Instructor Kevin Karplus

Office 315b Applied Sciences (459-4250)

Office hours M 3:30–5:00, Th 2:00–3:00

Classroom Thimann 101

Time M W F 2–3:10

Lab 316 Applied Sciences

1 What is this course about?

CE 222 (VLSI Digital System Design) teaches you the techniques and CAD tools used in academia (and some companies) for full-custom design and layout, in a bulk-CMOS logic process, of digital systems that will work {correctly} the first time. CE 222 prepares you for the follow-on course CE 223 (VLSI Design Project), which is the heart of the sequence. (Although CE 224 looks like part of the same sequence, and is a related subject, it can be taken completely independently of 222 and 223.)

CE 222 covers the principles of VLSI design and teaches you how to use the design tools; in CE 223 you will design a digital integrated circuit all the way from architecture to mask specification. Most of the real learning occurs in 223 and taking just CE 222 does not make much sense academically, so students who pass 222 are normally expected to take 223.

In CE 223, project groups (typically 3 people) specify, simulate, design, lay out, and verify the projects they proposed in the last three weeks of CE 222. The projects are physically small (2.2×2.2mm Tiny Chip™), but often fairly sophisticated. After CE 223, the MOSIS (MOS Implementation Service) silicon broker arranges fabrication of the completed chip layouts and delivers 4 chips in 40-pin ceramic DIPs (paid for by NSF). The prototype chips arrive late in Spring Quarter for your project group to test for functionality and speed on the lab's IMS XL100+ 100MHz IC tester. In return for funding fabrication costs of $440 per Tiny Chip project, the National Science Foundation requires that you submit a MOSIS report on the test results within 3 weeks of delivery, so 223 cannot be taken without a written promise to test the chips when they arrive.

To provide the discipline needed to complete a working chip in only ten weeks with no prior experience, the CE 223 project has to be more structured than projects in other courses, and everyone on the team must contribute fully. The teams form during 222, so you should start looking now for people you can stand being locked in an airless, sunless room with all Winter quarter.

The most sophisticated projects typically relate to someone's research, and are often suggested by professors. CE 223 students concurrently take independent study from the supervising professor, in order to get credit commensurate with the work. The CE 223 final report is also the report for the independent study.

2 Prerequisites

This course has deliberately loose prerequisites, so that people from either EE or CS backgrounds can take it effectively. It should be easily accessible to seniors in Computer Engineering, but has a 200-level number so that graduate students can get credit. The best preparation for the course is to have some knowledge of digital systems design (say CE 100 plus CE 110), some experience working on group design projects (software or hardware, preferably projects too large for one person to tackle), and a real love for creating things that work.

Extra knowledge of electronic circuits, of parallel architecture, of processor design, of computer arithmetic, of object-oriented design, of interfacing, of design-for-test, of dynamic programming, or any of a number of other subjects is useful, but not essential.

A willingness to work and facility at learning new tools quickly are essential.

Perhaps it is possible to get away with writing all your papers with WordPerfect or a Mac (though using LaTeX on the Suns will be easier for group writing and editing), but you still need to know some UNIX text editor, well enough to prepare input files of a few pages, and very soon. If you are not already familiar with a UNIX editor, I recommend emacs as the most versatile and powerful editor. You'll also need to know the rudiments of UNIX and using the X window system.

3 Why should you take this course?

All modern computers and nearly all digital systems are built around VLSI chips. Therefore, much of modern computer engineering is concerned with the design, simulation, optimization, testing, packaging, and interconnection of integrated circuits and the systems they comprise.

The VLSI course is somewhere between being a core course and learning a trade. It is not a theoretical course in "VLSI Appreciation"—the chips designed in CE223 are useful systems that really work! For undergraduates and MS students, your CE223 project will probably impress employers more than your thesis or anything else on your resumé. It seems that there are always jobs for people with VLSI design experience. We lose so many good students to industry this way!

Full-custom VLSI is the "assembly language" of the integrated circuit medium. Even though you may not intend to design VLSI at such a low level professionally, learning it exposes the full potential and limitations of the technology.

Since good system design depends so much on the implementation medium, this is the best way to learn system design. Full-custom VLSI is a particularly good systems-implementation technology to learn, not only because it is the technology of high-performance systems produced in volume, but also because in full-custom VLSI design the issues you deal with are fundamental design issues, and not just artificial quirks of the latest fashion in high-level design abstractions (which come and go like fads, and everyone's got a different one).

The design issues we face are fundamental enough to have kept the VLSI course relevant for fourteen years, with only incremental changes as new tools have been introduced (and one change from nMOS to cMOS many years ago). Full-custom design/layout tools haven't changed much since then, because the technology, bulk CMOS, has not changed significantly except in scale. The pincount and area that MOSIS provides for course projects has been kept nearly constant, because the time available for a project is as astonishing short as ever (perhaps shorter), and the scale of full-custom projects that can be completed by inexperienced designers has not increased.

A big part of what is learned in CE 223 is how to view a design using several different levels of abstraction, how to break a large project into manageable components, how to divide the labor and work cooperatively, and how to budget time to keep a project on track. All these skills are extremely useful in industry, and are hard to teach without a large project.

The enduring lessons of VLSI systems design are also valuable if your goal is systems architecture—particularly for parallel processing. VLSI has been a driving force behind that field, and not just because VLSI implementation is the only thing that makes parallel computers practical. Since parallel computers exist only for cost-effectiveness, not greater convenience of programming, the forms of parallelism that scale up well in VLSI are what inspire parallel architectures.

Parallel computers are a pragmatic expression of the kinds of parallelism inherent in VLSI technology, just as the von Neumann architecture reflected what was practical with vacuum tubes and mercury-delay-line memory. If you want to design better parallel architectures, or do useful theory about parallel computing, study the limitations and potential of VLSI.

If you want to write CAD tools for VLSI design, you would do well to study what chip designers do—and what better way to find a unfilled niche than to try designing with existing tools! It is amazing how many CAD tools for VLSI are written by people with no interest in, or knowledge of, chip design. The tools that designers like and use have generally involved chip designers in all stages of the program development. Caveat: the tools we are using are quite old now, as academic research in CAD has moved to more specialized fields. Check the literature to see what else is available before devoting a lot of time to developing a new CAD tool.

Similarly, if you are interested in testing, test generation, or design for test, you should first learn how to design in this medium. Much of the recent research in test generation and design-for-test has been in working with "realistic faults" which require a fairly detailed understanding of how custom and semi-custom chips are laid out—probably more detailed even than what we cover in CE 222.

A thin-film Multi-Chip Module (MCM) is just a big chip, with a large feature size and no transistors, which can be fabricated on old chip-making equipment that otherwise could only make obsolete military replacement parts. The importance of thermal management, transmission line reflections and losses, clock distribution, and power distribution are typical of bipolar chips and large chips in general, but CE 222 barely has time to touch on the problems specific to big chips, and many of these issues are ignored with relative impunity in the tiny chips designed in CE 223. Therefore students of MCMs also need a more advanced course, for which CE 222 would certainly be a prerequisite. The first step in learning to design big chips is to start by designing a small one, which is something you can do now in CE 223.

4 Books and supplies

The required text is the second edition of Weste and Eshraghian's *Principles of cMOS VLSI Design*.

I used to recommend, as an optional text, Glasser and Dobberpuhl's *The Design and Analysis of VLSI Circuits*, but it has gone out of print. This text is a bit too advanced for a first class in VLSI, but is an excellent reference for students who continue in VLSI past CE 223.

Another optional text, Thomas and Moorby's *The Verilog Hardware Description Language* will not be for sale in the bookstore—students did not find it a particularly useful introduction to Verilog the last time the course was taught. You will be expected to learn Verilog enough to use it for at least one assignment in 222, and for behavioral and architectural simulation of your chip in 223.

Both Glasser and Dobberpuhl and Thomas and Moorby will be put on reserve in the Science Library.

Rather than put together a reader containing the documentation for the design tools, I will put several copies in the lab, where they will be available for ready reference—it doesn't help much to take home the manuals anyway, as you can't really learn much about the tools without the screen in front of you. Other interesting books may be put on reserve in the library.

You will need **erasable** colored pencils in the following colors: carmine red, green, blue, brown, and purple. Colored pens and non-erasable pencils are less useful, because you will spend more time modifying your designs than you will creating them in the first place. Some of the design work is easier to do with pencil and paper (and eraser!) than on the computer, and so the pencils are really essential.

5 The VLSI Design Lab

The VLSI Design Lab is room 316 Applied Sciences. The lab contains documentation for all the hardware and software, and a small library of projects done by students in previous years. **None of the documentation is to leave the lab, ever.** The old project reports can be checked out for a day or two—please return them, as we have no duplicates. Keys to 316 will be available from Diane Brooks.

