PUNE INSTITUTE OF COMPUTER TECHNOLOGY
DHANKAWADI, PUNE - 43

LIST OF LAB EXPERIMENTS

ACADEMIC YEAR: 2018-19

LAB MANUAL

DEPARTMENT: INFORMATION TECHNOLOGY

CLASS: S.E. **DATE – 11/12/2018**
SEMISTER- II

SUBJECT: PROCESSOR INTERFACE LABORATORY

INDEX OF LAB EXPERIMENTS

EXPT NO	PROBLEM STATEMENT	
1.	Write Assembly Language Program (ALP) to add array of N numbers stored in the memory.	
2.	Write menu driven ALP to convert 4-digit Hex number into its equivalent BCD number and 5-digit BCD number into its equivalent HEX number. Make your program user friendly to accept the choice from user for i) HEX to BCD ii) BCD to HEX iii) EXIT. Display proper strings to prompt the user while accepting the input and displaying the result. Write near procedure to complete the task.	
3.	Write ALP to perform following operations on string: i. Find and display length ii. Display reverse iii. Check whether string is palindrome or not. Display proper strings to prompt the user while accepting the input and displaying the result. Write near procedures to complete the task.	
4.	Write menu driven ALP to perform string manipulations. The strings to be accepted from the user is to be stored in code segment Module_1 and write FAR PROCEDURES in code segment Module_2 to perform any two of the following string operations:	

	i) Concatenation of two strings ii) Comparison of two strings iii) Finding Number of occurrences of a sub-string in the given string iv) Finding number of alphabets, digits, special characters, lower & upper case alphabets, word and number of lines from the text. Note: Use PUBLIC and EXTERN directives. Create .OBJ files of both the modules and link them to create an .EXE file.	
5.	Write menu driven program in C using int86(), int86x(), intdos() and intdosx() functions for implementing following operations on file. i. To delete a file ii. To create a directory iii. To copy a file	
6.	Write 8051 ALP to add n, 8 bits numbers found in internal ram location 40H onwards and store results in R6 and R7.	
7.	Write 8051 ALP for the block transfer for internal / external memory.	
8.	Serial port programming: ISR based Connect two 8051 microcontrollers using serial ports. Send FFh and 00h alternatively to receiver. Output received byte to port1, see port pin waveform on CRO.	
9.	Write ALP to interface 8051 with: LCD to display message.	
10.	Write ALP to interface 8051 with: Stepper motor to rotate motor with different step angles and speeds.	

Subject Co-coordinator **Head of Department**

(Mr. S. R. Warhade) **(Prof. (Dr.) B. A. Sonkamble)**

EXPERIMENT NO.1

Title:

Add array of N numbers stored in the memory.

Problem Definition: Write Assembly Language Program (ALP) to add array of N numbers stored in the memory.
Display proper messages to guide user while accepting number and displaying result.

Aim: Study of code, data, stack segment initialization, add, loop, int instructions and termination of program using DOS function.

Software and Hardware Requirements: TASM, TD, TLink, PC

ALGORITHM:

1. Start.
2. Read counter as a number of array elements from user
3. Initialize counter .
4. Initialize array pointer.
5. Sum = 0.
6. Get the array element pointed by array pointer.
7. Add array element in the Sum.
8. Increment array pointer decrement counter.
9. Repeat steps 4, 5 & 6 until counter = 0.
10. Display Sum.
11. Stop.

Result :

How many numbers do you want to add in array ?

_

Enter numbers :

- , - , - , ----------

Sum of Array numbers is : ----

References:

1. Peter Abel, Niyaz Nizamuddin, "IBM PC Assembly Language and Programming", Pearson Education

2. “Advanced MSDOS Programming” by Ray Duncan
3. “8088 and 8086 Processors” by Triebel Singh

Questions :

1) Explain types of Models in detail.
2) Explain the directives for initialization of stack, data, code segments.

