



**Matrusri Engineering College
Saidabad, Hyderabad**

**Department
of
Electrical and Electronics Engineering**

Manual

Microprocessor and Microcontroller Laboratory
(Code PC752EE)

**B.E (EEE) VII semester
(2020-2021)**



Matrusri Engineering College Saidabad, Hyderabad

Vision, Mission, PEO'S, PO's ,PSO's and correlation:

Vision

To prepare individuals to thrive in and contribute to an ever changing global society by uniting the skills of engineering, technology, research and design

Mission

- M1. To provide state of the art resources that contributes to achieve excellence in teaching-learning, research and development activities
- M2. To produce qualified, motivated and well-rounded graduates possessing fundamental knowledge of Electrical & Electronics Engineering who can provide solutions and services to the community at large
- M3. To create and sustain environment of learning in which students acquire knowledge and learn to apply it professionally with due consideration of ethical and economical issues
- M4. To promote research & innovation and to keep abreast of emerging technologies in Electrical & Electronics Engineering in order to serve the needs of industry, scientific community and society

Program Education Objectives (PEOs)

- PEO1: To prepare students with good foundation in mathematics, science and engineering fundamentals required to comprehend, analyze, formulate solutions for real life engineering problems and facilitate them to pursue higher studies and /or to find entry level position in industries.
- PEO2: To inculcate effective communication skills, leadership, team work, multidisciplinary approach, and an ability to provide engineering solutions in a broader societal context.
- PEO3: To provide academic environment with an awareness of excellence, ethical codes and guidelines, and the life-long learning needed for a successful professional career.

Program Outcomes (POs)

- PO-1(Engineering Knowledge): Apply knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.
- PO-2(Problem Analysis): Identify, formulate, research literature and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.



Matrusri Engineering College

Saidabad, Hyderabad

PO-3(Design/ Development of Solutions):Design solutions for complex engineering problems and design system components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal and environmental considerations.

PO-4(Conduct investigations of complex problems): Using research-based knowledge and research methods including design of experiments, analysis and interpretation of data and synthesis of information to provide valid conclusions.

PO-5(Modern Tool Usage):Create, select and apply appropriate techniques, resources and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO-6(The Engineer and Society):Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice.

PO-7(Environment and Sustainability):Understand the impact of professional engineering solutions in societal and environmental contexts and demonstrate knowledge of and need for sustainable development.

PO-8(Ethics):Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.

PO-9(Individual and Team Work):Function effectively as an individual, and as a member or leader in diverse teams and in multi disciplinary settings.

PO-10(Communication):Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations and give and receive clear instructions.

PO-11(Project Management and Finance):Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO-12(Life-long Learning):Recognize the need for and have the preparation and ability to engage in independent and life- long learning in the broadest context of technological change.

Program Specific Outcomes

PSO-1: Apply appropriate techniques ,hardware and software tools to design ,analyze and test various systems in power electronics and power systems engineering adaptable to multi disciplinary environments

PSO-2: Identify the optimal solutions for industrial and societal electrical energy requirements by applying suitable design and control strategies



Matrusri Engineering College Saidabad, Hyderabad

Course Objectives:

1. To introduce the basics of microprocessors programming and their applications
2. To introduce students the basics of microcontrollers programming and their applications
3. Developing of assembly level programs and providing the basics of the processors.
4. To provide the solid foundation of interfacing the external devices to the processor according to the user requirements to create novel products and solutions for the real time problems

Course outcomes:

At the end of the course student will have ability to

1. Write the assembly level programming in micro processor.
2. Write the assembly level programming in micro controller.
3. Demonstrate the circuits for various applications using micro controllers
4. Apply the concepts of microprocessor in real time applications.



Matrusri Engineering College Saidabad, Hyderabad

MICROPROCESSOR & MICROCONTROLLERS LAB

PC752EE

List of Experiments:

For 8086:

Section 1: Using MASM/TASM

1. Programs for signed/unsigned multiplication and division.
2. Programs for finding average of N 16-bit numbers.
3. Programs for finding the largest number in an array.
4. Program for code conversion like BCD numbers to 7-Segment.
5. Programs for compute factorial of a positive integer number.

Section 2: Using 8086 (Interfacing)

1. 8279 – Keyboard Display: Write a small program to display a string of characters.
2. 8255 – PPI: Write ALP to generate triangular wave using DAC.
3. 8253 – Timer/Counter: Application of different modes
4. 8251 – USART: Write a program in ALP to establish Communication between two processors.
5. Traffic Signal Controller.

For 8051:

Section 3: Using 8051 Kit (Simple Programs)

1. Data Transfer – Block move, Exchange, sorting, finding largest element in an array.
2. Arithmetic Instructions: Multibyte operations.
3. Boolean & Logical Instructions (Bit manipulations).
4. Programs to generate delay, programs using serial port and on chit timer/counter.
5. Use of JUMP and CALL Instructions.

Section 4: Program Development using 'C' cross compiler for 8051

1. Square Wave Generation using timers.
2. Interfacing of Keyboard and 7-segment Display Module.
3. ADC interfacing for temperature monitoring.
4. DAC interfacing for Generation of Sinusoidal wave.
5. Stepper motor control (Clockwise, anticlockwise and in precise angles)

Additional Experiments Using Masm

1. Copying A String
2. Comparing a string



Matrusri Engineering College Saidabad, Hyderabad

INTRODUCTION TO MASM

The Microsoft macro assembler is an x86 high level assembler for DOS and Microsoft windows. It supports wide varieties of macro facilities and structured programming idioms including high level functions for looping and procedures

A program called **assembler** used to convert the mnemonics of instructions along with the data into the equivalent object code modules, these object code may further converted into executable code using linked and loader programs. This type of program is called as ASSEMBLY LANGUAGE PROGRAMMING. The assembler converts and Assembly language source file to machine code the binary equivalent of the assembly language program. In this respect, the assembler reads an ASCII source file from the disk and program as output. The major different between compilers for a high level language like PASCAL and an Assembler is that the compiler usually emits several machine instructions for each PASCAL statement. The assembler generally emits a single machine instruction for each assembler language statement.

Attempting to write a program in machine language is not particularly bright. This process is very tedious, mistakes, and offers almost no advantages over programming in assembly language. The major disadvantages over programming in assembly language over pure machine code are that you must first assemble and link a program before you can execute it. However attempting to assemble the code by hand would take for longer than the small amount of time that the assembler takes the perform conversion for you. An assembler like Microsoft Macro Assembler (MASM) provides a large number of features for assembly language programmers. Although learning about these features take a fair amount of time. They are so useful that it is well worth the effort.

Microsoft MASM version 6.11 contains updated software capable of processing printing instructions. Machine codes and instruction cycle counts are generated by MASM for all instructions on each processor beginning with 8086. To assemble the file PROG.ASM use this command: (better to use DOS command line)



Matrusri Engineering College Saidabad, Hyderabad

MASM PROG.ASM

The MASM program will assemble the PROG.ASM file. (To create PROG.OBJ from PROG.ASM)

To create PROG.EXE from PROG.OBJ, use this LINK command:

LINK PROG.OBJ

It converts the contents of PROG.OBJ into PROG.EXE.

To link more than one object file use + signs between their file names as in:

LINK PROGA+PROGB+PROGC

The following is a list of MASM reserved words:

ASSUME	assume definition
CODE	begin code segment
DATA	begin data segment DB define byte
DD	define double word
DQ	define quad word
DS	define storage
DUP	duplicate
DW	define word
ELSE	else statement
END	end program
ENDM	end macro
ENDIF	end if statement
ENDP	end procedure
ENDS	end segment
EQU	equate
IF	if statement
FAR	far reference
MACRO	define macro
.MODEL	model type
NEAR	near reference
OFFSET	offset
ORQ	origin
PARA	paragraph
PROC	define procedure
.EXIT	generate exit code
PUBLIC	public reference
SEG	locate segment



Matrusri Engineering College Saidabad, Hyderabad

SEGMENT	define segment
PTR	pointer

USING DEBUG TO EXECUTE THE 80x86 PROGRAM:

DEBUG is a utility program that allows a user to load an 80x 86 programs into memory and execute it step by step. DEBUG displays the contents of all processor registers after each instruction execute, allowing the user to determine if the code is performing the desired task. DEBUG only displays the 16-bit portion of the general purpose registers. Code view is capable of displaying the entire 32 bits. DEBUG is a very useful debugging tool. We will use DEBUG to step through a number of simple programs, gaining familiarity with Debug's commands as we do so. DEBUG contains commands that can display and modify memory, assemble instructions,

disassemble code already placed into memory, trace single or multiple instructions, load registers with data and do much more.

DEBUG loads into memory like any other program, in the first available slot.

The memory space used by DEBUG for the user program begins after the end of Debug's code. If an .EXE or .COM file were specified, DEBUG would load the

program according to accepted DOS conventions.

To execute the program file PROG.EXE use this command

DEBUG PROG.EXE

DEBUG uses a minus sign as its command prompt, so should see a "--" appear on display.

To get a list of some commands available with DEBUG is :

- T - trace (step by step execution)
- U - un assemble
- D - dump
- G - go (complete execution)



Matrusri Engineering College Saidabad, Hyderabad

H - Hex

To execute the program file PROG.ASM use the following procedure:

.MASM PROG.ASM

.LINK PROG.OBJ

.DEBUG PROG.EXE

ASSEMBLER DIRECTIVES:

The limits are given to the assembler using some pre defined alphabetical strings called Assembler Directives which help assembler to correctly understand. The assembly language programs to prepare the codes.

DB	GROUP	EXTRN
DW	LABEL	TYPE
DQ	LENGTH	EVEN
DT	LOCAL	SEGMENT
ASSUME	NAME	
END	OFFSET	
ENDP	ORG	
ENDS	PROC	
EQU	PTR	

DB-Define Byte: The DB drive is used to reserve byte of memory locations in the available on memory.

DW-Define Word: The DW drive is used to reserve 16 byte of memory location available on memory.

DQ-Define Quad Word (4 words): The DB directives is used to reserve 8 bytes of memory locations in the memory available.



Matrusri Engineering College

Saidabad, Hyderabad

DT-Define Ten Byte: The DT directive is used to reserve 10 byte of memory locations in the available memory.

ASSUME: Assume local segment name the Assume directive is used to inform the assembler. The name of the logical segments to be assumed for different segment used in programs.

END: End of the program the END directive marks the end of an ALP.

ENDP: End of the procedure.

ENDS: End of the segment.

EQU: The directive is used to assign a label with a variable or symbol. The directive is just to reduce recurrence of the numerical values or constants in the program.

OFFSET: Specifies offset address.

SEGMENT: The segment directive marks the starting of the logical segment.

EXECUTION OF ASSEMBLY LANGUAGE PROGRAMMING IN MASM SOFTWARE:

Assembly language programming has 4 steps.

1. Entering Program
2. Compile Program
3. Linking a Program
4. Debugging a Program

PROCEDURE:

1. Entering Program:-

Start Menu

Run

Cmd





Matrusri Engineering College Saidabad, Hyderabad

C:\cd MASM

C:\MASM> edit filename.asm



C:\MASM\filename.asm

This is editor

Enter program here

After entering program save & exit (ALT F & Press S or ALT F & Press X)

C:\MASM>

2. Compile the

Program:-C:\MASM> MASM

filename.asm

Microsoft @macro assembler version 5.10

Copy rights reserved© Microsoft

Corp 1981 All rights reserved

Object filename [OBJ];

List filename [NUL, LIST];

Cross Reference [NUL, CRF];

Press enter the screen shows c>

3. Linking a Program:-

c> link filename.obj



Microsoft @ overlay linker version 3.64

Copy rights reserved© Microsoft corp.

1983-88. All rights reserved



Matrusri Engineering College Saidabad, Hyderabad

Object module [.OBJ];

Run file [.EXE];

List [NUL MAP];

Libraries [LIB];

Press enter till screen chows c>

4. Debug a
Program:-C> debug
filename.exe

- (Screen shows dash)

- t ←

„t“ for trace the program execution by single stepping starting from the address
SEG.OFFSET. „q“ for Quit from Debug & return to DOS.

EXECUTION OF ASSEMBLY LANGUAGE PROGRAMMING IN MASM SOFTWARE:

1. To Study the various directives of Micro Assembler.
2. Switch on the computer, and go to DOS window displaying the C prompt i.e., C:\>
3. C:\>CD MASM611
4. C:\MASM611>CD BIN
5. C:\>MASM611\CD BIN>edit file_name.asm
6. Then the program is to be loaded in to the file. After entering, the program is to be saved using same method.
7. Alt+F is to be pressed. The monitor lists the commands. Enter Exit command and press Enter key.
8. Monitor displays C prompt , C:\>MASM611>CD BIN.
9. Enter
C:\MASM611\BIN>masm file_name.asm and then press enter key.
10. Enter the following command
C:\MASM611\BIN>link file_name.obj and then press enter key.
11. Then it displays



Matrusri Engineering College Saidabad, Hyderabad

RUN PROGRAM:FILE_NAME.EXE

Then press enter 3 times

12. C:\MASM611\BIN >debug file_name.exe and then press enter key.
13. Monitor then displays a hyphen –
14. debug the program line by line t (Trap) and press enter key
<t> <enter>.....Program executes line by line.
15. 'r' is to be pressed to get the register contents.
16. 'g' is to be pressed for entire program execution.
17. 'd' is to be pressed to get the data in memory locations.
18. 'q' is to be pressed to return to DOS prompt

PROCEDURE FOR PROGRAMS ON KEIL SOFTWARE

1. Click on Keil uvision3.
2. Click on **“Project”**, create a new project and save it in a new folder choose target option for **Atmel** and **AT89C51**.
3. Go to File, click on new file, and type the program.
4. Go to File, click on **“Save as”**, save the program with extension **.asm** on your particular folder where you saved your project.
5. Add your program to **Source Group 1** which is at Target1 (Project workspace) which is created after selecting the target in step 2.
6. To do this right clicks on Source Group 1 and select „Add files to Source Group 1“.
7. Search your code with .asm extension.
8. Now Click on **Translate current file** tab present file toolbar and check for errors. If error present then rectify.
9. Click on **Rebuild all target files** to add our program to the AT89C51 target.
10. Go to **Debug**, click on **Start/Stop debug session**.
11. For giving input data: Go to **view**, click on **Memory window**.
12. Enter **inputs** for corresponding memory addresses.
13. For internal memory type: i:0x20 for example



Matrusri Engineering College Saidabad, Hyderabad

14. For external memory type: x:0x2000 for example
15. Now click on “**Run**”, check the results
16. While in Debug don’t make any changes in the program.
17. After running again click **Start/Stop debug session** to edit mode for changes in program.

PROCEDURE FOR PROGRAMS ON 8051 KIT

1. Switch ON the power supply.
2. In the display it shows that ESA 51E.
3. In the next line it shows the symbol >.
4. Here you have to type A starting address of the program. i.e., >A8000
5. Then you start to type the each and every instruction of the program with the help of enter key.
6. Type all the instruction. You have press the Esc key.
7. Once again the same symbol appear in the display i.e., >.
8. Now you have to type the Data Memory address to load data i.e., MD 8900.
9. After loading the first data press enter key, and then it automatic shows the next address.
10. After enter the all data and instruction. Once again press the Esc key.
11. Now press the Key G and starting address of the instructions. i.e., G8000
12. You got the output directly in the display for this program.



Matrusri Engineering College Saidabad, Hyderabad

Section: 1

Ex.1 Signed / Unsigned Multiplication and Division using Masm

Date:

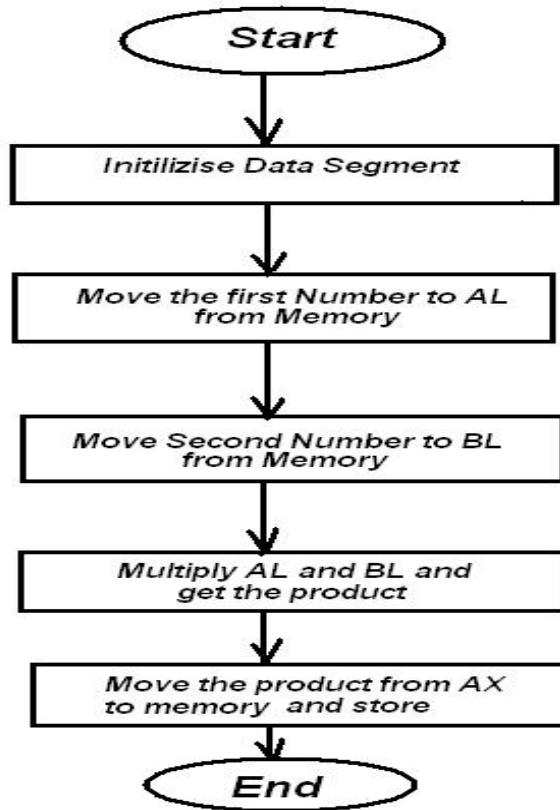
Aim: To study the operation of unsigned/signed Multiplication and division by using IDIV and IMUL instruction and to perform all arithmetic operations on signed numbers.

