RAM command word is

This command is written with A0 = 1. All subsequent write information with A0 = 0 will be to the Display RAM

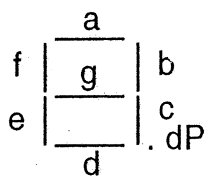
AI – Automatic increment flag. If AI = 1, the row address is incremented after each following read or write to the display RAM.

AAAA – selects one of the 16 rows of the display RAM.

EXAMPLE PROGRAM 1: To display A.

In the seven segment display, the segment definition and the correspondence between data bus, segments enabled and the 8279 bits are follows:

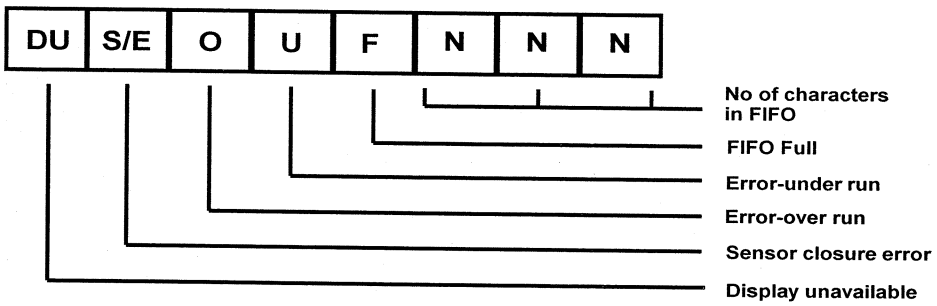
Database Buffer	D7	D6	D5	D4	D3	D2	D1	D0
8279 output	A3	A2	A1	A0	B3	B2	B1	B0
Segments	d	c	b	a	dp	g	f	e



ADDRESS	OPCODE	MNEMONICS	COMMENTS
4100	3E 00	MVI A, 00	Display mode set for left entry
4102	D3 C2	OUT C2	Out to Status or Control register
4104	3E CC	MVI A, CC	command word to clear Display
4106	D3 C2	OUT C2	
4108	3E 90	MVI A, 90	command word to write in display RAM
410A	D3 C2	OUT C2	
410C	3E 88	MVI A, 88	To display 'A'
410E	D3 C0	OUT C0	Out to DATA register
4110	3E FF	MVI A, FF	To blank the other digits
4112	D3 C0	OUT C0	
4114	D3 C0	OUT C0	
4116	D3 C0	OUT C0	
4118	D3 C0	OUT C0	
411A	D3 C0	OUT C0	
411C	76	HLT	

This Program will display A in the first digit and the rest will be blanked.

READ FIFO STATUS

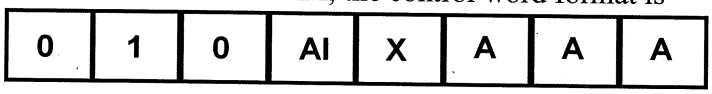


This status word indicates the number of characters in FIFO and indicates whether an error has occurred. There are two types of error possible: over run and under run. Over run occurs when the entry of another character into a full FIFO is attempted. Under run occurs when the CPU tries to rear an empty FIFO.

The FIFO status also has a bit to indicate that the display RAM was unavailable because a clear display or clear all command has not completed its clearing operation.

READ FIFO/SENSOR RAM

To read FIFO/SENSOR RAM, the control word format is



AI- Auto Increment Flag-Irrelevant, if scanned keyboard mode. For sensor matrix mode if AI = 1, then each successive read will be from subsequent row of sensor RAM.

EXAMPLE PROGRAM 3:

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4100	21 50 41	LXI H,4150	Initialise HL pair with the address of the lookup table.
4103	16 08	MVI D,08	
4105	3E 00	MVI A,00	Display mode set
4107	D3 C2	OUT C2	Command word to clear
4109	3E CC	MVI A,CC	Display.
410B	D3 C2	OUT C2	
410D	3E 90	MVI A,90	Command word to write
410F	D3 C2	OUT C2	Display RAM
4111	7E	MOV A,M	
4112	D3 CO	BAK: OUT C0	Blank display.
4114	23	INX H	
4155	15	DCR D	
4116	C2 12 41	JNZ BAK	
4119	DB C2	LOP : IN C2	Cheking for a key closure
411B	E6 07	ANI 07	
411D	CA 19 41	JZ 4119	
4120	3E 40	MVI A,40	Command word red FIFO RAM
4122	DB C2	OUT C2	For getting the key code.
4124	DB CO	IN C0	To get the corresponding
4126	E6 OF	ANI OF	Display code from the lookup table.
4128	6F	MOV L,A	
4129	26 42	MVI H,42	
412B	7E	MOV A,M	
412C	D3 CO	OUT C0	
412E	C3 19 41	JNZ 4119	
4150	FF FF FF FF	FF	; Lookup table
4154	FF FF FF FF	FF	
4200	OC 9F 4A 0B		
4204	99 29 28 8F		
4208	08 09 88 38		
420C	6C 1A 68 E8		

AAA - In the scaned keyboard mode-Irrelevent.

In the sensor matrix mode, it selects one of the 8 rows of sensor RAM.

In the scaned keyboard mode, the 8279 will automatically drive the data for subsequent read in the same sequence in which data firat entered the FIFO.

The FIFO RAM data format is

0	0	E	E	E	X	X	X

CNTL SHIFT SCAN RETURN

This cod will be entered into the FIFO when ever a key is pressed.

RESULT :

Thus the 8279 Keyboard and Display Interface was interfaced with 8085 Microprocessor and the prorgrames were verified.

LIQUID CRYSTAL ALPHANUMERIC DISPLAY INTERFACE

AIM

To interface LCD module with the 8085 microprocessor.

COMPONENTS REQUIRED

8085 microprocessor and liquid crystal alphanumeric display module.

COMPONENT DESCRIPTION

The LCD module includes CMOS VLSI microprocessor controller, which supplies character memory RAM, character generator RAM & ROM and all refresh, control and timing signals.

BASIC SYSTEM INTERFACE

Interfacing the LCD module with a microprocessor bus requires minimum hardware. A +5 volts DC and parallel TTL level ASCII are provided in order to display data. The 4 or 8 bit bus interfaces easily with most popular µp with a minimum of additional hardware and allows such control functions as cursor addressing, display shift and programmable characters to be easily implemented.

The control signals with which the LCD module communicates with microprocessor are the data bus, RS, $R/\overline{W}, E$. The following tables show the LCD interface pin connections.

Pin No	Symbol	FUNCTIONS
1	Vss	Power supply ground
2	V_{DD}	Power supply + 5 V to 25 V dc
3	Vo	LCD driving voltage
4	RS	Set high to read & write alphanumeric data & characters, generate data. Set low to read status and address information and to write control words.
5	$R/\overline{W}$	Set low to write to the module. Set high to read from the display.
6	E	Enable pulse: data is clocked into DAY STAR module and trailing edge of this pulse.
7 8 9 10	DB0 DB1 DB2 DB3	Lower order 4 lines of bi-directional tristate data bus used for data transfer between the mp and the display. These 4 are not used during 4-bit operation.
11 12 13 14	DB4 DB5 DB6 DB7	Higher order 4 bits of bi-directional data bus used for data transferbetween the microprocessor & the display. DB7 can be used for a busy flag.

DETAILED DESCRIPTION OF COMMANDS

1. CLEAR DISPLAY

	RS	R/$\overline{W}$	DB7			DB3			DB0
CODE	0	0	0	0	0	0	0	0	1

With ASCII "space" (20H) into all of the DD RAM, the cursor returns to address 0 (ADD = '00') and the display if it has been shifted returns to the original position.

2. RETURN HOME

	RS	R/$\overline{W}$	DB7			DB3			DB0
CODE	0	0	0	0	0	0	0	1	X

Returns the cursor to Address 0 (ADD = "00") and display if it has been shifted to the original position. The DDRAM contents remain unchanged.

3. ENTRY MODE SET

	RS	R/$\overline{W}$	DB7			DB4			DB0
CODE	0	0	0	0	0	0	0	1	I/D	S

I/D: This mode automatically increments (I/D = 1) or decrements (I/D = 0), the DD RAM address by one every time a character is written to (or) read from the DDRAM. The cursor moves to the right when incremented. The same applies to the left on reading of DDRAM.

S: Automatically shifts the entire display either to the right or left after each character is written. If S = 1, either to the left (when I/D = 1). Therefore the cursor looks as if still and only the display is moved. The display is not shifted, when reading from DDRAM (or) if S = 0.

4. DISPLAY ON/OFF CONTROL [DISPLAY BLANKING]

	RS	R/$\overline{W}$	DB7			DB4			DB0
CODE	0	0	0	0	0	0	1	D	C	B

D: The display is turned ON when D = 1 & OFF when D = 0. When the display is turned OFF by D = 0, the display data remains in the DDRAM and it can be displayed immediately by setting D = 1.

C: The cursor is displayed as the under bar, when C = 1 and not displayed when C = 0. Even if the cursor is not displayed, the function of I/D etc., does not change during display data write. The cursor is displayed using 5 dots in either the 8th line when the 5 x 7 dot character font is selected.

B: The character at the cursor position blinks when B = 1. The blink is done by switching between all block dots and the displayed character at 0.4-second intervals. The cursor under bar and the blink can be set concurrently.

5. CURSOR DISPLAY SHIFT

	RS	R/$\overline{W}$	DB7			DB4				DB0
CODE	0	0	0	0	0	1	S/C	R/L	X	X

Shifts the cursor position (or) display to the right and left without writing (or) reading the display data. This function is used for correction of display.

S/C	R/L	
0	0	Shifts the cursor position to the left (AC is decremented by one)Shifts
0	1	the cursor position to the right.Shifts the display to the left, the
1	0	cursor follows the display.Shifts the entire display to right, the cursor
1	1	follows the display shift.

