Douglas College

EFFECTIVE: MAY 2003
CURRICULUM GUIDELINES

A. Division: **Instructional**

Effective Date: May 2003

B. Department / Program Area: **Computing Science**

Revision: [X] New Course: [ ]

If Revision, Section(s) Revised: **K, M, N, P, Q**

Date of Previous Revision: **June 9, 1998**

Date of Current Revision: **November 18, 2002**

C: **CMPT-250** **D**: **Computer Systems Design and Architecture** **E:** **3**

Subject & Course No.	Descriptive Title	Semester Credits

F: Calendar Description:

This course introduces computer systems design and architecture. It begins with a review of the main digital circuit building blocks in a computer, the basic structure of a single bus computer, assembly language, and addressing modes. These concepts are formally extended by considering various architectures such as RISC and CISC and the relationship between the machine language and the architecture. Processor design in the context of pipelining, horizontal and vertical microprogramming, the ALU, and the memory is considered in depth.

G: Allocation of Contact Hours to Type of Instruction / Learning Settings

Primary Methods of Instructional Delivery and/or Learning Settings:

Lecture / Laboratory

Number of Contact Hours: (per week / semester for each descriptor)

Lecture **3 hours / week**
Laboratory **2 hours / week**

Number of Weeks per Semester: **14**

H: Course Prerequisites:
CMPT 150 with a minimum grade of C

I: Course Corequisites:
None

J: Course for which this Course is a Prerequisite:

K: Maximum Class Size:

Lecture **25**
Laboratory **25**

L: PLEASE INDICATE:

[ ] Non-Credit
[ ] College Credit Non-Transfer
[X] College Credit Transfer:

SEE BC TRANSFER GUIDE FOR TRANSFER DETAILS (www.bccat.bc.ca)

M: Course Objectives / Learning Outcomes:

The student should be able to:

- **demonstrate an understanding of the relationship between the machine language and the computer hardware in the context of functionality and complexity by**
  - **designing and implementing programs in machine and assembly language**
  - **functionally describing architectural support for operating systems and programming languages such as heaps, stacks, and task switching**
  - **describing the function of the hardware using a formal description language such as RTN (Register Transfer Notation)**
  - **virtually simulating the hardware functions**
    - **using a high level language such as VHDL, Verilog, or C++**
    - **using a logic circuit simulator such as LogicWorks**
  - **quantitatively describing the complexity and speed of various architectural components using mathematical functional notation and timing diagrams**
- **understand numbers of various bases and operations to be done on them by**
  - **mathematically defining fixed point and floating point numbers**
  - **designing arithmetic circuits or algorithms used to implement addition, subtraction, multiplication, and division**
- **understand the concept of microprogramming demonstrated by**
  - **describing the hardware using LogicWorks**
  - **implementing, using either a horizontal or vertical microprogramming architecture, some instructions**
- **functionally describe at the hardware level various computer architectures such as**
  - **RISC versus CISC**
  - **one bus versus multi bus organizations**
  - **the concepts of parallelism and pipelining**
- **describe at the hardware and software level techniques for**
  - **Input and Output devices and their interface**

N: Course Content:

1. **Components and Structure of a Computer**
   - **1.1. Registers, Counters, ALU**
   - **1.2. CPU structure**
     - **1.2.1. Harvard versus Princeton Architecture**
     - **1.2.2. Reduced Instruction Set Computers versus Complex Instruction Set Computer**
     - **1.2.3. Metrics for the comparison of processors**
   - **1.3. Control Unit**
     - **1.3.1. Finite State machines**
     - **1.3.2. Micro-programming**
   - **1.4. Advanced Computer Architectural Designs**
     - **1.4.1. Pipelines, branch prediction**
2. **Advanced Assembly Language Concepts**
   - **2.1. Macros, interrupts, system stack for parameter passing**
   - **2.2. I/O programming and interfacing**
   - **2.3. Multi-tasking**

O: Methods of Instruction:

There are three components to the course: lectures, labs., and assignments.

The lecture is used to introduce new material; usually via a sequence of theoretical concepts, examples, and practical considerations. The book is to be used as a close adjunct to the lecture notes and examples.

The two hour weekly lab. is used for the teaching and evaluation of processor, ALU, and memory designs, circuits using the software product LogicWorks and assembly language programs.

Assignments include, but are not limited to, logic designs some using LogicWorks others using VHDL or C++, microprograms and assembly language programs.

P: Textbooks and Materials to be Purchased by Students:

- **Portfolio for logic design assignments**
- **Two 3½" high density diskettes**

Q: Means of Assessment:

Evaluation will be carried out in accordance with Douglas College Policy. The final grade will be calculated from a particular distribution from the range below. The exact distribution will be given to the student on the first day of classes along with the course outline.

labs. (6 to 12)	**15% - 30%**
projects/assignments (1 to 6)	**25% - 40%**
tests (1 to 2) @ 15% - 25% each	**15% - 50%**
final examination	**20% - 40%**
class participation$_1$	**0% - 5%**

Note #1: participation includes (but is not limited to) short pop quizzes and/or handing-in (part-of) a homework assignment

R: Prior Learning Assessment and Recognition: specify whether course is open for PLAR

Not at this time

Course Designer(s):

Education Council / Curriculum Committee Representative:

Dean / Director:

Registrar: