EE371

Advanced VLSI Design

Mark Horowitz

Computer Systems Laboratory

Stanford University

horowitz@stanford.edu

Class Overview

This class builds on EE313 and EE271 to look at the circuit design issues in large digital VLSI chips. At the core of this class is the job of ‘circuit design’ and the tasks that a circuit designer does in industry. As we will see, much of the current effort is directed at dealing with the wires, and various signal integrity issues.

Like EE271 we will look at large functional blocks; like EE313 we will use SPICE a lot. Since simulating a large function block in SPICE is not fun, much of the class will be devoted to building up ways of evaluating a circuit without having to simulate the whole thing. This skill is really the art of circuit design.

I will assume everyone has taken EE313, and is familiar with MOS models, logical effort, and SRAM design.

.

Class Readings

I wish there was a good book on circuit design. But I have not found it yet. So the lecture notes are the principle reference material that you will use in the class. While the notes will cover the material in the class, they will not be as complete as the information that you would find in a textbook.

To supplement these notes, I will also copy some key papers on circuit design issues. There is a book we will often refer to -- The Design of High-Performance Microprocessor Circuits. It is a collection of papers that cover much of the material in the class. We will also have a number of readings from this book. You are responsible for the material in these articles, even if we don't completely cover them in class. So it would be a good idea to read the papers before coming to class, and asking questions about the material you did not understand.

To provide additional information and/or an alternative explanation of the material in the notes, references to readings from other textbooks will be included in the notes. While these readings are not required, they are often helpful in understanding the material.

Course Goals

Dilemma:

While there are many techniques that don't work, there are also many techniques that work. Often it is hard to know what the "right" answer, since many approaches will solve the problem. The best solution often depends on the application and changes with time.

Goal:

This class will give you some tools that are needed to solve circuit design problems, and will provide you problems to practice using these tools.It will also provide examples of current design practice, so you can evaluate them and come to your own conclusions about their efficacy. This experimentation will help you with building the foundation you need to choose the appropriate circuits and verification methods to your problems.

Caveats

1. People lie. Sometimes intentionally, sometimes by accident, and sometimes it is because what you heard is not what they meant. Never take anything on blind faith. Work it out -- make sure it works. Many clever circuits that are published either don't work or are very sensitive to certain conditions. Be careful.

2. (Corollary to 1) Don't trust me. I am human too and have my bias and blind spots. I will try to tell you what I know, but I am not perfect, and so errors will appear. If you don't think what I am saying is correct, PLEASE let me know. I am from MIT and I can take it.

3. There are no RIGHT answers, and there are no PERFECT circuits. Everything has its warts. A good circuit simply has the right set of warts to meet the constraints of the problem.

4. Simulation is no substitute for thinking. SPICE can make your job much easier, but it also can make it much harder. Like all tools it helps only if you use it well. And that requires thinking.

Class Topics

First part of class will be focused on learning the fundamental tools:

- What to model / what to verify
- Device models, MOS and wires
  - Assume everyone has a background in MOS models from EE313
  - Add information about matching issues and wires
- Worst-case design. Simulation methods. Margin testing

Next look at circuit "environments":

- Timings -- clocking methods and elements
- Electrical -- Power, ground signals

Then look at larger structures:

- Adders, multipliers
- High-speed IO

Logistics

Please come to class:

- I know it is televised, and on the internet, but if no-one comes to class, then no-one will ask me questions, and no-one will learn anything. I promise to be interesting, if you promise to attend lecture.
- And if people don't show up for class I will cut the in-campus broadcast

There is a review session for the class

- Time will be announced
  This session should be televised, and on the internet too, if you can't make it. It will explain the homework (some more) and answer other questions that you have. It will review new tools that you can use for the class.

Tools used in this class

- Sue (schematic editor)
- HSPICE

Homework

There will be a number of weekly problem sets, and a final design project.

Homework:

- Handled out on Thur, due next Thur in class. No late assignments. Will give you a day to look at assignment before the review session
- Homework will be to get experience with the technology and simulation and to experiment with new design techniques. There will usually be one, more open-ended design problem on each problem set.
- Grading the homework is hard. If we can't find graders, on Friday there will be a grading party, where the homework gets graded. Students in the class will be required to sign up for one grading party. At the party each student will get one problem, or part of a problem. The TAs or I will be around to help answer questions in grading.

HSPICE

The problem sets will use a version of SPICE called HSPICE. This version has a number of features (like parameter sweeps and optimization) that will make your life easier. While the homework can be done with other versions of SPICE, please let me know if you don't have HSPICE available in your location. You will need to get approval from me to use another simulator.

