UPES

THE NATION BUILDERS UNIVERSITY

UNIVERSITY OF PETROLEUM & ENERGY STUDIES

MICROPROCESSOR LAB MANUAL

FOR

ENGINEERING STUDENTS

B.TECH (SEMESTER –II, VI)

DEPARTMENT: Electronics Instrumentation & Control (EIC)

Dr. S.Choudhary
(HOD Electronics & Electrical Depts.)

Dr. Rajesh Garg
(HEAD Labs)

“CONTENTS”

8085 MICROPROCESSOR & PERIPHERALS LAB

INDEX

After conducting this lab students will be able to understand Assembly language programming on the 8085 Microprocessor Trainer Kit

SECTION- A

Note: Perform any *ten* experiments

SECTION-B

Note: Perform any *five* experiments

EXPERIMENT NO. 1

AIM:

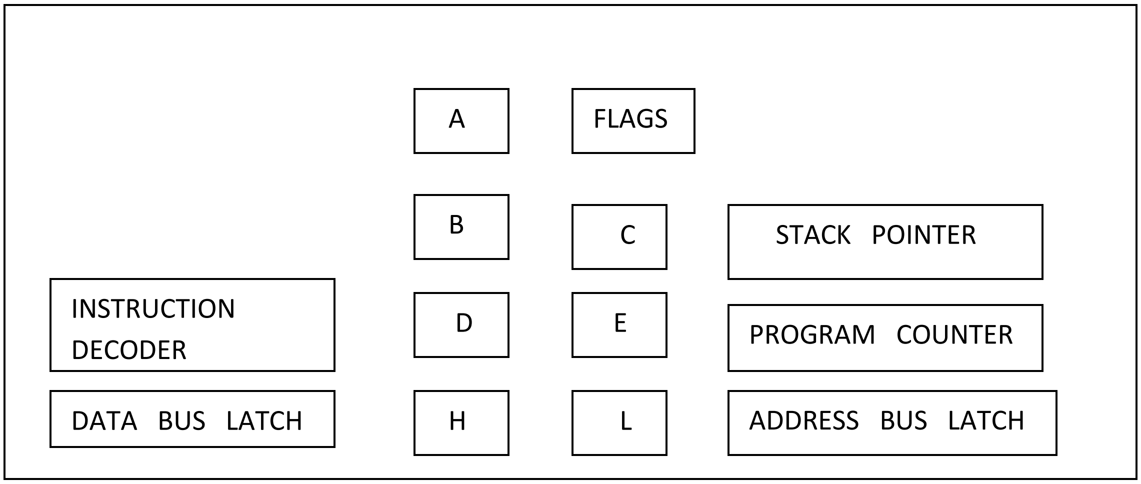
To explain the architecture of the 8085 Micro-Processor.

INTRODUCTION:

INTEL 8085 is a general purpose 8-bit Microprocessor capable of addressing 64KB of memory. It is enhanced version of its predecessor, the 8080 and its instruction set is upward compatible with that of the 8080A.

The microprocessor has a set of instructions designed internally, to manipulate data and communicate with peripherals. It can be programmed to perform functions on given data by selecting necessary instructions from its set.

The internal architecture of the 8085A determines how and what operations can be performed with the data. T o perform any operation, the microprocessor requires registers, an Arithmetic Logic Unit (ALU) and control Logic and internal buses. Certain registers of the processor can be written into and hence are available for manipulation or processing of input data. Use of certain write instructions makes this feasible. To know more about the features of the processor let us take a look into the internal architecture of the 8085A.



ACCUMULATOR:

The accumulator is the primary source and destination for one operand and two operand instructions. For example, all data transfers between the CPU and I/P devices are performed through the accumulator. In addition, many memory reference instructions move data between the accumulator and memory than between any other register and memory. All Arithmetic and Boolean instructions take one of the operands from the accumulator and return the result to the accumulator. So the accumulator should be loaded before any arithmetic or Boolean operation.

REGISTERS:

Apart from the accumulator we have got other registers such as B, C, D, E, H, and L . These are called secondary registers. Data stored in any of these six registers may be accessed with equal ease. Such data can be moved to any other registers or can be used as the second operand in two operand instructions. These six registers can be used to hold 8-bit data when used individually or 16-bit data when used in pair as BC as DE or HL.

Registers H and L comprise the primary data pointer for 8085A. Normally, these two registers will be used to hold the 16-bit memory address of data being accessed. It is usually called memory pointer. We can transfer data between any specified register and memory location addressed by H and L. It is good to address data memory via registers H and L ,whenever possible because it can make your program more efficient and easy to relocate.

FLAGS:

The ALU consists of five flip-flops that are set or reset according to data conditions in the Accumulator and other registers. The Microprocessor uses them to test for data conditions. The five flip-flops , referred to as flags ,are the carry flag (C), the Zero flag (Z), the sign flags (S), the parity flag (P), and the Auxiliary Carry flag (AC).

PROGRAM COUNTER (PC):

This register deals with sequencing the execution of instructions. The function of the PC is to point to the memory location from which the next byte is to be fetched for execution.

STACK POINTER (SP):

The stack pointer Register is also 16-bit and is used as a memory pointer. It point s to a memory pointer. It points to a memory location in RAM memory, called the STACK. T he beginning of the stack is defined by the user.

SHORT QUESTIONS:-

1. What is Microprocessor?
2. What is the technology used in Microprocessor?

3. What is the function of Flag , Register, Program counter and Stack pointer?
4. What are the meant by Low level and High level language?
5. What is machine level programming?
6. What jobs ALU of 8085 can perform?
7. How many hardware interrupts 8085 supports. ?
8. How many I/P ports can 8085 access?
9. Describe the accumulator register of 8085?
10. What is the function of RESET IN and RESET OUT function?

EXPERIMENT NO. 2

AIM:

To explain the classification of the instructions of 8085A & to define the various addressing mode in the 8085A.

8085 INSTRUCTION CLASSIFICATION:

The function of a microprocessor system is implemented by a sequence of data transfers between memory, processor and I/O devices and data transformations that occur in the registers within the microprocessor, manipulating a register under program control , addressing it and using it for data transfer and transformation requires a set of binary codes which comprise the INSTRUCTION SET OF A MICRO-PROCESSOR.

The 8085A instructions can be classified into the following five functional categories.

i) Data transfer (copy) operations,
ii) Arithmetic operations ,
iii) Logical operations
iv) Branching operations and
v) Machine-control operations.

Since data and instructions may reside anywhere in internal registers, external register or memory, locating them requires a particular addressing. The instructions in these five functional groups can be categorized according to their method of addressing the hardware registers and memory. This method is called the ADDRESSING MODE and six modes are available with 8085A which are explained in detail here,

i) Implied addressing
ii) Register addressing
iii) Immediate addressing
iv) Direct addressing
v) Register indirect addressing
vi) Combined addressing.

IMPLIED ADDRESSING:

The instructions using this mode have no explicit operands. Examples include

STC (Set Carry Flag)

DAA (Decimal Adjust Accumulator)

REGISTER ADDRESSING:

This is mode specifies the register or register pair that contains data. Both the source and the destination operand are registers.

For example,

```
MOV       B, C
```

Moves the contents of register C to register B.

IMMEDIATE ADDRESSING:

For an 8-bit data, this mode uses 2 bytes, with the first byte as the OP code, followed by 1 byte of data. On the other hand, for 16-bit data, this instruction contains 3 bytes, with the first byte as the OP code followed by 2 bytes of data. For example,

```
MVI     B,  05
```

Loads register B with the value 5 an

```
LXI     H,  2050
```

Loads H with 20 and L with 50.

DIRECT ADDRESSING:

Instructions using this mode specify the effective address as a part of the instruction. These instructions contain 3 bytes, with the first byte as the OP code followed by 2 bytes of address of data (the low-order byte of the address in byte 3) . Consider

```
LDA     2035
```

This instruction loads accumulator with the contents of memory location 2035. This mode is also called the absolute mode.

REGISTER INDIRECT ADDRESSING:

This mode contains a register pair which stores the address of data (the higher-order byte of the address in the first register of the pair, and low-order byte in the second). As an example,

LDAX B

Loads the accumulator with the contents of a memory location addressed by B and C register pair.

COMBINED ADDRESSING MODES:

Some instructions use a combination of addressing modes. A CALL instruction, for example, combines direct addressing and register indirect addressing. The direct address in a CALL instruction specifies the address of the desired subroutines; the register indirect address is that of stack pointer. The CALL instruction pushes the current contents of the program counter into the memory location specified by the stack pointer. The address that follows the CALL instruction is copied to pc and hence execution starts at the address of the subroutine.

SHORT QUESTIONS:-

1. What is an Instructions?
2. What is meant by Instruction Set?
3. In how many categories the instruction of 8085 be classified?
4. What is meant by 'addressing mode' different mode? Mention the different addressing mode?
5. Mention the Interrupts pin of 8085?
6. Explain the maskable and Non-maskable Interrupt?
7. What is meant by priority of interrupts?
8. What are the different types of data transfer operations possible?
9. Mention the different type of operations with asthmatic, logical, branch & machine control operation?
10. What are the different instruction word sizes in 8085?

EXPERIMENT NO. 03

PART (A):-

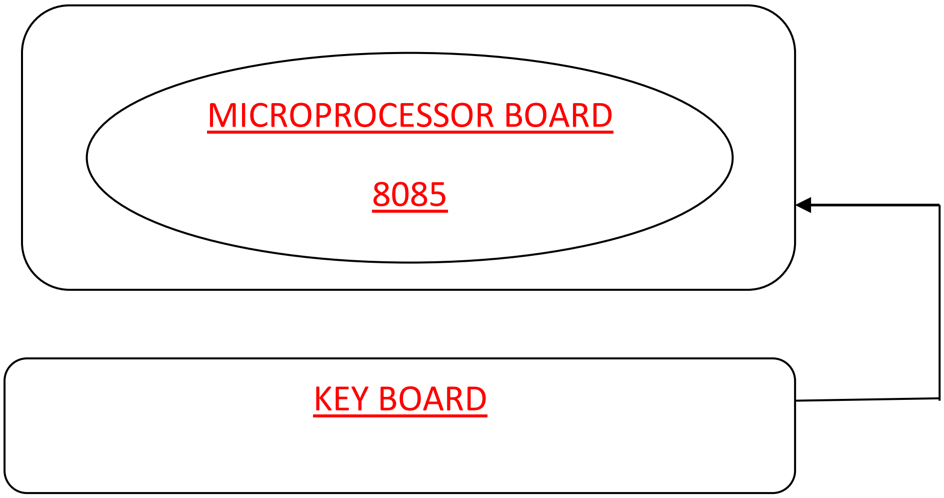
AIM:

Addition of two eight bit numbers in memory and result also in memory.

.

APPRATUS:

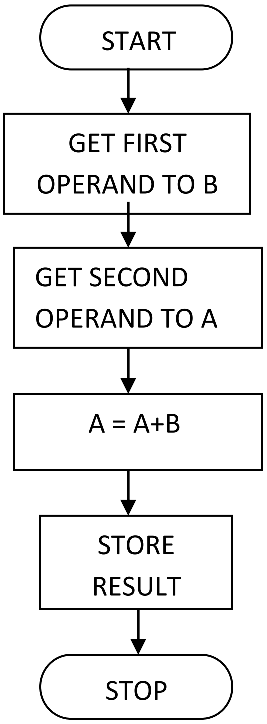
Microprocessor Kit 8085 , Key Board, Op-Code Sheet.



THEORY:

The first data is brought to Accumulator A and the second one in any one of the other registers, say B. The addition is done using ADD. The result is then stored at 4252. The ADD instruction affects flags depending on result.

FLOW CHART



PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected.
3. Make sure you are properly grounded.
4. Don't touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

- **i)** Key in the Opcodes from the address specified.
- **ii)** Enter data at 4250 and 4251 as specified in the example.
- **iii)** Execute the program and check for the result at 4252.
- **iv)** Change data at 4250 and 4251 and execute each time and check for result.

PROGRAM:

```
LDA     4250    ;    (4250)  = >  (A)  =  23

MOV   B , A     ;     (A)     = >   (B)
```

```
LDA   4251     ;   ( 4251)  = >  (A)   =  35

ADD    B       ;    (A)   +  (B)   = >  (A)

STA     4252    ;    (A)     =  >  (4252)  =  58

 HLT
```

OBJECT CODES:

Memory Address	Opcodes	Mnemonics
4200	3A	LDA 4250
4201	50	
4202	42	
4203	47	MOV B , A
4204	3A	LDA 4251
4205	51	
4206	42	
4207	80	ADD B
4208	32	STA 4252
4209	52	
420A	42	
420B	76	HLT

RESULT:

The two data to be added are at 4250 and 4251.The result is stored at 4252.

Data : (4250) = 23
(4251) = 35
Result (4252) = 58

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command.

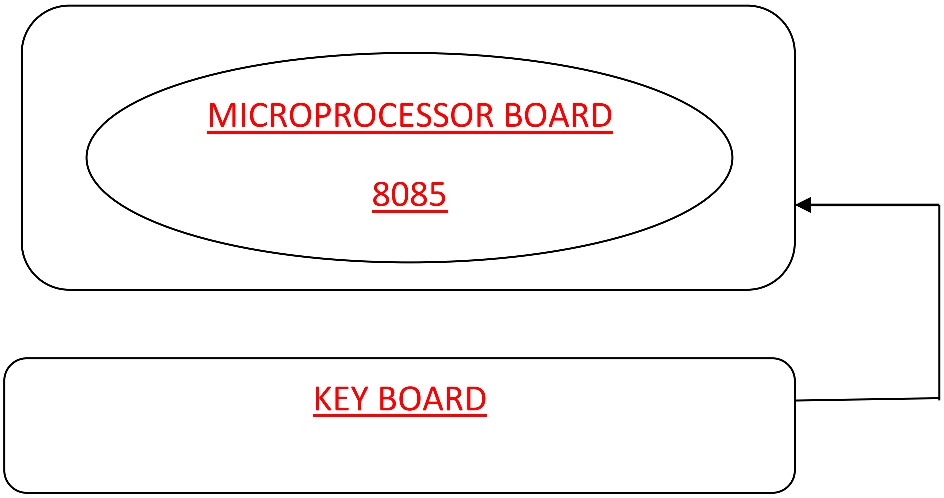
PART (B) :-

AIM:

Subtraction of two eight bit numbers in memory and result also in memory.

APPRATUS:

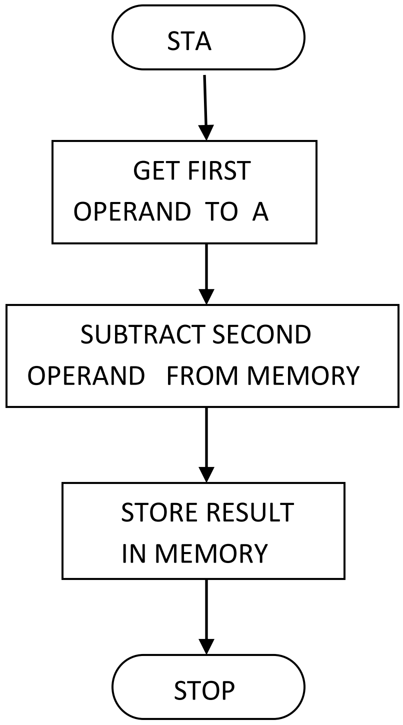
Microprocessor Kit 8085 , Key Board, Op-Code Sheet.



THEORY:

In this experiment, the HL register pair is first initialized to the start address of memory at which the data is stored. Then data is brought to accumulator A and the other one is subtracted from memory itself. The result from A is then stored into memory again using the HL register. The SUB instruction sets and clears flags according to result.

FLOW CHART



PRECAUTIONS :

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don’t touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

i) Key in the Opcodes from the address specified.

ii) Enter data that is needed for execution at 4250 and 4251.

PROGRAM:

```
     LXI       H,  4250          ;  Initialize memory pointer to 4250

    MOV      A,   M             ;  (4150) =>   (A)    =  49
```

```
 INX      H              ;  Point to next data.

 SUB      M               ;  (4251)  =>  (A)

INX      H              ;  Point to next location.

MOV     M,  A           ;   (A)     =>  (4252)

HLT
```

OBJECT CODES:

MEMORY ADDRESS	Opcodes	Mnemonics
4200	21	LXI H, 4250
4201	50	
4202	42	
4203	7E	MOV A, M
4204	23	INX H
4205	96	SUB M
4206	23	INX H
4207	77	MOV M, A
4208	76	HLT

RESULT:

Let the data to be subtracted be 24 from 49.

Data	:	(4250)	=	49
		(4251)	=	24
Result	:	(4252)	=	25

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command.

PART (C) :-

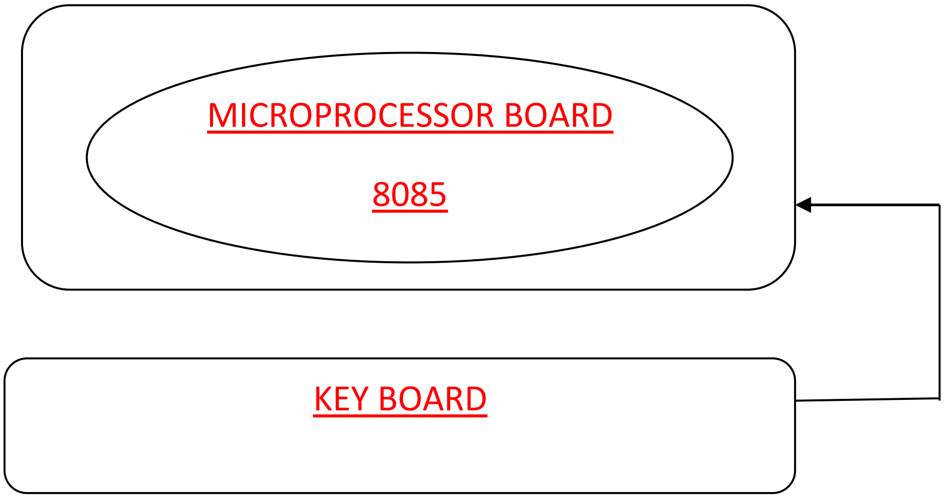
AIM:

Multiplication of two eight bit numbers in memory and result also in memory.

