SAULT COLLEGE OF APPLIED ARTS & TECHNOLOGY

SAULT STE. MARIE, ONTARIO

COURSE OUTLINE

Course Title: MICROPROCESSOR PROGRAMMING

Code No.: CET127-4 Semester: 3

Program: COMPUTER ENGINEERING TECHNOLOGY /
ELECTRICAL/ELECTRONIC TECHNOLOGY

Author: MARK ALLEMANG

Date: SEPT., 1994 Previous Outline Dated: JAN.,1994

APPROVED: ______________________ 94-09-06

Dean Date

TOTAL CREDITS : 4

PREREQUISITES: CET105 or CET110

I. PHILOSOPHY/GOALS:

This course introduces students to 8088 Assembly Language Programming. As a first course in assembly language programming it is necessary to study number systems and the fundamentals of microprocessor architecture in addition to the specific registers, addressing modes and instructions of the 8088 microprocessor. More significant assembly language programming skills will be developed in subsequent courses.

Initially the concentration is on the development of simple instruction sequences using DEBUG, an interactive debugging utility that is available as part of DOS. Later, the use of an Assembler such as Microsoft's MASM or Borland's Turbo Assembler will be used to assemble source programs. In addition, the Linker, and Debugging utilities are used to link and test more complex programs.

II. STUDENT PERFORMANCE OBJECTIVES:

Upon successful completion of this course the student will:

1. Be able to manipulate and convert numbers in binary, hexadecimal, decimal and octal number systems.

2. Describe the architecture of the 8088 microprocessor including its registers, memory organization, the stack and internal organization.

3. Be able to use the 8088 addressing modes appropriately.

4. Demonstrate the use of DEBUG to assemble, load, save, unassemble, single step, examine and troubleshoot simple instruction sequences.

5. Be able to write assembly language programs utilizing the subset of commonly-used 8088 instructions to solve simple problems.

6. Demonstrate proficiency in developing and troubleshooting assembly language programs utilizing an Assembler.

III. TOPICS TO BE COVERED:

1. Number Systems: Binary, Octal, Hexadecimal and Decimal.

2. 8088 Microprocessor Organization and Registers.

3. 8088 Addressing Modes.

4. The use of DEBUG to assemble and troubleshoot simple programs.

5. Instruction Set of the 8088 Microprocessor.

6. Using an Assembler as a Program Development Tool.

7. Writing Assembly Language programs to solve problems.

IV. LEARNING ACTIVITIES/REQUIRED RESOURCES

Note: The following Learning Activities may not be covered in the exact order specified below.

1. Number Systems: Binary, Octal, Hexadecimal and Decimal.

Learning Activities

- listen to a presentation on number systems bases including binary, octal, hexadecimal and decimal and how to convert from one to another
- complete exercises converting between the various bases
- listen to a presentation on using two's complement to represent negative numbers
- listen to a presentation on the nature of BCD and ASCII codes
- listen to a presentation on parity checking

Resources:
Text ch 2, overheads

==

2. 8088 Microprocessor Organization and Registers.

Learning Activities

- listen to a presentation on the architecture of a microcomputer system and introducing microprocessor architecture
- listen to a presentation on assembly language programming as to its nature, advantages and disadvantages

- listen to a presentation on the bus organization in typical microcomputer systems and the execution sequence of an instruction
- listen to a presentation on the 8088 memory address space and how the 8088 utilizes memory segmentation and the difference between physical and logical addresses
- listen to a presentation on the nature and functioning of the 8088 internal registers
- listen to a presentation on the operation of the stack in 8088 systems

Resources:
Text ch 1,4, overheads

==

3. 8088 Addressing Modes.

Learning Activities

- listen to a presentation on the various 8088 addressing modes
- write simple programs that make use of the various 8088 addressing modes

Resources:
Text ch 4, overheads

==

4. The use of DEBUG to assemble and troubleshoot simple programs.

Learning Activities

- listen to a presentation on the nature of machine code and the method of converting assembly language instructions to machine language
- listen to a presentation on DEBUG
- demonstrate the use of DEBUG on the PC to:
  - examine and modify the contents of memory and registers,
  - load and save machine code programs on disk,
  - assemble and unassemble instructions,
  - execute and single-step through programs and other tasks required in program debugging

Note: (A Practical Test will be given on the use of DEBUG).

Resources:
Text ch 3, overheads

==

5. Instruction Set of the 8088 Microprocessor.

Learning Activities

- listen to presentations on and be able to demonstrate the operation of the following groups of instructions and their proper utilization in programs:

- data transfer instructions
- arithmetic instructions
- logic instructions
- shift and rotate instructions
- flag-control instructions
- compare instructions
- jump and loop instructions
- subroutine handling
- string instructions

Resources:
Text ch 5, 8088 Instruction Set Summary, overheads

==

6. Using an Assembler as a Program Development Tool.

Learning Activities

- listen to a presentation on using TASM (Borland Turbo Assembler) including proper syntax, program initialization and pseudo operations (pseudo-ops)
- create assembler source files using an editor
- generate run modules using TASM and TLINK and then execute and troubleshoot assembler programs with Turbo Debug

Resources:
Text ch 6, overheads

==

7. Writing Assembly Language programs to solve problems.

Learning Activities

- listen to a presentation on the relationship between a program, DOS, BIOS and the computer hardware and how to use DOS and BIOS services to control the computer hardware
- write assembler programs to solve assigned problems using the instructions and addressing modes most appropriate to the task
- listen to a presentation on the use of the IN and OUT instructions and the I/O address space in 8088 systems
- demonstrate using IN and OUT instructions to directly control the computer I/O devices

Resources:
Text ch 8,9, overheads

==

V. METHOD OF EVALUATION:

3 THEORY TESTS (20% each)	60%
ASSIGNMENTS	25%
QUIZZES AND PRACTICAL TESTS	15%

(The percentages shown above may have to be adjusted to accurately evaluate student skills. Students will be notified of any changes made.)

GRADING SCHEME

A+	90	-	100%
A	80	-	89%
B	70	-	79%
C	55	-	69%
I	Incomplete		
R	Repeat		

TESTS

Written tests will be announced about one week in advance. Quizzes may be conducted without advance warning. No "re-write" opportunities exist for quizzes not written.

UPGRADING OF INCOMPLETES

When a student's course work is incomplete or final grade is below 55%, there is the possibility of upgrading to a pass when a student meets the following criteria:

1. The students attendance has been satisfactory.
2. An overall average of at least 40% has been achieved.
3. The student has not had a failing grade in all of the theory tests taken.
4. The student has made reasonable efforts to participate in class and complete assignments.

ASSIGNMENTS AND LAB ACTIVITIES:

Lab activities and assignments represent a very important component of this course in which practical 'hands-on' skills will be developed. Because of this, lab attendance is mandatory and the satisfactory completion of all assignments is required. It is the student's responsibility to discuss absences from regularly scheduled labs with the instructor so that alternate arrangements (where possible) can be made to complete the lab requirements.

A penalty for late assignments will be applied unless there are extenuating circumstances. A 10% per week penalty will be applied. After 4 weeks late assignments will not be accepted for credit.

It is acceptable that students consult with each other in relation to their assigned problems. However, it is unacceptable to copy programs written by someone else and submit them as your own work. Where plagiarism or copying is found and it is impossible to determine whose original work it is, a mark of zero will be assigned to all assignments involved.

ATTENDANCE:

Absenteeism will affect a student's ability to succeed in this course. Absences due to medical or other unavoidable circumstances should be discussed with the instructor, so that remedial activities can be scheduled. A Quiz or Test missed because of an unauthorized absence will result in a zero grade being assigned.

VI. REQUIRED STUDENT RESOURCES:

1. TEXTBOOK: **"Introduction to Programming in Assembly Language (IBM PC)"**
   by George Driver
   (West)

2. DISKETTES: 2 3-1/2" HD Diskettes.

3. 8088 INSTRUCTION SET SUMMARY.

VII. ADDITIONAL RESOURCE MATERIALS AVAILABLE:

VIII. SPECIAL NOTES:

Students with special needs (eg. physical limitations, visual or hearing impairments, or learning disabilities) are encouraged to discuss any required accommodations confidentially with the instructor.

Your instructor reserves the right to modify the course as deemed necessary to meet the needs of students or take advantage of new or different learning opportunities.