Directed Independent Adult Learning

COURSE SYLLABUS

COMPUTER ARCHITECTURE

COS-330-GS

THOMAS EDISON STATE COLLEGE

Higher Education.
For Adults with Higher Expectations.

Course Syllabus by Samuel Kohn, Ph.D.

Course Syllabus
COMPUTER ARCHITECTURE
COS-330-GS

January 2008

Course Essentials

Computer Architecture is an introduction to the organization and architecture of computer systems. The course begins with the standard von Neumann model and moves toward more recent architectural concepts. In line with ACM/IEEE-CS 2001 computing curriculum guidelines core areas for computer organization and architecture, this course covers:

- digital logic and digital systems
- machine level representation of data
- assembly level machine organization
- memory system organization and architecture
- interfacing and communication
- functional organization
- multiprocessing and alternative architectures

These guidelines are produced jointly by the Computer Society of the Institute for Electrical and Electronic Engineers (IEEE) and the Association for Computing Machinery (ACM).

This course should also provide a foundation for your studies of operating systems and other areas of computer science.

Advisory: It is advisable to have completed two computer science courses before taking this course. Also, it would be helpful (but it is not required) for students to take a course in discrete mathematics as co-requisite. However, it is expected that all students who are taking this course have the mathematical maturity gained in a year of college-level mathematics (such as calculus or discrete mathematics).

Goals and Objectives

All computer science is based on the computer. Thus, to be a computer professional requires an understanding of what makes up a computer. This course provides an introduction to a computer's components, characteristics, and interactions. The course deals with the interactions of programming with the underlying hardware. Tradeoffs, for example, among various components, such as CPU clock speed versus memory size, are covered.

Your overall goals will be to:

- Evaluate a computer system's architectures.
- Compare and contrast computer architectures.
- Design circuits at the gate level using and/or/not/multiplexors/decoders/encoders.
- Design microcode sequences that implement machine level instructions.

After completing **Computer Architecture** you should be able to:

- Use mathematical expressions to describe functions of simple combinational and sequential circuits.
- Describe the physical limitations of electronic circuits.
- Explain the pros and cons of using different formats to represent numerical data.
- Convert numerical data from one format to another.
- Discuss effects of fixed-length number representations on accuracy and precision.
- Describe internal representation of characters, strings, records, and arrays.
- Explain the basic organization and major functional units of the von Neumann machine.
- Comprehend how an instruction is executed in a von Neumann machine.
- Summarize how instructions are represented at machine levels and in context of a symbolic assembler.
- Identify different instruction formats, such as addresses per instruction and variable length versus fixed length formats.
- Write simple assembly language program segments.
- Demonstrate how fundamental high-level programming constructs are implemented at the machine-language level.
- Comprehend basic concepts of interrupts and I/O operations.
- Explain how interrupts are used to implement I/O control and data transfers.
- Identify various types of buses.
- Describe data access from a magnetic disk drive.
- Compare alternative implementations of datapaths.

- Discuss control points and the generation of control signals using hardwired or microprogrammed implementations.
- Explain basic instruction level parallelism using pipelining and describe hazards that may occur.
- Implement parallel processing beyond the classical von Neumann model.
- Describe alternative architectures such as SISD (Single Instruction, Single Data), SIMD (Single Instruction, Multiple Data), MISD (Multiple Instruction, Single Data), and MIMD (Multiple Instruction, Multiple Data), and VLIW (Very Long Instruction Word).

Course Materials

In addition to the Course Syllabus, you will need the following textbook to do the work of the course. It is available from the textbook supplier, MBS Direct.

Text

The Essentials of Computer Organization and Architecture, 2d ed., by Linda Null and Julia Lobur (Sudbury, Mass.: Jones and Bartlett, 2006).

Accompanying Software

Several types of software are available as downloads from the textbook's companion Web site. More items may become available in time. Check the "Student Resources" area of the Web site often:

MARIE Simulator Software: Allows you to assemble and run MARIE programs.

CAMERA Memory Tutorial Software: Allows you to apply concepts on cache and virtual memory.

Web site address: http://computerscience.jbpub.com/ECOA/index.cfm

Course Structure

Computer Architecture is a three-credit, 12-week course consisting of textbook readings, written assignments, and two examinations. Weekly study assignments are based on your textbook readings (see the “Course Calendar”).

In the weeks in which you are not scheduled to take an examination, you will complete a written assignment and submit it to your mentor for correction and grading. The course requires you to take two examinations, a midterm and a final.