.

APPRATUS:

Microprocessor Kit 8085 , Key Board, Op-Code Sheet.

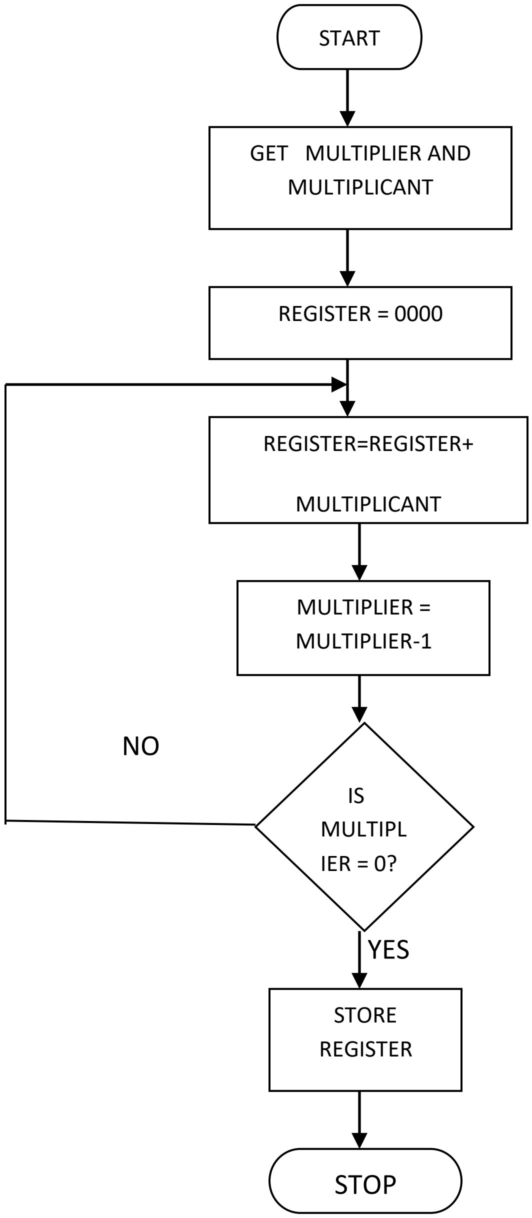


THEORY:

As far as the examples are concerned, they are very simple in the sense that 8085 offers instructions to do those tasks. Since there is no such instruction to multiply or to divide, some logic has to be applied to do these tasks. Multiplication can be done by done by repeated addition while division by repeated subtraction.

In this experiment, the multiplicand and the multiplier stored at memory locations are taken up by the register pair DE and resister B .Using the DAD instruction, 16-bit addition is done repeatedly by the multiplicand until the multiplier becomes zero. The result is again stored in a memory location.

FLOW CHART


START
GET MULTIPLIER AND MULTIPLICANT
REGISTER = 0000
REGISTER=REGISTER+ MULTIPLICANT
MULTIPLIER = MULTIPLIER-1
NO
IS MULTIPLIER = 0?
YES
STORE REGISTER
STOP

PRECAUTIONS :

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don’t touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

- **i)** Key in the Opcodes from the address specified.
- **ii)** Enter data at 4250 through 4253 for execution.
- **iii)** Execute the program and check for results at 4254 and 4255.
- **iv)** Try changing data and check results each time.

PROGRAM:

```
                 LDA     4252           ;          Load Multiplier

                 MOV     B, A           ;          Get the multiplier to B

                 LXI     D,  0000

                 LHLD    4250           ;

                 XCHG                   ;          Load multiplicand in DE

LOOP   :         DAD     D

                 DCR     B

                 JNZ     LOOP           ;          If not zero loop again.

                 SHLD    4254           ;          Else store result.

                 HLT
```

OBJECT CODES:

Memory Address	Opcodes	Mnemonics
4200	3A	LDA 4252
4201	52	
4202	42	
4203	47	MOV B,A
4204	11	LXI D, 0000
4205	00	
4206	00	
4207	2A	LHLD 4250
4208	50	
4209	42	
420A	EB	XCHG
420B	19	LOOP: DAD D

420C	05	DCR B
420D	C2	JNZ LOOP
420E	0B	
420F	42	
4210	22	SHLD 4254
4211	54	
4212	42	
4213	76	HLT

RESULT:

Let the multiplicand and the multiplier be at locations 4250 and 4252. The result will be stored at location 4254. The data at locations are:

Data : (4250) = 93 (4252) = 23
(4251) = 00 (4253) = 00
Result : (4254) = 19
(4255) = 14
93 × 23 = 1419

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command.

PART (D) :-

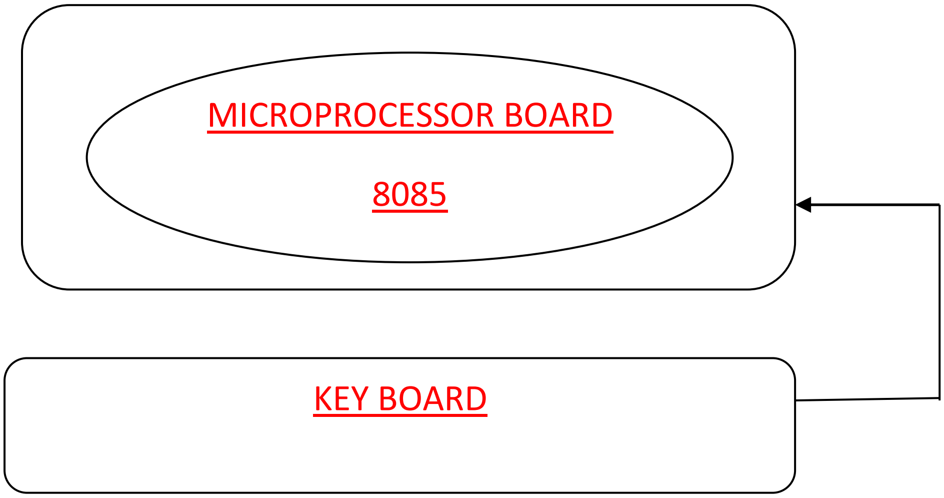
AIM:

Division of two eight bit numbers in memory and result also in memory

.

APPRATUS:

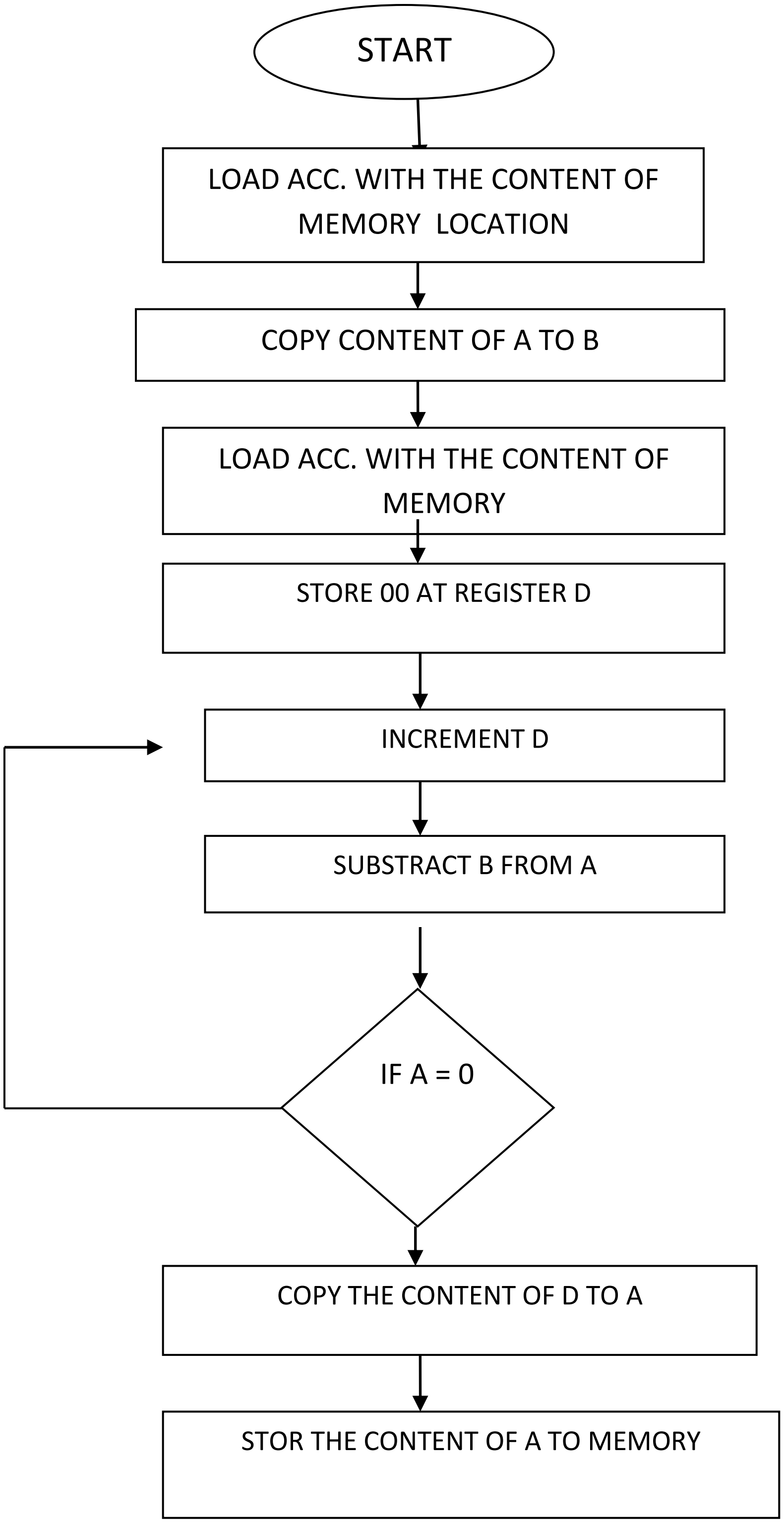
Microprocessor Kit 8085, Key Board, Op-Code Sheet.

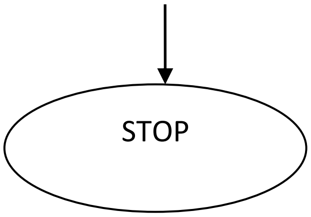


THEORY:

The divisor is subtracted from the 8 most significant bits of the dividend. If there is no borrow, the bit of the quotient is set to 1; otherwise 0. To line up the dividend and quotient properly the dividend is sifted left by one bit before each trial of subtraction. The dividend quotient shares a 16-bit register. Due to shift of dividend one bit of the registers falls vacant in each step. The quotient is stored in vacant position.

FLOW CHART :


START
LOAD ACC. WITH THE CONTENT OF MEMORY LOCATION
COPY CONTENT OF A TO B
LOAD ACC. WITH THE CONTENT OF MEMORY
STORE 00 AT REGISTER D
INCREMENT D
SUBSTRACT B FROM A
IF A = 0
COPY THE CONTENT OF D TO A
STOR THE CONTENT OF A TO MEMORY



PRECAUTIONS :

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don’t touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE

i) Key in the Opcodes from the address specified.

ii) Enter data at 4250 and 4251 as specified in the example.

iii) Execute the program and check for the result at 4252.

iv) Change data at 4250 and 4251 and execute each time and check for result.

PROGRAME:

```
            LDA     4250      ;   A <= 20 (4250)

            MOV     B, A      ;   A <= B

            LDA     4251      ;   A<= 05 (4251)

            MVI     D, 00     ;   D<=00

LOOP;       INR     D         ;   D<=0+1

            SUB     B         ;   B<= A-B;

            JNZ               ;   LOOP

            MOV     A, D      ;   A<=D

            STA     4252      ;   (4252) <= A;

            HLT
```

OBJECT CODES:

Memory Address	Opcodes	Mnemonics
4200	3A	LDA 4250
4201	50	
4202	42	
4203	47	MOV B,A
4204	3A	LDA 4251
4205	51	
4206	42	
4207	16	MVI D, 00
4208	00	
4209	14	INR D
420A	90	SUB B
420B	C2	JNZ
420C	09	
420D	42	
420E	7A	MOV A,D
420F	32	STA 4252
4210	52	
4211	42	
4212	76	HLT

RESULT:

Let the dividend and the divider is at locations 4250 and 4251. The result will be stored at location 4252. The data at locations are:

Data : (4250) = 20
(4251) = 05
Result : (4252) = 04

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command.

EXPERIMENT NO. 04

PART (A):-

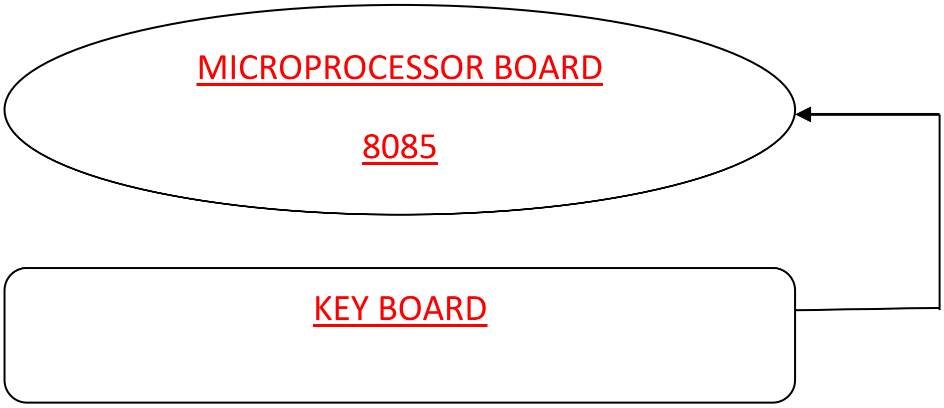
AIM:

Addition of two sixteen bit numbers in memory and result also in memory

.

APPRATUS:

Microprocessor Kit 8085 , Key Board, Op-Code Sheet.



THEORY:

The first data is brought to Accumulator A and the second one in any one of the other registers, say B. The addition is done using ADD. The result is then stored at 4252. The ADD instruction affects flags depending on result.

PRECAUTIONS :

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don't touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

i) Key in the Opcodes from the address specified.

ii) Enter data at 4250 and 4252 as specified in the example.

iii) Execute the program and check for the result at 4254 and 4256.

iv) Change data at 4250 and 4252 and execute each time and check for result.

PROGRAME & OBJECT CODES :

Memory Add.	Opcodes	Mnemonics	Comments
4200	2A, 50, 42	LHLD 4250	1st 16 bit in H-L pair
4203	EB	XCHG	Get 1st no. in D-E pair
4204	2A, 52, 42	LHLD 4252	2nd 16 bit no. in H-L pair
4207	0E, 00	MVI C, 00	Initial value C=00
4209	19	DAD D	1st no.+2nd no.
420A	D2, 0E , 42	JNC *;LOOP*	Is carry? No, go to the Label
420D	0C	INR C	Yes, Increment C.
420E	22, 54, 42 *;LOOP*	SHLD 4254	Store LSBs of sum in 4254 & 4255
4211	79	MOV A,C	MSBs of sum in accumulator.
4212	32, 56, 42	STA 4256	Store MSBs of the sum in 4256.
4215	76	HLT	Halt

RESULT:

DATA: 4250 – 98, LSBs of 1st number
4251 – 5 B, MSBs of 1st number
4252 – 4C, LSBs of 2nd number
4253 – 8E , MSBs of 2nd number

RESULT: 4254 – E4, LSBs of sum.
4255 – E9 , LSBs of sum.
4256 – 8E, MSB of 2nd sum.

DATA: 4250 – 45, LSBs of 1st number
4251 – A6, MSBs of 1st number
4252 – 23, LSBs of 2nd number
4253 –9B , MSBs of 2nd number

RESULT: 4254 – 68 LSBs of sum.
4255 – 41, LSBs of sum.
4256 – 01, MSB of 2nd sum.

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command.

EXPERIMENT NO. 05

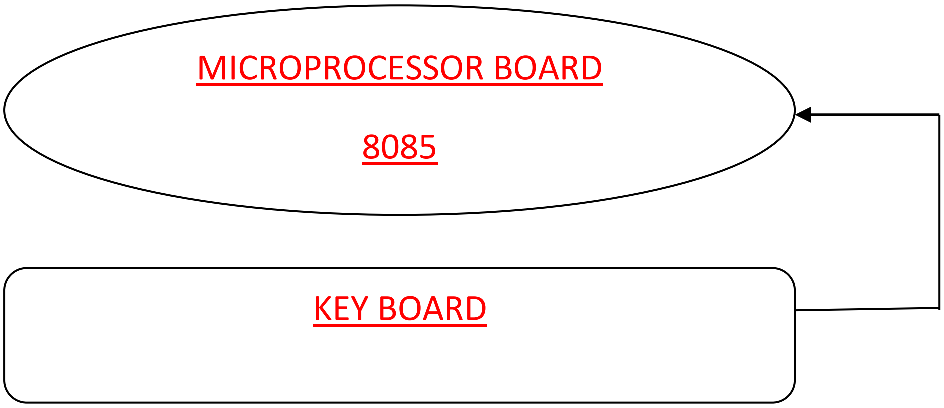
PART(A):

AIM:

1's complement of eight bit number

APPRATUS:

Microprocessor Kit 8085 , Key Board, Op-Code Sheet.



THEORY:

In this experiment to obtain ons's compliment of a number its 0 bits are replaced by 1 and 1 by 0.
The number is placed in the memory location 4250.
The result is stored the memory location 4251

PRECAUTIONS :

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don't touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

- **i)** Key in the Opcodes from the address specified.
- **ii)** Enter data at 4250 specified in the example.
- **iii)** Execute the program and check for the result at 4251.
- **iv)** Change data at 4250 and execute each time and check for result.

