EECS 151/251A
Spring 2019
Digital Design and Integrated Circuits

Instructor:
John Wawrzynek

Lecture 1

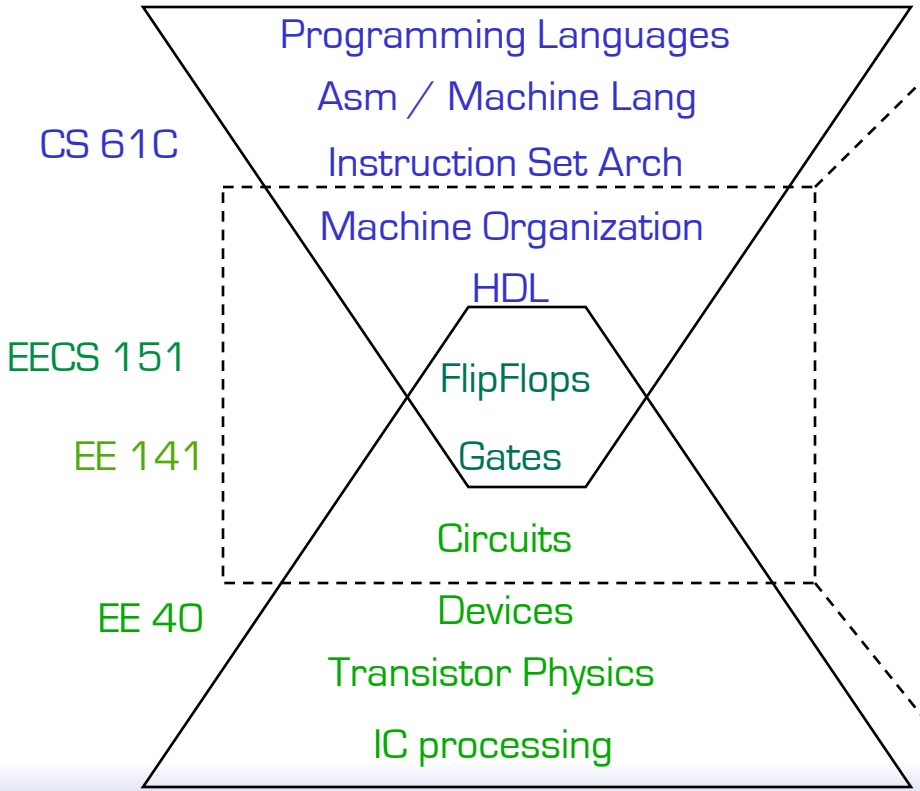
Class Goals and Outcomes

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)

ADMINISTRATIVIA

EECS151/251A Crew

Professor John Wawrzynek (Warznek)
631 Soda Hall
johnw@berkeley.edu
Office Hours:
Thur 2:30PM, & by appointment.

Reader: Prashanth Ganesh

Chris Yarp
Discussion Sections, FPGA Labs
Office Hours:
Tu 2:30, We 12:45

Arya Reais-Parsi
ASIC Labs
Office Hours: TBD

All TA office hours held in 125 Cory.

Course Information

- Basic Source of Information, class website:

http://inst.eecs.berkeley.edu/~eecs151/sp19/

- 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
- 1 Midterm + 1 Final

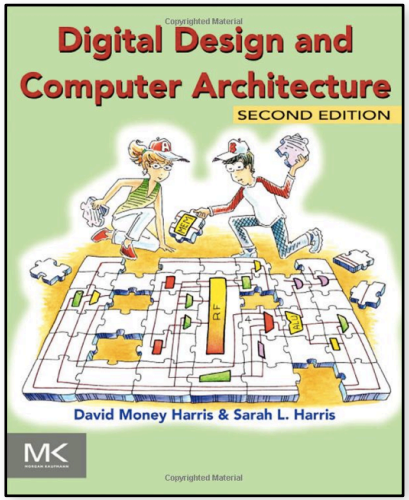
Lectures

- Slides available on website before the lecture
- Lectures are NOT videotaped, but …

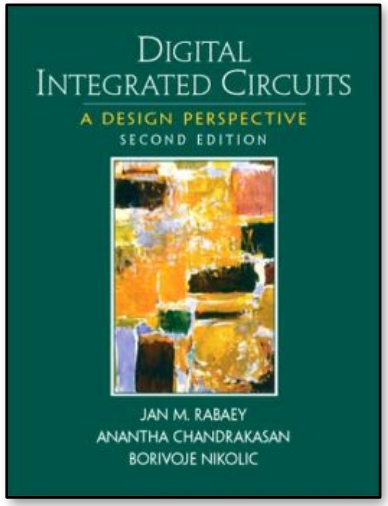
videos of Spring 2016 version of the class are available and will be posted on website

Class Textbooks

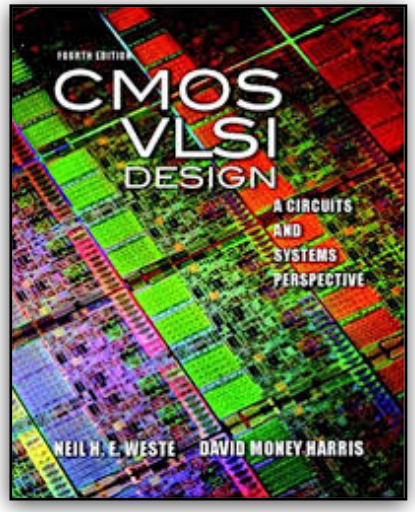
No Required Book this semester



Recommended (previously required)



Recommended



Useful

- Useful LA lab reference (EE151/251A):
  - Erik Brunvand: Digital VLSI Chip Design with Cadence and Synopsys CAD Tools

Discussion Sessions

- Start this week (Friday)!
- Review of important concepts from lecture (remember no text book)
- Help with problem sets
- Friday 9-10AM, 521 Cory Hall

Problem Sets

- Approximately 12 over the course of the semester (one per week)
- Posted on Friday, due on Monday 11:59pm, 10 days later
- Essential to understanding of the material
  - Hence …
- Ok to discuss with colleagues but need to turn in your own work / write-up
- Late turn-in: 20% point deduction per day, except with documented medical excuse
- Solutions posted Friday of due week

Labs

- Choose either FPGA or ASIC or both
- 7 FPGA / 7 ASIC lab exercises, done solo
  - Lab report (check off) due by next lab session
- Design Project lasts 7 weeks, done with partner
  - Project demo/interview RRR week
  - Project report due exam week
- All Labs held in 125 Cory
  - ASIC: W 5-8PM (Arya)
  - FPGA: W 2-5PM, Th 11AM-2PM (Chris)
- All Labs start this week!

Midterm and Final

- Midterm scheduled in evening. No lecture that day.
- Special review session in advance.
  - Midterm: Thur March 14 – 6PM-9PM
- Final: Fri May 17, 7-10PM

All exams are closed book – with one double sided 8.5x11 sheet of notes

Course Information

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

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

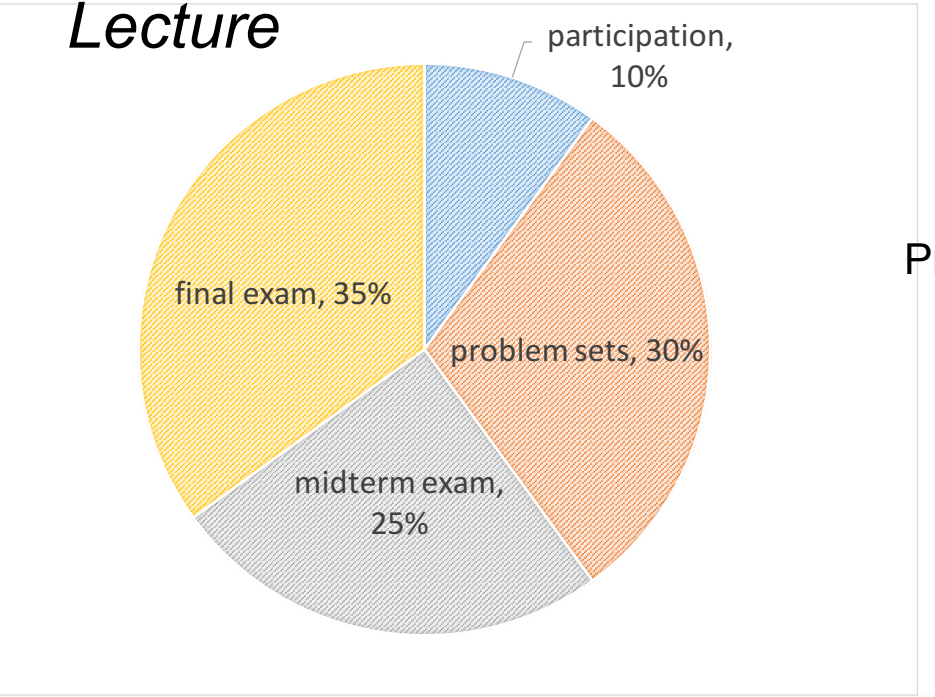
(make sure to register ASAP if you don't want to miss any of the action)
http://piazza.com/berkeley/spring2019/eecs151251a

Cheating Policy

- 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 problem sets, answers on exams, lab exercise checks, project design, and any required course turn-in material.
- 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 (obviously, okay to work with project partner). Okay to discuss homework with others. 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 of cheating reported to the office of student conduct.

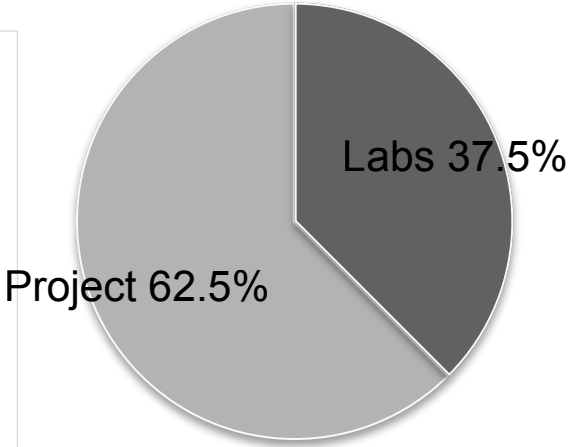
Grading Breakdown

Lecture

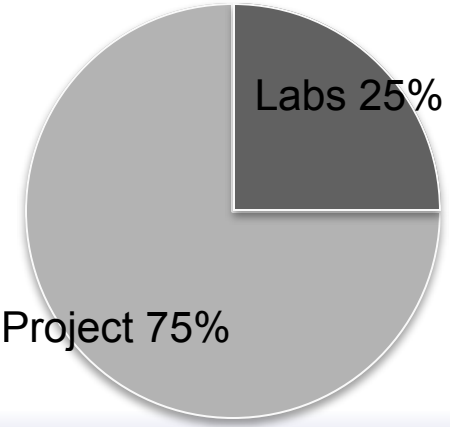


Labs

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.
- 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 project 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

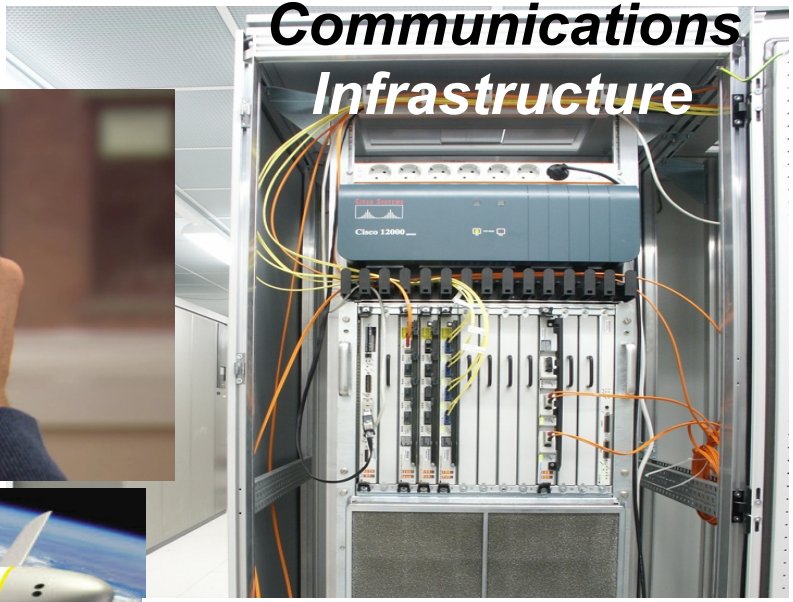
- Discussions and labs start this week
- HW 1 assigned later this week
- Register on Piazza as soon as possible
- Register for your EECS151 class account at inst.eecs.berkeley.edu/webacct
- If you are registering through concurrent enrollment:
  - See me in person at office hour Thursday (2:30 631 Soda)