The machines in the lab are now antiques, and you won't be using them much, except as X-terminals to other machines. It also looks like we'll be using them for Verilog simulation, since Verilog does not appear to be installed on any other machines in the Baskin Center.

CE 222 and CE 223 require many hours in front of a workstation, and, although people doing VLSI design have priority for the lab's workstations, as due dates approach, you will often find them all taken unless you work unpopular hours (6 a.m. to noon). There are other color X windows platforms around the Baskin Center (rooms 57, 162, 230, 310, and 314, for example). Most of these are in grad student offices, for which grad students already have keys. Some are dedicated to particular research groups, but the ones in 314 are for general grad student use. Undergrads in the course can get a key to 316 and a key code for 314.

Becoming very proficient at Magic, and the rest of the UC Berkeley suite of VLSI CAD tools, will be an ongoing assignment (not handed in). If you wait until some particular tool is required before you start learning to use it, you will have very little time to do the actual assignment. The Magic tutorials and CAD manuals are in the black binders in the bookcase behind the door.

6 Security

The lab's computers may have lost all their resale value, but they would be difficult to replace, and the chip tester is still a very expensive piece of equipment. The lab must always be locked when unoccupied—even if you are just leaving "for a minute".

To get a key to 316 Applied Sciences, you need to bring the form that we will supply, signed, to Diane Brooks in 253A. You must return the key when you stop taking VLSI courses. You also need to ask Diane for a keypad code for room 314 next door, where the laser printers are located. And you'll need an outside door key to get into the Baskin Center after 5 p.m. or on weekends (this class takes a lot of hours!). Unfortunately, the key to the outside doors nearest the lab is also the key to graduate student offices, which undergraduates can't get, so they have to enter via the second floor and take a roundabout path. By showing your ID at the Stockroom near the Thimann loading dock, you can get an Applied Sciences building key that gets you into the second floor from the back stairwell near the parking lot. From there, walk up the indoor stairs near the CE/CIS Board Office.

The whole department has been very concerned about security ever since the theft a few years ago, unapprehended despite numerous witnesses, of some Sun 3/60 CPU's (leaving behind their Sony Trinitrons, which is how the lab got some). Our insurance won't pay unless we take every precaution, so please don't prop open the outside door.

If you order food delivered, have them call just before they bring it, so that you can have someone meet them outside at the elevator. The pizza places that will deliver here have no trouble with this. The lab's phone can call out by dialing 6 first, but will make only local calls.

7 Lab Etiquette

When left idle, most workstations automatically blank the screen to avoid burning the current image into the phosphor of the monitor. To find out if it's already in use, tap the shift key. If a window system is running, this will light up the screen without affecting anything else. If nothing happens (and the monitor is on), then tapping the return key or space bar is the least dangerous thing to do. As a courtesy (and to save electricity, wear, and air conditioning), log out and turn off the monitors whenever they will be idle long enough to blank.

But do *not* turn off the CPU.

Our workstations are multi-tasking machines accessible over a network, not single-user PCs. Even when there is nobody coming in over the network, Unix is never completely idle. A power-off crash in the middle of some periodic background activity may corrupt some system file, so Unix should always be shutdown by a superuser before anyone cuts the power.

If power fails, the Suns stay off until someone presses the reset button on the surge protectors under the tables. If the power fails for long enough to trip the surge protectors, wait a while before resetting them—our power failures often come in bursts separated by a minute or two.

Room 316 (and other core rooms) were air conditioned at enormous cost, and so it is definitely uncool to prop the door open (even if the air circulation is not what it should be). The only time when propping the door open is ok is when the air in the hallway is cooler than the air in the lab (which sometimes happens late at night).

Other than emptying the trash, the janitors are afraid to clean the lab, lest they damage the computers. You must clean up your own mess, or it will stay that way until the between-quarter cleaning. They don't clean the whiteboard either, but don't expect anything important to stay there.

Please confine food to the big conference table in the middle of the room—that's what it's there for. The keyboards show the disgusting effects of people eating sticky stuff like chocolate and candy while working. Dropping corrosives like pizza sauce into a keyboard will ruin it, and a Pepsi spilled into a CPU would be too horrible for words. There is no maintenance on the old SUNs in 316 any more—if you break it, it stays broken.

If your carelessness ruins one of the beans, you may have to face the wrath of your fellow students who have to finish the quarter with one less machine.