PROGRAME & OBJECT CODES :

Memory Address	Opcodes	Mnemonics	Comments
4200	3A , 50, 42	LDA 4250	Get data in accumulator
4203	2F	CMA	Take its complement.
4204	32, 42, 51	STA	Store result in 4251
4207	76	HLT	Stop

RESULT:

DATA: 4250 – 96
RESULT: 4251 -- 69
DATA: 4250 – E4
RESULT: 4251 – 1B.

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command.

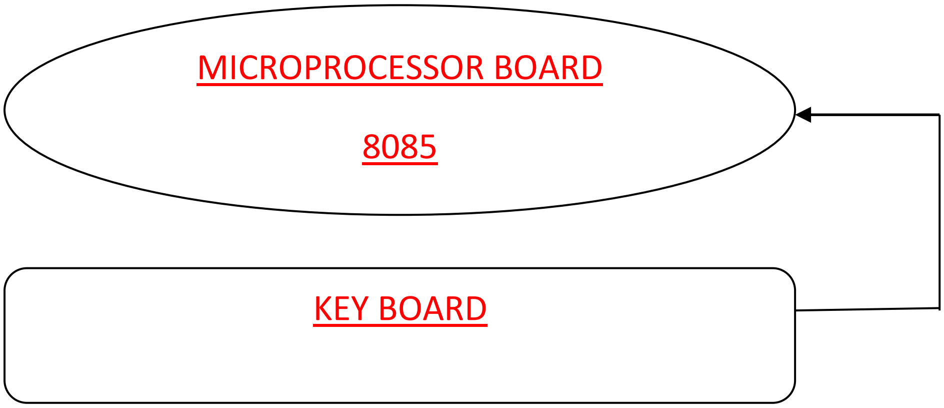
PART(B):

AIM:

2’s complement of eight bit number

APPRATUS:

Microprocessor Kit 8085 , Key Board, Op-Code Sheet.



THEORY:

In this experiment to obtain Two’s compliment of a number is obtain by adding 1 to the 1’s compliment of the number.

The number is placed in the memory location 4250.

The result is to be stored in the memory location 4251.

PRECAUTIONS :

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don’t touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

v) Key in the Opcodes from the address specified.
vi) Enter data at 4250 specified in the example.
vii) Execute the program and check for the result at 4251.
viii) Change data at 4250 and execute each time and check for result.

PROGRAME & OBJECT CODES :

Memory Address	Opcodes	Mnemonics	Comments
4200	3A , 50, 42	LDA 4250	Get data in accumulator
4203	2F	CMA	Take its 1's complement.
4204	3C	INR A	Take 2's complement.
4207	32, 51 ,42	STA 4252	Store result in 4251.
4208	76	HLT	Stop

RESULT:

DATA: 4250 – 96
RESULT: 4251 – 6A
DATA: 4250 – E4
RESULT: 4251 – 1C

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command.

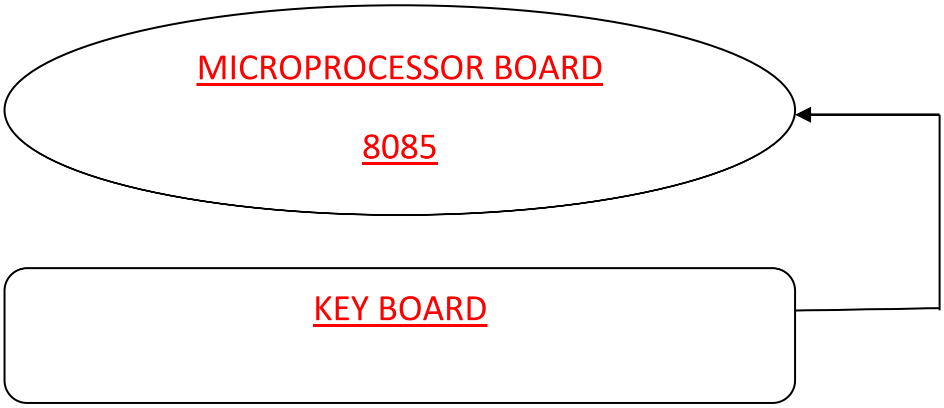
PART(C):

AIM:

1's complement of sixteen bit number

APPRATUS:

Microprocessor Kit 8085 , Key Board, Op-Code Sheet.



THEORY:

In this experiment, The 8 LSBs of the number are in the memory location 4250. The address 4250 is placed in H-L pair. The 8 LSBs of the number are transferred from 4250 to the accumulator. The instruction CMA takes one's complement of 8 LSBs. The 8 LSBs of the result are stored in the memory location 4252. The address 8 MSBs of the number is 4251, and it is placed in H-L pair .The 8 MSBs of the number are transferred from 4251 to the accumulator. The instruction CMA takes one's compliment of 8 MSBs. The 8 MSBs of the result are stored in memory location 4253.

PRECAUTIONS :

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don't touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Op-codes from the address specified.
2. Enter data at 4251 and 4252 specified in the example.
3. Execute the program and check for the result at 4253 and 4254.
4. Change data at 4251 and 4252 and execute each time and check for result.

PROGRAME & OBJECT CODES :

Memory Add.	Op-codes	Mnemonics	Comments
4200	21, 51,42	LXI H, 4251	Address OF LSBs of the number.
4203	7E	MOV A,M	8 LSBs of the number in accumulator.
4204	2F	CMA	Complement of 8 LSBs of the number.
4205	32, 53, 42	STA 4253	Store 8 LSBs of result.
4208	23	INX H	Address of 8 MSBs of the number.
4209	7E	MOV A, M	8 MSBs of the number in accumulator.
420A	2F	CMA	Complement of 8 MSBs of the number.
420B	32, 54, 42	STA 4254	Store 8 MSBs of the result.
420E	76	HLT	Stop

RESULT:

DATA: 4251 – 85, LSB of the number
4252 – 54, MSBs of the number.
RESULT: 4253 – 7A, LSBs of the result.
4254 – AB, MSBs of the result.

DATA: 4251 – 7E, LSB of the number
4252 – 89, MSBs of the number.
RESULT: 4253 – 81, LSBs of the result.
4254 – 76, MSBs of the result.

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command.

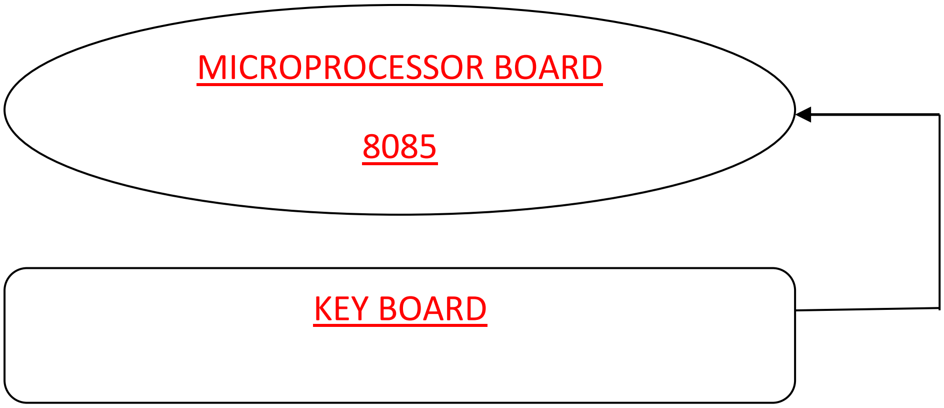
PART(D):

AIM:

2’s complement of sixteen bit number

APPRATUS:

Microprocessor Kit 8085 , Key Board, Op-Code Sheet.



THEORY:

In this experiment, The 8 LSBs of the number are in memory location 4251. They are taken first and 1’s compliment is obtained. To obtain 2’s complement 1 is added to 1’s complement. 8 LSBs of 2’s complement are stored in 4253. The carry resulting from the addition of 1 to 1’s complement is stored in register. After this 8 MSBs of the number are taken and 1’s complement is obtained. They carry is added to it. The 8 MSBs of the result are stored in 4254. In case of no carry the program jumps from JNC GO to INX H and the content of register B is not incremented. It remains zero. The addition of zero to 1’s complement of 8 MSBs does not affect result.

PRECAUTIONS :

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don’t touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Op-codes from the address specified.
2. Enter data at 4251 and 4252 specified in the example.
3. Execute the program and check for the result at 4253 and 4254.
4. Change data at 4251 and 4252 and execute each time and check for result.

PROGRAME & OBJECT CODES :

Memory Add.	Op-codes	Mnemonics	Comments
4200	21, 51,42	LXI H,4251	Address OF LSBs of the number.
4203	06, 00	MVI B,00	Use registers B to store carry.
4205	7E	MOV A,M	8 LSB in accumulator.
4206	2F	CMA	1's Complement of 8 LSBs of the number.
4207	C6, 01	ADI 01	2's complement of 8 LSBs of the number
4209	32, 03, 25	STA 4253	Store 8 LSBs of result.
420C	D2, 10, 42	JNC ***; GO***	Label
420F	04	INR B	Store carry.
4210	23	***GO;*** INX H	Address of 8 MSBs of the number.
4211	7E	MOV A,M	8 MSBs in accumulator.
4212	2F	CMA	1's Complement of 8 LSBs of the number.
4213	80	ADD B	Add carry
4214	32, 54, 42	STA 4254	Store 8 MSBs of the result.
4217	76	HLT	Stop

RESULT:

DATA: 4251 – 8C, LSB of the number
4252 – 5B, MSBs of the number.
RESULT: 4253 – 74, LSBs of the result.
4254 – A4, MSBs of the result.
2's complement of the number is A474.

DATA: 4251 – 00, LSB of the number

4252 – 5B, MSBs of the number.
RESULT: 4253 – 00, LSBs of the result.
4254 – A5, MSBs of the result.

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command.

EXPERIMENT NO. 06

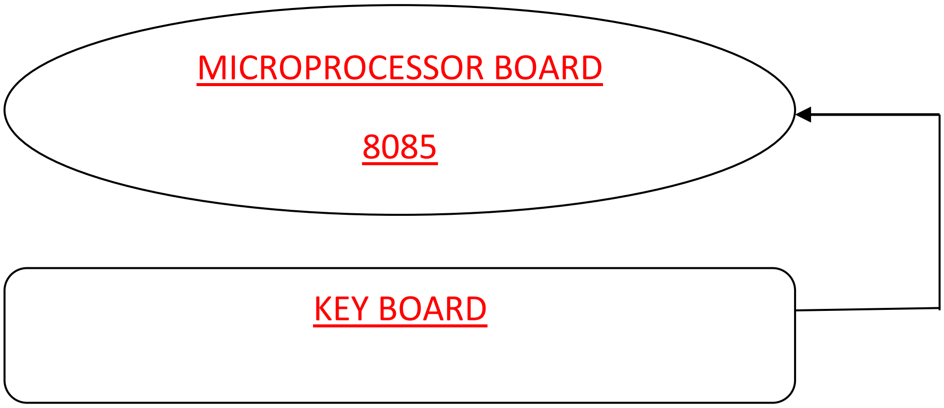
PART(A):

AIM:

Shifting left an eight bit number by one bit

APPRATUS:

Microprocessor Kit 8085 , Key Board, Op-Code Sheet.



THEORY:

In this experiment to obtain, the instruction LDA 4251 transfers the number from memory location 4251 to the accumulator. ADD A adds the contents of the accumulator to itself. The results are twice the number and thus the number is sifted left by one bit. This program does not take carry account after ADD instruction. If number to be handled is likely to produce carry the program may be modified to store it.

PRECAUTIONS :

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don’t touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data at 4250 specified in the example.
3. Execute the program and check for the result at 4251.
4. Change data at 4250 and execute each time and check for result.

PROGRAME & OBJECT CODES :

Memory Address	Opcodes	Mnemonics	Comments
4200	3A , 50, 42	LDA 4250	Get data in accumulator
4203	87	ADD A	Shift its left by one bit.
4204	32, 42, 51	STA 4251	Store result in 4251
4207	76	HLT	Stop

RESULT:

DATA: 4250 – 65
RESULT: 4251 -- CA

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command.

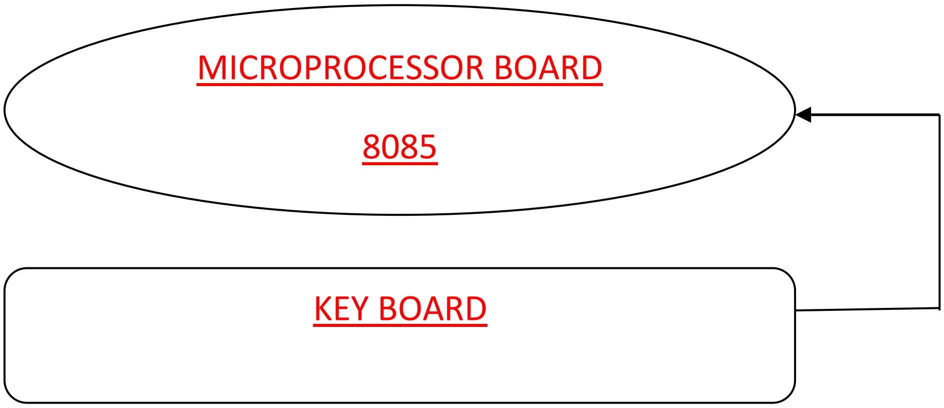
PART(B):

AIM:

Shifting left an eight bit number by two bit.

APPRATUS:

Microprocessor Kit 8085 , Key Board, Op-Code Sheet.



THEORY:

In this experiment to obtain, the instruction LDA 4251 transfers the number from memory location 4251 to the accumulator. ADD A adds the contents of the accumulator to itself. The results are twice the number and thus the number is sifted left by one bit. This program does not take carry account after ADD instruction. If number to be handled is likely to produce carry the program may be modified to store it.

PRECAUTIONS :

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don’t touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data at 4250 specified in the example.
3. Execute the program and check for the result at 4251.
4. Change data at 4250 and execute each time and check for result.

PROGRAME & OBJECT CODES :

Memory Address	Opcodes	Mnemonics	Comments
4200	3A , 50, 42	LDA 4250	Get data in accumulator
4203	87	ADD A	Shift its left by one bit..
4204	87	ADD A	Again Shift its left by one bit..
4205	32, 42, 51	STA 4251	Store result in 4251
4208	76	HLT	Stop

RESULT:

DATA: 4250 – 15
RESULT: 4251 -- 54

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command.

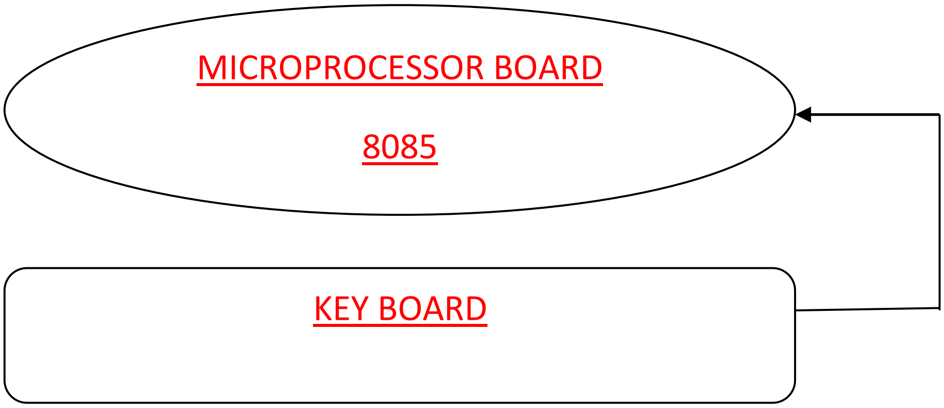
PART(C)

AIM:

Shifting left a sixteen bit number by one bit

APPRATUS:

Microprocessor Kit 8085 , Key Board, Op-Code Sheet.



THEORY:

In this experiment to obtain, the instruction LDA 4251 transfers the number from memory location 4251 to the accumulator. ADD A adds the contents of the accumulator to itself. The results are twice the number and thus the number is sifted left by one bit. This program does not take carry account after ADD instruction. If number to be handled is likely to produce carry the program may be modified to store it.

PRECAUTIONS :

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don’t touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data at 4250 specified in the example.
3. Execute the program and check for the result at 4252.
4. Change data at 4250 and execute each time and check for result.

PROGRAME & OBJECT CODES :

Memory Address	Opcodes	Mnemonics	Comments
4200	2A , 50, 42	LHLD 4250	Get data in H-L Pair
4203	29	DAD H	Shift its left by one bit..
4204	22, 52, 42	SHLD 4252	Store result in 4252 and 4253
4207	76	HLT	Stop

RESULT:

DATA: 4250 – 7596
RESULT: 4251 – EB2C
DATA
4250-96 H ,LSBs of the number
4251-75 H, MSBs of the number
RESULT
4252- 2C, LSBs of the result
4253- EB, MSBs of the result

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command.

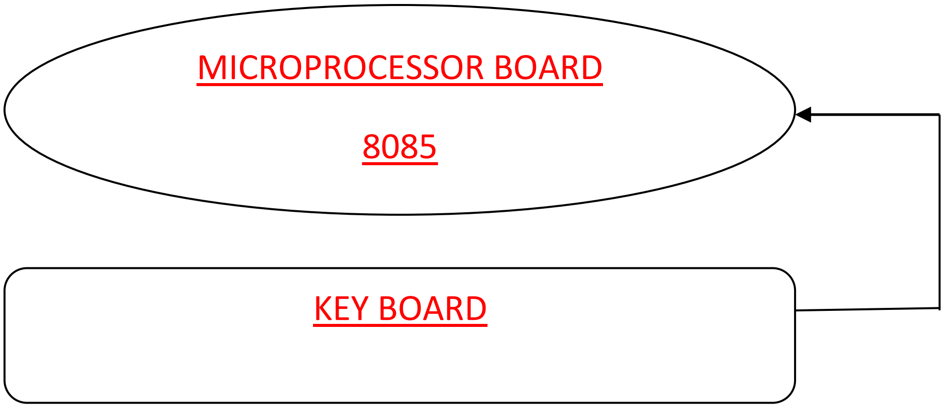
PART(D)

AIM:

Shifting left a sixteen bit number by two bit .