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

Electronics all around us

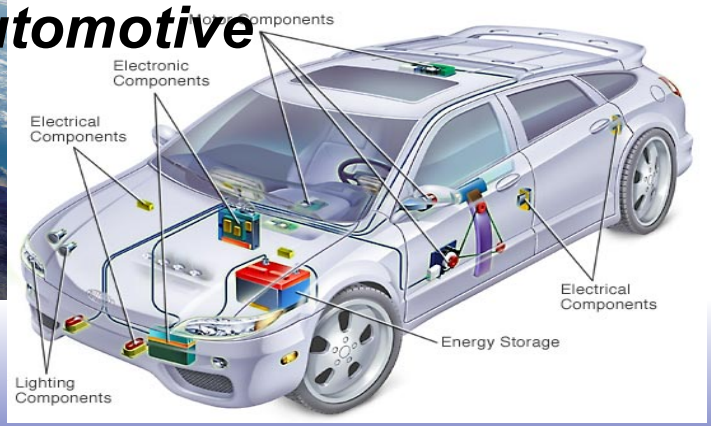
Consumer Products

Aerospace and Military

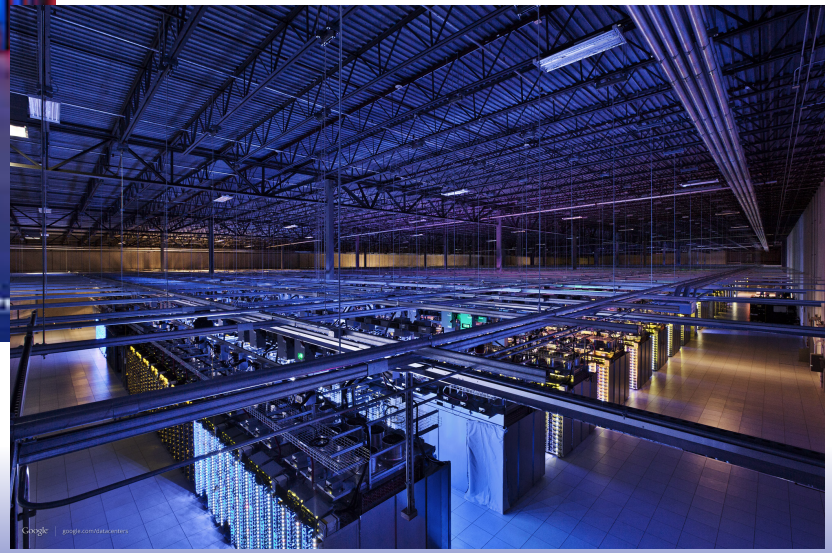


Communications Infrastructure

Automotive

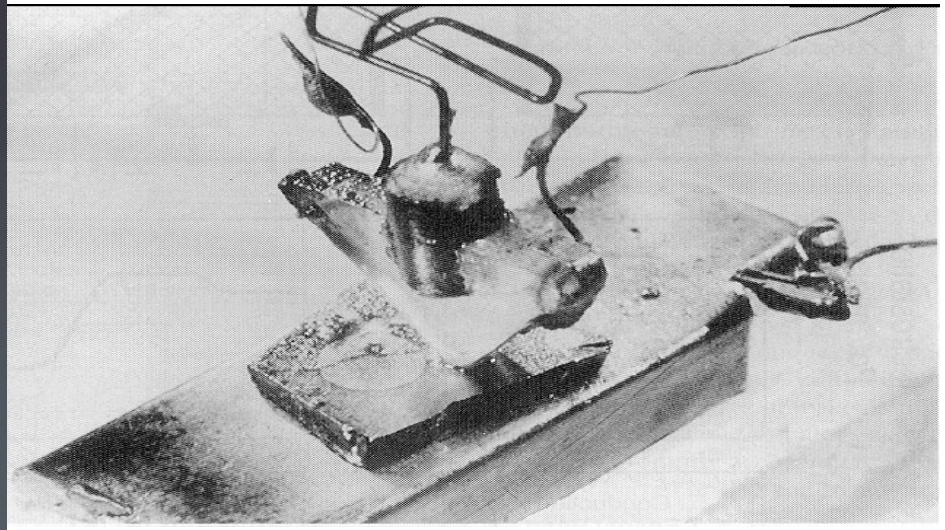


And then plenty more …



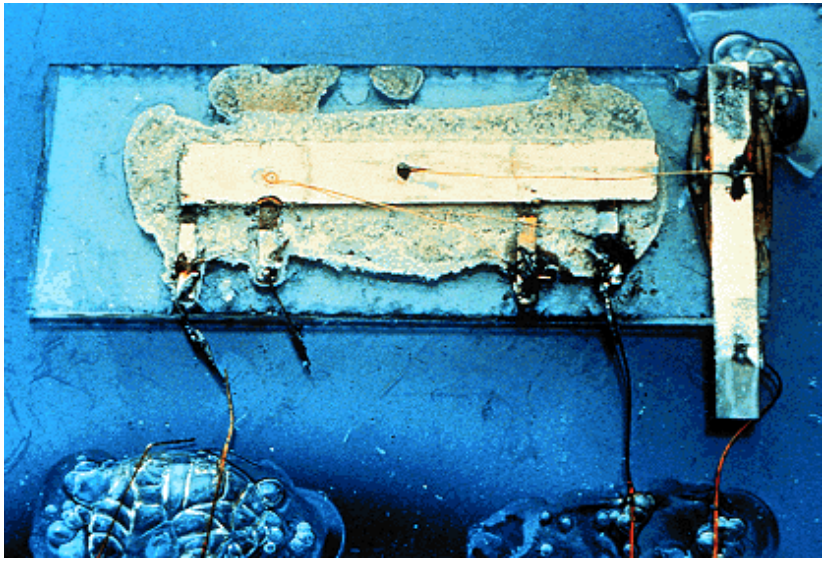
How did this all arise?

The Transistor Revolution



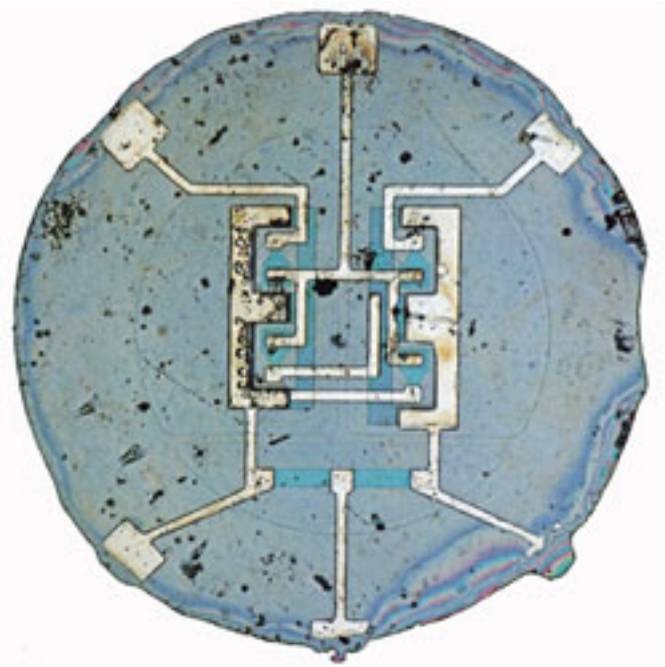
First transistor
Bell Labs, Dec 1947

First Integrated Circuits (1958-59)

