

DEPARTMENT OF
ELECTRONICS & COMMUNICATION ENGINEERING

LABORATORY MANUAL
FOR

MICROPROCESSORS &
MICROCONTROLLERS

(III B.Tech. - II – Sem.)



BALAJI INSTITUTE OF TECHNOLOGY & SCIENCE
Laknepally, Narsampet, Warangal



BALAJI INSTITUTE OF TECHNOLOGY & SCIENCE

Laknepally(V), Narsampet(M), Warangal(Dist).

Dept. of Electronics & Communication Engineering

List of Experiments:-

1. Programs for 16 Bit Arithmetic Operations for 8086 (Using Various Addressing Modes)
2. Program for Sorting an Array for 8086
3. Program for Searching A for A Number or Character in a String for 8086
4. Program for String Manipulations for 8086
5. Program for Digital Clock Design using 8086
6. Interfacing ADC and DAC to 8086
7. Parallel Communication between two Microprocessors using 8255
8. Serial Communication Between two Microprocessor Kits using 8251
9. Interfacing to 8086 and Programming to Control Stepper Motor
10. Programming Using Arithmetic, Logical and Bit Manipulation Instructions of 8051
11. Program and Verify Timer/Counter In 8051
12. Program and Verify Interrupt Handling in 8051
13. UART Operation in 8051
14. Communication between 8051 Kit and PC
15. Interfacing LCD to 8051
16. Interfacing Matrix/Keyboard to 8051
17. Data transfer from Peripheral to Memory through DMA Controller 8237/8257

1. Programs for 16 bit Arithmetic Operations for 8086 (Using Various Addressing Modes)

1.1) AIM: TO WRITE 8086 ALP TO ADD, SUB, MUL, DIV TWO 16-BIT NUMBERS.

APPARATUS:

1. 8086 mp kit -1 & Adapter-1
2. System-1
3. RS 232 Serial cable.
4. +5v Supply
5. 86/88e Driver Software.

PROGRAM CODE:-

```
XOR AX,AX
MOV ES,AX
MOV DI,3000
XOR BX,BX
XOR CX,CX
XOR DX,DX
MOV AX,5555
MOV BX,2222
ADD AX,BX
MOV [DI],AX
INC DI
INC DI
MOV AX,5555
SUB AX,BX
MOV [DI],AX
INC DI
INC DI
MOV AX,5555
```

```
MUL BX
MOV [DI],AX
INC DI
INC DI
INC DI
INC DI
MOV AX,5555
XOR DX,DX
DIV BX
MOV [DI],AX
INC DI
INC DI
MOV[DI],DX
INT 03
```

I/P:AX=5555 BX=2222

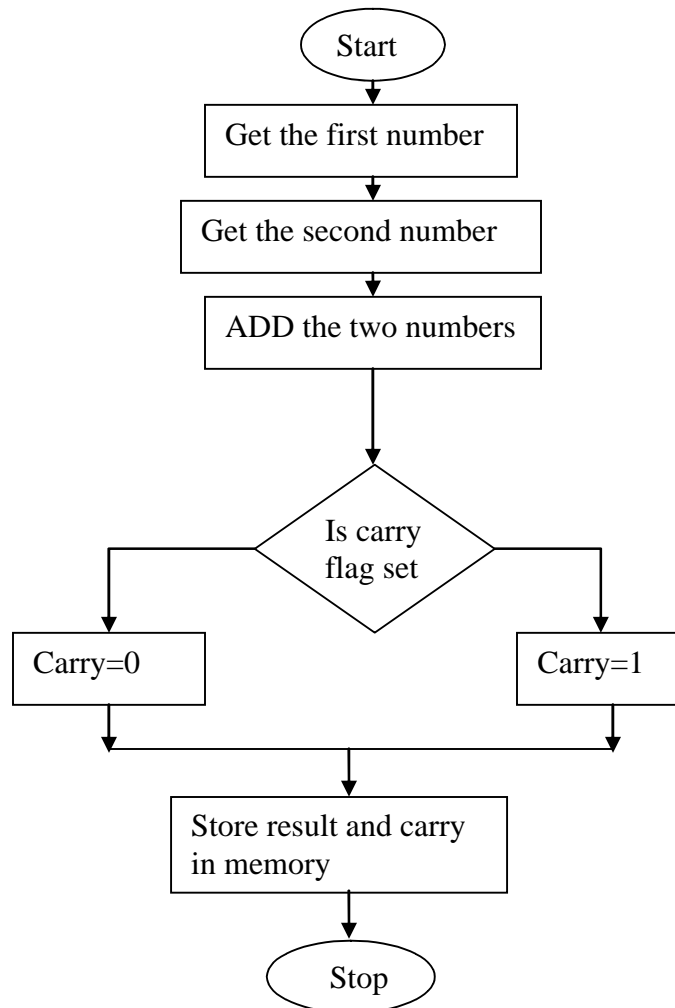
O/P: ES:DI

```
0000:3000 77}
0000:3001 77 } Add(AX)
0000:3002 33}
0000:3003 33} Sub(AX)
0000:3004 4A}
0000:3005 9F} AX
0000:3006 60}
0000:3007 0B } DX MUL(AX,DX)

0000:3008 02}
0000:3009 00} AX Quotient}

0000:300A 11}
0000:300B 11} DX Reminder } DIV
```

Flow chart:



1.2 ADDITION OF TWO 32-BIT DATA

AIM: To add two 32-bit data (Multibyte addition) using 8086 microprocessor.

