

The subject of this course is how the hardware of a computer works to make the computer execute programs stored in its memory as machine language. You will have to know what machine language means. Machine language is the programming language into which an assembler translates an assembly language program. The prerequisite subjects of assembly language (CSI333 at Albany) and discrete structures (CSI210 at Albany) are expected to enable you to master sections 6-1 through 6-7, 3-1 through 3-4, and 1-3 of Mano's text without much instruction.

We will look at machine language programs and the data that they process as sequences of words that contain bits. Some of you wrote interpreters in C or C++ for some machine language in CSI333. In this course, you will learn how the computer hardware directly performs that task.

The course will cover some topics in system architecture such as input/output devices and how they are controlled, memory organization of contemporary computers (caches and virtual memory), busses and elements that bear on computer performance (that means speed).

The course will cover most of Mano's book, with the topics of RISC, pipelines and caches taken from Hennessy and Patterson's book. In addition, binary decision logic specification diagrams and the Kline-McClusky logic design minimization algorithm will be covered with supplemental notes.

Computer hardware design relies on mathematical logic. Formal languages are used today for most computer design. Some topics from CSI210 will be used to express course material.

While this course (alone) will not prepare you to design computer chips or other hardware, it will help enable you to make some sense of computer hardware developments as they come along in your future. Hardware sets the practical limits on the speed, capacity and affordability of software systems.

An important benefit to software specialists of a hardware design course is an introduction to specification, design, verification and hierarchical organization in a concrete context. These methods of understanding are beneficial when applied in the generally less concrete contexts of software systems.

## 1 Staff Info

Lecture: LC-6, MWF 10:10AM-12:15PM. Prof. S. Chaiken, office LI67A, phone 442-4282, office hours: Tues, Thurs. and Fri. 2:30-3:45PM.

There will be two graduate student teaching assistants, to be announced. A tutorial/question and answer session on homework will be conducted by one of the TAs, to be announced. The Univ. at Albany newgroup `sunya.class.csi404` on server `cscnews.albany.edu`, a forum for course material discussion, will be activated soon. Finally, all official course announcements will be posted on Web page <http://www.cs.albany.edu/~sdc/CSI404>

## 2 SKNS

SKNS were given to all who applied to them before the class start. Further SKNS will be given by the professor on Wednesday, Jan. 31 or before **in exchange for a generally complete (B or better) written solution to Homework Assignment 1 below.**

## 3 Textbooks

Two textbooks are required. One is second edition of *Computer Organization and Design, the Hardware/Software Interface* by David A. Patterson and John L. Hennessy.

The other is the third edition of *Computer System Architecture* by M. Morris Mano. Most of the course topic details of the first 3/4 of the course will come from that book.

A few of the assignments will require the use of the GNU C/C++ compiler, assembler, binutils and possibly other Unix software. This is available on the Academic Computing Services Unix cluster ("eve.albany.edu") so you should get an account there unless you have your own suitable system.

Supplemental readings and class material on the World Wide Web may be assigned during the semester, as well as supplemental references to other books. You are expected to be able to read and do limited printing from the Web.

Pointers to or copies of supplemental software will also be provided. One recommended book is *The Indispensable PC Hardware Book* by Hans-Peter Messmer, a trade book from Addison-Wesley.

Examinations and Grading		
Midterm Exam 1	Wed. Feb. 21	20%
Midterm Exam 2	Wed. Apr. 4	20%
Final Exam	Tues. May 15, 10:30-12:30	35%
Homework	10 or 11 weekly assignments, equal weight	25%

## 4 Homework

Each homework assignment will be due at the beginning of a non-exam Wednesday class. Some answers will be discussed or made available then. No late homeworks will be graded. If you miss an assignment, you can make up for it by doing the others.

Generally, some of the assignment will be practice problems and others will be graded problems. The graded problems are more difficult and will apply skills and knowledge acquired from the practice problems. The TA's or professor will generally ask to see your work on practice problems and provide tutoring on them to support the graded problems.

## 5 Exams

No makeup exams will be given. If you miss a midterm exam, its weight will be added to the final's weight. If you miss the final, your score will be based on a 0 for the final.

Each midterm will cover the topics of the preceding third of the course. Of course each topic builds upon concepts of preceding topics. About 1/2 of the final exam questions will be based on material of the last third and the other half will be "comprehensive".