6. FUNCTION SET

	RS	R/$\overline{W}$	DB7			DB4				DB0
CODE	0	0	0	0	1	DL	N	F	X	X

DL:

Sets interface data length, data is sent or received in 8 bit length when DL = 1 and 4 bit length when DL = 0. When 4-bit length is selected data must be written or read twice for each byte of data.

N: Sets No. of display lines.

F: Sets character font.

N	F	No. of Display lines	Character Font	Duty Factor	Remarks
0	0	1	5 x 7 dots	1/8	-
0	1	1	5 x 10 dots	1/11	-
1	*	2	5 x 7 dots	1/16	Cannot display 2 lines with 5 x 10 dot character font.

7. SET CG RAM ADDRESS

	RS	R/$\overline{W}$	DB7								DB0
CODE	0	0	0	1	A	A	A	A	A	A	

Higher Order Bits — Lower order Bits

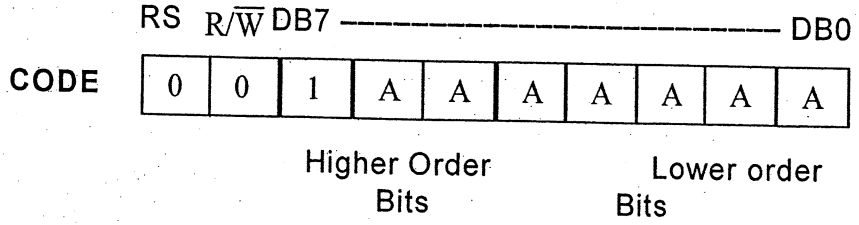
Sets the CG RAM address as a binary number "AAAAAA" in the address counter. Data pertaining to the CG RAM is written to or read from the host system after this command is executed.

Graphic or Control	ASCII (Hexadecimal)
"	22
#	23
$	24
%	25
&	26
'	27
(	28
)	29
*	2A
+	2B
‘	2C
-	2D
.	2E
/	2F
0	30
1	31
2	32
3	33
4	34
5	35
6	36
7	37
8	38
9	39
:	3A
;	3B
<	3C
=	3D
>	3E
?	3F
@	40
A	41
B	42
C	43
D	44
E	45
F	46
G	47
H	48
I	49
J	4A
K	4B
L	4C
M	4D
N	4E
O	4F
P	50

Graphic or Control	ASCII (Hexadecimal)
Q	51
R	52
S	53
T	54
U	55
V	56
W	57
X	58
Y	59
Z	5A
[	5B
\	5C
]	5D
^	5E
-	5F
`	60
a	61
b	62
c	63
d	64
e	65
f	66
g	67
h	68
i	69
j	6A
k	6B
l	6C
m	6D
n	6E
o	6F
p	70
q	71
r	72
s	73
t	74
u	75
v	76
w	77
x	78
y	79
z	7A
{	7B
|	7C
}	7D
~	7E
DEL Delete	7F

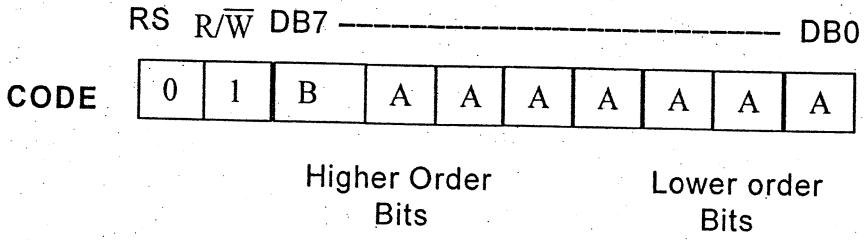
Table – 1 ASCII Code for Graphic or Control

8. SET DD RAM ADDRESS



Sets the DD RAM address as a binary number "AAAAAAA" in the address counter. Data pertaining to the DD RAM is written or read from the host system after this command is executed. When N=0 (1-line display), the value of "AAAAAAA" is "00" through "4F" (Hex). When N=1 (2-line display), the value of "AAAAAAA" is "00" through "27" (Hex) for the first line, and "40" through "67" (Hex) for the second line.

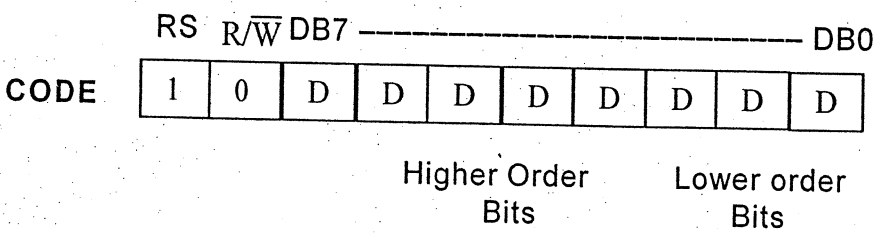
9. READ BUSY FLAG AND ADDRESS



Reads the Busy Flag (BF), which indicates that the system is internally operating on an instruction received previously. When BF = 1, it indicates that internal operation is going on and the next instruction is not accepted until BF returns to "0". Check the BF status before the next write operation.

At the same time, this value of the address counter is expressed in a binary number "AAAAAAA". The address counter is used for both CG and DD RAM address, and its value is determined by the previous instruction. Address contents are the same as items (7) and (8).

10. WRITE DATA TO CG OR DD RAM



Writes binary 8 bit data "DDDDDDDD" to the CG or the DD RAM. Whether the CG or the DD RAM is to be written is determined by the previous command. (CG RAM address setting or DD RAM address setting). After writing, the address is automatically incremented or decremented by one according to the entry mode. Display shift also follows the entry mode.

11. READ DATA FROM CG OR DD RAM

	RS	R/$\overline{W}$	DB7						DB0
CODE	1	1	D	D	D	D	D	D	D	D

Higher Order Bits — Lower order Bits

Reads binary 8-bit data "DDDDDDDD" from the CG or the DD RAM. Whether the CG RAM or the DD RAM is to be read is determined by the previous instruction. Prior to inputting read instructions, either the CG RAM address set instruction or the DD RAM address set instruction must be executed.

After reading data, the address is automatically incremented or decremented by one according to the entry mode. However, display shift is not performed regardless of the entry mode setting.

CHARACTER GENERATOR ROM (CG ROM)

The character generator ROM generates character patterns of 5 x 7 dots or 5 x 10 dots from 8 bit character codes. It can generate 160 5 x 7 dot character patterns and 32 5 x 10 dot character patterns.

BUSY FLAG

When data or commands are written to the Display Module it requires a certain period of time to place the data in the correct location or to execute the desired command. The time required to complete this process and to prepare for a new character or command is the appropriate Execution Time. The character Load Execution Time determines the maximum speed in which characters can be input (the Character Loading Rate). No data should be sent to the module until it has processed a previously entered character. To obtain the maximum character-loading rate, the Busy Flag (Data Bit 7) should be read to determine when the module is ready to accept the next character.

Character loading can also be accomplished while ignoring the status of the Busy Flag at a slight loss in loading rate. In this method, characters may be entered at a rate determined by the user to be less than the Display Module's Character Loading rate.

PROGRAMMING SEQUENCE

From the above discussion it should be clear that the sequence involved in displaying characters in the LCD module requires proper loading of the following commands.

i) Function Set

ii) Clear Display

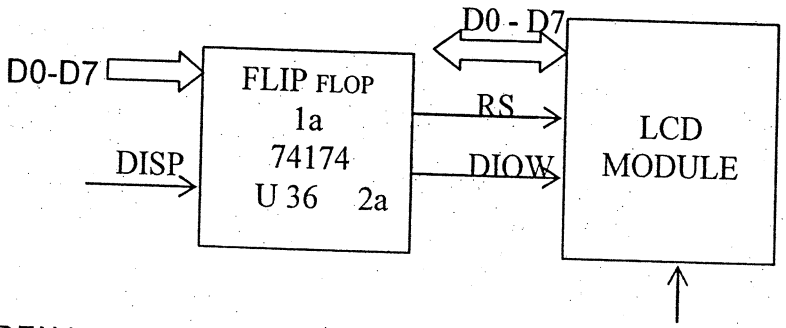
iii) Entry Mode Set

iv) Cursor or Display Shift

v) Set DD RAM Address

vi) Write Data to DD RAM

HARDWARE IMPLEMENTATION



DEN is the Enable pulse to LCD, which selects for I/O address 30H. Read/Write (DIOW) and register select (RS) control lines of LCD are provided by the Flip-flop (U 36) whose I/O address is 34H.

PROGRAM - DISPLAY A STRING IN LCD

OBJECTIVE

To initialise LCD and to display the string "INSTRUMENTATION" in the first row of the display.

THEORY

The LCD module must be sent data to set the desired functions, before sending the characters for display. Before reading or writing data command from/to the module the BUSY flag must be read to determine when the module is ready to accept the next character.

