Douglas College

Course Information

A: *Division* Instructional

B: *Department* Science and Technology

Date June 9, 1998

New Course X

Revision of Course _____

Dated _____

C: CMPT-250 (*Course Number*)

D: Computer Systems Design and Architecture (*Descriptive Title*)

E: 3 (*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.

Summary of Revisions

G: *Type of Instruction*

Lecture	3 hrs/week
Lab.	2 hrs/week
Seminar	
Clinical Experience	
Practicum	
Shop	
Studio	
Student Directed Learning	5 hrs/week (approx.)
Other	
Total	10 hrs/week

H: Course Prerequisites:
CMPT-150

I: *Course Corequisites*
None

J: *Course for which this course is a prerequisite*

K: *Maximum Class Size*
Class 34
Lab. 34

L: *College Credit*
Transfer X

M: *Transfer Credit*
Requested X
Granted _____

Course Equivalents

U.B.C.	ELEC259
S.F.U.	CMPT250

Course Designer

Vice-President - Instruction

Desmond Wilson

Dean

Registrar

N: *Textbook and Materials to be Purchased by Students*

- Heuring V.P. and Jordan H.F., Computer Systems Design and Architecture, Addison-Wesley
- Portfolio for logic design assignments
- Two 3½" high density diskettes

O: *Course Objectives*

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 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
    - memory access to primary, cache and virtual memory
    - Input and Output devices and their interface

P: *Course Content*

1 The General Purpose Computer
  1.1 Views: programmer's, architect's, hardware designer's
  1.2 Review of Simple-As-Possible (SAP-1) computer
    1.2.1 Relationship between instruction set and hardware
    1.2.2 Microprogramming a control unit for SAP
2 Computer Classification, Formal Description, and Assembly Programming
  2.1 Reduced Instruction Set Computers (RISC) versus
  Complex Instruction Set Computers (CISC)
    2.1.1 RISC
      2.1.1.1 RTN description of SRC computer
      2.1.1.2 SPARC
      2.1.1.3 PIC
    2.1.2 CISC
      2.1.2.1 MC68000
3 Processor Design
  3.1 Design Process
    3.1.1 One bus architecture
      3.1.1.1 SRC
    3.1.2 Harvard bus architecture
      3.1.2.1 PIC
  3.2 Advanced Concepts
    3.2.1 Pipelining
    3.2.2 Microprogramming
4 Arithmetic Units
  4.1 Review of number systems, addition, subtraction
  4.2 Multiplication and division
  4.3 Floating point numbers
5 Memory Systems
  5.1 Review of RAM, addressing
  5.2 Memory hierarchy
  5.3 Cache
  5.4 Virtual memory
6 Input and Output (I/O) Devices and their Interface
  6.1 Programmed I/O
  6.2 Interrupts
  6.3 Direct memory access
7 Architectural Support for Operating Systems and Programming Languages
  7.1 stacks
  7.2 heaps
  7.3 task switching

Q: *Method 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 3, assembly language programs.

Assignments include, but are not limited to, logic designs some using LogicWorks 3 others using VHDL or C++, microprograms and assembly language programs.

R: *Evaluation*

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 and necessary policies.

Distribution Range:

class participation[1]	=	0% - 5%
labs[2] (7 - 14)	=	15% - 25%
1 - 2 tests @ 15% - 20% each	=	15% - 40%
1 exam	=	20% - 30%
assignments (4 - 6)	=	20% - 35%

Notes:

1. participation includes (but is not limited to) short pop quizzes and/or handing in (part-of) a homework assignment
2. all labs are evaluated although some may be formative (i.e. not count towards the final grade)

Example Distribution:

class participation	=	5%
8 labs.	=	20%
midterm	=	20%
assignments	=	25%
exam	=	30%
Total	=	100%