APPARATUS:

1. 8086 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

```
MOV AX,[0300]
MOV BX,[0304]
ADD AX,BX
MOV [030A],AX
MOV AX,[0302]
MOV BX,[0306]
ADC AX,BX
MOV [030C],AX
INT A5
```

RESULT:

Input Data (Before Execution)

0000:0300	78
0000:0301	56
0000:0302	34
0000:0303	12
0000:0304	65
0000:0305	87
0000:0306	21
0000:0307	43

Output Data (After Execution)

0000:030A	DD
0000:030B	DD
0000:030C	55
0000:030D	55

1.3 SUBTRACTION OF TWO 32-BIT DATA

AIM: To subtract two 32-bit data (Multibyte subtraction) using 8086 microprocessor.

APPARATUS:

1. 8086 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

```
MOV AX,[0300]
MOV BX,[0304]
SUB AX,BX
MOV [030A],AX
MOV AX,[0302]
MOV BX,[0306]
SBB AX,BX
MOV [030C],AX
INT A5
```

RESULT:

Input Data (Before Execution)

0000:0300	44
0000:0301	44
0000:0302	44
0000:0303	44
0000:0304	11
0000:0305	11
0000:0306	11
0000:0307	11

Output Data (After Execution)

0000:030A	33
0000:030B	33
0000:030C	33
0000:030D	33

1.4 .MULTIPLICAION OF TWO 16-BIT DATA

AIM: To multiply two 16-bit data (Multibyte multiplication) using 8086 microprocessor.

APPARATUS:

1. 8086 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

```
MOV AX,[0300]
MOV BX,[0302]
MUL BX
INT A5
```

RESULT:

Input Data (Before Execution)

0000:0300	44
0000:0301	44
0000:0302	11
0000:0303	11

Output Data (After Execution)

AL	44
AH	44
DL	00
DH	00

1.5. DIVISION OF TWO 16-BIT DATA

AIM: To multiply two 16-bit data (Multibyte division) using 8086 microprocessor.

APPARATUS:

1. 8086 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

```
MOV AX,[0300]
MOV BX,[0302]
DIV BX
INT A5
```

RESULT:

Input Data (Before Execution)

0000:0300	44
0000:0301	44
0000:0302	22
0000:0303	22

Output Data (After Execution)

AL	22
AH	22
DL	00
DH	00

Program:1.6 MULTIPLICAION OF SIGNED NUMBERS

AIM: To multiply two signed numbers (8-bit data) using 8086 microprocessors.

APPARATUS:

1. 8086 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

```
MOV AX,0200
MOV DS,AX
MOV AL,[1500]
NEG AL
MOV BL,[1501]
MUL BL
MOV [1505],AX
INT A5
```

RESULT:

Input Data (Before Execution)

2000:1500	10
2000:1501	15

Output Data (After Execution)

2000:1505	B0
2000:1506	13

2.PROGRAM FOR SORTING AN ARRAY FOR 8086

2:1: To find the smallest number

AIM: Write a Program to find smallest number from a given array of numbers.

APPARATUS:

1. 8086 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

```
XOR AX,AX
MOV DS,AX
MOV SI,5000
MOV CL,06
MOV AL,[SI]
L1:  INC SI
     MOV BL,[SI]
     CMP AL,BL
     JL 7014(L2)
     XCHG AL,BL
L2:  LOOP 700B(L1)
     INT 03
```

RESULT: The smallest number is given array of number is:

AX=009D FL=F097

2:2 To find the Greatest number

AIM: Write a Program to find Greatest number from a given array of numbers

APPARATUS:

1. 8086 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

```
XOR AX,AX
MOV DS,AX
MOV SI,5000
MOV CL,06
MOV AL,[SI]
L1:  INC SI
     MOV BL,[SI]
     CMP AL,BL
     JG 7014(L2)
     XCHG AL,BL
L2:  LOOP 700B(L1)
     INT 03
```

RESULT: The smallest number is given array of number is:

AX=007C FL=F006

2:3 SORTING STRING OF 'N' NUMBERS IN ASCENDING/DESCENDING ORDER

AIM: To write a program to sort a given string of a number in ascending/descending 8086 microprocessor programming.

APPARATUS:

1. 8086 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

```

                                MOV CX,0005
                                DEC CX
Again    MOV DX,CX
                                MOV SI,0200
Up       MOV AL,[SI]
                                INC SI
                                MOV BL,[SI]
                                CMP AL,BL
                                JLE/JGE    Next
                                XCHG AL,BL
                                MOV [SI],BL
                                DEC SI
                                MOV [SI],AL
                                INC SI
Next    DEC DX
                                JNZ        Up
                                DEC CX
                                JNZ        Again
                                INT A5
```

RESULT

Input Data (Before Execution)

0000:0200	38
0000:0201	47
0000:0202	02
0000:0203	11
0000:0204	29

Output Data (After Execution)

	Ascending order	Descending order
0000:0200	02	47
0000:0201	11	38
0000:0202	29	29
0000:0203	38	11
0000:0204	47	02

2.4 SUM OF SQUARES OF 'N' NUMBER

AIM: To find the sum of squares of data string by using 8086 assembly language program.