Subject Co-coordinator **Head of Department**

(Mr. S. R. Warhade) **(Prof. (Dr.) B. A. Sonkamble)**

EXPERIMENT NO.2

Title: Conversion of Hex number to equivalent BCD and BCD number to Hex.

Problem Definition: Write an ALP to convert 4-digit hex number into its equivalent BCD number and 5 digits BCD number into its equivalent hex number. Make your program user friendly by asking the choice from the user for

0. Hex- BCD
1. BCD-Hex
2. Exit

Display proper messages to guide user while accepting number and displaying result.

Aim: Study of MUL, DIV instructions and procedures.

Software and Hardware Requirements: TASM, TD, TLink, PC

Algorithm:

1. Display menu and accept choice from user
2. If user enters 1, call hex-BCD procedure
3. If user enters 2,call BCD-hex procedure
4. If user enters any other choice go to last step.
5. Procedure to convert Hex number to its BCD equivalent-
    - Initialize ‘number’ and digit in data segment to zero.
    - Accept 4 digit no. and store it in BX.
    - Move number in data segment and in AX.
    - Set DX register to zero and move 000AH in BX.
    - Divide AX by BX
    - Store quotient in data segment.
    - Push remainder DX on stack and increment dig1. i.e. counter in data segment.
    - Repeat all the steps until quotient becomes zero.
    - Pop the contents from stack into DX, AND the data with 0FH to get digit in dl.
    - Add 30H to dl and display the digit using 02H function of INT 21H.
    - Decrement count of element and if not zero go to step one else stop.
    - Return.

6. Procedure for converting BCD to Hex.
    - Initialize ‘number’ in data segment to zero.
    - Move count (05) in ‘cnt’ variable in data segment

- Move 2710H in multiplier and even in CX.
- Take one digit input using 01H function of INT 21H.
- Multiply AL by CX and add it to 'number' in data segment.
- Divide multiplier by 000AH and decrement count.
- If count is not equal to zero go to step 4 else proceed.
- Move number to register BX
- Set CX to 0404H.
- Rotate BX 4 times to left to get MSB in LSB.
- And BL with 0FH to get single digit. Move it to DL, convert it to its ASCII value.
- Use 02H function of INT 21H to display the digit.
- Decrement CH and if not zero go to rotate step, else return

Result: Choose from following options:

1. Convert hex to BCD
2. Convert BCD to hex
3. Exit

Option selected: 1
Enter the hex number- FFFF
BCD equivalent number is – 65535 (

Continue (Y/N): Y

Choose from following options:

1. Convert hex to BCD
2. Convert BCD to hex
3. Exit

Option selected: 2
Enter the 5-digit BCD number- 65535
Hex equivalent number is – FFFF

Continue (Y/N): N

Conclusion: The program for conversion of 4 digit hex number to BCD equivalent of conversion of 5 digit BCD number to Hex equivalent has been performed successfully.

References: 1. **"**Microprocessor and interfacing Second edition" by Douglas Hall
2. "Advanced MSDOS Programming" by Ray Duncan
3. "8088 and 8086 Processors" by Triebel Singh

Questions:

1. List and Explain stack related instructions.
2. Explain ENDP, PROC directives.
3. Explain RET, PUSH, POP, CALL, DIV, MUL instructions in detail.

Subject Co-coordinator **Head of Department**

(Mr. S. R. Warhade) **(Prof. (Dr.) B. A. Sonkamble)**

EXPERIMENT NO. 3

Title: Operations on string

Problem Definition: Write an ALP for following operations on string entered by user.

1. Calculate length of string.
2. Reverse the string.
3. Check whether the string is palindrome or not.

Make the program user friendly by providing menu like-

1. Enter the string
2. Calculate length of string
3. Reverse the string
4. Check palindrome
5. Exit.

Display appropriate messages.

Aim: Study of near procedure and string instructions.