APPRATUS:

Microprocessor Kit 8085 , Key Board, Op-Code Sheet.



THEORY:

In this experiment to obtain, the instruction LDA 4251 transfers the number from memory location 4251 to the accumulator. ADD A adds the contents of the accumulator to itself. The results are twice the number and thus the number is sifted left by one bit. This program does not take carry account after ADD instruction. If number to be handled is likely to produce carry the program may be modified to store it.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don't touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data at 4250 specified in the example.
3. Execute the program and check for the result at 4252.
4. Change data at 4250 and execute each time and check for result.

PROGRAME & OBJECT CODES :

Memory Address	Opcodes	Mnemonics	Comments
4200	2A , 50, 42	LHLD 4250	Get data in H-L Pair
4203	29	DAD H	Shift its left by one bit.
4204	29	DAD H	Again Shift its left by one bit.
4205	22, 52, 42	SHLD 4252	Store result in 4252 and 4253
4208	76	HLT	Stop

RESULT:

DATA: 4250 – 1596
RESULT: 4251 – 5658

DATA

4250-96 H, LSBs of the number

4251-15 H, MSBs of the number

RESULT

4252- 58, LSBs of the result

4253- 56, MSBs of the result

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.

2. Make space whenever you doing program on kit.

3. You should properly enter the program and exit through proper command

EXPERIMENT NO. 07

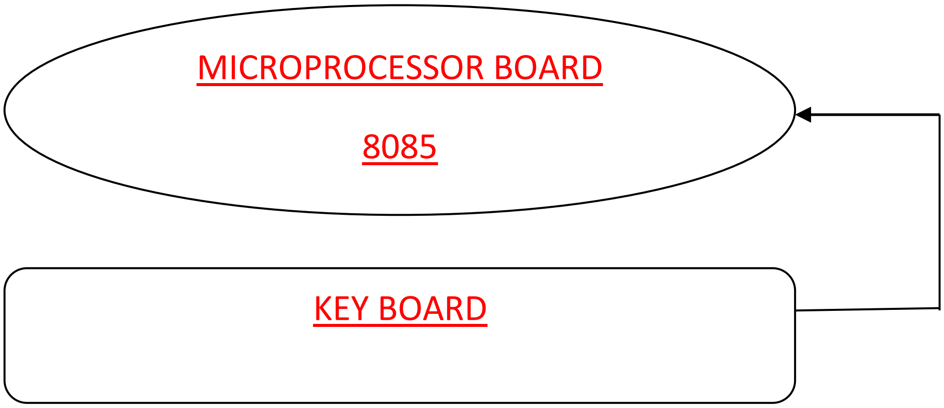
PART(A):

AIM:

Mask off Least significant 4 bits of an eight bit number

APPRATUS:

Microprocessor Kit 8085 , Key Board, Op-Code Sheet.



THEORY:

In this experiment to obtain, the instruction LDA 4251 transfers the number from memory location 4251 to the accumulator. ADD A adds the contents of the accumulator to itself. The results are twice the number and thus the number is sifted left by one bit. This program does not take carry account after ADD instruction. If number to be handled is likely to produce carry the program may be modified to store it.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don't touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data at 4250 specified in the example.
3. Execute the program and check for the result at 4251.
4. Change data at 4250 and execute each time and check for result.

PROGRAME & OBJECT CODES :

Memory Address	Opcodes	Mnemonics	Comments
4200	3A , 50, 42	LDA 4250	Get data in accumulator
4203	E6,F0	ANI F0	Mask off the most significant 4 bits
4205	32, 51, 42	STA 4251	Store result in 4252
4208	76	HLT	Stop

RESULT:

DATA: 4250 –A6
RESULT: 4251 –A0

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command

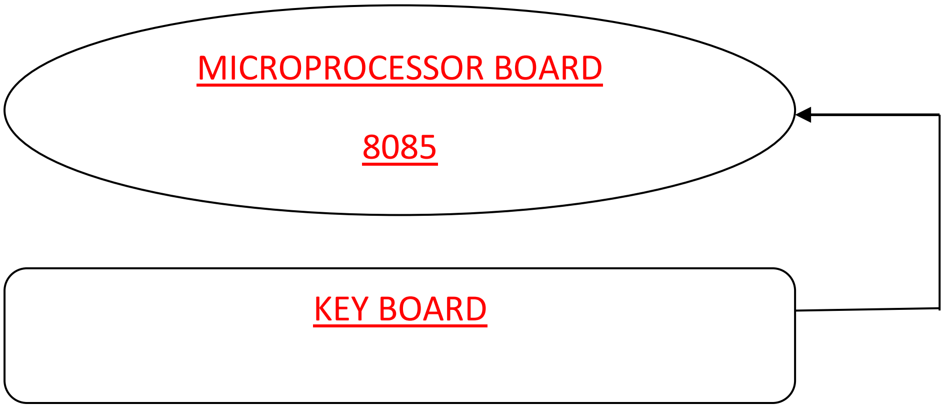
PART(B):

AIM:

Mask off Most significant 4 bits of an eight bit number

APPRATUS:

Microprocessor Kit 8085 , Key Board, Op-Code Sheet.



THEORY:

In this experiment to obtain, the instruction LDA 4251 transfers the number from memory location 4251 to the accumulator. ADD A adds the contents of the accumulator to itself. The results are twice the number and thus the number is sifted left by one bit. This program does not take carry account after ADD instruction. If number to be handled is likely to produce carry the program may be modified to store it.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don’t touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data at 4250 specified in the example.
3. Execute the program and check for the result at 4251.
4. Change data at 4250 and execute each time and check for result.

PROGRAME & OBJECT CODES :

Memory Address	Opcodes	Mnemonics	Comments
4200	3A , 50, 42	LDA 4250	Get data in accumulator
4203	E6,0F	ANI 0F	Mask off the most significant 4 bits
4205	32, 51, 42	STA 4251	Store result in 4252
4208	76	HLT	Stop

RESULT:

DATA: 4250 –A6
RESULT: 4251 –06

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command

EXPERIMENT NO. 08

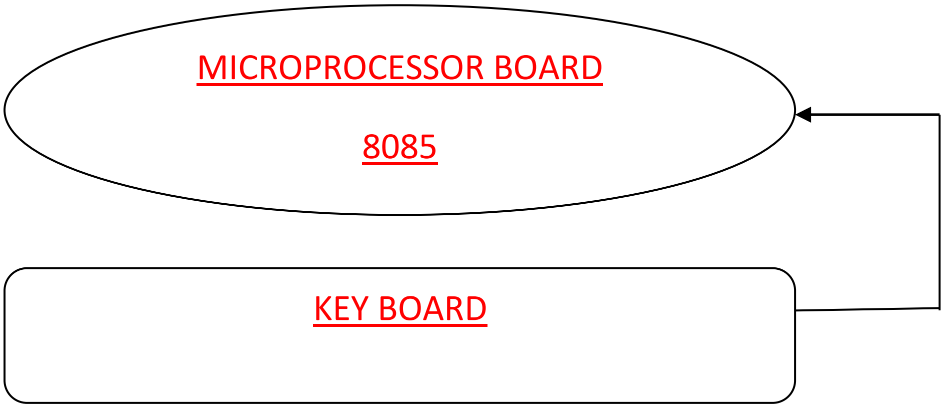
PART(A):

AIM:

Program for Mutibyte addition

APPRATUS:

Microprocessor Kit 8085 , Key Board, Op-Code Sheet.



THEORY:

A byte consists of 8-bits. In this experiment two multibyte hex numbers are to be added. Each number consists of 4 bytes. An 8-bit microcomputer takes one byte of the number at a time and adds them with carry. A counter is initiated to count the byte.
The count is placed in memory location 4500 H.
The 1st number is placed in the memory location 4501 to 4504 H.
The 2nd number is placed in the memory location 4601 to 4604 H.
The sum is in the memory location 4501 to 4504.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don’t touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data at 4500 and 4601 specified in the example.
3. Execute the program and check for the result at 4501 to 4504.
4. Change data at 4500 & 4601 and execute each time and check for result.

PROGRAME & OBJECT CODES :

Memory Address	Opcodes	Mnemonics	Comments
4200	21, 00, 45	LXI H, 4500	Address of byte coumt in H-L pair.
4203	4E	MOV C,M	Byte count in register C.
4204	23	INX H	Address of 1st byte of 1st number.
4205	11, 01, 46	LXI D, 4601	Address of 1st byte of 2nd number.
4208	B7	ORA A	Clear carry.
4209	1A	**LOOP:** LDAX D	Get byte of 2nd number in accumulator.
420A	8E	ADC M	Byte of 2nd number + byte of 1st number + carry.
420B	77	MOV M,A	Store sum in memory addressed by H-L pair.
420C	23	INX H	Increment the content of H-L pair.
420D	13	INX D	Increment the content of D-E pair.
420E	0D	DCR C	Decrement count
420F	C2,09, 42	JNZ **:LOOP**	Is count = 0? No, jump to LOOP.
42012	76	HLT	Stop.

RESULT:

```
   3A9C8A67,   1st number
+  9B476C8B,   2nd number
    D5E3F6F2      SUM
```

DATA

4500- 04
4501-67
4502-8A
4503-9C
4504-3A

4601-8B
4602-6C
4603-47
4604-9B

The sum is stored in the memory location 4501 to 4504 H.

RESULT

4501-F2
4502-F6
4503-E3
4504-D5

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command.

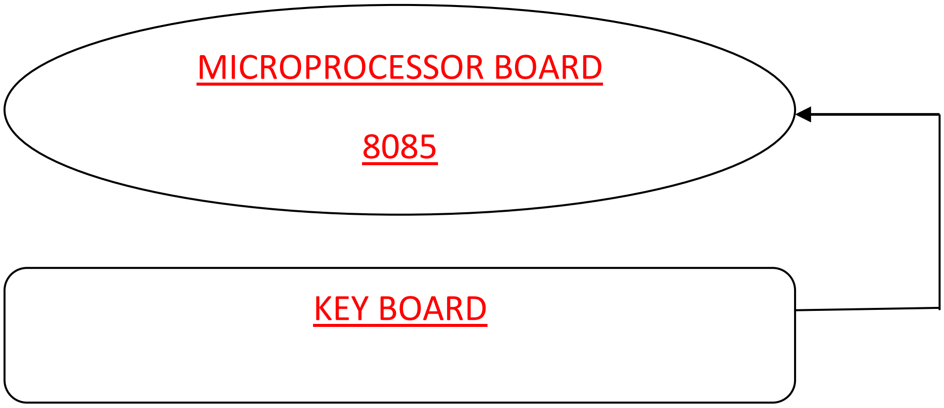
PART(B):

AIM:

Program for Mutibyte subtraction

APPRATUS:

Microprocessor Kit 8085 , Key Board, Op-Code Sheet.



THEORY:

A byte consists of 8-bits. In this experiment two multibyte hex numbers are to be subtracted. Each number consists of 4 bytes. An 8-bit microcomputer takes one byte of the number at a time and subtract them with borrow. A counter is initiated to count the byte.
The count is placed in memory location 4500 H.
The 2nd number is placed in the memory location 4501 to 4504 H.
The 1st number is placed in the memory location 4601 to 4604 H.
The result is placed in the memory location 4501 to 4504H.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don’t touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data at 4500 and 4601 specified in the example.
3. Execute the program and check for the result at 4501 to 4504.
4. Change data at 4500 & 4601 and execute each time and check for result.

PROGRAME & OBJECT CODES :

Memory Address	Opcodes	Mnemonics	Comments
4200	21, 00, 45	LXI H, 4500	Address of byte coumt in H-L pair.
4203	4E	MOV C,M	Byte count in register C.
4204	23	INX H	Address of 1st byte of 1st number.
4205	11, 01, 46	LXI D, 4601	Address of 1st byte of 2nd number.
4208	B7	ORA A	Clear carry.
4209	1A	**LOOP:** LDAX D	Get byte of 2nd number in accumulator.
420A	8E	SBB M	Subtract byte of 2nd number from byte of 1st number with borrow.
420B	77	MOV M,A	Store sum in memory addressed by H-L pair.
420C	23	INX H	Increment the content of H-L pair.
420D	13	INX D	Increment the content of D-E pair.
420E	0D	DCR C	Decrement count
420F	C2,09, 42	JNZ **:LOOP**	Is count = 0? No, jump to LOOP.
42012	76	HLT	Stop.

RESULT:

```
   EC8B945A,   1st number
-  4B3CA8B9,   2nd number
   A14EEBA1
```

DATA

4500- 04
4501-B9
4502-A8
4503-3C
4504-4B

4601-5A
4602-94
4603-8B
4604-EC

The sum is stored in the memory location 4501 to 4504 H.

RESULT

4501-A1
4502-EB
4503-4E
4504-A1

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command.

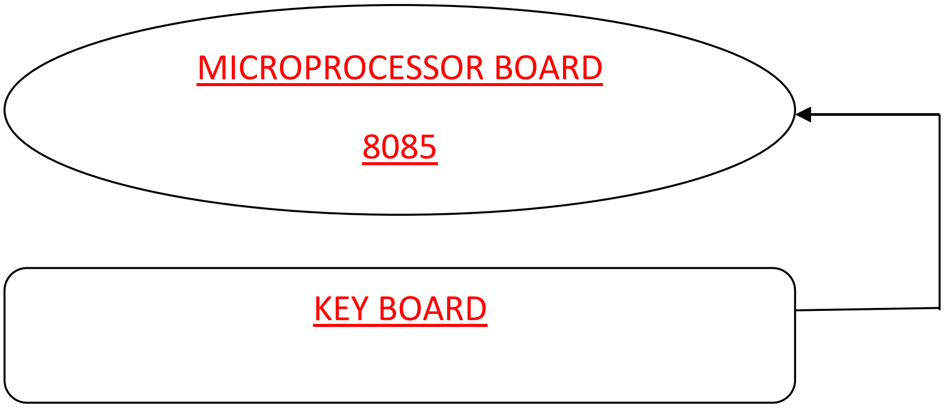
PART(C):

AIM:

Mutibyte decimal addition

APPRATUS:

Microprocessor Kit 8085 , Key Board, Op-Code Sheet.



THEORY:

A byte consists of 8-bits. In this experiment two multibyte decimal numbers are to be added. Each number consists of 4 bytes. DAA instruction has been incorporated and placed after instruction ADC M. DAA also produces carry which is taken into account while adding the next byte the instruction ADC M.
A counter is initiated to count the byte.
The count is placed in memory location 4500 H.
The byte of the 1st number is placed in the memory location 4501 to 4505 H.
The byte of 2nd number is placed in the memory location 4601 to 4605 H.
The sum is stored in the memory location 4501 to 4505.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don't touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data at 4500 and 4601 specified in the example.
3. Execute the program and check for the result at 4501 to 4505.
4. Change data at 4500 & 4601 and execute each time and check for result.

PROGRAME & OBJECT CODES :

Memory Address	Opcodes	Mnemonics	Comments
4200	21, 00, 45	LXI H, 4500	Address of byte coumt in H-L pair.
4203	4E	MOV C,M	Byte count in register C.
4204	23	INX H	Address of 1st byte of 1st number.
4205	11, 01, 46	LXI D, 4601	Address of 1st byte of 2nd number.
4208	B7	ORA A	Clear carry.
4209	1A	**LOOP:** LDAX D	Get byte of 2nd number in accumulator.
420A	8E	ADC M	Byte of 2nd number + byte of 1st number + carry.
420B	27	DAA	Decimal adjust.
420C	77	MOV M,A	Store sum in memory addressed by H-L pair.
420D	23	INX H	Increment the content of H-L pair.
420E	13	INX D	Increment the content of D-E pair.
420F	0D	DCR C	Decrement count
4210	C2,09, 42	JNZ **:LOOP**	Is count = 0? No, jump to LOOP.
42013	76	HLT	Stop.

RESULT:

```
  0096876958  1st numbers
+ 0087358769  2nd number
  0184235727  SUM
```

DATA

4500- 05
4501-58
4502-69
4503-87
4504-96
4505-00

4601-69
4602-87
4603-35
4604-87
4605-00

The sum is stored in the memory location 4501 to 4505 H.

RESULT

4501-27
4502-57
4503-23
4504-84
4505-01

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.

2. Make space whenever you doing program on kit.

3. You should properly enter the program and exit through proper command.

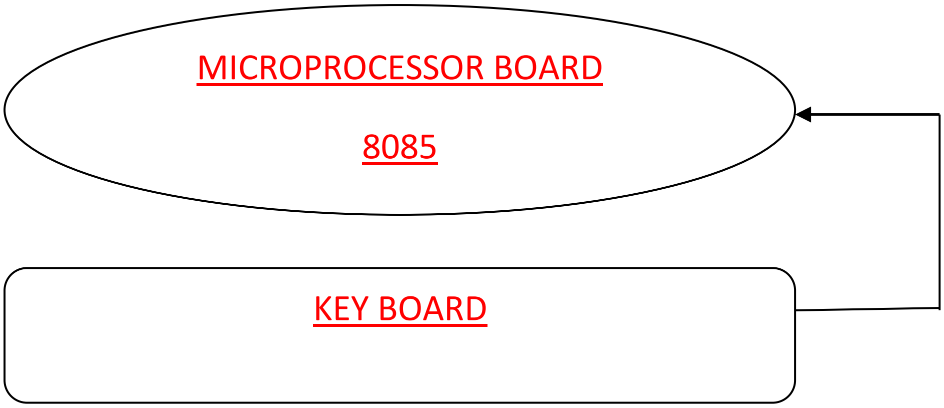
PART(D):

AIM:

Mutibyte decimal subtraction

APPRATUS:

Microprocessor Kit 8085 , Key Board, Op-Code Sheet.