Apparatus required: PC, MASM Software Tool

Flow chart:



Matrusri Engineering College Saidabad, Hyderabad



Program:

(a) Unsigned 8-bit multiplication:

ASSUME CS:CODE,DS:DATA

DATA SEGMENT

NUM1 DB 15H ;First number

NUM2 DB 25H ;Second number

RESULT DW ? ;result in memory

DATA ENDS

CODE SEGMENT

START: MOV AX,DATA ;



Matrusri Engineering College Saidabad, Hyderabad

MOV DS,AX ; Data Segment initialization
MOV AL,NUM1 ; move first number to al register from memory
MOV BL,NUM2 ; move second number to bl
MUL BL ;multiply content of al and bl registers
MOV SI,OFFSET RESULT ;move address of the result into dx register
MOV [SI],AX ;move the result to memory
INT 3H ;terminate the program
CODE ENDS
END START ; end of program

Note: For multiplication and division one operand is accumulator and second operand

May be any register or memory location

Example: mul bl means al X bl , mul cx means ax X cx

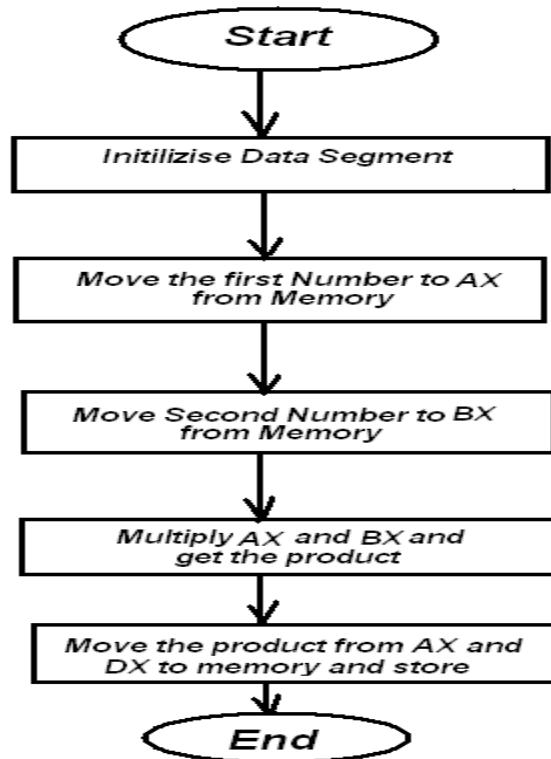
Result: AL = 15h, BL = 25h

AX = AL X BL = 0309h

(b) Unsigned 16-bit multiplication:



Matrusri Engineering College Saidabad, Hyderabad



(b) Unsigned 16-bit multiplication:

Program:

ASSUME CS:CODE,DS:DATA

DATA SEGMENT

NUM1 DW 1234H ; First Number

NUM2 DW 2222H ; Second Number

RESULT DD ? ; Memory For Result

DATA ENDS

CODE SEGMENT

START: MOV AX,DATA

MOV DS,AX

MOV AX,NUM1 ; move first no. To ax



Matrusri Engineering College Saidabad, Hyderabad

MOV BX,NUM2 ; move 2nd no. To bx
MUL BX ; Multiply The 2 Nos.
MOV SI,OFFSET RESULT ; Set Si For Address Of The Result
MOV [SI],AX ; Move The Lsw Of The Result To Memory
MOV [SI+2],DX ; Move The Msw Of The Result
INT 3H
CODE ENDS
END START

Result: Data : AX = 1234h, BX=2222h

Product : 026D52E8

AX = 026D (Most Significant Word of result)

DX= 52E8 (Least Significant word of result)

(c) Unsigned 8-bit Division:

Program:

ASSUME CS:CODE,DS:DATA

DATA SEGMENT

NUM1 DB 75H ; First Number
NUM2 DB 25H ; Second Number
RESULT DW ? ; Memory For Result

DATA ENDS

CODE SEGMENT

START: MOV AX,DATA

MOV DS,AX



Matrusri Engineering College Saidabad, Hyderabad

XOR AX,AX

MOV AL,NUM1 ; move the dividend to AL

MOV BL,NUM2 ; move the divisor to BL

DIV BL ; perform division

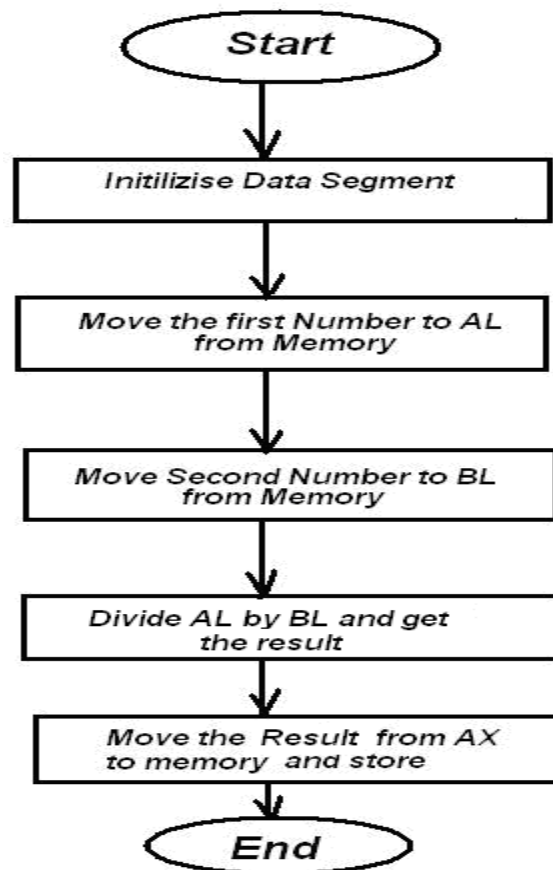
MOV DX,OFFSET RESULT ; get the address for result in DX

MOV [DX],AX ; move the result to memory

INT 3H

CODE ENDS

END START



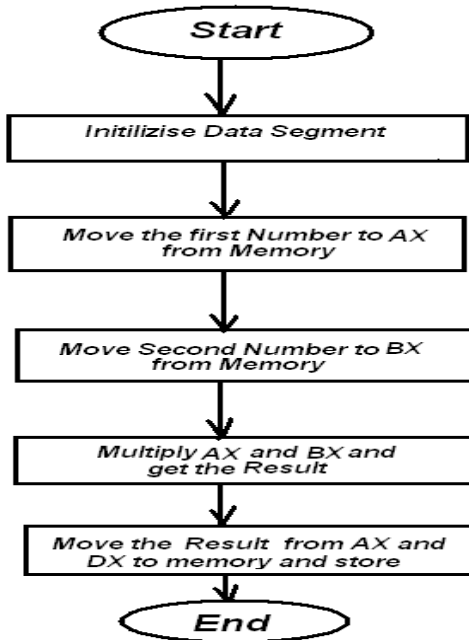
Data: AL = 75h, BL = 25h

Result: : AL = 03h (Quotient) , AH = 06h (Reminder)



Matrusri Engineering College Saidabad, Hyderabad

(d) Unsigned 16 bit division:



Program:

ASSUME CS:CODE,DS:DATA

DATA SEGMENT

NUM1 DW 2468H ; first number

NUM2 DW 0002H ; second number

RESULT DD ? ; memory for result

DATA ENDS

CODE SEGMENT

START: MOV AX,DATA

MOV DS,AX

MOV AX,NUM1 ; move the dividend to AX

MOV BX,NUM2 ; move divisor to BX

DIV BX ; perform 16 bit division



Matrusri Engineering College Saidabad, Hyderabad

MOV SI,OFFSET RESULT ; set si for address of the result

MOV [SI],DX ; move the result to memory

MOV [SI+2],AX

INT 3H ; terminate the program

CODE ENDS

END START

Data: AX = 2468h, BX = 0002h

Result: AX = 1234h (Quotient), DX = 0000h (remainder)

Note: For sign multiplication and division **IMUL** and **IDIV** instructions are used in place of **MUL** and **DIV** instructions.

Questions:

1. In 8-bit multiplication where the result is stored? In 16-bit multiplication where is result is stored?
2. How do you defined 8-bit and 16-bit numbers? Give examples.
3. When overflow flag, divide by zero flag will set?
4. Explain DAA and DAS instruction with examples?
5. Draw the flag register and mention various flag bits?



Matrusri Engineering College Saidabad, Hyderabad

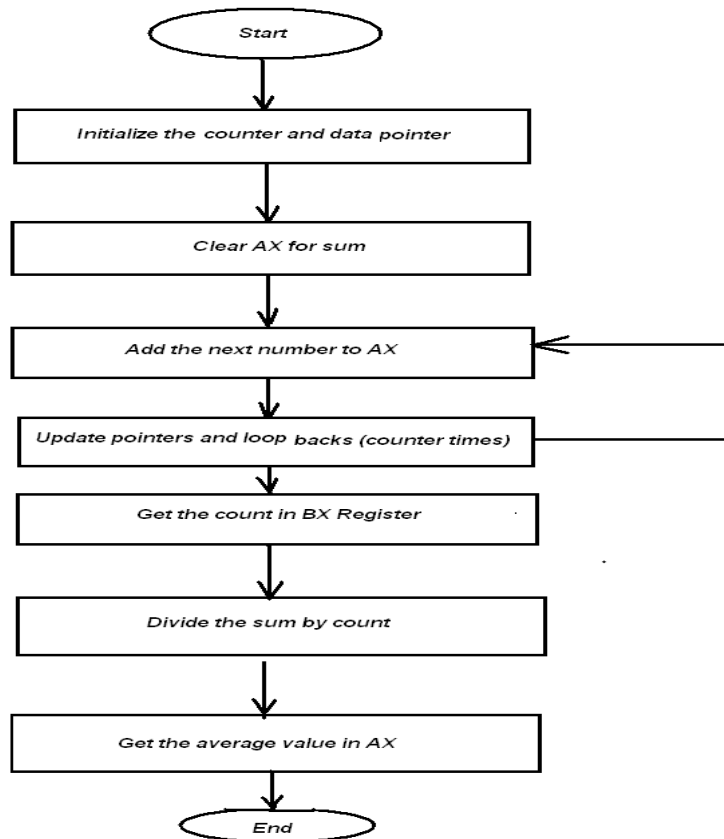
Ex.2 Find the Average of N-16 bit Numbers using Masm

Date:

Aim: To find the average of N-16 bit numbers using ALP program in Masm.