PROGRAM

ADDRESS	OPCODE	MNEMONICS	COMMENTS
5000	CD 4C 50	CALL 504C	BUSY FLAG CHECK
5003	3E 00	MVI A,00	Make RS and R/W low
5005	D3 3D	OUT 3D	
5007	3E 38	MVI A,38	Function set
5009	D3 3C	OUT 3C	
500B	CD 4C 50	CALL 504C	BUSY FLAG CHECK
500E	3E 00	MVI A,00	
5010	D3 3D	OUT 3D	
5012	3E 01	MVI A,01	Clear display
5014	D3 3C	OUT 3C	
5016	CD 4C 50	CALL 504C	BUSY FLAG CHECK
5019	3E 00	MVI A,00	
501B	D3 3D	OUT 3D	
501D	3E 06	MVI A,06	Entry mode set
501F	D3 3C	OUT 3C	
5021	CD 4C 50	CALL 504C	BUSY FLAG CHECK
5024	3E 00	MVI A,00	
5026	D3 3D	OUT 3D	
5028	3E 0C	MVI A,0C	Display control
502A	D3 3C	OUT 3C	
502C	CD 4C 50	CALL 504C	BUSY FLAG CHECK
502F	3E 00	MVI A,00	
5031	D3 3D	OUT 3D	
5033	3E 80	MVI A,80	Set DD ram address
5035	D3 3C	OUT 3C	
5037	21 58 50	LXI H,5058	Lookup table address
503A	0E 0F	MVI C,0F	Number of characters
503C	CD 4C 50	CALL 504C	BUSY FLAG CHECK
503F	3E 01	MVI A,01	Make RS =1 and R/W =0
5041	D3 3D	OUT 3D	
5043	7E	MOV A,M	Write data to DDRAM
5044	D3 3C	OUT 3C	
5046	23	INX H	
5047	0D	DCR C	
5048	C2 3C 50	JNZ 503C	repeat
504B	76	HLT	

SUBROUTINE PROGRAM FOR BUSY FLAG CHECK

ADDRESS	OPCODE	MNEMONICS	COMMENTS
504C	3E 02	MVI A, 02	Make RS=0 and R/W =1
504E	D3 3D	OUT 3D	
5050	DB 3C	IN 3C	Check busy flag
5052	E6 80	ANI 80	
5054	C2 50 50	JNZ 5050	
5057	C9	RET	

Enter the following lookup table (ASCII data for the message INSTRUMENTATION) starting from location 5058 using SU command

Look up table 5058 : 49,4E,53,54,52,55,4D,45,4E,54,41,54,49,4F,4E

RESULT

The LCD module is interfaced with the 8085 microprocessor and the result is verified.

EXCERCISE

1. Display a rolling message "How are You" using display shift option.

2. Blink the message "Hello"

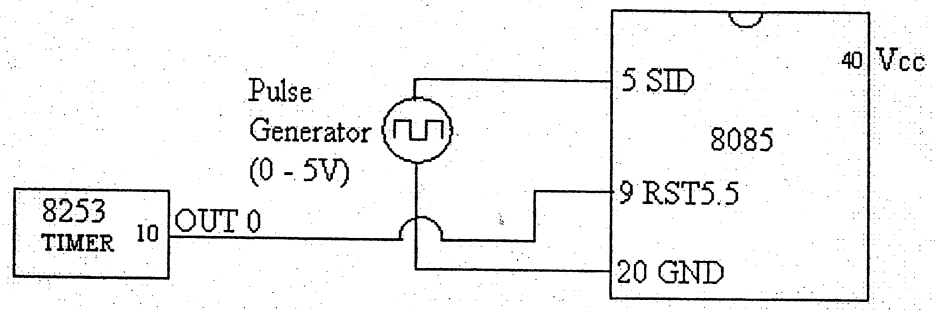


Figure 14.1. Connection Diagram

TABULAR COLUMN

Pulse generator frequency (Hz)	Microprocessor display in Hex	Equivalent reading in Decimal (Hz)

FREQUENCY MEASUREMENT USING SID LINE OF THE 8085 MICROPROCESSOR

AIM

To measure the frequency of a signal using SID line of the 8085 microprocessor.

THEORY OF OPERATION

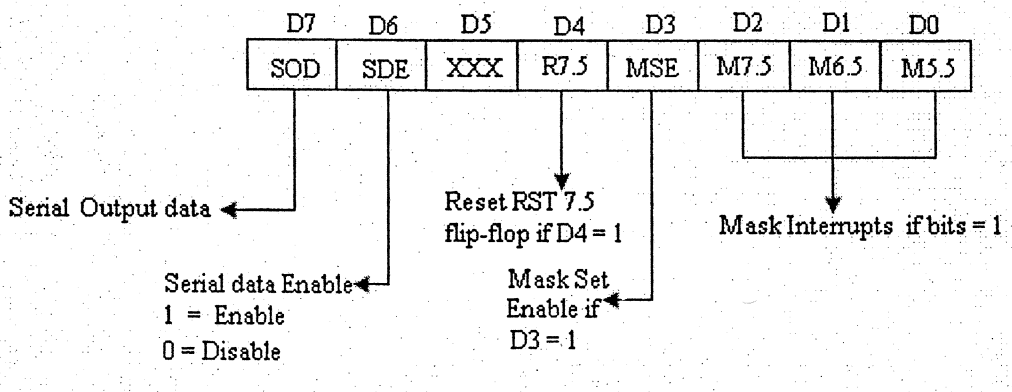
The number of cycles/second is determined to evaluate frequency. The signal output from the pulse generator (0-5V range) is applied to SID line of the 8085 microprocessor. The RIM instruction reads the SID line and stores it in the 7th bit of the accumulator. The program detects the rising edge and the subsequent falling edge of the signal. Once it is detected, it increments the content of HL pair, which is initialized to zero at the beginning. This way the number of pulses are counted till the timer generates an interrupt to the processor. The maximum delay produced by the 8253 (Appendix-C) timer is 1/23 of a second. The HL pair has the number of pulses counted for 1/23 of a second. The interrupt to processor is given through RST 5.5 line as shown in Fig.14.1. In the ISR, the content of HL pair is multiplied by 23 gives the total no of pulses per second (in Hex) and this is displayed using the display routine.

After entering the program, connections in the 8085-microprocessor kit are given as per the connection diagram shown in Fig. 14.1

ADDRESS	OPCODE	MNEMONICS	COMMENTS
4100	3E 30	MVI A, 30	Control word for the timer
4102	D3 0B	OUT 0B	Port address of control word register
4104	3E FF	MVI A, FF	Move the LSB and MSB of the count to counter 0 of timer.
4106	D3 08	OUT 08	Port address of counter 0
4108	3E FF	MVI A, FF	FF is moved to A register
410A	D3 08	OUT 08	Port address of counter 0
410C	21 00 00	LXI H, 00 00	Initialize HL pair to store the no of pulses counted.
410F	FB	EI	Enabling interrupt RST 5.5
4110	3E 1E	MVI A, 1E	To set RST5.5
4112	30	SIM	Set interrupt mask
4113	20	RIM	Identifying the low to high Transition.
4114	E6 80	ANI 80	Content of accumulator is compared with 80h
4116	FE 80	CPI 80	Jump on carry to specified address
4118	DA 13 41	JC 4113	Identify the high to low transition
411B	20	RIM	Jump if zero to specified address
411C	E6 80	ANI 80	Increment the count by one
411E	FE 80	CPI 80	Store the pulse count in 4200
4120	CA 1B 41	JZ 411B	Jump to specified address
4123	23	INX H	
4124	22 00 42	SH LD 4200	
4127	C3 13 41	JMP 4113	

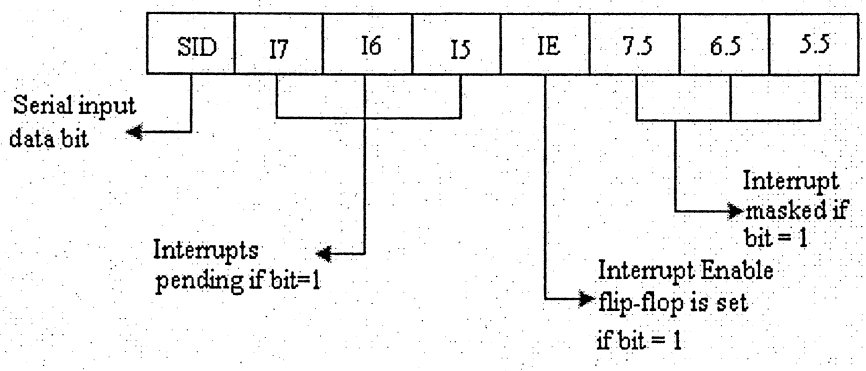
SIM: Set Interrupt Mask

This is a multipurpose instruction and used to implement the 8085 interrupts (RST 7.5, 6.5 and 5.5) and serial data output.



RIM: Read Interrupt Mask

This is a multipurpose instruction used to read the status of interrupts 7.5, 6.5, 5.5 and to read serial data input bit.



Interrupt service routine :

ADDRESS	OPCODE	MNEMONICS	COMMENTS
002C	C3 5E 40	JMP 405E	Jump to the specified address
405E	C3 06 43	JMP 4306	Jump to the specified address
4306	2A 00 42	LHLD 4200	Multiplying the content stored in
4309	EB	XCHG	4200 by 23,to get the number of
430A	21 00 00	LXI H, 0000	pulses counted for one second.
430D	06 17	MVI B, 17	Move immediate eight bit data
430F	19	DAD D	Add register pair to H and L registers
4310	05	DCR B	Decrement the content of B register
4311	C2 OF 43	JNZ 430F	Jump to specified address if [B]'"0
4314	22 00 45	SHLD 4500	Store H and L registers direct
4317	3A 01 45	LDA 4501	Packed to unpacked conversion
431A	E6 F0	ANI F0	AND immediate with accumulator
431C	0F	RRC	Rotate accumulator right
431D	0F	RRC	
431E	0F	RRC	
431F	0F	RRC	
4320	3200 46	STA 4600	Store the content of accumulator to the specified address
4323	3A 01 45	LDA 45 01	Load accumulator direct
4326	E6 0F	ANI 0F	AND immediate with accumulator
4328	32 01 46	STA 4601	Store the content of accumulator to the specified address
432B	3A 00 45	LDA 4500	Load accumulator with the content of specified address
432E	E6 F0	ANI F0	AND immediate with accumulator
4330	0F	RRC	
4331	0F	RRC	Rotate accumulator right
4332	0F	RRC	
4333	0F	RRC	
4334	32 02 46	STA 4602	Store the content of accumulator to the specified address
4337	3A 00 45	LDA 4500	Load accumulator with the content of specified address
433A	E6 0F	ANI 0F	AND immediate with accumulator
433C	32 03 46	STA 4603	Store the content of accumulator to the specified address
433F	21 00 46	LXI H, 4600	Call the display routine to display
4342	3E 03	MVI A, 03	the frequency in Hex
4344	0E 09	MVI C, 09	Segment address
4346	CD 05 00	CALL 0005	Call monitor sub routine
4349	76	HLT	Halt

Control word for the timer in the program is (30) Hex. i.e. counter 0 is selected and the counter is loaded as LSB first and MSB next, counter operated in Mode 0 and as a binary counter.

