MICROPROCESSOR AND INTERFACING LAB

MICROPROCESSOR AND INTERFACING LAB

LIST OF EXPERIMENTS
V SEM.(ECE, CSE, IT,BME)

1. STUDY ARCHITECTURE OF 8085 & FAMILIARIZATION WITH ITS HARDWARE, COMMANDS & OPERATION OF MICROPROCESSOR KIT.
2. WRITE A PROGRAM USING 8085 & VERIFY FOR :
   (A) ADDITION OF TWO 8-BIT NUMBERS.
   (B) ADDITION OF TWO 16-BIT NUMBERS. (WITH CARRY)
3. WRITE A PROGRAM USING 8085 & VERIFY FOR :
   (A) SUBTRACTION OF TWO 8-BIT NUMBERS. (DISPLAY OF BARROW)
   (B) SUBTRACTION OF TWO 16-BIT NUMBERS. (DISPLAY OF BARROW)
4. WRITE A PROGRAM USING 8085 FOR MULTIPLICATION OF TWO 8-BIT NUMBERS BY REPEATED ADDITION METHOD CHECK MINIMUM NUMBER OF ADDITION & TEST FOR TYPICAL DATA.
5. WRITE A PROGRAM USING 8085 FOR MULTIPLICATION OF TWO 8-BIT NUMBERS BY BIT ROTATION METHOD & VERIFY.
6. WRITE A PROGRAM USING 8085 FOR DIVISION OF TWO 8-BIT NUMBERS BY REPEATED SUBTRACTION METHOD& TEST FOR TYPICAL DATA.
7. WRITE A PROGRAM USING 8085 FOR DIVISION OF TWO 8 -BIT NUMBERS BY BIT ROTATION METHOD & TEST FOR TYPICAL DATA.
8. WRITE A PROGRAM USING 8085 FOR FINDING SQUARE OF A NUMBER USING LOOK-UP TABLE & VERIFY.
9. WRITE A PROGRAM USING 8085 FOR FINDING SQUARE-ROOT OF A NUMBER.
10. STUDY OF 8086 MICROPROCESSOR KIT.
11. WRITE A PROGRAM USING 8086 FOR ADDITION OF TWO 16 BIT NUMBERS.
12. WRITE A PROGRAM USING 8086 FOR DIVISION OF A DEFINED DOUBLE WORD BY ANOTHER WORD & VERIFY.
13. WRITE A PROGRAM USING 8086 FOR COPYING 12 BYTES OF DATA FROM SOURCE TO DESTINATION & VERIFY
14. WRITE A PROGRAM USING 8086 & VERIFY FOR FINDING THE LARGEST NUMBER FROM AN ARRAY.
15. WRITE A PROGRAM USING 8086 FOR ARRANGING AN ARRAY OF NUMBERS IN DESCENDING ORDER & VERIFY.

EXPERIMENT NO. 1

AIM : STUDY OF 8085-MICROPROCESSOR KIT.

APPARATUS: 8085 microprocessor kit.

.

THEORY :

Intel 8085 is an 8-bit microprocessor. It is 40-pin IC package fabricated on a single LSI chip. It uses a single +5 V supply. Its clock speed is about 3 MHz. It consists of three main sections: -

1.ALU (Arithmetic and logic unit):-

The ALU performs the arithmetic and logical operation, addition, subtraction, logical AND, OR, EX-OR, Complement, Increment, Decrement, shift, clear.

2.Timing and Control Unit:-

It generates timing and control signals, which are necessary for the execution of instruction.

3.Registers: -

These are used for temporary storage of data and instruction. INTEL 8085 has following registers: -

i) One 8 bit accumulator
ii) Six 8 bit registers (B, C, D, E, H, L)
iii) One 16 bit stack pointer, SP
iv) One 16 bit program counter, PC
v) Instruction register
vi) Status register
vii) Temporary registers

PC contains the address of next instruction.
IR holds the instruction until it is decoded.
SP holds the address of the stack top.
Accumulator is used during execution of program for temporary storage of data.

Status flags are as follows: -

i) Carry (CS)
ii) Zero (Z)
iii) Sign (S)
iv) Parity (P)
v) Auxiliary Carry (AC)

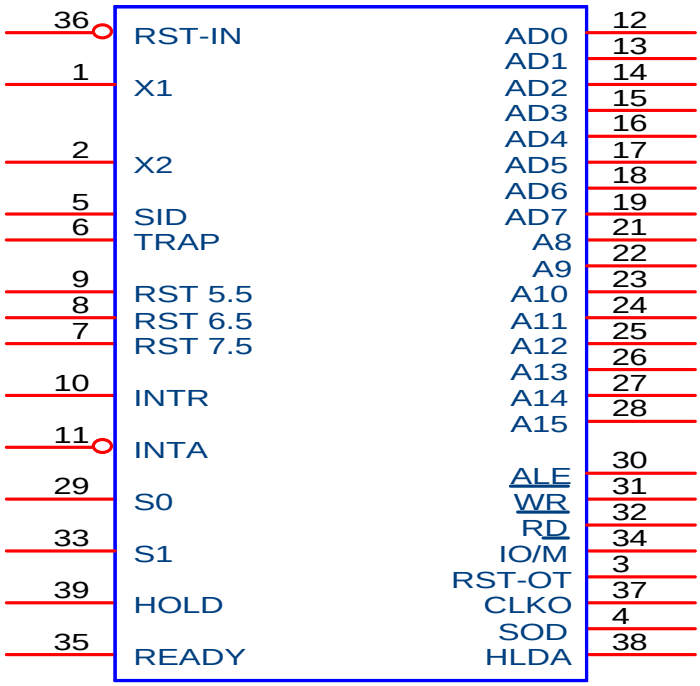
PSW

This 8-bit program status word includes status flags and three undefined bits.

Data and Address bus

Data bus is 8- bit wide and 8 bits of data can be transmitted in parallel. It has 16-bit wide address bus as the memory addresses are of 16 bits.

CIRCUIT DIAGRAM(PIN DIAGRAM):-



PIN CONFIGURATION

A8-A15 (Output):-

These are address bus and used for the most significant bits of memory address.

AD0-AD7 (Input/Output):-

These are time multiplexed address data bus. These are used for the least significant 8 bits of the memory address during first clock cycle and then for data during second and third clock cycle

ALE (Address Latch Enable)

It goes high during the 1st clock cycle of a machine. It enables the lower 8 bits of address to be latched either in the memory or external latch.

IO/M

It is status signal, when it goes high; the address on address bus is for I/O device, otherwise for memory.

So, S1

These are status signals to distinguish various types of operation

S1	So	Operation
0	0	Halt
0	1	Write
1	0	Read
1	1	Fetch

RD (output)

It is used to control read operation.

WR (output)

It is used to control write operation.

HOLD (input)

It is used to indicate that another device is requesting the use of address & data bus.

HLDA (output)

It is acknowledgement signal used to indicate HOLD request has been received.

INTR (input)

When it goes high, microprocessor suspends its normal sequence of operations.

INTA (output)

It is interrupt acknowledgement signal sent by microprocessor after INTR is received.

RST 5.5,6.5,7.5 and TRAP

These are various interrupt signals. Among them TRAP is having highest priority

RESET IN (input)

It resets the PC to zero.

RESET OUT(output)

It indicates that CPU is being reset.

X1, X2 (input)

This circuitry is required to produce a suitable clock for the operation of microprocessor. .

Clk (output)

It is clock output for user. Its frequency is same at which processor operates.

SID (input)

It is used for data line for serial input.

SOD (output)

It is used for data line for serial output.

Vcc

+5 volts supply

Vss

Ground reference

EXPERIMENT NO. 2(A)

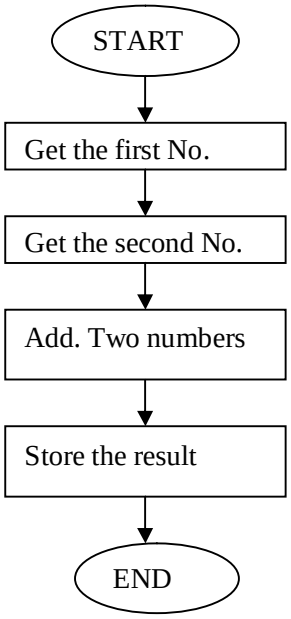
AIM : WRITE A PROGRAM USING 8085 & VERIFY FOR :
(a) ADDITION OF TWO 8-BIT NUMBERS.

APPARATUS : 8085 microprocessor kit, 5V power supply, Keyboard.

THEORY (Program)

Memory address	Machine code	Mnemonics	Operands	Commands
7000	21,01,75	LXI	H,7501	Get address of 1st no. in HL pair
7003	7E	MOV	A,M	Move Ist no. in accumulator
7004	23	INX	H	HL points the address 7502H
7005	86	ADD	M	Add the 2nd no.
7006	23	INX	H	HL points 7503H
7007	77	MOV	M,A	Store result in 7503H.
7008	CF	RST 1		Terminate

CIRCUIT DIAGRAM / BLOCK DIAGRAM:-



PROCEDURE:-

	SCIENTECH
ANSHUMAN	Reset
S	Exmem
Enter Enter	Starting Address
Program Address	Next
Write Program	Write Program
Execution Steps	**Execution Steps**
Esc	Reset
G	GO
Enter-enter	Starting Address
Prog. Address	Fill
Enter	Reset
S	Exmem
Enter	Result Address
Any key-2	
Enter-2	
Register Name	

INPUT DATA

7501- 13H
7502- 12H

OUTPUT DATA

7503- 25H

PRECAUTIONS:-

Make sure that all the machine codes should be as per specified in the program.

