

MICROPROCESSOR AND MICROCONTROLLER LABORATORY

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2016 -2017)

SEMESTER – IV

Subject Code	15CSL48	IA Marks	20
Number of Lecture Hours/Week	01 I + 02 P	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

CREDITS – 02

Course objectives: This course will enable students to

- To provide practical exposure to the students on microprocessors, design and coding knowledge on 80x86 family/ARM. To give the knowledge and practical exposure on connectivity and execute of interfacing devices with 8086/ARM kit like LED displays, Keyboards, DAC/ADC, and various other devices.

Description

Demonstration and Explanation hardware components and Faculty in-charge should explain 8086 architecture, pin diagram in one slot. The second slot, the Faculty in-charge should explain instruction set types/category etc. Students have to prepare a write-up on the same and include it in the Lab record and to be evaluated.

Laboratory Session-1: Write-up on Microprocessors, 8086 Functional block diagram, Pin diagram and description. The same information is also taught in theory class; this helps the students to understand better.

Laboratory Session-2: Write-up on Instruction group, Timing diagrams, etc. The same information is also taught in theory class; this helps the students to understand better.

Note: These TWO Laboratory sessions are used to fill the gap between theory classes and practical sessions. Both sessions are evaluated as lab experiments for 20 marks.

Experiments

- Develop and execute the following programs using 8086 Assembly Language. Any suitable assembler like MASM/TASM/8086 kit or any equivalent software may be used.
- Program should have suitable comments.
- The board layout and the circuit diagram of the interface are to be provided to the student during the examination.
- Software Required: Open source ARM Development platform, KEIL IDE and Proteus for simulation

SOFTWARE PROGRAMS: PART A

1. Design and develop an assembly language program to search a key element “X” in a list of ‘n’ 16-bit numbers. Adopt Binary search algorithm in your program for searching.
2. Design and develop an assembly program to sort a given set of ‘n’ 16-bit numbers in ascending order. Adopt Bubble Sort algorithm to sort given elements.
3. Develop an assembly language program to reverse a given string and verify whether it is a palindrome or not. Display the appropriate message.
4. Develop an assembly language program to compute nCr using recursive procedure. Assume that ‘n’ and ‘r’ are non-negative integers.

<p>5. Design and develop an assembly language program to read the current time and Date from the system and display it in the standard format on the screen.</p> <p>6. To write and simulate ARM assembly language programs for data transfer, arithmetic and logical operations (Demonstrate with the help of a suitable program).</p> <p>7. To write and simulate C Programs for ARM microprocessor using KEIL (Demonstrate with the help of a suitable program)</p> <p>Note : To use KEIL one may refer the book: Insider's Guide to the ARM7 based microcontrollers, Hitex Ltd.,1st edition, 2005</p>
HARDWARE PROGRAMS: PART B
<p>8. a. Design and develop an assembly program to demonstrate BCD Up-Down Counter (00-99) on the Logic Controller Interface.</p> <p>b. Design and develop an assembly program to read the status of two 8-bit inputs (X & Y) from the Logic Controller Interface and display X*Y.</p> <p>9. Design and develop an assembly program to 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 is it necessary for the student to compute these values).</p> <p>10. Design and develop an assembly program to drive a Stepper Motor interface and rotate the motor in specified direction (clockwise or counter-clockwise) by N steps (Direction and N are specified by the examiner). Introduce suitable delay between successive steps. (Any arbitrary value for the delay may be assumed by the student).</p> <p>11. Design and develop an assembly language program to</p> <p>a. Generate the Sine Wave using DAC interface (The output of the DAC is to be displayed on the CRO).</p> <p>b. Generate a Half Rectified Sine waveform using the DAC interface. (The output of the DAC is to be displayed on the CRO).</p> <p>12. To interface LCD with ARM processor-- ARM7TDMI/LPC2148. Write and execute programs in C language for displaying text messages and numbers on LCD</p> <p>13. To interface Stepper motor with ARM processor-- ARM7TDMI/LPC2148. Write a program to rotate stepper motor</p>
<p>Study Experiments:</p> <p>1. Interfacing of temperature sensor with ARM freedom board (or any other ARM microprocessor board) and display temperature on LCD</p> <p>2. To design ARM cortex based automatic number plate recognition system</p> <p>3. To design ARM based power saving system</p>
<p>Course Outcomes: After studying this course, students will be able to</p> <ul style="list-style-type: none"> • Learn 80x86 instruction sets and gains the knowledge of how assembly language works. • Design and implement programs written in 80x86 assembly language • Know functioning of hardware devices and interfacing them to x86 family • Choose processors for various kinds of applications.
<p>Graduate Attributes</p> <ul style="list-style-type: none"> • Engineering Knowledge • Problem Analysis • Modern Tool Usage • Conduct Investigations of Complex Problems • Design/Development of Solutions

Conduction of Practical Examination:

- All laboratory experiments (all 7 + 6 nos) are to be included for practical examination.
- Students are allowed to pick one experiment from each of the lot.
- Strictly follow the instructions as printed on the cover page of answer script for breakup of marks
- PART –A: Procedure + Conduction + Viva: 10 + 25 +05 (40)
- PART –B: Procedure + Conduction + Viva: 10 + 25 +05 (40)
- Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.