APPARATUS:

1. 8086 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

```
MOV SI,[0300]
MOV CL,[SI]
MOV DI,0220
MOV BX,0000
MOV AH,00
Up INC SI
MOV AL,[SI]
MUL AL
ADD [DI],AX
DEC CL
JNZ      Up
INT A5
```

RESULT

Input Data

0000:0300	05
0000:0301	01
0000:0302	02
0000:0303	03
0000:0304	04
0000:0305	05

Output Data

BX	0037
----	------

2.5 SUM OF CUBES OF 'N' NUMBERS

AIM: To find the sum of cubes of an array of size 10 by using 8086 assembly language program.

APPARATUS

1. 8086 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

```
MOV SI,0200
MOV DI,0220
MO CL,0A
MOV AX,0000
MOV [DI],AX
MOV AL,[SI]
MOV BL,AL
MUL AL
MUL BL
ADD [DI],AX
Up INC SI
DEC CL
JNZ Up
INT A5
```

RESULT

Input Data

0000:0200	01
0000:0201	02
0000:0202	03
0000:0203	04
0000:0204	05
0000:0205	06
0000:0206	07
0000:0207	08
0000:0208	09
0000:0209	0A

Output Data

0220	D1
0221	0b

3 Program for Searching A for A Number in a Sting for 8086.

AIM: PAM for searching a number or a character in a string.

APPARATUS:

1. 8086 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

XORW AX, AX

XORW BX, BX

MOVW DS,AX

MOVW SI, 6000

MOVW CX, 0007

MOVB AL,09

MOVB BL,[SI]

CMP AL, BL

JZ : L1

INCW SI

LOOP L2

MOV DX, SI

INT 03

RESULT:

Input Data (Before Execution)

0000:6000	26
0000:6001	86
0000:6002	95
0000:6003	09
0000:6004	48
0000:6005	43
0000:6006	91

Output Data (After Execution):register Contents:

AX=0009 BX=0009 CX=0004 DX=0000 SP=0100 SI=6003

FL=F046

RESULT: A PROGRAM FOR SEARCHING OF AN NUMBER (OR)
CHARACTER IN A STRING HAS BEEN PERFORMED.

3.2 TO SEARCH A STRING USING SCAS-SCAN THE STRING.

AIM: WAP to find the required string (or) number from the given string using SCAS Required element is EE and mentions the address in the register 'DX'.

APPARATUS:

1. 8086 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

```
XOR DX, DX
MOV AX, AX
MOV ES, AX
MOV DI, 6000
MOV CL, 07
MOV AL, 0EE
REPZ
SCASB
DEC DI
MOV DX, DI
INT 03
```

RESULT:

Input Data (Before Execution)

0000:6000	- 55
0000:6001	- AB
0000:6002	-D5
0000:6003	-FD
0000:6004	- 9L
0000:6005	- EE
0000:6006	-34
0000:6007	-66

Here DX IS Not equal to zero and the required element is present in the register Dx having the Address 6005.

4. PROGRAM FOR STRING MANIPULATIONS FOR 8086 COMPARISON OF TWO STRINGS.

AIM: To compare two data strings using 8086 microprocessor programming

APPARATUS:

1. 8086 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

```
MOV AX,2000
MOV DS,AX
MOV ES,AX
MOV SI,0100
MOV DI,0200
MOV CX,0005
MOV BX,0000
CLD
REP CMPSB
JE Last
MOV BX,FFFF
INT A5
```

RESULT

Input Data (Before Execution)

DS:SI		ES:DI	
2000:0100	01	2000:0200	01
2000:0101	02	2000:0201	02
2000:0102	03	2000:0202	03
2000:0103	04	2000:0203	04
2000:0104	05	2000:0204	05

Output Data (After Execution)

BX FFFF

4.1 LENGTH OF A DATA STRING

AIM: To find the length of a given string of data using 8086 microprocessor programming.

APPARATUS:

1. 8086 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

```
MOV AX,2000
MOV ES,AX
MOV DI,0100
MOV CX,0000
MOV AL,00
CLD
Again SCASB
      JZ   Last
      INC CX
      JMP  Again
Last  INT A5
```

RESULT

Input Data (Before Execution)

2000:0100	44
2000:0101	67
2000:0102	49
2000:0103	20
2000:0104	00

Output Data (After Execution)

CX 0004

4.2 MOVING A STRING OF DATA

AIM: To move a sting or a block of data from one segment to the other segment using 8086 microprocessor programming.

APPARATUS:

1. 8086 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

```
MOV AX,2100
MOV DS,AX
MOV AX,2200
MOV ES,AX
MOV SI,2000
MOV DI,0000
MOV CX,0005
CLD
REP MOVSB
INT A5
```

RESULT

Input Data (Before Execution)

2000:2000	23
2000:2001	45
2000:2002	21
2000:2003	78
2000:2004	69

Output Data (After Execution)

2200:0000	23
2000:0001	45
2000:0002	21
2000:0003	78
2000:0004	69

4.3 REVERSE OF A STRING

AIM: To write a program to reverse of a string 8086 microprocessor programming.

