LABORATORY MANUAL

MICROPROCESSORS AND MICROCONTROLLERS-II

T.E (ELEX) (Sem. VI)

Index

PIC

16	Addition of five numbers.
17	Subtraction of two numbers.
18	Multiple of 4
19	Multiplication two 16 bit number
20	Display a count of 0-F on seven segment display
21	Interface LCD to Port C and Port D of PIC

Lab Session 1

Title: TO ADD TWO 8 BIT NUMBERS.

Objective:

Program involves storing the two 8-bit no in memory locations and adding them taking into consideration the carry generated. The objective of this program is to give an overview of arithmetic instructions of 8086 for 8-bit operands.

Pre-Requisites:

1. Knowledge of TASM directives.
2. Knowledge of Arithmetic Instructions of 8086 for 8-bit operands.

Algorithm:

1. Initialize the data segment.
2. Store two 8-bit numbers in memory locations.
3. Move the 1st number in any one of the general purpose register.
4. Move the 2nd number in any other general purpose register.
5. Add the 2 numbers.
6. Store the result in memory location.
7. Check for carry flag. If carry flag is set then store '1' as MSB of result.
8. Stop

Post Lab Assignments:

1. Explain arithmetic instructions of 8086.
2. Explain TASM directives.
3. Explain the instances when the carry flag is set.

Lab Session 2

Title : TO MULTIPLY TWO 16 BIT NUMBERS

Objective: Program involves storing the two 16 bit numbers in memory locations and multiplying them the objective of this program is to give an overview of arithmetic instructions of 8086 for 16 bit operation.

Pre-Requisites:

1. Instructions of microprocessor 8086
2. Addressing mode of microprocessor 8086.
3. Flag register of microprocessor 8086
4. Knowledge of TASM directories.

Algorithm:

1. Initialize the data segment.
2. Store two16 bit numbers in memory locations.
3. Move the 1st number in any one of the general purpose register.
4. Move the 2nd number in any other general purpose register.
5. Multiply the 2 numbers.
6. Store the result in memory location.
7. Stop.

Post Lab Assignment**s:**

1. List of condition of the flag in the status register after MUL instructions has been executed.
2. Some 8086 mnemonics which is used for multiplication and storing a Data.

Lab Session 3

Title: TO IMPLEMENT BLOCK TRANSFER.

Objective:

Program involves transferring source string from a particular location in source segment (Data Segment) to the desired location in destination segment (Extra Segment).
The objective of this program is to give an overview of the String instructions of 8086.

Pre-Requisites:

1. Knowledge of TASM directives.
2. Knowledge of String Instructions of 8086.

Algorithm:

1. Initialize the data segment.
2. Store the source string in consecutive memory location.
3. Initialize the extra segment.
4. Allocate consecutive memory locations for transfer.
5. Load the effective address of source string in SI register.
6. Load the effective address of destination string in DI register.
7. Initialize the Direction flag for Auto increment or Auto decrement.
8. Store number of bytes to be transferred in any of the general purpose registers.
9. Transfer the source string using appropriate string instructions (MOVSB / MOVSW)
10. Decrement count.
11. Check if count = 0.If yes then stop else repeat steps 9 - 11.
12. Stop

Post Lab Assignments:

1. Explain string instructions.
2. Explain the purpose of direction flag.
3. Explain the REP prefix.
4. Explain the difference between CMP and CMPSB

Lab Session 4

Title: TO FIND AVERAGE OF 10 NUMBERS.

Objective:

Program involves averaging of a ten numbers.

Pre-Requisites:

1. Knowledge of TASM directives.
2. Knowledge of CMP and Jump Instructions of 8086.

Algorithm:

1. Initialize the data segment.
2. Clear carry flag.
3. Store initial address of list in a register.
4. Move 0A to register AL.
5. Store count in register AL.
6. Reset AL registers (content of AX)
7. Add one number from array to contents of AL
8. Decrement count
9. Increment SI
10. If count $\neq 0$ go to step 8
11. Divide content of AX with that of BL.
12. Store answer in register.
13. Stop.

Post Lab Assignments:

1. Write different instructions used.

Lab Session 5

Title: TO FIND EVEN AND ODD NUMBERS IN A GIVEN STRING.

Objective:

Program involves counting the even and odd numbers in a given array. The objective of this program is to give an overview of the String instructions of 8086.

Pre-Requisites:

1. Knowledge of TASM directives.
2. Knowledge of String Instructions of 8086.

Algorithm:

1. Initialize the data segment.
2. Initialize the array.
3. Load the effective address of array in any general purpose register.
4. Load total number of elements of the array in any register.
5. Test LSB of CX register. If LSB =1then no. is odd else no. is even.
6. Store result in DX register.
7. Stop.