EXPERIMENT NO. 2(B)

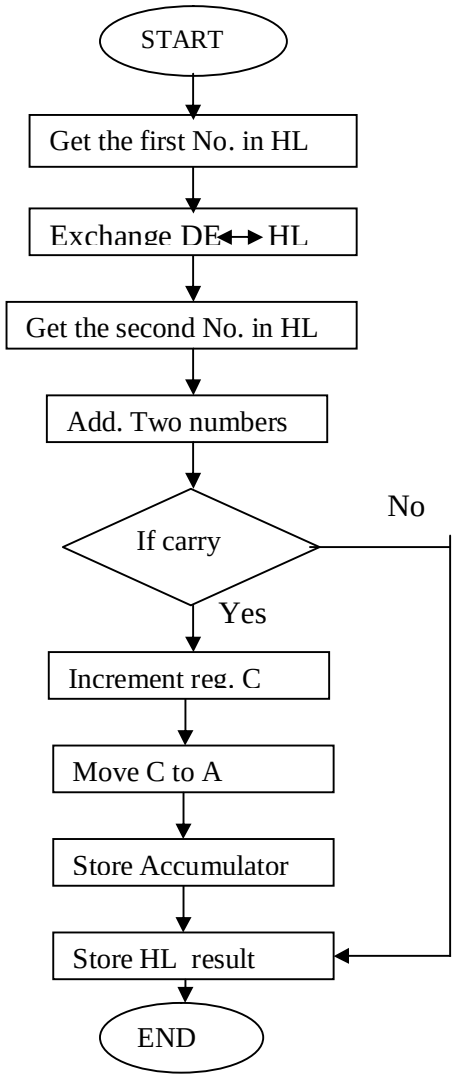
AIM : WRITE A PROGRAM USING 8085 & VERIFY FOR :
(b) ADDITION OF TWO 16-BIT NUMBERS(WITH CARRY).

APPARATUS : 8085 microprocessor kit, 5V power supply, Keyboard.

THEORY (Program)

Memory address	Label	Machine code	Mnemonics	Operands	Commands
7000		2A,01,76	LHLD	7601H	Get 1st no. in HL pair from memory (7601)
7003		EB	XCHG		Exchange cont. of DE ↔ HL
7004		2A,03,76	LHLD	7603H	Get 2st no. in HL pair from location 7603
7007		0E,00	MVI	C,00H	Clear reg. C.
7009		19	DAD	D	Get HL+DE & store result in HL
700A		D2,12,70	JNC	7012(loop)	If no carry move to loop/if carry then move to next step.
700D		0C	INR	C	Increment reg C
700E		79	MOV	A,C	Move carry from reg. C to reg. A
7011		32,02,75	STA	7502	Store carry at 7502H
7012	loop	22,00,75	SHLD	7500	Store result in 7500H.
7015		CF	RST1		Terminate

CIRCUIT DIAGRAM / BLOCK DIAGRAM:-



PROCEDURE:-

ANSHUMAN	SCIENTECH
S	Reset
Enter Enter	Exmem
Program Address	Starting Address
Write Program	Next
Execution Steps	Write Program
	Execution Steps
Esc	Reset
G	GO
Enter-enter	Starting Address
Prog. Address	Fill
Enter	Reset
S	Exmem
Enter	Result Address
Any key-2	
Enter-2	
Register Name	

INPUT DATA

7601 : 13H
7602 : 31H
7603 : 12H
7604 : 10H

OUTPUT DATA

7500 : 25H
7501 : 41H
7502 : 00H

PRECAUTIONS:-

Make sure that all the machine codes should be as per specified in the program.

EXPERIMENT NO. 3(A)

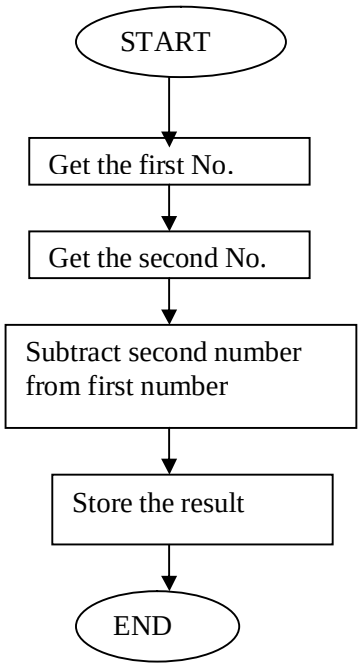
AIM : WRITE A PROGRAM USING 8085 & VERIFY FOR :
(A) SUBTRACTION OF TWO 8-BIT NUMBERS. (DISPLAY OF BARROW).

APPARATUS : 8085 microprocessor kit, 5V power supply, Keyboard.

THEORY(Program) :

Memory address	Opcode	Mnemonics	Operands	Comments
7000	21,01,75	LXI	H, 7501	Get address of ist no. in HL pair
7003	7E	MOV	A, M	Move Ist no. in accumulator
7004	23	INX	H	HL points 7502H.
7005	96	SUB	M	Substract 2nd no. from Ist no.
7006	23	INX	H	HL points 7503 H.
7007	77	MOV	M, A	Move contents of acc. to memory
7008	CF	RST 1		Stop

CIRCUIT DIAGRAM / BLOCK DIAGRAM :-



PROCEDURE:-

	SCIENTECH
ANSHUMAN	Reset
S	Exmem
Enter Enter	Starting Address
Program Address	Next
Write Program	Write Program
Execution Steps	**Execution Steps**
Esc	Reset
G	GO
Enter-enter	Starting Address
Prog. Address	Fill
Enter	Reset
S	Exmem
Enter	Result Address
Any key-2	
Enter-2	
Register Name	

INPUT DATA

7501 : 20H
7502 : 10H

OUTPUT DATA

7503 : 10H

PRECAUTIONS:-

Make sure that all the machine codes should be as per specified in the program.

EXPERIMENT NO. 3 (B)

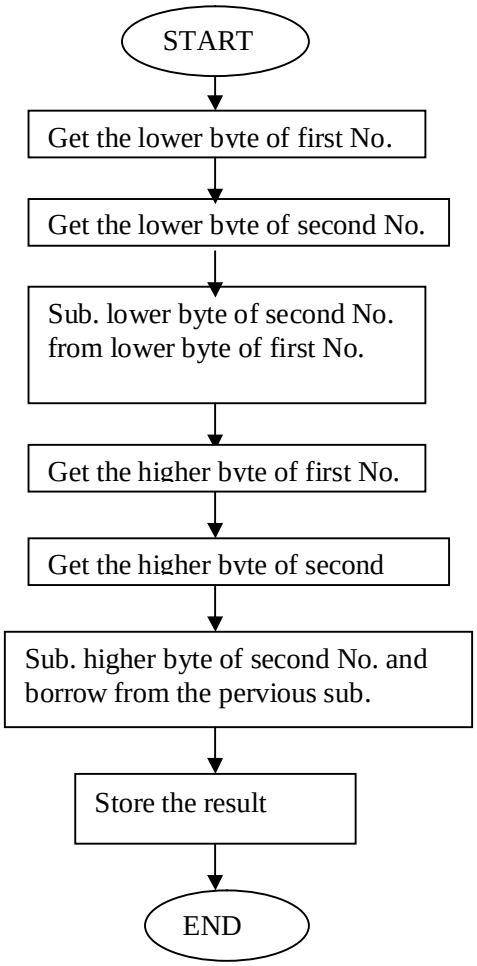
AIM : WRITE A PROGRAM USING 8085 & VERIFY FOR :
(B) SUBTRACTION OF TWO 16-BIT NUMBERS. (DISPLAY OF BARROW)

APPARATUS : 8085 microprocessor kit, 5V power supply, Keyboard.

THEORY (Program) :

Memory Address	Machine Code	Mnemonics	Operands	Comments
7000	2A, 01,75	LHLD	7501 H	Get 1st 16 bit no. in HL pair
7003	EB	XCHG		Exchange HL pair with DE.
7004	2A, 03,75	LHLD	7503 H	Get 2nd 16 bit no. in HL pair
7007	7B	MOV	A, E	Get lower byte of ist no.
7008	95	SUB	L	Subtract lower byte of 2nd no.
7009	6F	MOV	L, A	Store the result in reg. L
700A	7A	MOV	A, D	Get higher byte of Ist no.
700B	96	SBB	H	Subtract higher byte of 2nd no. with borrow
700C	67	MOV	H,A	Move from acc. To H
700D,E, F	22,05,75	SHLD	7505H	Store 16 bit result at 7505&7506
7010	CF	RST 1		Terminate

CIRCUIT DIAGRAM / BLOCK DIAGRAM :-



PROCEDURE:-

	SCIENTECH
ANSHUMAN	Reset
S	Exmem
Enter Enter	Starting Address
Program Address	Next
Write Program	Write Program
Execution Steps	**Execution Steps**
Esc	Reset
G	GO
Enter-enter	Starting Address
Prog. Address	Fill
Enter	Reset
S	Exmem
Enter	Result Address
Any key-2	
Enter -2	
Register Name	

INPUT DATA

7501 : 30H
7502 : 40H
7503 : 10H
7504 : 20H

OUTPUT DATA

7505 : 20H
7506 : 20H

PRECAUTIONS:-

Make sure that all the machine codes should be as per specified in the program.

EXPERIMENT NO. 4

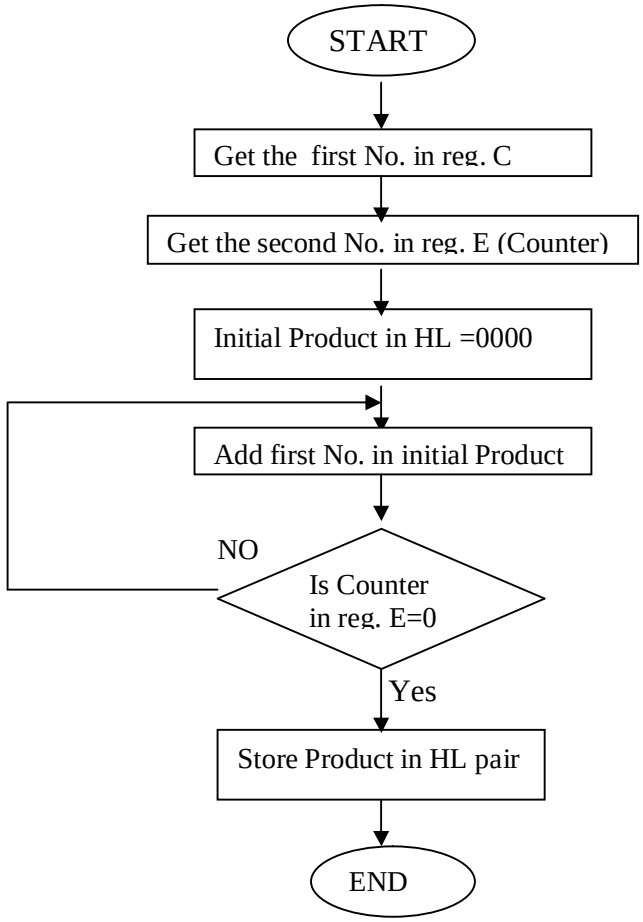
AIM : WRITE A PROGRAM USING 8085 FOR MULTIPLICATION OF TWO 8-BIT NUMBERS BY REPEATED ADDITION METHOD CHECK MINIMUM NUMBER OF ADDITION & TEST FOR TYPICAL DATA

APPARATUS : 8085 microprocessor kit, 5V power supply, Keyboard.

THEORY (Program) :

Memory Address	Label	Machine Code	Mnemonics	Operands	Comments
7000		0E,25	MVI	C,25	Move the no. in reg. C
7002		1E,05	MVI	E,05	Move the no. in reg. E
7004		06,00	MVI	B,00	Clear reg. B
7006		21,00,00	LXI	H,0000	Initial Product=0000
7009	UP1:	09	DAD	B	HL+BC=>HL
700A		1D	DCR	E	Decrement reg. E
700B		C2,09,70	JNZ	UP1(7009)	Jump if not zero to location up1
700E		22,00,75	SHLD	7500	Store HL at 7500
7011		CF	RST 1		Terminate

CIRCUIT DIAGRAM / BLOCK DIAGRAM:-



PROCEDURE:-

	SCIENTECH
ANSHUMAN	Reset
S	Exmem
Enter Enter	Starting Address
Program Address	Next
Write Program	Write Program
Execution Steps	**Execution Steps**
Esc	Reset
G	GO
Enter-enter	Starting Address
Prog. Address	Fill
Enter	Reset
S	Exmem
Enter	Result Address
Any key-2	
Enter	
Name	
Register	

INPUT DATA

1) Reg.C : 25H
   Reg.E : 05H
   Reg.B : 00H

OUTPUT DATA

HL pair : 00B9H

PRECAUTIONS:-

Make sure that all the machine codes should be as per specified in the program.

EXPERIMENT NO. 5

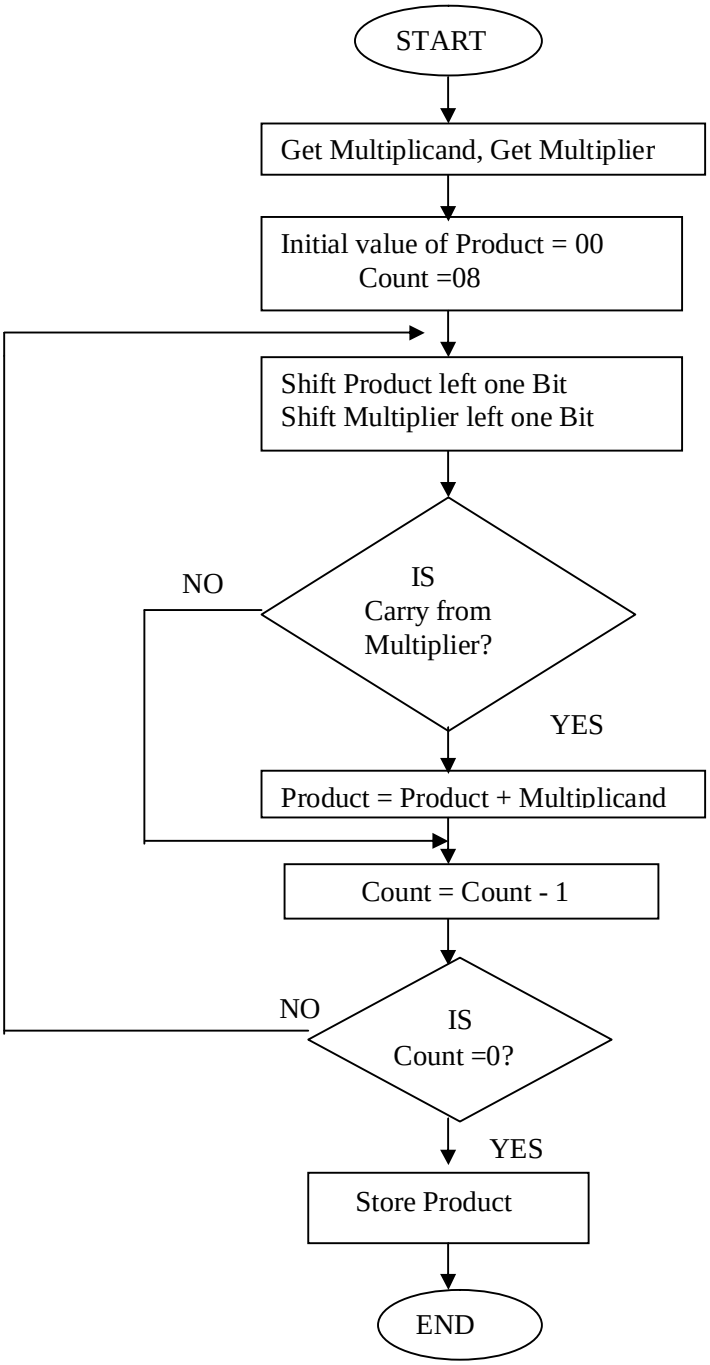
AIM : WRITE A PROGRAM USING 8085 FOR MULTIPLICATION OF TWO 8-BIT NUMBERS BY BIT ROTATION METHOD & VERIFY.

APPARATUS : 8085 microprocessor kit,5 V power supply, Keyboard.

THEORY(Program)

Memory Address	Label	Machine Code	Mnemonics	Operands	Comments
7000		2A,01,75	LHLD	7501 H	Get Multiplicand in H-L pair.
7003		EB	XCHG		Exchange HL pair with DE pair
7004		3A,03,75	LDA	7503 H	Get 2nd no. in acc.
7007		21,00,00	LXI	H,0000	Initial product in HL=00
700A		0E,08	MVI	C,08H	Count=08 in reg .C
700C	Loop	29	DAD	H	Shift partial product left by 1 bit
700D		17	RAL		Rotate multiplication by 1 bit. Is multiplier = 1?
700E		D2,12,70	JNC	Ahead(7012)	No, go to ahead
7011		19	DAD	D	Product=Product + Multiplicand
7012	Ahead	0D	DCR	C	Decrement Count
7013		C2,0C,70	JNZ	Loop(700C)	
7016		22,04,75	SHLD	7504	Store result
7019		CF	RST 1		Terminate

CIRCUIT DIAGRAM / BLOCK DIAGRAM :-



PROCEDURE:-

	SCIENTECH
ANSHUMAN	Reset
S	Exmem
Enter Enter	Starting Address
Program Address	Next
Write Program	Write Program
Execution Steps	**Execution Steps**
Esc	Reset
G	GO
Enter-enter	Starting Address
Prog. Address	Fill
Enter	Reset
S	Exmem
Enter	Result Address
Any key-2	
Enter-2	
Register Name	

INPUT DATA

7501- 25H
7502- 00H
7503- 05H

OUTPUT DATA

7504- B9H
7505- 00H

PRECAUTIONS:-

Make sure that all the machine codes should be as per specified in the program.

EXPERIMENT NO. 6

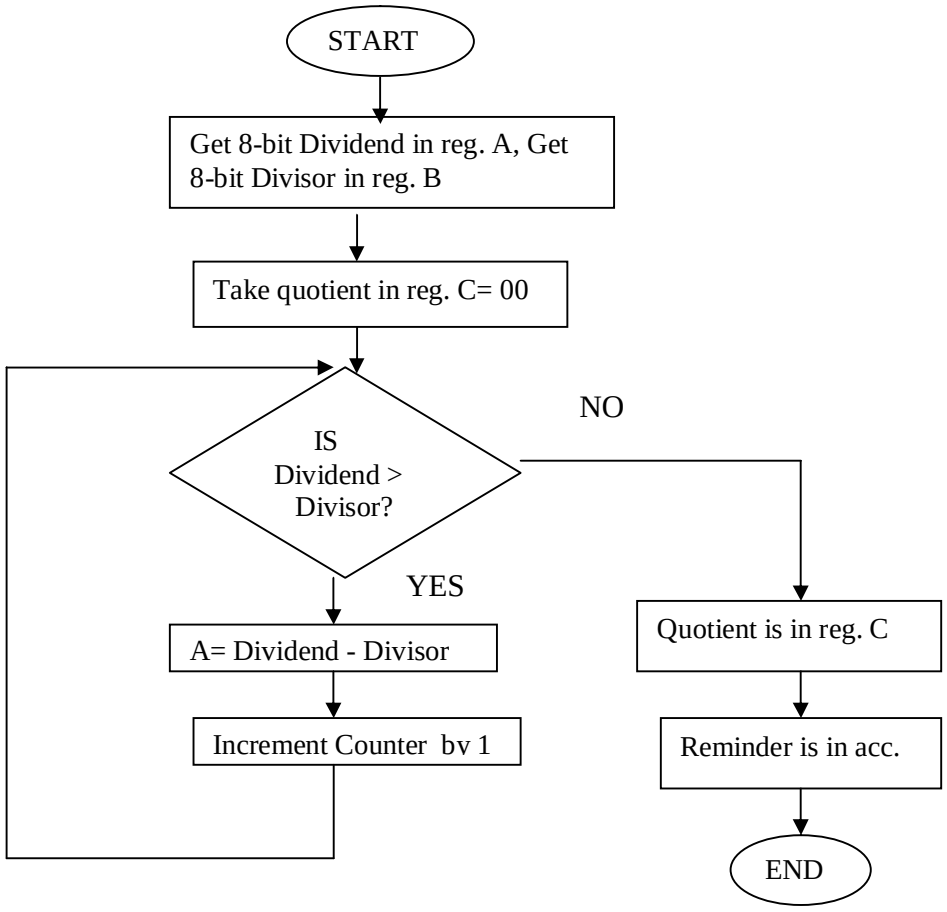
AIM : WRITE A PROGRAM USING 8085 FOR DIVISION OF TWO 8-BIT NUMBERS BY REPEATED SUBTRACTION METHOD& TEST FOR TYPICAL DATA.

APPARATUS : 8085 microprocessor kit, 5V power supply, Key board.

THEORY (Program) :

Memory Address	Label	Machine Code	Mnemonics	Operands	Comments
7000		3A,01,75	LDA	Divisor(7501)	
7003		47	MOV B,A		Take divisor in reg,B
7004		3A,02,75	LDA	Dividend(7502)	Take dividend in reg,A
7007		0E,00	MVI	C,00	Quotient=00
7009		B8	CMP	B	
700A		DA,13,70	JC	Loop(7013)	
700D	loop1	90	SUB	B	Dividend-divisor=>A
700E		0C	INR	C	C=C+1
700F		B8	CMP	B	Is dividend < divisor
7010		D2,0D,70	JNC	Loop1(700D)	If not,go back
7013	loop	32,03,75	STA	Remainder(7503)	Store Remainder
7016		79	MOV	A,C	
7017		32,04,75	STA	Quotient(7504)	Store Quotient
701A		CF	RST 1		Terminate.

CIRCUIT DIAGRAM / BLOCK DIAGRAM:-



PROCEDURE:-

	SCIENTECH
ANSHUMAN	Reset
S	Exmem
Enter Enter	Starting Address
Program Address	Next
Write Program	Write Program
Execution Steps	**Execution Steps**
Esc	Reset
G	GO
Enter-enter	Starting Address
Prog. Address	Fill
Enter	Reset
S	Exmem
Enter	Result Address
Any key-2	
Enter-2	
Register Name	

INPUT DATA

7501- Divisor
7502-Dividend

OUTPUT DATA

7503-Remainder
7504-Quotient

PRECAUTIONS:-

Make sure that all the machine codes should be as per specified in the program.

EXPERIMENT NO. 7

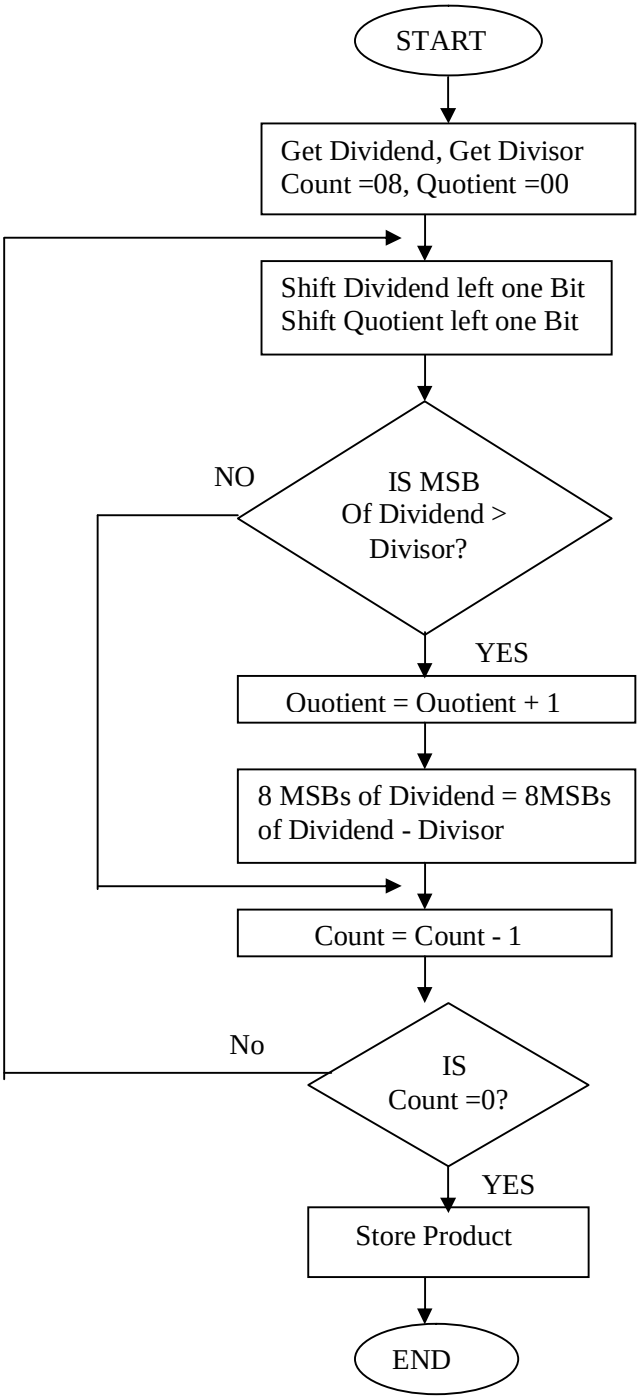
AIM : WRITE A PROGRAM USING 8085 FOR DIVISION OF TWO 8 -BIT NUMBERS BY BIT ROTATION METHOD & TEST FOR TYPICAL DATA.

APPARATUS : 8085 microprocessor kit, Keyboard, and 5V Power Supply.

THEORY (Program)

Memory Address	Label	Machine Code	Mnemonics	Operands	Comments
7000		2A, 01,75	LHLD	7501 H	Enter the 16 bit address in HL pair
7003		3A, 03,75	LDA	7503 H	Get divisor from 7503
7006		47	MOV	B, A	Divisor in register B
7007		0E, 08	MVI	C, 08	Count = 08 in register C.
7009	Loop	29	DAD	H	Shift dividend and quotient left by one bit.
700A		7C	MOV	A, H	Most significant bits of dividend in acc.
700B		90	SUB	B	Subtract divisor from MSB of dividend.
700C		DA, 11, 70	JC	Ahead(7011)	Is MSB of dividend>divisor? No, go to AHEAD.
700F		67	MOV	H, A	MSB of dividend in reg. H
7010		2C	INR	L	Yes, add 1 to quotient.
7011	Ahead	0D	DCR	C	Decrement count.
7012		C2, 09,70	JNZ	Loop(7009)	Is count=0?No, jump to loop.
7015		22,04,75	SHLD	7504 H	Store quotient in 7504 and remainder in 7505 H.
7018		CF	RST1		Stop.

CIRCUIT DIAGRAM / BLOCK DIAGRAM:-



PROCEDURE:-

	SCIENTECH
ANSHUMAN	Reset
S	Exmem
Enter Enter	Starting Address
Program Address	Next
Write Program	Write Program

Execution Steps	**Execution Steps**
Esc	Reset
G	GO
Enter-enter	Starting Address
Prog. Address	Fill
Enter	Reset
S	Exmem
Enter	Result Address
Any key-2	
Enter-2	
Register Name	

INPUT DATA

7501- LSB of dividend
7502- MSB of dividend
7503- Divisor

OUTPUT DATA

7504- Quotient
7505- Remainder

PRECAUTIONS:-

Make sure that all the machine codes should be as per specified in the program.

EXPERIMENT NO. 8

AIM : WRITE A PROGRAM USING 8085 FOR FINDING SQUARE OF A NUMBER USING LOOK-UP TABLE & VERIFY

APPARATUS : 8085 microprocessor kit, 5V power supply, Keyboard.