Apparatus required: PC, MASM Software Tool

Flow Chart:



Programs:

ASSUME CS:CODE,DS:DATA

DATA SEGMENT

LIST DW 0053H,0019H,0025H,004BH ; list of array



Matrusri Engineering College Saidabad, Hyderabad

COUNT EQU 04H

; length of array

DATA ENDS

CODE SEGMENT

START: MOV AX,DATA

MOV DS,AX

MOV CX,COUNT ; move the count to CX

MOV BX,OFFSET LIST ; point BX to starting address of array

MOV AX,0000H ; clear ax

L1: ADD AX,[BX] ; add the 1st no. to ax

ADD BX,0002H ; get the address of 2nd no.

LOOP L1 ; add the all numbers to Ax by loop

MOV BX,COUNT ; move count to Bx

DIV BX ; divide the total sum by count to get average

INT 3H ; terminate the program

CODE ENDS

END START

Result:

Data: 0053h, 0019h, 0025h, 004bh

Sum = 00DCh

Average = Sum/Count = 0037h

Questions:

1. How do you define array of numbers in assembly language.
2. How does Queue differentiate between single byte and two byte instructions?
3. What is difference between JUMP and LOOP instructions?
4. What is the difference between control transfer instruction and sequential instructions?



Matrusri Engineering College Saidabad, Hyderabad

Ex.3

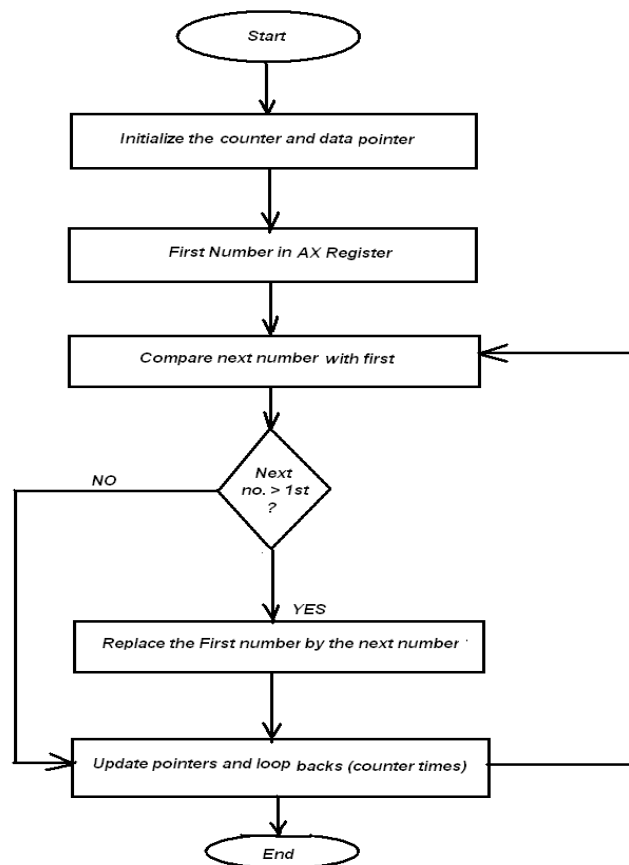
Find the Largest number in an array using Masm

Date:

Aim: To find Largest number of a given list of number using ALP program in Masm.

Apparatus required: PC, MASM Software Tool

Flow Chart:



Program:

ASSUME CS:CODE,DS:DATA

DATA SEGMENT

LIST DW 1234H,2354H,0056H,045AH ; list of array for 16bit numbers



Matrusri Engineering College Saidabad, Hyderabad

```
COUNT EQU 04H ; length of array
LARGEST DW 01H DUP(?) ; Memory location for result
DATA ENDS
CODE SEGMENT
START: MOV AX,DATA
MOV DS,AX
MOV SI,OFFSET LIST ; point si to starting address for array
MOV CL,COUNT ; move the length of array for count
MOV AX,[SI] ; move first no. to ax register
AGAIN: CMP AX,[SI+2] ; compare first and second numbers
JNL NEXT ; if carry is set 1st no. is lesser than 2nd
MOV AX,[SI+2] ; take the next no. in the array and compare
NEXT: ADD SI,02H
DEC CL ; decrement the count
JNZ AGAIN ; compare until count becomes 0.
MOV SI,OFFSET LARGEST ; move the address for largest no.
MOV [SI],AX ; store the largest in memory for AX.
INT 21H ;terminate the program
CODE ENDS
END START
```

Note: Instead of JNC change as JC for smallest number

Result: The largest number is **2354h**

The smallest number is **0056h**

Questions:

1. Give a list of all conditional JUMP instructions?
2. What are the branch instructions and string instructions?



Matrusri Engineering College Saidabad, Hyderabad

3. What are the different addressing modes of 8086?
4. What is difference between machine language and assembly language programmes?

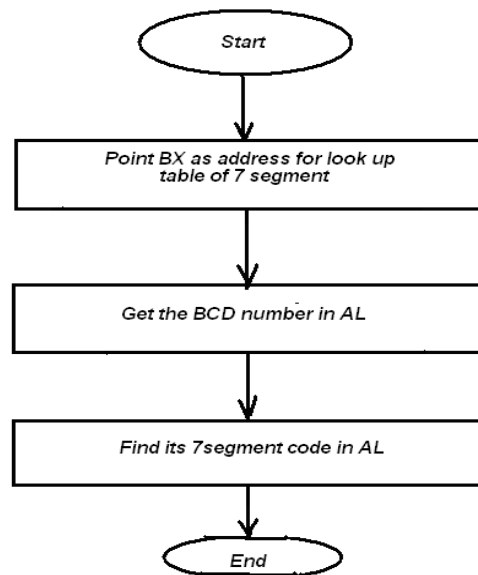
Ex.4 **Code conversion BCD number to 7-segment**

Date:

Aim: To convert the given BCD number to 7-segment

Apparatus required: PC, MASM Software Tool

Flow chart:



Program:

ASSUME CS:CODE,DS:DATA

DATA SEGMENT

**CODELIST DB 3FH,06H,5BH,4FH,47H,6DH,7DH,07H,7FH,67H ; list of array for
7segment value**

Char Equ 05 ; get the bcd no. to find 7segment value

DATA ENDS

CODE SEGMENT

START: MOV AX,DATA

MOV DS,AX



Matrusri Engineering College Saidabad, Hyderabad

MOV BX,OFFSET CODELIST ; set bx as starting address for look up table of
7 segment codes

MOV AL,CHAR ; get the bcd no. in AL

XLAT ; Exchange Equivalent 7 segment value

MOV AH,4CH

INT 21H

CODE ENDS ; Terminate The Program

END START

Result:

The given BCD number is: 05

The equivalent 7segment code is: 6dh

Questions:

1. What is meant by look-up table technique?
2. Explain XLAT instructions.
3. What is the difference between BCD and Decimal Number?
4. What are the DOS commands related to MASM?



Matrusri Engineering College Saidabad, Hyderabad

Ex.5

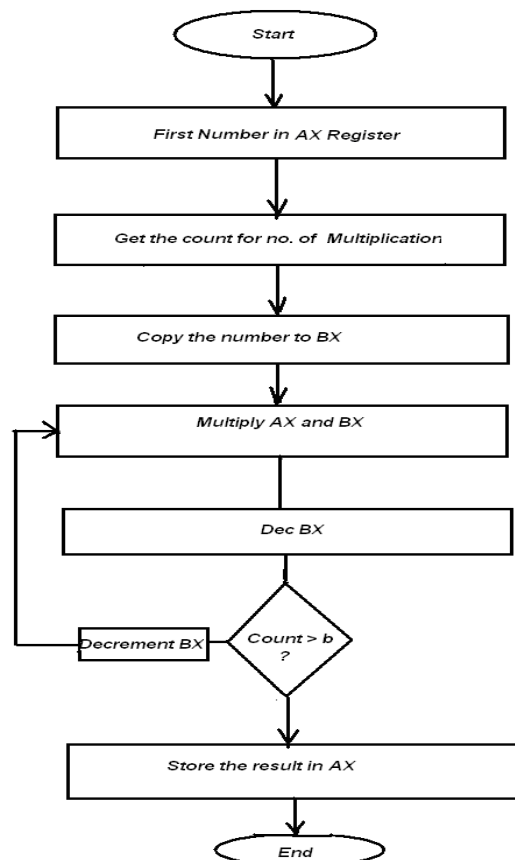
Find the Factorial using Masm

Date:

Aim: To find the factorial of a number between 1 and 8.