Software and Hardware required: TASM, TD, TLink, and PC

Algorithm:

1. Declare data segment with appropriate data members.
2. Initialize data and stack segment
3. Using macro display the menu
    1. Enter the string
    2. Calculate length of string
    3. Reverse the string
    4. Check palindrome
    5. Exit.
4. Accept option from user using INT21H function 01H.
5. If option is 1 go to step 7, option 2- go to step 8, option 3- go to 9, option 4- go to step 10, option 5-goto step 11
6. Display ‘enter a valid option’ and go to step 3.
    1. Call accepts procedure.
    2. Call length procedure
    3. Call reverse procedure
    4. Call palindrome procedure
    5. Stop.

Procedures-

1. Accept procedure- accept string using INT 21H function 0AH and return to main program.
2. Length procedure- Take count from 1st location of string, display the length of string using INT 21H function 02H.Return to main program.
3. Reverse Procedure- Traverse till the last char. Of the string so that SI will point to last location. Move contents of SI to DL register. Display character on screen by using 02H. Move the contents in another array pointed by DI. Increment SI, decrement DI, decrement counter. Repeat steps till count is zero. Return to main program.
4. Palindrome procedure: Call the reverse proc and store the reversed string in a new string. Compare two strings char. By char. If both are same display msg 'String is palindrome' else display msg.'string is not palindrome'. Return to main program.

Result: Choose from following options-

Enter the string.
Calculate length of string
Reverse the string
Check palindrome
Exit.

Option selected: 1
Enter the string: Malayalam
Option selected: 2
Length of string: 9

Conclusion: The program to perform string operations like accept, length, reverse, palindrome on a string using near procedures.

References: 1. **"**Microprocessor and interfacing Second edition" by Douglas Hall
2. "Advanced MSDOS Programming" by Ray Duncan
3. "8088 and 8086 Processors" by Triebel Singh

Questions:

1. To compare the two strings which instructions we use?
2. Explain LODS, STOS instructions
3. Explain 0AH function of INT21H.
4. What is procedure?
5. List the differences between near and Far procedure
6. Explain CLD and STD instructions?
7. What do you mean by prefix?
8. Explain the use of REP, REPE, REPZ, and REPNZ
9. Which instruction set we can write instead of LOOP?

Subject Co-coordinator **Head of Department**

(Mr. S. R. Warhade) **(Prof. (Dr.) B. A. Sonkamble)**

EXPERIMENT NO. 4

Problem Definition: Write an ALP to perform string manipulations. The string to be accepted from the user is to be stored in code segment module-1 and write FAR procedure in code segment module-2 for following operations on the strings-

- Concatenation of two strings
- Compare the strings
- Number of occurrences of sub strings in given string.

Aim: Study of FAR procedures

Software and Hardware requirements: TASM, TD, TLink, PC

Algorithm:

1. Declare data segment with appropriate data members.
2. Initialize stack segment and data segment.
3. Using macro, display the menu.-
    - Concatenate two strings
    - Compare two strings
    - Find number of occurrences of substrings in a string.
    - Exit
4. Accept option from user using 01H function of INT 21H.
5. If option=1, call concatenate procedure then return to step 3.
6. If option=2, call compare procedure then return to step3.
7. If option=3, call sub procedure then return to step3.
8. If option=4, go to step 9.
9. Display 'enter valid option!' and go to step3
10. Stop.

Procedures:

1. Concatenate procedure-
    - Accept string1 from user using 0AH function of INT 21H.
    - Accept string2 from user using 0AH function of INT 21H.
    - Load length of first string in CX.
    - Load the address of source string1 in SI and DI.
    - Traverse first string till the last character.
    - Get string2 count in cl and append string2 to string1.
    - Display the concatenated string.
    - Return to the main program.

Compare procedure-

- Accept string1 using 0AH function of INT 21H.
- Accept string2 using 0AH function of INT 21H.
- Load length of string1 from string1+1 location to cl.
- Load length of string2 from string2+1 location to ch.
- Compare lengths of strings, if not equal display 'strings not equal'
- Load number of characters to be compared in CX.
- Compare two strings char by char. If not equal, display 'strings not equal'.
- Display 'strings are equal' and return to the main program.
- Return to the main program.