APPARATUS:

1. 8086 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

```
MOV AX,2000
MOV DS,AX
MOV CX,0005
MOV SI,0200
MOV DI,0209
Next MOV AL,[SI]
XCHG AL,[DI]
MOV [SI],AL
INC SI
DEC DI
JNZ Next
INT A5
```

RESULT

Input Data (Before Execution)

2000:0200	00
2000:0201	01
2000:0202	02
2000:0203	03
2000:0204	04
2000:0205	05
2000:0206	06
2000:0207	07
2000:0208	08
2000:0209	09

Output Data (After Execution)

2000:0200	09
2000:0201	08
2000:0202	07
2000:0203	06
2000:0204	05
2000:0205	04
2000:0206	03
2000:0207	02
2000:0208	01
2000:0209	00

5 PROGRAM FOR DIGITAL CLOCK DESIGN USING 8086

AIM: Write a Program for a Digital Clock

REQUIREMENTS:

Hardware:-

- 1) PC
- 2) 8086 MP Trainer Kit
- 3) RS 232 Connector
- 4) +5v Adaptor

Software:-

WIN 86 Driver Software.

Program that places a message on the screen every 10 seconds , using int lah;

```
CODE SEGMENT
TIMEDELAY:
MOV SP,1000H
MOV DI ,10XD
TIMEOUT:
MOV AH,00H
INT 1AH
MOV BX,DX
TIMER:
MOV AH, 00H
INT 1AH
SUB DX,BX
CMP DX,182XD
JC TIMER
MOV AH,09H
CS MOV DX,MSG
INT 21H
DEC D1
JNZ TIMEOUT
```

```
MOV AX,4C00H
```

```
INT 21H
```

```
MSG:
```

```
DB 'TEN MORE SECONDS HAVE PASSED $'
```

```
CODE ENDS
```

```
19
```

```
20
```

RESULT:-We have designed digital clock using 8086.

6.ANALOG TO DIGITAL CONVERSION

AIM: To interface analog to digital converter with 8086 microprocessor through 8255 and display the digital equivalent of the analog input voltage.

APPARATUS:

1. 8086 Trainer kit
2. Key board
3. SMPS
4. CRO
5. Interfacing cable with probe

PROGRAM CODE:

```
MOV DX,8807
MOV AL,81
OUT DX,AL
MOV DX,8803
MOV AL,00
OUT DX,AL
MOV DX,8807
MOV AL,09
OUT DX,AL
MOV AL,08
OUT DX,AL
MOV AL,83
OUT DX,AL
INT AC
Repeat MOV DX,8807
MOV AL,0D
OUT DX,AL
MOV AL,0C
OUT DX,AL
MOV DX,8805
UP IN AL,DX
```

```
AND AL,02
JNZ      Up
Again   IN AL,DX
AND AL,02
JZ      Again
MOV AL,0B
MOV DX,8807
OUT DX,AL
MOV DX,8803
IN AL,DX
MOV CL,AL
MOV DX,8807
MOV AL,0A
OUT DX,AL
INT AB
MOV AL,02
MOV DX,CX
NOP
MOV DH,00
INT AE
MOV AH,0B
INT A1
AND AL,FF
JZ      Repeat
INT A3
```

RESULT: 8255 PPI is interfaced with 8086 in mode 0 with port A, B, and C as output ports.

8255 address	PORT A	8801
	PORT B	8803
	PORT C	8805
	CWR	8807

Output:

6. PARALLEL COMMUNICATION BETWEEN TWO MICROPROCESSORS USING 8255

AIM: Parallel communication between 2 microprocessors using 8255

SOFTWARE USED: Term 86E or Win 86E

HARDWARE REQUIRES:

- 1) PC
- 2) ESA 86/88E Unit
- 3) RS 232 Cable
- 4) Adopter +5V

TRANSMITTING PROGRAM :

```
MOV DX,0FFE6
MOV AL,80
OUT DX,AL
MOV DX,0FFE0
MOV AL,29
OUT DX,AL
INT 03
```

RECEIVING PROGRAM:

```
MOV DX,0FFE6
MOV AL,90
OUT DX,AL
MOV SI,5000
MOV CL,06
MOV DX,0FFE0
IN [AL],DX
L1: MOV [SI],AL
    INC SI
    LOOP L1
INT 03
```

RESULT: AX = 0099, SI = 5006, FL = F006 ,

S 5000	0000:5000 ---29
	0000:5001 ---29
	0000:5002 ---29
	0000:5003 ---29
	0000:5004 ---29
	0000:5005 ---29

8. SERIAL COMMUNICATION BETWEEN TWO MICROPROCESSOR KITS USING 8251

AIM:To write a program for serial communication between two microprocessor kits using 8251

REQUIREMENTS:

HARDWARE:

- 1) PC
- 2) 8086 MP Trainer Kit
- 3) RS 232 Connector
- 4) +5V Adaptor
- 5) 8251 Study Card

PROGRAM CODE:

```
START:  MOV AL,EEH
        OUT 81H,AL
        MOV AL,15H;
        OUT 81H,AL
RECEIVE: IN AL,81H;
        ROR AL,1
        ROR AL,1
        JNC RECEIVE
        IN AL,80H;
        NOT AL;
TRANS:  IN AL,81H;
        ROR AL,1
        JNC TRANS
        MOV AL,BL;
        OUT 80H,AL
        JMP RECEIVE;
        ENDS
```

RESULT:-

9.INTERFACING TO 8086 AND PROGRAMMING TO CONTROL STEPPER MOTOR

AIM: Write a program to rotate stepper motor in clock wise direction.