Apparatus required: PC, MASM Software Tool

Flow Chart:



Program:

ASSUME CS:CODE

CODE SEGMENT

START: MOV AX,0006H ; ax=0006h you can change the value for ax and



Matrusri Engineering College Saidabad, Hyderabad

; check

MOV CX,0000H ; clear Cx

ADD AX,CX ; add values of ax and cx

JZ EXIT1 ; if sum is 0 store the factorial of 0h, if sum is not 0

MOV BX,AX ; then move value in ax to bx

LOOP: DEC BX ; decrement bx value

JZ EXIT ; if BX is 0 terminate the program, otherwise

MUL BX ; multiply AX and BX

JMP LOOP ; continue the loop for multiplication until BX = 0

EXIT1: MOV AX,0001H

EXIT: INT 3H

CODE ENDS

END START

Result:

The factorial value for 0006h is : AX= 02D0h

Questions:

1. Write a program to find Fibonacci series for 5 numbers?
2. What is the factorial for number 0009H
3. What is purpose of assembler directives?
4. What type of errors will be indicated by assembler?



Matrusri Engineering College Saidabad, Hyderabad

Section: 2 Using 8086 Kit (interfacing)

Ex.1

Key Board Interface

Date:

Aim: To Interface key board with 8086 Microprocessor.

Apparatus required: 8086 Microprocessor Kit with power supply, 8279 Keyboard

Study card.

Program:

Address	Opcode	Label	Mnemonics	Comments
0000:2000	BB 00 00		MOV BX,00	
0000:2003	BA C2 FF		MOV DX,0FFC2	
0000:2006	B0 00		MOV AL,00	
0000:2008	EE		OUT DX,AL	Routine to clear
0000:2009	B0 90		MOV AL,90	All display LEDs
0000:200B	EE		OUT DX,AL	
0000:200C	B9 08 00		MOV CX,08	
0000:200F	B0 00		MOV AL,00	
0000:2011	BA C0 FF		MOV DX,0FFC0	
0000:2014	EE	RPT	OUT DX,AL	



Matrusri Engineering College Saidabad, Hyderabad

0000:2015	E2 FD		LOOP RPT	
0000:2017	BE 50 20	BCK	MOV SI,2050	
0000:201A	BA C2 FF		MOV DX,0FFC2	Read 8279 status
0000:201D	EC	KEY	INB AL,DX	To check if any
0000:201E	24 07		AND AL,07	character is
0000:2020	74 FB		JZ KEY	available
0000:2022	B0 40		MOV AI,40	
0000:2024	EE		OUT DX,AL	
0000:2025	BA C0 FF		MOV DX,0FFC0	Read FIFO RAM
0000:2028	EC		INB AL,DX	Mask SIFT and
0000:2029	24 1F		AND AL,1F	CTRL Keys
0000:202B	8A DB		MOV BL,AL	Point to key
0000:202D	03 F3		ADD SI,BX	Value
0000:202F	BA C2 FF		MOV DX,0FFC2	Address to
0000:2032	B0 94		MOV AL,94	Display 0
0000:2034	EE		OUT DX,AL	
0000:2035	BA C0 FF		MOV DX,0FFC0	



Matrusri Engineering College Saidabad, Hyderabad

0000:2038	B0 F3		MOV AL,F3	
0000:203A	EE		OUT DX,AL	
0000:203B	BA C2 FF		MOV DX,0FFC2	Address to
0000:203E	B0 95		MOV AL,95	Display scanned
0000:2040	EE		OUT DX,AL	value
0000:2041	BA C0 FF		MOV DX,0FFC0	
0000:2044	8A 04		MOV AL,[SI]	
0000:2046	EE		OUT DX,AL	
0000:2047	E9 CD FF		JMP BCK	

Look-up table for scan key values

Address	Opcode	Mnemonics
0000:2050	F3 60 B5 F4	DB F3, 60, B5, F4
0000:2054	66 D6 D7 70	DB 66, D6, D7, 70
0000:2058	F7 76 77 C7	DB F7, 76, 77, C7
0000:205C	93 E5 97 17	DB 93, E5, 97, 17

Result:

Questions:

1. What are the operating modes of 8279?



Matrusri Engineering College Saidabad, Hyderabad

2. Give the FIFO status of 8279?
3. What is meant by debouncing?

Ex.2

Generate of Triangular & Square waves using DAC

Date:

Aim: To generate the triangular and square waves using Digital to Analog converter.

Apparatus required: 8086 Microprocessor Kit with power supply, DAC Interface **study card**,
CRO

Program:

For Triangular wave:

Address	Opcode	Label	Mnemonics	Comments
			MOV AX,0000	
			MOV CS,AX	
			MOV ES,AX	
			MOV DX,0FFE6	Initialize all 8255
			MOV AL,80	Ports as o/p ports
			OUT DX,AL	
		RPT	MOV CX,0FF	Set count
			MOV AL,00	Start from 0
		UP	INC AL	Increment data for positive slope
			MOV DX,0FFE0	



Matrusri Engineering College Saidabad, Hyderabad

			OUT DX,AL	Output at port A & B
			MOV DX,0FFE2	
			OUT DX,AL	
			LOOP UP	
			MOV CX,0FF	Set count
			MOV AX,CX	Start from FF
		DOWN	DEC AL	Decrement data for Negative slope
			MOV DX,0FFE0	
			OUT DX,AL	O/p port A & B
			MOV DX,0FFE2	
			OUT DX,AL	
			LOOP DOWN	
			JMP RPT	

For Square Wave:

Address	Opcode	Label	Mnemonics	Comments
			MOV AX,0000	
			MOV CS,AX	
			MOV ES,AX	
			MOV DX,0FFE6	



Matrusri Engineering College Saidabad, Hyderabad

			MOV AL,80	
			OUT DX,AL	
		START	MOV AL,00	
			MOV DX,0FFE0	
			OUT DX,AL	
			CALL DELAY	
			MOV AL,FF	
			MOV DX,0FFE0	
			OUT DX,AL	
			CALL DELAY	
			JMP START	

Delay program:

Address	Opcode	Label	Mnemonics	Comments
2040			MOV BX,00FF	
2043		L1	DEC BX	
2044			JNZ L1	
2047			RET	

Result:

Questions:



Matrusri Engineering College Saidabad, Hyderabad

1. Give the pin diagram DAC 0808?
2. What is difference between triangular and ramp wave
3. How do you change the amplitude and time period in square wave?
4. What is the control word format of 8255?
5. What are the modes of operation of 8255?

Ex.3

Traffic Light Interface

Date:

Aim: To control flow of traffic by Traffic Light controller using 8086 kit.

Apparatus Required: 8086 Microprocessor Kit, Power Supply and Traffic Light Interface Module.

Theory: The interface provides a set of six LEDs at each of the four corners of a four junction road

The organization of these LEDs is identical at each of the four corners. The organization is described below with reference to the LEDs at south-west corner only.

RED: Referred to as SOUTH END henceforth

A: Referred to as SOUTH AMBER henceforth

L: Referred to as SOUTH LEFT henceforth

S: Referred to as SOUTH STRAIGHT henceforth

R: Referred to as SOUTH RIGHT henceforth

DL: Referred to as SOUTH PEDESTRAIN

(DL – set of two LEDs, one on either side of row)

The first five LEDs will be ON and OFF depending on the state of corresponding port line. The last one marked as DL is a set of two dual-color LEDs and they both will be either RED or GREEN depending on the state of the corresponding port line. The 24 LEDs and their corresponding port lines are summarized.



Matrusri Engineering College Saidabad, Hyderabad

DIRECTION	LED	PORTLINE
SOUTH	RED	PA3
	AMBER	PA2
	LEFT	PA0
	STRAIGHT	PC3
	RIGHT	PA1
	PEDESTRAIN	PC6
EAST	RED	PA7
	AMBER	PA6
	LEFT	PA4
	STRAIGHT	PC2
	RIGHT	PA5
	PEDESTRAIN	PC7
NORTH	RED	PB3
	AMBER	PB2
	LEFT	PB0
	STRAIGHT	PC1
	RIGHT	PB1
	PEDESTRAIN	PC4
WEST	RED	PB7
	AMBER	PB6
	LEFT	PB4
	STRAIGHT	PC0
	RIGHT	PB5
	PEDESTRAIN	PC5



Matrusri Engineering College Saidabad, Hyderabad

Example: determine the port lines values for the following traffic situation.

Vehicles from WEST are allowed to go NORTH or EAST.

Vehicles from EAST are allowed to go WEST.

Pedestrian crossing is allowed on SOUTH.

No other vehicle movement/pedestrian crossings are allowed.

