TAGORE INSTITUTE OF ENGINEERING AND TECHNOLOGY

Deviyakurichi-636112, Attur (TK), Salem (DT). Website: www.tagoreiet.ac.in

(Approved by AICTE, New Delhi and Affiliated to Anna University, Chennai)**Accredited by NAAC**

Department of Computer Science And Engineering

III Year- V Semester – Computer Science And Engineering

EC8681-MicroProcessors and MicroControllers Laboratory

LAB MANUAL

Academic Year 2020-2021

(2017 Regulation)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

V SEM – CSE

EC8681-MicroProcessors and MicroControllers Laboratory

S.NO	NAME OF EXPERIMENTS
	8086 Programs using kits and MASM
1.	Basic arithmetic and Logical operations
2.	Move a data block without overlap
3.	Code conversion, decimal arithmetic and Matrix operations.
4.	Floating point operations, string manipulations, sorting and searching
5.	Password checking, Print RAM size and system date
6.	Counters and Time Delay
	Peripherals and Interfacing Experiments
7.	Traffic light controller
8.	Stepper motor control
9.	Digital clock
10.	Key board and Display
11.	Printer status
12.	Serial interface and Parallel interface
13.	A/D and D/A interface and Waveform Generation
	8051 Experiments using kits and MASM

14.	Basic arithmetic and Logical operations
15.	Square and Cube program, Find 2,,s complement of a number
16.	Unpacked BCD to ASCII

EXPT NO : ADDITION OF TWO 16-BIT DATA USING KITS

DATE :

Aim:

To write and execute an assembly language program for addition of two 16-bit data by using 8086 microprocessor trainer kit.

Apparatus Required:

Sl.No.	Apparatus Required	Quantity
1	8086 Microprocessor Kit	1
2	Power chord	1
3	Keyboard PS/2 port	1

Algorithm:

1. Load the first data in AX register.
2. Load the second data in BX register.
3. Clear the CL register.
4. Add the two data and get the result in AX register.
5. Store the sum in memory.
6. Check for carry, if carry 1 go to step 7 else if carry 0 go to step 8.
7. Increment C register.
8. Store the carry in memory.
9. Terminate the program.

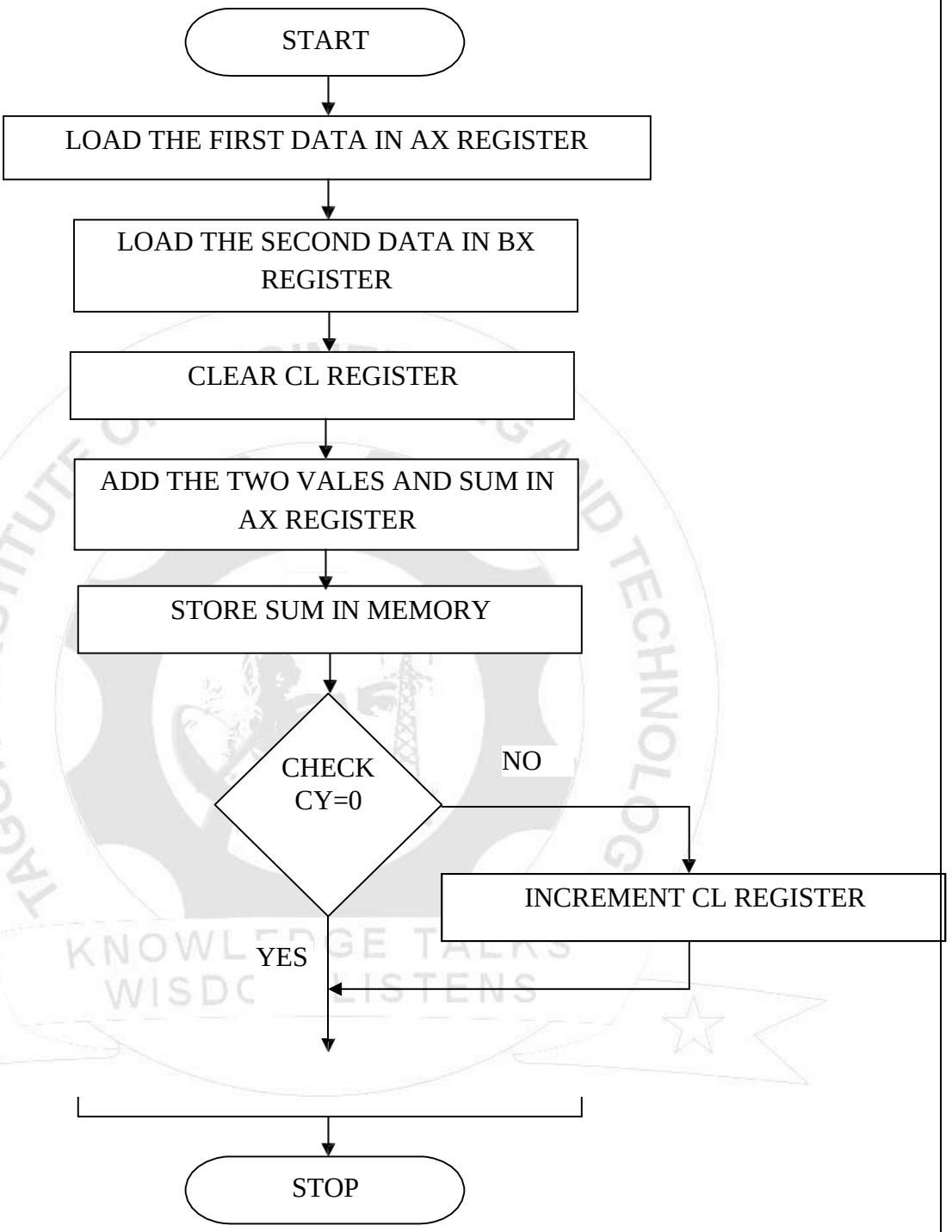
Program:

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
1000		MOV AX, 1100H	8B 06 00 11	; Load the first data in AX register.
1004		MOV BX,1102H	8B 1E 02 11	; Load the second data in BX register.
1008		MOV CL,00H	C6 C1 00	; Clear the CL register.
100B		ADD AX,BX	01 D8	; Add the two data and sum in AX register.
100D		MOV [1104H],AX	89 06 04 11	; Store the sum in to memory location 1104H
1011		JNC L1	73 02	; If CY=0 go to L1 otherwise do the following
1013		INC CL	FE C1	; Increment CL register.
1015	L1:	MOV [1106H],CL	88 0E 06 11	; Store the carry into memory location 1106H
1019		HLT	F4	;Terminate the program

SAMPLE INPUT AND OUTPUT:

INPUT		OUTPUT	
ADDRESS	**DATA**	**ADDRESS**	**DATA**
1100H	48H	1104H	9AH
1101H	F0H	1105H	00H
1102H	52H	1106H	01H
1103H	10H		

Flowchart:



Result:

Thus the assembly language program for addition of two 16-bit data was written and executed successfully by using 8086 microprocessor trainer kit.

EXPT NO. : SUBTRACTION OF TWO 16-BIT DATA

DATE :

Aim:

To write and execute an assembly language program for subtraction of two 16-bit data by using 8086 microprocessor trainer kit.

Apparatus Required:

Sl.No.	Apparatus Required	Quantity
1	8086 Microprocessor Kit	1
2	Power chord	1
3	Keyboard PS/2 port	1

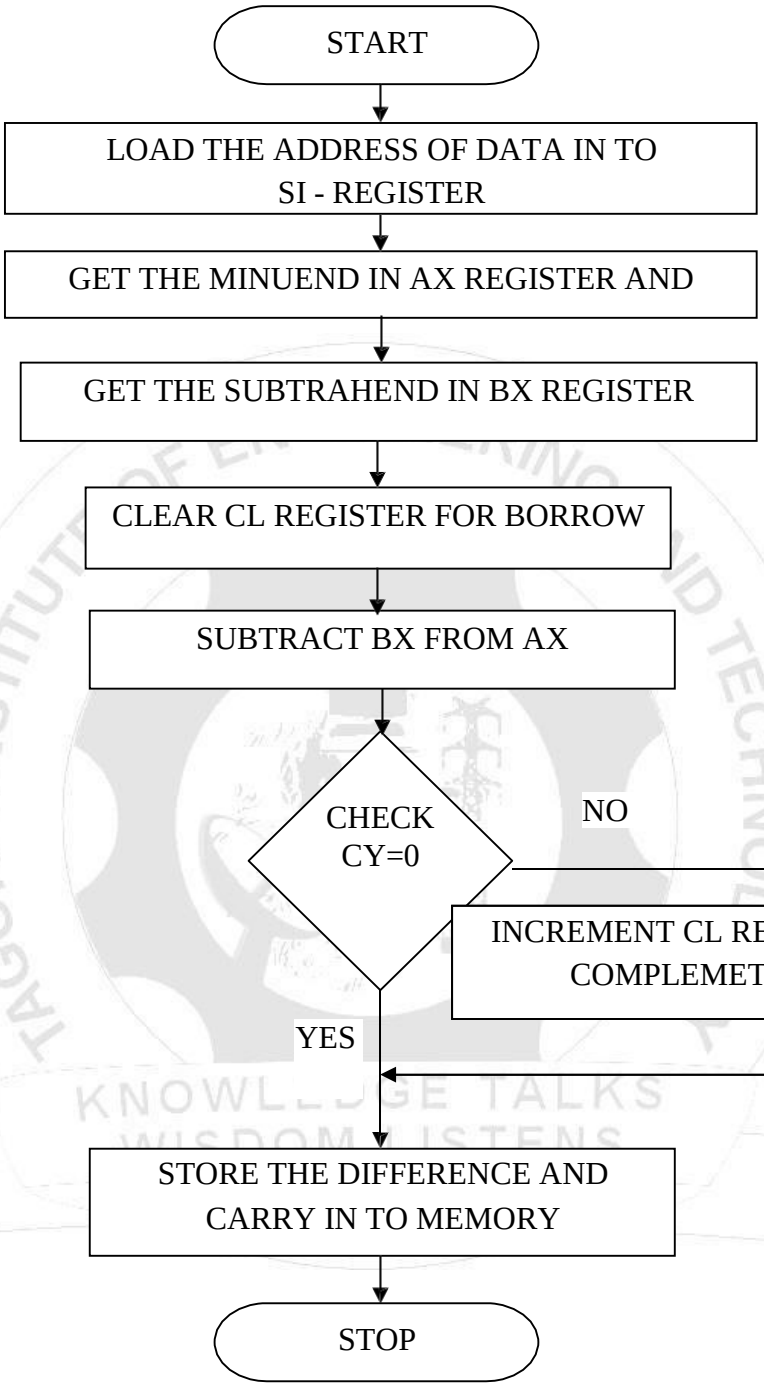
Algorithm:

1. Set SI register as pointer for data.
2. Get the minuend in AX register.
3. Get the subtrahend in BX register.
4. Clear the CL register to account borrow.
5. Subtract the content of BX from AX and the difference will be in AX.
6. Check for carry, if carry 1 go to step 7 else if carry 0 go to step 9.
7. Increment CL register.
8. Take 2's complement of difference in AX register.
9. Store the magnitude of difference and sign into memory.
10. Terminate the program.

Program:

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
2000		MOV SI,1100H	C7 C6 00 11	; Load the address of data in to SI register
2004		MOV AX,[SI]	8B 04	; Get the minuend in AX –register.
2006		MOV BX,[SI+2]	8B 5C 02	; Get the subtrahend in BX register.
2009		MOV CL,00H	C6 C1 00	; Clear the CL register to account borrow.
200C		SUB AX,BX	29 D8	; Subtract the content of BX from AX and the difference will be in AX.
200E		JNC L1	73 08	; If CY=0 go to L1 otherwise do the following
2010		INC CL	FE C1	; Increment CL register.
2012		NOT AX	F7 D0	; Take 1's Complement.
2014		ADD AX, 0001H	81 C0 01 00	; Add 0001H with AX for 2's complement.
2018	L1:	MOV [SI+4],AX	89 44 04	; Store difference into memory location 1104H.
201B		MOV [SI+6],CL	88 4C 06	; Store the borrow into memory location 1106H
201E		HLT	F4	;Terminate the program

Flowchart:


START
LOAD THE ADDRESS OF DATA IN TO SI - REGISTER
GET THE MINUEND IN AX REGISTER AND
GET THE SUBTRAHEND IN BX REGISTER
CLEAR CL REGISTER FOR BORROW
SUBTRACT BX FROM AX
CHECK CY=0
NO
INCREMENT CL REGISTER AND TAKE 2'S COMPLEMET OF AX REGISTER
YES
STORE THE DIFFERENCE AND CARRY IN TO MEMORY
STOP

SAMPLE INPUT AND OUTPUT:

INPUT		OUTPUT	
ADDRESS	**DATA**	**ADDRESS**	**DATA**
1100H	0CH	1104H	BEH
1101H	84H	1105H	2EH
1102H	CAH	1106H	01H
1103H	B2H		

Result:

Thus the assembly language program for subtraction of two 16-bit data was written and executed successfully by using 8086 microprocessor trainer kit.

EXPT NO. : MULTIPLICATION OF TWO 16-BIT DATA

DATE :

Aim:

To write and execute an assembly language program for multiplication of two 16-bit data by using 8086 microprocessor trainer kit.

Apparatus Required:

Sl.No.	Apparatus Required	Quantity
1	8086 Microprocessor Kit	1
2	Power chord	1
3	Keyboard PS/2 port	1

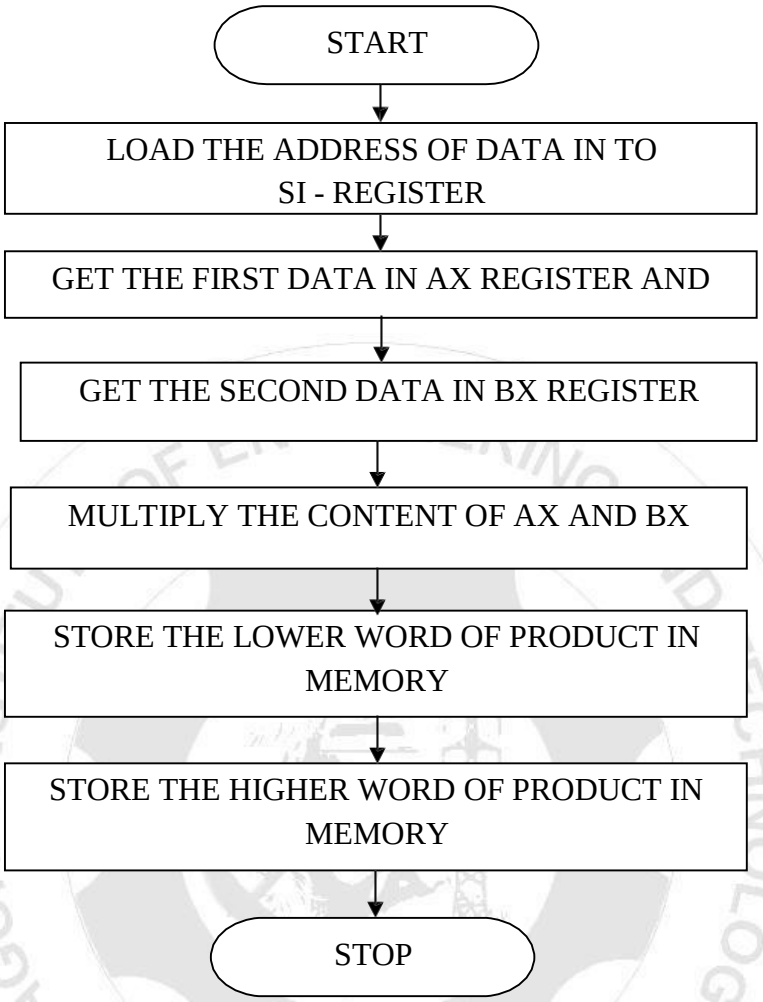
Algorithm:

1. Set SI register as pointer for data.
2. Get the first data in AX register.
3. Get the second data in BX register.
4. Multiply the content of AX and BX register.
5. Store the product (AX &DX)into memory.
6. Terminate the program.

Program:

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
3000		MOV SI,1100H	C7 C6 00 11	; Load the address of data in to SI register
3004		MOV AX,[SI]	8B 04	; Get the first data in AX –register.
3006		MOV BX,[SI+2]	8B 5C 02	; Get the second data in BX register.
3009		MUL BX	F7 E3	; Multiply AX and BX register.
300B		MOV [SI+4],AX	89 44 04	; Store the lower 16-bit of product in Memory.
300E		MOV [SI+6],DX	89 54 06	; Store the higher 16-bit of product in Memory.
3011		HLT	F4	; Terminate the program.

Flowchart:



KNOWLEDGE TALKS
WISDOM LISTENS

SAMPLE INPUT AND OUTPUT:

INPUT		OUTPUT	
ADDRESS	**DATA**	**ADDRESS**	**DATA**
1100H	1AH	1104H	20H
1101H	EFH	1105H	8AH
1102H	50H	1106H	C2H
1103H	CDH	1107H	BFH

Result:

Thus the assembly language program for multiplication of two 16-bit data was written and executed successfully by using 8086 microprocessor trainer kit.