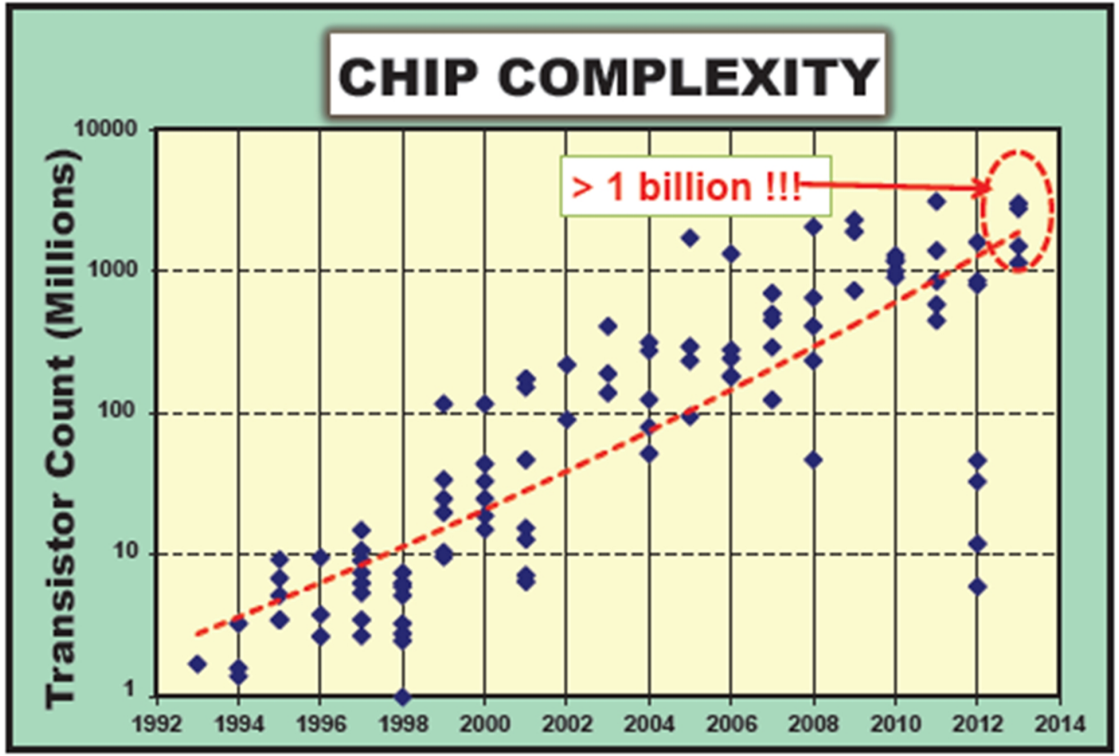


Jack Kilby, Texas Instruments

Bob Noyce, Fairchild



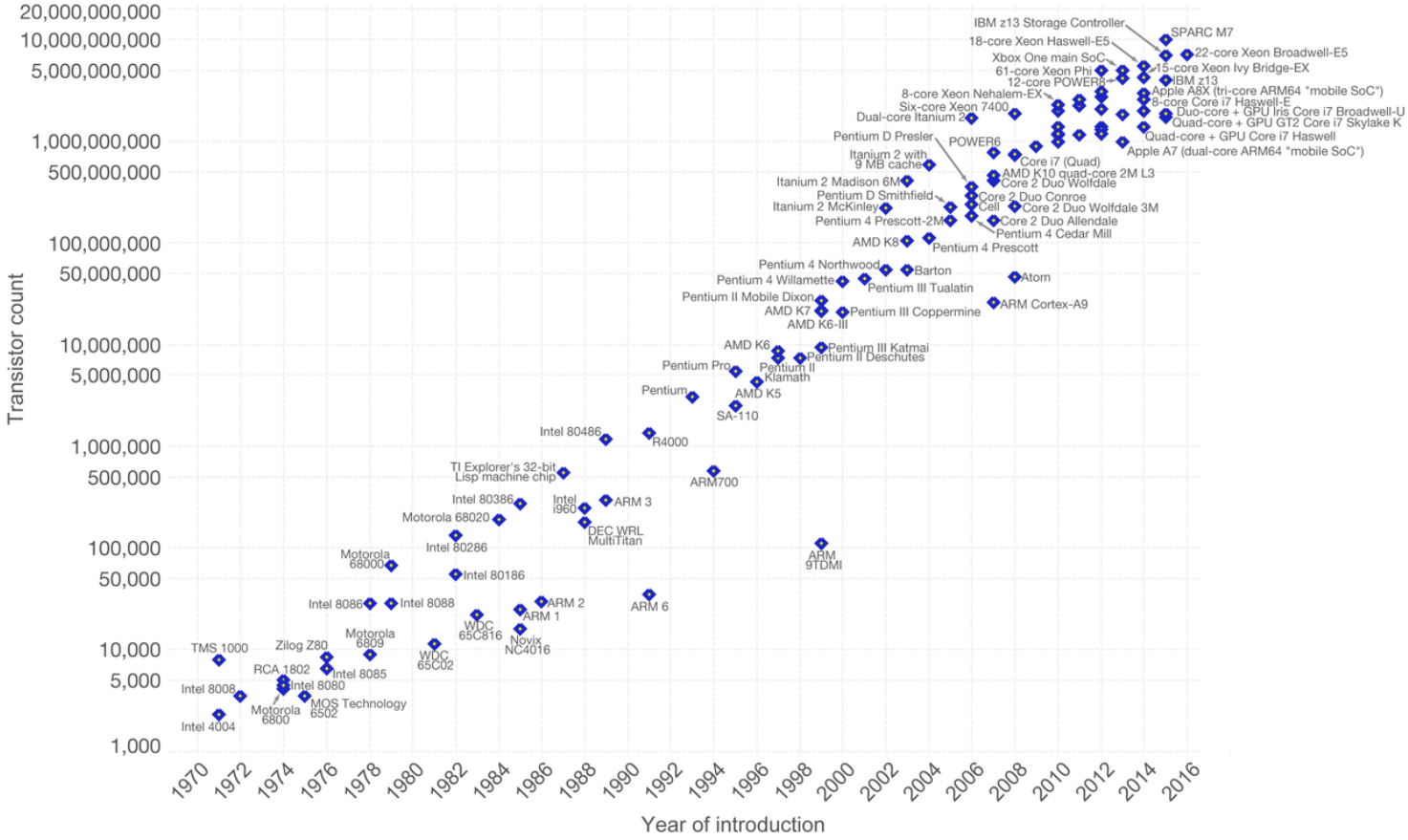
Transistor Counts



Moore's Law – The number of transistors on integrated circuit chips (1971-2016)

Our World in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are strongly linked to Moore's law.