DIRECTION	RED	AMBER	LEFT	STRAIGHT	RIGHT	PEDESTRIAN
SOUTH	ON	OFF	OFF	OFF	OFF	GREEN
EAST	OFF	OFF	ON	ON	OFF	RED
NORTH	ON	OFF	OFF	OFF	OFF	RED
WEST	OFF	OFF	OFF	ON	OFF	RED

Program:

Address	Opcode	Label	Mnemonics	Comments
2000	B0 80	START	MOV AL,80H	Initialization of 8255 Mode 0
2002	BA E6 FF		MOV DX,OFFE6H	
2005	EE		OUT DX,AL	All ports as O/P
2006	BE 38 20	AGAIN	MOV SI,2038H	Table of port



Matrusri Engineering College Saidabad, Hyderabad

2009	8A 04	NEXTS T	MOV AL,[SI]	values
200B	BA E0 FF		MOV DX,0FFE0H	
200E	EE		OUT DX,AL	Port A Value
200F	46		INC SI	
2010	83 C2 02		ADD DX,2	
2013	8A 04		MOV AL,[SI]	
2015	EE		OUT DX,AL	Port B Value
2016	46		INC SI	
2017	80 C2 02		ADD DX,2	
201A	8A 04		MOV AL,[SI]	
201C	EE		OUT DX,AL	Port C Value
201D	46		INC SI	
201E	E8 09 00		CALL DELAY	Calling Delay
2021	81 FE 56 20		CMP SI,2056H	
Checking for the end of the data values				
2025	75 E2		JNZ NEXTST	
2027	E9 DC FF		JMP AGAIN	
202A	B9 FF 00	DELAY	MOV CX,0FFH	Delay routine



Matrusri Engineering College Saidabad, Hyderabad

202D	51	DLY5	PUSH CX	
202E	B9 FF 03		MOV CX,03FFH	
2031	90	DLY10	NOP	
2032	E2 FD		LOOP DLY10	
2034	59		POP CX	
2035	E2 F6		LOOP DLY5	
2037	C3		RET	

2038	10 81 7A	PORT VALUES	10H,81H,7AH	State1
203B	44 44 F0		44H, 44H,F0H	All ambers ON
203E	08 11 E5		08H,11H,E5H	State 2
2041	44 44 F0		44H,44H,F0H	All ambers ON
2044	81 10 DA		81H,10H,DAH	State 3
2047	44 44 F0		44H,44H,F0H	All ambers ON
204A	11 08 B5		11H,08H,B5H	State 4
204D	44 44 F0		44H,44H,F0H	All ambers ON
2050	88 88 00		88H,88H,00H	State5
2053	44 44 F0		44H,44H,F0H	All ambers ON
2056	00		00H	Dummy



Matrusri Engineering College Saidabad, Hyderabad

Result:

Section: 3 Using 8051 Kit

EX: 1(a) Move A Block Of Data From Source To Destination

Aim: Write a program to move a block of data from source to destination

Apparatus required: 8051 kit, Power Supply and Keyboard.

Programs:

Address	Opcode	Label	Mnemonics	Comments
			MOV R0,#30	R0 □ SOURCE POINTER
			MOV R1,#60	R1 □ DESTINATION POINTER
			MOV R2,#05	R □ COUNTER
		AGAIN	MOV A,@R0	
			MOV @R1,A	
			INC R0	
			INC R1	
			DJNZ R2,AGAIN	DECREMENT R2 AND JUMP TO AGAIN AS LONG AS R2≠0
			SJMP	

Result:



Matrusri Engineering College Saidabad, Hyderabad

Source Address	Data	Destination Address	Data
30	10	60	10
31	20	61	20
32	30	62	30
33	40	63	40
34	50	64	50

EX: 1(b) Exchange A Block Of Data Between Source And Destination

Aim: Write a program to exchange a block of data between source and destination **Apparatus**

required: 8051 kit, Power Supply and Keyboard.

Programs:

Address	Opcode	Label	Mnemonics	Comments
			MOV R0,#40	R0 □ SOURCE POINTER
			MOV R1,#70	R1 □ DESTINATION POINTER
			MOV R2,#05	R □ COUNTER
		AGAIN	MOV A,@R0	
			XCH A,@R1	
			MOV @R0,A	
			INC R0	



Matrusri Engineering College Saidabad, Hyderabad

			INC R1	
			DJNZ R2,AGAIN	DECREMENT R2 AND JUMP TO AGAIN AS LONG AS R2=0
			SJMP	

Source Address	Data	Destination Address	Data
40	10	70	60
41	20	71	70
42	30	72	80
43	40	73	90
44	50	74	A0

Result:

Source Address	Data	Destination Address	Data
40	60	70	10
41	70	71	20



Matrusri Engineering College Saidabad, Hyderabad

42	80	72	30
43	90	73	40
44	A0	74	50

EX: 1(c)

Find the largest number in the array

Date:

Aim: Find the largest number in the given set of numbers.

Apparatus required: 8051 kit, Power Supply and Keyboard.

Programs:

Address	Opcode	Label	Mnemonics	Comments
8000	90 89 00	Loop4:	Mov DPTR,#8900	The total No. 'N' of data bytes are stored in 8900h data memory.
8003	E0		Movx A,@DPTR	
8004	FA		Mov R2,A	Reg.R2 is used as counter
8005	90 89 01		Mov DPTR,#8901	Data will be put from 8901h onwards
8008	E0		Movx A,@DPTR	



Matrusri Engineering College Saidabad, Hyderabad

8009	1A		Dec R2	Decrement counter
800A	F9		Mov R1,A	
800B	A3	Loop2:	INC DPTR	Increment data memory
800C	E0		Movx A,@DPTR	
800D	FB		Mov R3,A	
800E	99		SUBB A,R1	Comparing 2 Nos.
800F	50 0B		JNC LOOP1	
8011	DA F8	LOOP3 :	DJNZ R2,LOOP2	
8013	E9		MOV A,R1	
8014	F5 71		MOV 71,A	
8016	12 13 9E		LCALL 139E	Display the largest no.
8019	80 FE		SJMP 8019	
801B	EB	LOOP1	MOV A,R3	
801C	F9		MOV R1,A	
801D	02 80 11		LJMP LOOP3	
8020	12 13 9E		LCALL 139E	Display the largest no.
8023	02 80 00		LJMP LOOP4	

Procedure:

1. Switch ON the power supply.



Matrusri Engineering College Saidabad, Hyderabad

2. In the display it shows that ESA 51E.
3. In the next line it shows the symbol >.
4. Here you have to type A starting address of the program. i.e., >A8000
5. Then you start to type the each and every instruction of the program with the help of enter key.
6. Type all the instruction. You have press the Esc key.
7. Once again the same symbol appear in the display i.e., >.
8. Now you have to type the Data Memory address to load data i.e., MD 8900.
9. After loading the first data press enter key, and then it automatic shows the next address.
10. After enter the all data and instruction. Once again press the Esc key.
11. Now press the Key G and starting address of the instructions. i.e., G8000
12. You got the output directly in the display for this program.

Result:

The largest number in the given set of numbers is.....

Questions:

1. What is the difference between MOVC and MOVX instructions?
2. Explain the memory organization of 8051?
3. What are the different addressing modes of 8051?
4. Give applications of Microcontrollers.



Matrusri Engineering College Saidabad, Hyderabad

EX: 2

Arithmetic Instructions

Date:

Aim: Perform the Arithmetic Operation like Addition, Subtraction, Multiplication and Division.

Apparatus required: 8051 kit, Power Supply and Keyboard.

Programs:

(a) Addition: Add two 16-bit numbers FC45H and 02ECH.

Address	Opcode	Label	Mnemonics	Comments
8000			CLR C	Make CY=0
			MOV A,#45	Load the low byte into A
			ADD A,#0EC	Add the low byte now A=31 and CY=1
			MOV R0,A	Save the low byte of sum into R0
			MOV A,#02	Load the high byte into A
			ADDC A,#0FC	Add the high bytes with carry; 02+FC+1=FFH



Matrusri Engineering College Saidabad, Hyderabad

			MOV R1,A	Save the high byte of result to R1
HERE			SJMP HERE	Halt the program

(b)Subtraction: subtract two 16-bit numbers 1296h from 2762h

Address	Opcode	Label	Mnemonics	Comments
8000			CLR C	Clear CY
			MOV A,#62	A=62H
			SUBB A,#96	62H-96H=CC with CY=1
			MOV R5,A	Saving LSB of result in R5
			MOV A,#27	Save the result
			SUBB A,#12	27H-12H-1=14H
			MOV R6,A	Save the result
HERE			SJMP HERE	Halt the program

(c)Multiplication:

Address	Opcode	Label	Mnemonics	Comments
8000	74 82		MOV A,#82	Move the first data in Reg.A
8002	75 F0 02		MOV 0F0,#02	Move the Second data in Reg.B
8005	A4		MUL AB	Multiply the 2 data's
8006	FA		MOV R2,A	Move the result to Reg.R2
8007	80 80 07		SJMP 8007	Halt the program

(a) Division: convert 9Ch to decimal number using division

Address	Opcode	Label	Mnemonics	Comments
8000			MOV A,#9C	The Hex num is loaded in Reg.A
			MOV 0F0,#0A	Move the divisor into B
			DIV AB	Divide the hex num by 10



Matrusri Engineering College Saidabad, Hyderabad

			MOV R0,0F0	The remainder is in B, move it to reg R0
			MOV 0F0,#0A	Reload the divisor into B
			DIV AB	Divide the quotient by 10
			MOV R1,B	Move the remainder to R1
			MOV R2,A	Move the last quotient to R2

Result: the decimal number is in now R2, R1, R0 with R0 holding lowest digit.