APPARATUS:

1. 8086 Trainer kit
2. Key board
3. SMPS
4. Stepper motor interfacing kit

PROGRAM CODE:.

```
MOV DX,8006
MOV AL,80
OUT DX,AL
MOV CL,01
MOV DX,8000
MOV AL,88
OUT DX,AL
CALL      Delay
ROR AL,CL
JMP      Up
INT A5
```

DELAY PROGRAM

```
Delay    MOV CX,FFFF
         INT AA
         RET
```


PROGRAMMING WITH 8051

10 PROGRAMMING USING ARITHMETIC, LOGICAL AND BIT MANIPULATION INSTRUCTIONS OF 8051

10.1 ADDITION OF TWO NUMBERS

AIM: To add two numbers by using 8051 microcontroller.

APPARATUS:

1. 8051 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

```
MOV A,#24
MOV F0,#42
ADD A,F0
RET
```

RESULT

Input Data

A	24
B	42

Output Data

A	66
---	----

10.2 SUBTRACTION OF TWO NUMBERS

AIM: To subtraction two numbers by using 8051 microcontroller

APPARATUS:

1. 8051 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE;

```
MOV A,#44  
MOV F0,#37  
CLR C  
SUB A,F0  
RET
```

RESULT

Input Data

A	44
B	37

Output Data

A	0D
---	----

10.3 MULTIPLICATION OF TWO NUMBERS

AIM: To multiply the given two numbers by using 8051 microcontroller

APPARATUS:

1. 8051 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

```
MOV A,#22  
MOV F0,#11  
MUL AB  
RET
```

RESULT

Input Data

A	22
B	11

Output Data

A	42
B	02

10.4 DIVISION OF TWO NUMBERS

AIM: To multiply the given numbers by using 8051 microcontroller

APPARATUS:

1. 8051 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE;

```
MOV A,#22
MOV F0,#11
DIV A
RET
```

RESULT

Input Data

A	22
B	11

Output Data

A	02
B	00

10.5 USE OF SWAP INSTRUCTION

AIM: To show the use of SWAP instruction of 8051 microcontroller

APPARATUS:

1. 8051 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

```
MOV A,#50  
SWAP A  
MOV R0,A  
RET
```

RESULT:

Input Data

A 50

Output Data

A 05

10.6 LARGEST OF TWO NUMBERS

AIM: To find the largest of two numbers using 8051 microcontroller.

APPARATUS:

1. 8051 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

```
MOV R0,#05
MOV R1,#08
MOV A,R0
CLR C
SUBB A,R1
JNC      Down
MOV A,R1
RET
Down    MOV A,R0
RET
```

RESULT:

Input Data

R0	05
R1	08

Output Data

A	08
---	----

10.7 FACTORIAL OF A NUMBERS

AIM: To find the factorial of a given numbers using 8051 microcontroller.

APPARATUS:

1. 8051 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

```
MOV A,#05
MOV R1,#04
MOV F0,R1
MUL AB
DJNZ R1,FB
RET
```

RESULT:

Input Data

A	05
R1	04

Output Data

A	78
---	----

10.8 EQUALIZING THE GIVEN NUMBERS

AIM: To increment and decrement the given two numbers until they are equal using 8051 microcontroller programming.

APPARATUS:

1. 8051 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

```
MOV R2,#04
MOV R5,#08
INC R2
Up MOV A,R2
DEC R5
CJNE A,05 Up
RET
```

RESULT:

Input Data

A	05
R1	04

Output Data

A	78
---	----

10.9 SQUARE OF A NUMBERS

AIM: To find the square of a given number using subroutine using 8051 microcontroller programming.

APPARATUS:

1. 8051 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

```
MOV R0,#02
MOV R1,#03
MOV A,R0
LCALL    Routine
RET
```

SUBROUTINE:

```
Routine  MOV A,R1
          MOV F0,R1
          MUL AB
          RET
```

RESULT:

Input Data

R1 03

Output Data

A 09

10.10 CUBE OF A NUMBERS

AIM: To find the cube of a given number using 8051 microcontroller programming.

APPARATUS:

1. 8051 Trainer kit
2. Key board
3. SMPS
- 4.

PROGRAM CODE:

```
MOV R1,#04
MOV A,R1
MOV F0,R1
MUL AB
MOV F0,R1
MUL AB
RET
```

RESULT:

Input Data

R1 04

Output Data

A 40

10.11 UNPACKED TO PACKED BCD CONVERSION

AIM: To pack the given two BCD numbers using 8051 microcontroller programming.