THEORY:

The DAA instruction is not used after SUB or SBB instruction for decimal adjustment. It is used only after ADD, ADI,ACI or ADC instruction. Therefore, for decimal subtraction the number which is to be subtracted is converted 10's compliment and then it is added to the other number.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don't touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data at 4500 and stack pointer at 4601 specified in the example.
3. Execute the program and check for the result at 4501 to 4504.
4. Change data at 4500 & 4601 and execute each time and check for result.

PROGRAME & OBJECT CODES :

Memory Address	Opcodes	Mnemonics	Comments
4200	31, 00, 47	LXI SP,4700	Initialise stack pointer.
4203	21, 00, 45	LXI H,4500	Address for byte count in H-L pair
4206	4E	MOV C,M	Byte count in register C.
4207	23	INX H	Address of 1st byte of 1st number.
4208	11, 01, 46	LXI D,4601	Address of 1st byte of 1st number in D-E pair.
420B	1A	LDAX D	1st byte of 2nd number in accumulator.
420C	47	MOV B,A	Get in register B.
420D	3E, 99	MVI A,99	Get 99 in accumulator.
420F	90	SUB B	Take 9’s compliment of 1st byte.
4210	3C	INR A	10’s compliment.
4211	86	ADD M	Add it to 1st byte of 1st number
4212	27	DAA	Decimal adjust
4213	F5	PUSH PSW	Save carry.
4214	77	MOV M,A	Store result.
4215	23	**LOOP:** INX H	Next byte of 1st number.
4216	13	INX D	Address for next byte of 2nd number.
4217	1A	LDAX D	Next byte of 2nd number in accumulator.
4218	47	MOV B,A	Get it in register B.
4219	3E, 99	MVI A,99	99 in accumulator.

421B	90	SUB B	Take its 9’s compliment.
421C	47	MOV B,A	
421D	F1	POP PSW	Get carry of previous byte calculation
421E	78	MOV A,B	9’s compliment in accumulator.
421F	8E	ADC M	Add to it byte of 1st number and carry.
4220	27	DAA	Decimal adjust.
4221	F5	PUSH PSW	Save Carry.
4222	77	MOV M,A	Store result
4223	0D	DCR C	Decrement count.
4224	C2, 15, 42	JNZ **:LOOP**	
4227	76	HLT	Stop.

RESULT:

57849900 1st numbers
- 28975899 2nd number
288740017 Result

DATA
4500- 04
4501-00
4502-99
4503-84
4504-57

4601-99
4602-58
4603-97
4604-28

The sum is stored in the memory location 4501 to 4505 H.

RESULT
4501-70
4502-09
4503-18
4504-70

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command.

EXPERIMENT NO. 09

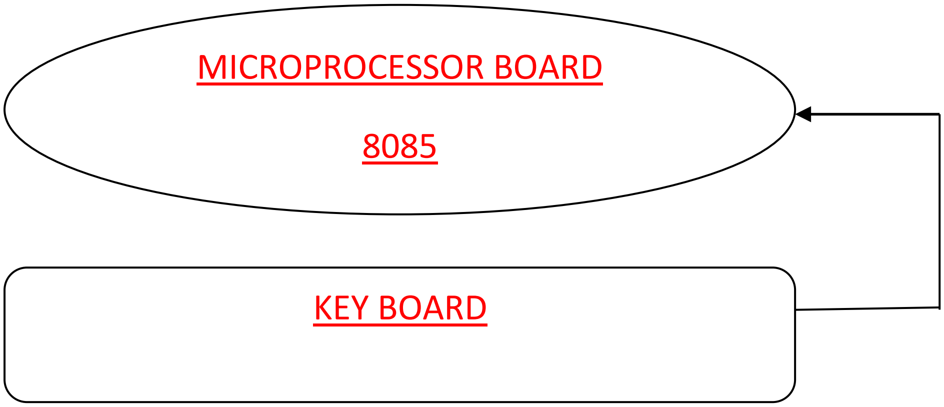
PART(A):

AIM:

Program for finding a smallest number in a data array

APPRATUS:

Microprocessor Kit 8085 , Key Board, Op-Code Sheet.



THEORY:

The 1st number of the series is placed in the accumulator and it is compared with 2nd number which is in the memory. The smaller of the two is placed in the accumulator. Again this number which is in the accumulator is compared with the 3rd number of the series and smaller number is placed in the accumulator. This process of comparison is repeated till all the number of the series is compared and smallest number is stored in the desired memory location.

As there are three numbers in the series, count =03

The count is placed in the memory location 4500H.

The number is placed in the memory location 4501 to 4503H.

The result is to be stored in the memory location 4550H.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don't touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data at 4500 specified in the example.
3. Execute the program and check for the result at 4550.
4. Change data at 4500 and execute each time and check for result.

PROGRAME & OBJECT CODES :

Memory Address	Opcodes	Mnemonics	Comments
4200	21, 00, 45	LXI H, 4500	Get address for count in H-L pair.
4203	4E	MOV C,M	Count in register C.
4204	23	INX H	Get address of 1st number in H-L pair.
4205	7E	MOV A,M	1st number in accumulator .
4206	0D	DCR C	Decrement count.
4207	23	**LOOP:** INX H	Address of next number in H-L pair.
4208	BE	CMP M	Compare next number with previous smallest. Is previous smallest less than next number?
4209	DA, 0D, 42	JC **:AHEAD**	Yes, smaller number in accumulator. Go to AHEAD
420C	7E	MOV A,M	No, get next number in accumulator.
420D	0D	**AHEAD:** DCR C	Decrement count
420E	C2, 07, 42	JNZ **:LOOP**	
4211	32, 50, 45	STA 4550	Store smallest number in 4550.
42014	76	HLT	Stop.

RESULT:

DATA	**DATA**
4500- 03	4500-05
4501-86	4501-EB
4502-58	4502-D4
4503-75	4503-3C

	4504-0F
	4505-A8
RESULT	**RESULT**
4550-58.	4550-0F

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command.

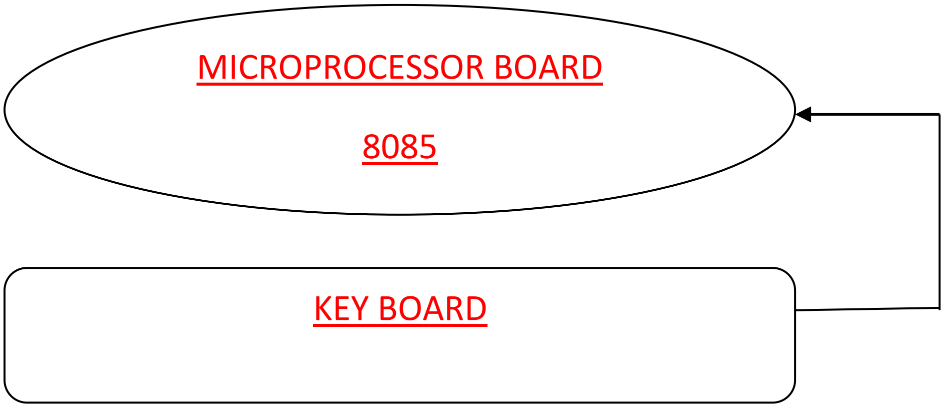
PART(B):

AIM:

Program for finding a largest number in a data array

APPRATUS:

Microprocessor Kit 8085 , Key Board, Op-Code Sheet.



THEORY:

The 1st number of the series is placed in the accumulator and it is compared with 2nd number which is in the memory. The larger of the two is placed in the accumulator. Again this number which is in the accumulator is compared with the 3rd number of the series and larger number is placed in the accumulator. This process of comparison is repeated till all the number of the series is compared and largest number is stored in the desired memory location.

As there are three numbers in the series, count =03

The count is placed in the memory location 4500H.

The number is placed in the memory location 4501 to 4503H.

The result is to be stored in the memory location 4550H.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don't touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data at 4500 specified in the example.
3. Execute the program and check for the result at 4550.
4. Change data at 4500 and execute each time and check for result.

PROGRAME & OBJECT CODES :

Memory Address	Opcodes	Mnemonics	Comments
4200	21, 00, 45	LXI H, 4500	Get address for count in H-L pair.
4203	4E	MOV C,M	Count in register C.
4204	23	INX H	Get address of 1st number in H-L pair.
4205	7E	MOV A,M	1st number in accumulator.
4206	0D	DCR C	Decrement count.
4207	23	**LOOP:** INX H	Address of next number in H-L pair.
4208	BE	CMP M	Compare next number with previous maximum. Is next number > previous maximum?
4209	D2, 0D, 42	JNC **:AHEAD**	No, larger number is in accumulator. Go to AHEAD
420C	7E	MOV A,M	Yes, get larger number in accumulator.
420D	0D	**AHEAD:** DCR C	Decrement count
420E	C2,07, 42	JNZ **:LOOP**	
4211	32, 50, 45	STA 4550	Store smallest number in 4550.
4214	76	HLT	Stop.

RESULT:

DATA	**DATA**
4500- 03	4500-06
4501-98	4501-38
4502-75	4502-94
4503-99	4503-EB
	4504-A8

	4505-B5
	4506-FB
RESULT	**RESULT**
4550-99.	4550-FB

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command.

EXPERIMENT NO. 10

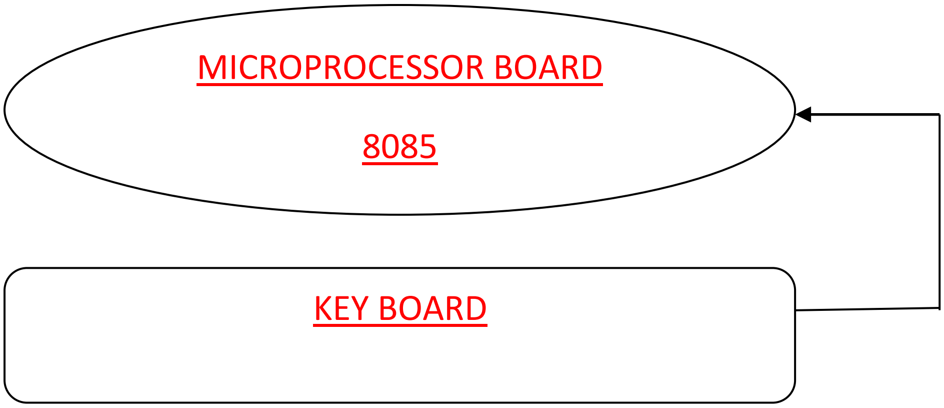
PART(A):

AIM:

Program for arranging data array in ascending order

APPRATUS:

Microprocessor Kit 8085 , Key Board, Op-Code Sheet.



THEORY:

In the programmer three counts have been used. Count 1 is stored in register B to check whether all the numbers of the series have been arranged in ascending order. Count 2 is stored in register C tocheck whether all the numbers of the series have been compared while selecting the smallest number of the series. After the selection of the smallest number it is to be ascertained which number was the smallest. At this stage register C is free . The register C is again used in SUBROUTINE-2 to store count 3 to check which number has been selected as the smallest number. The smallest number is replaced by FF in the series. Again the program to find smallest number is taken up to select the smallest number in the modified series.

SUBROUTINE-1 is to find the smallest number in a data array. SUBROUTINE-2 is to check which number is the smallest, and to replace it by FF.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don't touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data from 4500, specified in the example.
3. Execute the program and check for the result from 4601.
4. Change data from 4500 and execute each time and check for result.

PROGRAME & OBJECT CODES :

Memory Address	Opcodes	Mnemonics	Comments
4100	11, 01, 46	LXI D, 4601	Memory location to store result
4103	21, 00, 45	LXI H,4500	Count address in H-L pair.
4106	46	MOV B,M	Count in register B to check whether all number have been arranged in ascending order.
4107	CD, 00, 44	**START:** CALL 4400	Call Subroutine-! To find smallest number.
410A	12	STAX D	Store the result
410B	CD, 50, 42	CALL 4250	Call Subroutine-2 to check which number is smallest.
410E	13	INX D	
410F	05	DCR B	Have all numbers been arranged in ascending order?
4110	C2, 07, 41	JNZ **:START**	No, repeat process.
4113	76	HLT	Stop

SUBROUTINE-1: To Find the Smallest Number

Memory Address	Opcodes	Mnemonics	Comments
4400	21, 00, 45	LXI H, 4500	Count address in H-L pair.
4403	4E	MOV C,M	Count in register C.
4404	3E,FF	MVI A,FF	Get FF in accumulator.

4406	23	**LOOP:** INX H	
4407	BE	CMP M	Compare next number with previous smallest. Is next number< previous smallest ?
4408	CD, 0C,44	JC **:AHEAD**	No,smallest number is in the accumulator. Go to AHEAD.
440B	7E	MOV A,M	
440C	0D	**AHEAD:** DCR C	
441D	C2, 06, 44	JNZ **:LOOP**	
4410	C9	RET	

SUBROUTINE-2 :

Memory Address	**Opcodes**	**Mnemonics**	**Comments**
4250	21, 00, 45	LXI H, 4500	
4253	4E	MOV C,M	Count to check which number was smallest.
4254	23	**BEHIND:** INX H	Get next number.
4255	BE	CMP M	Compare the next number with smallest number which is in the accumulator.
4256	CA, X,X	JZ **:FORWARD**	Is the present number the smallest one? Yes, go to FORWARD.
4259	0D	DCR C	Decrement count.
425A	C2,X,X	JNZ **:BEHIND**	No, jump to take up next number.
425D	3E, FF	**FORWARD:** MVI A, FF	
425F	77	MOV M, A	Replace the smallest number by FF.
4260	C9	RET	

RESULT:

1): DATA	**RESULT**
4500- 05	4601-15
4501-E5	4602-96
4502-A9	4603-A9

4503-96	4604-B4
4504-B4	4605-E5
4505-15	

2): DATA	**RESULT**
4500- 06	4601-30
4501-45	4602-45
4502-B8	4603-98
4503-FF	4604-B8
4504-E8	4605-E8
4505-98	4606-FF
4506-30	

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.

2. Make space whenever you doing program on kit.

3. You should properly enter the program and exit through proper command.

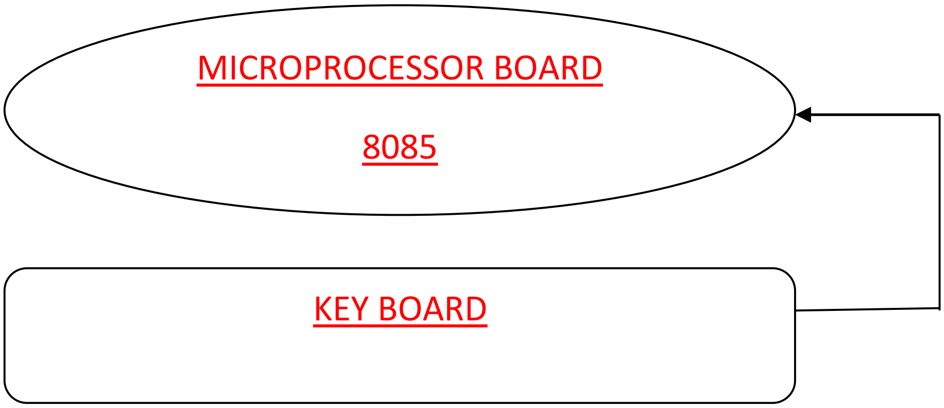
PART(B):

AIM:

Program for arranging data array in descending order

APPRATUS:

Microprocessor Kit 8085 , Key Board, Op-Code Sheet.



THEORY:

In the programmer three counts have been used. Count 1 is stored in register B to check whether all the numbers of the series have been arranged in descending order. Count 2 is stored in register C to check whether all the numbers of the series have been compared while selecting the highest number of the series. After the selection of the highest number it is to be ascertained which number was the smallest. At this stage register C is free. The register C is again used in SUBROUTINE-2 to store count 3 to check which number has been selected as the largest number. The largest number is replaced by 00 in the series. Again the program to find largest number is taken up to select the largest number in the modified series.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don't touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data from 4500, specified in the example.
3. Execute the program and check for the result from 4601.
4. Change data from 4500 and execute each time and check for result.

PROGRAME & OBJECT CODES :

Memory Address	Opcodes	Mnemonics	Comments
4100	21, 00, 45	LXI H, 4500	Address for count.
4103	4E	MOV C,M	Count for number of passes in register C.
4104	21, 00, 45	**BACK:** LXI H,4500	
4107	56	MOV D,M	Count for number of Comparison in register D.
4108	23	INX H	
4109	7E	MOV A,M	1st number in accumulator.
410A	23	**LOOP:** INX H	Address of next number.
410B	46	MOV B, M	Next number in register B.
410C	B8	CMP B	Compare next number with previous greatest number.
410D	D2, 16, 41	JNC **:AHEAD**	If previous greater number > next number, go to AHEAD.
4110	2B	DCX H	
4111	77	MOV M,A	Place smaller of the two compared numbers in memory
4112	78	MOV A,B	Place greater of the two numbers in accumulator.
4113	C3, 18, 41	JMP **:GO**	
4116	2B	**AHEAD:** DCX H	
4117	70	MOV M,B	Place smaller of the two compared numbers in memory.
4118	23	**GO:** INX H	
4119	15	DCR D	Decrease the count for comparisons.
411A	C2, 0A, 41	JNZ **:LOOP**	
411D	77	MOV M, A	Place the greatest number after a pass in the memory.
411E	0D	DCR C	Decreased the count for

			passes
411F	C2, 04, 41	JNZ **:BACK**	
4122	76	HLT	Stop

RESULT:

DATA	**RESULT**
4500- 04 (COUNT)	4601-15
4501-60	4602-25
4502-40	4603-40
4503-50	4604-50
4504-15	4605-60
4505-25	

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command.