Questions:

1. How many register banks are there in RAM and how do you select a particular register bank.
2. What is the purpose of B register?
3. Explain DPTR, PSW registers?
4. Draw the pin diagram of 8051?
5. How do you perform signed arithmetic operations.

EX: 3

Boolean and Logical Instructions

Date:

Aim: Perform the Boolean and Logical Instructions Operation like AND, OR, EX-OR and 2's Complement.

Apparatus required: 8051 kit, Power Supply and Keyboard.

Programs: AND OPERATION

Address	Opcode	Label	Mnemonics	Comments
			MOV A,#35H	Value 35h is stored in A register



Matrusri Engineering College Saidabad, Hyderabad

			ANL A,#0FH	The value 35h is AND ed with 0Fh
			SJMP	Program is terminated

OR OPERATION

Address	Opcode	Label	Mnemonics	Comments
			MOV A,#04H	Value 04h is stored in A register
			ORL A,#68H	The value 04h is OR ed with 68h
			SJMP	Program is terminated

EX-OR OPERATION

Address	Opcode	Label	Mnemonics	Comments
			MOV A,#64H	Value 64h is stored in A register
			XRL A,#78H	The value 64h is AND ed with 78h
			SJMP	Program is terminated

2's COMPLEMENT OPERATION

Address	Opcode	Label	Mnemonics	Comments
---------	--------	-------	-----------	----------



Matrusri Engineering College Saidabad, Hyderabad

			MOV A,#85H	Value 85h is stored in A register
			CPL A	Complement the value in A register
			ADD A,#01H	2's complement is obtained
			LCALL 03	Program is terminated

Result:

Questions:

1. What is the function of ALU in 8051?
2. What are the alternate functions of Port3
3. Explain the following ALE, EA and PSEN pins
4. Write an instruction to set first bit of port 0?
5. What are the bit addressable registers?

EX: 4

Jump and Call Instructions

Date:

(a) AIM: Find the sum of the values 79h, f5h and E2h using jump instruction. Put the sum in Registers R0 (Low byte) and R5 (High Byte)

Apparatus required: 8051 kit, Power Supply and Keyboard.

Program:

```
MOV A,#0           ; Clear Accumulator
MOV R5,A           ; clear R5
ADD A,#79h         ;A=0+79h=79h
```



Matrusri Engineering College Saidabad, Hyderabad

JNC N1	;If no carry add next number
INC R5	;If CY=1, increment R5
N1: ADD A,#0F5h	; A= 79+F5 = 6E and CY=1
JNC N2	;jump if CY=0
INC R5	; CY=1, Then increment R5 (R5=1)
N2: ADD A,#0E2h	;A=6E+E2=50 and CY=1
JNC OVER	;jump if CY=0
INC R5	; if CY=1, increment R5
OVER: MOV R0,A	; Now R0=50h and R5=02h

(B)AIM:To find the factorial of number 5 using CALL and RETURN instructions.

Program:

```
        MOV A,#05
        MOV R0,A
        LCALL 9000
        SJMP
9000: CJNE R0,#01,9004
        RET
9004: DEC R0
        MOV 0F0,R0
        MUL AB
        LJMP 9000
        SJMP
```

Result: A=78

C) using CALL instruction



Matrusri Engineering College Saidabad, Hyderabad

(i) Convert the given ASCII number to Decimal

(ii) Decimal to ASCII Numbers

Program:

(a) ASCII to Decimal

```
MOV R0,#34h
```

```
MOV R1,#35h
```

```
LCALL 9000
```

```
SJMP
```

```
9000: CLR C
```

```
MOV A,R0
```

```
SUBB A,#30
```

```
SWAP A
```

```
MOV R3,A
```

```
MOV A,R1
```

```
SUBB A,#30
```

```
ADD A,R3
```

```
RET
```

```
SJMP
```

Result: A=45h

(b) Decimal to ASCII

```
MOV A,#45
```

```
MOV R0,A
```

```
LCALL 9000
```

```
SJMP
```

```
9000: CLR C
```



Matrusri Engineering College Saidabad, Hyderabad

ANL A,#0F

ADD A,#30

MOV R1,A

MOV A,R0

SWAP A

ANL A,#0F

ADD A.#30

MOV R2,A

RET

SJMP

Result: R1=35 ; R2=34

Questions:

1. Explain the different JUMP and CALL instructions.
2. Explain DJNZ instructions?
3. Which register is used for serial communication?
4. After reset what is default address of stack pointer?

Ex.5 STEPPER MOTOR CONTROL BY 8051 MICROCONTROLLER

Date:

Aim: To control the Stepper Motor either by Clockwise Direction or Anticlockwise direction.

Apparatus required: 8051 kit, Power Supply and Keyboard, Stepper Motor Interface Module with power supply.

Programs:

SEG	EQU	0E8H	
P2	EQU	0A0H	
DPL	EQU	82H	
OUTPUT	EQU	03FAH	GETCH



Matrusri Engineering College Saidabad, Hyderabad

EQU

12A5H

GETKB

EQU

0142H

DSPCHR

EQU

03DCH

Note: Press Key “C” for clock wise rotation and “A” for anticlockwise rotation.

Address	Opcode	Label	Mnemonics	Comments
8000	90 80 75		MOV DPTR,#MSG	To display
8003	12 03 FA		LCALL OUTPUT	The message
8006	90 E1 02		MOV DPTR,#E102H	Check for serial
8009	E0		MOVX A,@DPTR	Or keyboard mode
800A	20 E3 05	CHK	JB ACC.3,KBD	
800D	12 12 A5		LCALL GETCH	Get the key code
8010	80 03		SJMP L1	
8012	12 01 42	KBD	LCALL GETKB	Get the key code
8015	B4 41 05	L1	CJNE A,#41H,L2	Checking for ‘A’
8018	12 03 DC		LCALL DSPCHR	Display character
801B	80 1B		SJMP L3	Jump to routine
801D	B4 61 05	L2	CJNE A,#61H,L4	Checking for ‘a’
8020	12 03 DC		LCALL DSPCHR	Display character
8023	80 13		SJMP L3	Jump to routine
8025	B4 43 05	L4	CJNE A,#43H,L5	Checking for ‘C’
8028	12 03 DC		LCALL DSPCHR	Display character
802B	80 22		SJMP L6	Jump for routine
802D	B4 63 05	L5	CJNE A,#63H,L7	Checking for ‘c’
8030	12 03 DC		LCALL DSPCHR	Display character
8033	80 1A		SJMP L6	Jump to routine
8035	02 00 00	L7	LJMP 0	

Routine for Clockwise Rotation.



Matrusri Engineering College Saidabad, Hyderabad

Address	Opcode	Label	Mnemonics	Comments
8038	75 A0 E8	L3	MOV P2,#SEG	Configure 8255
803B	78 03		MOV R0,#03H	For all ports O/P
803D	74 80		MOV A,#80H	
803F	F2		MOVX @R0,A	
8040	74 88		MOV A,#88H	Initialize bit Pattern. Bit PA0 - PA3 used to Control motor
8042	78 00	LOOP	MOV RO,#00H	O/P the pattern
8044	F2		MOVX @RO,A	To port A
8045	C0 E0		PUSH A	
8047	12 80 66		LCALL DELAY	
804A	D0 E0		POP A	
804C	23		RL A	
804D	80 F3		SJMP LOOP	

Routine For Anticlockwise Rotation.

Address	Opcode	Label	Mnemonics	Comments
804F	75 A0 E8	L6	MOV P2,#SEG	
8052	78 03		MOV R0,#03H	
8054	74 80		MOV A,#80H	
8056	F2		MOVX @R0,A	
8057	74 88		MOV A,#88H	
8059	78 00	LOOP 1	R0,#00H	
805B	F2		MOVX @R0,A	
805C	C0 E0		PUSH A	
805E	12 80 66		LCALL DELAY	
8061	D0 E0		POP A	



Matrusri Engineering College Saidabad, Hyderabad

8063	03		RR 3	
8064	80 F3		SJMP LOOP1	
8066	7A 03	DELAY	MOV R2,#03H	Delay routine
8068	90 05 00	DELAY2	MOV DPTR,#500H	
806B	A3	DELAY1	INC DPTR	
806C	E5 82		MOV A ,DPL	
806E	45 82		ORL A,DPL	
8070	70 F9		JNZ DELAY1	
8072	DA F4		DJNZ R2,DELAY 2	
8074	22		RET	

8075	45	4 E	54	4 5	52
807A	20	4F	50	5 4	49
807F	4 F	4 E	28	4 1	2 F
8084	43	29	3A	0 0	

Result:

USING „C“ Cross Compiler (KEIL Software)

ASSEMBLY LANGUAGE PROGRAM FOR ACTIVATING PORTS & GENERATION OF SQUARE WAVE

AIM: Write an assembly language program for generating square waveform using keil software (AT89C51).