Post Lab Assignments:

1. Explain shift & rotate instructions.

Lab Session 6

Title: ARRANGING NUMBER IN ASCENDING ORDER.

Objective:

Program involves sorting an array in ascending order using Bubble sort algorithm.
The objective of this program is to give an overview of the Compare and Jump instructions. Use of Indirect Addressing mode for array addressing is expected.

Pre-Requisites:

1. Knowledge of TASM directives.
2. Knowledge of CMP and Jump Instructions of 8086.

Algorithm:

1. Initialize the data segment.
2. Initialize the array to be sorted.
3. Store the count of numbers in a register.
4. Store count-1 in another register.
5. Load the effective address of array in any general purpose register.
6. Load the first element of the array in a register.
7. Compare with the next element of the array.
8. Check for carry flag.
9. If carry=0 first number > second number. Swap the 2 numbers.
10. Increment to the contents of the SI register so that it points to the next element of the array.
11. Decrement (count-1) by 1.
12. Check if (count-1) =0. If no then repeat steps 7 to 11.
13. Decrement count by 1.
14. Check if count = 0.If no then repeat steps 6 through
15. Stop.

Post Lab Assignments:

1.Explain CMP and Jump Instructions.

Lab Session 7

Title: FINDING THE GREATEST NUMBER FROM THE GIVEN ARRAY.

Objective:

To find the greatest number from given array of data.
The objective of this program is to give an overview of the compare and jump instruction.

Pre-Requisites:

1. Knowledge of TASM directives.
2. Knowledge of CMP and Jump instructions of 8086.

Algorithm:

1. Initialize the data segment.
2. Initialize the array.
3. Initialize data pointer with the starting address of the array.
4. Initialize a register as counter=number of element in the array.
5. Load the first element of the array in a register
6. Increment the data pointer.
7. Compare the number in register with the next element of array.
8. Check for carry flag.
9. If carry=0,first number (in register)>second number. Go to step 10.
If carry=1, first number (in register) <second number. Copy the greater number in register.
10. Decrement counter by 1.
11. Check if counter=0.If not, Repeat steps 6 to 11.
12. The greatest number is obtained in the register.
13. Stop

Post Lab Assignments:

1. Explain Various Jump instructions of 8086.

Lab Session 8

Title: DISPLAY CHARACTER FROM KEYBOARD UNTIL 0 IS ENTERED.

Objective:

To Read Character from Keyboard and display on screen until 0 is pressed.

Pre-Requisites:

1. Knowledge of TASM directives.
2. Knowledge of DOS Interrupt.

Theory: Instructions used in program are:

- MOV AH,08H
  INT 21H

Read Input From Keyboard without echo and store at AL.

- **MOV AH,02H**
  INT 21H

Display Character on screen. Character should be in DL register.

Algorithm:

1. Initialize the data segment.
2. Read input from keyboard.
3. Compare input with ASCII value of ZERO.
4. If result is 0, go to step 7.
5. Move content of AL to DL, to display it on screen.
6. Display character on screen.
7. Stop

Lab Session 9

Title: DISPLAY A TO Z ON SCREEN.

Objective:

To store A to Z Alphabets on an array and display them on user screen.

Pre-Requisites:

1. Knowledge of TASM directives.
2. Knowledge of String instructions of 8086 and DOS interrupt.

Algorithm:

1. Initialize the data segment.
2. Store all Alphabets in array.
3. Initialize counter to 1AH.
4. Load starting Address of array in to SI.
5. Get each character in DL.
6. Display Character on user screen.
7. Increment SI.
8. Decrement counter.
9. Repeat step 5 to 8 until count becomes Zero.
10. Stop

Lab Session 10

Title: PASSWORD VERIFICATION

Objective:

The objective is to make use of string instruction and MACRO, to check whether the entered password by the user is correct or not.

Pre-Requisites:

1. Knowledge of TASM directives.
2. Knowledge of DOS interrupt.
3. Knowledge of string instruction and MACRO.

Algorithm:

1. Store Initial password into Array.
2. Write Macro for printing output message.
3. Write Macro to display '*'.
4. Initialize the data segment.
5. Set the counter value=no. of character present in password.
6. Load Effective address of stored password in BX.
7. Take input from the keyboard.
8. Compare input with the password string.
9. If zero=0, both value are equal. Go to step 10.
   If zero is not equal to 0. Go to step 15.
10. display '*' Macro.
11. Increment BX.
12. Decrement counter by 1.
13. Check if counter=0.If not, Repeat step 7 to 12.
14. Display Macro message for correct password, Go to step 16.
15. Display '*' macro and Macro message for wrong password.
16. End

.

Post Lab Assignments:

1. Explain Macro.

Lab Session 11

Title: TO WRITE A PROGRAM FOR 3 X 3 MATRIX MULTIPLICATION.