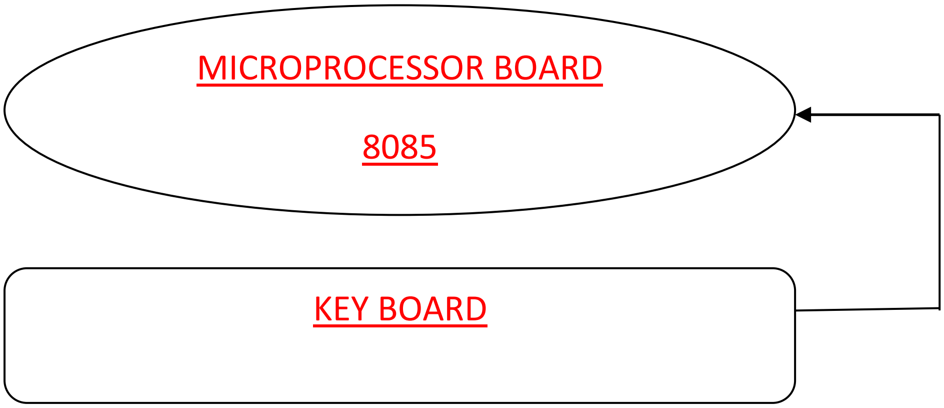
PART(C):

AIM:

Program for Find the square form look up table

APPRATUS:

Microprocessor Kit 8085 , Key Board, Op-Code Sheet.



THEORY:

The squares of data are stored in certain memory location in the tabular form. This is called look-up table. For this example the square of number from 00 to 99 are stored in memory locations 4600 to 4609 H. The values of squares are in decimals are decimal. The data from the index and it is transferred from memory to the accumulator and then to register L. It forms the LSBs of the memory location , where square of the data is placed. The MSBs of the address is moved to register H. Now the address of the desired memory location where of the square of the resides in H-L pair. The square of the data is now moved to the accumulator and then it is stored.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don't touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data from 4500, specified in the example.
3. Execute the program and check for the result from 4601.
4. Change data from 4500 and execute each time and check for result.

PROGRAME & OBJECT CODES :

Memory Address	Opcodes	Mnemonics	Comments
4100	21, 00, 45	LDA 4500	Get data in accumulator.
4103	4E	MOV L, A	Get data in register L.
4104	21, 00, 45	MVI H, 46	Get 46 in register H.
4107	56	MOV A, M	Square of data in accumulator.
4108	23	STA 4501	Store square in 4501
4109	7E	HLT	Stop

RESULT:

DATA	**RESULT**
4500- 07 D	4601-49 D

Look –up Table:

Address	*Square*
(Hex)	*(Decimal)*
4600-	00
4601	01
4602	04
4603	09
4604	16
4605	25
4606	36
4607	49
4608	64
4609	81

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command.

EXPERIMENT NO. 11

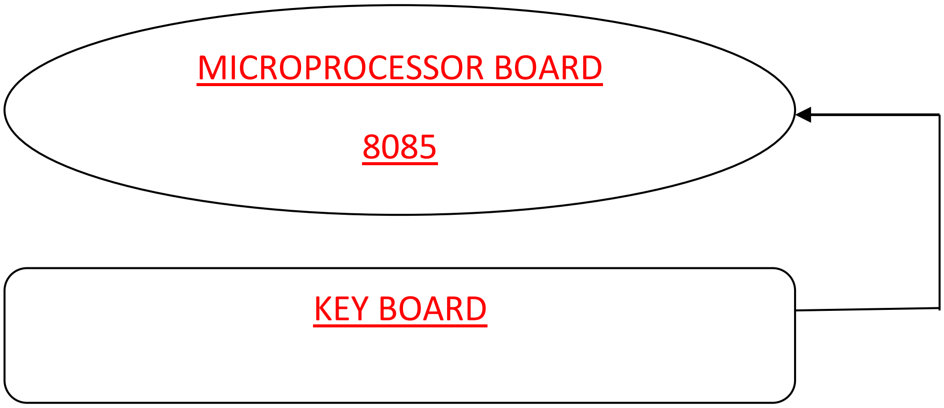
PART(A):

AIM:

Program for ASCII to DECIMAL Conversion.

APPRATUS:

Microprocessor Kit 8085 , Key Board, Op-Code Sheet.



THEORY:

Conversion of an ASCII number to decimal is very simple because all the decimal numbers form sequence in ASCII. Any ASCII number can be converted to decimal just by substracting 30 from it.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don’t touch the live wire

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data from 4150, specified in the example.
3. Execute the program and check for the result from 4151.
4. Change data from 4150 and execute each time and check for result.

PROGRAME & OBJECT CODES :

Memory Address	Opcodes	Mnemonics	Comments
4100	21, 50,41	LXI H,4150	Point to data
4103	7E	MOV A,M	Get operand
4104	DE, 30	SUI 30	Convert to Decimal.
4106	FE, 0A	CPI 0A	Check it is a valid Decimal number.
4108	DA, 0D, 41	JC :LOOP	Yes, Store Result.
410B	3E,FF	MVI A,FF	No , make result = FF.
410D	23	LOOP: INX H	Increment in H-L pair
410E	77	MOV M,A	Move from Accu. To Memory.
410F	76	HLT	Stop

RESULT:

DATA	RESULT
4150- 35	4151- 05

DATA	RESULT
4150- 3A	4151- FF

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command.

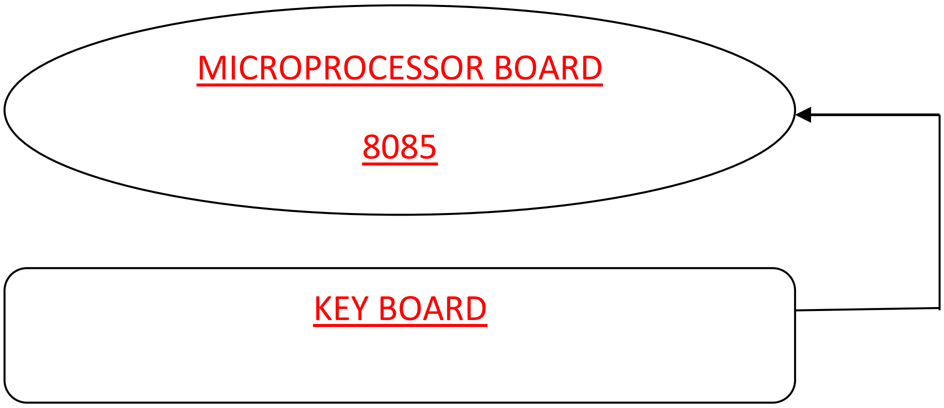
PART(B):

AIM:

Program for BCD to HEX Conversion.

APPRATUS:

Microprocessor Kit 8085 , Key Board, Op-Code Sheet.



THEORY:

Out of Two BCD digits at 4150 and 4151, the one at 4150 is the MSD. The logic is to multiply the MSD by ten using repeated addition. Then add the LSD to it.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don’t touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data from 4150 & 4151 specified in the example.
3. Execute the program and check for the result from 4152.
4. Change data from 4150 & 4151 and execute each time and check for result.

PROGRAME & OBJECT CODES:

Memory Address	Opcodes	Mnemonics	Comments
4100	21, 50,41	LXI H,4150	Point to data
4103	7E	MOV A,M	Initialize memory pointer to 4150 (A) = (4150)- MSD
4104	87	ADD A	MSD X 2.
4105	47	MOV B,A	Save MSD X 2
4106	87	ADD A	MSD X 4
4107	87	ADD A	MSD x 8
4108	80	ADD B	MSD X 10
4109	23	INX H	Point to LSD.
410A	86	ADD M	Add to form hex equivalent.
410B	23	INX H	(A) => (4152)
410C	77	MOV M,A	Move from Acc. to memory
410D	76	HLT	Stop.

RESULT:

DATA	RESULT
4150 - 02 (MSD)	4152 – 1D H = 29 D
4151 - 09 (LSD)	

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command

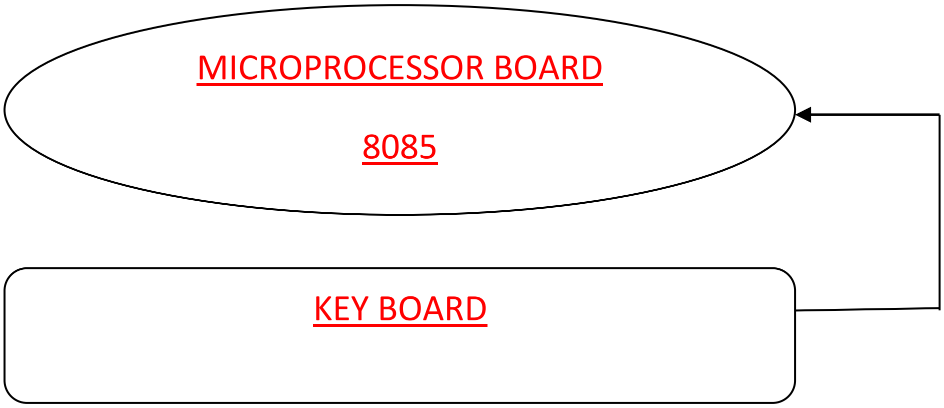
PART(C):

AIM:

Program for HEX to DECIMAL Conversion.

APPRATUS:

Microprocessor Kit 8085, Key Board, Op-Code Sheet.



THEORY:

In this experiment the hex number is converted into its equivalent decimal number using the following logic. First count the number of hundreds, the numbers of tens and units present in that hex number. Using these three, add up to get the equivalent decimal number.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don't touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data from 4150, specified in the example.
3. Execute the program and check for the result from 4151 and 4152.
4. Change data from 4150 and execute each time and check for result.

PROGRAME & OBJECT CODES :

Memory Address	Opcodes	Mnemonics	Comments
4100	21, 50,41	LXI H,4150	Point to data
4103	01, 00, 00	LXI B,0000	Initialize Hundreds= 0,Tens=0.
4106	7E	MOV A,M	Get hex data to A.
4107	D6, 64	LOOP: SUI 64	
4109	DA, 10, 41	JC :LOOP1	
410C	04	INR B	Hundreds = Hundreds +1.
410D	C3, 07, 41	JMP :LOOP	
4110	C6, 64	LOOP1: ADI 64	If subtracted extra, add it clear carry flag.
4112	D6, 0A	LOOP2: SUI 0A	.
4114	DA, 1B, 41	JC	
4117	0C	INR C	Tens = Tens +1
4118	C3, 12, 41	JMP :LOOP2	
411B	C6, 0A	LOOP3: ADI 0A	If subtracted extra, add it again.
411D	23	INX H	A= Units.
411E	70	MOV M,B	Store Hundreds.
411F	47	MOV B,A	Combine Tens in C and
4120	79	MOV A,C	Units in A to form a
4121	07	RLC	Single 8-bit number.
4122	07	RLC	
4123	07	RLC	
4124	07	RLC	
4125	80	ADD B	
4126	23	INX H	
4127	77	MOV M,A	Store Tens and units.
4128	76	HLT	stop

RESULT:

DATA	RESULT
4150 – A9	4151 – 01
	4152 – 69

A9 H = 169 D

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.

2. Make space whenever you doing program on kit.

3. You should properly enter the program and exit through proper command

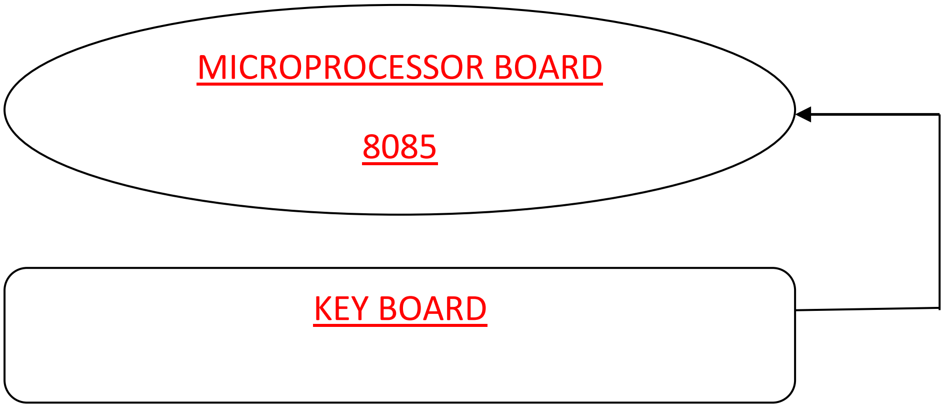
PART(D):

AIM:

Program for HEX to BINARY Conversion.

APPRATUS:

Microprocessor Kit 8085, Key Board, Op-Code Sheet.



THEORY:

First get the data and rotate it right. Depending upon carry store either 0 or 1 in memory. Do the rotation 8 times for the 8 bits in that number.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don't touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data from 4150, specified in the example.
3. Execute the program and check for the result from 4150 to 4157.
4. Change data from 4150 and execute each time and check for result.

PROGRAME & OBJECT CODES :

Memory Address	Opcodes	Mnemonics	Comments
4100	21, 50,41	LXI H,4150	Point to data
4103	06, 08	MVI B,08	Counter for 8- bit.
4105	3E, 5A	MVI A,5A	
4107	0F	LOOP: RRC	Check least significant bit.
4108	DA, 10, 41	JC :LOOP1	
410B	36, 00	MVI M,00	Store zero if no carry
410D	C3, 12, 41	JMP :NEXT	
4110	36, 01	LOOP1: MVI M,01	Store one if there is a carry.
4112	23	NEXT: INX H	
4113	05	DCR B	Check for counter.
4114	C2, 07, 41	JNZ: LOOP	
4117	76	HLT	Stop

RESULT:

DATA	RESULT
4150 – 5A = 0101 1010 B	4150 – 0
	4151 – 1
	4152 - 0
	4153 - 1
	4154 - 1
	4155 - 0
	4156 - 1
	4157 - 0

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command

"SECTION B"

EXPERIMENT NO. 01

PART (A):

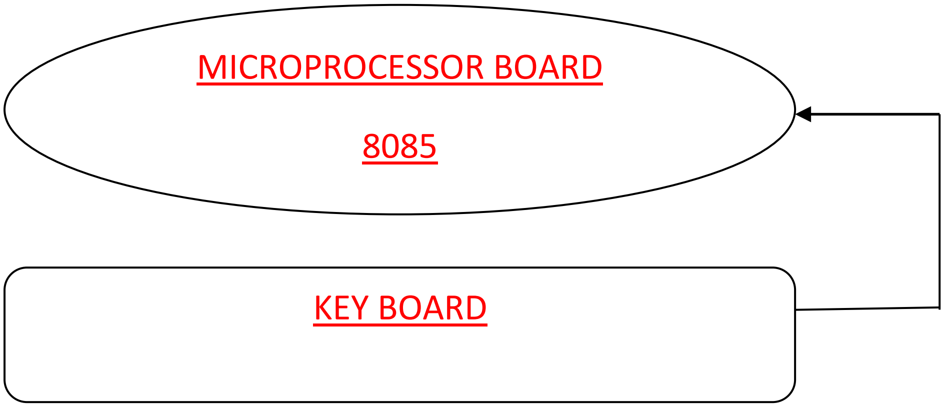
AIM:

Interfacing of 8085 Microprocessor with DAC 0800 to generate Square wave form.

.

APPRATUS:

Microprocessor Kit 8085, DAC 0800 card, Key Board, Op-Code Sheet.



THEORY:

The basic idea behind the generation of waveforms is the continuous generation of analog output of DAC.

With 00(Hex) as input to DAC2, the analog output is -5V.Similarly with FF(Hex) as input, the output is +5V.Outputting digital data 00 and FF at regular intervals, to DAC2, results I a square wave of amplitude 15 volts.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don't touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data from 4100, specified in the example.
3. Execute the program and check for the result on CRO.

PROGRAME & OBJECT CODES :

Memory Address	Opcodes	Mnemonics
4100	3E, 00	START: MVI A,00
4102	D3, C8	OUT 0C8
4104	CD, 11, 41	CALL DELAY
4107	3E, FF	MVI A,0FF
4109	D3, C8	OUT 0C8
410B	CD, 11, 41	CALL DELAY
410E	C3, 00, 41	JMP START
4111	06, 05	DELAY: MVI B,05
4103	0E, FF	L1: MVI C,0FF
4115	0D	L2: DCR C
4116	C2, 14, 41	JNZ L2
4119	05	DCR B
411A	C2, 13, 41	JNZ L1
411D	C9	RET

RESULT:

Execute the program and using CRO, verify that the waveform at the DAC2 output is a Square wave. Modify the frequency of the Square wave, by varying the time delay.

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command

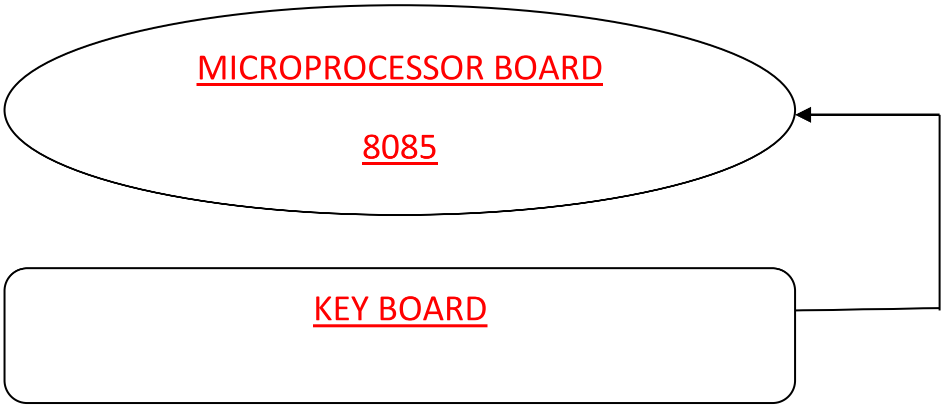
PART (B):

AIM:

Interfacing of 8085 Microprocessor with DAC 0800 to generate Triangular wave form

APPRATUS:

Microprocessor Kit 8085, DAC 0800 card, Key Board, Op-Code Sheet.