(Note : For the details of 8253 Timer, Internal Block Diagram and the Control word format refer Data sheets given in Appendix - C)

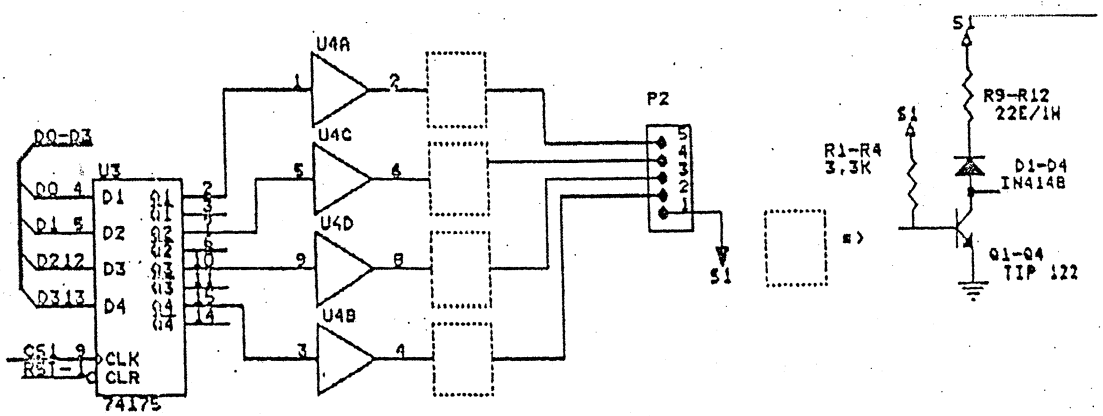
(Note : For the details of Monitor System Calls refer Data sheets given in Appendix - A)

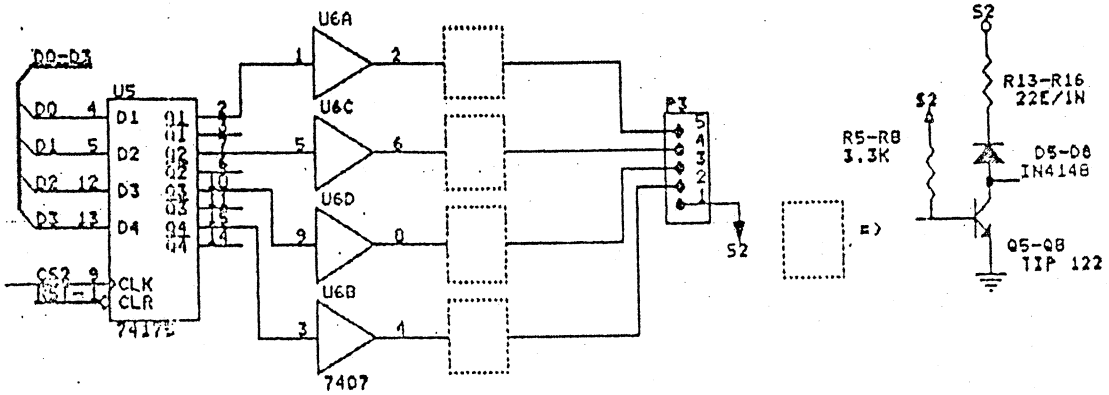
RESULT

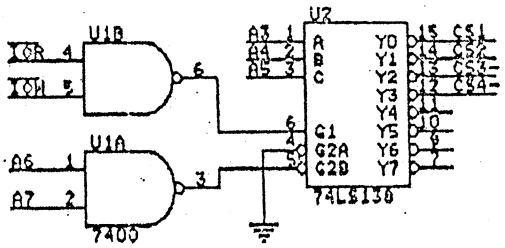
The frequency of the applied pulse signal from the signal generator is measured using the SID line of the Microprocessor 8085.

EXERCISE

1. Display the frequency, which you have calculated using the above program, in decimal.
2. How will you measure the frequency of a sine wave using the SID line of the 8085 Microprocessor.


U4A
U4C
U4D
U4B
P2
D0-D3
U3
D0
D1
D212
D313
D1
D2
D3
D4
CS1
RST
CLK
CLR
74175
S1
R1-R4
3.3K
R9-R12
22E/1W
D1-D4
IN4148
Q1-Q4
TIP 122


U6A
U6C
U6D
U6B
P3
D0-D3
U5
D0
D1
D2
D3
D1
D2
D3
D4
CS2
RST
CLK
CLR
74175
7407
S2
R5-R8
3.3K
R13-R16
22E/1W
D5-D8
IN4148
Q5-Q8
TIP 122


U1B
U1A
U2
IOR
IOW
A6
A7
A3
A4
A5
A
B
C
G1
G2A
G2B
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
7400
74LS138

Exp. No. :
Date :

STEPPER MOTOR INTERFACE

AIM

To interface a Stepper Motor with 8085 microprocessor kit.

THEORY

A motor in which the rotor is able to assume only discrete stationary angular position is a stepper motor. The rotary motion occurs in a stepwise manner from one equilibrium position to the next.

Stepper motor control is a very popular application of microprocessor in control area. They are widely used in (simple position control systems in the open and closed loop mode) a variety of applications such as computer peripherals (printers, diskdriver etc.) and in the areas of process control machine tools, medicine, numerically controlled machines and Robotics.

WAVE SCHEME

The stepper motor windings A1, B1, A2, B2 can be cyclically excited with a DC current to run the motor in the clockwise direction. By reversing the phase sequence as A1, B2, A2, B1, we can obtain anticlockwise stepping.

Table Wave Switching Scheme

Anticlockwise					Clockwise				
Step	A1	A2	B1	B2	Step	A1	A2	B1	B2
1	1	0	0	0	1	1	0	0	0
2	0	0	0	1	2	0	0	1	0
3	0	1	0	0	3	0	1	0	0
4	0	0	1	0	4	0	0	0	1

2-PHASE SCHME

In this scheme, any two adjacent stator windings are energized.

Table 2-Phase Switching Scheme

Anticlockwise					Clockwise				
Step	A1	A2	B1	B2	Step	A1	A2	B1	B2
1	1	0	0	1	1	1	0	1	0
2	0	1	0	1	2	0	1	1	0
3	0	1	1	0	3	0	1	0	1
4	1	0	1	0	4	1	0	0	1

HARDWARE DESCRIPTION

ADDRESS DECODING

The 74138 chip (U2) generates the address decoding logic to generate the device select pulses, CS1 and CS2 for selecting the ICs 74175 (U3 & U5). The 74175 (U3 & U5) latches the data bus to the stepper motor driving circuitry. Address lines A7 & A6 from VXT Bus is NANded and the O/P is connected to 138 enable signal. Similarly IOW & IOR signals are nanded and the NAND gate O/P is connected to Pin 6 of 74138, Pin 4 is grounded.

74175 at U3 is selected with the address

A7	A6	A5	A4	A3	A2	A1	A0	
1	1	0	0	0	X	X	X	= C0 [HEX]

74175 at U4 is selected with the address

A7	A6	A5	A4	A3	A2	A1	A0	
1	1	0	0	1	X	X	X	= C8 [HEX]

74125 is selected with the address

A7	A6	A5	A4	A3	A2	A1	A0	
1	1	0	1	0	0	X	X	= D0 [HEX]

A7	A6	A5	A4	A3	A2	A1	A0	
1	1	0	1	1	X	X	X	= D8 [HEX]

1) Program to run a stepper motor for 1 revolution or 360° or 200 steps

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENTS
4200		0E 32	MVI C, 32	Counter for one revolution
4202	START	21 20 42	LXI H, LOOK UP	Load HL pair with Look up table address
4205		06 04	MVI B, 04	Load the register B with counts for data
4207	REPT	7E	MOV A, M	Load the acc. with content of Look up table address
4208		D3 C0	OUT C0	Out the data to Port A
420A		11 03 03	LXI D, COUNT	Load DE pair with delay counts (rpm)
420D	DELAY	00	NOP	No operation
420E		1B	DCX D	Decrement DE pair
420F		7B	MOV A, E	Load the acc. With E
4210		B2	ORA D	Content of A & D are logically ORed
4211		C2 0D 42	JNZ DELAY	Jump on non-zero to delay
4214		23	INX H	Increment the HL pair
4215		05	DCR B	Decrement the register B
4216		C2 07 42	JNZ REPT	Jump on non-zero to REPT
4219		0D	DCR C	Decrement the register C
421A		C2 02 42	JNZ START	Jump on non-zero to START
421D		76	HLT	
4220	LOOK UP	09 05 06 0A		

1 revolution = 360°
Step Angle = 1.8 °/step
4 steps = 7.2° (since the motor is a 4 step sequence stepper motor)

To run the stepper motor for 'K' revolutions

$$\text{No of steps for one revolution} = \frac{360^\circ}{1.8^\circ} = 200 \text{ steps}$$

$$\text{Number of counts for 1 revolution} = \frac{200}{4} = 50 = 32H$$

Number of counts for 'K' revolution = 32 * K (Hex)

2) Program to run the Stepper Motor in both Forward and Reverse directions for one revolution

