



**EECS 151/251A**

**Spring 2018**

**Digital Design and Integrated  
Circuits**

Instructors:

Nicholas Weaver & John Wawrzynek

**Lecture 1**



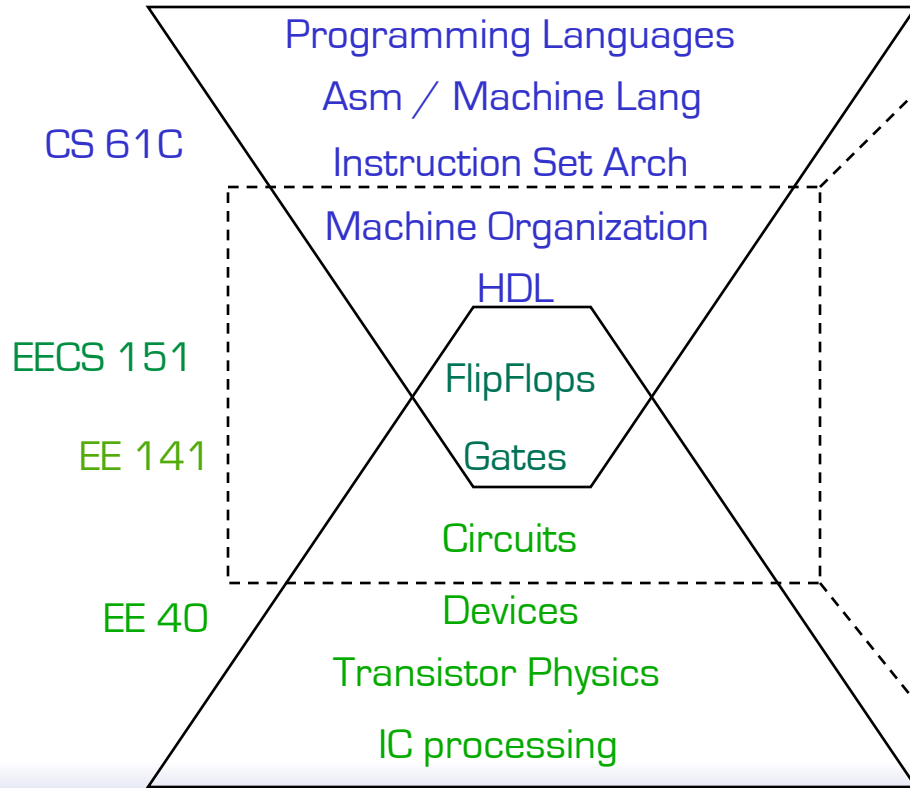
## **Class Goals and Outcomes**

# What this class is all about?

- ❑ **Introduction to digital integrated circuit and system engineering**
  - Key concepts needed to be a good digital designer
  - Discover you own creativity!
- ❑ **Learn models that allow reasoning about design behavior**
  - Manage design complexity through abstraction and understanding of tools
  - Allow analysis and optimization of the circuit's performance, power, cost, etc.
- ❑ **Learn how to make sure your circuit and system works**
  - *Do you want your transistor/block to be the one that screws up a 1 billion transistor chip?*

*Digital design is not a spectator sport!  
Learn by doing.*

# Course Focus



## Deep Digital Design Experience

Fundamentals of Boolean Logic

Synchronous Circuits

Finite State Machines

Timing & Clocking

Device Technology & Implications

Controller Design

Arithmetic Units

Memories

Testing, Debugging

Hardware Architecture

Hardware Design Language (HDL)

Design Flow (CAD)



**ADMINISTRATIVA**

# EECS151 Crew



**Dr. Nicholas Weaver**

329 Soda Hall

[nweaver@icsi.berkeley.edu](mailto:nweaver@icsi.berkeley.edu)

Office Hours:

M 1-3pm & by appointment  
& just drop by if my door  
is open



**Professor John  
Wawrzynek (Warznek)**

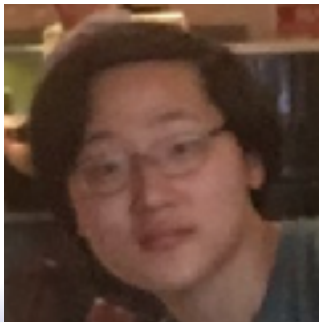
631 Soda Hall

[johnw@berkeley.edu](mailto:johnw@berkeley.edu)

Office Hours:

Tu,Th 2:30PM, & by  
appointment.

**Taehwan Kim**



**Arya Reais-Parsi**

*All TA office hours held in 125 Cory. Check website for days and times.*

# Course Information

- Basic Source of Information, class website:

<http://inst.eecs.berkeley.edu/~eecs151/sp18/>

- Lecture notes and video modules
- Assignments and solutions
- Lab and project information
- Exams
- Piazza Discussion Forum
- Many other goodies ...



Print only what you need: Save a tree!

# *Class Organization*

- ❑ Lectures
- ❑ Discussion sessions
- ❑ Office hours
- ❑ Problem Sets
- ❑ Labs – FPGA or ASIC or both
- ❑ Design project
- ❑ 2 Midterms + 1 Final



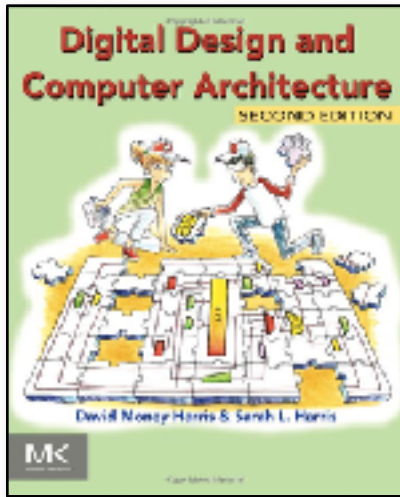
# Lectures

- ❑ Slides available on website before the lecture
- ❑ Lectures are NOT videotaped, but ... videos of Spring 2016 are available and will be posted on website

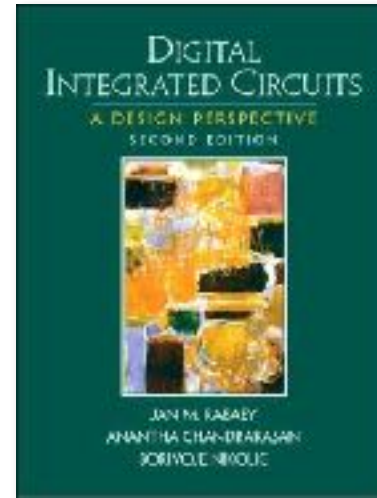
[https://www.youtube.com/playlist?list=PLkFD6\\_40KJIwtkVBMW61fI9cwzOk\\_ruah](https://www.youtube.com/playlist?list=PLkFD6_40KJIwtkVBMW61fI9cwzOk_ruah)



# Class Textbooks



*Required*



*Recommended*

□ Useful LA lab reference (EE151/251A):

- Erik Brunvand: Digital VLSI Chip Design with Cadence and Synopsys CAD Tools

# *Discussions*

- Not recorded – but material posted
- F 11:00AM-12 | 106 Moffitt Library
- Lead GSI: Arya Reais-Parsi
  - Only one may be viable given class size and GSI assignment

# *Problem Sets*

- ❑ Approximately 10 over the course of the semester
- ❑ Posted on Thursday, due on Friday the next week at 11:59pm
- ❑ Essential to understanding of the material
  - Hence ...
- ❑ Ok to work with colleagues but need to turn in your own work / write-up



# Labs

- ❑ Choose either FPGA or ASIC or both
- ❑ 7 FPGA / 7 ASIC labs
- ❑ Lab report due by next lab session
- ❑ Lab times – all in 125 Cory
  - ASIC: W 5-8PM (Taehwan)
  - FPGA: W, Th 2-5PM (Arya)
  - All labs start next Week

# Midterms and Final

- ❑ Midterms scheduled in class plus extra time. No lecture that day. Special review session in advance.
  - Midterm 1: Th Feb 15 – 5pm - 8pm
  - Midterm 2: Th Mar 22 – 5pm - 8pm
- ❑ Final: Fri May 11, 11:30-2:30PM



*All exams are open book – it is understanding that matters!*

# Course Information

- ❑ For interactions between faculty, GSIs and fellow students – we are using Piazza

For fastest response **post your questions on Piazza.**



(make sure to register ASAP if you don't want to miss any of the action)  
<http://piazza.com/berkeley/spring2018/eecs151251a>

# *Snippets of Knowledge*

- ❑ Can work together on homework
  - But you must turn in your own solution
- ❑ Lab reports due in lab 1 week after the lab session
- ❑ Labs done solo
- ❑ Project is done in groups of 2
- ❑ No late assignments
  - Solutions available shortly after due date/time
- ❑ Don't even think about cheating!