2. Substring procedure-
    - Initialize data segment, declare buffer to store strings.
    - Move count of main string in cl and count of substring in dl.
    - Compare substring with main string till there is match, else return.
    - Compare all other substrings with main string.
    - If they match ,increment counter containing number of occurrences
    - Repeat comparison for remaining length of main string.(length of main string-length of substring+1)
    - Display counter value which indicates number of occurrences of substring.
    - Return to the main program.

Result:

Enter first string: Good

Enter second string: morning
Menu: 1.Concatenate 2. Compare 3. Substring Occurrences

Enter a choice: 1
Concatenated string is: Good Morning

Enter first string: Good

Enter second string: Good
Menu: 1.Concatenate 2. Compare 3. Substring occurrences
Enter a choice: 2

THE STRINGS ARE EQUAL.

Enter first string: abcdabc Enter second string: abc

Menu: 1.Concatenate 2.Compare 3.Substring occurrences

Enter a choice: 3

Number of occurrences: 2

Conclusion: The string manipulation operations are successfully implemented using FAR procedure

References: 1. **"**Microprocessor and interfacing Second edition" by Douglas Hall
2. "Advanced MSDOS Programming" by Ray Duncan
3. "8088 and 8086 Processors" by Triebel Singh

Questions:

1. Explain all directives related to Procedure
2. Explain SCAS instruction.
3. Explain following directives
   a) SEGMENT b) END c) ENDS
4. Explain concept of LINKER
5. Explian all directives related to Procedure

Subject Co-coordinator **Head of Department**

(Mr. S. R. Warhade) **(Prof. (Dr.) B. A. Sonkamble)**

EXPERIMENT NO. 5

Title: C programming using int86( ), int86x( ), intdos( ), intdosx( ) functions.

Problem Definition: Write a program in C using int86( ), int86x( ), intdos( ), intdosx( ) functions to delete a file and create a directory.

Aim: To study ROM/BIOS functions and use of int86( ), int86x( ), intdos( ), intdosx( ) functions.

Hardware and Software requirements: IBM PC, Turbo C.

Algorithm:

Delete file

1. Start
2. Declare all the libraries like ‘stdio.h’, ‘conio.h’, ‘dos.h’.
3. Get the file name that is to be deleted.
4. Call function 41h of INT 21h. To call this function, intdos( ) function should be invoked.
5. Check the carry flag to check whether the function is successful.
6. Display appropriate messages.
7. Stop.

Create directory

1. Start.
2. Declare all the libraries like ‘stdio.h’, ‘conio.h’, ‘dos.h’.
3. Enter the name of the directory that is to be created.
4. Call function 39h of INT 21h. To call this function, intdosx( ) function should be invoked.
5. Check the carry flag to check whether the function is successful.
6. Display appropriate messages.
7. Stop.

Conclusion: We have successfully used intdos( ) and intdosx( ) functions in C programming.

Result: 1. “File successfully deleted”
2. “ Directory successfully created.”

References: 1. “TSR programming through C” - Yashvant Kanitkar

Questions:

1. Which are the ways to interact with the hardware through C?
2. Compare all the ways to interact with the hardware.
3. What are ROM/BIOS functions?
4. Explain int86( ), int86x( ), intdos( ), intdosx( ) functions.
5. What is the difference between int86( ) and int86x( ) functions?
6. Explain with details the following functions:
    - 41 h of INT 21h
    - 39h of INT 21h
    - 02h of INT 13h
7. How the DOS or ROM/BIOS functions are called through C?

Subject Co-coordinator **Head of Department**

(Mr. S. R. Warhade) **(Prof. (Dr.) B. A. Sonkamble)**

EXPERIMENT NO.6

Title: 8051 programming.