ADDRESS	LABEL	OPCODE	MNEMONICS	COMMENTS
4100	START	0E 20	MVI C, 32 H	Counter for one revolution in forward direction
4102	FORWD	21 3F 41	LXI H, FORLOOK	Load the HL pair with lookup table address(forward direction)
4105		CD 21 41	CALL ROTATE	Call ROTATE
4108		0D	DCR C	Decrement C
4109		C2 02 41	JNZ FORWD	Jump on non-zero to FORWD
410C		CD 35 41	CALL STOP	Call STOP
410F		0E 20	MVI C, 32 H	Counter for one revolution in reverse direction
4111	REVES	21 43 41	LXI H, REVLOOK	Load the HL pair with lookdown table address(reverse direction)
4114		CD 21 41	CALL ROTATE	Call ROTATE
4117		0D	DCR C	Decrement C
4118		C2 11 41	JNZ REVES	Jump on non-zero to REVES
411B		CD 35 41	CALL STOP	Call STOP
411E		C3 00 41	JMP START	Jump to START
4121	ROTATE	06 04	MVI B, 04H	Load the register 'B' with number of data in lookup table
4123	REPT	7E	MOV A,M	Move the content of HL to accumulator
4124		D3 C0	OUT 0C0H	Out the data in port A

4126		11 03 03	LXI D, 0303H	Load the' DE' pair with delay counts
4129	LOOP1	1B	DCX D	Decrement 'DE' pair
412A		7B	MOV A,E	Move the content of E to A
412B		B2	ORA D	Contents of A&D are ORed
412C		C2 29 41	JNZ LOOP1	Jump on non-zero to LOOP1
412F		23	INX H	Increment 'HL' pair
4130		05	DCR B	Decrement B
4131		C2 23 41	JNZ REPT	Jump on non-zero to REPT
4134		C9	RET	
4135	STOP	11 FF FF	LXI D, FFFFH	Load the 'DE' pair with delay counts
4138	LOOP2	1B	DCX D	Decrement 'DE' pair
4139		7B	MOV A,E	Move the content of E to A
413A		B2	ORA D	Contents of A&D are Ored
413B		C2 38 41	JNZ LOOP2	Jump on non-zero to LOOP2
413E		C9	RET	
413F	FOR LOOK	09 05 06 0A		
4143	REV LOOK	0A 06 05 09		

RESULT

Thus the stepper motor was interfaced with 8085 microprocessor.

EXERCISE

1. What are the different types of steeper motor?
2. Define the following terms?
   i) Step angle ii) Stepping rateiii) Holding torque of a stepper motor?
3. What is the difference between the stepper motor and servo motor?
4. List the application of stepper motor?

To rotate the motor at specified rpm speed

Let 'Y' be the specified rpm

$$Y \text{ rpm} = \frac{Y * 360^{\circ}}{60} = 6Y \text{ deg/sec}$$

$$6Y \text{ deg} = 1\text{sec}$$

To find the delay 'D' for 1.8° rotation

$$D = \frac{1.8}{6Y} = \frac{0.3}{Y} \text{ sec}$$

Let 'X' be the count loaded in DE pair for a delay of 'D' sec.

Mnemonics	**Number of T states**	**Number of times occurred**	**Total T states**
LXI D, X	10	1	10
NOP	4	X	4X
DCX D	6	X	6X
MOV A, E	4	X	4X
ORA D	4	X	4X
JNZ	10	X-1	10(X-1)
NO JUMP	7	1	7
			(28X+7) T

Equating the T states to delay

$(28X+7)\ T = D$

$$(28X + 7) = \frac{D}{T} = D * F \qquad \text{Where } F = \frac{1}{T} = 1.5 \text{ Mhz}$$

$$X = \frac{(D * 1.5x10^{6}) - 7}{28}$$

Thus count can be calculated for a given delay of 'D' sec.

APPENDIX - A

MONITOR SYSTEM CALLS, DATA FORMAT FOR SEVEN SEGMENT DISPLAY

OVERVIEW

This Appendix gives the importance of monitor system calls, explanation of some Function Calls and the Data format for Seven Segment Display

MONITOR SYSTEM CALLS :

The user can write their own program easily and very efficiently, using the Monitor Program resourses. These resources are essentially SYSTEM SUBROUTINES accessed through a PRINCIPAL ENTRY POINT. Instead of providing as an Interrupt function, it has been implemented as a CALL function to the monitor.

Before making s SYSTEM CALL to the monitor, initialise the Function number, Information number and the pointer, if necessary. The mentioned task under that Function number would be done after the CALL is made and system returns back to the main program.

The principal entry point for this CALL function is 0005. So, to call a function, set the specified registers so that the function you are calling knows what to do. Now CALL 0005 and the desired task is done. The use of these CALLs makes your programs small and efficient.

Some of the Function Calls are explained below.

DISPLAY DATA (Function 03)

This function displays the data, which are in the memory addressed by HL register pair. Using C register, data can be displayed in any area of the 7 segment Display [Data Display Area, Address Display Area and Status Display Area].

Input	:	A = 03
		C = 00 to OB
		HL = Starting address of the data in the memory
		CALL 05
Output	:	None
Result	:	Displays data in the display
Registers affected	:	A , BC , DE , HL

a) When C=00, the data from the memory pointed by the HL register will be displayed in the leftmost digit of the display. Similarly, when C=01 or 02 or 03 upto 07, data will be displayed to the particular digit.

b) When C = 08, 2 digit data will be displayed in the data field, two digits Will be taken from the memory pointed by HL and HL+1.

c) When C = 09, 4 digit data will be displayed in the address field.

d) When C = 0A, 2 digit data will be displayed in the status field.

e) When C = 0B, 8 digit data will be displayed in all the eight digits.

The following Alphanumeric can be displayed in the 7 segment display. The subroutine takes care of all conversations for the 7 segment display. The user has to load the corresponding data in the memory and is not requered to do any conversation.

BLANK DISPLAY (Function 01)

This function blanks the 7 Segment display

Input	:	A = 01 C = 00, 01, 02, 03 CALL 05
Output	:	None
Result	:	Blanks the address field of Display
Registers affected	:	A, B, C, HL, DE

a) If c = 00, the Address field display is blanked

b) If c=01, the data field display is blanked

c) If c=02, the Status filed display is blanked

d) If c=03, all the digits of display are blanked

DATA FORMAT FOR THE 7 SEGMENT DISPLAY

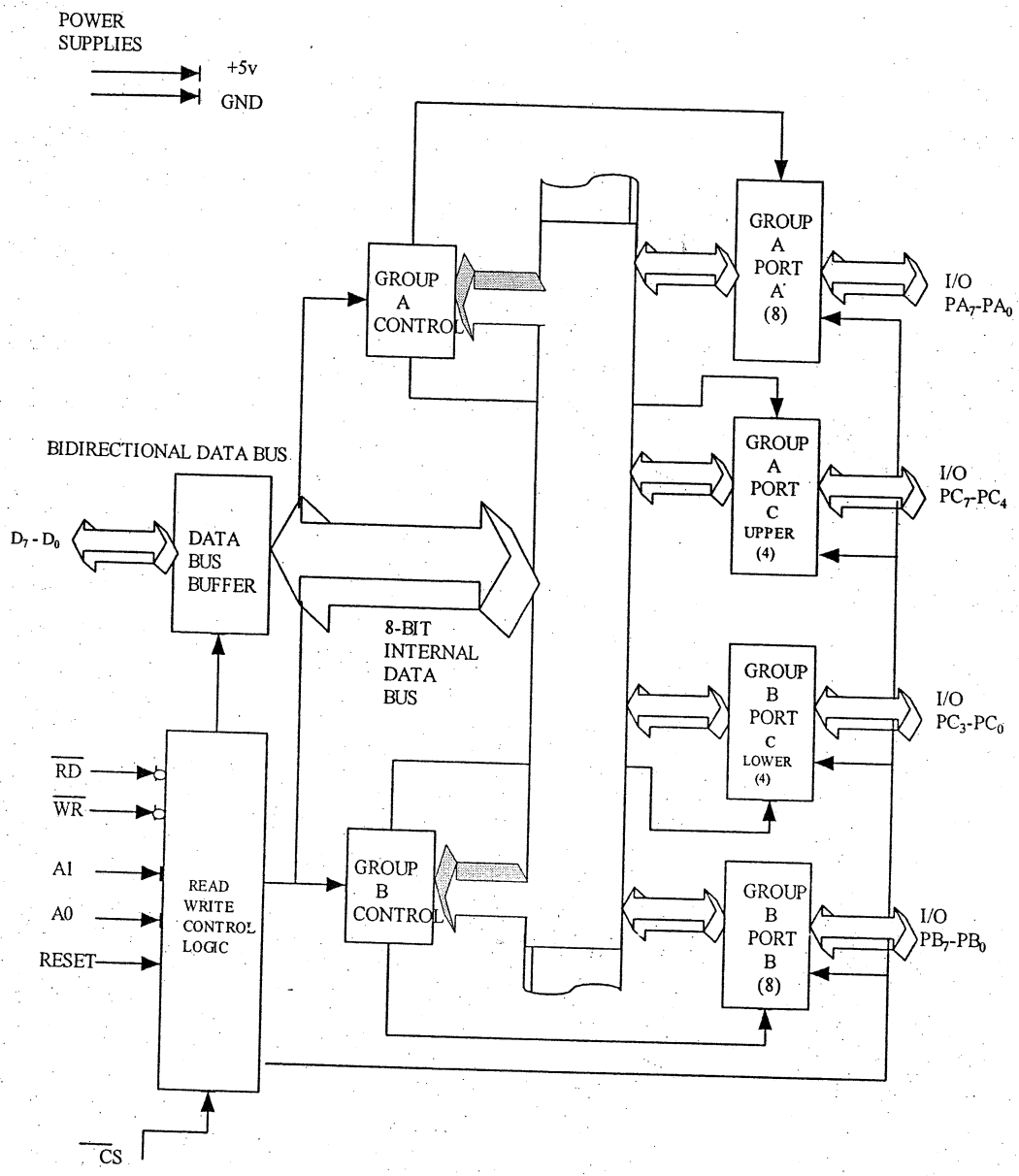
ALPHANUMERIC TO BE DISPLAYED CORRESPONDING DATA TO BE LOADED IN MEMORY	
0	00
1	01
2	02
3	03
4	04
5	05
6	06
7	07
8	08
9	09
A	0A
B	0B
C	0C
D	0D
E	0E
F	0F
BLANK	10
DOT	11
HYPHEN (-)	12
G	13
H	14
L	15
t	16
N	17
n	18
o	19
P	1A
R	1B
r	1C
U	1D
Y	1E
Mu	1F
u	20
1	21
x	26