THEORY:

The digital to analog con be classified as current output, voltage output and multiplying type. The current output DAC provides current as the output signal. The voltage output DAC internally convert current signal into voltage signal. The voltage output DAC is slower than the current output DAC because of the delay in converting the current signal into the voltage signal.
The following Program will generate triangular wave at DAC2 output. The program is self explanatory.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don't touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data from 4100, specified in the example.
3. Execute the program and check for the result on CRO.

PROGRAME & OBJECT CODES :

Memory Address	Opcodes	Mnemonics
4100	2E, 00	START: MVI L,00
4102	7D	L1 : MOV A,L
4103	D3, C8	OUT OC8
4105	2C	INR L
4106	C2, 02, 41	JNZ L1
4109	2E, FF	MVI L,0FF
410B	7D	L2: MOV A,L
411C	D3, C8	OUT 0C8
410E	2D	DCR L
411F	C2, 0B, 41	JNZ L2
4112	C3, 00, 41	JMP START

RESULT:

Execute the program and using CRO, verify that the waveform at the DAC2 output is a Square wave. Modify the frequency of the Square wave, by varying the time delay.

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command

PART (C):

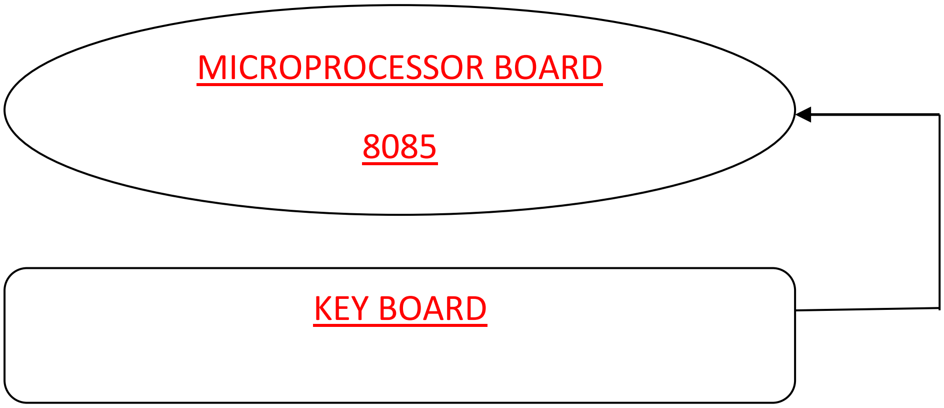
AIM:

Interfacing of 8085 Microprocessor with DAC 0800 to Saw-tooth wave form.

.

APPRATUS:

Microprocessor Kit 8085, DAC 0800 card, Key Board, Op-Code Sheet.



THEORY:

Output digital data from 00 to FF in constant steps of 01 to DAC1. Repeat this sequence again and again. As a result a Saw-tooth wave will be generated at DAC1 output.

The digital to analog con be classified as current output, voltage output and multiplying type. The current output DAC provides current as the output signal. The voltage output DAC internally convert current signal into voltage signal. The voltage output DAC is slower than the current output DAC because of the delay in converting the current signal into the voltage signal.

PRECAUTIONS:

1. All steps should be followed carefully.

2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don't touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data from 4100, specified in the example.
3. Execute the program and check for the result on CRO.

PROGRAME & OBJECT CODES :

Memory Address	Opcodes	Mnemonics
4100	3E,00	START: MVI A,00
4102	D3, C0	L1: OUT 0C0
4103	3C	INR A
4105	C2,02, 41	JNZ L1
4106	C3, 00, 41	JMP START

RESULT:

Execute the program and using CRO, verify that the waveform at the DAC2 output is a Square wave. Modify the frequency of the Square wave, by varying the time delay.

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command

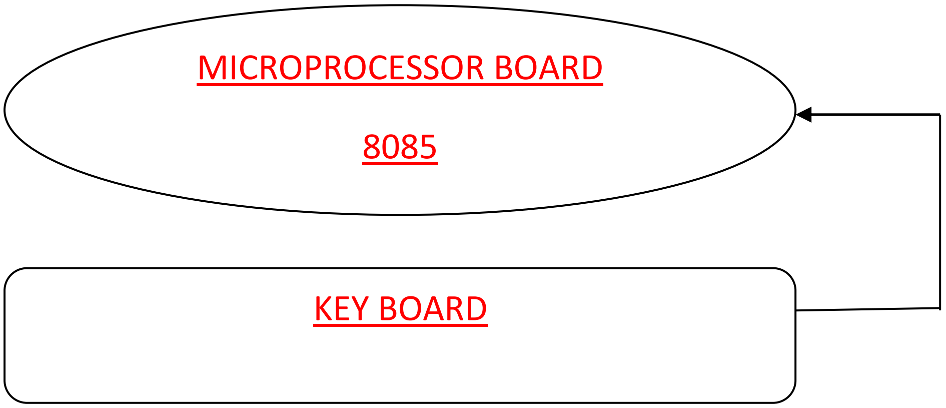
EXPERIMENT NO. 02

AIM:

Keyboard Interfacing 8279 IC with 8255 PPI IC and to display rolling message “HELP US” in the display.

APPRATUS:

Microprocessor Kit 8085, 8279 interfacing card, Key Board, Op-Code Sheet.



THEORY:

The data is fetched from address 412CH and displayed in the first digit of the display. The next data is displayed in the second digit of the display, since in the command word for ‘write display RAM’ the auto increment flag is set. A time delay is given between successive digits for a lively display.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don’t touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data from 4100, specified in the example.
3. Execute the program and check for the result on CRO.

PROGRAME & OBJECT CODES:

```
ORG          4100H
CNT          EQU          0C2H
DAT          EQU          0C0H
POINTER     EQU          412CH
```

Memory Address	Opcodes	Mnemonics
4100	21, 2C, 41	START: LXI H,POINTER
4103	16,0F	MVI D,0F
4105	3E, 10	MVI A,10
4107	D3, C2	OUT CNT
4109	3E, CC	MVI A,0CC
410B	D3, C2	OUT CNT
410D	3E, 90	MVI A,90
410F	D3, C2	OUT CNT
4111	7E	LOP: MOV A,M
4112	D3, C0	OUT DAT
4114	CD, 1F, 41	CALL DELAY
4117	23	INX H
4118	15	DCR D
4119	C2, 11, 41	JNZ LOP
411C	C3, 00, 41	JMP START
411F	06, A0	DELAY: MVI B,0A0
4121	0E, FF	LOP1: MVI C,0FF
4123	0D	LOP2: DCR C
4124	C2, 23, 41	JNZ LOP2
4127	05	DCR B
4128	C2, 21, 41	JNZ LOP1
412B	C9	RET
412C	FF, FF, FF, FF	
4130	FF, FF, FF, FF	
4134	98, 68, 7C, C8	

4138	1C, 29, FF, FF	

RESULT:

Execute the program and check the message on display.

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command

EXPERIMENT NO. 03

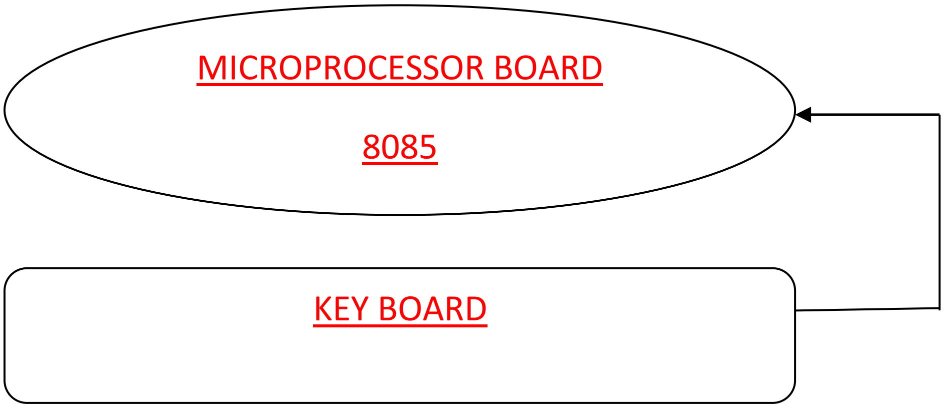
PART (A):

AIM:

Stepper motor interfacing with 8085 microprocessor, Movement of motor clockwise and anticlockwise

APPRATUS:

Microprocessor Kit 8085, stepper motor interfacing card, Key Board, Op-Code Sheet.



THEORY:

Stepper motor control is a very popular application of microprocessor in control area.They are widly in (simple position control system in the open and closed loop mode) a variety of applications such as computer peripherals (printers, disk driver etc) and in the area of process control machine tools, medicine, numerically controlled machine and robotics.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don’t touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data from 4100, specified in the example.
3. Execute the program and check for the result .

PROGRAME & OBJECT CODES:

Memory Address	Opcodes	Mnemonics
4100	21, 1A, 41	START: LXI H,LOOK UP
4103	06, 04	MVI B,04
4105	7E	REPT: MOV A,M
4106	D3, C0	OUT OCOH
4108	11,03,03	LXI D, 0303H
410B	00	DELAY: NOP
410C	1B	DCX D
410D	7B	LOP: MOV A,E
410E	B2	ORA D
410F	C2, 0B, 41	JNZ :DELAY
4112	23	INX H
4113	05	DCR B
4114	C2, 05, 41	JNZ :REPT
4117	C3, 00, 41	JMP : START
411A	09, 05, 06, 0A	LOOK UP: DB 09 05 06 0A

RESULT:

Enter the above program starting from location 4100. Execute the same. The stepper motor rotates. Speed can be varied by varying the count at DE pair. Direction can be varied by entering the data in the look-up table in the reverse order.

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command

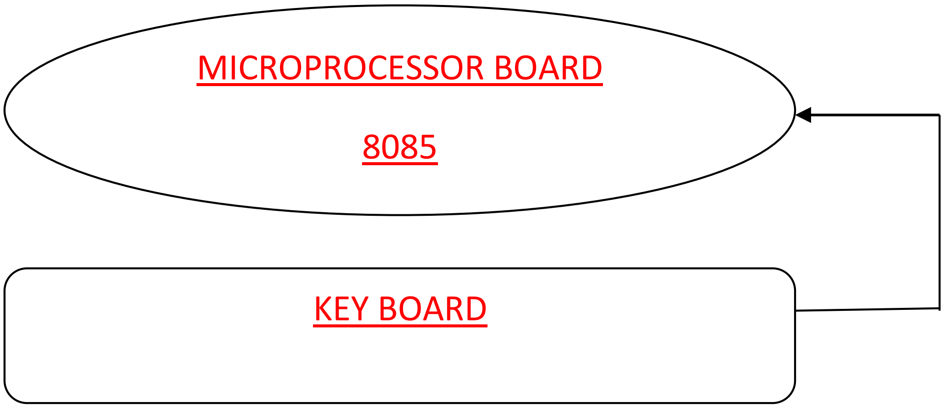
PART (B):

AIM:

To run a stepper motor for required angle within 360^{o}c which is equivalent to 256 steps

APPRATUS:

Microprocessor Kit 8085, stepper motor interfacing card, Key Board, Op-Code Sheet.



THEORY:

Stepper motor control is a very popular application of microprocessor in control area.They are widly in (simple position control system in the open and closed loop mode) a variety of applications such as computer peripherals (printers, disk driver etc) and in the area of process control machine tools, medicine, numerically controlled machine and robotics.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don't touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data from 4100, specified in the example.
3. Execute the program and check for the result .

PROGRAME & OBJECT CODES:

Memory Address	Opcodes	Mnemonics
4100	0E HEX DATA	MVI C,HEX DATA
4102	21, 20, 41	START: LXI H, LOOK UP
4105	06,04	MVI B,04
4107	7E	REPT: MOV A,M
4108	D3, C0	OUT CO
410A	0D	DCR C
410B	CA, 24, 41	JZ END
410E	11, 03, 03	LXI D, COUNT
4111	00	DELAY: NOP
4112	1B	DCX D
4113	7B	MOV A,E
4114	B2	ORA D
4115	C2, 11, 41	JNZ DELAY
4118	23	INX H
4119	05	DCR B
411A	C2, 07, 41	JNZ REPT
411D	C2, 02, 41	JUMP START
4120	09, 05, 06, 0A	LOOK UP: DB 09 05 06 0A
4124	76	END: HLT

RESULT:

Enter the above program and execute it. By converting the required step in decimal to hex and entering the hex data at 4101 the motor rotates for so much steps and then stops.

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command

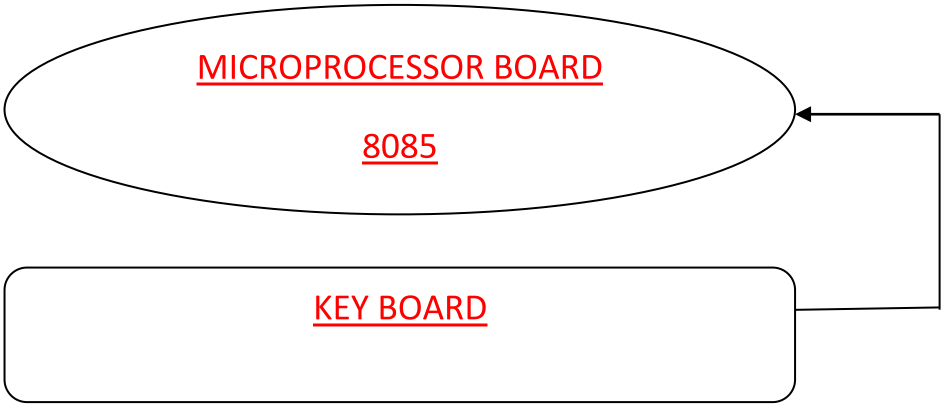
EXPERIMENT NO. 04

AIM:

To write a program to control temperature for a given set point using ON_OFF controller unit.

APPRATUS:

Microprocessor Kit 8085, DAC interfacing card, Key Board, Op-Code Sheet.



THEORY:

Connect power connector of ITB-005 to connector P3 of VBMB-002. Connect the output of ITB 005 at P2to ADC input (Pin no.3 of P2) in VBMB-002. Execute the following program. We are using simple counter method, to convert Analog signal to Digital signal.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don't touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data from 4100, specified in the example.
3. Execute the program and check for the result.

PROGRAME & OBJECT CODES:

```
ORG         4100H
DAC1        EQU         OCOH
OUTPUT   EQU         0DOH
```

Memory Address	Opcodes	Mnemonics
4100	3E, FF	INIT: MVI A,FF
4102	D3, C8	OUT C8; OUT VOLT TO DAC2
4104	3E, 00	START: MVI A,00
4106	47	REPEAT : MOV B,A
4107	D3, C0	OUT CO
4109	DB, DO	IN DO ;CHECK COMPARATOR OUTPUT
410B	E6, 01	ANI 01
410D	C2, 15, 41	JNZ FINAL
4110	78	MOV A,B
4111	3C	INR A
4112	C3, 06, 41	JMP REPEAT
4115	78	FINAL: MOV A,B
4116	32, 00, 50	STA 5000
4119	E6, F0	ANI OF
411B	32, 02, 50	STA 5002
411E	78	MOV A,B
411F	E6, F0	ANI FO
4121	0F	RRC
4122	0F	RRC
4123	0F	RRC
4124	0F	RRC
4125	32, 01, 50	STA 5001
4128	21, 01, 50	LXI H.5001
412B	3E, 03	MVI A,03
412D	0E, 08	MVI C,08
412F	CD, 05, 00	CALL 0005
4132	3A, 00, 50	LDA 5000
4135	FE, C0	CPI CO ;CONTROL DATA CHECK
4137	DA, 00, 41	JC INIT
413A	3E, 7F	MVI A, 7F
413C	D3, C8	OUT C8

413E	C3, 04, 41	JMP START
		END

RESULT:

For the digital data 80 to FF equals to temperature 0 to 100 degree. Since Digital to analog convertor is Bipolar one, We can add 80 (hex decimal value) for getting data corresponds to temperature.

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command

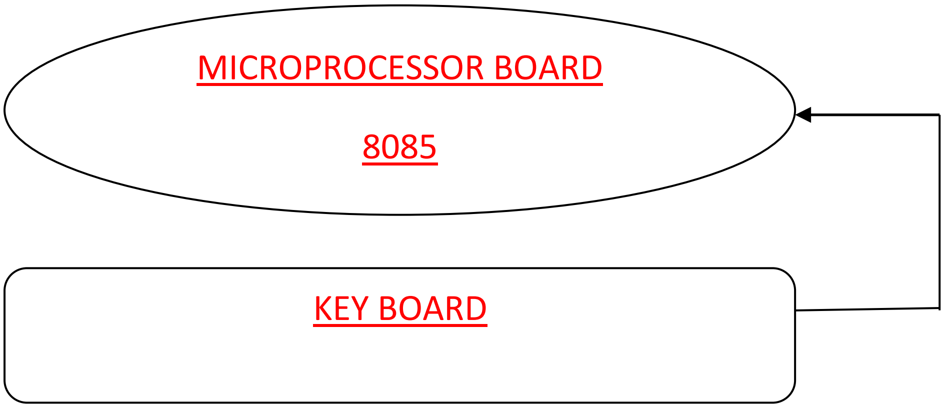
EXPERIMENT NO. 05

AIM:

Interfacing 8085 microprocessor with 0809 ADC

APPRATUS:

Microprocessor Kit 8085, 0809 ADC interfacing card, Key Board, Op-Code Sheet.



THEORY:

There are numerous way in which an analog signal can be converted to digital form. The Analog-to-Digital convertors can classified into two general group based on the conversion technique.

The Most commonly used A/D convertor are successive approximation and integration type convertors. The successive approximation A/D convertors are used in applications such as data loggers and instrumentation, where conversion speed is important. On the other hand, Integration type convertors are used in application such as digital meters, panel meters and monitoring system, where the conversion accuracy is critical.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don’t touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Place jumper J2 in A position.
2. Place jumper J5 in A position.
3. Enter and execute the program.
4. Vary the analog input (using trim pot) and view the corresponding digital value in the LED display.