Don't be excluded from the camaradarie that comes from shared hard work—we encourage you to get away from the screen from time to time. You'll probably end up taking your breaks in the lab (what else is open on Science Hill in the middle of the night?), and talking across the room is inherently loud above the noise of machines and air conditioning. We realize that when several people take a break it can get loud and rowdy in the lab, so it is probably best for everyone to try to synchronize their breaks. Take the time to get to know your classmates, and talk about something besides your projects for a while. This class can be a memorable social experience.

8 Evaluation

Evaluation will be based on three main components:

- Homework, consisting mainly of small design exercises to familiarize you with the tools.
- Exams, to test your familiarity with the basic concepts of the design method. All exams will be open book and open notes, but no living references are to be used.
- Project proposal, a written description of what you plan to do in 223. This quarter you will be expected to form teams of three people, write a proposal for a chip design, revise the proposal after consultation with me, and do a block diagram and behavioral description of the proposed chip. You are required to write a project proposal, even if you do not plan to continue in 223.

Although there is no formal evaluation of class participation, I do notice who comes and who doesn't, and may comment on it in the narrative. Please come to class even if you haven't finished the homework—I cover a lot of material that isn't in the book.

The criterion for passing 222 is my judgement that you are adequately prepared to do 223. Inability to cooperate on a group project, meet deadlines, or do competent work are sufficient grounds for failing the course.

Do all the homework assignments—you'll regret it in 223 if you don't learn the stuff now. If you get grossly overloaded, it is better to do part of each homework assignment than to skip one entirely.

9 Collaboration versus individual work

The project proposal this quarter and the projects themselves will be team efforts, and you are expected to co-operate fully with the members of your own team (and to a lesser extent with other teams) in completing the project.

Learning the design tools is also a co-operative effort. Feel free to ask others how to use the tools, and to share your own tips and tricks.

The homeworks and exams are strictly individual work. I need some way to distinguish the individuals within a group, and to identify those who are not keeping up with the class. The penalty for cheating (claiming someone else's work as your own) is to be deleted from the class with a permanent mark in your record giving the reason—don't risk it![1] If you are not clear on the boundary between permitted co-operation and cheating, please come and talk with me.

10 Tentative schedule

I've once again rearranged the topics of the course, and attempted to identify the parts of the books that cover the topics we'll discuss. The following schedule is only tentative—we may spend more or less time on some of the topics, and we may need to insert other topics.

The "other topics" listing in the schedule will probably be some selection from the following topics, depending on class interest and how much the schedule slips. Uncovered topics will be covered in 223, if needed.

Read	subjects
5.5.7	synchronization failure
	self-timed systems
4.4	inductance & switching noise
5.6	I/O pad design
7.1–2	fault models; fault simulation
7.3	design for testing
	logic minimization
	electromigration

[1]Because campus regulations do not permit instructors to fail students for cheating except on essential work, all homework and exams in this course are essential. If circumstances warrant, late papers or makeup exams will be permitted.

Preliminary Schedule			
date	Read	subjects	assignments
Sept 29	1.1–1.5.6, 5.4.1	digital amplifiers, switch logic	
Oct 2	1.5.7,5.5.1–2	dynamic & static storage	start learning wirec and rsim
4	5.5.(3,4,5,9,10,15)	clocking disciplines	assign circuit design 1
6	5.4.4,5.4.6,5.4.7	precharging and domino logic	
9	5.5.8	precharging and domino logic	circuit 1 due, assign circuit 2
11	1.6	Verilog intro	start MAGIC tutorials
13		Verilog intro	
16	3.1–3.2	layers, colors	circuit 2 due, assign Verilog
18	3.4	geometric design rules	
20		geometric design rules	
23	5.3	layout example	Verilog due, assign MAGIC 1
25		stick diagrams	
27	8.2.1	adder structures	
30	8.2.7	multiplier structures	
Nov 1	8.4	state machines and PLAs	
3		systolic arrays	MAGIC 1 due,assign PLA
6	4.2,4.3.1,4.3.3	sheet resistance, capacitance	[ICCAD]
8	3.5	cMOS latchup	[ICCAD]
10	4.7	power estimation	PLA due,Assign MAGIC 2
13		MIDTERM EXAM	
15		project discussions	
17		project discussions	start project proposal
20	4.5.4.3	RC timing	MAGIC 2 due
22		RC timing	assign RC timing
24		THANKSGIVING	
27	5.5.10,5.5.16	clock generation	
29	8.3	RAM	
Dec 1		other topics	RC timing due
4		other topics	
6		other topics	project proposal due
8		other topics	
Dec 12		FINAL EXAM Tues., 4–7 p.m.	