APPENDIX - B

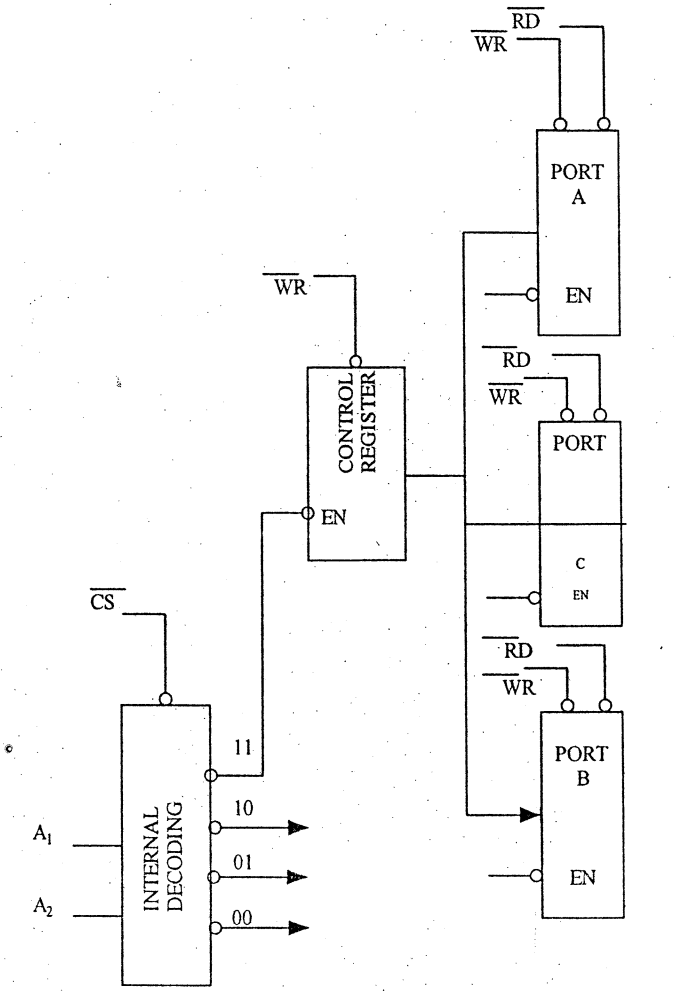
8255 PROGRAMMABLE PERIPHERAL INTERFACE (PPI)

OVERVIEW

This Appendix gives the details of 8255 Programmable Peripheral Interface Block Diagram, PIN configuration and Control Word Format for I/O Mode and BSR Mode

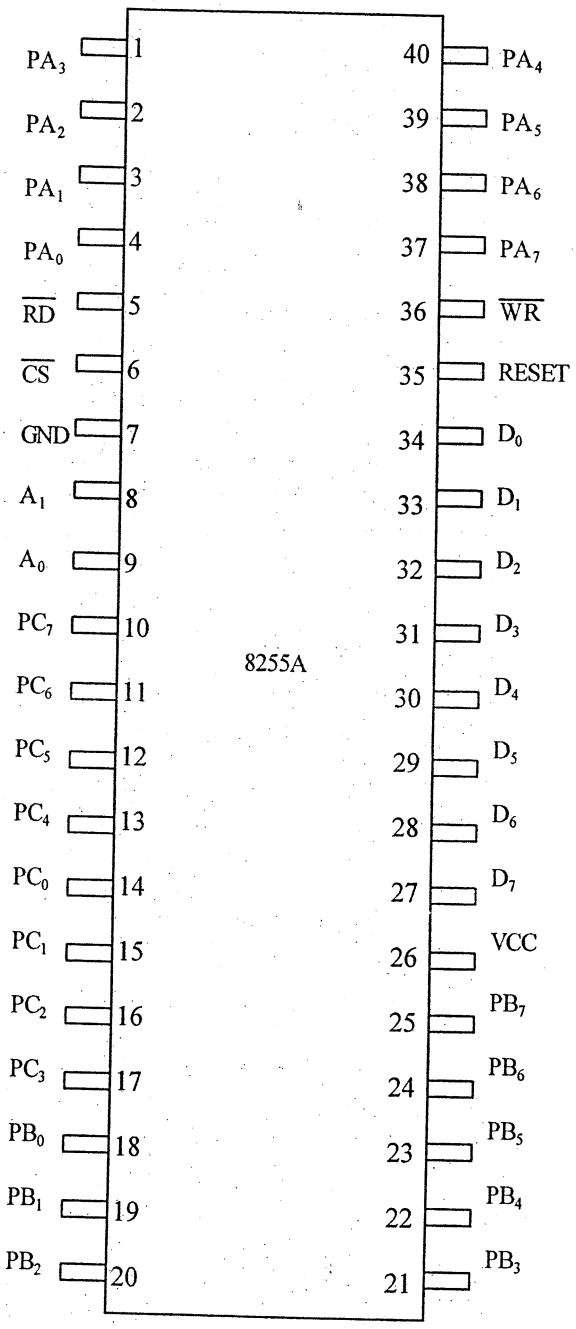


8255A BLOCK DIAGRAM

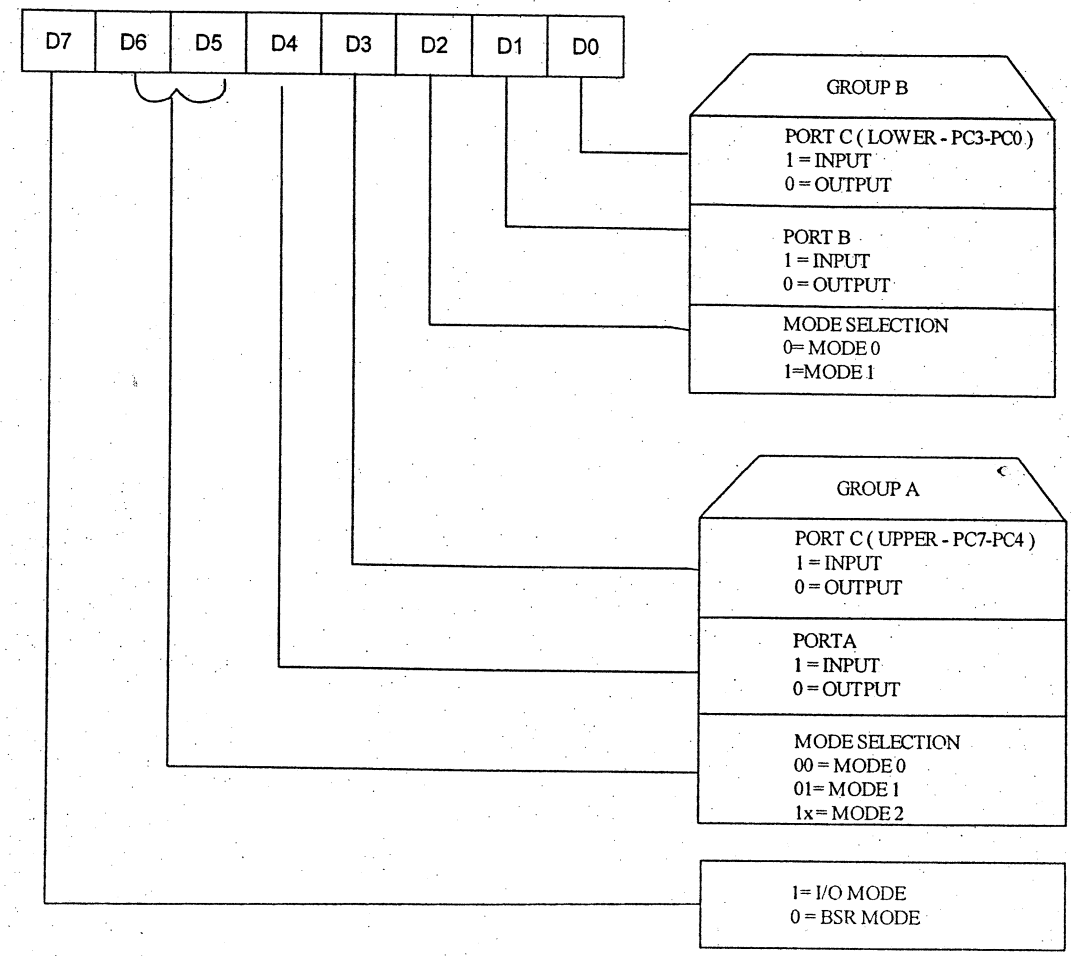


(b)