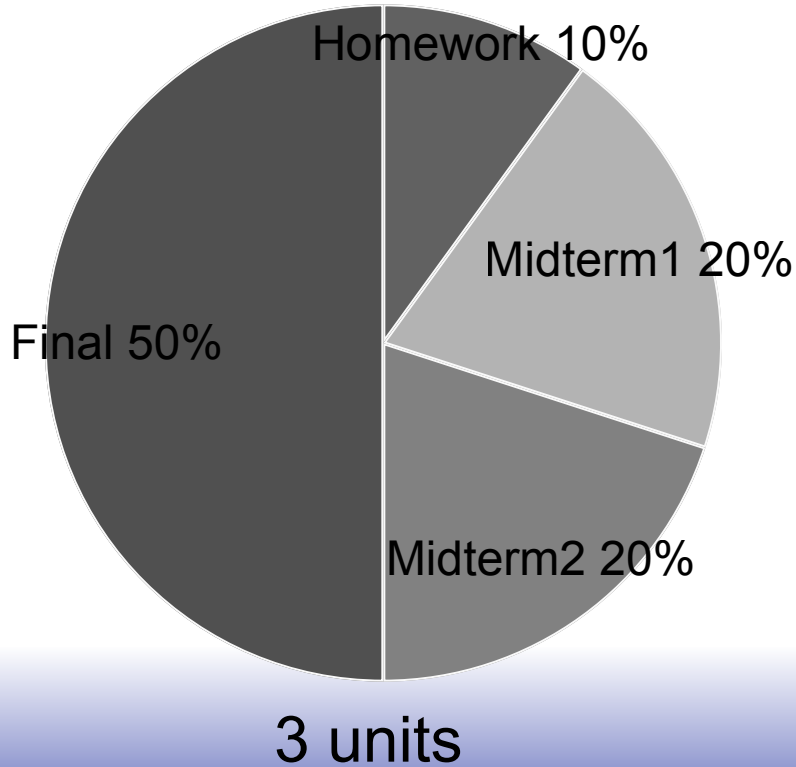


# Cheating

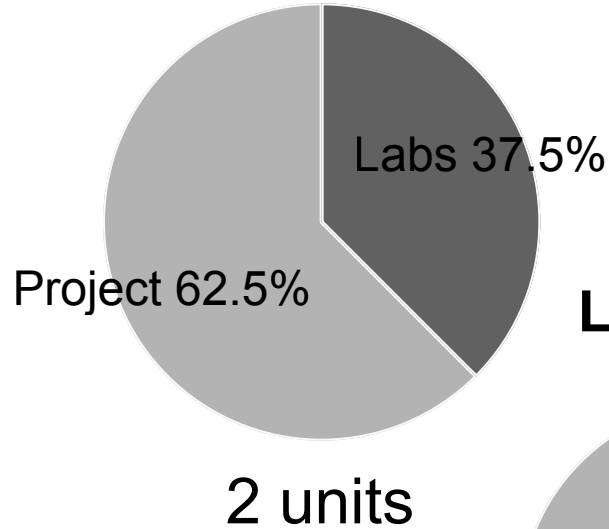
- Details of our cheating policy on the class web site. Please read it and ask questions.
- If you turn in someone else's work as if it were your own, you are guilty of cheating. This includes homework sets, answers on exams, Verilog code, block diagrams, etc.
- Also, if you knowingly aid in cheating, you are guilty.
- We have software that compares your submitted work to others.
- However, it is okay to discuss with others lab exercises and the project. Okay to work together on homework. But everyone must turn in their own work.
- Do not post your work on public repositories like github (private o.k.)
- **If we catch you cheating, you will get **negative points** on the assignment: It is better to not do the work than to cheat!**  
If it is a midterm exam, final exam, or final project, you get an F in the class.  
All cases reported to the office of student conduct.

# Grading Breakdown

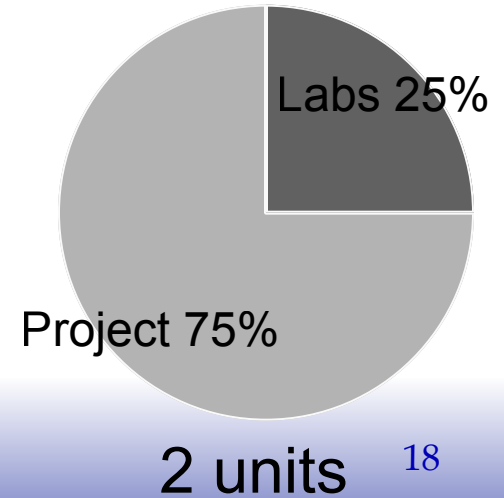
## EECS 151/251A Lecture



## LA – ASIC



## LB – FPGA



# *Tips on How to Get a Good Grade*

The lecture material is not the most challenging part of the course.

- You should be able to understand everything as we go along.
- Do not fall behind in lecture and tell yourself you “will figure it out later from the notes or book”.
- Notes will be online before the lecture (usually the night before). Look at them before class. Do assigned reading (only the required sections).
- Ask questions in class and stay involved in the class - that will help you understand. Come to office hours to check your understanding or to ask questions.
- Complete all the homework problems - even the difficult ones.
- The exams will test your depth of knowledge. You need to understand the material well enough to apply it in new situations.

You need to enroll in both the lab and the course.

- Take the labs very seriously. They are an integral part of the course.
- Choose your partner carefully. Your best friend may not be the best choice!
- Most important (this comes from 30+ years of hardware design experience):
  - Be well organized and neat with homework, labs, project.
  - In lab, add complexity a little bit at a time - always have a working design.
  - Don't be afraid to throw away your design and start fresh.

# *Getting Started*

- ❑ Discussions and labs start next week
- ❑ HW 1 assigned later this week

# Course Accounts

- ❑ A new online system to get account:
  - go to <http://inst.eecs.berkeley.edu/webacct>,  
or
  - click "WebAcct" on <http://inst.eecs.berkeley.edu>.
- ❑ Need Calnet login to get account and password immediately
- ❑ All enrolled and waitlisted students are included. Will need SIDs for others (e.g. CE).



# Digital Integrated Circuits and Systems – From The Past to the Future ...

# Electronics all around us



**Consumer Products**



**Aerospace and Military**



**Toys**



**Communications Infrastructure**

**Automotive**



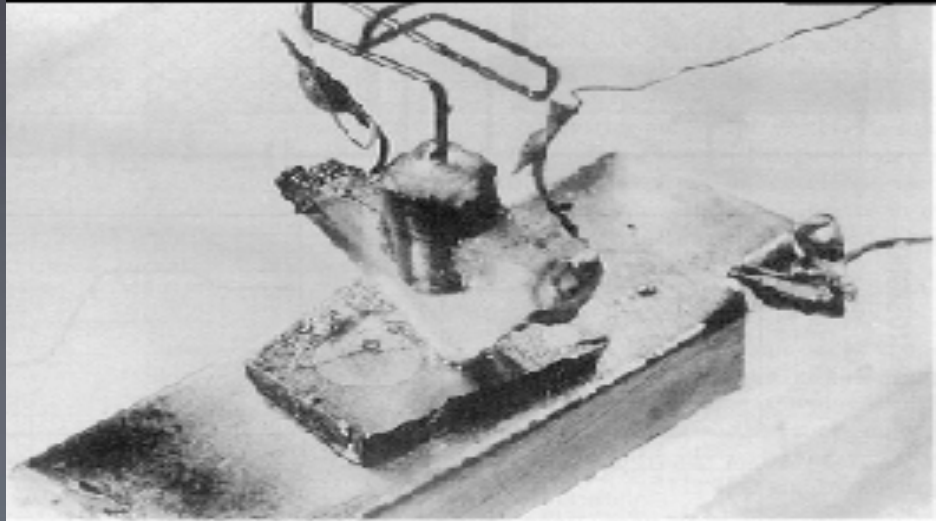
*And then plenty more ...*





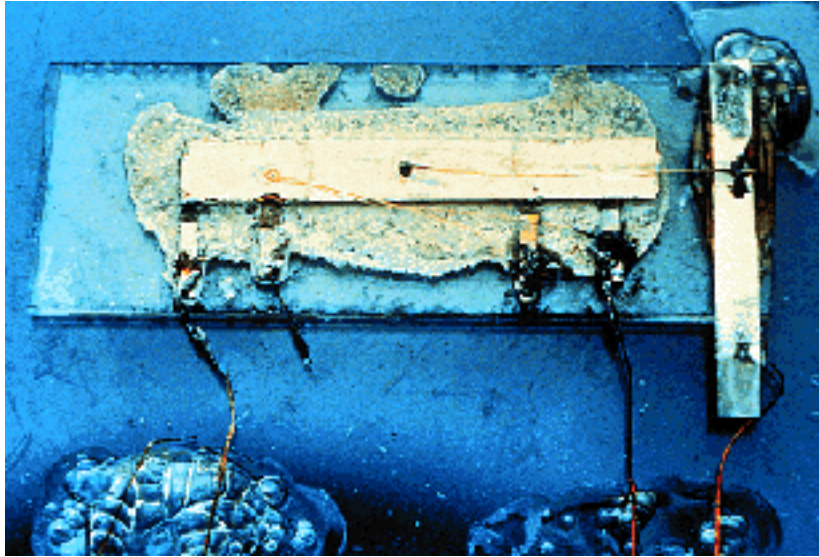
*How did this all arise?*

# *The Transistor Revolution*



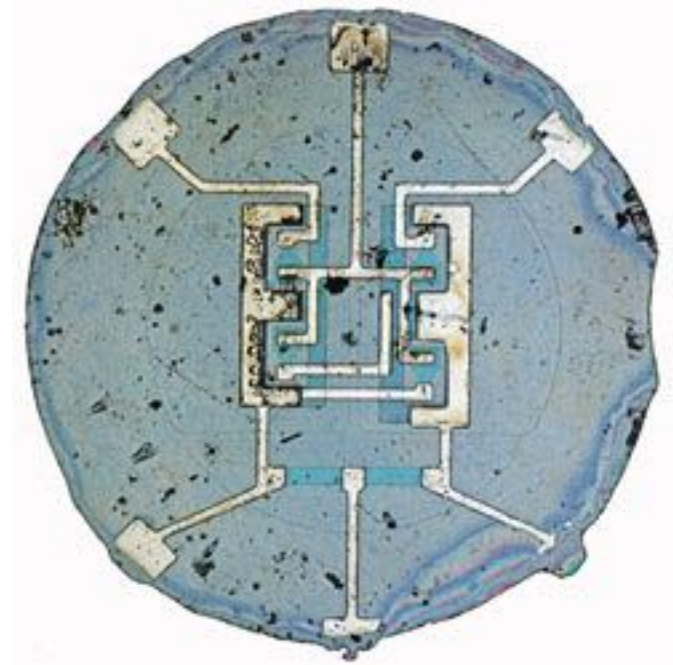
First transistor  
Bell Labs, Dec 1947

# First Integrated Circuits (1958-59)

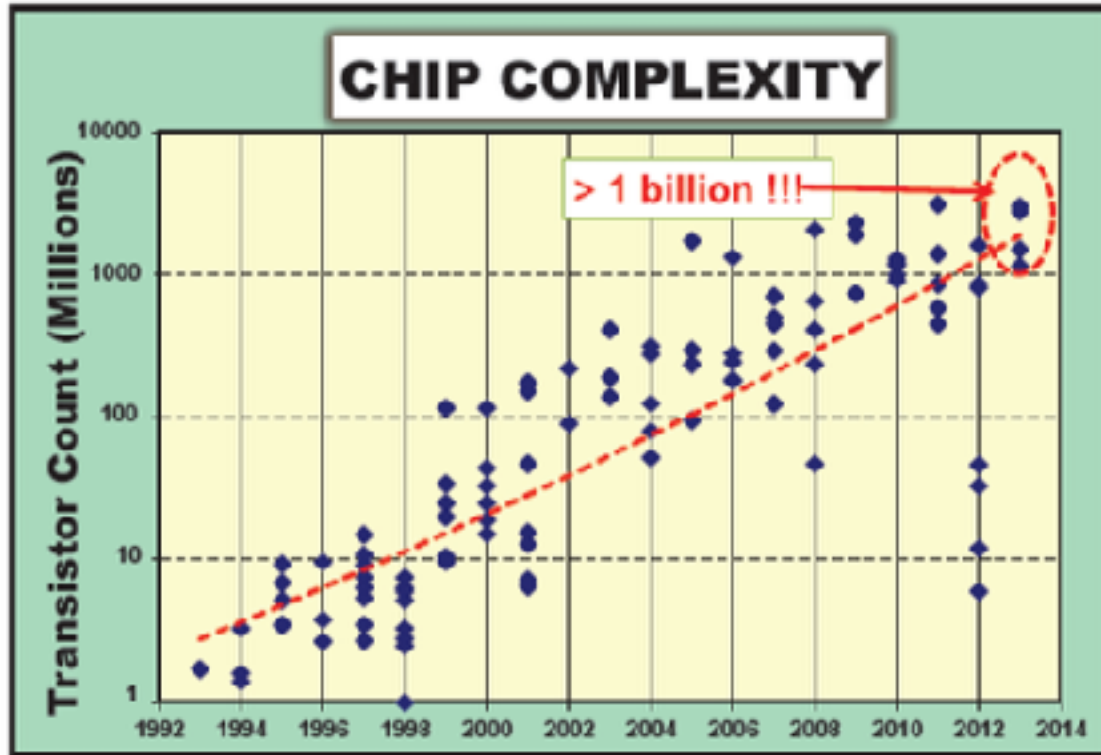


*Jack Kilby, TI*

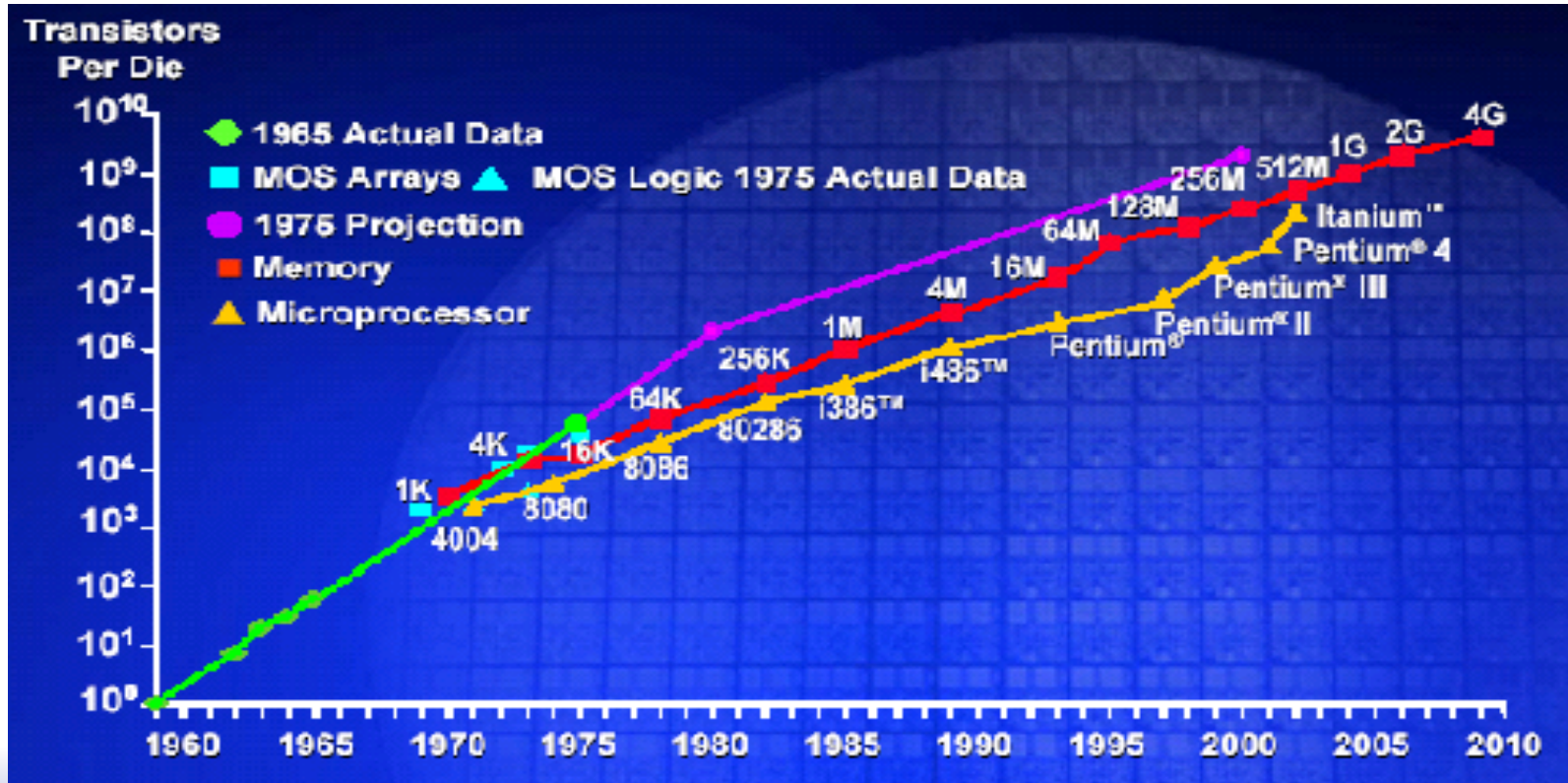
*Bob Noyce, Fairchild*



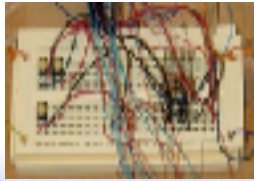
# Transistor Counts



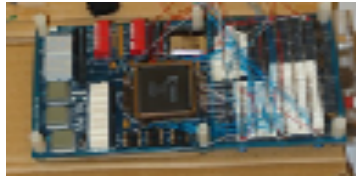
# Moore's Law – 2x stuff per 1-2 yr



# CS150 Project Complexity



1980 Pong game  
10's of logic gates



1995 MIDI synthesizer  
1000's of logic gates



2000-2010 eTV tuner  
10K's logic gates

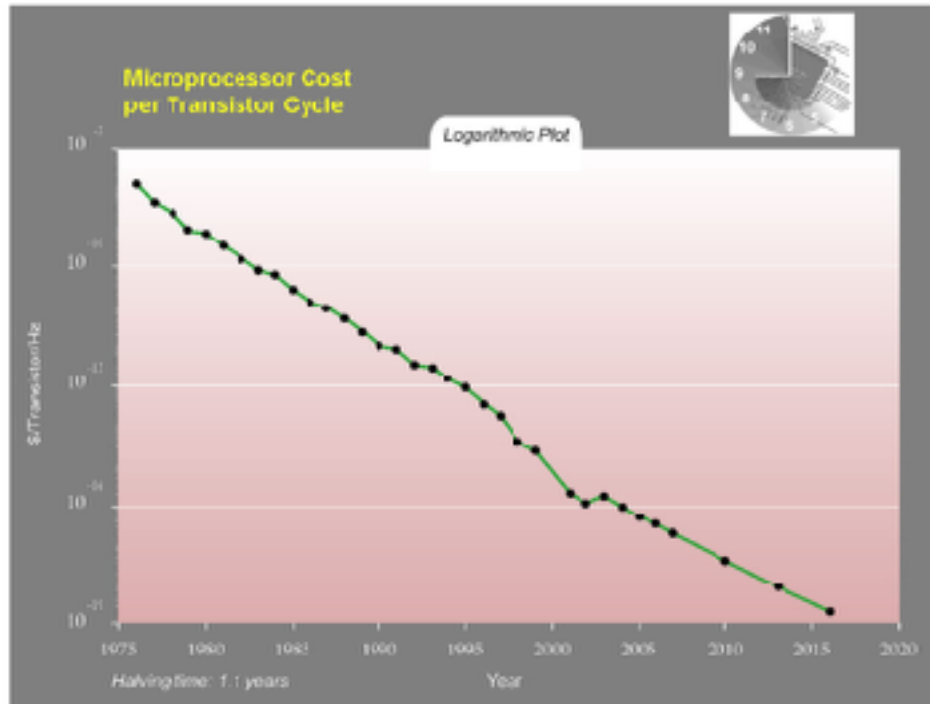


2010-2017 MIPS CPU or BYO  
1M logic gates



2018 MIPS CPU  
Programmable SOC:  
dual-core ARM, 85K  
logic cells, 220 MACC

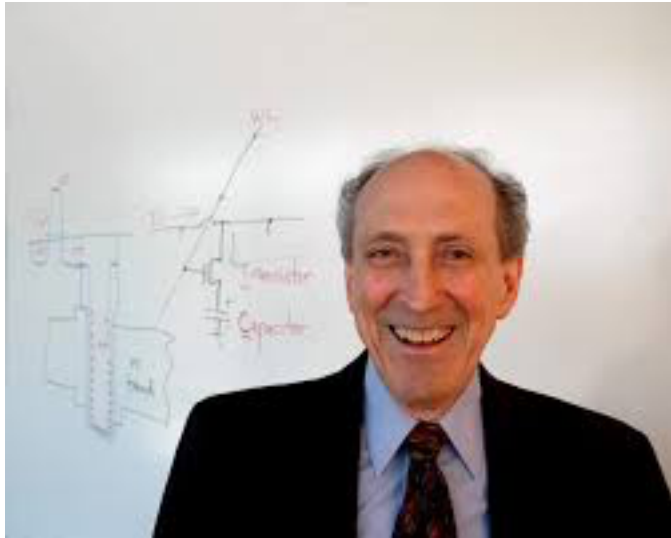
# *The most important outcome: cost*



# *The other outcome*

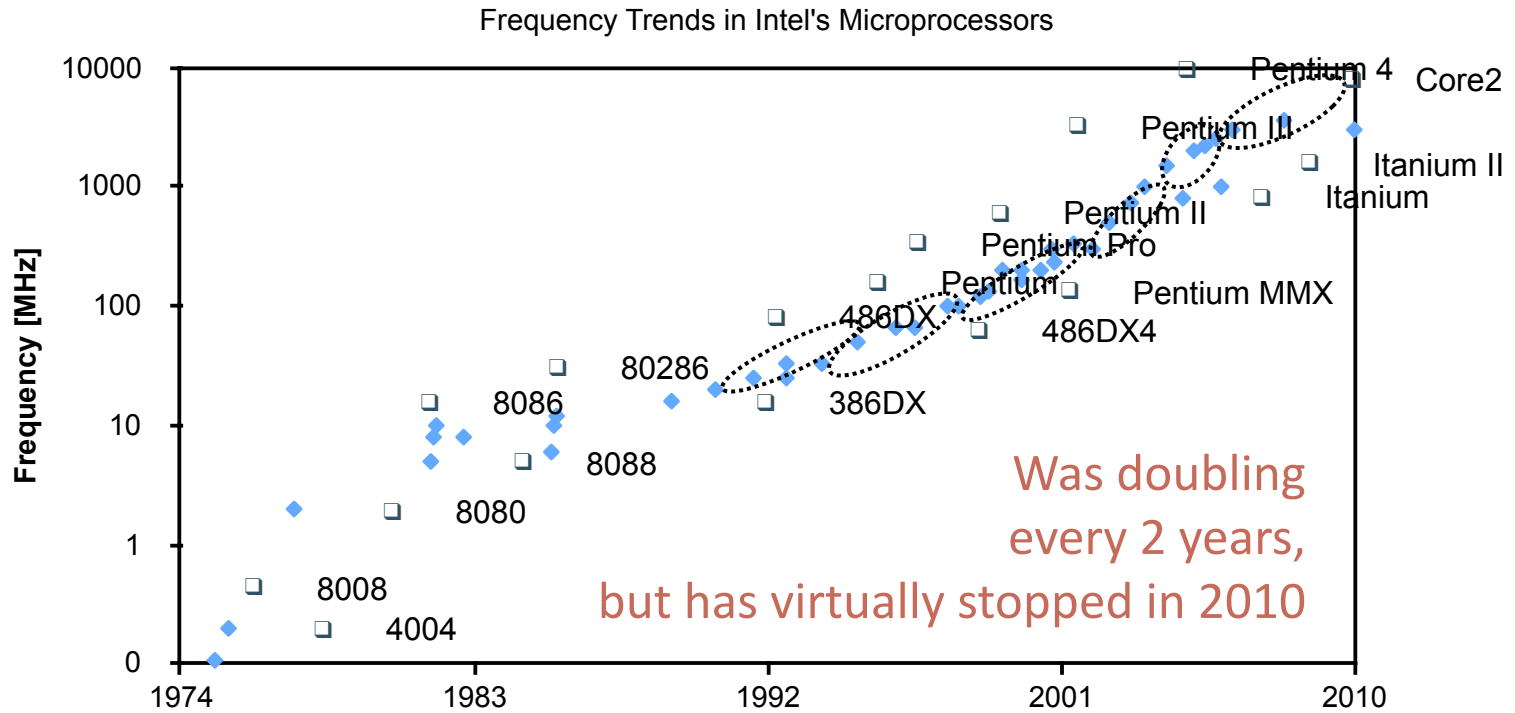


# Dennard Scaling (1974)



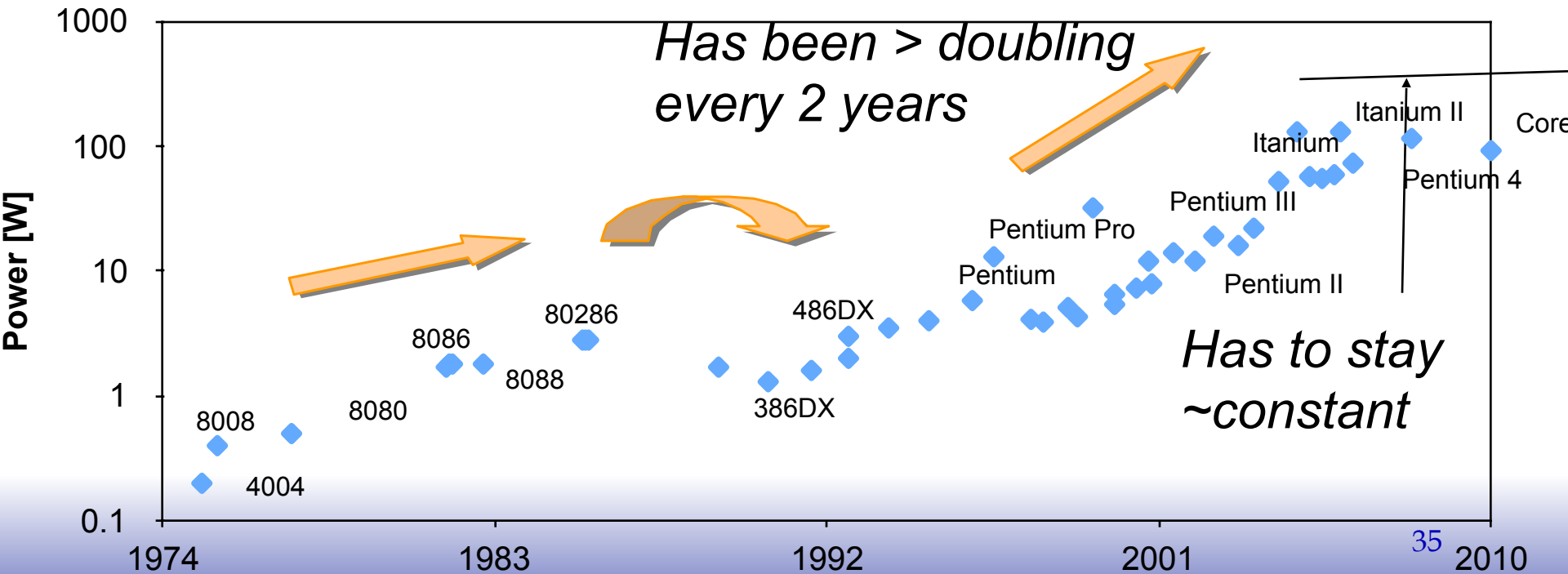
- ❑ Voltages and currents should be scaled proportional to the dimensions of the transistor
- ❑ If so, delay and power should scale as the technology
  - Delay  $\sim C \cdot V / I_{av}$
  - $P \sim C V^2 / \text{Delay}$

# Frequency

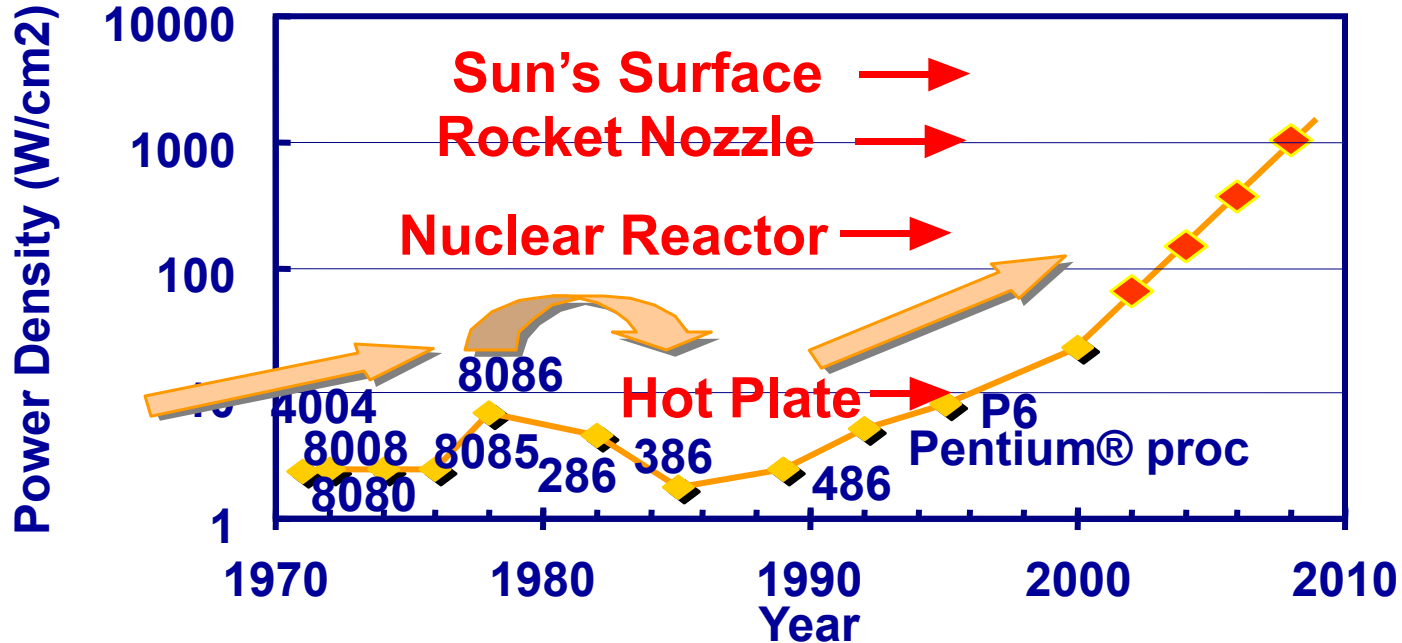


# Power Dissipation

Power Trends in Intel's Microprocessors



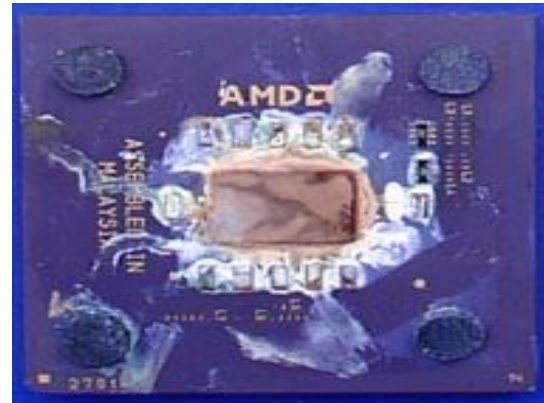
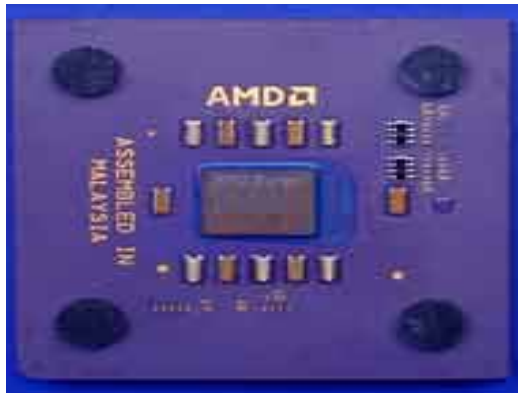
# Cause: Power Density



S. Borkar

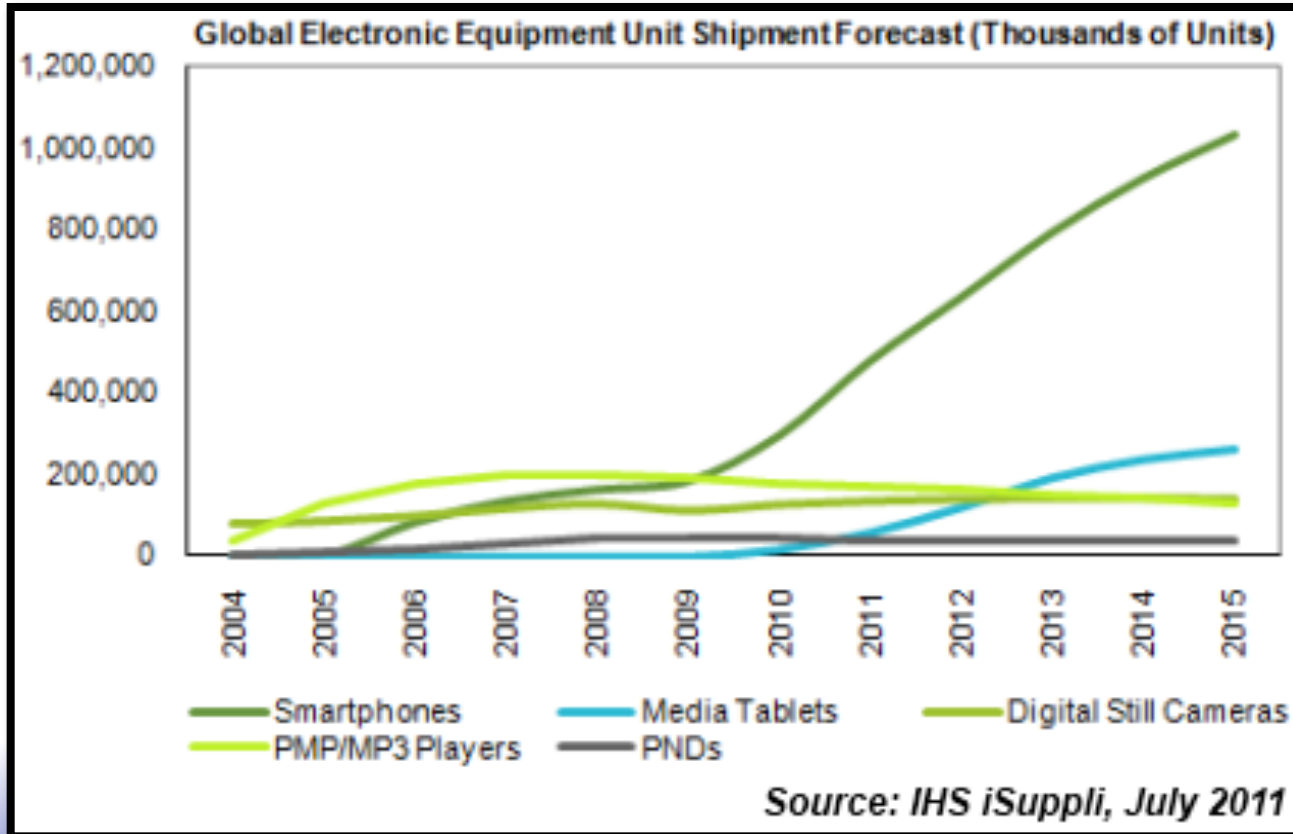
Power density too high for cost-effective cooling

# Not enough cooling...



\*Pictures from [http://www.tomshardware.com/2001/09/17/hot\\_spot/](http://www.tomshardware.com/2001/09/17/hot_spot/)  
"We know computers really run on *magic smoke*,  
because when the magic smoke escapes the chip stops working"

# Way Beyond Microprocessors



# *Some quick questions*

- ❑ True or False: Moore's law allows us to predict future properties in the same way that the law of gravity allows us to predict the path of a planet around the sun.

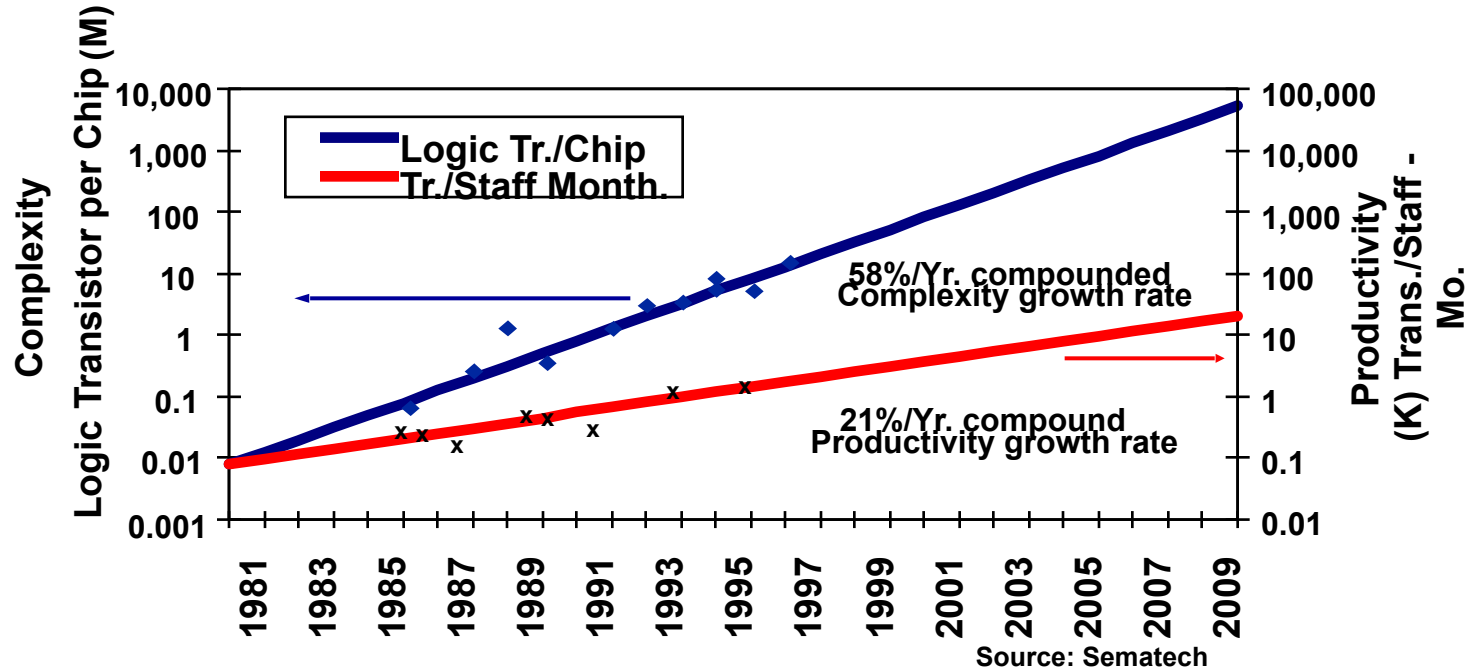
# *Some quick questions*

- ❑ True or False: Moore's law says that computer performance doubles every 18-24 months



# *The other Demon: Complexity*

# Complexity and Productivity Trends



**Complexity outpaces design productivity**

Courtesy, ITRS Roadmap

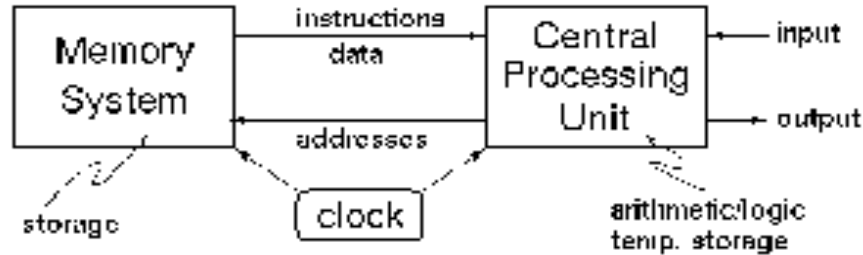
# *The answer*

- Design methodology!
  - Abstraction
  - Hierarchy
  - Reuse

# *Digital System Design: A few basic concepts*

# Example Digital Systems

- General Purpose Desktop/Server Digital Computer
  - Often designed to maximize performance. "Optimized for speed"



- Handheld Calculator



- Usually designed to minimize cost.  
"Optimized for low cost"

- Of course, low cost comes at the expense of speed.

# Example Digital Systems

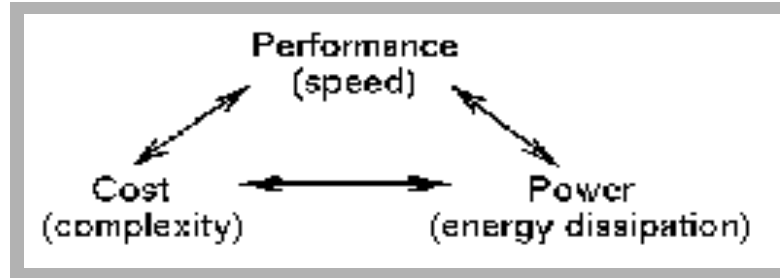
## □ Digital Watch



*Designed to minimize power.  
Single battery must last for years.*

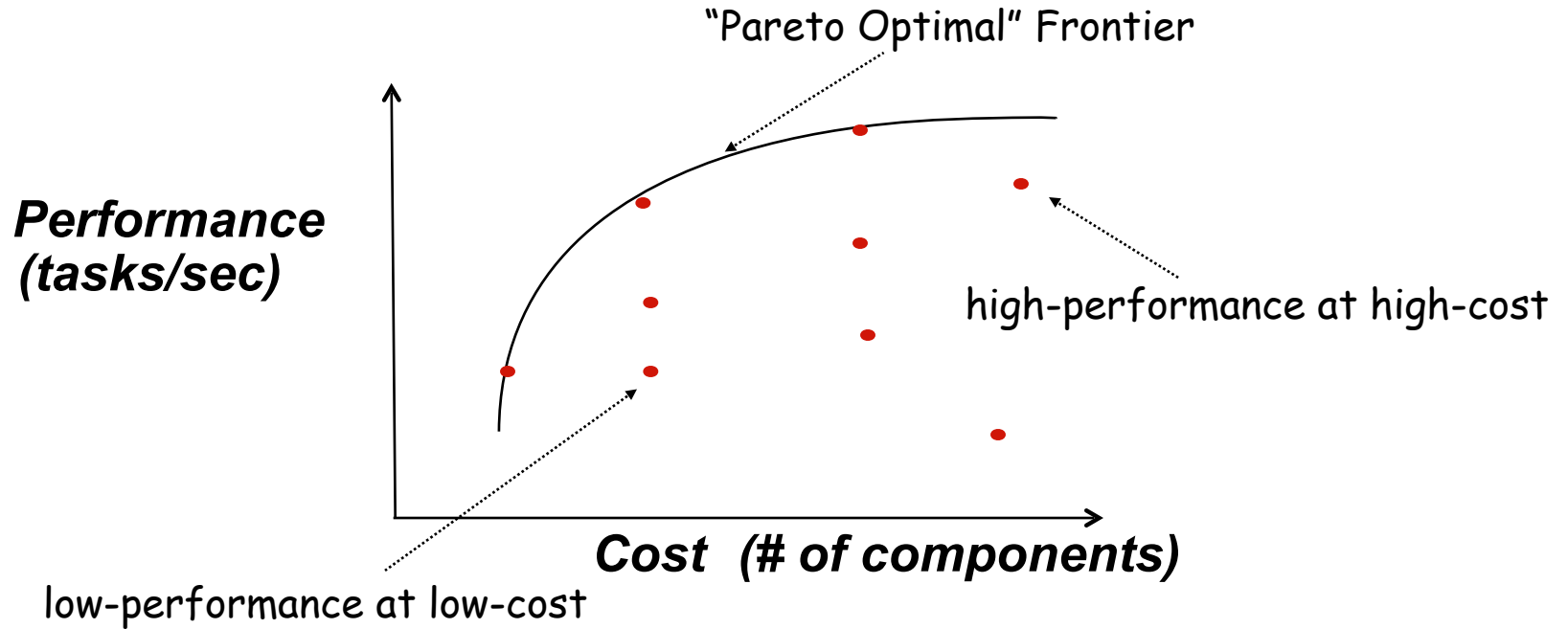
- Low power operation comes at the expense of:
  - lower speed
  - higher cost

# Basic Design Tradeoffs



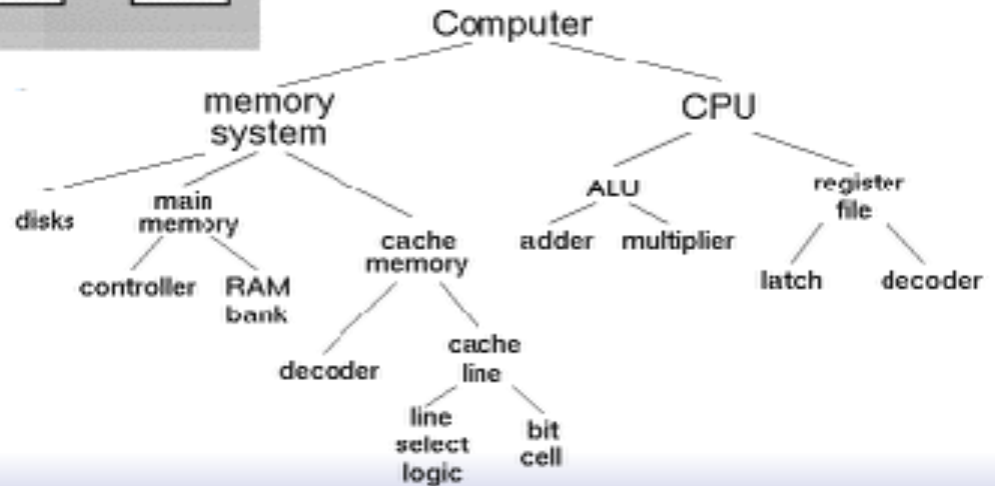
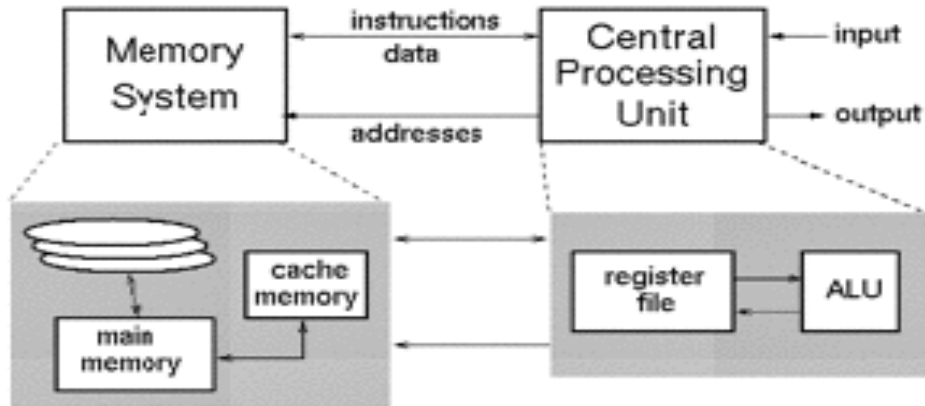
- Improve on one at the expense of the others
- Tradeoffs exist at every level in the system design
- Design Specification
  - Functional Description
  - Performance, cost, power constraints
- Designer must make the tradeoffs needed to achieve the function within the constraints

# Design Space & Optimality





# Hierarchy & Design Representation



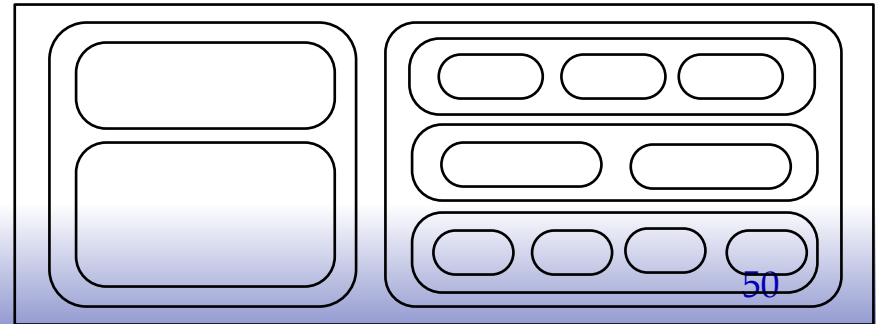
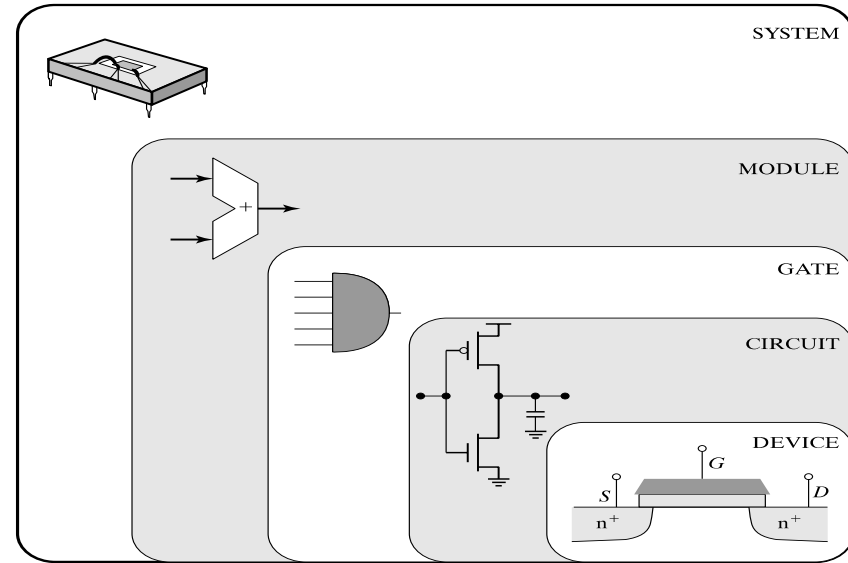
# Hierarchy in Designs – Complexity Control

## □ Design Abstraction

- Hide details and reduce number of things to handle at any time

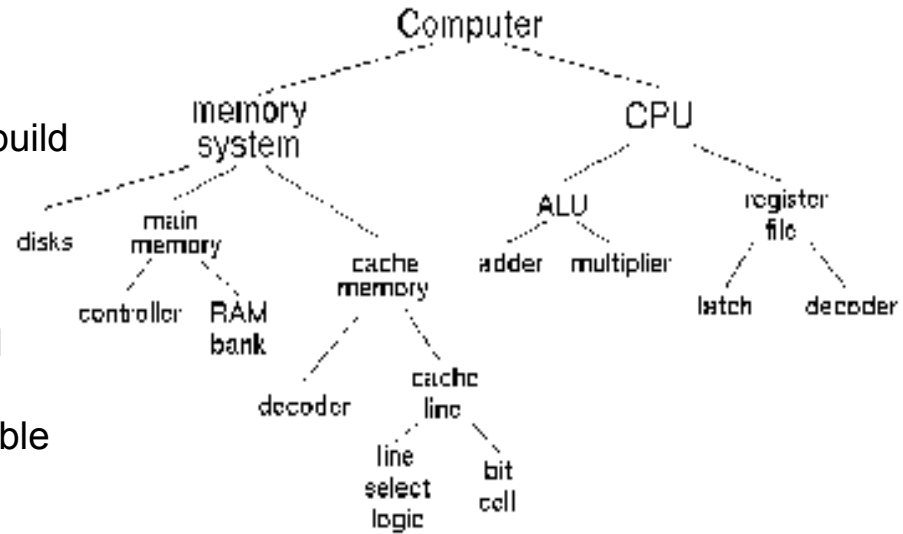
## □ Modular design

- Divide and conquer
- Simplifies implementation and debugging



# Design Methodologies

- Top-Down Design
  - Starts at the top (root) and works down by successive refinement.
- Bottom-up Design
  - Starts at the leaves & puts pieces together to build up the design.
- Which is better?
  - In practice both are needed & used
  - Top-down to handle the complexity (divide and conquer)
  - Bottom-up since structure influenced by available primitives (in a well designed system)



# Digital Design: What's it all about?

Given a functional description and performance, cost, & power constraints, come up with an implementation using a set of primitives.

- How do we learn how to do this?
  1. Learn about the primitives and how to use them.
  2. Learn about design representations.
  3. Learn formal methods and tools to manipulate the representations.
  4. Look at design examples.
  5. Use trial and error - CAD tools and prototyping. Practice!
- Digital design is in some ways more an art than a science. The creative spirit is critical in combining primitive elements & other components in new ways to achieve a desired function.
- However, unlike art, we have objective measures of a design:

*Performance Cost Power*

# *Next Lecture*

- Design flow and substrates
  - RTL design abstraction
  - FPGA fabric overview
  - ASIC std-cell overview
  - Design tools (synthesis, p&r, verification)