## 6 Incompletes

A grade of "I" will only be given for genuine extenuating circumstances that arose beyond your control after the midterm point. Both of the following conditions must be met:

1. Your work must be in good standing as of March 14, the midterm point. That means that your score for homework due before then must be equivalent to at least a C and your Midterm Exam 1 grade must also be equivalent to at least a C. This implies that if you miss Midterm Exam 1 or hardly submit any homework, you are not eligible for an incomplete.
2. Written documentation must be supplied about the extenuating circumstance either by you or the University administration.

Under no circumstances will the condition for completing an incomplete be that the entire course be retaken later without a new registration.

## 7 Disability

Alternative arrangements will be made for clients of the Office of Disabled Student Services according to their recommendations and policies. See me at least one week before about any examination, and arrange for Internet accessibility if needed as soon as possible.

## 8 Cheating

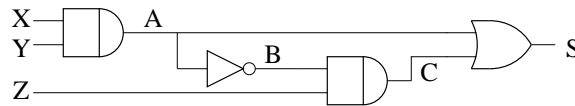
You are welcome to discuss class work, but you are expected to solve each homework problem without copying, seeing or hearing what another person has written. If copying is suspected, the professor or the TA may ask you to explain your solution. Verified copying will result in a zero assignment grade for both parties.

The examinations will be given in two separate rooms to enable appropriate proctoring. Details will be announced before the exam. Cheating in an exam will result in course failure and a report to the Dean of Undergraduate Studies according to the regulations in the Undergraduate Bulletin.

## 9 Homework 1: Due Jan. 31, beginning of class

Readings: Mano, sections 1.1 to 1.5, 2.2 and 2.3 (2.1 is optional). Mano, section 3-1 to 3-3 (prereq. review) P & H: sections B-1, B-2 and B-3.

Problem 1: Figure out the truth table for the combinational circuit below (i.e., *analyze* it) by filling in the columns from left to right. Note that each column label corresponds to a wire (i.e., *node*) in the circuit at which a logic level can be observed.



X	Y	Z	A	B	C	S
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

Problem 2: Consider the Boolean algebra expression below which is a formula for the Boolean function  $F(X, Y, Z)$ .

$$(X + Y)' + Z(Y + Z'X)'$$

1. Draw the combinational circuit that is represented by the expression. Label each node with the subexpression that it corresponds to.
2. Figure out the truth table by filling in the columns from left to right. Note that each column label corresponds to a subexpression.

X	Y	Z	$X + Y$	$(X + Y)'$	$Z'$	$Z'X$	$Y + Z'X$	$(Y + Z'X)'$	$(X + Y)' + Z(Y + Z'X)'$
0	0	0							
0	0	1							
0	1	0							
0	1	1							
1	0	0							
1	0	1							
1	1	0							
1	1	1							

Problem 3: Draw the Karnaugh maps for each of the Boolean functions as expressed by the two truth tables. For each, use the method of section 1-4 to write a simplified sum of products. Write the number for each corresponding

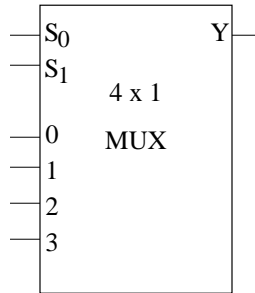
square in the given space first.

square number	A	B	C	F
	0	0	0	0
	0	0	1	1
	0	1	0	0
	0	1	1	1
	1	0	0	1
	1	0	1	1
	1	1	0	0
	1	1	1	0

square number	A	B	C	F
	0	0	0	0
	0	0	1	1
	0	1	0	1
	0	1	1	1
	1	0	0	0
	1	0	1	1
	1	1	0	0
	1	1	1	1

Explain why the arrangement of the squares shown in figure 1-7 is used for K-maps rather than some other such as 1,2,3,4 for the top row. (For more practice, do problem 1-8 in Mano; do not hand in.)

Problem 4: Draw the circuit diagram that shows how a 8-to-1 single bit multiplexer can be built from two instances of a 4-to-1 single bit multiplexer and a few **AND**, **OR** and **NOT** gates. Use the idea of “expansion” which is illustrated for decoders in Figure 2-3 of Mano. Your diagram must show the two multiplexers as two blocks like the one below—You must **not** draw the gates “inside” that implement each MUX.



Besides the two 4-to-1 MUX instances, your design should not contain more than two each of **AND**, **OR** and **NOT** gates.

Also, explain in English why the circuit you gave does what it is supposed to do. (Warning: No credit for a diagram without an explanation in your own words!)

Practice: Mano 3-1, 2, 3, 4, 5, 10, 13, 15, 16

Problem 5: Mano 3-9, Mano 3-6, Mano 3-18.