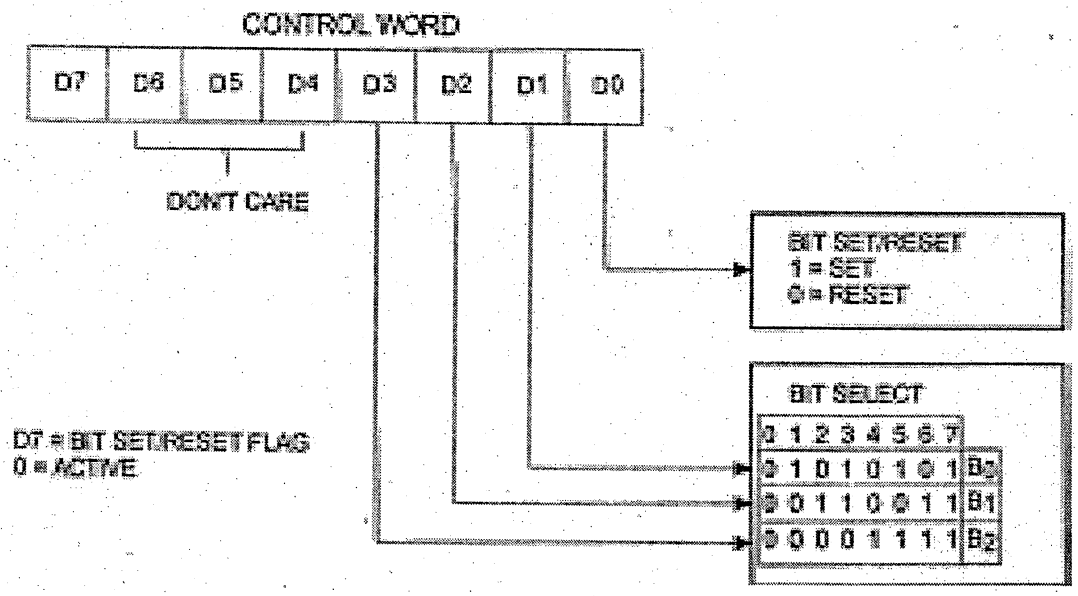
D7-D0	data bus (bi dirctional)
RESET	reset input
$\overline{cs}$	Chip select
$\overline{RD}$	read input
$\overline{WR}$	write input
A_0,A_1	port address
PA7-PA0	port A(Bit)
PB7-PB0	port B (bit)
PC7-PC0	port C (bit)
Vcc	+5Volts
GND	0 Volts



8255 PPI - PIN Configuration



8255A Control Word Format for I/O Mode



8255A Control Word Format for BSR Mode

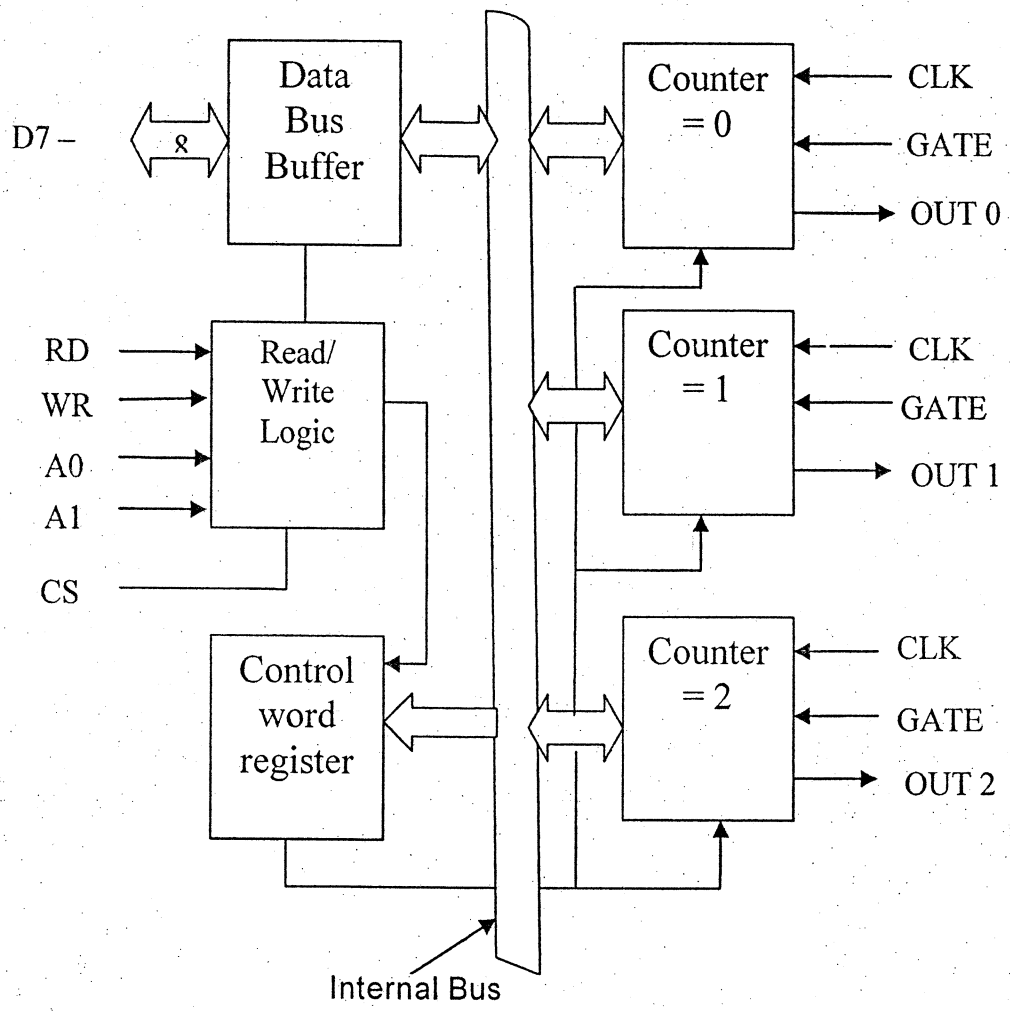
APPENDIX - C

PROGRAMMABLE INTERVAL TIMER 8253 (8254)

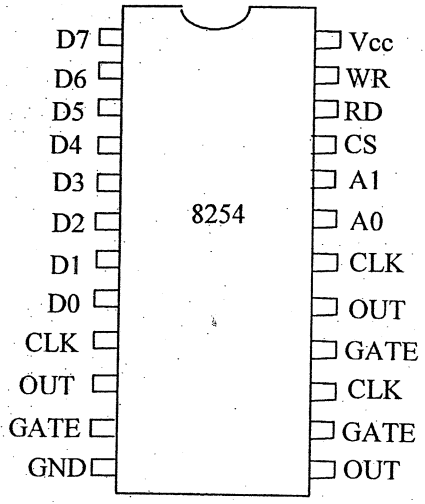
OVERVIEW

This Appendix gives the details of the Functional Block Diagram of 8253 Programmable Interval Timer, PIN out details and the Control Word Format and Mode definitions.

8254 Block diagram



PIN Configuration



PIN Names

D7 – D0	Data Bus (8 bit)
CLK N	Counter Clock inputs
GATE N	Counter Gate Inputs
OUT N	Counter outputs
RD	Read Counter
WR	Write Command or Data
CS	Chip select
A0 – A7	Counter select
Vcc	+5 volts
GND	ground

8253 Control Word Format

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC-Select Counter :

SC1 SC0

0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Read-Back Command

RW-Read/Write:

RW1RW0

0	0	Counter Latch Command
0	1	Read/Write LSB only
1	0	Read/Write MSB only
1	1	Read/Write LSB first, then MSB

M-Mode:

M2 M1M0

0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

BCD:

0	Binary Counter 16-bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

Note : Don't care Bits(x) should be 0 to Ensure compatibitlity with future

APPENDIX - D

SUMMERY OF 8085 INSTRUCTION AND MACHINE CODES

OVERVIEW

This Appendix lists the Instructions (alphabetical order) and Machine Codes of 8085 Microprocesscr.

OPCODE IN HEX	MNEMONIC	OPCODE IN HEX	MNEMONIC	OPCODE IN HEX	MNEMONIC
CE	ACI d8	E4	CPO addr16	0A	LDAX B
8F	ADC A	CC	CZ addr16	1A	LDAX D
88	ADC B	27	DAA	2A	LHLD addr16
89	ADC C	09	DAD B	01	LXI B,addr16
8A	ADC D	19	DAD D	11	LXI D,addr16
8B	ADC E	29	DAD H	21	LXI H,addr16
8C	ADC H	39	DAD SP	31	LXI SP,addr16
8D	ADC L	3D	DCR A	7F	MOV A,A
8E	ADC M	05	DCR B	78	MOV A,B
87	ADD A	0D	DCR C	79	MOV A,C
80	ADD B	15	DCR D	7A	MOV A,D
81	ADD C	1D	DCR E	7B	MOV A,E
82	ADD D	25	DCR H	7C	MOV A,H
83	ADD E	2D	DCR L	7D	MOV A,L
84	ADD H	35	DCR M	7E	MOV A,M
85	ADD L	0B	DCX B	47	MOV B,A
86	ADD M	1B	DCX D	40	MOV B,B
C6	ADI d8	2B	DCX H	41	MOV B,C
A7	ANA A	3B	DCX SP	42	MOV B,D
A0	ANA B	F3	DI	43	MOV B,E
A1	ANA C	FB	EI	44	MOV B,H
A2	ANA D	76	HLT	45	MOV B,L
A3	ANA E	DB	IN addr8	46	MOV B,M
A4	ANA H	3C	INR A	4F	MOV C,A
A5	ANA L	04	INR B	48	MOV C,B
A6	ANA M	0C	INR C	49	MOV C,C
E6	ANI d8	14	INR D	4A	MOV C,D
CD	CALL addr16	1C	INR E	4B	MOV C,E
DC	CC addr16	24	INR H	4C	MOV C,H
FC	CM addr16	2C	INR L	4D	MOV C,L
2F	CMA	34	INR M	4E	MOV C,M
3F	CMC	03	INX B	57	MOV D,A
BF	CMP A	13	INX D	50	MOV D,B
B8	CMP B	23	INX H	51	MOV D,C
B9	CMP C	33	INX SP	52	MOV D,D
BA	CMP D	DA	JC addr16	53	MOV D,E
BB	CMP E	FA	JM addr16	54	MOV D,H
BC	CMP H	C3	JMP addr16	55	MOV D,L
BD	CMP L	D2	JNC addr16	56	MOV D,M
BE	CMP M	C2	JNZ addr16	5F	MOV E,A
D4	CNC addr16	F2	JP addr16	58	MOV E,B
C4	CNZ addr16	EA	JPE addr16	59	MOV E,C
F4	CP addr16	E2	JPO addr16	5A	MOV E,D
EC	CPE addr16	CA	JZ addr16	5B	MOV E,E
FE	CPI d8	3A	LDA d16	5C	MOV E,H