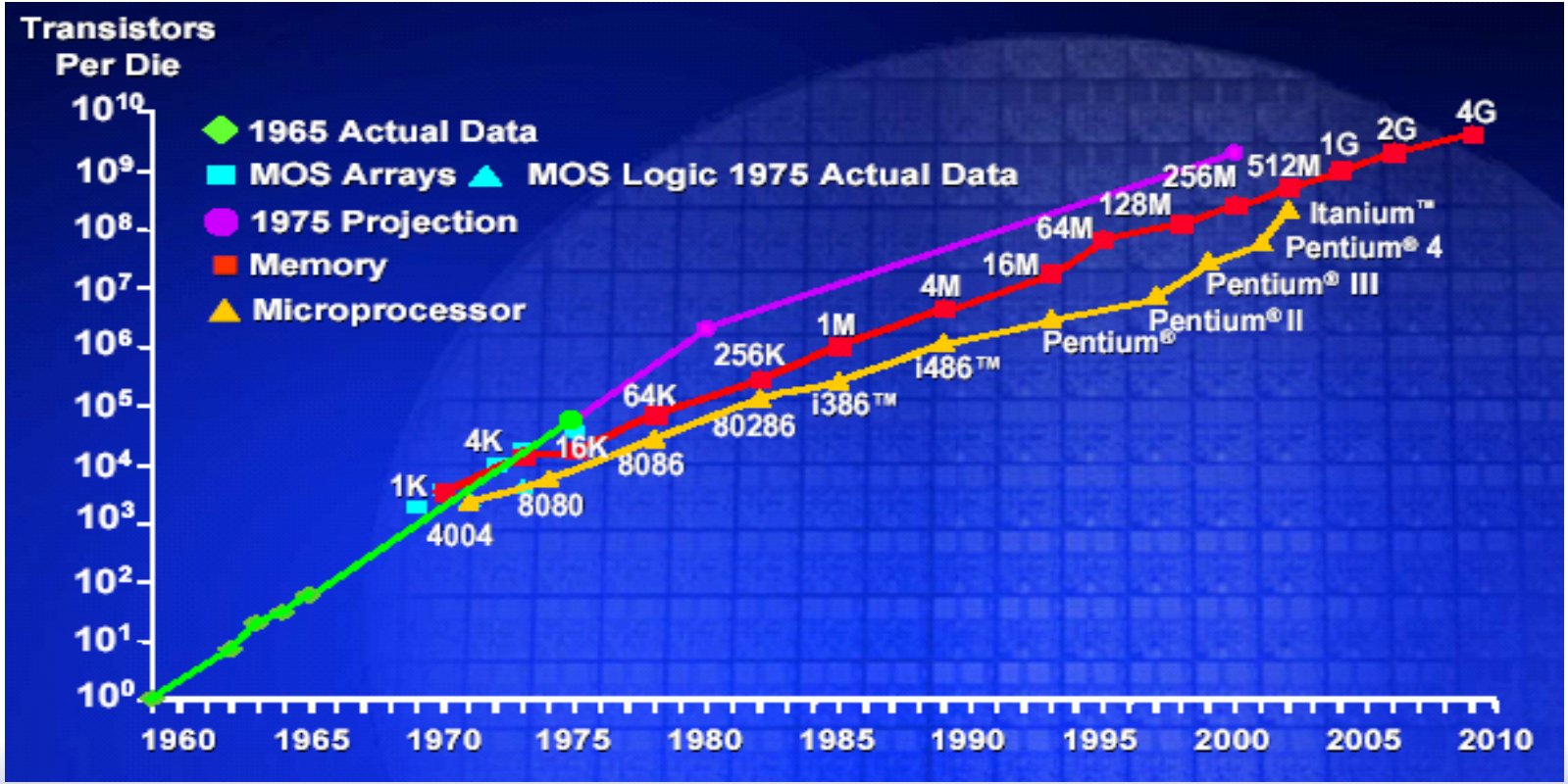


Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)

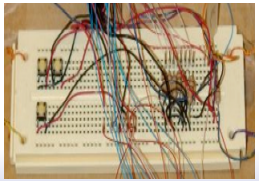
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

Licensed under CC-BY-SA by the author Max Roser.

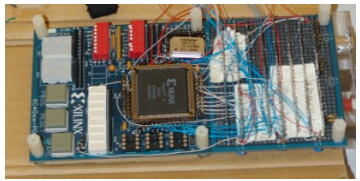
Moore's Law – 2x transistors per 1-2 yr



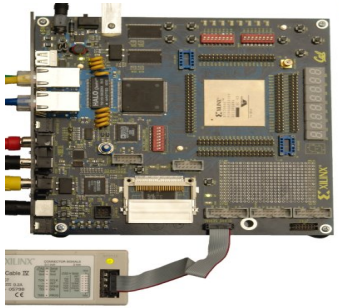
CS150/EECS151 Project Complexity



1980 Pong game
10's of logic gates

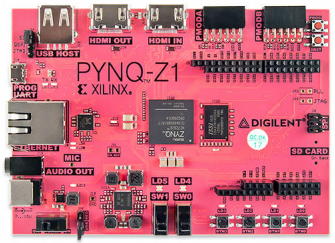


1995 MIDI synthesizer
1000's of logic gates



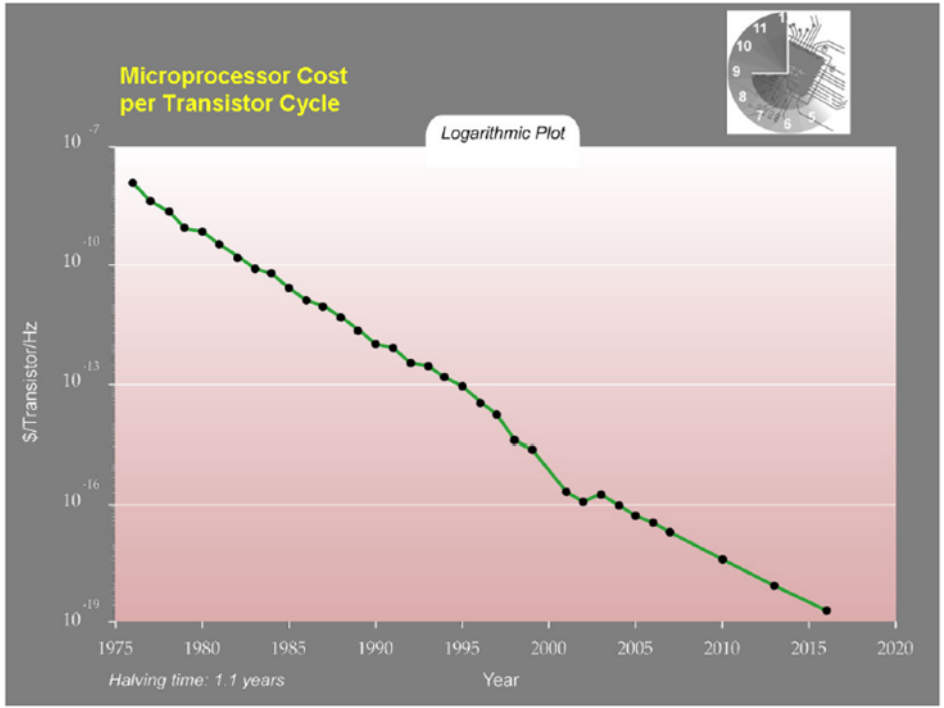
2000-2010 eTV tuner
10K's logic gates

2010-2017 MIPS CPU or BYO
1M logic gates



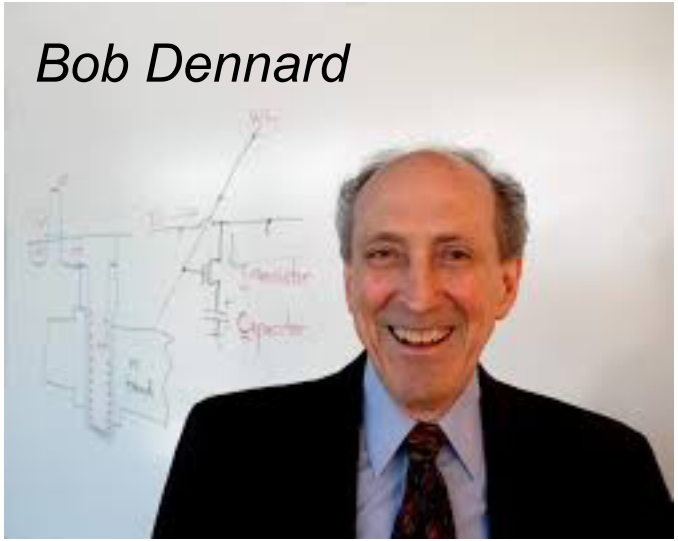
2018 MIPS CPU
Programmable SOC:
dual-core ARM, 85K
logic cells, 220 MACC