- There is a library provided for the class
  - We are going to try something new this year. Rather than using our old 0.35μ CMOS technology, we are going to use a version of a 90nm technology. We are working on getting the corner models worked out, but I hope it will be step for the class. I hope that will make some of the questions more interesting
  - Wire models will also be provided, and will be updated for a 90nm technology
- Some of the assignments will require you to estimate parameters for a 'scaled' technology
- One of the review sessions on Friday will explain how to use the key features in HSPICE. I assume that you have all used it in EE313.

Project

The project this year will be a little simple compared to some of the previous projects. It will look at trying to build a simple 64 bit adder. The only trick will be that the goal of the project is to fill a certain area with adders and the wires that interconnect them (and the registerfile to feed them). You will be limited on power as well, so the key will be to make the right trade-offs to get the maximum aggregate performance. The homeworks should give you the background that you need to do the project.

Or it will be something completely different. But it will probably be an adder.

This project will require verilog for the design, as well as a schematic, and will be done in groups of 2 people. You should find a partner that you can work with early in the quarter. Try to find someone you can work with and who complements your skills

Honor Code

Please remember you are bound by the honor code:

- I will trust you not to cheat
- I will try not to tempt you

But if you are found cheating it is very serious

- There is a formal hearing
- You can be thrown out of Stanford

Save yourself and me a huge hassle and be honest

Lecture 1:

Models, Simulation, and Circuit Design

Computer Systems Laboratory

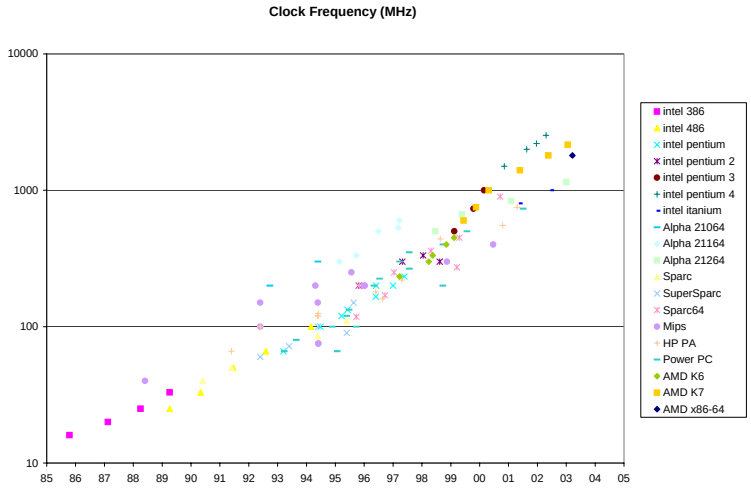
Stanford University

horowitz@stanford.edu

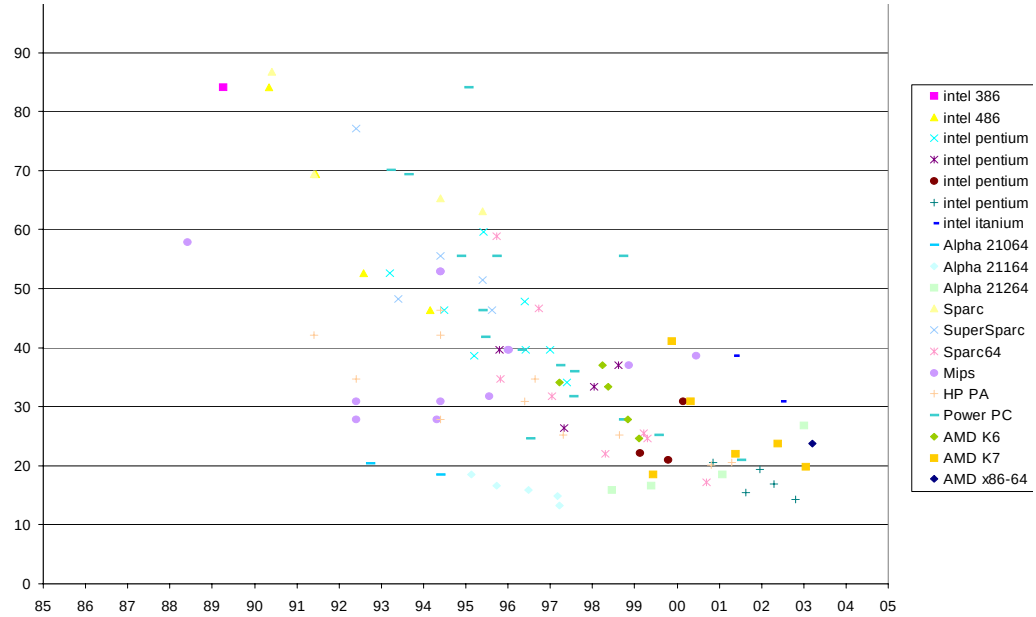
Integrated Circuit Trends

Two trends have made circuit designers in high demand:

- Process scaling and demands for high-clock rate processors



Cycle Time in FO4



What These Trends Mean

The cycle time of digital systems is falling faster than the speed of the gates

- Need to build functions and wires that are faster (in terms of speed of the basic building blocks) than last time

Long wires are becoming wet noodles

- Pushing information through them is getting harder
- Need to worry about wire delay and noise coupling issues

Faster Functions

There are two approaches to building faster functions:

- Build faster logic gates
  - Here you use some experience with transistors to craft faster basic gates
  - Dynamic logic, and certain passgate logic is the best example of this
  - Customize the layout to make the critical wires short
- Build faster architectures
  - Leverage all those additional transistors that are possible
  - Do more things in parallel, do some speculative calculations
  - Decrease the time for the high-level function, by using many standard gates

Today, the faster architecture approach is ALWAYS used, so digital circuit designers must deal with environments where there are a large number of gates and wires to deal with.

Example: Adder

If you want to build a fast 64 bit adder:

- Look over the different architectures proposed
  - Ripple, carry bypass, carry select,
  - Tree adders
  - Ling adder (reformulation of logic equations)
- Look over the circuit issues
  - Static CMOS
  - Pass transistor logic
  - Dynamic Logic
  - Number of stages of logic
  - Clever implementation of individual logic gates, or groups of logic gates

Circuit Design

Most people think about:

- Innovative configurations of transistors that perform some function better.
  - Where better might be smaller, faster, lower power, etc.

That is part of the job. The part that takes more time is:

- Making sure that this collection of transistors will work

OR

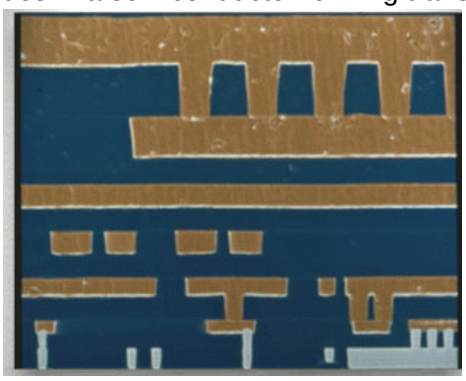
- Figuring out why this collection of transistors does not work, or only works on a few parts.

To do either, you need to be able to reason about circuits ...

The Problem

ICs are very complicated, contain many millions of tiny 3D structures

- Interleaving of conductors and insulators
- Diffusions of impurities in a semiconductor forming transistors



Could talk about the system in terms of 3D electric fields, and carriers

- But takes a long time for a computer to simulate a single transistor

The Solution: Models

Need to use some kind of model to simulate anything

- In the 3-D simulation has some models for how the electrons behave
  - This relies on other models for carriers in crystal structures
  - Maxwell's equations
- Simulating complex designs mean that the base models must account for more effects in each element you 'model'
  - Simulate transistors, not carriers
  - Assume wires are equipotentials (quastatic models)

To reason about a system (either you or a machine) one needs a model of it.

Models

Are an approximation of the real world

- Must leave many details out
- Must (to be useful) retain the important ‘details’
- Appropriate level depends on questions you want to answer

CAUTION:

- Simulation and analysis do not tell you what the circuit will do
- It tells you what your MODEL of the circuit will do
- So remember:

  Garbage in, garbage out

Some of the hardest work is figuring out the right model for a problem

Modeling / Simulation Problem

There are really two problems:

- Need to generate the correct model of the circuit
- Need to stimulate that circuit in ways that exercise the problem

  Add coupling noise the critical time

  Set initial conditions for the worst-case charge-sharing

  Inject substrate noise

SPICE limitation:

- Only evaluates the model of the circuit that you gave it
- Does the evaluation for the conditions you specify

  Answers the question you ask,

  But does not tell you whether it was the right question

What Model - The SPICE Approximation

Use lumped element model:

- Quasi-static (size is small compared to wavelength/4)
- Approximate devices by terminal characteristics
- Connections in the model are equipotentials

Major equation:

- $\left(\mathbf{i} = \mathbf{C} \cdot \frac{\Delta \mathbf{V}}{\Delta \mathbf{t}}\right) \rightarrow \Delta \mathbf{t} = \frac{\mathbf{C}\Delta \mathbf{V}}{\mathbf{i}}$ or $\Delta \mathbf{t} = \frac{\mathbf{Q}}{\mathbf{i}}$ charge control model

- So I need to model devices on the chip by their terminal iV and CV behavior - Figuring out the right model can be hard.

  (Sometimes need inductor equation too)

What Needs to be Modeled?

Transistors

- nMOS, pMOS

Wires

- They are not ideal connectors
- How complex?

  Resistance effects, iR drops in lines?

  Coupling, Inductance?

Circuit Environment

- Temperature, Power Supply, Substrate Voltage, Chip ‘Gnd’ vs. Board ‘Gnd’
  - Modeling of the package and power and ground distribution nets
  - Modeling of heat generation and flow

Appropriate model depends on question being asked