Problem Definition: Write a 8051 Program to
a) Add ‘n’ 8-bit nos. found in internal RAM location 40h onwards and store result in R6 and R7 register

Aim: To understand concept of register bank, bit addressable area, internal memory organization, various addressing modes,

Hardware and Software requirements: IBM PC, Keil software

Algorithm:
1. Start
2. Load count value in register.
3. Point Ro to internal memory location
4. Clear Accumulator
5. Add accumulator with memory contents pointed by R0
6. If carry is not generated go to step8
7. Increment R6
8. Increment R0
9. Decrement counter
10. If counter is not zero go to step 5
11. Store result of final addition in register R7
12. Stop.

Conclusion:
We have studied to access internal memory.

Result:
Internal memory
I: 0x40 01 02 03 04
After addition
R6=0 R7=0A

Questions

1. List the features of 8051.
2. Explian various addressing modes of 8051 with example.
3. Explain internal memory organization of 8051.
4. Why 8051 program start at 0030h location onwards?
5. 8051 works on what frequency?
6. How to switch from one bank to other in 8051?
7. What is PSW?

Subject Co-coordinator **Head of Department**

(Mr. S. R. Warhade) **(Prof. (Dr.) B. A. Sonkamble)**

EXPERIMENT NO. 7

Problem Definition: Write an 8051 Program to
Transfer the block from internal memory location 30h to internal memory location 40h.

Aim: To understand concept of internal memory organization, various addressing modes and external memory accessing.

Hardware and Software requirements: IBM PC, Keil software

Algorithm:
1. Start
2. Load count value in register.
3. Point Ro to internal memory location 30h
4. Point R1 to internal memory location 40h
5. Copy contents of memory pointed by R0 to Accumulator
6. Copy contents of Accumulator to memory pointed by R1
8. Increment R0
9. Increment R1
10. Decrement counter
11. If counter is not zero go to step 3
12. Stop.

Conclusion:
We have studied to access internal memory using indirect addressing mode.

Result:
Internal memory
I: 0x30 01 02 03 04 05
I: 0x40 00 00 00 00 00
After block transfer
I: 0x40 01 02 03 04 05

References: 1.The 8051 Microcontroller and Embedded Systems-Mazidi
2. The 8051 Microcontroller –K. Ayala

Questions

1. List the features of 8051.
2. Explain various addressing modes of 8051 with example.
3. Explain internal memory organization of 8051.
4. Why 8051 program start at 0030h location onwards?
5. 8051 works on what frequency?
6. How to switch from one bank to other in 8051?
7. What is PSW?

Subject Co-coordinator **Head of Department**

(Mr. S. R. Warhade) **(Prof. (Dr.) B. A. Sonkamble)**

EXPERIMENT NO. 8

Title: 8051 serial port interface

Problem Definition: Serial port programming: ISR based Connect two 8051 microcontrollers using serial ports. Send FFh and 00h alternatively to receiver.

Algorithm:

1) The TMOD register is loaded with the value 20H, indicating the use of Timer 1 in mode 2 (8-bit auto-reload) to set the baud rate.
2) The TH1 is loaded with one of the values in Table 10-4 to set the baud rate for serial data transfer (assuming XTAL = 11.0592 MHz).
3) The SCON register is loaded with the value 50H, indicating serial mode 1,

   where an 8-bit data is framed with start and stop bits.

   i) TR1 is set to 1 to start Timer 1.
   ii) TI is cleared by the "CLR TI" instruction.
   iii)The character byte to be transferred serially is written into the SBUF register.
   a)The TI flag bit is monitored with the use of the instruction " JNB TI, xx" to see if the character has been transferred completely.

4) **To transfer the next character, go to Step 5.**

   To understand the importance of the role of TI, look at the following sequence of steps that the 8051 goes through in transmitting a character via TxD.

   i) The byte character to be transmitted is written into the SBUF register.

   ii) The start bit is transferred.

   iii) The 8-bit character is transferred one bit at a time.