Objective:

The objective is to multiply 3 X 3 matrix.

Pre-Requisites:

1. Knowledge of TASM directives.
2. Knowledge of DOS interrupts.
3. Knowledge of string instruction and MACRO.

Algorithm:

1. Initialize Data segment.
2. Set the counter
3 .Multiply row1 of matrix1 with column1 of matrix2 and add all multiplication Answer.
4. Repeat step 3 for all other rows n columns.
5. Store final answer at the memory location defined for matrix3.

Post Lab Assignments:

1. Explain Indirect addressing mode of 8086 with examples.

Lab Session 12

Title: TO WRITE A PROGRAM FOR 3 X 3 MATRIX ADDITION

Objective:

The objective is to multiply 3 X 3 matrix.

Pre-Requisites:

1. Knowledge of TASM directives.
2. Knowledge of DOS interrupts.
3. Knowledge of string instruction and MACRO

Algorithm:

1. Initialize the data segment.
2. Initialize counter = 9
3. Initialize pointer DI to matrix 1.
4. Initialize pointer Bx to matrix 2.
5. Initialize pointer SI to result matrix 3.
6. Get the number from matrix 1.
7. Add number from matrix 1 with matrix 1 number.
8. Save the carry if any.
9. Save the result in result matrix 3.
10. Increment DI, BX, and SI to point to next element.
11. Decrement count.
12. Check if count = 0,if not go to step VI else go to step XIII
13. Display the result.
14. Stop.

Lab Session 13

Title: TO COMPUTE POWER OF NUMBER X^N USING DOS INTERRUPT.

Objective:

The objective is to make use of string instruction and MACRO, to check whether the entered password by the user is correct or not.

Pre-Requisites:

1. Knowledge of TASM directives.
2. Knowledge of DOS interrupts.
3. Knowledge of string instruction and MACRO.

Algorithm:

1. Initialize data segment
2. Write a macro for reading a value from the keyboard
3. Write a macro for printing input message and result
4. Initialize the code segment.
5. Call the macro to input the number X & N.
6. Write a loop for calculating x^N .
7. Convert the above computed value obtained in step 5 to ASCII value.
8. Call the macro to point the result.

Post Lab Assignments:

1. Explain Macros and Procedure .Explain its difference.

Lab Session 14

Title: TO BINARY NUMBER INTO GRAY CODE NUMBER.

Objective:

The objective is to make use of string instruction and MACRO, to check whether the entered password by the user is correct or not.

Pre-Requisites:

1. Knowledge of TASM directives.
2. Knowledge of DOS interrupts.
3. Knowledge of string instruction and MACRO.

Algorithm:

1.Get the number whose equivalent is to be found.
2. Add number with itself
3. XOR this added result with the original number.
4. Shift the XOR ed number by 1 bit position to the right to get the gray equivalent.
5. Display the result.
6. Stop.

Lab Session 15

Title: TO WRITE A PROGRAM TO FIND WHETHER GIVEN STRING IS PALINDROME USING DOS INTERRUPTS.

Objective:

The objective is to make use of string instruction and MACRO, to check whether the entered password by the user is correct or not.

Pre-Requisites:

1. Knowledge of TASM directives.
2. Knowledge of DOS interrupts.
3. Knowledge of string instruction and MACRO.

Algorithm:

I. Initialize the data and stack memory
II. Using Macro display Menu.
    1. Accept
    2. Length
    3. Reverse
    4. Palindrome
    5. Exit
III. Accept choice from user using INT 21h, function 01h
IV. Is choice is =1 jump to step XI else go to step V
V. Is choice is =2 jump to step XIV else go to step VI
VI. Is choice is =3 jump to step XVII else go to step VII
VII. Is choice is =4 jump to step XX else go to step VIII
VIII. Is choice is =5 jump to step XXIII else go to step IX
IX. Display wrong choice
X. Jump to step II
XI. Call procedure accept.
XII. Accept string using INT 21H, function 0AH.
XIII. Return to main program and go to step II.
XIV. Call procedure Length.
XV. Calculate the length of string and display it using INT 21H, Function 02h.
XVI. Return back to main program and jump back to step II.
XVII. Call procedure Reverse.
XVIII. Reverse the string and display it.
XIX. Return back to main program and jump to step II
XX. Call procedure palindrome.
XXI. Check if string is palindrome, if yes display string is palindrome else string is not a palindrome.
XXII. Return back to main program and jump to step II.
XXIII. Terminate the program and stop.

Post Lab Assignments: Explain different addressing mode of 8086 with example.

Lab Session 16

Title: TO WRITE A PROGRAM TO ADD 5 DATA BYTES STORED IN THE PROGRAM MEMORY. DISPLAY THE RESULT AT PORT B AND PORT C.