EXPT NO. : DIVISION OF TWO 16-BIT DATA

DATE :

Aim:

To write and execute an assembly language program for division of two 16-bit data by using 8086 microprocessor trainer kit.

Apparatus Required:

Sl.No.	Apparatus Required	Quantity
1	8086 Microprocessor Kit	1
2	Power chord	1
3	Keyboard PS/2 port	1

Algorithm:

1. Copy the data from the address 2000H into AX register.
2. Copy the data from the address 2002H into BX register.
3. Divide the AX register with BX register.
4. Store the quotient (AX) and remainder (DX) into memory.
5. Terminate the program.

Program:

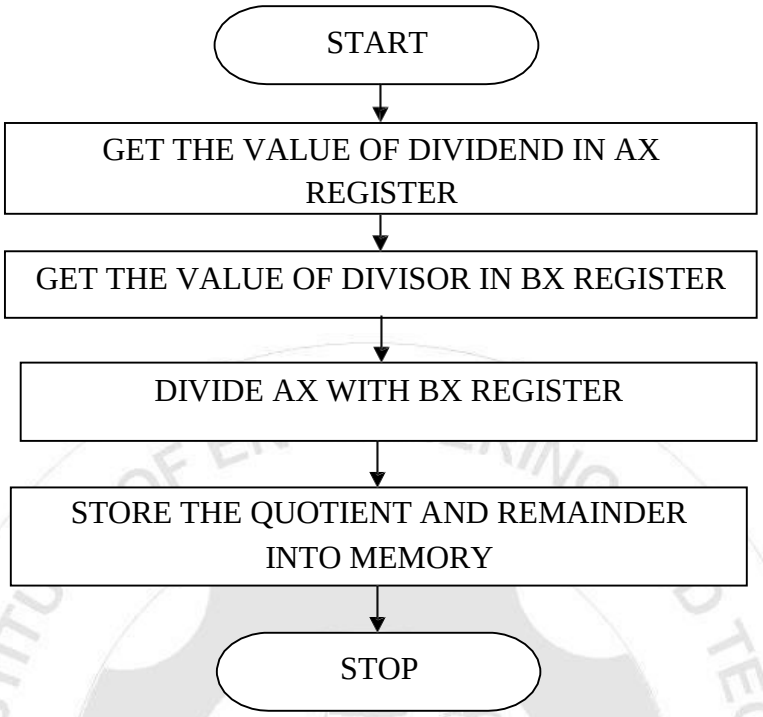
ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
1000		MOV AX,[2000H]	8B 06 00 20	; Get the data in AX register from 2000H
1004		MOV BX,[2002H]	8B 1E 02 20	; Get the data in BX register from 2002H
1008		DIV BX	F7 F3	; Divide the dividend by divisor data.
100A		MOV[2006H],AX	89 06 06 20	; Store Quotient in AX into 2006H address.
100E		MOV [2008H],DX	89 16 08 20	; Store Remainder in DX into 2008H address.
1012		HLT	F4	; Terminate the program.

KNOWLEDGE TALKS
WISDOM LISTENS

SAMPLE INPUT AND OUTPUT:

INPUT		OUTPUT	
ADDRESS	DATA	ADDRESS	DATA
2000H	FFH	2006H	01H
2001H	FFH	2008H	00H
2002H	FFH		
2003H	FFH		

Flowchart:



Result:

Thus the assembly language program for Division of two 16-bit data was written and executed successfully by using 8086 microprocessor trainer kit.

USING MASM

ADDITION,SUBTRACTION,MULTIPLICATION

Aim: To Write and execute an Assembly language Program (ALP) to 8086 processor to add, subtract and multiply two 16 bit unsigned numbers. Store the result in extra segment.

Software Required:

1. Pc with windows(95/98/XP/NT/2000)
2. MASM Software

Programs:

(A) 16-BIT ADDITION

```
ASSUME CS: CODE, DS: DATA, ES: EXTRA
DATA SEGMENT
        NUM1 DW 23C5H
        NUM2 DW 6789H
DATA ENDS
EXTRA SEGMENT
        RESULT DW   0000H
EXTRA ENDS
CODE SEGMENT
START:      MOV AX,DATA
            MOV DS,AX
            MOV AX,EXTRA
            MOV ES,AX
            SUB AX,AX
            MOV AX, NUM1
            MOV BX, NUM2
            ADD AX,BX
            MOV RESULT,AX
            INT 3H
CODE ENDS
END START
```

Result:

INPUT: NUM1 23C5H
NUM2 6789H

OUTPUT: RESULT 8B4EH

(B) 16-BIT SUBTRACTION

```
ASSUME CS:CODE,DS:DATA,ES:EXTRA
DATA SEGMENT
        NUM1 DW   23C5H
        NUM2 DW  1789H
        RESULT DW   0000H
DATA ENDS
EXTRA SEGMENT
        RESULT DW   0000H
EXTRA ENDS
CODE SEGMENT
START:  MOV AX,DATA
        MOV DS,AX
        MOV AX,EXTRA
        MOV ES,AX
        SUB AX,AX
        MOV AX, NUM1
        MOV BX, NUM2
        SUB AX,BX
        MOV RESULT,AX
        INT 3H
CODE ENDS
END START
```

Result:

INPUT: NUM1 23C5H
NUM2 1789H

OUTPUT: RESULT 0C3CH

(C) 16-BIT MULTIPICATION

```
ASSUME CS:CODE,DS:DATA,ES:EXTRA

DATA SEGMENT
        NUM1 DW 23C5H
        NUM2 DW 1789H
DATA ENDS

EXTRA SEGMENT
        RESULT DW   0000H
EXTRA ENDS
CODE SEGMENT
START:      MOV AX,DATA
            MOV DS,AX
            MOV AX,EXTRA
            MOV ES,AX
            SUB AX,AX
            MOV AX, NUM1
            MOV BX, NUM2
             MUL AX,BX
            MOV DI,OFFSET RESULT
            MOV [DI],AX
            MOV RESULT,AX
            INT 3H
CODE ENDS
END START
```

Result:

INPUT:	NUM1	23C5H
	NUM2	1789H
OUTPUT:	DX:AX	0349D76DH

(D)DIVISION

Aim: To Write and execute an Assembly language Program (ALP) to 8086 processor to divide a 32 bit unsigned number by a 16 bit unsigned numbers. Store the result in stack segment.

Software Required:

1. Pc with windows(95/98/XP/NT/2000)
2. MASM Software

PROGRAM:

```
ASSUME CS:CODE,DS:DATA,SS:STACK
DATA SEGMENT
        NUM1 DD   2A2CFA6H
        NUM2 DW   590H
DATA ENDS

STACK SEGMENT
        QUOITENT DW ?
        REMAINDER DW ?
STACK ENDS

CODE SEGMENT
START:      MOV AX,DATA
            MOV DS,AX
            MOV AX,STACK
            MOV SS,AX
            SUB AX,AX
            MOV SI,OFFSET  NUM1
            MOV AX,[SI]
            INC SI
            INC SI
```

```
        MOV DX,[SI]
        MOV BX, NUM2
        DIV BX
        MOV QUOITENT,AX
        MOV REMAINDER,DX
        INT 3H
CODE ENDS
END START
```

Result:

INPUT: NUM1 2A2CFA6H

NUM2 590H

OUTPUT: AX: DX:

EXPT NO. : MOVE A DATA BLOCK WITHOUT OVERLAP

DATE :

Aim:

To write and execute an assembly program for moving a data block without overlap using MASM Software.

Software Required:

1. Pc with windows(95/98/XP/NT/2000)
2. MASM SOFTWARE

Program:

```
.MODEL SMALL
.DATA
        Array1 DW 1111h,2222h,3333h,4444h,5555h
        Array2 DW 5 DUP (0)
        Count DW 0005H
.CODE
        MOV AX,@DATA
        MOV DS,AX
        LEA SI,Array1
        LEA DI,Array2
        MOV CX,Count

NEXT:   MOV AX,[SI]
        MOV [DI],AX
        INC SI
        INC SI
        INC DI
        INC DI
        LOOP NEXT

        MOV AH,4Ch
        INT 21h
END
```

Result:

Thus the assembly language program for 16-bit data was written and executed successfully by using 8086 microprocessor trainer kit.

EXPT NO. : BINARY TO BCD CODE CONVERSION

DATE :

Aim:

To write and execute an assembly program for converting binary to BCD code by using 8086 microprocessor trainer kit.

Apparatus Required:

Sl.No.	Apparatus Required	Quantity
1	8086 Microprocessor trainer kit	1
2	Keyboard PS/2 port	1
3	Power chord	1

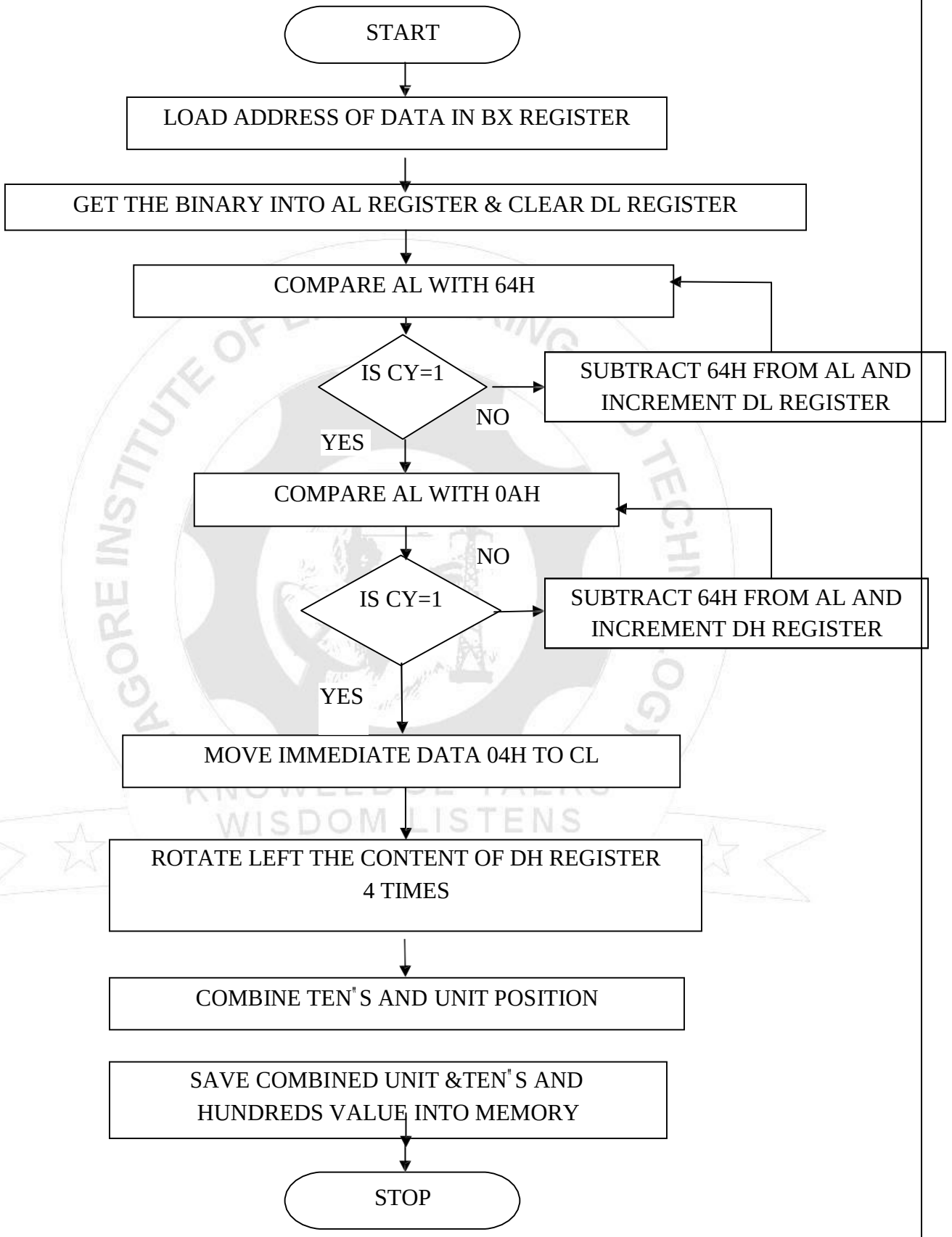
Algorithm:

1. Load the address of the data in BX register.
2. Get the binary data in AL register.
3. Clear DX for starting hundreds and tens.
4. Compare whether data is than 100(64H), if less than 100 go to step 6 otherwise do the following.
5. Subtract 100(64H) and increment 100 Register, repeat it until no more hundred.
6. Compare whether data is than 10(0AH), if less than 10 go to step 8 otherwise do the following.
7. Subtract 10(0AH) and increment 10 Register; repeat it until no more Ten.
8. Rotate 10"s counter to upper nibble position.
9. Combine units and tens counter and save in it and hundred counter into memory.
10. Terminate the program.

Program:

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
1000		MOV BX,1100H	C7 C3 00 11	; Load the address of the data in BX register.
1004		MOV AL,[BX]	8A 07	; Get the binary data in AL register.
1006		MOV DX,0000H	C7 C2 00 00	; Clear DX for counting 100"s and 10"s.
100A	L2:	CMP AL,64H	80 F8 64	; Compare whether data is than 100.
100D		JC L1	72 08	; if data is less than 100 go to L1
100F		SUB AL,64H	80 E8 64	; Subtract 100.
1012		INC DL	FE C2	; Increment Hundred register.
1014		JMP L2	E9 EA EF	; Go to L2.
1017	L1:	CMP AL,0AH	80 F8 0A	; Compare whether data is than 10.
101A		JC L3	72 08	; if data is less than 10 go to L3
101C		SUB AL,0AH	80 E8 0A	; Subtract 10.
101F		INC DH	FE C6	; Increment Ten register.
1021		JMP L1	E9 F3 FF	; Go to L1.
1024	L3:	MOV CL,04H	C6 C1 04	; Move 04H immediate data to CL register.
1027		ROL DH,CL	D2 C6	; Rotate the 10"s counts to Upper nibble position.
1029		ADD AL,DH	00 F0	; Combine Ten"s and Unit position.
102B		MOV[BX+1],AL	88 47 01	; Save Ten"s and Unit into memory.
102E		MOV[BX+2],DL	88 57 02	; Save hundreds into memory.
1031		HLT	F4	; Terminate the program.

Flowchart:


START
LOAD ADDRESS OF DATA IN BX REGISTER
GET THE BINARY INTO AL REGISTER & CLEAR DL REGISTER
COMPARE AL WITH 64H
IS CY=1
NO
SUBTRACT 64H FROM AL AND INCREMENT DL REGISTER
YES
COMPARE AL WITH 0AH
NO
IS CY=1
SUBTRACT 64H FROM AL AND INCREMENT DH REGISTER
YES
MOVE IMMEDIATE DATA 04H TO CL
ROTATE LEFT THE CONTENT OF DH REGISTER 4 TIMES
COMBINE TEN"S AND UNIT POSITION
SAVE COMBINED UNIT &TEN"S AND HUNDREDS VALUE INTO MEMORY
STOP

SAMPLE INPUT AND OUTPUT:

INPUT		OUTPUT	
ADDRESS	**DATA**	**ADDRESS**	**DATA**
1100	E4H	1101	28H
		1102	02H

Result:

Thus the assembly language program for converting binary to BCD code was written and executed successfully by using 8086 microprocessor trainer kit.

EXPT NO : BCD TO BINARY CODE CONVERSION

DATE :

Aim:

To write and execute an assembly program for converting BCD to binary code by using 8086 microprocessor trainer kit.