Written Assignments

The written assignments in the course are built around associated textbook chapters. The assignments consist of exercises from the text.

Take the time to familiarize yourself with the "Written Assignments" section of the syllabus, and read through the assignment questions before you begin each lesson. Your answers to the assignment questions should be well developed and convey your understanding of the course materials. Formulate responses in your own words citing text materials where appropriate and in an appropriate manner. Please type or input all work.

Examinations

Computer Architecture requires you to take two examinations, a midterm and a final. Refer to the Student Handbook for information on all exam policies and procedures.

You may take examinations only during the designated exam weeks, at approved locations, and with approved proctors. In this regard, you must submit a "Proctor Request Form" with the necessary documentation to the Office of Test Administration during the first week of the semester.

Both the midterm and the final exam are two hours long. The exams consist of both objective (true/false, classification, multiple choice) and short answer/essay questions. The essay questions focus on major course concepts while the objective questions are more closely tied to topics covered in the "Review of Essential Terms and Concepts" sections at the end of each chapter.

The midterm will cover the material in textbook chapters 1-4, and the final will cover textbook chapters 5-9.

Grading

Your final grade will be determined as follows:

Written Assignments	50 percent
Midterm Examination	30 percent
Final Examination	20 percent

Letter grades for assignments and examinations equate to numerical grades as follows:

A	=	93–100	C+	=	78–79
A–	=	90–92	C	=	73–77
B+	=	88–89	C–	=	70–72
B	=	83–87	D	=	60–69
B–	=	80–82	F	=	Below 60

To receive credit for the course, you must earn a letter grade of D or higher on the weighted average of all assigned course work (e.g., exams, assignments, projects, papers, etc.). You will receive a score of 0 for any work not submitted.

Strategies for Success

To succeed in this course, consider following these steps:

1. Read carefully the entire "Course Essentials" section of this syllabus, making sure that all aspects of the course are clear to you and that you have all the materials required for the course.
2. Take the time to read the entire Student Handbook. The handbook answers many questions about how to proceed through the course, how to schedule exams and arrange for proctors, and how to get the most from your educational experience at Thomas Edison State College.
3. Before you begin working on your assignments, take the time to fill in the dates for the current semester on the "Course Calendar." The Week-by-Week dates you will need to plan your semester's work are located in the General Course Instructions. Once you fill out the calendar, you will know exactly when to begin your assignments, when your written assignments are due, and when to schedule examinations.
4. Each week, consult the "Course Calendar" in the syllabus to determine which chapter in the textbook you are to read and what exercises you are to do for your assignments. It is essential that you follow the "Course Calendar" each week to ensure that you stay on track throughout the course.
5. When reading and studying the text you will find it valuable to use the *Student Lecture Companion: A Note-Taking Guide.* This volume will help you take notes more efficiently and provides a visual guide that follows the chapter topics presented in the textbook.
6. When submitting your answers it is very strongly recommended that you repeat the question before your answer. This will help you to be sure that you fully understand the question before working on an answer. (Writing out the question facilitates this process.) In addition, having the questions

with the answers will make your assignments a better study tool for the midterm and final.

7. Be sure to take advantage of all the sections in your textbook that will help you master the subject matter presented in the course. For example, don't overlook the Questions in the "Review of Essential Terms and Concepts" at the end of each chapter. Although you are usually not required to send in responses to these questions, working out the problems yourself will greatly enhance your understanding of the material. In addition, you are encouraged to test yourself on Exercises where solutions are in the back of the book. These exercises have a blue diamond (♦) before them. Also note that the more challenging exercises have an asterisk (*) before them.
8. Finally, focus on the examples worked out in the text. A good strategy is to pretend that each example is an exercise and try doing the exercise. After you do the exercise you can compare your answer with the answer provided in the text. It is recommended that you do this even after you have read the explanation and the solution in the text.

Areas of Focus

Following are some study notes that will help you focus your reading of the text. This section is organized by chapter. Refer to this material before you read a chapter as well as while you are reading.