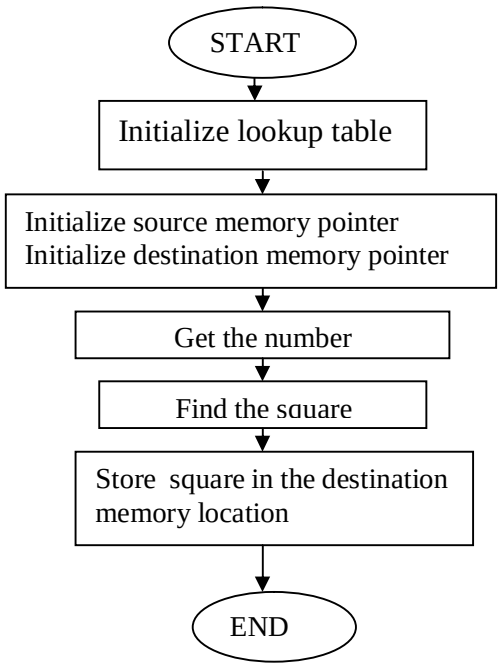
THEORY(Program) :

Memory Address	Machine Code	Mnemonics	Operands	Comments
2000	3A,00,25	LDA	2500 H	Get 1st no. in acc.
2003	6F	MOV	L,A	Move From A into reg. L
2004	26,26	MVI	H,26H	Get 26 in reg H
2006	7E	MOV	A,M	Square of data in accumulator
2007	32,01,25	STA	2501 H	Store square in 2501 H.
200A	CF	RST 1		Stop

LOOK-UP TABLE

Address		Square
2600	-	00
2601	-	01
2602	-	04
2603	-	09
2604	-	16
2605	-	25
2606	-	36
2607	-	49
2608	-	64
2609	-	81

CIRCUIT DIAGRAM / BLOCK DIAGRAM:-



PROCEDURE:-

ANSHUMAN
S
Enter Enter
Program Address
Write Program

Execution Steps

Esc
G
Enter-enter
Prog. Address
Enter
S
Enter
Any key-2
Enter-2
Register Name

SCIENTECH
Reset
Exmem
Starting Address
Next
Write Program

Execution Steps

Reset
GO
Starting Address
Fill
Reset
Exmem
Result Address

INPUT DATA

2500- 07H

OUTPUT DATA

2501- 49H

PRECAUTIONS:-

Make sure that all the machine codes should be as per specified in the program.

EXPERIMENT NO. 9

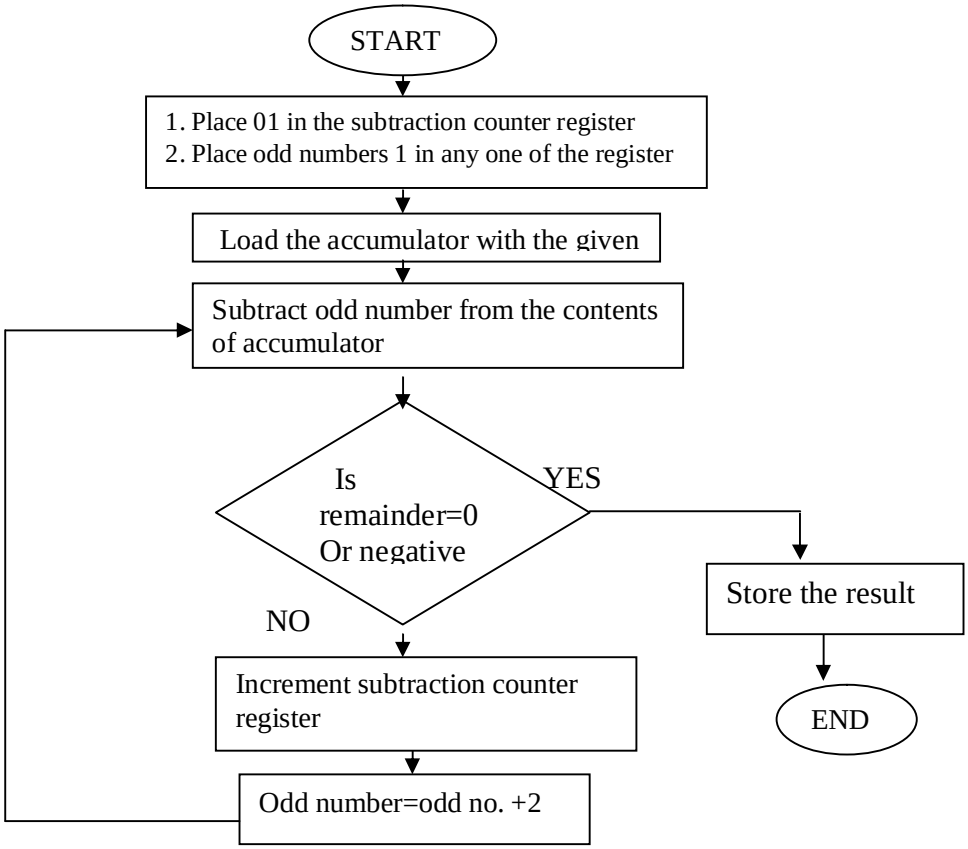
AIM :WRITE A PROGRAM USING 8085 FOR FINDING SQUARE-ROOT OF A NUMBER

APPARATUS : 8085 microprocessor kit, 5V power supply, Keyboard.

THEORY(Program):

Memory Address	Label	Machine Code	Mnemonics	Operands	Comments
2000		0E,01	MVI	C,01H	Place 01 in reg.C
2002		06,01	MVI	B,01H	Place odd number 1 in reg.B
2004		3E,36	MVI	A,36	Load accumulator with the given number
2006	Loop	90	SUB	B	Subtract odd number from the accumulator
2007		CA,10,20	JZ	Ahead(2010)	If accumulator contents are zero, go to Ahead
200A		0C	INR	C	Increment reg. C
200B		04	INR	B	Increment odd number
200C		04	INR	B	Increment odd number
200D		C3,06,20	JMP	Loop(2006)	Repeat subtraction
2010	Ahead	79	MOV	A,C	Move the contents of C reg. to accumulator
2011		32,50,20	STA	2050H	Store the result in the memory location 2050H.
2014		CF	RST1		Stop

CIRCUIT DIAGRAM / BLOCK DIAGRAM:-



PROCEDURE:-

	SCIENTECH
ANSHUMAN	Reset
S	Exmem
Enter Enter	Starting Address
Program Address	Next
Write Program	Write Program

Execution Steps	**Execution Steps**
Esc	Reset
G	GO
Enter-enter	Starting Address
Prog. Address	Fill
Enter	Reset
S	Exmem
Enter	Result Address
Any key-2	
Enter	
Name	
Register	

INPUT DATA

2500-10H
2501- 00H

OUTPUT DATA

2550- 04H

PRECAUTIONS:-

Make sure that all the machine codes should be as per specified in the program.

EXPERIMENT NO .10

AIM : STUDY OF 8086 MICROPROCESSOR KIT.

APPARATUS: 8086 microprocessor kit.

.

THEORY : The 8086 is a 16-bit, N-channel, HMOS microprocessor. The term HMOS is used for "high-speed MOS". The 8086 uses 20 address lines and 16 data lines. It can directly address up to 2^{20} = 1Mbytes of memory. The 16-bit data word is divided into a low-order byte and a high-order byte. The 20 address lines are time multiplexed lines. The 16 low-order address lines are time multiplexed with data, and the 4 high-order address lines are time multiplexed with status signals

OPERATING MODES OF 8086

There are two modes of operation for Intel 8086, namely the minimum mode and the maximum mode. When only one 8086 CPU is to be used in a microcomputer system the 8086 is used in the minimum mode of operation. In this mode the CPU issues the control signals required by memory and I/O devices. In case of maximum mode of operation control signals are issued by Intel 8288 bus controller which is used with 8086 for this very purpose. When MN/$\overline{MX}$ is high the CPU operates in the minimum mode. When it is low the CPU operates in the maximum mode.

Pin Description For Minimum Mode

For the minimum mode of operation the pin MN/$\overline{MX}$ is connected to 5V d.c supply. The description of the pins from 24 to 31 for the minimum mode is as follows:

$\overline{INTA}$(Output): Pin no. 24 Interrupt acknowledge. On receiving interrupt signal the processor issues an interrupt acknowledge signal. It is active LOW.
ALE(Output) : Pin no. 25 Address latch enable. It goes HIGH during T1. The microprocessor sends this signal to latch the address into the Intel 8282/8283 latch.
$\overline{DEN}$(Output) : Pin no. 26 Data enable. When Intel 8286/8287 octal bus transceiver is used this signal acts as an output enable signal. It is active LOW.
DT/$\overline{R}$(Output) : Pin no. 27 Data Transmit/Receive. When Intel 8286/8287 octal bus transceiver is used this signal controls the direction of data flow through the transceiver. When it is High data are sent out. When it is LOW data are received.
M/$\overline{IO}$(Output) : Pin no. 28.Memory or I/O access. When it is HIGH the CPU wants to access memory. When it is LOW the CPU wants to access I/O device.
$\overline{WR}$ (Output) : Pin no. 29. Write. When it is LOW the CPU performs memory or I/O write Operation.
HLDA (Output) : Pin no. 30.HOLD acknowledge. It is issued by the processor when it receives HOLD signal. It is active HIGH signal. When HOLD request is removed HLDA goes LOW.
HOLD (Output) : Pin no. 31.Hold. when another device in microcomputer system wants to use the address and data bus, it sends a HOLD request to CPU through this pin. It is an active HIGH signal.