Apparatus Required:

Sl.No.	Apparatus Required	Quantity
1	8086 Microprocessor trainer kit	1
2	Keyboard PS/2 port	1
3	Power chord	1

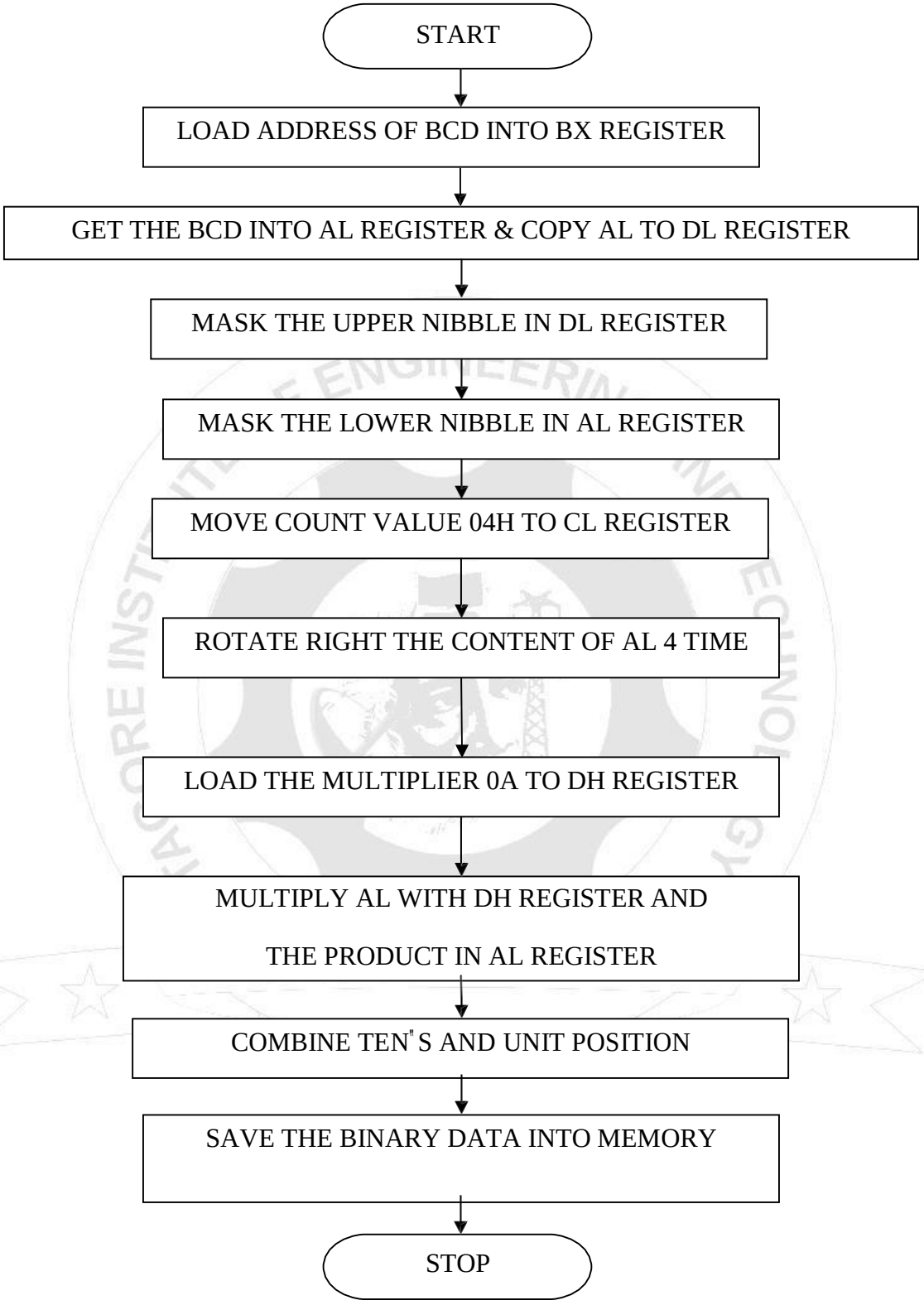
Algorithm:

1. Load the address of the data in BX register.
2. Get the binary data in AL register.
3. Copy AL to DL register.
4. Mask the upper nibble in DL and lower nibble in AL.
5. Move immediate data 04H into CL register and rotate right AL by 4 times.
6. Move DH by 0AH and multiply DH with AL register.
7. Add AL and DL and store result into memory.
8. Terminate the program.

Program:

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
1000		MOV BX,1100H	C7 C3 00 11	; Load the address of the data in BX register.
1004		MOV AL,[BX]	8A 07	; Get the binary data in AL register.
1006		MOV DL,AL	88 C2	; Copy data in AL to DL register.
1008		AND DL,0FH	80 E2 0F	; Mask the upper nibble.
100B		AND AL,0F0H	80 E0 F0	; Mask the lower nibble.
100E		MOV CL,04H	C6 C1 04	; Move the immediate data 04H to CL register.
1011		ROR AL,CL	D2 C8	; Rotate the upper nibble to lower nibble.
1013		MOV DH,0AH	C6 C6 0A	; set multiplier as 0AH.
1016		MUL DH	F6 E6	; Multiply DH with AL register.
1018		ADD AL,DL	00 D0	; Add AL and DL register.
101A		MOV[BX+1],AL	88 47 01	; Save the binary data into memory.
101D		HLT	F4	; Terminate the program.

Flowchart:


START
LOAD ADDRESS OF BCD INTO BX REGISTER
GET THE BCD INTO AL REGISTER & COPY AL TO DL REGISTER
MASK THE UPPER NIBBLE IN DL REGISTER
MASK THE LOWER NIBBLE IN AL REGISTER
MOVE COUNT VALUE 04H TO CL REGISTER
ROTATE RIGHT THE CONTENT OF AL 4 TIME
LOAD THE MULTIPLIER 0A TO DH REGISTER
MULTIPLY AL WITH DH REGISTER AND
THE PRODUCT IN AL REGISTER
COMBINE TEN" S AND UNIT POSITION
SAVE THE BINARY DATA INTO MEMORY
STOP

SAMPLE INPUT AND OUTPUT:

INPUT		OUTPUT	
ADDRESS	**DATA**	**ADDRESS**	**DATA**
1100	75H	1101	4BH

Result:

Thus the assembly language program for converting BCD to binary code was written and executed successfully by using 8086 microprocessor trainer kit.

EXPT NO. : BINARY TO ASCII CODE CONVERSION

DATE :

Aim:

To write and execute an assembly program for converting binary to ASCII code by using 8086 microprocessor trainer kit.

Apparatus Required:

Sl.No.	Apparatus Required	Quantity
1	8086 Microprocessor trainer kit	1
2	Keyboard PS/2 port	1
3	Power chord	1

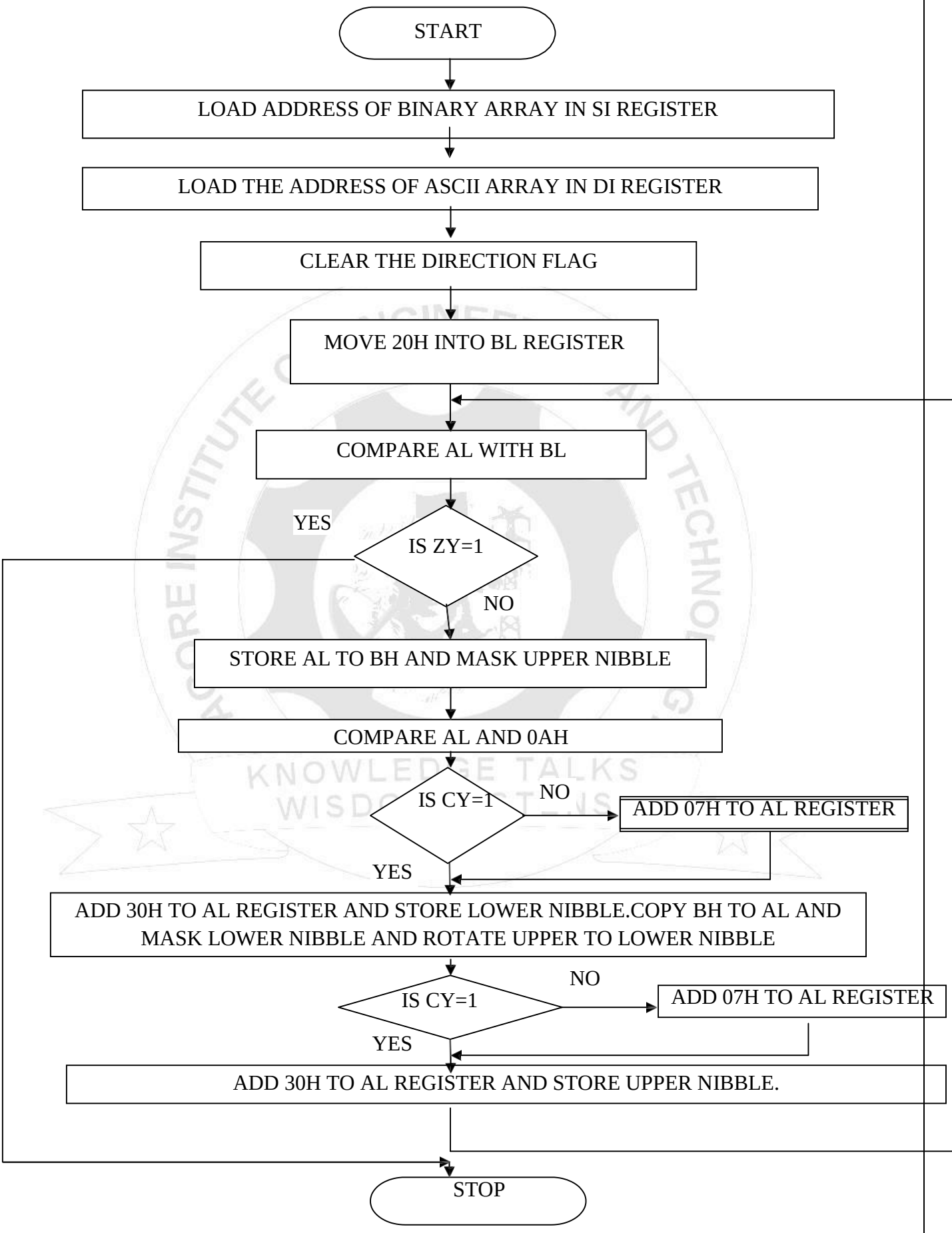
Algorithm:

1. Set SI for Binary array and DI for ASCII array.
2. Clear DF for auto increment of SI & DI.
3. Load the end of character in BL register.
4. Load the byte of binary into AL register and check for end of string.
5. If ZF=1 then go to step 13 otherwise do the following.
6. Mask the upper nibble of binary data.
7. Check whether lower nibble less 0AH then add 30H with lower nibble otherwise add 37H with lower nibble.
8. Store ASCII code for lower nibble into memory.
9. Rotate upper nibble into lower nibble position.
10. Check whether lower nibble less 0AH then add 30H with lower nibble otherwise add 37H with lower nibble.
11. Store the upper nibble into memory.
12. Go to step 4.
13. Terminate the program.

Program:

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
1000		MOV SI,1100H	C7 C6 00 11	; Set SI pointer for binary array.
1004		MOV DI,1400H	C7 C7 00 14	; Set DI pointer for ASCII array.
1008		CLD	FC	; Clear DF for auto increment SI&DI.
1009		MOV BL,20H	C6 C3 20	; Load the end of character in BL register.
100C	L1:	LODSB	AC	; Load byte of binary data into AL register.
100D		CMP AL,BL	38 D8	; Check for end of string.
100F		JE L2	74 2A	; If ZF=1 then go to L2, otherwise continue.
1011		MOV BH,AL	88 C7	; Save the binary data into BH Register.
1013		AND AL,0FH	80 E0 0F	; Mask the upper nibble of binary data.
1016		CMP AL,0AH	80 F8 0A	; Check AL is less than 0AH
1019		JL L3	7C 03	; If AL is less than 0AH go to L3.
101B		ADD AL,07H	80 C0 07	; If AL is greater than 0AH add 07H to AL register.
101E	L3:	ADD AL,30H	80 C0 30	; Add 30H to AL register.
1021		STOSB	AA	; Store lower nibble of ASCII.
1022		MOV AL,BH	88 F8	; Get the data into AL register.
1024		AND AL,F0H	80 E0 F0	; Mask the lower nibble.
1027		MOV CL,04H	C6 C1 04	; Move immediate data 04H to CL register.
102A		ROL AL,CL	D2 C0	; Rotate the upper nibble to lower nibble position.
102C		CMP AL,0AH	80 F8 0A	; Check AL is less than 0AH
102F		JL L4	7C 03	; If AL is less than 0AH go to L4.
1031		ADD AL,07H	80 C0 07	; If AL is greater than 0AH add 07H to AL register.
1034	L4:	ADD AL,30H	80 C0 30	; Add 30H to AL register.
SAMPLE INPUT AND OUTPUT:			AA	; Store upper nibble of ASCII.
1037		STO SB		
1038		JMP L1	E9 D1 FF	; Go to L1.
103B	L2:	HLT	F4	; Terminate the program.

FLOWCHART:


START
LOAD ADDRESS OF BINARY ARRAY IN SI REGISTER
LOAD THE ADDRESS OF ASCII ARRAY IN DI REGISTER
CLEAR THE DIRECTION FLAG
MOVE 20H INTO BL REGISTER
COMPARE AL WITH BL
YES
IS ZY=1
NO
STORE AL TO BH AND MASK UPPER NIBBLE
COMPARE AL AND 0AH
IS CY=1
NO
ADD 07H TO AL REGISTER
YES
ADD 30H TO AL REGISTER AND STORE LOWER NIBBLE.COPY BH TO AL AND MASK LOWER NIBBLE AND ROTATE UPPER TO LOWER NIBBLE
IS CY=1
NO
ADD 07H TO AL REGISTER
YES
ADD 30H TO AL REGISTER AND STORE UPPER NIBBLE.
STOP

SAMPLE OUTPUT

INPUT		OUTPUT	
ADDRESS	**DATA**	**ADDRESS**	**DATA**
1100	4EH	1400	45
1101	15H	1401	34
1102	87H	1402	35
1103	C0H	1403	31
1104	20H	1404	37
		1405	38
		1406	30
		1407	43

Result:

Thus the assembly language program for converting binary to ASCII code was written and executed successfully by using 8086 microprocessor trainer kit.

EXPT NO. : ASCII TO BINARY CODE CONVERSION
DATE :

Aim:

To write and execute an assembly program for converting ASCII to binary code by using 8086 microprocessor trainer kit.

Apparatus Required:

Sl.No.	Apparatus Required	Quantity
1	8086 Microprocessor trainer kit	1
2	Keyboard PS/2 port	1
3	Power chord	1

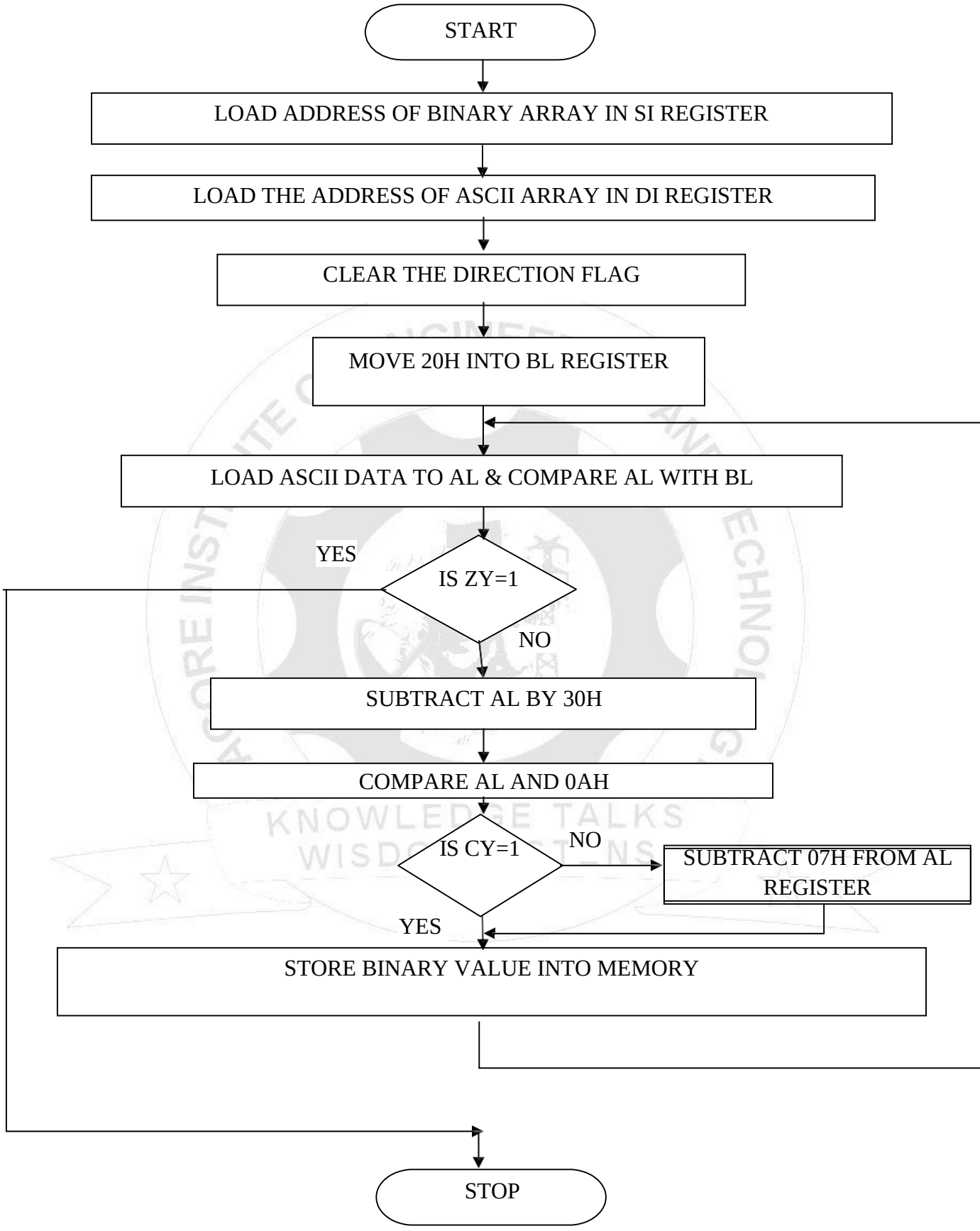
Algorithm:

1. Set SI for Binary array and DI for ASCII array.
2. Clear DF for auto increment of SI & DI.
3. Load the end of character in BL register.
4. Load the byte of binary into AL register and check for end of string.
5. If ZF=1 then go to step 10 otherwise do the following.
6. Subtract AL with 30H.
7. Check whether AL less 0AH then go to step 8 otherwise subtract 07H with AL.
8. Store binary code into memory.
9. Go to step 4.
10. Terminate the program.