Chapter 1

Please focus on the following:

- Computer organization versus computer architecture
- Main components of a computer
- Principle of equivalence of hardware and software
- Computer terminology, including how to interpret a computer advertisement in a newspaper or elsewhere.
- What a computer looks like from the inside
- Standards organizations
- Historical development
- The computer level hierarchy
- The von Neumann model and non-von Neumann models

Chapter 2

Please focus on the following:

- Number systems
- Decimal to binary conversions
- Signed versus unsigned numbers
- Signed integer representation
- Binary arithmetic
- Floating point representation
- Character representation such as ASCII, EBCDIC, Unicode, and BCD
- Codes for data recording and transmission
- Error detection and correction

Chapter 3

Please focus on the following:

- Boolean algebra and common Boolean functions such as AND, OR and NOT together with truth table and Boolean identities.
- Logic gates and digital circuits using logic gates
- Digital components
- Combinational circuits such as adders, decoders, multiplexers, and parity checkers
- Sequential circuits and their use in retaining and remembering values
- Clocks and their use in synchronizing circuits
- SR, JK and D flip-flops and their use in building components such as particularly registers and memory

Chapter 4

Please focus on the following:

- CPU (Central Processing Unit) basics and organization

- Datapath and its use for storage units and arithmetic and logic units connected by buses
- Registers as addresses, program counters, or data necessary for program execution
- ALU (Arithmetic Logic Unit) and how it carries out the logic operations and arithmetic operations and how it is integrated into the computer system
- Control Unit and its use in extracting instructions from memory, decoding instructions, assuring data is in the right place at the right time, letting ALU know which registers are to be used, servicing interrupts, and turning on appropriate ALU circuitry to execute operations
- Buses, such as address buses and data buses, and how they permit system component communication
- Clocks and how they regulate speed of instructions and synchronize system components
- Clock frequency and clock cycle time and their role in determining how quickly a computer can function
- Input/Output (I/O) subsystems and how they are integrated with the computer system
- Memory organization and addressing
- Interrupts that alter the normal flow of execution in a computer system and how they are used for I/O, error handling, and other miscellaneous events
- MARIE (Machine Architecture that is Really Intuitive and Easy) architecture and how it is used to provide an introduction to a simple computer. Basic computer organization and fundamental concepts such as the CPU, fetch-decode-execute cycle, the data path, clocks and buses, register transfer notation are illustrated through a basic architecture and ISA (Instruction Set Architecture). MARIE has the classical von Neumann design with a program counter, an accumulator, an instruction register, 4096 bytes of memory, and two addressing modes. Assembly language programming is used for reinforcing computer architecture concepts such as instruction format, instruction mode, data format, and control.
- Instruction processing such as the fetch-decode-execute cycle
- Register transfer notation
- Assemblers and how they convert assembly language (using mnemonics) into machine language (which consists entirely of binary values or strings of zeros and ones). Assemblers take a programmer's assembly language

program (which is really a symbolic representation of the binary numbers) and convert it into binary instructions, or the machine code equivalent. Assembly language requires intimate knowledge about the architecture and the datapath.

- Hardwired control versus microprogramming the differences between these two methods
- Case studies of real architectures with a focus on register sets, CPU speed, and instruction set architectures
- Please note: A word does not have to be 32 bits. Word length is whatever architecture specifies. Many machines have words of more than 8 bits, but are still byte-addressable machines.

Chapter 5

Please focus on the following:

- Instruction formats such as instruction length, little versus big endian, register usage (and the use of stacks), and expanding opcodes
- Addressing modes (although addressing is an instruction design issue, it requires special focus)
- Instruction-level pipelining that allows fetch-decode-execute cycles to be overlapped, resulting for faster execution time. However, resource conflicts, conditional branching, and data dependencies can slow this process down. It should be noted that there are different pipelining types. Instruction-level pipelining is not the only type of pipelining.
- ISA (Instruction Set Architecture), including examples such as Intel, MIPS, and the Java virtual machine

Chapter 6

Please focus on the following:

- Types of memory including the two basic categories are RAM (Random Access Memory) and ROM (Read Only Memory)
- The memory hierarchy including how system memory (registers, cache, and main memory), online memory (hard disk), near line memory (optical disk), and offline memory (tapes and floppy disks) work. Locality of reference (or the clustering of memory references) is integral in understanding how a memory hierarchy works.

- Cache memory and how it speeds up memory accesses by storing recently used data closer to the CPU (in a memory that requires less access time)
- Virtual memory and how it increases available address space by using the hard disk as an extension of RAM. Techniques include paging and segmentation (including advantages and disadvantages) and TLBs (translation look-aside buffer) as a method for improving performance of paging systems.
- Memory management in the Pentium memory hierarchy

Chapter 7

Please focus on the following:

- Amdahl's Law, which recognizes a component's interrelationships within the system
- I/O architectures and I/O control methods (programmed I/O, interrupt-driven I/O, DMA, and channel I/O) and I/O bus operation. Focus on differences among the I/O control methods. For example, please note how interrupt-driven I/O differs from DMA.
- RAID (Redundant Array of Independent Disks) devices for redundancy (in different ways) in storing data facilitating improved performance and increased availability for systems employing. Some RAID levels are theoretical and have not been deployed.
- Data compression and how it economizes on disk usage and reduces transmission time in data communications. Focus on the mathematical concepts.