APPARATUS:

1. 8051 Trainer kit
2. Key board
3. SMPS
- 4.

PROGRAM CODE:

```
MOV A,#04
MOV F0,#05
SWAP A
ADD A,F0
RET
```

RESULT:

Input Data

A	04
B	05

Output Data

A	45
---	----

10.12 CHECKING AN 'N'TH BIT

AIM: To check whether n-th bit of given data is '1' or '0' using 8051 microcontroller programming.

APPARATUS:

1. 8051 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

```
MOV A,#10
AND A,#08
JNZ      Down
MOV A,#55
RET
Down    MOV A,#FF
RET
```

RESULT:

Input Data

A 10

Output Data

A FF

10:13 COMPARE TWO NUMBERS

AIM: To compare two numbers and store 00 in A if data1=data2, FF in A if data1>data2 and 55 in A if data1<data2 using 8051 microcontroller programming.

APPARATUS:

1. 8051 Trainer kit
2. Key board
3. SMPS

PROGRAM CODE:

```
                MOV A,#09
                CJNE A,#08      Down
                MOV A,#00
                RET
Down           JC          Last
                MOV A,#FF
                RET
Last          MOV A,#55
                RET
```

RESULT:

Input Data
A 09

Data2 in prog. 08

Output Data

A FF

11:PROGRAM & VERIFY TIMER AND COUNTER FUNCTIONS

11.1:Blink LED'S

AIM:WAP to blink LED'S Timer 0,mode1,initial value000H.

HARDWARE REQUIREMENTS:

- 1) PC
- 2) 8051 micro controller trainer kit
- 3) RS 232 Connector
- 4) +5V Adaptor

SOFTWARE REQUIREMENTS:-

- 1) KEIL micro version
- 2) Flash magic software

PROGRAM CODE:

```
ORG 0000H
AKR:MOV A,#00
MOV TMOD,#01
MOV TH0,#00
MOV TL0,#00
MOV P0,A
SETB TR0
JNB TF0,$
CLR TR0
CLR TF0
CPL A
SJMP AKR
END
```

RESULT: we observed that the LED'S are blinked using timer / counter..

11.2 SUB TITLE:GENERATE THE SQUARE WAVE

AIM: WAP to generate the square wave (timer0,mode1,initial value 0000h)

HARDWARE REQUIREMENTS:

- 1) PC
- 2) 8051 micro controller trainer kit
- 3) RS 232 Connector
- 4) +5V Adaptor

SOFTWARE REQUIREMENTS:-

- 1) KEIL micro version
- 2) Flash magic software

PROGRAM CODE:

```
Program
ORG 000H
SETB P2.0
MOV TMOD,#01H
K: MOV TH0,#00
MOV TL0,#00
SETB TR0
JNB TF0,$
CLR TR0
CLR TF0
CPL P2.0
SJMP K
END
```

RESULT:Amplitude =6v,frequency=3.8*50m sec.We have observed the square wave using timer/counter of 8051.

12:PROGRAM &VERIFY INTERRUPT HANDLING IN 8051.

12.1:INTERRUPT HANDLING.

AIM: WAP to transmit data serially data ECEA

HARDWARE REQUIREMENTS:

- 1) PC
- 2) 8051 micro controller trainer kit
- 3) 3)RS 232 Connector
- 4) +5V Adaptor

SOFTWARE REQUIREMENTS:-

- 1) KEIL micro version
- 2) Flash magic software

PROGRAM CODE:

```
ORG 0000H
LJMP RK
ORG 0003H
SJMP LEDS
ORG 0013H
SJMP TRANS
ORG 0030H
RK:    MOV IE, #85H
      MOV TCON, #01H
      SJMP $
LEDS:  MOV R0,#0FFH
      MOV A,#00H
L1:    MOV P0,A
      MOV TMOD,#00H
      MOV TH0,#00H
      MOV TL0,#00H
      SETB TR0
      JNB TF0,$
```

```
CLR TR0
CPL A
DJNZ PR0,L1
RET1
TRANS: MOV R1,#0AH
MOV TMOD,#20H
MOV TH1,#FDH
MOV SCON,#40H
CLR T1
SETB TR1
L2 MOV SBUF,# 'X'
JNB T1,$
CLR T1
DJNZ R1,L2
RET1
```

RESULT: VERIFIED THE FUNCTION OF INTERRUPT HANDLING IN 8051.

13 Program & verify UART operation in 8051.

13.1: Transmit data serially.

AIM: WAP to transmit data serially data ECEA.

HARDWARE REQUIREMENTS:

- 1) PC
- 2) 8051 micro controller trainer kit
- 3) 3)RS 232 Connector
- 4) +5V Adaptor

SOFTWARE REQUIREMENTS:-

- 1) KEIL micro version
- 2) Flash magic software

PROGRAM CODE:

```
ORG 0000H
MOV TMOD,#20H
MOV TH1,30FDH
MOV SCON,#40H
CLR T1
SETB TR1
RK    MOV SBUF #'B'
      ACALL TRANS
      MOV SBUF #'1'
      ACALL TRANS
      MOV SBUF #'T'
      ACALL TRANS
      MOV SBUF #'S'
      ACALL TRANS
      LJMP RK
TRANS JNB T1,$
      CLR T1
      RET1
```

END

RESULT: Verified the UART operation, transmitted the data serially.