Program:

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
1000		MOV SI,1100H	C7 C6 00 11	; Set SI pointer for binary array.
1004		MOV DI,1400H	C7 C7 00 14	; Set DI pointer for ASCII array.
1008		CLD	FC	; Clear DF for auto increment SI&DI.
1009		MOV BL,20H	C6 C3 20	; Load the end of character in BL register.
100C	L3:	LODSB	AC	; Load byte of binary data into AL register.
100D		CMP AL,BL	38 D8	; Check for end of string.
100F		JE L1	74 0C	; If ZF=1 then go to L1, otherwise continue.
1011		SUB AL,30H	80 E8 30	; Subtract 30H from AL register.
1014		CMP AL,0AH	80 F8 0A	; Compare Al and 0AH.
1017		JC L2	72 03	; Check AL is less than 0AH
1019		SUB AL,07H	80 E8 07	; Subtract 07H from AL register.
101C	L2:	STOSB	AA	; Store binary value into memory.
101D		JMP L3	E9 ED FF	; Go to L3.
1020	L1:	HLT	F4	; Terminate the program.

FLOW CHART:


START
LOAD ADDRESS OF BINARY ARRAY IN SI REGISTER
LOAD THE ADDRESS OF ASCII ARRAY IN DI REGISTER
CLEAR THE DIRECTION FLAG
MOVE 20H INTO BL REGISTER
LOAD ASCII DATA TO AL & COMPARE AL WITH BL
YES
IS ZY=1
NO
SUBTRACT AL BY 30H
COMPARE AL AND 0AH
IS CY=1
NO
SUBTRACT 07H FROM AL REGISTER
YES
STORE BINARY VALUE INTO MEMORY
STOP

SAMPLE INPUT AND OUTPUT:

INPUT		OUTPUT	
ADDRESS	**DATA**	**ADDRESS**	**DATA**
1100	42H	1400	0BH
1101	37H	1401	07H
1102	46H	1402	0FH
1103	39H	1403	09H
1104	38H	1404	08H
1105	20H		

Result:

Thus the assembly language program for converting ASCII to binary code was written and executed successfully by using 8086 microprocessor trainer kit.

EXPT NO. : SEARCH SMALLEST ELEMENT IN AN ARRAY
DATE :

Aim:

To write and execute an assembly language program for searching smallest element in an array of data by using 8086 microprocessor trainer kit.

Apparatus Required:

Sl.No.	Apparatus Required	Quantity
1	8086 Microprocessor Kit	1
2	Power chord	1
3	Keyboard PS/2 port	1

Algorithm:

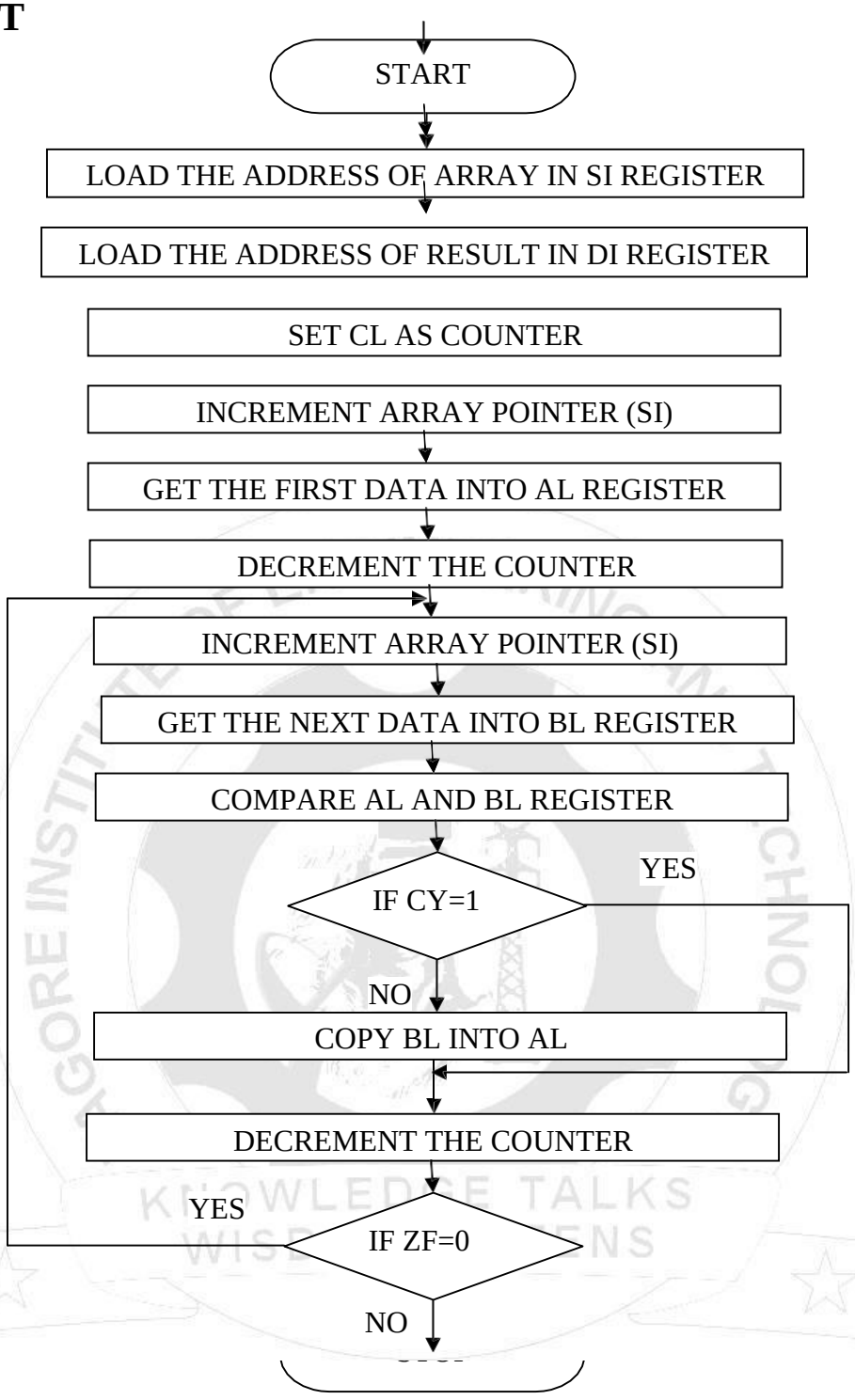
1. Set SI register pointer for array.
2. Set DI register pointer for result.
3. Set CL register as count for element in the array.
4. Increment the address pointer (SI).
5. Set first data as smallest.
6. Decrement the count.
7. Increment the address pointer (SI).
8. Get the next data into BL register.
9. Compare current smallest data in AL with BL register.
10. If carry is set (AL is less than BL) go to step 12, otherwise do the following.
11. If BL is small then copy BL into AL register.
12. Decrement the count.
13. If counter is not zero then go to step 7, otherwise do the following.
14. Store the smallest data into memory.
15. Terminate the program.

Program:

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
1000		MOV SI,1100H	C7 C6 00 11	; Set SI register as pointer for array.
1004		MOV DI,1200H	C7 C7 00 12	; Set DI register as pointer for result.
1008		MOV CL,[SI]	8A 0C	; Set CL as count for elements in the array.
100A		INC SI	46	; Increment the address pointer.
100B		MOV AL,[SI]	8A 04	; Set first data as smallest.
100D		DEC CL	FE C9	; Decrement the count.
100F	L1	INC SI	46	; Make SI pointer points the next data in array.
1010		MOV BL,[SI]	8A 1C	; Get the next data in BL register.
1012		CMP AL,BL	38 D8	; Compare current smallest data in AL with BL.
1014		JC L2	72 02	; If carry is set then AL is less than BL, Hence proceed to L2.
1016		MOV AL,BL	88 D8	; If carry is not set, then make BL as current smallest.
1018	L2	DEC CL	FE C9	; Decrement the count.

101A		JNZ L1	75 F3	; If count is not zero repeat search.
101C		MOV[DI],AL	88 05	; Store the smallest data in memory.
101E		HLT	F4	; Terminate the program.

FLOWCHART



SAMPLE INPUT AND OUTPUT:

INPUT		OUTPUT	
ADDRESS	**DATA**	**ADDRESS**	**DATA**
1100H	06H	1200H	2DH
1101H	4EH		
1102H	2DH		
1103H	30H		
1104H	98H		
1105H	ACH		
1106H	FEH		

Result:

Thus the assembly language program for searching smallest element in an array of data was written and executed successfully by using 8086 microprocessor trainer kit.

EXPT NO. : SORTING OF AN ARRAY
DATE :

Aim:

To write and execute an assembly language program for sorting an array of data by using 8086 microprocessor trainer kit.

Apparatus Required:

Sl.No.	Apparatus Required	Quantity
1	8086 Microprocessor Kit	1
2	Power chord	1
3	Keyboard PS/2 port	1

Algorithm:

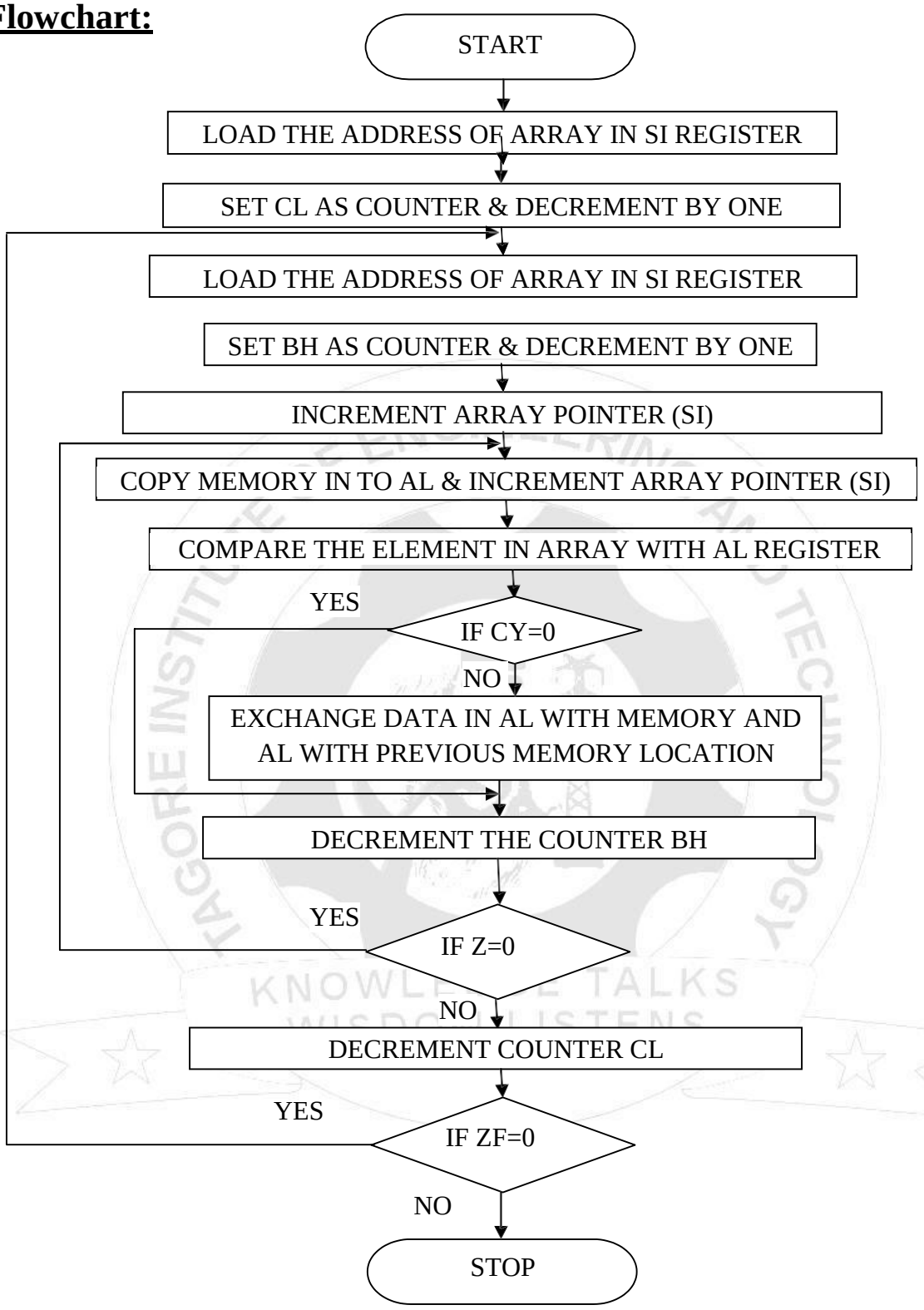
1. Set SI register pointer for array and Set CL register as count for N-1 repetitions.
2. Decrement the count (CL).
3. Set SI register pointer for array and set BH register as count for N-1 repetitions.
4. Decrement the count (BH).
5. Increment the address pointer (SI) and Get the first element into AL register.
6. Increment the address pointer (SI).
7. Compare with next element of array in memory.
8. If AL is greater than memory then go to step13.
9. If AL is less than memory exchange, current memory and previous memory.
10. Decrement the count (BH) and check BH is not zero go to step 8, otherwise do the following.
11. Decrement the count (CL).
12. If CL is not zero go to step 4, otherwise do the following.
13. Terminate the program.

Program:

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
1000		MOV SI,1100H	C7 C6 00 11	; Set SI register as pointer for array.
1004		MOV CL,[SI]	8A 0C	; Set CL as count for elements in the array.
1006		DEC CL	FE C9	; Decrement the count.
1008	L3:	MOV SI,1100H	C7 C6 00 11	; Set SI register as pointer for array.
100C		MOV BH,[SI]	8A 3C	; Set BH as count for elements in the array.
100E		DEC BH	FE CF	; Decrement the count.
1010		INC SI	46	; Increment the address pointer.
1011	L2:	MOV AL,[SI]	8A 04	; copy data into AL register.
1013		INC SI	46	; Make SI pointer points the next data in array.
1014		CMP AL,[SI]	3A 04	; Compare current data in AL with memory.
1016		JNC L1	73 05	; If carry is reset go to L1, otherwise do the following.
1018		XCHG AL,[SI]	86 04	; Exchange AL with SI pointed memory.
101A		XCHG AL,[SI-1]	86 44 FF	; Exchange AL with SI-1 pointed memory.
101D	L1:	DEC BH	FE CF	; Decrement the count.

101F		JNZ L2	75 F0	; If zero is reset go to L2, otherwise do the following.
1021		DEC CL	FE C9	; Decrement the count.
1023		JNZ L3	75 F3	; If count is not zero repeat search.
1025		HLT	F4	; Terminate the program.

Flowchart:



SAMPLE INPUT AND OUTPUT:

INPUT		OUTPUT	
ADDRESS	**DATA**	**ADDRESS**	**DATA**
1100H	07H	1101H	FFH
1101H	AAH	1102H	BBH
1102H	77H	1103H	AAH
1103H	FFH	1104H	77H
1104H	22H	1105H	44H
1105H	11H	1106H	22H
1106H	44H	1107H	11H
1107H	BBH		

Result:

Thus the assembly language program for sorting of an array of data was written and executed successfully by using 8086 microprocessor trainer kit.

EXPT NO. : SORTING USING MASM

DATE :

Aim: To Write and execute an Assembly language Program (ALP) to 8086 processor to sort the given array of numbers in ascending and decending order.

Software Required:

1. Pc with windows(95/98/XP/NT/2000)
2. MASM Software

Programs:

(A) ASCENDING ORDER

```
ASSUME CS:CODE,DS:DATA
DATA SEGMENT
        ARRAY DB 0500H,0300H,0001H,0004H,0002H
        COUNT DB 05H
DATA  ENDS
CODE  SEGMENT
START:      MOV AX,DATA
            MOV DS,AX
            SUB AX,AX
            MOV DX,COUNT-1
            AGAIN:MOV CX,DX
            MOV SI,OFFSET ARRAY
BACK:       MOV AX,[SI]
            CMP AX,[SI+2]
            JBE FORW
            XCHG AX,[SI+2]
            MOV [SI],AX
FORW:       INC SI
            INC SI
            LOOP BACK
            DEC DX
            JNZ AGAIN
            INT 03H
CODE ENDS
END START
```

EXPT NO. : STRING MANIPULATION – COPY A STRING

DATE :

Aim:

To write and execute an assembly language program for copying a string by using 8086 microprocessor trainer kit.

Apparatus Required:

Sl.No.	Apparatus Required	Quantity
1	8086 Microprocessor Kit	1
2	Power chord	1
3	Keyboard PS/2 port	1

Algorithm:

1. Initialize source address into SI.
2. Initialize destination address into DI.
3. Initialize array size.
4. Clear direction flag.
5. Copy the contents of source into destination until count reaches zero.
6. Terminate the program.