APPARATUS:



Matrusri Engineering College Saidabad, Hyderabad

1. Keil software
2. P.C.

PROGRAM(1):

MOV SP,#7H

BACK: CLR P1.0

ACALL DELAY

SETB P1.0

ACALL DELAY

SJMP BACK

DELAY:MOV R1,#0FFH

AGAIN:DJNZ R1,AGAIN

RET

END

PROGRAM(2):

MOV SP,#7 ;initialize stack pointer

;since we are using subroutine programe

BACK:MOV P1,#00H ; send 00h on port 1 to generate

;low level of square wave

ACALL DELAY ; wait for some time

MOV P1,#0FFH ;send ffh on port 1 to generate

;high level of square wave

ACALL DELAY ; wait for some time

SJMP BACK ;repeat the sequence

DELAY:MOV R1,#0FFH ;load count



Matrusri Engineering College Saidabad, Hyderabad

AGAIN:DJNZ R1,AGAIN

;decrement count and repeat the
process

;until count is zero

RET

;return to main programe

EXPECTED RESULTS:

P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0

Program (1) : Activating PORT1

P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0

Program (2) : Activating Individual PORT1 pin 0

Exp. No.

ASSEMBLY LANGUAGE PROGRAM FOR ADDITION OF TWO NUMBERS



Matrusri Engineering College Saidabad, Hyderabad

AIM: Write an assembly language program for adding two 8-bit numbers using keil Software (AT89C51).

APPARATUS:

1. Keil software
2. P.C.

PROGRAM:

MOV A, #05H

MOV B, #02H

ADD A, B

END

RESULT:

In accumulator, a= 7h



Matrusri Engineering College Saidabad, Hyderabad

Exp. No.

ASSEMBLY LANGUAGE PROGRAM FOR MULTIBYTE ADDITION

AIM: Write an assembly language program for multibyte addition using keil software (AT89C51).

APPARATUS:

1. Keil software
2. P.C.

PROGRAM:

MOV R0,#20H

MOV R1,#30H

MOV R3,#04H

CLR C

CLR A

AGAIN: MOV A,@R0

ADDC A,@R1

MOV @R1,A

INC R0

INC R1

DJNZ R3,AGAIN

END

RESULT:

Inputs:

i: 0x20 -- 01h, 02h, 03h, 04h

i: 0x30 -- 05h, 06h, 07h, 08h



Matrusri Engineering College Saidabad, Hyderabad

Output:

i: 0x30 -- 06h, 08h, 0Ah, 0Ch

Exp. No.

ASSEMBLY LANGUAGE PROGRAM FOR ASCENDING ORDER OF A GIVEN NUMBERS

AIM: Write an assembly language program for arranging in ascending/descending order using keil software (AT89C51).

APPARATUS:

1. Keil software
2. P.C.

PROGRAM FOR ASCENDING ORDER:

```
MOV R0,#5          ; INITIALIZE COUNTER 1
AGAIN: MOV DPTR,#2000H ; initialize memory pointer
MOV R1,#4          ; initialize counter 2
BACK:  MOV R2,DPL    ; save lower byte of memory address
MOVX A,@DPTR       ; Get the number
MOV B,A           ; Save the number
INC DPTR           ; Increment the memory pointer
MOVX A,@DPTR       ; Get the next number
CJNE A,B,n         ; If not equal check for greater or less
AJMP SKIP          ; Otherwise go to skip
n:      JNC SKIP     ;If
MOV DPL,R2         ;Exchange
MOVX @DPTR,A
INC DPTR
```



Matrusri Engineering College Saidabad, Hyderabad

MOV A,B

MOVX @dptr,A

SKIP: DJNZ R1,BACK ;If R1 not equal to 0 go to BACK

DJNZ R0,AGAIN ;If R0 not equal to 0 go to
AGAIN

RESULT:

Inputs:

x: 0x2000 -- 05h, 02h, 01h, 04h

Output:

x: 0x2000 -- 01h, 02h, 04h, 05h



Matrusri Engineering College Saidabad, Hyderabad

Exp. No.

ASSEMBLY LANGUAGE PROGRAM FOR DESCENDING ORDER OF A GIVEN NUMBERS

AIM: Write an assembly language program for arranging in ascending/descending order using keil software (AT89C51).

APPARATUS:

1. Keil software
2. P.C.

PROGRAM FOR DESCENDING ORDER:

```
MOV R0,#5          ; INITIALIZE COUNTER 1

AGAIN:  MOV DPTR,#2000H ; initialize memory pointer

MOV R1,#4          ; initialize counter 2

BACK:    MOV R2,DPL          ; save lower byte of memory
          address

MOVX A,@DPTR      ; Get the num ber

MOV B,A          ; Save the number

INC DPTR          ; Increment the memory pointer

MOVX A,@DPTR      ; Get the next number
```



Matrusri Engineering College Saidabad, Hyderabad

CJNE A,B,n ; If not equal check for greater or less
AJMP SKIP ; Otherwise go to skip
n: JC SKIP ;If
MOV DPL,R2 ;Exchange
MOVX @DPTR,A
INC DPTR
MOV A,B
MOVX @dptr,A
SKIP: DJNZ R1,BACK ;If R1 not equal to 0 go to BACK
DJNZ R0,AGAIN ;If R0 not equal to 0 go to AGAIN

RESULT:

Inputs:

x: 0x2000 -- 05h, 02h, 01h, 04h

Output:

x: 0x2000 -- 05h, 04h, 02h, 01h



Matrusri Engineering College Saidabad, Hyderabad

Exp. No.

ASSEMBLY LANGUAGE PROGRAM FOR DATA TRANSFER

AIM: Write an assembly language program for block move from one address to another address using keil software (AT89C51).

APPARATUS:

1. Keil software
2. P.C.

PROGRAM:

MOV R0,#20H

MOV R1,#30H

MOV R3,#10H

CLR A

AGAIN:MOV A,@R0

MOV @R1,A



Matrusri Engineering College Saidabad, Hyderabad

INC R0

INC R1

DJNZ R3,AGAIN

END

RESULT:

Inputs:

i: 0x20 -- 01h, 02h, 03h, 04h,05h,06h,07h,08h,09h,0Ah

Output:

i: 0x30 -- 01h, 02h, 03h, 04h,05h,06h,07h,08h,09h,0Ah

ADDITIONAL EXPERIMENTS USING MASM

COPYING A STRING

Aim: Write a program in Assembly language to copy a string of 6 bytes.

Apparatus: i. 8086 kit and its power supply

ii. MASM software

Program:

ON KIT

CLD

MOV SI,5000

MOV DI,6000

MOV CX,0006

REP

MOSB

INT 03



Matrusri Engineering College Saidabad, Hyderabad

ON MASM

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

DATA SEGMENT

SOURCE_STR equ 2000H

LEN_STR DW 01 DUP(?)

DATA ENDS

EXTRA SEGMENT

DEST_STR EQU 5000H

EXTRA ENDS

CODE SEGMENT

START: MOV AX,DATA

MOV,DS,AX

MOV AX,EXTRA

MOV ES,AX

XOR CX,CX

MOV SI,OFFSET SOURCE_STR

MOV DI,DEST_STR

MOV AL,'\$'

L2: CMP AL,[SI]

JZ L1

INC CX

INC SI

JMP L2

L1: MOV SI,OFFSET SOURCE_STR

MOV LEN_STR,CX

CLD

REP MOVSB

INT 03

CODE ENDS

END START



Matrusri Engineering College Saidabad, Hyderabad

Result:

CX=0006

Input

5000	05
5001	A8
5002	B6
5003	CF
5004	F7
5005	A2

Output

6000	05
6001	A8
6002	B6
6003	CF
6004	F7
6005	A2

String is being copied.

COMPARE TWO STRINGS

Aim: Write an assembly language program to compare two strings.

Apparatus:

1. 8086 trainer kit and power supply
2. MASM software.

Procedure:

1. Clear direction flag (DF=0 for auto increment mode)
2. Set data segment, extra segment, source and destination index
3. Set counter
4. Compare source string bytes and destination strings bytes until counter value=0

Program:

```
CLD
MOV SI,5000
MOV DI,6000
MOV CX,0006
```



Matrusri Engineering College Saidabad, Hyderabad

```
REPE
CMPSB
INT 03
IN MASM
ASSUME CS:CODE,DS:DATA,ES:EXTRA
DATA SEGMENT
SOURCE_STR EQU 2000H COUNT EQU 3000H
DATA ENDS
EXTRA SEGMENT
DEST_STR EQU 5000H
EXTRA ENDS
CODE SEGMENT
STAR: MOV AX,DATA
MOV DS,AX
MOV ES,AX
XOR CX,CX
MOV SI,COUNT MOV CX,[SI]
MOV SI,SOURCE_STR
MOV DI,DEST_STR
CLD
REPE CMPSB
INT 03
CODE ENDS
END START
```

Result:

Case(i):Equal

5000 10 6000 10



Matrusri Engineering College Saidabad, Hyderabad

5001	20	6001	20
5002	0D	6002	0D
5003	AB	6003	AB
5004	03	6004	03
5005	01	6005	01

Case(ii): Unequal

5000	10	6000	10
5001	20	6001	20
5002	0A	6002	0b