13.2) RECEIVE data serially.

AIM:-W.A.P to RECEIVE data serially.

HARDWARE REQUIREMENTS:

- 1) PC
- 2) 8051 micro controller trainer kit
- 3) RS 232 Connector
- 4) +5V Adaptor

SOFTWARE REQUIREMENTS:-

- 1) KEIL micro version
- 2) Flash magic software

PROGRAM CODE:

```
ORG 0000H
MOV TMOD,#20H
MOV TH1,#0FDH
RK  MOV SCON,#50H
CLR R1
SETB TR1
JNB R1,$
MOV A,SBUF
MOV SBUF,A
SJMP RK
END
```

14: TITLE: COMMUNICATION BETWEEN 8051 MICRO CONTROLLER.

AIM:

HARDWARE REQUIREMENTS:

- 1) PC
- 2) 8051 micro controller trainer kit
- 3) RS 232 Connector
- 4) +5V Adaptor

KIT&PC serial communication

8051>H

HELP MENU

D Display data,program,internal,bit memory or registers

E Edit data,program,internal,bit memory or registers

S Single step from specified address ,press SP to terminate

G Execute the program till user break

B Set address till where the program is to be executed

C Clear break points

F10 key followed by 4 key at the PC to upload data to a file (DOS)

T Test the onboard peripherals

: Download a file from pc mem to the SDA-SI-MEL kit(DOS)

A Assembler

Z Disassembler

TEST FOR ONBOARD PERIPHERALS

For SDA SI-MEL kit , following menu is displayed on pressing the option "T"

8051>T

ALS- SDA SI-MEL kit Test monitor

- 1.Test internal Data RAM
- 2.Test external data Memory(U6)
- 3.Test external data Memory(U7)
- 4.8253 loop test
- 38
- 5.Test 8253
- 6.Exit

Select(1-6):

Suppose the user presses the key '1', following message is displayed if the internal data RAM is OK.

Testing internal data Ram: pass

After displaying the message, is displayed once again waits for user to enter a key

EDITING MEMORY COMMAND:

8051>E

EDIT(R,B,M,P,D)..D-EXTERNAL DATA RAM

Enter STA address=0400

0400=7f:55 press 'N' key to go to the next address

0401 =D5:66

0402=D3:77

0403=73:88

0404=6F:12

0405=CB:01

0406=A7:02 press 'p' key to the previous address

0407=6F:03

0408=7B:04

0409=29:05

040A=6F:06

040B73:07

040C=FF:08

040D=7D:09 press 'CR' key to have same address

040E=09:09 press 'ESC' key to about the command

39

8051>E

EDIT(R,B,M,D)..B-BITS

Enter STA address=00

00=0:1

01=0:1

02=0:0

03=0:1

03=0:

03=1:

02=0:

8051>E

EDIT(R,B,,P,D)...R-REGISTORS

ACC=0:33

PSW=00:44

DPH=00:55

DPL=00:00

8051>E

EDIT (R,B,M,P,D)..-P=PROGRAM CODE

8000=FF:78

8001=FF:10

8002=FF:79

8003=ff:20

8004=FF:7A

8005=FF:12

8007=FF:00

8008=FF:03

8009FF:0F

8051>E

EDIT (R,B,M,P,D)...-M-INTERNAL RAM

40

0000=00:12

0001=00:34

0002=00:00

DISPLAY COMMAND

8051>D

EDIT(R,B,M,P,D)..-EXTERNAL DATA RAM

Enter END address=040F

0500 55 66 77 88 12 01 02 03 05 06 07 08 09 04 D7

SETTING BREAK COMMAND:

8051>B

BR_NO:R

BR_ADD 0000

ERROR! ONLY A BREAKS ALLOWED

8051>B

BR_NO:0

ERROR! BREAK NUMBERS MUST BE BETWEEN 1&8

CLEAR BREAK COMMAND:

8015>C

BR_NO:A Clears all the break point set by the user

8051>C

BR_NO:1Clears the break point number 1

41

PROGRAMAE EXECUTION COMMAND:

8051>G

PROGRAM EXECUTION

ENTER START ADDRESS=8000

ACC PSW DPH DPL PCL SP B R0 R1 R2 R3 R4 R5 R6 R7 33 44 55 00 10 34 00 00

00 00 00 00

ASSEMBEL MEMORY COMMAND

8051>A

ENTER START ADDRESS=8000

DISASSEMBLE MEMORY COMMAND

8051>Z

42

RESULT:

15-WRITE A PROGRAM TO VERIFY LCD USING 8051

AIM:-W.A.P to LCD

HARDWARE REQUIREMENTS:

- 1) PC
- 2) 8051 micro controller trainer kit
- 3) 3)RS 232 Connector
- 4) +5V Adaptor