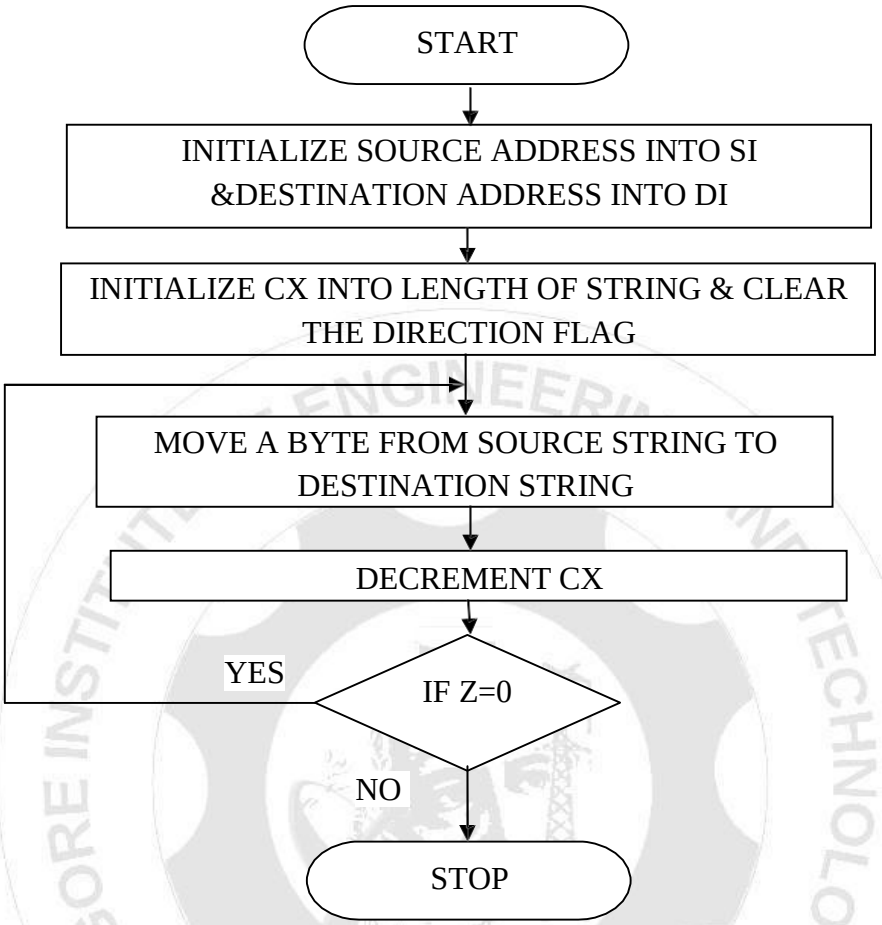
Program:

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
1000		MOV SI,1200H	C7 C6 00 12	; Initialize source address into SI
1004		MOV DI,1300H	C7 C7 00 13	; Initialize destination address into DI
1008		MOV CX,0006H	C7 C1 06 00	; Initialize array size
100C		CLD	FC	; Clear direction flag
100D		REP MOVSB	F3 A4	; Copy the contents of source into destination until count reaches zero
100F		HLT	F4	; Terminate the program.

SAMPLE INPUT AND OUTPUT:

INPUT		OUTPUT	
ADDRESS	DATA	ADDRESS	DATA
1200H	02H	1300H	02H
1201H	01H	1301H	01H
1202H	02H	1302H	02H

Flowchart:



Result:

Thus the assembly language program for copying a string was written and executed successfully by using 8086 microprocessor trainer kit.

EXPT NO. : DISPLAY A STRING IN TO MONITOR
DATE :

Aim:

To write an assembly language program for displaying string manipulation into the monitor in 8086 using MASM software.

Apparatus Required:

Sl.No.	Apparatus Required	Quantity
1	System	1
2	MASM Software	1

PROCEDURE:

1. Start the program.
2. In the code segment initialize the address of 1200h to DX register.
3. Interrupt command is used to display data on monitor.
4. Perform the termination using terminate interrupt for program.
5. Stop the program.

PROGRAM:

```
Code segment

        Assume cs: code, ds: code

        Org 1000h

        Mov ah, 09h

        Mov dx, 1200h

        Int 21h

        Mov ah, 4ch

        Int 21h

        Org 1200h

        Db "VI micro$"

Code    ends

        End
```

OUTPUT:

VI micro

RESULT: Thus the assembly language program for displaying string manipulation into the monitor in 8086 using MASM software was written and executed successfully and the output was verified.

EXPT NO. : **FILE MANIPULATION**

DATE :

Aim:

To write an assembly language program for file manipulation in 8086 using MASM software.

Apparatus Required:

Sl.No.	Apparatus Required	Quantity
1	System	1
2	MASM Software	1

PROCEDURE:

1. Start the program.
2. In the data segment define byte for file.
3. In code segment, move the data segment values to DS register.
4. Initialize the address of 1200h to DX register.
5. Interrupt command (4ch) is used to open file.
6. Perform the termination using terminate interrupt for the program.
7. Stop the program.

PROGRAM:

```
Code    segment
assumecs:code,ds:code
org 1000h
movax,data
movds,ax
mov ah,5bh
mov dx,1200h
mov cx,0
int 21h
mov ah,4ch
int 21h
code    ends
data    segment
org 1200h
db 'e:\man\v.asm'
data    ends
        End
```

OUTPUT:

'v.asm' file created in **'E'** drive **'man'** folder.

RESULT: Thus the assembly language program for file manipulation in 8086 using MASM software was written and executed successfully and the output was verified.

EXPT NO. : READ CHARACTER CODE FROM KEYBOARD

DATE :

Aim:

To write an assembly language program to read scan code and character code from a keyboard control in 8086 using MASM software.

Apparatus Required:

Sl.No.	Apparatus Required	Quantity
1	System	1
2	MASM Software	1

PROCEDURE:

1. Start the program.
2. Initialize the address of 1200h to SI register.
3. Interrupt command (4ch) is used to open file.
4. Perform the termination using terminate interrupt for the program.
5. Stop the program.

PROGRAM:

```
Code    segment
        Assume cs: code, ds: code
        Org  1000h
        Mov ah, 0h
        Int 16h
        Movsi, 1200h
        Mov [si], al
        Incsi
        Mov  [si],  ah
        Mov  ah,  4ch
        Int 21h
Code ends
        End
```

OUTPUT:

Display the character code for **'a'** is **'61'.**

RESULT:

Thus the assembly language program to read scan code and character code from a keyboard control in 8086 using MASM software was written and executed successfully and the output was verified.

EXPT NO. : TO INSERT OR DELETE A STRING

DATE

Aim: To Write and execute an Assembly language Program (ALP) to 8086 processor to insert or delete a character/ number from the given string.

Software Required:

Pc with windows(95/98/XP/NT/2000)

MASM Software

Program:

(A) STRING INSERTION

```
ASSUME CS:CODE,DS:DATA,ES:EXTRA
DATA SEGMENT
        STRING1 DB 'MICROPROCESSOR INTERFACING LAB$'
        STRING2 DB 'AND '
        STRLEN EQU ($-STRING1)
DATA ENDS
EXTRA SEGMENT
        STRING3 DB 38 DUP(0)
EXTRA ENDS
CODE SEGMENT
START:  MOV AX,DATA
        MOV DS,AX
        MOV AX,EXTRA
        MOV ES,AX
        MOV SI,OFFSET STRING1
        MOV DI,OFFSET STRING3
        CLD
        MOV CX,15
        REP MOVSB
        CLD
        MOV SI,OFFSET STRING2
        MOV CX,4
        REP MOVSB
        MOV SI,OFFSET STRING1
```

```
        MOV SI,15
        MOV CX,15
        REP MOVSB
        INT 3H
CODE ENDS
END START
```

Result:

INPUT: **STRING1:** 'MICROPROCESSOR INTERFACING LAB'

STRING2: 'AND '

OUTPUT: **STRING3:** 'MICROPROCESSOR AND INTERFACING LAB'

(B) STRING DELETION

```
ASSUME CS:CODE,DS:DATA,ES:EXTRA
DATA SEGMENT
        STRING1 DB 'MICROPROCESSOR AND INTERFACING LAB$'
        STRLEN EQU ($-STRING1)
DATA ENDS
EXTRA SEGMENT
        STRING2 DB STRLEN-3 DUP(0)
EXTRA ENDS
CODE SEGMENT
START:  MOV AX,DATA
        MOV DS,AX
        MOV AX,EXTRA
        MOV ES,AX
        MOV SI,OFFSET STRING1
        MOV DI,OFFSET STRING2
        CLD
        MOV CX,15
        REP MOVSB
        CLD
```

```
        MOV SI,19
        MOV CX,15
        REP MOVSB
        INT 3H
CODE ENDS
END START
```

Result:

INPUT: **STRING1:** MICROPROCESSOR AND INTERFACING LAB'

OUTPUT: **STRING2:** 'MICROPROCESSOR INTERFACING LAB'

EXPT NO. : TO CHECK PASSWORD

Aim: To Write and execute an Assembly language Program (ALP) to 8086 processor verify the password.

Software Required:

Pc with windows(95/98/XP/NT/2000)

MASM Software

Program:

```
ASSUME CS:CODE,DS:DATA,ES:EXTRA
DATA SEGMENT
        STRING1 DB 'MRCET'
        STRLEN EQU ($-STRING1)
        SANE DB 'STRINGS ARE UNEQUAL$'
        SAE DB 'STRINGS ARE EQUAL$'
DATA ENDS
EXTRA SEGMENT
        STRING2 DB 'MRCET'
EXTRA ENDS
CODE SEGMENT
START:  MOV AX,DATA
        MOV DS,AX
        MOV AX,EXTRA
        MOV ES,AX
        MOV SI,OFFSET STRING1
        MOV DI,OFFSET STRING2
        CLD
        MOV CX,STRLEN
        REP CMPSB
        JZ GO
        MOV AH,09H
        MOV DX,OFFSET SANE
        INT 21H
        JMP EXITP
```

```
GO:     MOV AH,09H
        MOV DX,OFFSET SAE
        INT 21H
EXITP:  INT 03H
CODE ENDS
END START
```

Result:

TAGORE INSTITUTE OF ENGINEERING AND TECHNOLOGY
KNOWLEDGE TALKS
WISDOM LISTENS

EXPT NO. : INTERFACING STEPPER MOTOR WITH 8086

DATE :

Aim:

To write and execute an assembly program for stepper motor speed control by using 8086 microprocessor trainer kit.

Apparatus Required:

Sl.No.	Apparatus Required	Quantity
1	8086 Microprocessor trainer kit	1
2	Keyboard PS/2 port	1
3	Stepper motor interface kit with Motor	1
4	FRC connector	1
5	Power chord	1

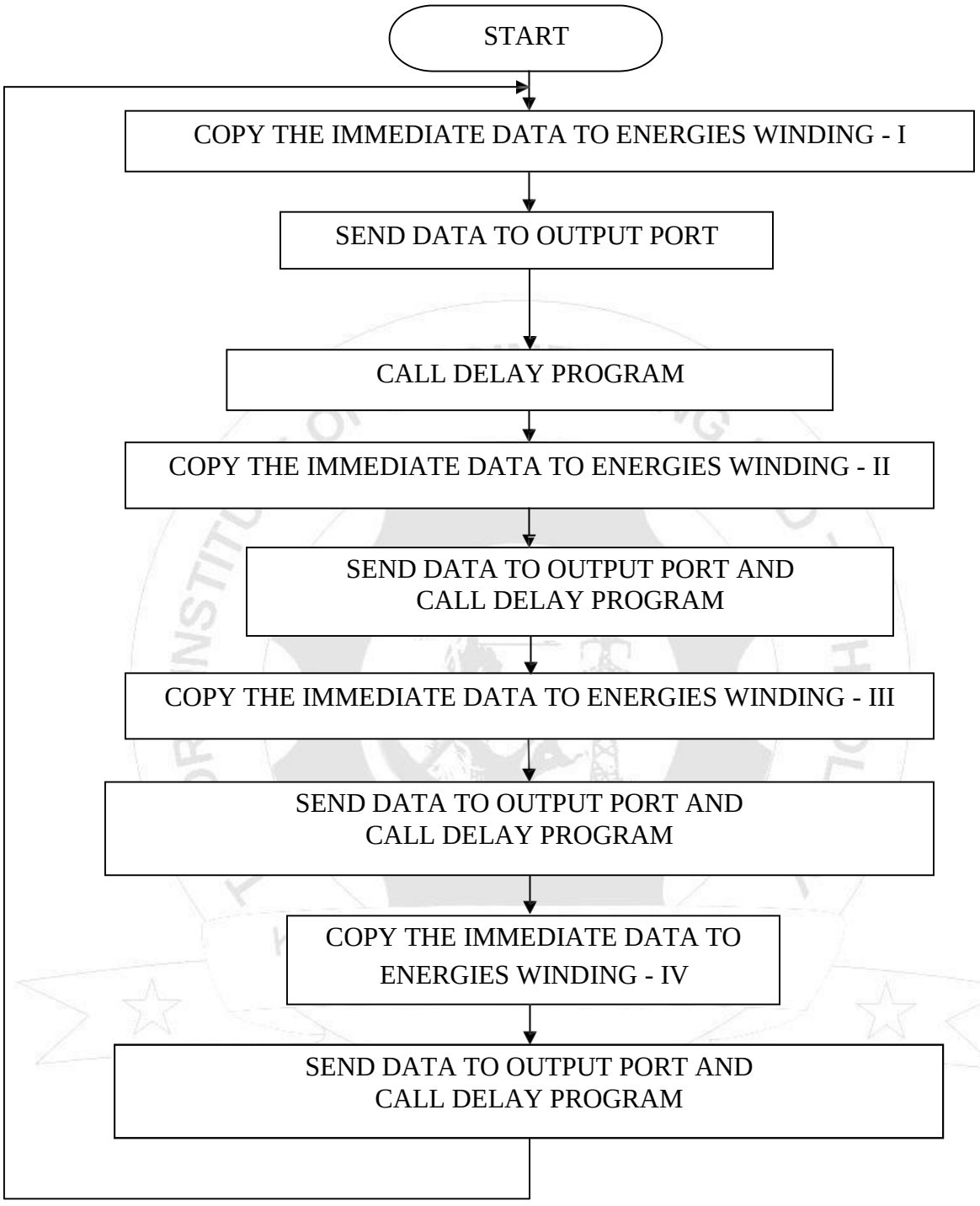
Algorithm:

1. Copy the immediate data 09H to Accumulator for energies of the winding-I.
2. Send the data to output port and call the delay program.
3. Copy the immediate data 05H to Accumulator for energies of the winding-II.
4. Send the data to output port and call the delay program.
5. Copy the immediate data 06H to Accumulator for energies of the winding-III.
6. Send the data to output port and call the delay program.
7. Copy the immediate data 0AH to Accumulator for energies of the winding-IV.
8. Jump to Step 1 and repeat the same.

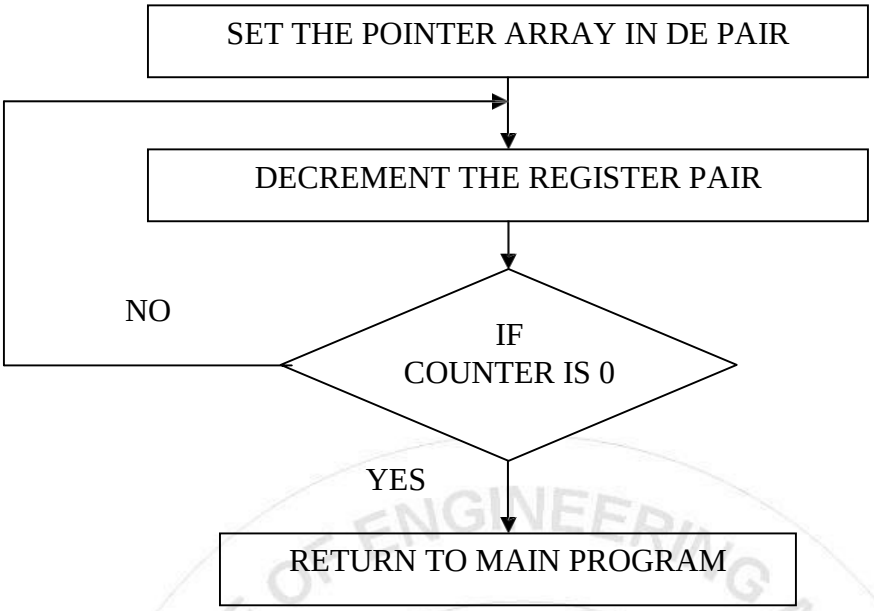
DELAY:

1. Set the pointer array to the register pair.
2. Decrement the register pair.
3. Check whether the counter is not equal to zero then go to step 2.
4. Return to main program.