OPCODE IN HEX	MNEMONIC		OPCODE IN HEX	MNEMONIC		OPCODE IN HEX	MNEMONIC	
5D	MOV	E,L	C1	POP	B	97	SUB	A
5E	MOV	E,M	D1	POP	D	90	SUB	B
67	MOV	H,A	E1	POP	H	91	SUB	C
60	MOV	H,B	F1	POP	PSW	92	SUB	D
61	MOV	H,C	C5	PUSH	B	93	SUB	E
62	MOV	H,D	D5	PUSH	D	94	SUB	H
63	MOV	H,E	E5	PUSH	H	95	SUB	L
64	MOV	H,H	F5	PUSH	PSW	96	SUB	M
65	MOV	H,L	17	RAL		D6	SUI	d8
66	MOV	H,M	1F	RAR		EB	XCHG	
6F	MOV	L,A	D8	RC		AF	XRA	A
68	MOV	L,B	C9	RET		A8	XRA	B
69	MOV	L,C	20	RIM		A9	XRA	C
6A	MOV	L,D	07	RLC		AA	XRA	D
6B	MOV	L,E	F8	RM		AB	XRA	E
6C	MOV	L,H	D0	RNC		AC	XRA	H
6D	MOV	L,L	C0	RNZ		AD	XRA	L
6E	MOV	L,M	F0	RP		AE	XRA	M
77	MOV	M,A	E8	RPE		EE	XRI	d8
70	MOV	M,B	E0	RPO		E3	XTHL	
71	MOV	M,C	0F	RRC				
72	MOV	M,D	C7	RST	0			
73	MOV	M,E	CF	RST	1			
74	MOV	M,H	D7	RST	2			
75	MOV	M,L	DF	RST	3			
3E	MVI	A, d8	E7	RST	4			
06	MVI	B, d8	EF	RST	5			
0E	MVI	C, d8	F7	RST	6			
16	MVI	D, d8	FF	RST	7			
1E	MVI	E, d8	C8	RZ				
26	MVI	H, d8	98	SBB	B			
2E	MVI	L, d8	99	SBB	C			
36	MVI	M, d8	9A	SBB	D			
00	NOP		9B	SBB	E			
B7	ORA	A	9C	SBB	H			
B0	ORA	B	9D	SBB	L			
B1	ORA	C	9E	SBB	M			
B2	ORA	D	DE	SBI	d8			
B3	ORA	E	22	SHLD	addr16			
B4	ORA	H	30	SIM				
B5	ORA	L	F9	SPHL				
B6	ORA	M	32	STA	addr16			
F6	ORI	d8	02	STAX	B			
D3	OUT	addr8	12	STAX	D			
E9	PCHL		37	STC				

d8	→	8-bit data	addr16	→	16-bit address
d16	→	16-bit data	M	→	Memory
addr8	→	8-bit address	PSW	→	Program Status Word

5

MONITOR SYSTEM CALLS

5.1 INTRODUCTION

The monitor program is written in a **MODULAR APPROACH.** The users can now write their own program easily and very efficiently, using the monitor program resources.

These resources are essentially **SYSTEM SUBROUTINES** accessed through a **PRINCIPAL ENTRY POINT.** Instead of providing as an Interrupt function, it has been implemented as a CALL function to the monitor.

Before making a SYSTEM CALL to the monitor, initialise the Function number, Information number and the pointer, if necessary. The mentioned task under that Function number would be done after the CALL is made and system returns back to the main program.

The principal entry point for this CALL function is 0005. So, to call a function, set the specified registers so that the function you are calling knows what to do. Now CALL 0005 and the desired task is done.

The use of these CALLs makes your programs small and efficient. The following is a summary of the FUNCTION CALLS.

5.2 SYSTEM CALLS DESCRIPTION

5.2.1 RESET (Function 00)

This function resets the Microprocessor kit

Input	:	A = 00 CALL 05
Output	:	None
Registers affected	:	All Registers

5.2.2 BLANK DISPLAY (Function 01)

This function blanks the 7 segment display

```
Input          :  A = 01
                  C = 00,01,02,03
                  CALL 05
Output         :  None
Result         :  Blanks the Address field of Display
Registers      :  A,B,C,HL,DE
affected
```

a) If C=00, the Address field display is blanked
b) If C=01, the data field display is blanked
c) If C=02, the Status field display is blanked
d) If C=03, all the digits of display are blanked

5.2.3 DISPLAY DOT (Function 02)

This function displays the DOT in any of the digit depending on the C register content.

```
Input          :  A = 02
                  C = 00, 01, 02, 03
                  CALL 05
Output         :  None
Result         :  Display dot
Registers      :  A,B,C,HL,DE
affected
```

a) When C=00, dot will be displayed in the right most digit of the data field.

b) When C=01, dot will be displayed in the right most digit of the address field.

c) When C=02, dot will be displayed in the right most of the upper two digits of the address field.

d) When C=03, dot will be displayed in the right most digit of the status field.

5.2.4 DISPLAY DATA (Function 03)

This function display the data, which are in the memory addressed by HL Register. Using C register, data can be displayed in any area of the 7 segment Display (Data Display Area, Address Display Area and Status Display Area).

Input	:	A = 03 C = 00 to 0B HL = Starting address of the data in the memory CALL 05
Output	:	None
Result	:	Displays data in the display
Registers affected	:	A,BC,DE,HL

a) When C=00, the data from the memory pointed by the HL register will be displayed in the leftmost digit of the display. Similarly, when C=01 or 02 or 03 upto 07, data will be displayed to the particular digit.

b) When C=08, 2 digit data will be displayed in the data field, two digits will be taken from memory pointed by HL and HL+1.

c) When C=09, 4 digit data will be displayed in the address field.

d) When C=0A, 2 digit data will be displayed in the status field.

e) When C=0B, 8 digit data will be displayed in all the eight digits.

The following Alphanumeric can be displayed in the 7 segment display of the Micro-85 EB .

The subroutine takes care of all conversions for the 7 segment display. The user has to load the corresponding data in the memory and is not required to do any conversion.

DATA FORMAT FOR 7 SEG DISPLAY

ALPHANUMERIC TO BE DISPLAYED	CORRESPONDING DATA TO BE LOADED IN MEMORY
0	00
1	01
2	02
3	03
4	04
5	05
6	06
7	07
8	08
9	09
A	0A
B	0B
C	0C
D	0D
E	0E
F	0F
BLANK	10
D0T	11
HYPHEN (-)	12
G	13
H	14
L	15
t	16
N	17
n	18
o	19
P	1A
R	1B
r	1C
U	1D
Y	1E
Mu	1F
u	20
l	21
x	26

5.2.5 DISPLAY THE MESSAGE "Err" (Function 04)

This function will display "Err" in the display and after a delay it will jump to command prompt mode.

```
Input          :  A  = 04
                  CALL 05
Output         :  None
Result         :  Displays "Err" in display
Registers      :  All registers
affected
```

5.2.6 READ FROM KEYBOARD (Function 05)

This function will read a key and return the key code in the A register.

```
Input          :  A  = 05
                  CALL 05
Output         :  A  = code of the key
Result         :  Returns key code in the A register
Registers      :  A
affected
```

5.2.7 HEX TO ASCII & ASCII TO HEX CONVERSION (Function 06)

a) Hex to ASCII.

When Reg 'C' is 00 this function converts a hex value to its equivalent ASCII value. The 8 bit hex data will be converted to 2 ASCII values.

```
Input          :  A = 06
                  C = 00
                  L = hex value
                  CALL 05
Output         :  DE register pair will have the ASCII
                  values D-Higher byte, E-Lower bytes
Result         :  DE will have the ASCII values
Registers      :  A,DE
affected
```

b) ASCII to Hex.

When Reg 'C' is 01 this function converts two ASCII values to its equivalent 8 bit Hex value.

Inputs	:	A = 06 C = 01 H = Higher byte ASCII value L = Lower byte ASCII value CALL 05
Output	:	A contains the hex value
Result	:	A will be returned with the 8 bit hex value
Registers affected	:	A,DE

5.2.8 CALCULATE LENGTH OF A BLOCK

This function calculates the length of a block of memory.

Input	:	A = 07 HL = Starting address of the memory BC = End address of block CALL 05
Output	:	DE contains the Block Size
Result	:	The Block size is returned in DE register in hex. If HL>BC, then a calculation error is displayed.
Registers affected	:	A,BC,DE

5.2.9 GENERATE TONE (Function 08) (Optional)

This function generates a tone for driving a speaker.

Input	:	A = 08 C = 00 or 01 CALL 05
Output	:	None
Result	:	Generates a constant frequency for the speaker.
Registers affected	:	A,BC,HL

a) When C=00, it generates 1 KHz
b) When C=01, it generates 2 KHz

5.2.10 ROUTINE FOR DELAY (Function 09)

Input	:	A = 09 HL = Delay value CALL 05
Output	:	None
Result	:	The delay time is equal to the value of HL * 6.9 Micro Seconds.
Registers affected	:	A,H,L.