CSci/ECE 374
Computer Organization and Architecture
Spring 2006
Department of Electrical and Computer Engineering
North Dakota State University

Instructor: Fei Dai, Ph.D.
ECE 101P; Tel.: (701) 231-7217; E-mail: fei.dai@ndsu.edu

Lecture: Monday, Wednesday and Friday, 4:00pm to 4:50pm
Location: ECE 123

Office hours: Monday, Wednesday and Friday, 1:00pm to 2:00pm (or by appointment)

Grading:
Homework = 20% (6 to 7 sets due in 1 week after it is assigned)
Unannounced quizzes = 10% (5 quizzes anytime during the class)
Term project = 10% (simulation or report)
Two tests = 15% each (50 minute in class tests)
Final exam = 30% (2 hour exam)

Homework, project, tests, and exam:

Homework and project must be submitted before 4:00pm on the due date in the class. Late homework and makeup exam will be allowed only for the cases of family emergency and illness with adequate supporting evidence (such as doctor’s note).

Term Project:

A term project is either a program that simulate the behavior of a computer component (cache, instruction dispatcher, virtual memory, etc) or a research report of a specific architectural problem. Each project is assigned to a group of up to three (3) students, and is due by the end of the semester. Each group should determine a project topic during the middle of the semester.

Text:
Structured Computer Organization
5th Edition
Andrew S. Tanenbaum
Peason Prentice Hall, 2005

Prerequisites:
ECE 173 Introduction to Computing (ECE students)
ECE 275 Digital System I (ECE students)
CSci 373 Assemble Language (CS students)

Disabilities:

Any students with disabilities, who require special accommodation in this class, are encouraged to express their concerns or requests to the instructor as soon as possible.

Academic dishonesty/plagiarism:

Homework and examinations in this course must comply with the Code of Academic Responsibility and Conduct as cited in "Rights & Responsibilities of Community: A Code of Student Conduct" (1993), pages 29-30 (available at http://www.ndsu.edu/policy/335.htm).

Grading policy:

The final grade will be on a relative basis. The following grade ranges can be used as minimum criteria:

90%+	A
80%+	B
70%+	C
60%+	D

Broad objectives:

To learn about:

- Components of modern computer and their working principles
- Basic digital logic circuits for CPU, memory, and I/O devices
- Microarchitecture design principles
- Instruction set architecture design
- Operating system concepts
- Assemble language concepts
- Parallel computer architecture

Specific objectives:

To learn to:

- Compare modern computers for each level of organization (A, K)
- Design and use Hamming error-correction codes (A, K)
- Design adder, shifter, memory, and bus circuits (A, C)
- Implement IJVM instruction using Mic-1 microinstructions (A, C, E, K)
- Analyze the performance of pipeline, cache, and out-of-order execution systems (A, B, E, K)
- Design CPU instruction formats (A, C, E, K)
- Analyze the performance of paging and segmentation systems (A, B, E, K)
- Explain the compiling, linking, and loading processes of a program (A, K)
- Analyze the performance of parallel computers (A, B, E, K)

Accreditation Board for Engineering and Technology (ABET) outcomes:

A. An ability to apply knowledge of mathematics, science and engineering
B. An ability to design and conduct experiments, as well as to analyze and interpret data
C. An ability to design a system, component, or process to meet desired needs
D. An ability to function on multi-disciplinary teams
E. An ability to identify, formulate, and solve engineering problems
F. An understanding of professional and ethical problems
G. An ability to communicate effectively
H. The broad education necessary to understand the impact of engineering solutions in a global and societal context
I. A recognition of the need for, and ability to engage in life-long learning
J. A knowledge of contemporary issues
K. An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice
L. An ability to design, develop, implement, and improve integrated systems that include people, materials, information, equipment and energy
M. An ability to integrate systems using appropriate analytical, computational and experimental practices

Assessment tools:

- Tests and exam
- Project
- Homework
- Unannounced quizzes

Main Topics: 1) Computer systems, 2) Digital logic, 3) Microarchitecture, 4) Instruction Set Architecture, 5) Operating system, 6) Assembly level, and 7) Parallel computers.

Chapter 1
1.1.1 Languages, Levels, and Virtual Machine
1.1.2 Contemprary Multilevel Machines
1.3.1 Technological and Economic Forces

Chapter 2
2.1.1 CPU Organization
2.1.3 RISC versus CISC
2.1.5 Instruction-Level Parallelism
2.1.6 Processor-Level Parallelism
2.2.3 Byte Ordering
2.2.4 Error-Correcting Codes
2.2.5 Cache Memory
2.4.1 Buses

Chapter 3
3.1.1 Gates
3.2.1 Integrated Circuits
3.2.2 Combinational Circuits
3.2.3 Arithmetic Circuits
3.3.1 Latches
3.3.2 Flip-Flops
3.3.6 RAMs and ROMs
3.4.2 Computer Buses
3.4.5 Bus Arbitration
3.7.1 I/O Chips

Chapter 4
4.1.1 The Data Path
4.1.2 Microinstructions
4.2.1 Stacks
4.2.2 The IJVM Memory Model
4.3.1 Microinstructions and Notations
4.3.2 Implementation of IJVM Using the Mic-1
4.4.2 Reducing the Execution Path Length
4.5.1 Cache Memory
4.5.3 Out-of-Order Execution and Register Renaming

Chapter 5
5.3.1 Design Criteria for Instruction Formats
5.3.2 Expanding Opcodes
5.4.1 Addressing Modes
5.4.2 Immediate Addressing

5.4.3 Direct Addressing
5.4.4 Register Addressing
5.4.5 Register Indirect Addressing
5.4.6 Indexed Addressing
5.4.7 Based-Indexed Addressing
5.4.8 Stack Addressing
5.4.9 Addressing Modes for Branch Instructions
5.4.14 Discussion of Addressing Modes
5.5.2 Dyadic Operations
5.5.6 Loop Control
5.6.5 Interrupts

Chapter 6
6.1.1 Paging
6.1.2 Implementation of Paging
6.1.3 Demand Paging and the Working Set Model
6.1.4 Page Replacement Policy
6.1.5 Page Size and Fragmentation
6.1.6 Segmentation
6.1.7 Implementation of Segmentation
6.2.2 Implementation of Virtual I/O Instruction
6.3.3 Processor Synchronization Using Semaphores

Chapter 7
7.1.3 Format of an Assembly Language Statement
7.1.4 Pseudoinstructions
7.3.1 Two-Pass Assemblers
7.3.2 Pass One
7.3.3 Pass Two
7.4.1 Tasks Performed by the Linker
7.4.2 Binding Time and Dynamic Relocation
7.4.3 Dynamic Linking

Chapter 8
8.3.1 Multiprocessors vs. Multicomputers
8.4.1 Interconnection Networks
8.4.4 Communication Software for Multicomputers
8.4.7 Performance