Flowchart:


START
COPY THE IMMEDIATE DATA TO ENERGIES WINDING - I
SEND DATA TO OUTPUT PORT
CALL DELAY PROGRAM
COPY THE IMMEDIATE DATA TO ENERGIES WINDING - II
SEND DATA TO OUTPUT PORT AND CALL DELAY PROGRAM
COPY THE IMMEDIATE DATA TO ENERGIES WINDING - III
SEND DATA TO OUTPUT PORT AND CALL DELAY PROGRAM
COPY THE IMMEDIATE DATA TO ENERGIES WINDING - IV
SEND DATA TO OUTPUT PORT AND CALL DELAY PROGRAM

Delay Flowchart:



Program:

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
1000	L1:	MOV AL,09H	C6 C0 09	; Copy the immediate data 09H to AL-Reg energies winding I.
1003		OUT 0C0,AL	E6 C0	; Send the data to output port.
1005		CALL 1200H	E8 F8 01	; Call the delay program.
1008		MOV AL,05H	C6 C0 05	; Copy the immediate data 05H to AL-Reg energies winding II.
100B		OUT 0C0,AL	E6 C0	; Send the data to output port.
100D		CALL 1200H	E8 F0 01	; Call the delay program.
1010		MOV AL,06H	C6 C0 06	; Copy the immediate data 06H to AL-Reg energies winding III.
1013		OUT 0C0,AL	E6 C0	; Send the data to output port.
1015		CALL 1200H	E8 E8 01	; Call the delay program.
1018		MOV AL,0AH	C6 C0 0A	; Copy the immediate data 0AH to AL-Reg energies winding IV.
101B		OUT 0C0,AL	E6 C0	; Send the data to output port.
101D		CALL 1200H	E8 E0 01	; Call the delay program.
1020		JMP L1	E9 DD FF	; Jump to L1 for repeat the same.
Delay Program				
1200		MOV DX, 0FFFFH	C7 C2 FF FF	; Copy FFFF into DX register.
1204	L2:	DEC DX	4A	; Decrement DX register.
1205		JNZ L2	75 FD	; in not zero jump to L2.
1207		RET	C3	; Return to the main program.

Result:

Thus the assembly language program for stepper motor speed control was written and executed successfully by using 8086 Microprocessor trainer kit.

EXPT NO. : SERIAL COMMUNICATION

DATE :

AIM:

To implement serial communication between two kits.

APPARATUS REQUIRED

S.no	Required Components	specification	Qty
1.	MICROPROCESSOR	MICRO 86/88 KIT	1
2.	POWER CHORD	+5V SUPLLY	1
3.	KEY BOARD		1
4.	9-9 cable	RS232	1

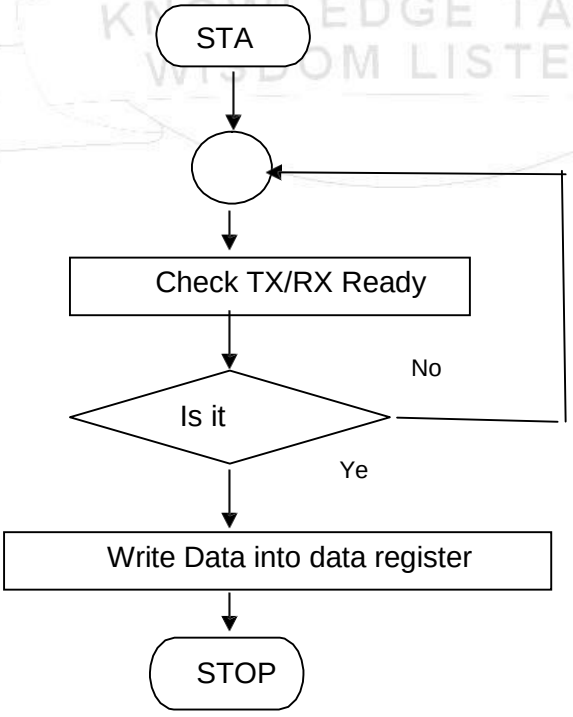
SOFWARE SPECIFICATION

- MASM SOFTWARE

ALGORITHM:

1. Initialize 8253 and 8251 to check the transmission and reception of a character
2. Initialize8253 to give an output of 150Khz at channel 0 which will give a 9600 baud rate of 8251.
3. The command word and mode word is written to the 8251 to set up for subsequent operations
4. The status word is read from the 8251 on completion of a serial I/O operation, or when the host CPU is checking the status of the device before starting the next I/O operation

FLOW CHART:



PROGRAM FOR TRANSMITTER

```
        CODE SEGMENT
        ASSUME CS:CODE,DS:CODE
        ORG 1000H
        MOV SI,1500H
        MOV AL,36H
        OUT 16H,AL
        MOV AL,40H
        OUT 10H,AL
        MOV AL,01H
        OUT 10H,AL
    RELOAD: MOV CL,05H
    CHECK: IN AL,0AH
        AND AL,04H
        JZ CHECK
        MOV AL,[SI]
        OUT 08H,AL
        INC SI
        CMP AL,3FH
        JNZ RELOAD
        DEC CL
        JNZ CHECK
        INT 02
        CODE ENDS
        END
```

PROGRAM FOR RECEIVER

```
    CODE SEGMENT
    ASSUME CS:CODE,DS:CODE
    ORG 1000H
    MOV SI,1500H
    MOV AL,36H
    OUT 16H,AL
    MOV AL,40H
    OUT 10H,AL
    MOV AL,01H
    OUT 10H,AL
RELOAD: MOV CL,05H
CHECK: IN AL,0AH
    AND AL,02H
    JZ CHECK
    IN AL,08H
    MOV [SI],AL
    INC SI
    CMP   AL,3FH
    JNZ  RELOAD
    DEC CL
    JNZ CHECK
    INT 02
    INT 02
    CODE ENDS
    END
```

PROCEDURE

1. Enter the transmitter program in one kit and receiver program in another kit
2. Connect RS232 cable b/w two kit
3. Now the data will be received serially to the receiver kit in the location 1500.

RESULT:

Thus the assembly language program for stepper motor speed control was written and executed successfully by using 8086 Microprocessor trainer kit.

EXPT NO. : PARALLEL COMMUNICATION

DATE :

AIM:

To implement parallel communication between two kits.

APPARATUS REQUIRED

S.no	Required Components	specification	Qty
1.	MICROPROCESSOR	MICRO 86/88 KIT	1
2.	POWER CHORD	+5V SUPLLY	1
3.	KEY BOARD		1
4.	26 core cable		1

SOFWARE SPECIFICATION

- MASM SOFTWARE

PROGRAM FOR TRANSMITER SIDE

```
        code segment
        assume cs:code,ds: code
        org 1000h
        mov al,82h
        OUT 26h,al
        mov al,3fh
        OUT 20h,aL
loop: in al,22h
        sub al,3fh
        jnz loop
        mov al,24h
        out 20h,al
        call delay
        int 02
delay:  mov  bl,05h
lion:   mov   dl,0ffh
l2: dec dl
        jnz l2
        dec bl
        jnz lion
        ret
        code ends
        end
```

RECEIVER SIDE

```
        code segment
        assume cs:code,ds:code
        org 1000h
        mov al,90h
        out 26h,al
check: in al,20h
       sub  al,3fh
       jnz check
       mov al,3fh
       out 22h,al
       mov cl,08h
       call delay
       mov si,1200h
l1:     in al,20h
       mov [si],al
       call delay
       inc si
       dec  cl
       jnz  l1
       int 02
delay:  mov  bl,05h
lion:   mov   dl,0ffh
l2: dec dl
       jnz l2
       dec bl
       jnz lion
       ret
       ret
       code ends
       end
```

PROCEDURE

1. Enter the transmitter program in one kit and receiver program in another kit
2. Connect 26 core cable b/w two kit
3. Now the data will be received parallel on the receiver kit in the location 1200.

RESULT

Thus the assembly language program for subtraction of two 8-bit data was written and executed successfully by using 8051 microcontroller trainer kit.

EXPT NO. : INTERFACING 8279 WITH 8086

DATE :

AIM:

To write a program to roll your name in the 7 segment display

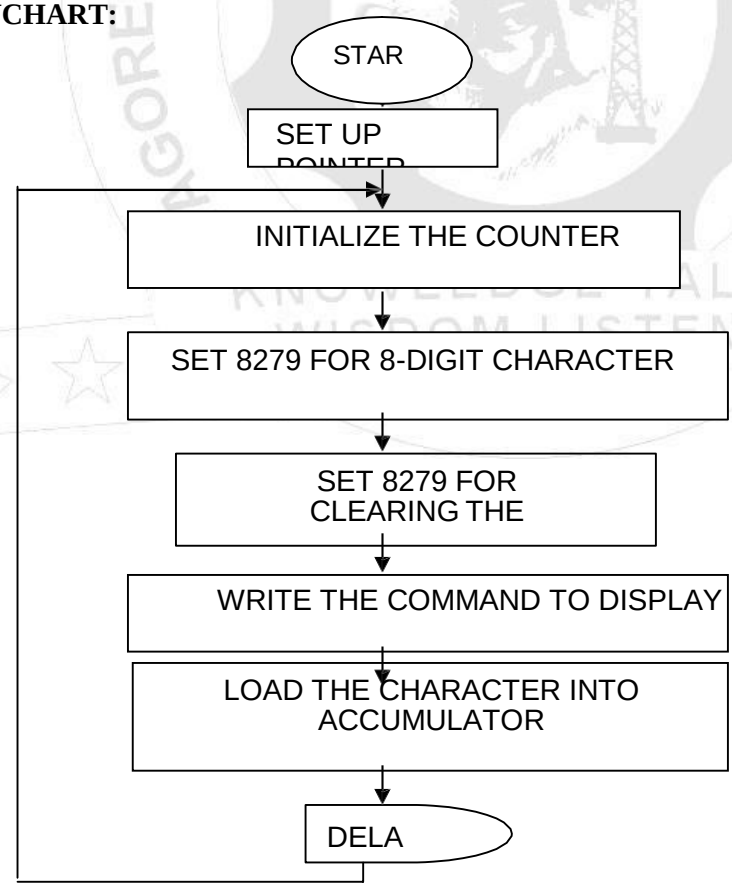
APPARATUS REQUIRED

S.no	Required Components	specification	Qty
1.	MICROPROCESSOR	MICRO 86/88 KIT	1
2.	POWER CHORD	+5V SUPLLY	1
3.	KEY BOARD		1
4	8279 INTERFACING	VBMB 001	1

ALGORITHM :

- Initialize the counter
- Set 8279 for 8 digit character display, right entry
- Set 8279 for clearing the display
- Write the command to display
- Load the character into accumulator and display it
- Introduce the delay
- Repeat from step 1.

FLOWCHART:



PROGRAM:

MEMORY ADDRESS	OPCODES	MNEMONICS	COMMENTS

		START:MOV SI,1200
		MOV CX,count
		MOV AL,10
		OUTC2,AL
		MOV AL,CC
		OUT C2,AL
		MOV AL,90
		OUT C2,AL
		NEXT:MOV AL.[SI]
		OUT C0,AL
		CALL DELAY
		INC SI
		LOOP NEXT
		JMP START
		
		DELAY:MOV
		DX,A0FF
		LOP:DEC DX
		JNZ LOP
		RET
			Initialize SI to 1200.
			Initialize count.
			Right entry.
			
			Clear display.
			
			RAM write display.
			
			Delay routine.

PROCEDURE:

i. Enter the program from location 1000.
ii. Connect the 8279 interface kit to 8086 Processor.
iii. Execute the Program
iv. The name will be displayed in the 7 segment

RESULT:

Thus the assembly language program for subtraction of two 8-bit data was written and executed successfully by using 8051 microcontroller trainer kit.

EXPT NO. : INTERFACING ADC WITH 8086

DATE :

AIM

To write a program to convert the analog into digital voltage

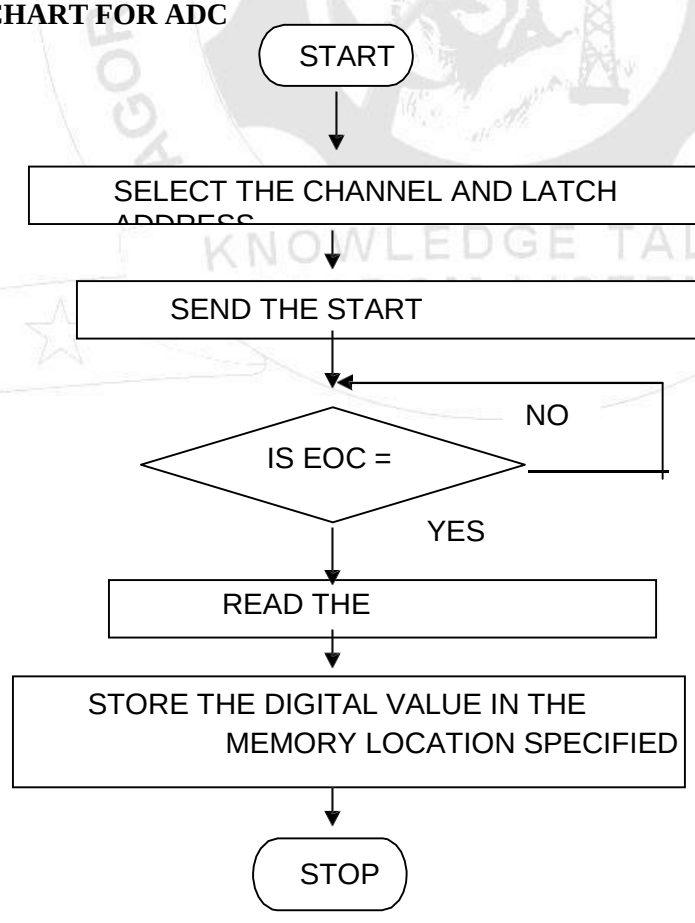
APPARATUS REQUIRED

S.no	Required Components	specification	Qty
1.	MICROPROCESSOR	MICRO 86/88 KIT	1
2.	POWER CHORD	+5V SUPLLY	1
3.	KEY BOARD		1
4	ANOLOG TO DIGITAL CONVERTER	VBMB 003	1

ALGORITHM:

(i) Select the channel and latch the address.
(ii) Send the start conversion pulse.
(iii) Read EOC signal.
(iv) If EOC = 1 continue else go to step (iii)
(v) Read the digital output.
(vi) Store it in a memory location

FLOW CHART FOR ADC



PROGRAM:

MEMORY ADDRESS	OPCODES	MNEMONICS	COMMENTS

		MOV AL,10
		OUT C8,AL
		MOV AL,18
		OUT C8,AL
		HLT
			Channel selection and ALE low
			
			ALE high
			
			Terminate the Program

PROCEDURE:

i. Enter the program from location 1000.
ii. Connect the ADC interface kit to 8086 Processor
iii. Vary the analog input using trim pot and give SOC by pressing the switch
iv. See the corresponding digital value in the led display

TAGORE INSTITUTE OF ENGINEERING AND TECHNOLOGY
KNOWLEDGE TALKS
WISDOM LISTENS

RESULT:

Thus the assembly language program for subtraction of two 8-bit data was written and executed successfully by using 8051 microcontroller trainer kit.

EXPT NO. : INTERFACING DAC WITH 8086

DATE :

AIM:

To write a program to

i. generate square waveform in dac2 output

ii. generate 0V in DAC 1 output.

APPARATUS REQUIRED

S.no	Required Components	specification	Qty
1.	MICROPROCESSOR	MICRO 86/88 KIT	1
2.	POWER CHORD	+5V SUPLLY	1
3.	KEY BOARD		1
4	DIGITAL TO ANALOG CONVERTER	VBMB 003	1

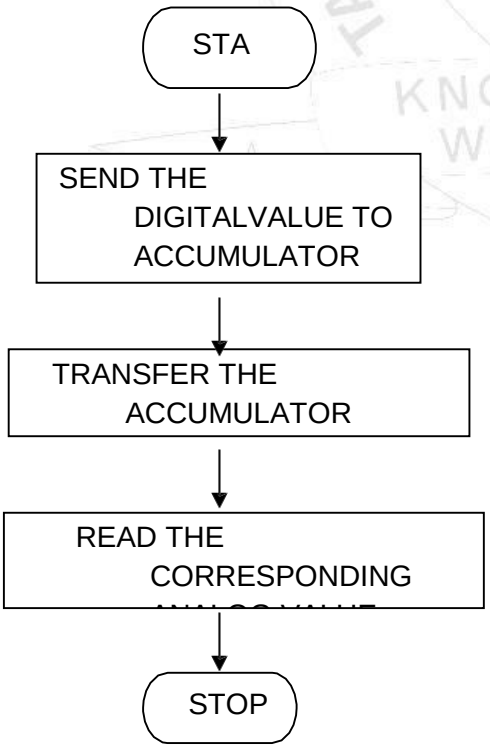
ALGORITHM

Square Waveform:

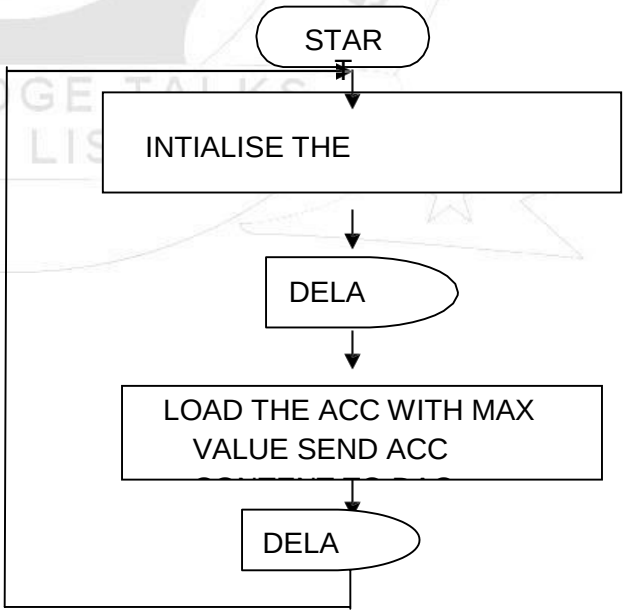
(i) Send low value (00) to the DAC.
(ii) Introduce suitable delay.
(iii) Send high value to DAC.
(iv) Introduce delay.
(v) Repeat the above procedure.

FLOWCHART:

MEASUREMENT OF ANALOG VOLTAGE



SQUARE WAVE FORM



PROGRAM TO GENERATE 0 V IN THE DAC1 OUTPUT

MEMORY ADDRESS	OPCODES	MNEMONICS	COMMENTS
		MOV AL,7FH OUT C0,AL HLT	7F(hex)=0v Out the data in DAC1 Terminate the Program

PROCEDURE

i. Enter the program from location 1000.
ii. Connect the DAC interface kit to 8086 Processor
iii. View the 0v output in the CRO

PROGRAM TO GENERATE SQUARE WAVE IN DAC2 OUTPUT

MEMORY ADDRESS	OPCODES	MNEMONICS	COMMENTS
		START:MOV AL,00 OUT C8,AL CALL DELAY MOV AL,FF OUT C8,AL CALL DELAY JMP START DELAY:MOV DX,A0FF LOP:DEC DX JNZ LOP RET	Low data in DAC 2 Delay High data in DAC 2 out put Process repeat Delay routine

PROCEDURE:

i. Enter the program from location 1000.
ii. Connect the DAC interface kit to 8086 Processor
iii. View the square wave output in the CRO

RESULT:

Thus the assembly language program for subtraction of two 8-bit data was written and executed successfully by using 8051 microcontroller trainer kit.

EXPT NO. : INTERFACING TRAFFIC LIGHT WITH 8086
DATE :

AIM:
To develop Traffic light Control system using 8086

APPARATUS REQUIRED:
Microprocessor trainer kit, Traffic light controller kit, power supply, data cable etc

PROGRAM:

```
START:
MOV AX,@DATA
MOV DS,AX
MOV AL,80H
MOV DX,CWR
OUT DX,AL
MOV AL,F3H
MOV DX,PORTC
OUT DX,AL
MOV AL,FFH
MOV DX,PORTA
OUT DX,AL
MOV AL,FFH
MOV DX,PORTB
OUT DX,AL
MOV CL,03H
CALL DELAY
TOP:
MOV AL,EEH
MOV DX,PORTA
OUT DX,AL
MOV AL,EEH
MOV DX,PORTB
OUT DX,AL
MOV CL,02H
CALL DELAY

MOV AL,FCH
MOV DX,PORTC
OUT DX,AL
MOV AL,7DH
MOV DX,PORTA
OUT DX,AL
MOV AL,57H
MOV DX,PORTB
OUT DX,AL
MOV CL,15H
CALL DELAY

MOV AL,E7H
MOV DX,PORTB
OUT DX,AL
MOV AL,FDH
MOV DX,PORTA
```

```
OUT DX,AL
MOV AL,EDH
MOV DX,PORTA
OUT DX,AL
MOV CL,02H
CALL DELAY

MOV AL,F7H
MOV DX,PORTB
OUT DX,AL
MOV AL,F0H
MOV DX,PORTC
OUT DX,AL
MOV AL,F1H
MOV DX,PORTA
OUT DX,AL
MOV CL,15H
CALL DELAY

MOV AL,FBH
MOV DX,PORTA
OUT DX,AL
MOV AL,FBH
MOV DX,PORTB
OUT DX,AL
MOV AL,50H
MOV DX,PORTC
OUT DX,AL
MOV CL,15H
CALL DELAY

MOV AL,FEH
MOV DX,PORTA
OUT DX,AL
MOV AL,FEH
MOV DX,PORTB
OUT DX,AL
MOV CL,03H
CALL DELAY

MOV AL,FFH
MOV DX,PORTA
OUT DX,AL
MOV AL,AFH
MOV DX,PORTC
OUT DX,AL
MOV AL,EEH
MOV DX,PORTA
OUT DX,AL
MOV AL,EEH
MOV DX,PORTB
OUT DX,AL
MOV CL,02H
CALL DELAY

MOV AL,BFH
```

```
MOV DX,PORTA
OUT DX,AL
MOV AL,BFH
MOV DX,PORTB
OUT DX,AL
MOV CL,15H
CALL DELAY
JMP TOP

DELAY:
MOV BX,10H
D1:
MOV CX,0FFFFH
D2:
LOOP D2
DEC BX
JNZ D1
INT 03H
END START
```

The sequence of traffic followed in the program is given below.

a) From road north to East
From road east to north
From road south to west
From road west to south
From road west to north

b)From road north to East
From road south to west
From road south to north
From road south to east

c) From road north to south
From road south to north
Pedestrian crossing at roads west & east

d) From road east to west
From road west to east
Pedestrian crossing at roads north & south

RESULT:

Thus the assembly language program for addition of two 8-bit data was written and executed successfully by using 8051 microcontroller trainer kit.

EXPT NO. : ADDITION OF TWO 8-BIT DATA
DATE :

Aim:

To write and execute an assembly program for addition of two 8-bit data by using 8051 microcontroller trainer kit.

Apparatus Required:

Sl.No.	Apparatus Required	Quantity
1	8051 Microcontroller trainer kit	1
2	Keyboard PS/2 port	1
3	Power chord	1

Algorithm:

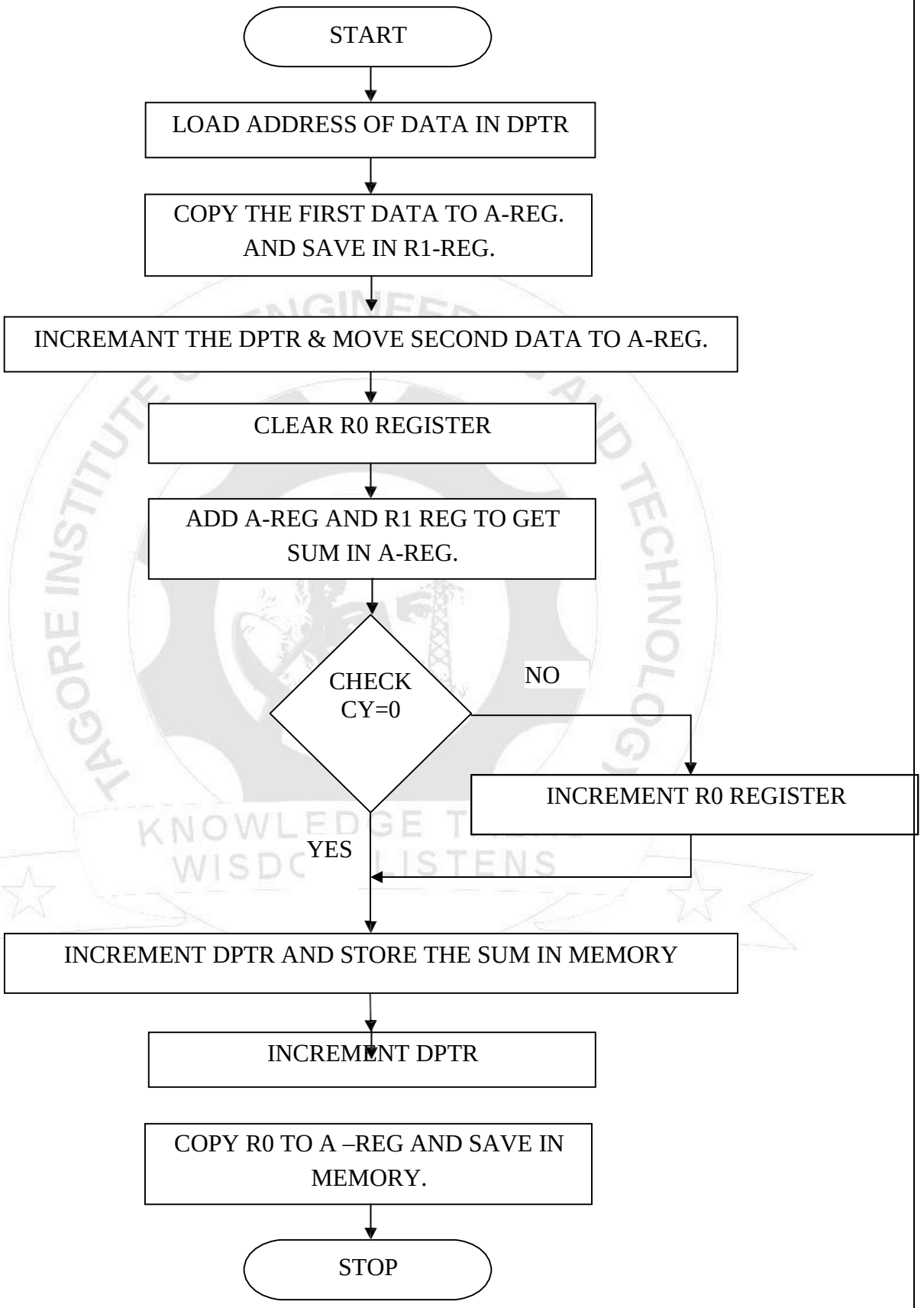
1. Set DPTR as pointer for data (local address of data in DPTR).
2. Copy the first data from external memory to accumulator and save it in R1-register.
3. Copy the Second data from external memory to accumulator.
4. Clear R0 register account for carry.
5. Add the content of R1 register to accumulator.
6. Checks for carry, if carry is not set go to step 8 otherwise do the following.
7. Increment the R0-register.
8. Increment the DPTR and Save the sum in external memory.
9. Terminate the program.

Program:

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
4100		MOV DPTR,#4200H	90 42 00	; Set the DPTR to external Ram address 4200 memory location.
4103		MOVX A,@DPTR	E0	; Copy the content of external address in DPTR to A-register.
4104		MOV R1,A	F9	; Copy the content of A-register to R1 register.
4105		INC DPTR	A3	; Increment the DPTR.
4106		MOVX A,@DPTR	E0	; Copy the content of external address in DPTR to A-register.
4107		MOV R0,#00H	78 00	; Copy the data 00H to R0 register.
4109		ADD A,R1	29	; Add the content of R1 to A register.
410A		JNC L1	50 01	; if CY is 0 then go to L1
410C		INC R0	08	; Increment the R0 register.
410D	L1:	INC DPTR	A3	; Increment the DPTR.
410E		MOVX @DPTR,A	F0	; Copy the content of A-register to external address in DPTR.
410F		INC DPTR	A3	; Increment the DPTR.
4110		MOV A,R0	E8	; Copy the content of R0 register to A-register.

4111		MOVX @DPTR,A	F0	; Copy the content of A-register to external address in DPTR.
4112	L2:	SJMP L2	80 FE	; Remain idle in infinite loop.

Flowchart:



SAMPLE INPUT AND OUTPUT:

INPUT		OUTPUT	
ADDRESS	**DATA**	**ADDRESS**	**DATA**
4200	E2H	4202	27H
4201	45H	4203	01H

Result:

Thus the assembly language program for addition of two 8-bit data was written and executed successfully by using 8051 microcontroller trainer kit.

EXPT NO. : SUBTRACTION OF TWO 8-BIT DATA

DATE :

Aim:

To write and execute an assembly program for subtraction of two 8-bit data by using 8051 microcontroller trainer kit.

Apparatus Required:

Sl.No.	Apparatus Required	Quantity
1	8051 Microcontroller trainer kit	1
2	Keyboard PS/2 port	1
3	Power chord	1

Algorithm:

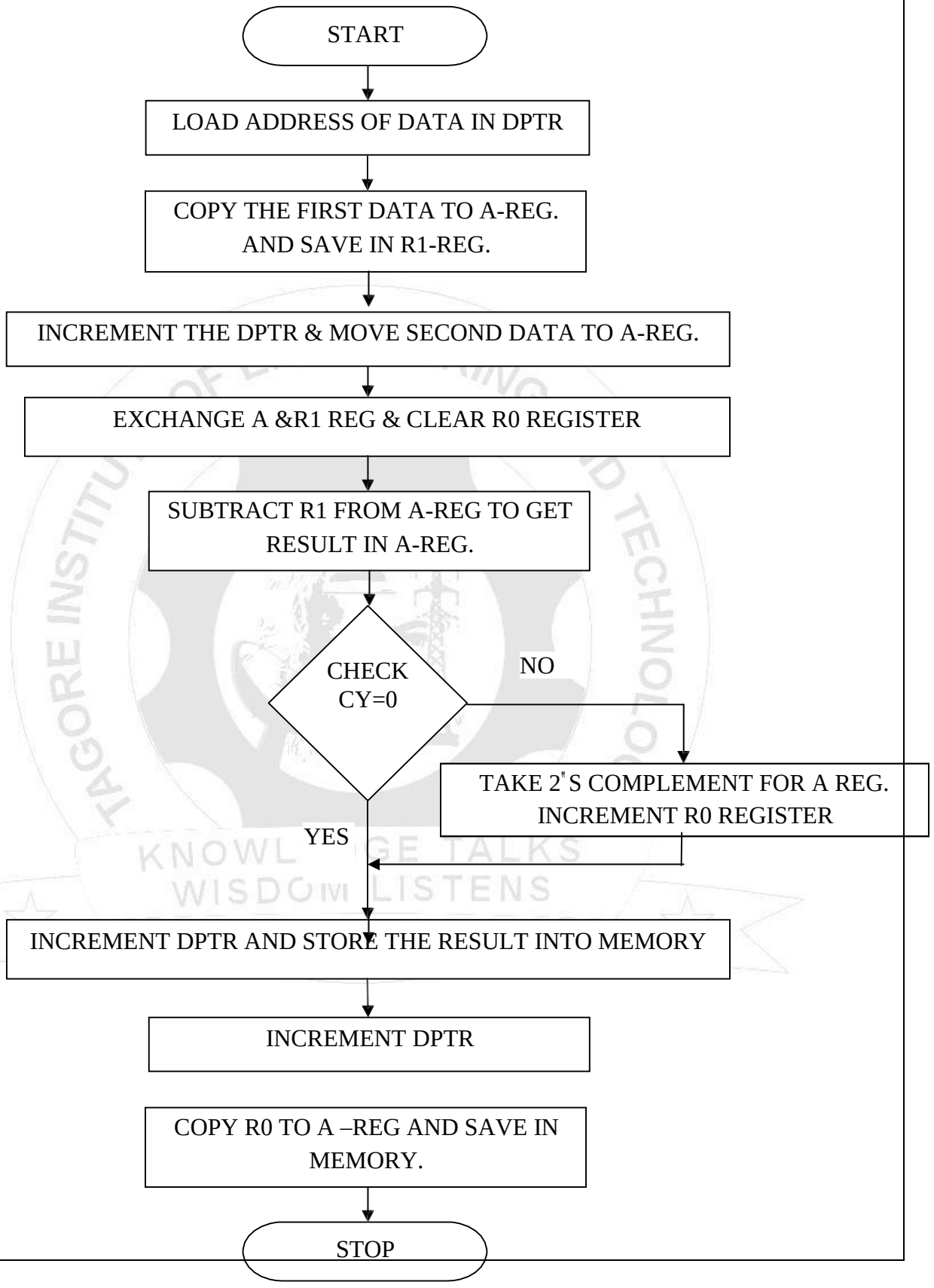
1. Set DPTR as pointer for data (local address of data in DPTR).
2. Copy the first data from external memory to accumulator and save it in R1-register.
3. Increment DPTR & Copy the Second data from external memory to accumulator.
4. Exchange the content of R1 and A register.
5. Clear R0 register to account for sign.
6. Clear the carry flag and Subtract the content of R1 from A-register.
7. Check CF=0 then go to step 9, otherwise do the following.
8. Complement the content of A-register and increment one to get 2's complement of result in A-register also increment R0 register by one to negative result.
9. Increment the DPTR and Save the sum in external memory.
10. Increment the DPTR , Copy R0-register to Accumulator and save sign bit in external memory
11. Terminate the program.

Program:

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
4100		MOV DPTR,#4200H	90 42 00	; Set the DPTR to external RAM address 4200 memory location.
4103		MOVX A,@DPTR	E0	; Copy the content of external address in DPTR to A-register.
4104		MOV R1,A	F9	; Copy the content of A-register to R1 register.
4105		INC DPTR	A3	; Increment the DPTR.
4106		MOVX A,@DPTR	E0	; Copy the content of external address in DPTR to A-register.
4107		XCHG A,R1	C9	; Exchange the content of R1 and A-Register.
4108		MOV R0,#00H	78 00	; Copy the data 00H to R0 register.
410A		CLR C	C3	; Clear the C-register.
410B		SUBB A,R1	99	; Subtract R1-reg and CY from A-register.
410C		JNC L1	50 03	; if CY is 0 then go to L1
410E		CPL A	F4	; Complement the A-register.
410F		INC A	04	; Increment the A-register
4110		INC R0	08	; Increment the R0 register.
4111	L1:	INC DPTR	A3	; Increment the DPTR.
4112		MOVX @DPTR,A	F0	; Copy the content of A-register to external address in DPTR.
4113		INC DPTR	A3	; Increment the DPTR.