PROGRAME & OBJECT CODES:

ORG 4100H
DAC1 EQU OCOH
OUTPUT EQU 0DOH

Memory Address	Opcodes	Mnemonics
4100	3E, 10	MVI A, 10 ;SELECT CHANNEL 0 AND MAKE
4102	D3, C8	OUT 0C8 ;ALE LOW
4104	3E, 18	MVI A, 18 ; MAKE ALE HIGH
4106	D3, C8	OUT 0C8
4108	3E, 01	MVI A, 01 ;SOC SIGNAL HIGHG
410A	D3, D0	OUT 0D0
410C	AF	XRA A
410D	AF	XRA A
410E	AF	XRA A
410F	3E, 00	MVI A, 00 ;SOC SIGNAL LOW
4111	D3, D0	OUT 0D0
4113	76	HLT

RESULT:

Initiating the analog to digital conversion process by means of software is sometime necessary. Execute the program given, which converts the analog input at channel 0 and displays the output with the LEDs.

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command

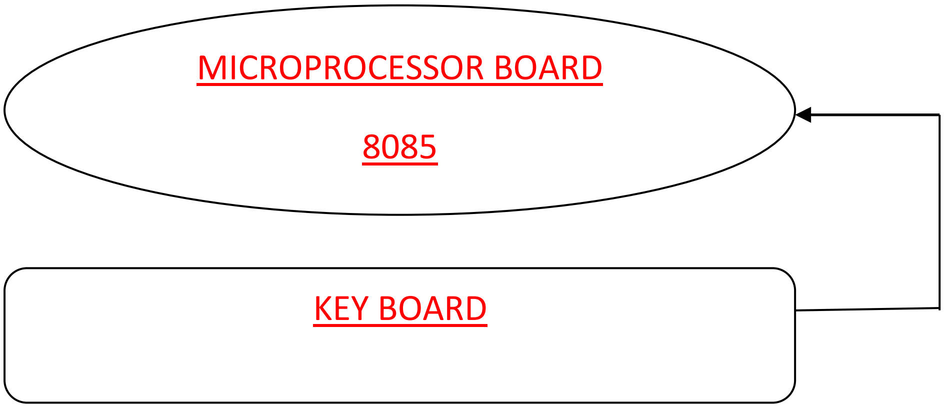
EXPERIMENT NO. 06

AIM:

Case study on Microprocessor application with Protective relay.

APPRATUS:

Microprocessor Kit 8085, 0809 ADC interfacing card, Key Board, Op-Code Sheet.



THEORY:

The Microprocessor increases flexibility and reduces the number of relay units. Fault current and voltage contain harmonics which are filtered by analog filter before they are fed to the relay. The analog filter is band pass filters. They may be designed to pass 40-70 Hz signals. At low frequency the analog filters are slow. The research is going on to develop faster relays using digital filter technique.

Over current relay are widely used for the protection of distribution lines, large industrial motors and equipment etc.The microprocessor being very fast can sense a number of circuits, using multiplexer, and send a tripping signal to the circuit breaker of the faulty circuit.

There may be a fat tripping of an over current relay due to transients. To solve this problem the program can be slightly modified as shown below. If the faulty current exceeds the pick-up value, the microprocessor measure the faulty current after small delay once again to confirm whether it is a fault current or transient and the microprocessor will send a trip signal to disconnect the faulty part of the system.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don't touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data from 4110, specified in the example.
3. Execute the program and check for the result.

PROGRAME & OBJECT CODES:

Memory Address	Opcodes	Mnemonics
4110	3E, 98	MVI A, 98 ; Initialize ports
4112	D3, 03	OUT 03
4114	3E, 00	LOOP; MVI A,00 ;Signal to multiplexer to switch on S1
4116	D3, 02	OUT 02
4118	3E, 08	MVI A, 08 ; Start of conversion pulse.
411A	D3, 02	OUT 02
411C	3E, 00	MVI A,00
411E	D3, 02	OUT 02
4120	DB, 02	READ; IN 02 ;Check whether conversion is over
4122	17	RAL
4123	D2, 20, 41	JNC READ
4126	DB, 00	IN 00 ; Read Idc.
4128	2F	CMA
4129	D6, 80	SUI 80
412B	32, 50, FD	STA 4500 ; Store Idc. To check program
412E	21, 50, 45	LXI H, 4550
4131	46	MOV B,M ; Count for look-up table for register B
4132	23	SEARCH; INX H
4133	BE	CMP M
4134	D2, 3E, 41	JNC FORWARD
4137	05	DCR B
4138	C2, 32, 41	JNZ SEARCH
413B	CA, 14,41	JZ LOOP
413E	24	FORWARD; INR H
413F	46	MOV B,M ; Count for delay
4140	0E, FF	BACK ; MVI C,FF
4142	16, FF	CHUNK; MVI D,FF
4144	15	GO; DCR D
4145	C2, 44, 41	JNZ GO

4148	0D	DCR C
4149	C2, 42, 41	JNZ CHUNK
414C	05	DCR B
414D	C2, 40, 41	JNZ BACK
4150	3E, 01	MVI A, 01
4152	D3, 01	OUT 01 ; Trip signal.
4154	76	HLT

LOOK-UP TABLE:

MEMORY		DIGITAL VALUE
ADDRESS		FOR DIFFERENT
4550	---- ------------	0C (COUNT)
4551	-----------------	7F
4552	-----------------	7A
4553	-----------------	6D
4554	-----------------	66
4555	-----------------	60
4556	-----------------	5A
4557	---------------- -	53
4558	-------------- --	4D
4559	----------------	46
455A	----------------	40
455B	----------------	3A
455C	-----------------	33

RESULT:

Microprocessor application with Protective relay has been done through the kit.

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.

2. Make space whenever you doing program on kit.

3. You should properly enter the program and exit through proper command

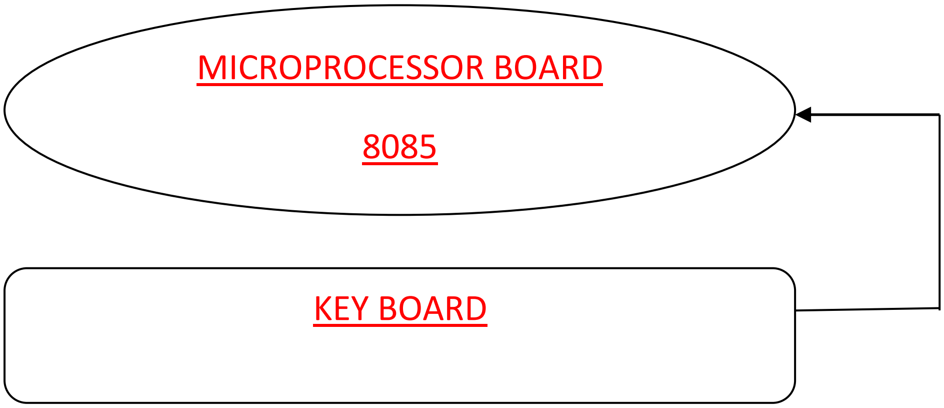
EXPERIMENT NO. 07

AIM:

Interfacing of 8085 Microprocessor with seven segment display/ LED interfacing.

APPRATUS:

Microprocessor Kit 8085, 0809 ADC interfacing card, Key Board, Op-Code Sheet.



THEORY:

The Seven- Segment LED display is a multiple display. It can display all decimal digits and some letters. It is very popular among multiple displays as it has the smallest number of separately controlled light emitting diode (LED). Multiple displays of 9-segment LED,14-segmant LED and dot matrix type are available. These displays give better representation of alphanumeric characters but require complex circuitry.

In seven-segment displays there are seven light emitting diode (LED) in which each LED can be controlled separately. To display a digit or letter the desired segment are made ON, there are two type of 7- segment display namely- common anode type and common cathode type.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don’t touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data from 4100, specified in the example.
3. Execute the program and check for the result.

PROGRAME & OBJECT CODES:

Display of Decimal number 0 to 9

Memory Address	Opcodes	Mnemonics
4100	3E, 98	MVI A,98 ;Get control word.
4102	D3, 03	OUT 03 ; Initialize ports.
4104	21, 00, 45	ABOVE ; LXI H,4500 ; Get count
4107	5E	MOV E,M ;Count in register E.
4108	23	LOOP; INX H
4109	7E	MOVA,M ; get next number
410A	D3, 01	OUT 01 ; Output at Port B
410C	06, 0F	MVI B,0F
410E	0E, FF	BEHIND; MVI C,FF
4110	16, FF	BACK ; MVI D,FF
4112	15	GO ; DCR D ;Delay subroutine using 3 registers
4113	C2, 12, 41	JNZ GO
4116	0D	DCR C
4117	C2, 10, 41	JNZ BACK
411A	05	DCR B
411B	C2, 0E, 41	JNZ BEHIND
411E	1D	DCR E ; Decrement count
411F	C2, 08, 41	JNZ LOOP ;Go to LOOP to get next number
4122	C3, 04, 41	JMP ABOVE ; Go to ABOVE to re-start

DATA :

4550	-------- 0A
4551	-------- 00
4552	-------- 01
4553	-------- 02
4554	-------- 03
4555	-------- 04
4556	-------- 05

4557 --------- 06
4558 --------- 07
4559 --------- 08
455A --------- 09

RESULT:

The 7-Segment display will display the decimal number 0 to 9 one by one. If the program is executed using two displaying units the 1st unit displays 0 to 9 and the 2nd unit displays always 0, i.e. MSBs of the numbers. MSBs for all the numbers are 0.If we want that the 2nd unit should also display 0 to 9, the data be fed as above.

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command
.

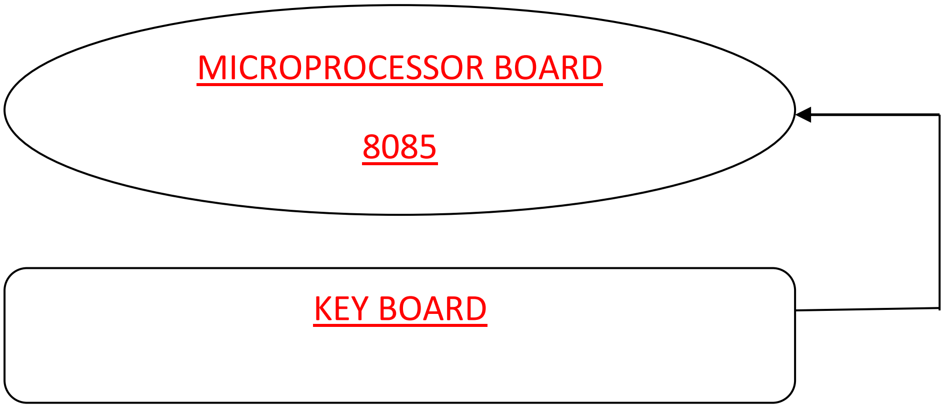
EXPERIMENT NO. 08

AIM:

Case study on Microprocessor application with Traffic Light controller.

APPRATUS:

Microprocessor Kit 8085, 8255 IC, key Board, Op-Code Sheet.



THEORY:

All ports of 8255 have been programmed as output ports. The control word to make all ports output ports in mode 0 operations is 80. The connection of pins of the ports to LED has been made through buffers (7407). Positive logic has been used to switch on LEDs. Three types LEDs have been red, yellow, and green. Green light glows to allow crossing yellow to make alert, and red does not allow crossing.

PRECAUTIONS:

1. All steps should be followed carefully.
2. Make sure all power sources are disconnected
3. Make sure you are properly grounded.
4. Don't touch the live wire.

EXPERIMENTAL PROCEDURE: STEP WISE –

1. Key in the Opcodes from the address specified.
2. Enter data from 4100, specified in the example.
3. Execute the program and check for the result.

PROGRAME & OBJECT CODES:

Memory Address	Opcodes	Mnemonics
4100	3E, 80	MVI A, 80 ; Get control word for 8255
4102	D3, 0B	OUT 0B ; Initialize ports of 8255.2.
4104	3E, 01	LOOP; MVI A,01
4106	D3, 09	OUT 09 ; Red ON for south
4108	D3, 08	OUT 08 ; Red ON for north
410A	3E, 44	MVI A, 44 ; Green ON for east & north
410C	D3, 04	OUT 0A
410E	CD, 00, 45	CALL DELAY 1
4111	3E, 22	MVI A,22 ; Yellow ON for east and west
4113	D3, 0A	OUT 0A
4115	3E, 02	MVI A,02
4117	D3, 09	OUT 09 ; Yellow ON for south
4119	D3, 08	OUT 08 ; Yellow ON for north
411B	CD, 13, 45	CALL DELAY II ; Red ON for east & west
411E	3E, 11	MVI A,11
4120	D3, 0A	OUT 0A
4122	3E, 04	MVI A, 04
4124	D3, 08	OUT 08 ; Green ON for north
4126	D3, 09	OUT 09 ; Green ON for south
4128	CD, 00, 45	CALL DELAY I
412B	3E, 22	MVI A,22 ; Yellow ON for east & west
412D	D3, 0A	OUT 0A
412F	3E, 02	MVI A,02
4131	D3, 09	OUT 09 ; Yellow ON for south
4133	D3, 08	OUT 08 ; Yellow ON for north
4135	CD, 13, 45	CALL DELAY II
4138	C3, 04, 41	JMP LOOP
DELAY I		
4500	06, 20	MVI B,20
4502	0E, FF	G03; MVI C,FF

4504	16, FF	G02; MVI C,FF
4506	15	G01; DCR D
4507	C2, 06, 45	JNZ G01
450A	0D	DCR C
450B	C2, 04, 45	JNZ G02
450E	05	DCR B
450F	C2, 02, 45	JNZ G03
4512	C9	RET
DELAY II		
4513	06,10	MVI B, 10
4515	C3, 02, 45	JMP G03

RESULT:

Three types LEDs have been red, yellow, and green. Green light glows to allow crossing yellow to make alert, and red does not allow crossing. Green light for right turns has not been shown. One can add some more LEDs for this purpose connect them to ports and make addition to the program. The program will become a longer one. Once the user understands the circuit and program illustrated in this, the further extension is very easy.

SOURCES OF ERROR:

1. Select op-codes carefully whenever you doing programming.
2. Make space whenever you doing program on kit.
3. You should properly enter the program and exit through proper command

SHORT QUESTIONS

1. What is an Instruction?
2. What is meant by Instruction Set?
3. In how many categories the instruction of 8085 be classified?
4. What is meant by ‘addressing mode’ different mode? Mention the different addressing mode?
5. Mention the Interrupts pin of 8085?
6. Explain the makeable and Non-makeable Interrupt?
7. What is meant by priority of interrupts?
8. What are the different types of data transfer operations possible?
9. Mention the different type of operations with asthmatic, logical, branch & machine control operation?
10. What are the different instruction word sizes in 8085?
11. What is the technology used in Microprocessor?
12. What is the function of Flag , Register, Program counter and Stack pointer?
13. What are the meant by Low level and High level language?
14. What is machine level programming?
15. What jobs ALU of 8085 can perform?
16. How many hardware interrupts 8085 supports. ?
17. How many I/P ports can 8085 access?
18. Describe the accumulator register of 8085?
19. What is the function of RESET IN and RESET OUT function?
20. What is another name of microprocessor?
21. What is bus?
22. What are the difference buses and what jobs they do in microprocessor?
23. Why are the different buses buffered?
24. Explain the two type of software?
25. Draw the software hierarchy of a microprocessor?
26. What is an editor?
27. What is an OS (Operating System) and what is its function?
28. What is a locator?
29. What is the difference assembly language used for 8085 microprocessor?
30. What is a coprocessor?
31. Draw a typical coprocessor configuration and discuss the same.
32. What is a coprocessor trap?
33. Explain the three fields contained in a coprocessor instruction.
34. What is a debugger?
35. How does a debugger help in debugging a program?
36. What is meant by the term ‘long word’?
37. Distinguish between KB, MB, GB, TB and PB.
38. Compare signed magnitude number and complementary numbers.
39. In how many groups can the signals of 8085 be classified?
40. Draw the architecture of 8085 and mention its various function blocks.
41. What is the technology used in the manufacture of 8085?
42. What is meant by the statement that 8085 is an 8-bit microprocessor?

43. What is the operating frequency of 8085?
44. Draw the block diagram of the built-in clock generator 8085.
45. What are the widths of data bus (DB) and address bus (AB) of 8085?
46. What is the distinguishing feature of DB and AB?
47. The address capability of 8085 is 64 KB. Explain.
48. Does 8085 have serial I/O control?
49. How many instructions 8085 can support?
50. Mention the addressing modes of 8085.

APPLICATION OF MICROPROCESSOR 8085

1. Delay Subroutine using one, two or three register.
2. In 7 segments LED display.
3. In Microprocessor based Protective Relay.
4. In measurement of Electrical Quantities.
5. In measurement of Physical Quantities.
6. In moving of Stepper motor-
  (a). Permanent Magnet Type stepper motor.
  (**b).** Variable Reluctance stepper motor.
  (c). Interfacing of stepper motor.
7. Microprocessor based Traffic control light.
8. To Generate Square Waves(Pulse) using Microprocessor-
  (a). To Generate Square wave or Pulse using I/O port.
  (b). To Generate Square wave or Pulse using SOD Line.
9. Microprocessor based control of Firing Circuit of a Thyristor.
10. Interfacing of Digital Multiplexer/Data Selector.
11. Interfacing of Digital/Decoder.

REFERENCES:

1. Microprocessor Architecture, Programming and Application with the 8085- Ramesh S Gaonkar,4th Edition, Penram International.
2. L.A. Leventhal, “Introduction to Microprocessor; Software, Hardware, Programming” ; Prentice-Hall of India, 1983.
3. B. Ram, “Advanced Microprocessors and Interfacing”, Tata McGraw-Hill, 2001.