Electrical Engineering M216A
Design of VLSI Circuits and Systems

C.K. Ken Yang
University of California at Los Angeles
yang@ee.ucla.edu
Lecture Notes

The lecture notes will include some material not covered in the principle textbook and will be the primary course material. However, the level of details and description in the notes will not be as complete as the information that you would find in a textbook. I will also try to include additional information to help you understand the material.

Some of the notes used in this course are courtesy of M. Horowitz and D. Harris.
To provide additional information and/or an alternative explanation of the material in the notes, readings from other textbooks will be included in the notes. While these readings are not required, they are often helpful in understanding the material.

- Weste, Harris, *CMOS VLSI Design*
- Rabaey, *Digital Integrated Circuits: a System Perspective (2nd ed)*
- Glasser, Dobberpuhl, *The Design and Analysis of VLSI Circuits*
- Shoji, *CMOS Digital Design* (interesting collection of topics),
- Uyemura, *Circuit Design for CMOS VLSI;*
- Fabricius, *Introduction to VLSI Design.*
Course Information

• Background
  – This class will assume a background in digital logic, and some understanding of transistors and RC circuits. The class will also use a number of CAD tools that run on Windows and Unix workstations. You will learn how to use an layout/schematic editor Electric, a switch-level simulator Irsim, and the Verilog functional simulator. You will also use Synopsys logic synthesis tools and Silicon Ensemble place and route tool. Because of the number of CAD tools, you will only receive a brief introduction to them. Many of them will be scripted.

• Disclaimer:
  – This class provides an overview on the principles that govern VLSI design. It will introduce a number of possible design styles that you can explore further in other classes. If you are interested in more circuit design issues, take EE215B. If you are interested in CAD programs, take CS258D,E,F.
Lecture 1

Overview of VLSI: Complexity, Wires, and Switches

C.K. Ken Yang
yang@ee.ucla.edu

Courtesy of M. Horowitz
Stanford University
Overview

• Reading
  • Weste & Harris; 1.1-3, 1.5, 1.7-12
• Background
  - VLSI is a maturing field; it has its beginning back in the early 60's with SSI, small scale integration, when a few bipolar transistors and resistors were fabricated on the same chip. Today chips are both simpler and more complex. They typically only contain two active elements (nMOS and pMOS transistors) and wires. But there might be hundreds of millions of these transistors on the chip, and these chips can do amazing functions. You also find chips in everything. This lecture will look at why this has happened and what makes VLSI design challenging. It will also take a quick look at the basic elements that make up VLSI chips: MOS transistors and wires.
The Big Picture

- Want to go from this:
• To this:
Magnified
Why?

- Easier to move/control electrons than real stuff
  - Electronic calculators, not mechanical
  - Move information, not things (phone, fax, WWW, etc.)
- Building electronics:
  - Started with tubes, then miniature tubes
  - Transistors, then miniature transistors
  - Components were getting cheaper, but:
    - There is a minimum cost of a component (storage, handling …)
    - Total system cost was proportional to complexity
- Integrated Circuits changed that
  - Print a circuit, like you print a picture,
    - Create components in parallel
    - Cost no longer depended on # of devices
  - What happens as resolution goes up?
Moore’s Law

First stated by Intel’s Gordon Moore in the early 80’s. Saw that the resolution of the printing process was improving exponentially (0.7x feature size every 3 years) and predicted that it would continue into the future.

Since the cost of the printing process (called wafer fabrication) is growing at a slower rate, it implies that the cost per function, is dropping exponentially. At each new generations, each gate cost about 1/2 what it did 3 years ago. Shrinking an existing chip makes it cheaper!
Bad News

• Although the cost of manufacturing IC's remained approximately constant, the design cost did not. In fact, while designer productivity has improved with time, it has not increased at the same rate as the complexity of the chips.

• So the cost of the chip design is growing exponentially with the complexity of the circuit. The integrating of a system on a piece of silicon has an attractive manufacturing cost but frightening design cost and risk. Meanwhile, the market demands increasingly complex ICs.

• In addition, the number of custom IC designers was (and is) fairly limited. Even if you were willing to take the risk, where would you find the people to do the design?
Real Trend Plots

Die Size Trends

Transistor Per Integrated Circuit Trends

MOS Logic Process Trends

Sense of Scale

- What fits on a VLSI Chip today?
  - State of the art logic chip
    - 20mm on a side (400mm²)
    - ~65nm feature (drawn gate) length
    - <0.24μm wire pitch
    - >8-level metal
  - For comparison
    - 32b RISC processor
      - 8K \(\lambda\) x 16K\(\lambda\)
    - SRAM
      - about 32\(\lambda\) x 32\(\lambda\) per bit
      - 8K x 16K is 128Kb, 16KB
    - DRAM
      - 8\(\lambda\) x 16\(\lambda\) per bit
      - 8K x16K is 1Mb, 128KB

![Diagram showing the scale of a VLSI chip compared to a 32b RISC processor, 32b RISC Processor, and 64b FP Processor.]
Technology Scaling

- Number of ‘grids’ per chip doubles every 3 years
  - more functionality per chip
  - harder to design
- Two problems
  - What do you do with all that space -- what function?
  - How do you make sure it works
The Challenge in VLSI Design – Managing Complexity

• Simplify the design problem
  • Can’t understand 10M transistors, or 100M rectangles
  • Need to make less complex (and less numerous) models
    – Abstraction
      • Simplified model for a thing, works well in some subset of the design space
    – Modeling Constraints
      • Needed to ensure that the abstractions are valid
      • Might work if you violate constraints, but guarantees are off

• Understand the underlying technology
  – Provide a feeling for what abstractions and constraints are needed.
  – Determine efficient solutions (make the right tradeoffs).

• CAD tools use the abstractions and constraints to help us manage the complexity.
  – They do not replace the need to understand the technology.
  – In fact, we now need to understand how tools work.
Reality of VLSI Design – Juggling Tradeoffs

• Bottom line is $$$$ 
• To the VLSI designer, the external “constraints” and issues are multi-dimensional.

\[ \text{Performance} \]

- Portables (power - performance/area)
- DRAM (area - features/performance)
- DSP (design time/area - performance)
- Military (robustness - power/performance)
VLSI Design

Besides all that,

I think it is fun.

I hope you agree.
What is on an Integrated Circuit?

• Actually only two types of “devices”:
  – Conducting layers which form the wires on the IC.
    • There are many layers of wires (used to have 1 layer of metal, now advanced processes have 5-10 metal layers). Wires have electrical properties like resistance and capacitance.
    • (Requires insulators and contacts between layers.)
  – Transistors (the free things that fit under the wires).
    • There are a few kinds of transistors. In this class we will study MOS ICs, so we will work with MOS transistors. These transistors can be thought of as a voltage controlled switch. The voltage on one terminal of the transistor determines whether the other two terminals are connected or not.
Physical Topology of an Integrated Circuit

- The transistors are built in the silicon, and then there are lots of wiring layers deposited on top. In cross-section it looks like (abstractly):

  **Many** more metal layers

In the technology that we will use in the class (which can be scaled from $2\mu$ to $0.13\mu$) there are 7 primary layers. The top five layers are metal wires, and then there is a polysilicon layer and a diffusion layer (together poly and diff can form “active” devices – more on that later).
- Another View:

- Chip consists of
  - *transistors*: fabricated on the silicon surface and
  - *wires*: that connect the transistors fabricated on layers of metal separated by insulators
- Most of the area are the wires
Transistors

The voltage on the gate (poly connection) controls the current that flows between the source and drain (diffusion terminals). The transistor model is often displayed by drawing its current-voltage curve. We will talk about more later.
A MOSFET as a Switch

- Three terminal device
  - source, drain
    - two ends of conductive path
  - gate
    - controls conductive path
  - operation
    - conducts when gate is high
    - open circuit when gate is low
  - caveat
    - passes 0s well, not 1s

This description is for nMOS transistors. For pMOS everything is reversed. The source is the higher voltage terminal, and the transistor is on when the gate is much lower than the source. More on pMOS later.
Switch Networks

• Since transistors can be modeled as switches
  – Draw an abstract switch as
    • Control (gate) terminal is on top

• We can build switch networks between two non-control terminals
  – Define function of a switch network as the inputs conditions that connect the two terminals of the network
  – Structure of switch network sets its logic functions:
    • ‘OR’ functions are constructed by parallel switches
    • ‘AND’ function are constructed by series switches
  – Not quite logic “gates” (later)
    • But are components that build a gate.

• A VLSI chip is comprised of millions of these switch networks.
The end of the design process must create a set of drawings, one for each layer needed in the manufacturing process

- **Layout drawings are complicated**
  - There are many rules about the geometry to make sure the circuits can be reliably manufactured
    - Minimum width of wire, minimum spacing between wires, alignment rules
  - The layers represent transistors and wires, and need to create the correct function
  - Many rectangles for each transistor and wire, and there are millions of transistors and wires.
- **Different layers are represented by different colors**
  - People used to draw the layout on mylar (10s of transistors)
  - But not any more, now use CAD tools (to help with abstraction, visualization, and constrain to design rules), and pre-made cells (for partitioning, hierarchy)
Simple Layout Example of a Simple Processor

- Example from old student project
- Use hierarchy to hide complexity
- Pads around chip
- Major blocks are shown
  - Design is broken into a controller that controls dataflow.
- Colored regions are really many wires
Layout

• This picture is an expanded view of a portion of the layout of the other page.
• The next two slides:
  – Controller layout
    • Handle instruction inputs
  – Datapath layout
    • Moving data around
Controller Layout

- Right half shows cells in the design
- Left half has the cells expanded to show the layout layers
- This design style has random wires
Datapath Layout

- Wires here are more regular.
  - Words are 16bits wide.
  - Each path is repeated 16x
- Again
  - Cells on right
  - Expanded cells on left
- Transistor density is higher
A Slightly More Powerful Processor

• By converting rectangles into transistors, transistors into gates, gates into functions, and functions into an architecture, we result in something quite remarkable.
• Pentium 4

2.2cm
Abstractions and Disciplines
How to Deal with $10^8$ Transistors

- Digital abstraction
  - signals are 1 or 0
- Switch abstraction
  - MOSFETs as simple switches
- Gate abstraction
  - Unidirectional elements
  - Separable timing
- Synchronous abstraction
  - Race free logic
  - Function does not depend on timing

- Constrain the design space to simplify the design process
  - strike a balance between design complexity and absolute performance

- Partition the problem
  (Use hierarchy)
  - Module is a box with pins
  - apply recursively
Design Levels

• **Specification**
  – what the system (or component) is supposed to do

• **Architecture**
  – high-level design of component
    • state defined
    • logic partitioned into major blocks

• **Logic**
  – gates, flip-flops, and the connections between them

• **Circuit**
  – transistor circuits to realize logic elements

• **Device**
  – behavior of individual circuit elements

• **Layout**
  – geometry used to define and connect circuit elements

• **Process**
  – steps used to define circuit elements

Can describe design at many different levels of abstraction

High-lighted levels we will discuss in this class
Design Procedure and Tools

- Concept
  - divider

- Architecture
  - subtract/compare

- Logical Implementation
  - ab+bc+ac
  - xor

- Circuit Implementation
  - transistors

- Physical layout + Verify
  - mask layers (rectangles)

- C-modelling

- Behavior modelling
  - Verilog or VHDL

- Logic Synthesis
  - Design Analyzer (Synopsys)
  - Verify Synthesis
    - Static Timing

- Place and Route
  - Silicon Ensemble (Cadence)
  - Verify P&R
    - Dynamic Timing

- Layout
  - Electric
A More Realistic Design Flow

Wire Model

3-D RLC Modeling Tool

ρ, σ, μ
Layers
Layout rules

Parasitic Extraction Library

Standard Cell Library

Device model

Schematic Entry

Layout Entry

Cell Characterization

Synthesis Library (Timing/Power/Area)

Place & Route Library (Ports)

C-Model

Verilog Behavioral Model

Functional

Verilog Structural RTL

Functional Static Timing

Block Layout

Floorplan

DRC/ERC/LVS
Static/Dynamic Timing w/ extract
Power/Area
Scan/Testability
Clock Routing/Analysis

Global Layout

Floorplan

Synthesis

Structural Model

Block Layout

P & R

P & R
M216A Goals

• Understanding the basic building blocks of VLSI.
  – Transistors/Wires
  – Logic Gates and Layout
  – Datapath Blocks
• Be able to conceptually model system.
  – Logic Optimization
  – State Machine Design (RTL)
• Be able to build a system (using a subset of the tools)
  – Verilog Modeling
  – Synthesis
  – Place and Route
• Understanding the constraints and tradeoffs
  – Delay analysis (gates and interconnects)
  – Clocking methodology.
  – System integration issues (Power/Ground routing, Noise)