4114		MOV A,R0	E8	; Copy the content of R0 register to A-register.
4115		MOVX @DPTR,A	F0	; Copy the content of A-register to external address in DPTR.
4116	L2:	SJMP L2	80 FE	; Remain idle in infinite loop.

Flowchart:



SAMPLE INPUT AND OUTPUT:

INPUT		OUTPUT	
ADDRESS	**DATA**	**ADDRESS**	**DATA**
4200	04H	4202	01H
4201	05H	4203	01H

Result:

Thus the assembly language program for subtraction of two 8-bit data was written and executed successfully by using 8051 microcontroller trainer kit.

EXPT NO. :. MULTIPLICATION OF TWO 8-BIT DATA

DATE :

Aim:

To write and execute an assembly program for multiplication of two 8-bit data by using 8051 microcontroller trainer kit.

Apparatus Required:

Sl.No.	Apparatus Required	Quantity
1	8051 Microcontroller trainer kit	1
2	Keyboard PS/2 port	1
3	Power chord	1

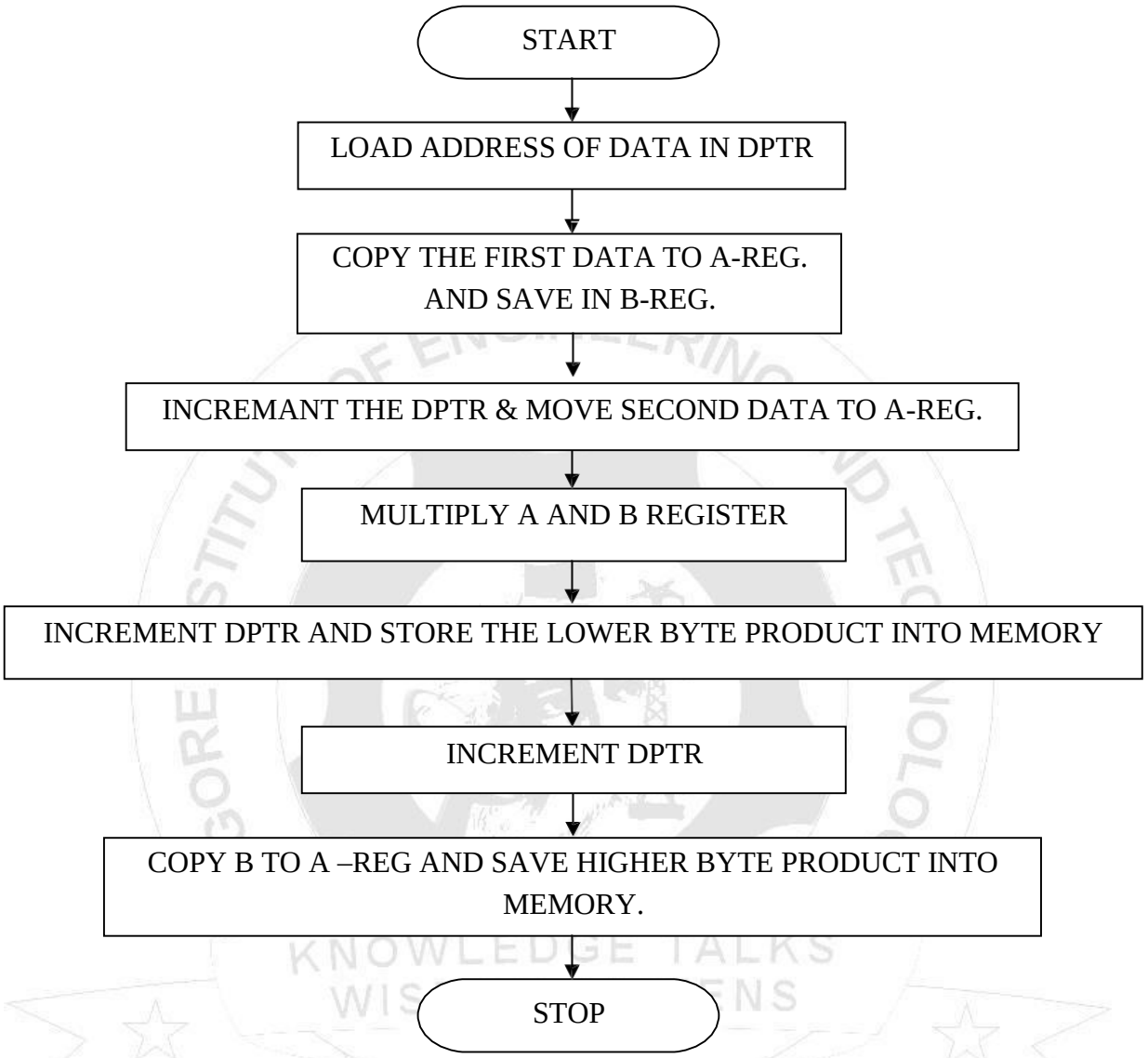
Algorithm:

1. Set DPTR as pointer for data (local address of data in DPTR).
2. Copy the first data from external memory to accumulator and save it in B-register.
3. Increment the DPTR and Copy the Second data from external memory to accumulator.
4. Perform multiplication to get the product in A-register and B-register.
5. Increment the DPTR and Save A-register into memory.
6. Increment the DPTR and move the B-register into A register and save it in memory.
7. Terminate the program.

Program:

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
4100		MOV DPTR,#4200H	90 42 00	; Set the DPTR to external Ram address 4200 memory location.
4103		MOVX A,@DPTR	E0	; Copy the content of external address in DPTR to A-register.
4104		MOV B,A	F5 F0	; Copy the content of A-register to B register.
4106		INC DPTR	A3	; Increment the DPTR.
4107		MOVX A,@DPTR	E0	; Copy the content of external address in DPTR to A-register.
4108		MUL AB	78 00	; Perform Multiplication.
4109		INC DPTR	A3	; Increment the DPTR.
410A		MOVX @DPTR,A	F0	; Copy the content of A-register to external address in DPTR.
410B		INC DPTR	A3	; Increment the DPTR.
410C		MOV A,B	E5 F0	; Copy the content of B register to A-register.
410E		MOVX @DPTR,A	F0	; Copy the content of A-register to external address in DPTR.
410F	L1:	SJMP L1	80 FE	; Remain idle in infinite loop.

Flowchart:



SAMPLE INPUT AND OUTPUT:

INPUT		OUTPUT	
ADDRESS	**DATA**	**ADDRESS**	**DATA**
4200	B2H	4202	16H
4201	03H	4203	02H

Result:

Thus the assembly language program for multiplication of two 8-bit data was written and executed successfully by using 8051 microcontroller trainer kit.

EXPT NO. : DIVISION OF TWO 8-BIT DATA

DATE :

Aim:

To write and execute an assembly program for division of two 8-bit data by using

8051 microcontroller trainer kit.

Apparatus Required:

Sl.No.	Apparatus Required	Quantity
1	8051 Microcontroller trainer kit	1
2	Keyboard PS/2 port	1
3	Power chord	1

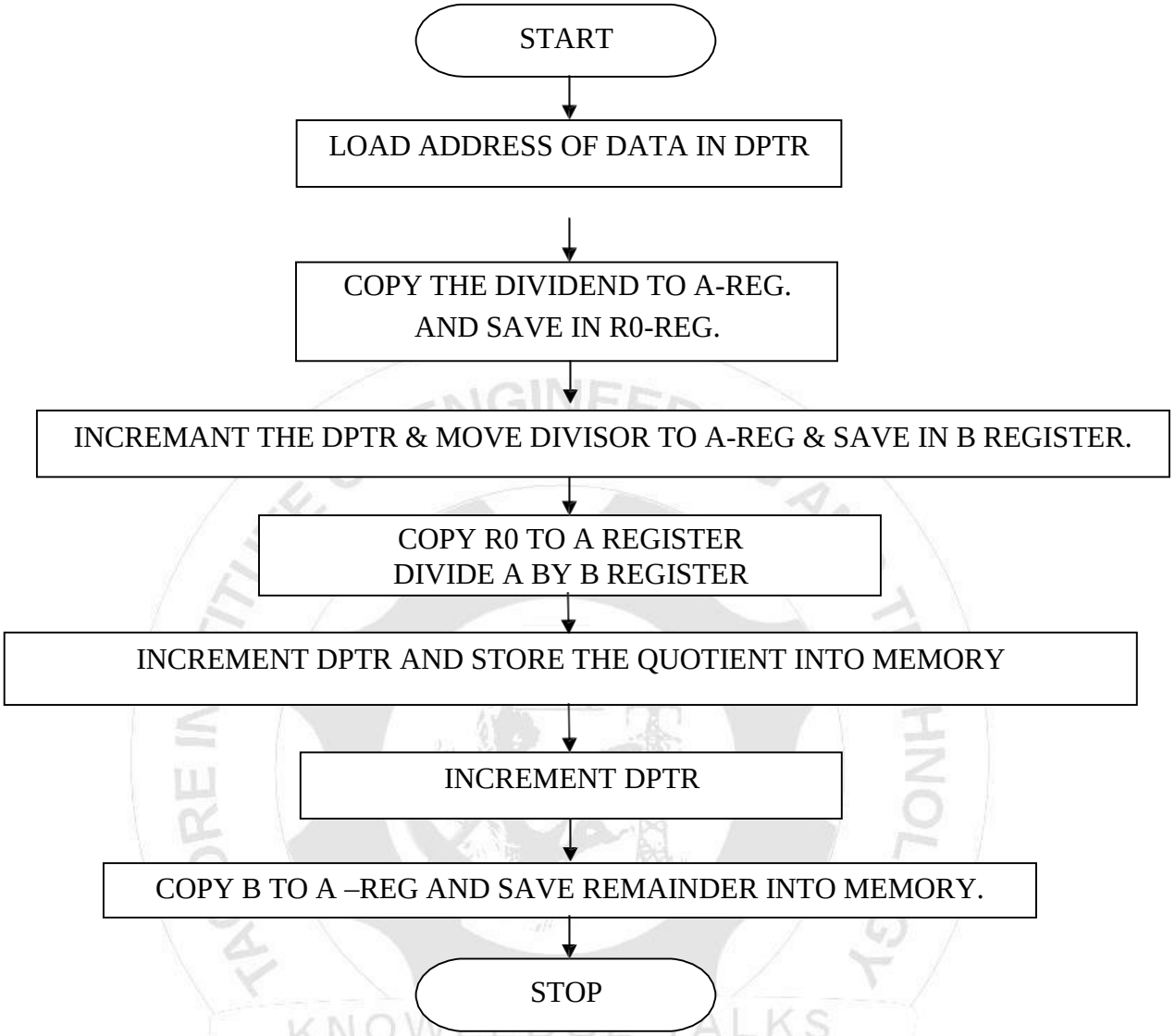
Algorithm:

1. Set DPTR as pointer for data (local address of data in DPTR).
2. Copy the dividend from external memory to accumulator and save it in R0-register.
3. Increment the DPTR and Copy the divisor from external memory to accumulator and store in B register.
4. Copy the dividend from R0 to A-register.
5. Perform division to get the quotient in A-register and remainder in B-register.
6. Increment the DPTR and Save quotient into memory.
7. Increment the DPTR and move the B-register into A register and save remainder in memory.
8. Terminate the program.

Program:

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
4100		MOV DPTR,#4200H	90 42 00	; Set the DPTR to external Ram address 4200 memory location.
4103		MOVX A,@DPTR	E0	; Copy the content of external address in DPTR to A-register.
4104		MOV R0,A	F8	; Copy the content of A-register to R0 register.
4105		INC DPTR	A3	; Increment the DPTR.
4106		MOVX A,@DPTR	E0	; Copy the content of external address in DPTR to A-register.
4107		MOV B,A	F5 F0	; Copy the content of A register to B-register.
4109		MOV A,R0	E8	Copy the content of R0-register to A register.
410A		DIV AB	84	; Perform Multiplication.
410B		INC DPTR	A3	; Increment the DPTR.
410C		MOVX @DPTR,A	F0	; Copy the content of A-register to external address in DPTR.
410D		INC DPTR	A3	; Increment the DPTR.
410E		MOV A,B	E5 F0	; Copy the content of B register to A-register.
4110		MOVX @DPTR,A	F0	; Copy the content of A-register to external address in DPTR.
4111	L1:	SJMP L1	80 FE	; Remain idle in infinite loop.

Flowchart:



SAMPLE INPUT AND OUTPUT:

INPUT		OUTPUT	
ADDRESS	**DATA**	**ADDRESS**	**DATA**
4200	02H	4202	02H
4201	01H	4203	00H

Result:

Thus the assembly language program for division of two 8-bit data was written and executed successfully by using 8051 microcontroller trainer kit.

EXPT NO. : 2'S COMPLEMENT OF A NUMBER

DATE :

Aim:

To write and execute an assembly program for 2" s complement by using 8051 microcontroller trainer kit.

Apparatus Required:

Sl.No.	Apparatus Required	Quantity
1	8051 Microcontroller trainer kit	1
2	Keyboard PS/2 port	1
3	Power chord	1

PROGRAM:

```
ORG 0000H
MOV DPTR, #0030H
MOVX A, @DPTR
CPL A
ADD A, #01H
MOV DPTR, #0090H
MOVX @DPTR, A
MOV PCON, #02H
```

INPUT: A=05H OUTPUT: A=OBH

RESULT:

Thus the assembly language program for 2" S complement was written and executed successfully by using 8051 microcontroller trainer kit.

EXPT NO. : UNPACKED BCD TO ASCII

DATE :

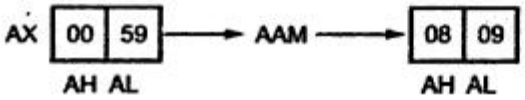
Aim:

To convert BCD number into ASCII by using 8051 micro controller.

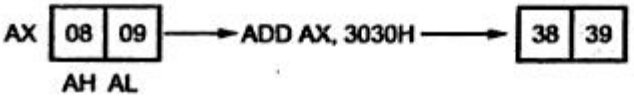
Apparatus Required:

Sl.No.	Apparatus Required	Quantity
1	8051 Microcontroller trainer kit	1
2	Keyboard PS/2 port	1
3	Power chord	1

Algorithm :



Note : 59H → 89 Decimal



Note : 38H and 39H are the ASCII equivalents of 8 and 9 respectively

1. Save contents of all registers which are used in the routine.
2. Get the data in AL register and make AH equal to 00.
3. Use AAM instruction to convert number in its decimal equivalent in the unpacked format.
4. Add 30H in each digit to get its ASCII equivalent.
5. Display digit one by one using function 2 of INT 21H.
6. Restore contents of registers.

PROGRAM:

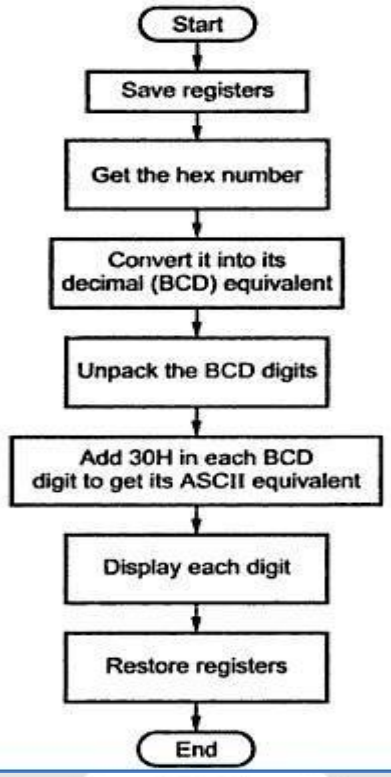
Routine : Convert Binary to ASCII for number less than 100

Passing Parameter : Hex number in AL register.

```
; Routine to convert binary number into its
; Decimal and then ASCII equivalent, and then display the number

BTA PROC NEAR

     PUSH DX            ; Save registers
     PUSH BX
     PUSH AX

     MOV AH, 00H        ; Clear AH
     AAM                ; Convert to BCD
     ADD AX, 3030H      ; Convert to ASCII
     MOV BX,AX          ; Save result
     MOV DL, BH         ; Load first digit (MSD)
     MOV AH, 02         ; Load function number
     INT 21H            ; Display first digit (MSD)
     MOV DL, BL             ; Load second digit (LSD)
     INT 21H                ; Display second digit (LSD)

     POP AX                 ; Restore registers
     POP BX
     POP DX
     RET
     ENDP
```

TAGORE INSTITUTE OF TECHNOLOGY
KNOWLEDGE TALKS
WISDOM LISTENS

FLOWCHART:



TAGORE INSTITUTE OF ENGINEERING AND TECHNOLOGY
KNOWLEDGE TALKS
WISDOM LISTENS

RESULT:

The given number is converted into ASCII using 8051 microcontroller kit.