CSE 214: MICROPROCESSORS LAB

IV SEM B.Tech (CSE)

(2014)

Prepared by	Approved by
Dr. N.Gopalakrishna Kini	H.O.D

DEPT. OF COMPUTER SCIENCE & ENGG.

Manipal Institute of Technology., MANIPAL

INSTRUCTIONS TO STUDENTS

1. Students should be regular and come prepared for the lab practice.

2. In case a student misses a class, it is his/her responsibility to complete that missed experiment(s).

3. Students should bring the observation book and lab journal. Prescribed textbook and class notes can be kept ready for reference if required.

4. Once the program(s) get executed, they should show the results to the instructors and copy the same in their observation book.

5. Their observation book as well as the lab journal should be complete with any other information related to the program they execute.

6. Questions for lab tests and exam need not necessarily be limited to the questions in the lab manual, but could involve some variations and / or combinations of the questions.

CONTENTS

Module I: Assembly language programming (ALP)

Module II: Interface programming

Module-I : ASSEMBLY LANGUAGE PROGRAMMING

WEEK 1: Introduction to MASM & simple programs

Tutorial: 8086 Architecture to be explained briefly. Detailed description of the MASM toolkit is to be given in lab. Simple programs are to be given for the familiarization of the masm.

1. Addition of 2 8-bit numbers.
2. Addition of 2 16-bit numbers.
3. Subtraction of 2 8-bit numbers.
4. Subtraction of 2 16-bit numbers.

WEEK 2: Arithmetic, Branching Instructions

1. Multiply two 8-bit numbers.
2. Multiply a 32-bit number by a 16-bit number
3. Multiply two 32-bit numbers.
4. Divide a 16-bit number by an 8-bit number.
5. Divide a 32-bit number by a 16 bit number.

WEEK 3: Arithmetic and branching instructions (Contd…)

Tutorial: Explanation about arithmetic (normal and BCD arithmetic), logical and branching instructions.

1. Find the sum of 10 numbers in an array.
2. Find GCD and LCM of two numbers.
3. Generate Fibonacci series (up to N terms).
4. Unpack a 2 byte packed BCD number and represent in ASCII format.
5. Convert the given 4 byte unpacked BCD in ASCII format to packed BCD format.

WEEK 4: List Operations (Arrays)

1. Sort an array in ascending order.
2. Insert 1 or more elements into a sorted list. The number to be inserted and the elements can be assumed to be present in locations.
3. Delete 1 or more elements from a sorted list. The number to be deleted and the elements can be assumed to be present in locations.
4. Insert 1 or more elements in an unsorted list at any given position.
5. Delete 1 or more elements from an unsorted list at any given position.
6. Generate prime numbers between two numbers N1 and N2.

WEEK 5: String Operations

Tutorial: Explanation about String primitives.

Assuming each string is terminated by '$', write 8086 programs to:

1. Find the length of the given string.
2. Check if the given string is a palindrome or not.
3. Concatenation of two strings.
4. Reverse the given string.
5. Find the substring in the main string.

WEEK 6: String instructions (Use procedures and macros wherever required), DOS Interrupts

1. Read a string from the keyboard and count the number of words in the string, where the words (group of characters) may be separated by 1 or more spaces.
2. Read a string from the keyboard and reverse the case of each letter and display it on the screen.
3. Write a recursive procedure to calculate factorial of a number.
4. Create a file and write into it.
5. Read a file and display its contents on the screen.
6. Accept a string from KB and append it to an existing file

MODULE-II : INTERFACE PROGRAMMING

WEEK 7: Logic Controller Interfacing

Tutorial: Logic Controller Interfacing.

1. Develop and execute an assembly language program to read the status of eight inputs from the logic controller interface, to complement those values and to display these complemented values using the same interface.
2. Develop and execute an assembly language program to read the status of eight inputs from the logic controller interface and to display FF if any input is high and to display 00 otherwise.

WEEK 8: Display Interfacing

Tutorial: Interfacing 7 Segment Display.

1. Develop and execute an assembly language program to display a 4 digit BCD number on the display interface.
2. Display messages FIRE and HELP alternately with flickering effects on a 7- segment display interface for a suitable period of time. Ensure a flashing rate that makes it easy to read both the messages (Examiner does not specify these delay values nor it is necessary for the student to compute these values).

WEEK 9: Keyboard Interfacing

1. Develop and execute an assembly language program to a keypad interface for key closure. The row number and column number of the key pressed are to be stored in memory locations.
2. Develop and execute an assembly language program to scan a keypad and to store the code of the key pressed in a memory location.

WEEK 10: Elevator Interfacing

1. Simulate the elevator motion by interfacing the elevator kit .

    i. Initially the elevator should be in the ground floor, with all requests in OFF state.

    ii. When a request is made from a floor, the elevator should move to that floor, wait there for a couple of seconds, and then come down to ground floor and stop. If some requests occur during going up or coming down they should be ignored.

WEEK 11: DAC Interfacing

1. Develop and execute an assembly language program to generate a sine wave using the DAC interface. (The output of the DAC is to be displayed on the CRO. The ON/OFF times can be any arbitrary values. No need to compute these values.)
2. Develop and execute an assembly language program to generate a triangular waveform using a DAC interface. (The output of the DAC is to be displayed on a CRO. The slope can be any arbitrary value. No need to compute this)

WEEK 12: Stepper Motor Interfacing

1. Write an assembly language program to rotate the stepper motor by a fixed angle in a fixed interval time.
2. Drive a Stepper Motor interface to rotate the motor in anti-clockwise direction by N steps (N is specified by the examiner). Introduce suitable delay between successive steps. (Any arbitrary value for the delay may be assumed by the student).

WEEK 13

LAB EXAM

Additional Programs:

1. Multiply two signed numbers.
2. Sort the elements of an array of bytes (signed and unsigned).
3. Add the elements of 2 arrays , each element being 32 bit numbers. Assume no overflow.
4. Reverse the case of each character(lower to upper and vice versa).
5. Accept the password from keyboard and check for its validity.
6. Read a string from the keyboard and count the number of vowels in it.
7. Display a digital clock on screen by reading the real time clock.
8. Read the status of 3 inputs from the logic controller interface. Assuming that these three inputs from the logic controller interface. Assuming that these three inputs represented a binary number X (0«X«7), Display 2*X using the same interface.
9. Develop and execute an assembly language program to convert a 16 bit binary value (assumed to be unsigned integer) to BCD and to display it on the display interface.
10. Develop and execute an assembly language program to generate a rectangular pulse using the DAC interface.

References:

1. Douglas V.Hall, “Microprocessors and Interfacing”, Tata McGraw Hill Publications Ltd., Revised Second Edition 2006.
2. Respective user manual for interfacing kit.