Objective: The objective is to make use of PIC Arithmetic instructions and use it.

Pre-Requisites:

1. Knowledge of MPLAB.
2. Knowledge of PIC 18 instruction set.

Algorithm:

1. Initialise Port C as output Port.
2. Initialise counter for data bytes.
3. Initialize FSRO pointer for data register.
4. Clear WREG register to save the sum.
5. Set up the register to save carry and clear it.
6. Use ADD Instruction with direct addressing mode & post increment mode so that it refers to data pointed by FSRO.
7. Check if carry is set if yes increment carry register initiated in steps else go to by FSRO.
8. Decrement the counter initiated in step 2.
9. Check if counter =0, if yes display result at PORT B & PORT C else go to step 6.

Post Lab Assignments:

1. Explain Different Arithmetic Instructions.
2. Explain different indirect addressing modes for data registers.

Lab Session 17

Title: TO WRITE A PROGRAM TO SUBTRACT 2 NUMBERS.

Objective: The objective is to make use of PIC Arithmetic instructions and use it.

Pre-Requisites:

1. Knowledge of MPLAB.
2. Knowledge of PIC 18 instruction set.

.

Algorithm:

1. Initialise Port C as output Port.
2. Initialise counter for data bytes.
3. Initialize FSRO pointer for data register.
4. Clear WREG register to save result.
5. Set up the register to save carry and clear it.
6. Use subtract Instruction with indirect addressing mode & pre increment mode so that it refers to data pointed by FSRO +1.
7. Check if carry is set if yes increment carry register initiated in steps else go to by FSRO.
8. Display result at PORT B.

Post Lab Assignments:

1. Explain Different subtraction instruction.
2. Explain NEGF instruction.

Lab Session 18

Title: TO WRITE A PROGRAM TO DISPLAY 0-F ON SEVEN SEGMENT DISPLAY

Objective: The objective is to make use of PIC18 and seven segment display.

Pre-Requisites:

1. Knowledge of MPLAB
2. Knowledge of PIC instruction.

Algorithm:

1. Initialise Port D as output Port.
2. Clear counting register for counting 0 to F.
3. Initialize registers count with value D'16'.
4. Get code of the value of counter from the table pointer and give it to Port D.
5. Decrement the counter. If it is Zero then go to step 6 otherwise step 4.
6. End

Post Lab Assignments:

1. Generate the delay of 19.328 ms

Lab Session 19

Title: TO WRITE A PROGRAM TO MULTIPLY TWO 16BIT NUMBERS

Objective: To make use of PIC.

Pre-Requisites:

1. Knowledge of MPLAB
2. Knowledge of PIC instruction

Algorithm:

1. Assume multiplier (M) & Multiplicand (N) stored at data memory location M1, M1+1, & N1, N1+1 respectively and store result at memory location PR, PR+1, PR+2, PR+3 .
2. Compute the partial product $M1_{L,}$ $N1_{L}$ & save it in location PR & PR+1.
3. Compute partial product $M1_{H}$ $N1_{H}$ and save it on location PR+1 & PR+ 2. The Carry Flag may set to 1 after this addition.
4. Compute partial product $M1_{H}$ $N1_{L}$ & add it to memory location PR+1 & PR+2.
5. Add carry flag to memory location PR+3
6. Compute partial product $M1_{L,}$ $N1_{H}$ & add it to memory location PR+1 & PR+2. Carry flag may be set to 1 after this addition.
7. Add Carry flag to memory location PR+3

Post Lab Assignments:

1. Explain different Rotate & skip & logical instructions

Lab Session 20

Title: TO WRITE A PROGRAM TO FIND NO. OF ELEMENT IN AN ARRAY OF 8-BIT ELEMENT THAT ARE MULTIPLE OF 4.

Objective: The objective is to make use of PIC18 logical and Branch instruction set.

Pre-Requisites:
1.Knowledge of MPLAB
2. Knowledge of PIC instruction.

Algorithm:

1. Start
2. Clear any one register from data memory to save result.(as counter)
3. Fsr0 pointer to data registers.
4. Get data Byte & increment pointer
5. If data byte =0 then go to step 7 otherwise step 6.
6. AND 03 with data if result 0, increment counter otherwise go to step 4.

Post Lab: 1. Explain all the instruction of Bit manipulation.

Lab Session 21

Title: Interfacing LCD to Port C & Port D

Objective: The objective is to make use of PIC18 with LCD instruction set.

Pre-Requisites: 1.Knowledge of MPLAB
2. Knowledge of PIC instruction.

Algorithm:

1. Initialize Port C & Port D as an output port.
2. Write command into the command register to initialize LCD like return home, clear display, function settings,
3. Give enough amount of delay.
4. Write data into the data register to display on the LCD.

Post Lab: 1. Write different command codes of LCD.