SOFTWARE REQUIREMENTS:-

- 1) KEIL micro version
- 2) Flash magic software

PROGRAM CODE:

```
μOrg 000h;starting  
  
Mov a,#38h,Intiliaze for two lines  
  
acall command  
  
acall delay  
  
Mov a,#0ch;cursor on  
  
acall command  
  
acall delay  
  
Mova,#01h;clear LCD  
  
acall command  
  
acall delay  
  
Mov a,#06h; shift the cursor towards right  
  
acall command  
  
acall delay  
  
Mov a,#80h; cursor at line 1
```

acall command

acall delay

Mov tmod, #20h; select the timer 1 with mode 2

Mov th1, #0fd; select the baudrate as 9600bps

Mov scon, #50h; initialize transmitter with mode 1

clr ri

Setb tr1; start the timer 1

Rk:jnbri,\$; wait for character

Mov a,sbuf,a

Acall data

Clr ri

Sjmp rk

RESULT:

16.KEY BOARD INTERFACING WITH 8051

AIM:

HARDWARE REQUIREMENTS:

- 1.PC
- 2.8051 MC Trainer Kit
- 3.RS 232 Connector
- 4.+5V Adaptor
- 5.4X4 Matrix Keyboard

PROGRAM:

Keyboard subroutine. This program sends the ASCII code for pressed key to p0.1;p1.0-p1.3 connected to rows p2.0-p2.3 connected to columns

```
MOV P2,#0FFH    ;make p2 an input poart
K1:MOV P1,#0     ;ground all rows at once
MOV A,P2        ;Read all col.ensure all keys open
ANL A,#00001111b ;masked unused bits
CJNE A,#00001111B,K1;check til all keys released
K2:ACALL DELAY  ;call 20ms delay
MOV A,P2;see if any key is pressed
ANL A,#00001111B;mask unused bits
CJIT A,#00001111B,OVER;key pressed, await closure
SJMP K2;check if key pressed
OVER:ACALL DELAY; wait 20m debounce time
MOV AP2;check key closure
```

```
ANL A,#00001111B,OVER;key pressed ,fine row
SJMP K2;if none,keep polling
OVER1:MOV P1,#1111110B;ground row 0
MOV A,P2;read all columns
ANL A,#00001111B0;mask unused bits
CJNE A,#00001111B,ROW_1; key row 0,fine the col.
MOV A,P1,#1111110B;ground row 2
MOV A,P2;read all columns
ANL A,#00001111B;mask unused bits
CJNE A,#00001111B,ROW_1;key row 1 find the col.
MOV P1,#11111011B ;ground row 2
MOV A,P2;read all columns
ANL A,#00001111B;mask unused bits
CJNE A,#00001111B,ROW_2key row 2,find the col.
Mp1,#11111011B;ground row 3
MOV A,P2; read all columns
ANL A,#00001111B;mask unused bits
CJNE A,#00001111B,ROW;key row 3,find the col.
LJMP K2
ROW0:MOV DPTR,#KCODE0 ;set DPTR=start of row 0
SJMP FIND ;find col.key belongs to
ROW1:MOV DPTR,#KCODE1 ; set DPTR=start of row 1
SJMP FIND ; find col.key belongs to
ROW2:MOV DPTR,#KCODE2 ; set DPTR=start of row2
SJMP FIND ; find col.key belongs to
```

ROW3:MOV DPTR,#KCODE3 ; set DPTR=start of row3

FIND:RRC A ; if any CY bit is low

JNC MATCH ;if zero,get the ASCII code

INC DPTR ;point to next col.Address

SJMP FIND ;keep searching

MATCH:CLR,A ;set A=(match if found)

MOVC A,@+DTPR; get ASCII code from table

MOV P0,A;display pressed key

LJMP K1

ASCII LOOK-UP TABLE FOR EACH ROW

ORG 300H

KCODE0:DB '0','1','2','3' ;ROW 0

KCODE1:DB '4','5','6','7' ; ROW 1

KCODE2:DB '8','9','A','B' ;ROW 2

KCODE3:DB 'C','D','E','F' ;ROW 3

END

END

17)Data transfer from peripheral to memory through DMA

controller 8257

AIM: To interface DMA Controller 8257 with 8086,for moving 2KB of data from a peripheral device to memory.

REQUIREMENTS:

SOFTWARE: TERM 86E or WIN 86E

HARDWARE: Personal computer

RS-232C

FRC cable

+5V adaptor

ESA 86/88 kit

DMA kit

PROGRAM :

MOV AX,DATA; initialize data segment

MOV DS,AX;

MOV AX,DMAH; load DMA address register with lower
byte of DMA address

OUT 80H,AX;

MOV AX,DMAH; load byte of DMA address register of
channel 0

OUT 80H,AX;

MOV AX,TCH; load higher byte TC register

OUT 81H,AX;

MOV AX,MSR; MODE SET Register initialization,

;The F/L flip flap is assumed to be cleared

OUT 88H,AX;

MOV AH,4CH;

Latch segment address on A16-A19

INT 21H; externally,i.e.0010(2H) and wait for the DMA

request

; after the request is serviced the

;CPU may continue the execution

Input :

MSR: Mode set register content 4141H

DMAL: DMA address lower byte 0000h

DMAH: Terminal count higher byte 5050H

TCL: Terminal count byte FFFFH

TCH: Terminal count higher byte 4747H

RESULT: