Diablo Valley College

COMSC-260: Assembly Language Programming/Computer Organization

Course Description

A course covering the basics of machine architecture, machine language, assembly language, operating system interface, and interfacing with high level languages. Topics include data representation, instruction representation and execution, addressing, indexing, macros, subroutine linkages, storage and time efficiency issues, interrupt handling, virtual memory, cache memory, and dynamic address translation.

Approved On

Aug 2 2005

Course Notes

None

General Information

Department:	Computer Science
Division:	Math and Computer Science
Units:	4.00 to 4.00
Max Day Class Size:	35
Max Night Class Size:	35
Materials Fee:	$0.00
Grade Code:	Student choice
Repeatability	0

Number of Hours

Per Semester

Lecture:	54.00 to 54.00
Laboratory:	54.00 to 54.00
Activity:	0.00 to 0.00

By Arrangement

Lecture:	0.00 to 0.00
Laboratory:	0.00 to 0.00
Activity:	0.00 to 0.00

Discipline Eligible for Teaching this Course

Master Degree Disciplines

Computer Science

Non-Master Degree Disciplines

Additional Information

SAM code: B

TOP code: 0706.00
CIP code: 11.0701
CAN code:
Non-Degree Applicable:
Library: Yes
Media: Yes

Approved for

- ✓ CSU Transferable
- CSU Lower Division Transfer Pattern
- ✓ UC Transferable
- DVC General Education
  - False
- CSU General Education
  - False
- IGETC

- ✓ AA/AS Degree Requirement or Elective
- Certificate of Accomplishment (<18 Units)
- ✓ Certificate of Achievement (>=18 Units)

Objectives

The following measurable course objectives are the basis for student learning outcomes. The students will be able to:

A. Describe computer hardware architecture
B. Use all types of data representation
C. Access and use the operating system
D. Program in assembly language
E. Handle hardware interrupts
F. Interface with high level languages
G. Perform simple hardware simulation

Content

A. Introduction
   1. Purpose of assembly language
   2. Machine language
B. Data representation
   1. Character representation
   2. Integer representation—unsigned and signed
   3. Floating point representation
   4. Other numeric representation
C. Hardware overview
   1. Summary of microprocessors
   2. System bus
   3. Channels, controllers and interfaces
   4. Direct Memory Access
   5. Pipelining
   6. Cache memory
   7. Real memory vs. virtual memory

8. Interrupts and the interrupt vectors

D. Elements of Assembly Language
   1. From source program to execution
   2. Assembler directives

E. I/O through the operating system

F. Machine Instructions
   1. Addressing
   2. Arithmetic instructions
   3. Conditional and unconditional transfer
   4. Stack operations
   5. Logical instructions
   6. String operations
   7. Floating point instructions

G. Subprogram calling
   1. Calling conventions
   2. Creating library of reusable functions

H. Macros
   1. Parameter passing
   2. Macro instructions

I. Interrupt handling
   1. Interrupt vector table
   2. Terminate and Stay Resident programs

J. Conditional assembly

K. Disk operations
   1. File accessing
   2. Direct disk I/O
   3. Boot sector
   4. File allocation table
   5. Directory layout

L. Video Basics
   1. BIOS

M. Memory Management
   1. Real memory
   2. Expanded memory
   3. Extended memory
   4. Virtual memory
   5. Dynamic address translation

N. Hardware simulation
   1. Verilog basics
   2. Modeling structure
   3. Hardware simulation

Methods

- ✓ Lecture
- ✓ Laboratory
- ✓ Demonstration
- Discussion
- Distance education
- Other

Assignments

Reading 1: Read the chapter on the organization of computers and be able to give a general overview of the organization of computer.

Reading 2:	Read the chapter on arithmetic flags and operations and be able to discuss how conditions are tested, how flags are set and how condition jumps are performed.
Writing, problem solving, performance 1:	Write a virtual memory simulation program.
Writing, problem solving, performance 2:	Design a half adder using a simulator.
Lab, field activity, product or report:	
Other:	

Evaluation

Sample One:	Lab: Write a program to perform addition, subtraction, multiplication on very long integers.
Figure 1:	
Sample Two:	Exam: Describe in detail the interrupt handling process
Figure 2:	
Other:	
Figure 3:	
Frequency of Evaluation:	Evaluations will adhere to the DVC "Fairness in Grading" guidelines and will include as a minimum: • Evaluation of students within the first quarter of the course and notifying student of the results • Counting a final examination for no more than one-half the course grade • Basing final grades on at least three students' tests and/or reports Additional Information:(1) At least 3 exams, (2) Four lab exercises

Textbooks

Book One

Author: William B. Jones
Title: Assembly Language, Programming for the IBM PC Family
Year: 2001

Book Two

Author: Michael D. Ciletti
Title: Starter's Guide to Verilog 2001
Year: 2004

Other

Description:

Distance Education

Prerequisite

Course Name	Exit Skills	Target
COMSC - 265 Or Equiv.	Be able to use all program structures (sequence, alternation, and iteration).	Apply program structures to understanding machine architecture and machine language programming.
COMSC - 265 Or Equiv.	Be able to write properly modularized programs including calling subroutines by value and reference.	Learn how subroutine calls are handled by computer hardware and system software.
COMSC - 265 Or Equiv.	Be able to use primary and secondary memory devices.	Learn how primary and secondary memory is organized and accessed by machine language programs and systems software.

Corequisite

Recommended

Course Name	Exit Skills	Target

Limitation