Pin Description For Maximum Mode

For the maximum mode of operation the pin MN/$\overline{MX}$ is made LOW. It is grounded. The description of the pins from 24 to 31 is as follows:
QS1,QS0(Output): Pin no. 24,25 Instruction Queue status. Logic are given below:

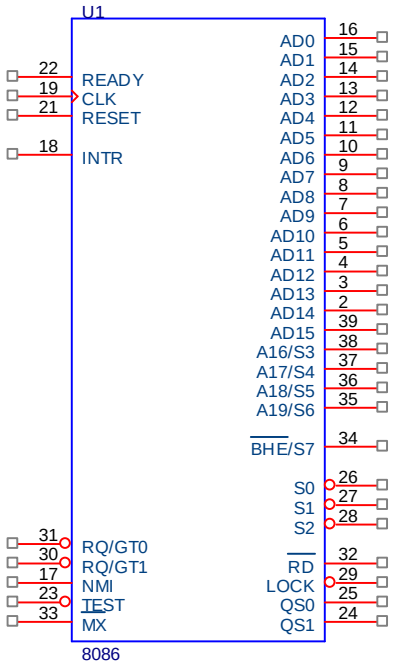
QS1	QS0	
0	0	No operation
0	1	1st byte of opcode from queue
1	0	Empty the queue
1	1	Subsequent byte from queue

$\overline{S0}$,$\overline{S1}$,$\overline{S2}$(Output) : Pin nos. 26,27,28.status signals. These signals are connected to the bus controller Intel 8288.The bus controller generates memory and I/O access control signals. Table for status signals is :

$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	
0	0	0	Interrupt acknowledge
0	0	1	Read data from I/O port
0	1	0	Write data into I/O port
0	1	1	Halt
1	0	0	Opcode fetch
1	0	1	Memory read
1	1	0	Memory write
1	1	1	Passive state.

$\overline{LOCK}$(Output) : Pin no. 29.It is an active LOW signal. When it is LOW all interrupts are masked and no HOLD request is granted. In a multiprocessor system all other processors are informed by this signal that they should not ask the CPU for relinquishing the bus control.

$\overline{RQ}$ / $\overline{GT_1}$, $\overline{RQ}$ / $\overline{GT_0}$ (Bidirectional) : Pin no. 30,31. Local bus Priority control. Other processors ask the CPU through these lines to release the local bus. **$\overline{RQ}$ / $\overline{GT_1}$** has higher priority than **$\overline{RQ}$ / $\overline{GT_0}$**

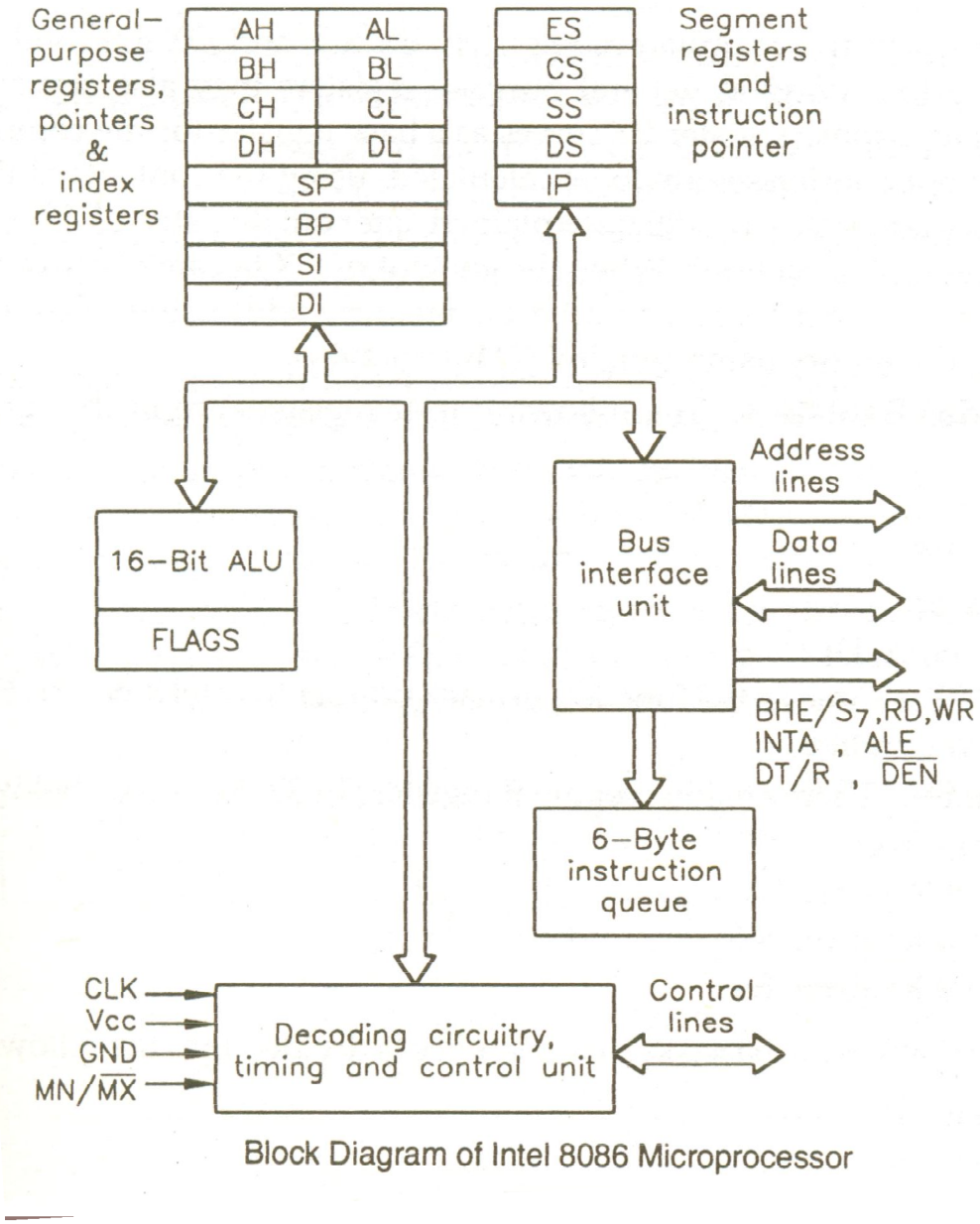


PIN DIAGRAM OF 8086

FUNCTIONAL UNITS OF 8086 :

The 8086 contains two functional units: a bus interface unit (BIU) and an execution unit(EU). The general purpose registers, stack pointer, base pointer and index registers, ALU, flag register(FLAGS), instruction decoder and timing and control unit constitute execution unit(EU). The segment registers, instruction pointer and 6-byte instruction queue are associated with the bus interface unit(BIU)

BLOCK DIAGRAM OF 8086:



Block Diagram of Intel 8086 Microprocessor

REGISTERS OF 8086 : The Intel 8086 contains the following registers:

a) General Purpose Register
b) Pointer and Index Registers
c) Segment Registers
d) Instruction Registers
e) Status Flags

EXPERIMENT NO. 11

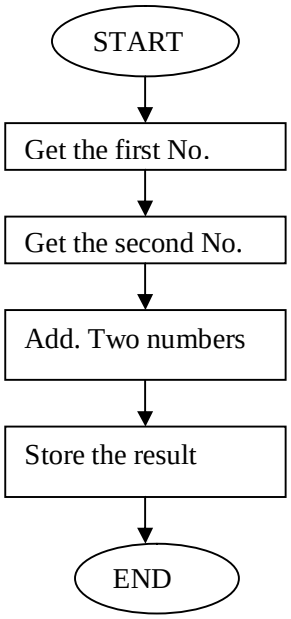
AIM : WRITE A PROGRAM USING 8086 FOR ADDITION OF TWO 16 BIT NUMBERS.

APPARATUS : 8086 microprocessor kit, 5V power supply, Keyboard.

THEORY(Program):

Memory Address	Machine Code	Mnemonics	Oprands	Comments
1000	B8,34,12	MOV	AX,1234	Load 1234 in AX
1003	BA,65,87	MOV	DX,8765	Load 8765 in DX
1006	03,C2	ADD	AX,DX	Add DX with AX
1008	8B,C8	MOV	CX,AX	Move answer to CX
1009	CD,A5	INT A5		Jump to command mode saving all registers.

CIRCUIT DIAGRAM / BLOCK DIAGRAM:-



PROCEDURE:-

ANSHUMAN
S
Enter Enter
SRC-SEGM Address
Enter
Program Address
Write Program

SCIENTECH
Reset
O
EB/AX

Starting Address
Next
Write Program

Execution Steps	**Execution Steps**
Esc	Reset
G	GO
Enter-enter	Starting Address
SRC-SEGM Add	Fill
Enter	Reset
Prog. Address	O
Enter	EB/AX
S	
Enter	Result Address
Any key-2	
Enter-2	
Register Name	

INPUT DATA

1000-1234(H)
1001-8765(H)

OUTPUT DATA

AX -9999(H)

PRECAUTIONS:-

Make sure that all the machine codes should be as per specified in the program.

EXPERIMENT NO. 12

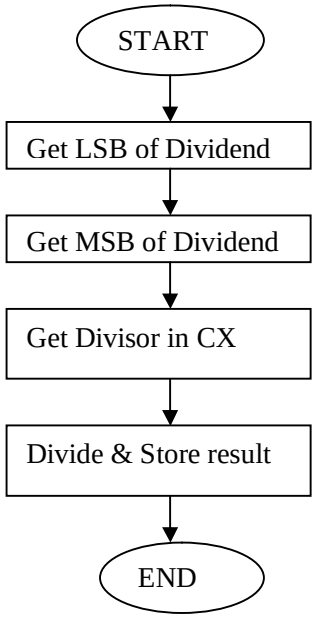
AIM : WRITE A PROGRAM USING 8086 FOR DIVISION OF A DEFINED DOUBLE WORD BYANOTHER WORD & VERIFY.

APPARATUS : 8086 microprocessor kit, 5V power supply, Keyboard.

THEORY(Program)

Memory Address	Machine Code	Mnemonics	Operands	Comments
1000	B8,78,56	MOV	AX,5678H	Move 5678 to AX
1003	BA,34,12	MOV	DX,1234H	Move 1234 to DX
1006	B9,25,25	MOV	CX,2525	Move 2525 to CX
1009	F7,F1	DIV	CX	Divide AX&DX by CX
100b	CD,A5	INT	A5	

CIRCUIT DIAGRAM / BLOCK DIAGRAM:-,



PROCEDURE:-

ANSHUMAN
S
Enter Enter
SRC-SEGM Address
Enter
Program Address
Write Program

SCIENTECH
Reset
O
EB/AX

Starting Address
Next
Write Program

Execution Steps	Execution Steps
Esc	Reset
G	GO
Enter-enter	Starting Address
SRC-SEGM Add	Fill
Enter	Reset
Prog. Address	O
Enter	EB/AX
S	
Enter	Result Address
Any key-2	
Enter-2	
Register Name	

INPUT DATA

AX : 5678H
DX : 1234H
CX : 2525H

OUTPUT DATA

AX : 7D77(Quotient)
DX : 0145(Remainder)

PRECAUTIONS:-

Make sure that all the machine codes should be as per specified in the program.

EXPERIMENT NO. 13

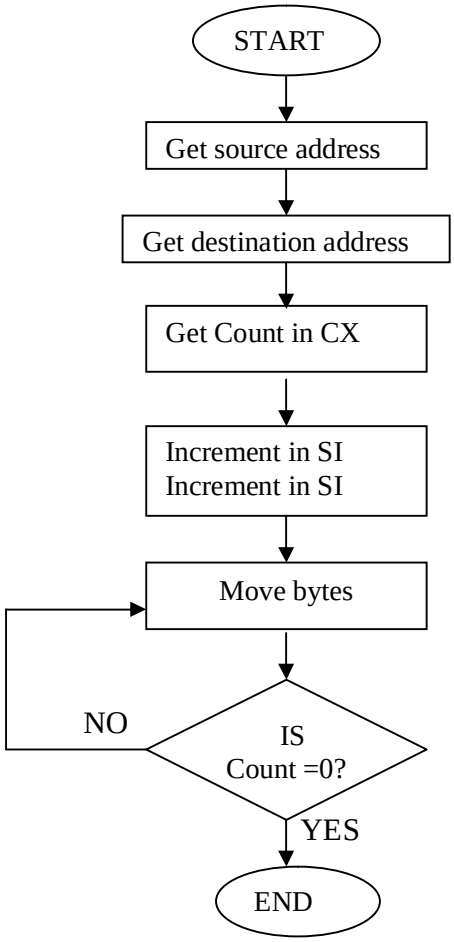
AIM : WRITE A PROGRAM USING 8086 FOR COPYING 12 BYTES OF DATA FROM SOURCE TO DESTINATION & VERIFY

APPARATUS : 8086 microprocessor kit, 5V power supply, Keyboard.

THEORY(Program)

Memory Address	Label	Machine Code	Mnemonics	Operands	Comments
0101		FC	CLD		Clear direction flag DF
0102		BE,00,03	MOV	SI,0300	Source address in SI
0105		BF,02,02	MOV	DI,0202	Destination address in DI
0108		8B,0C	MOV	CX,[SI]	Count in CX
010A		46	INC	SI	Increment SI
010B		46	INC	SI	Increment SI
010C	BACK	A4	MOV	SB	Move byte
010D		E2,FD	LOOP	BACK	Jump to BACK until CX becomes zero
010F		CC	INT		Interrupt program

CIRCUIT DIAGRAM / BLOCK DIAGRAM:-



INPUT DATA

0300	: 0B
0301	: 00
0302	: 03
0303	: 04
0304	: 05
0305	: 06
0306	: 15
0307	: 07
0308	: 12
0309	: 08
030A	: 09
030B	: 0A
030C	: 0B
030D	: 0E

OUTPUT DATA

0202	: 03
0203	: 04
0204	: 05
0205	: 06
0206	: 15
0207	: 07
0208	: 12
0209	: 08
020A	: 09
020B	: 0A
020C	: 0B
020D	: 0E

PRECAUTIONS:-

Make sure that all the machine codes should be as per specified in the program.

EXPERIMENT NO. 14

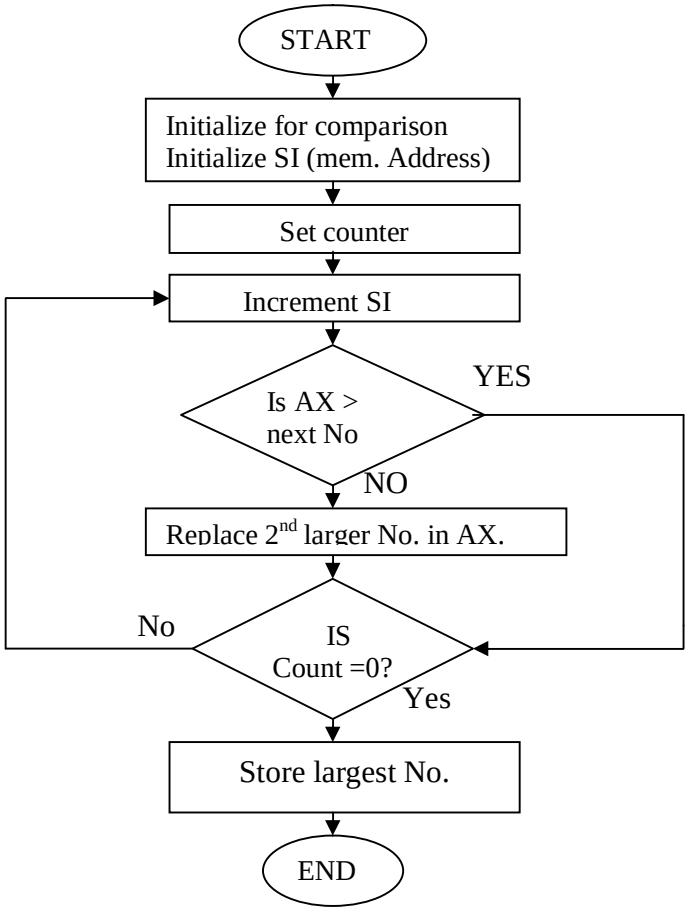
AIM : WRITE A PROGRAM USING 8086 & VERIFY FOR FINDING THE LARGEST NUMBER FROM AN ARRAY.

APPARATUS : 8086 microprocessor kit, 5V power supply, Keyboard.

THEORY(Program)

Memory Address	Label	Machine Code	Mnemonics	Operands	Comments
0101		B0,00,00	MOV	AX,0000	Initial value for comparison
0104		BE,00,02	MOV	SI,0200	Memory address in SI
0107		8B,0C	MOV	CX,[SI]	Count in CX
0109	BACK	46	INC	SI	Increment SI
010A		46	INC	SI	Increment SI
010B		3B,04	CMP	AX,[SI]	Compare previous largest number with next number
010D		73,02	JAE	GO	Jump if number in AX is larger i.eCF=0
010F		8B,04	MOV	AX,[SI]	Save next larger number in AX
0111	GO	E2,F6	LOOP	BACK	Jump to BACK until CX becomes zero
0113		A3,51,02	MOV	(0251),AX	Store largest number in memory
0116		CC	INT3		Interrupt program

CIRCUIT DIAGRAM / BLOCK DIAGRAM:-



PROCEDURE:-

	SCIENTECH
ANSHUMAN	Reset
S	Exmem
Enter Enter	Starting Address
Program Address	Next
Write Program	Write Program

Execution Steps	**Execution Steps**
Esc	Reset
G	GO
Enter-enter	Starting Address
Prog. Address	Fill
Enter	Reset
S	Exmem
Enter	Result Address
Any key-2	
Enter-2	
Register Name	

INPUT DATA

0200	: 05H
0201	: 00H
0202	: 41H
0203	: 83H
0204	: 58H
0205	: 72H
0206	: 39H
0207	: 46H
0208	: 53H
0209	: 84H
020A	: 30H
020B	: 96H

OUTPUT DATA

251	: 30H
252	: 96H

PRECAUTIONS:-

Make sure that all the machine codes should be as per specified in the program.

Experiment No. 15

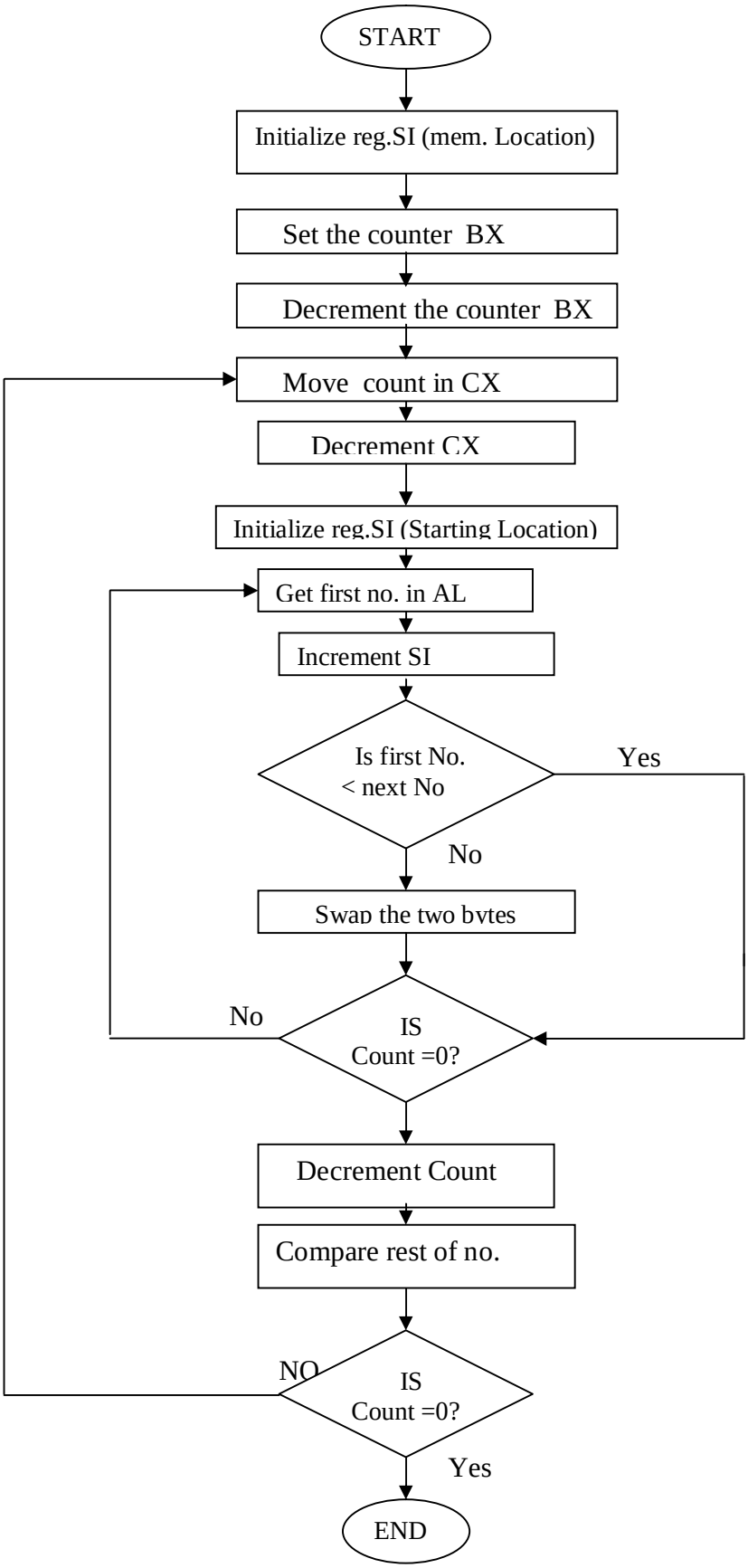
AIM : WRITE A PROGRAM USING 8086 FOR ARRANGING AN ARRAY OF NUMBERS IN DESCENDING ORDER & VERIFY.

APPARATUS : 8086 microprocessor kit, 5V power supply, Keyboard.

THEORY(Program)

Memory Address	Label	Machine Code	Mnemonics	Operands	Comments
0200		BE,00,03	MOV	SI,0300	Initialize SI Reg. with Memory Location. 0300.
0203		8B,1C	MOV	BX,[SI]	BX has no. of bytes
0205		4B	DEC	BX	Decrement the no. of bytes by one
0206	(3)	8B 0C	MOV	CX (SI)	Move no. of bytes in CX
0208		49	DEC	CX	Decrement the no. of bytes by one
0209		BE,02,03	MOV	SI,0303	Initialize SI reg. with the starting address of string
020C	(2)	8A,04	MOV	AL,[SI]	Move first data byte of string into AL
020E		46	INC	SI	Point at the next bytes of the string
020F		3A,04	COMP	AL,[SI]	Com. the two bytes of string.
0211		73,06	JAE	(1)	If two bytes are equal or 1st byte is above that the second byte branch to (1)
0213		86,04	XCHG	AL,[SI]	Else
0215		4E	DEC	SI	Second byte is less than first byte and swap the two bytes.
0216		88,04	MOV	[SI],AL	
0218		46	INC	SI	Point at next location of string
0219	(1)	E2,F1	LOOP	(2)	Loop if CX is not zero
021B		4B	DEC	BX	
021C		BE,00,03	MOV	SI,0300	
021F		75,E5	JNZ	(3)	
0221		F4	HLT		Halt.

CIRCUIT DIAGRAM / BLOCK DIAGRAM:-



PROCEDURE:-

	SCIENTECH
ANSHUMAN	Reset
S	Exmem
Enter Enter	Starting Address
Program Address	Next
Write Program	Write Program
Execution Steps	**Execution Steps**
Esc	Reset
G	GO
Enter-enter	Starting Address
Prog. Address	Fill
Enter	Reset
S	Exmem
Enter	Result Address
Any key-2	
Enter-2	
Register Name	

INPUT DATA

0300	: 05
0301	: 00
0302	: 20
0303	: 25
0304	: 28
0305	: 15
0306	: 07

OUTPUT DATA

0302	: 28
0303	: 25
0304	: 20
0305	: 15
0306	: 07

PRECAUTIONS:-

Make sure that all the machine codes should be as per specified in the program.

Questions-Answers based on practicals

Q.l Explain MOV r,M ?
Q.2 How many T-state are in MOV instruction?
Q.3 Explain the addressing mode ofMOV r,M?
Q.4 How many machine cycles are in MOV instruction?
Q.5 What is MOV M,r ?
Q.6 Which flag is affected in MOV instruction?
Q.7 What is MVI r,data?
Q.8 How many T-state are in MVI instruction?
Q.9 Explain the addressing mode of MVI r,data?
Q.10 How many machine cycles are in MVI instruction?
Q.11 Explain LXI rp,data 16 ?
Q.12 How many T-state are in LXI instruction?
Q.13 Explain the addressing mode of LXI rp,data?
Q.14 How many machine cycles are in LXI instruction?
Q.15 What is LDA addr ?
Q.16 How many T-state are in LDA instruction?
Q.17 Explain the addressing mode ofLDA addr?
Q.18 How many machine cycles are in LDA instruction?
Q.19 What is STA addr?
Q.20 How many T-state are in STA instruction?
Q.21 Explain the addressing mode ofSTA addr?
Q.22 How many machine cycles are in STA instruction?
Q.23 What is LHLD addr?
Q.24 How many T-state are in LHLD instruction?
Q.25 Explain the addressing mode ofLHLD addr?
Q.26 How many machine cycles are in LHLD instruction?
Q.27 What is SHLD addr ?
Q.28 How many T-state are in SHLD instruction?
Q.29 Explain the addressing mode of SHLD addr?
Q.30 How many machine cycles are in SHLD instruction?
Q.31 What is LDAX rp?
Q.32 How many T-state are in LDAX instruction?
Q.33 Explain the addressing mode ofLDAX rp?
Q.34 How many machine cycles are in LDAX instruction?
Q.35 What is STAX rp ?
Q.36 How many T-state are in STAX instruction?
Q.37 Explain the addressing mode ofSTAX rp?
Q.38 How many machine cycles are in STAX instruction?
Q.39 What is XCHG ?
Q.40 How many T-state are in XCHG instruction?
Q.4l Explain the addressing mode ofXCHG?
Q.42 How many machine cycles are in XCHG instruction?
Q.43 What is ADD r ?
Q.44 How many T-state are in ADD instruction?
Q.45 Explain the addressing mode of ADD?
Q.46 How many machine cycles are in ADD instruction?
Q.47 What is ADC r?
Q.48 How many T-state are in ADC r instruction?
Q.49 Explain the addressing mode of ADC ?

Q.50 How many machine cycles are in ADC instruction?
Q.51 Explain ADI data?
Q.52 How many T-states are in ADI instruction?
Q.53 Explain the addressing mode of ADI?
Q.54 How many machine cycles are in ADI instruction?
Q 55Explain DAD rp ?
Q.56How many T-states are in DAD instruction?
Q.57Explain the addressing mode of DAD.
Q.58How many machine cycles are in DAD instruction?
Q.59 Explain DAA.
Q.60 What is INX rp?

Answers

A.1 Move the content of memory to register
A.2 Four T-state
A.3 Register indirect
A.4 two machine cycle
A.5 move the content of register to memory
A.6 none
A 7 move immediate data to register
A.8 seven T-states
A.9 immediate
A.10 three machine cycles
A.11 load register pair immediate
A.l2 ten T -states
A.l3 immediate
A.14 three machine cycles
A.15 load accumulator direct
A.l6 thirteen T -states
A.17 direct
A.l8 four
A.l9 store accumulator direct
A.20 thirteen T -states
A.21 direct
A.22 four
A.23 load H - L pair direct
A.24 sixteen T -states A25 direct
A.26 five
A.27 store H-L pair direct
A.28 sixteen T –states
A.29 direct
A.30 five
A.31 Load accumulator indirect
A.32 seven
A.33 register indirect
A.34 two
A.35 Store accumulator indirect
A.36 seven
A.37 register indirect
A.38 two
A.39 Exchange the contents ofH-L pair with D-E pair
A.40 four
A.41 register
A.42 one
A.43 Add register to accumulator
A.44 four
A.45 register
A.46 two
A.47 Add register with carry to accumulator
A.48 four
A.49 register
A.50 one
A.51 Add immediate data to accumulator

A.52 seven T-states
A.53 immediate
A.54 two
A.55 Add register pair to HL pair
A.56 ten
A.57 register
A.58 three
A.59 Decimal adjust accumulator
A.60 Increment register pair