   In the programming of the 8051 to receive character bytes serially, the following steps must be taken.

   i) The TMOD register is loaded with the value 20H, indicating the use of Timer
   1 in mode 2 (8-bit auto-reload) to set the baud rate.

   ii) TH1 is loaded with one of the values in Table 10-4 to set the baud rate (assum
   ing XTAL = 11.0592MHz).

iii) The SCON register is loaded with the value 50H, indicating serial mode 1,
where 8-bit data is framed with start and stop bits and receive enable is turned
on.

1) TR1 is set to 1 to start Timer 1.
2) RI is cleared with the "CLR RI" instruction.
   i) The RI flag bit is monitored with the use of the instruction "JNB RI, xx" to
   see if an entire character has been received yet.
3) When RI is raised, SBUF has the byte. Its contents are moved into a safe place.
4) To receive the next character, go to Step 5.

Conclusion:

We have studied how to interface 8051 microcontroller with RS232 standard serial interface.

References: 1.The 8051 Microcontroller and Embedded Systems-Mazidi
2. The 8051 Microcontroller –K. Ayala

Questions :

1) What is mean by RS232 standard ?
2) Explain USART ?
3) What is MAX232 ?

Subject Co-coordinator **Head of Department**

(Mr. S. R. Warhade) **(Prof. (Dr.) B. A. Sonkamble)**

EXPERIMENT NO. 9

Title: 8051 LCD interface
Problem Definition: **Write ALP to interface 8051 with: LCD to display message.**

Algorithm:

1)LCD initialization

The steps that has to be done for initializing the LCD display is given below and these steps are common for almost all applications.

i) Send 38H to the 8 bit data line for initialization

ii) Send 0FH for making LCD ON, cursor ON and cursor blinking ON.

iii) Send 06H for incrementing cursor position.

iv) Send 01H for clearing the display and return the cursor.

2) Sending data to the LCD.

The steps for sending data to the LCD module is given below. I have already said that the LCD module has pins namely RS, R/W and E. It is the logic state of these pins that make the module to determine whether a given data input is a command or data to be displayed.

i) Make R/W low.

ii) Make RS=0 if data byte is a command and make RS=1 if the data byte is a data to be displayed.

iii) Place data byte on the data register.

iv) Pulse E from high to low.

v) Repeat above steps for sending another data.

Conclusion:

We have studied how to interface 8051 microcontroller with 16x2 LCD .

References: 1.The 8051 Microcontroller and Embedded Systems-Mazidi
2. The 8051 Microcontroller –K. Ayala

Questions :

1) How many types of LCD?
2) What is mean by LCD commands?
3) How LCD initialized?

Subject Co-coordinator **Head of Department**

(Mr. S. R. Warhade) **(Prof. (Dr.) B. A. Sonkamble)**

EXPERIMENT NO. 10

Title: 8051 stepper motor interface
Problem Definition: Write ALP to interface 8051 with: Stepper motor to rotate motor with different step angles and speeds.

Algorithm:
The following steps show the 8051 connection to the stepper motor and its programming.

1. Use an ohmmeter to measure the resistance of the leads. This should identify which
   COM leads are connected to which winding leads.
2. The common wire(s) are connected to the positive side of the motor's power supply.
   In many motors, +5 V is sufficient.
3. The four leads of the stator winding are controlled by four bits of the 8051 port (e.g Pl.O
   - P1.3). However, since the 8051 lacks sufficient current to drive the stepper motor
   windings, we must use a driver such as the ULN2003 to energize the stator.

One and two electromagnets are alternatively energized at a single time and in coding:

- First pin is high
- Then first and second both get high
- Second alone goes high
- Second along with third goes high
- In same way, alternatively one and two pins go high

Conclusion:
We have studied how to interface 8051 microcontroller with stepper motor.

References: 1.The 8051 Microcontroller and Embedded Systems-Mazidi
2. The 8051 Microcontroller –K. Ayala

Questions:

1) How many types of stepper motor ?
2) What is mean by step angle ?
3) Why there is need of driver ULN2003 ?

Subject Co-coordinator **Head of Department**

(Mr. S. R. Warhade) **(Prof. (Dr.) B. A. Sonkamble)**