Chapter 9

Please focus on the following:

- RISC (Reduced Instruction Set Computer) machines and the RISC versus CISC (Complex Instruction Set Computer) debate (including clock cycle time and overlapping register windows)
- Flynn's Taxonomy for categorizing computer architectures and its basis in number of instructions and number of data streams; specifically SISD (Single Instruction stream, Single Data stream), SIMD (Single Instruction stream, Multiple Data streams), MISD (Multiple Instruction streams,

Single Data stream), and MIMD (Multiple Instruction streams, Multiple Data streams) combinations.

- Parallel and multiprocessor architectures including SIMD and MIMD architectures, superscalar, VLIW (Very Long Instruction Word), and vector processors
- Interconnection networks
- Alternative parallel processing approaches to the traditional von Neumann architecture such as dataflow machines (which use a data-driven architecture), neural networks, and systolic arrays

Supplementary Textbooks and Other References:

William Stallings, *Computer Organization and Architecture: Designing for Performance,* 6th ed. (Pearson Education: Upper Saddle River, NJ, 2003).

Andrew S. Tanenbaum, *Structured Computer Organization*, 3d ed. (Prentice Hall, Englewood Cliffs, NJ, 1990).

Vincent P. Heuring and Harry F. Jordan*, Computer Systems Design and Architecture*, 2d ed, (Pearson Education: Upper Saddle River, NJ, 2004)

John P. Hayes*, Computer Architecture and Organization,* 3d ed. (Boston: WCB/McGraw-Hill, 1998).

John L. Hennessy and David A. Patterson, *Computer Architecture: A Quantitative Approach,* 3d ed, (San Francisco: Morgan Kaufmann, 2003).

John D. Carpinelli, *Computer Systems Organization and Architecture* (Boston: Addison-Wesley, 2001).

Nicholas P. Carter, *Schaum's Outline of Computer Architecture* (New York: McGraw-Hill, 2002).

Course Calendar

Using the table of week-by-week dates in the General Course Instructions, write the dates for the current semester in the second column. In the last column, fill in the date for sending each assignment and taking the examinations.

Week	**Date**	**Reading Assignments**	**Written Assignment**	**Date Due or Exam Date**
1		Chapter 1	**1** Submit by Sunday of Week 1	
2		Chapter 2	**2** Submit by Sunday of Week 2	
3		Chapter 3	**3** Submit by Sunday of Week 3	
4			**4** Submit by Sunday of Week 4	
5		Chapter 4	**5** Submit by Sunday of Week 5	
6			**6** Submit by Sunday of Week 6	
7		**MIDTERM EXAMINATION** **(covers material from Chaps. 1-4)**		

8		Chapter 5	**7** Submit by Sunday of Week 8	
9		Chapter 6	**8** Submit by Sunday of Week 9	
10		Chapter 7	**9** Submit by Sunday of Week 10	
11		Chapter 9	**10** Submit by Sunday of Week 11	
12		**FINAL EXAMINATION (covers material from Chaps. 5-7, 9)**		

Written Assignments

The following assignments consist of questions from the "Exercises" section at the end of each assigned chapter in the textbook. Complete all the listed questions for each chapter.

Please answer each question as completely as possible. Before submitting your work, be sure to look over your answers for correctness. For directions on how to prepare and submit your assignments, see the Student Handbook section of the Course Manual.

Assignment 1

Chapter 1: Exercises 2, 6, 8, 10

Assignment 2

Chapter 2: Exercises 2, 4, 7, 8, 12, 19, 22, 23, 30, 36, and 44

Assignment 3

Chapter 3: Exercises 2, 4, 8, 19, 25, and 26

Assignment 4

Chapter 3: Exercises 30, 32, 36, and 40

Assignment 5

Chapter 4: Exercises 4, 11, 16, 17, 21, and 29

Assignment 6

Chapter 4: Complete the "True or False" Items on page 241

Assignment 7

Chapter 5: Exercises 2, 4, 6, 8, 9, 12, 15, and 19

Assignment 8

Chapter 6: Exercises 2, 4, 7, 12, and 16

Assignment 9

Chapter 7: Exercises 2, 4, 11, 19, 22, and 31.

Assignment 10

Chapter 9: Exercises 2, 10, 12, 16, 18, 22, and 24