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

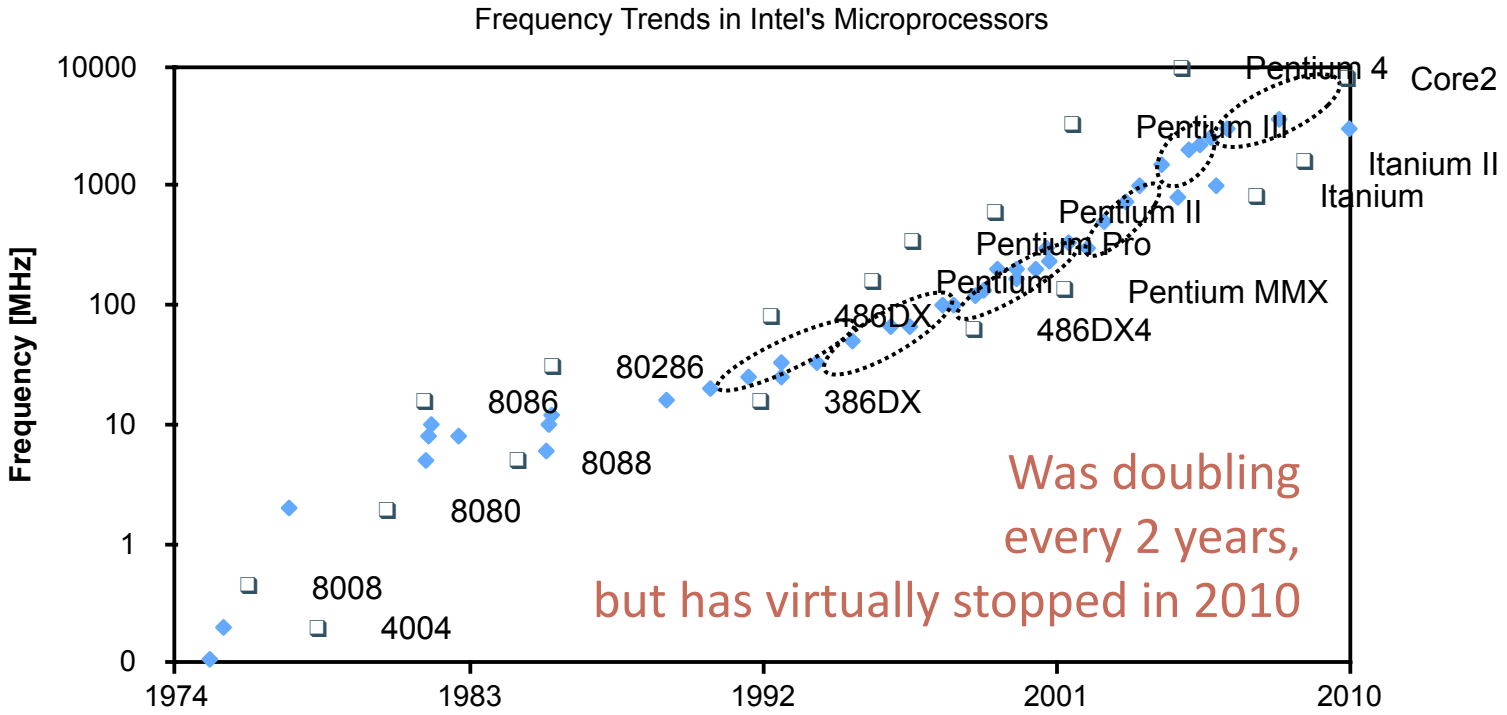
$$Delay \approx C \cdot V / I_{avg}$$

$$P \approx C \cdot V^2 / Delay$$

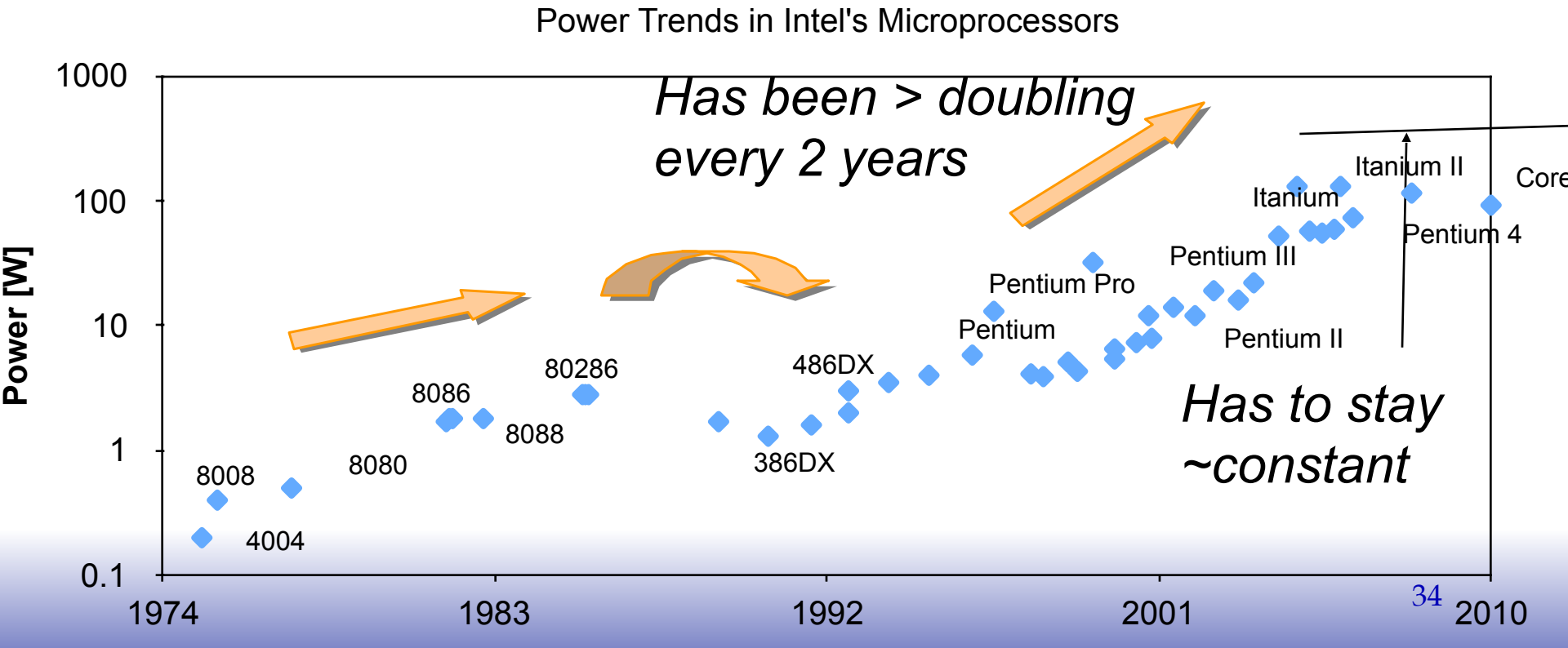
- And, in theory, power *density* constant!

Dennard, Robert H.; Gaensslen, Fritz; Yu, Hwa-Nien; Rideout, Leo; Bassous, Ernest; LeBlanc, Andre (October 1974). "Design of ion-implanted MOSFET's with very small physical dimensions" (PDF). *IEEE Journal of Solid State Circuits*. **SC-9** (5).

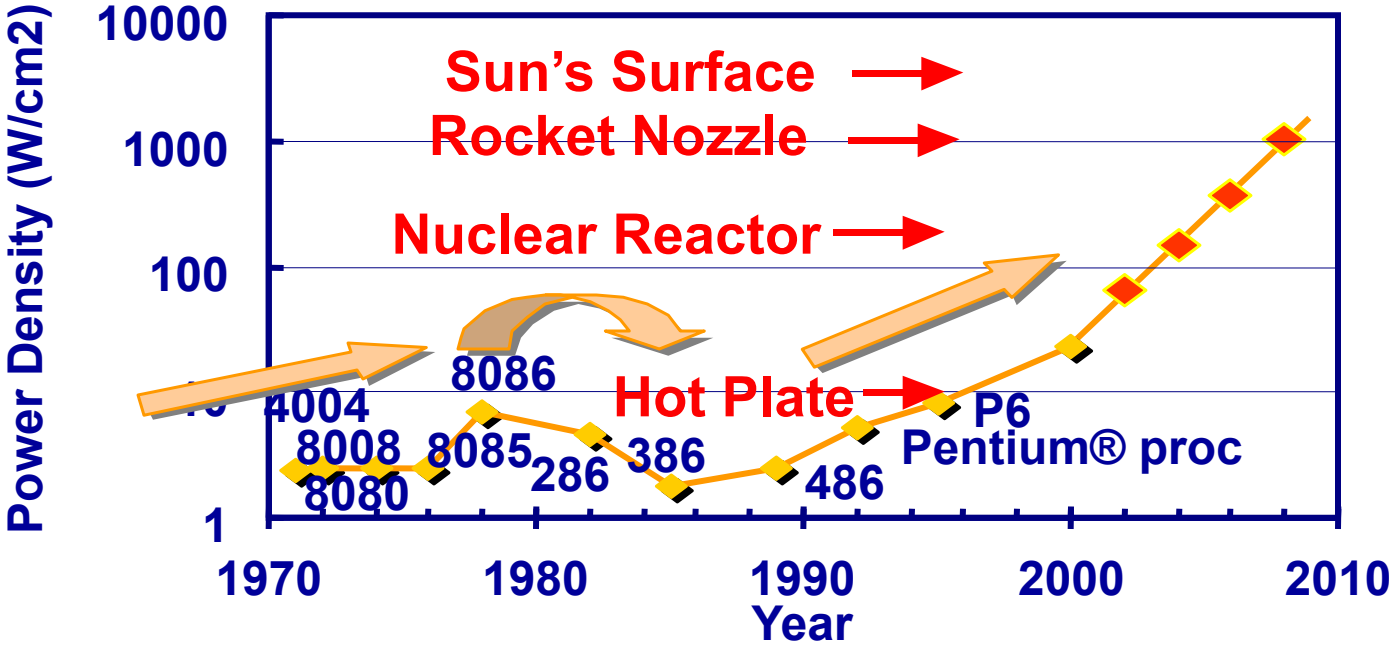
Frequency


Frequency Trends in Intel's Microprocessors
Frequency [MHz]
10000
1000
100
10
1
0
1974
1983
1992
2001
2010
Pentium 4
Core2
Pentium III
Itanium II
Itanium
Pentium II
Pentium Pro
Pentium
Pentium MMX
486DX
486DX4
80286
386DX
8086
8088
8080
8008
4004
Was doubling
every 2 years,
but has virtually stopped in 2010

Power Dissipation



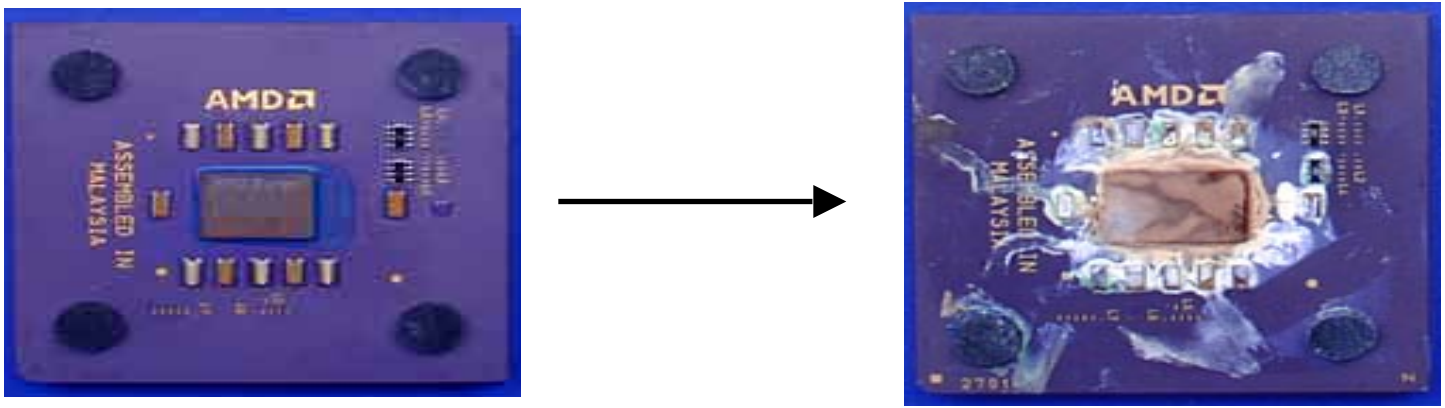
Cause: Power Density



S. Borkar

Power density too high for cost-effective cooling

Not enough cooling…



*Pictures from http://www.tomshardware.com/2001/09/17/hot_spot/

Some quick questions

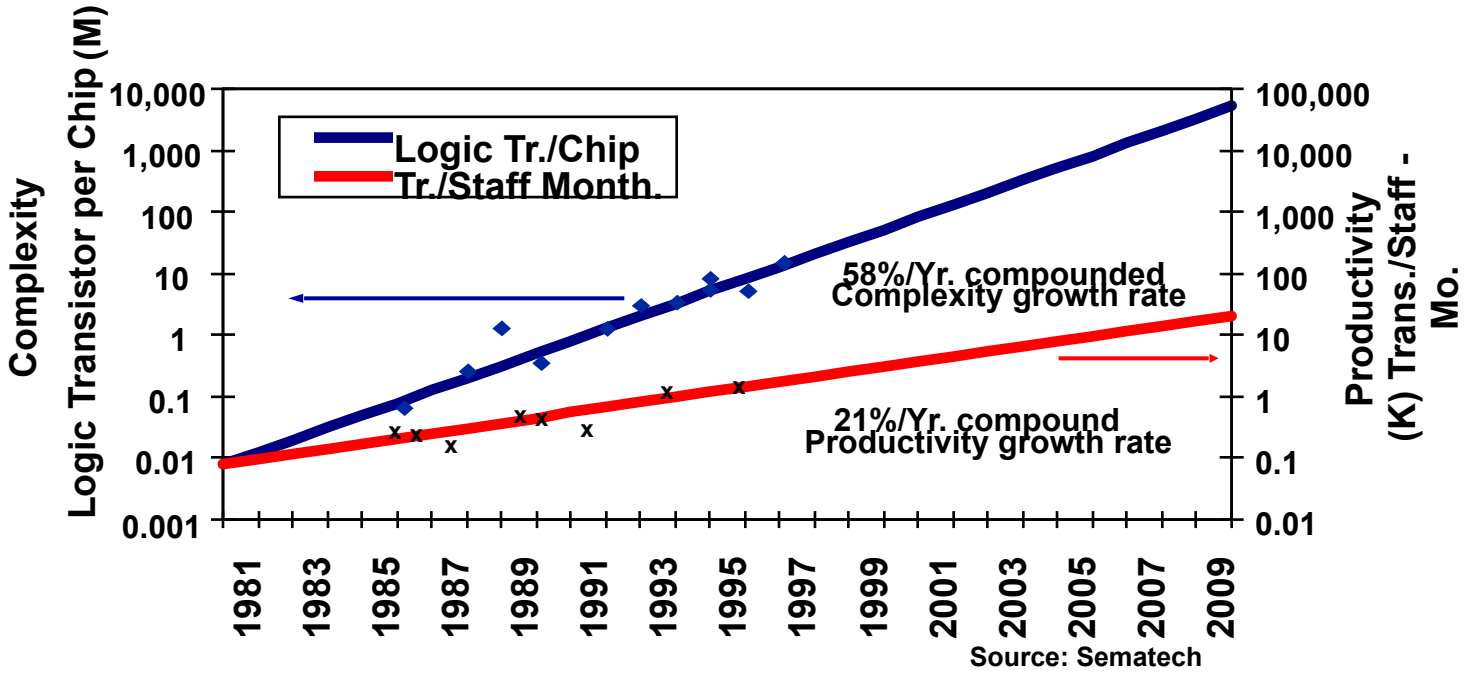
- True or False: Moore's law allows us to predict future properties in the same way that the law of gravity allows us to predict the path of a planet around the sun.

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

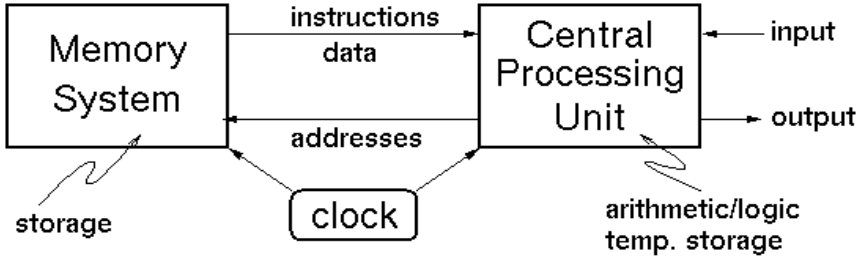
The answers

- Design methodology!
  - Abstraction
  - Hierarchy
  - Reuse
- Computer Aided Design tools

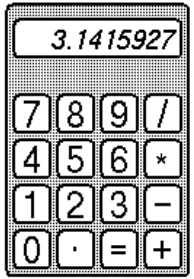
Digital System Design: A few basic concepts

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.

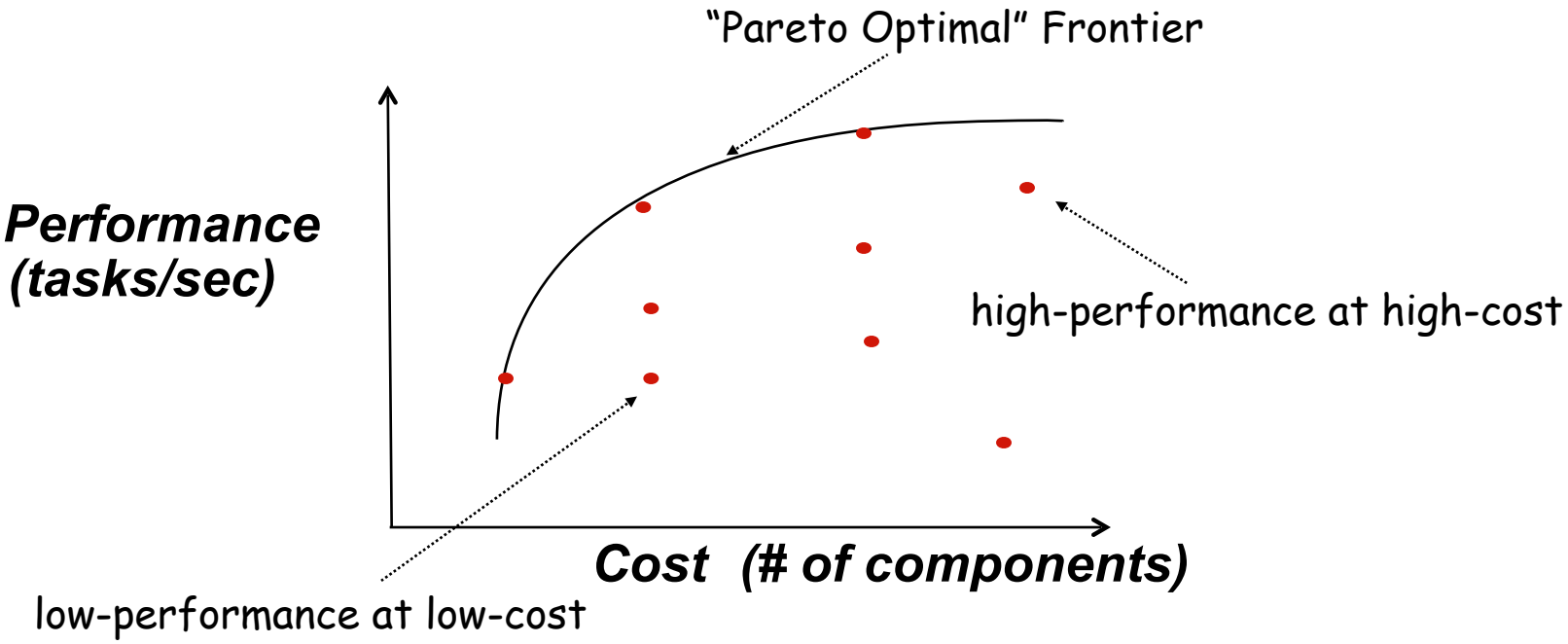
Example Digital Systems

- Digital Watch

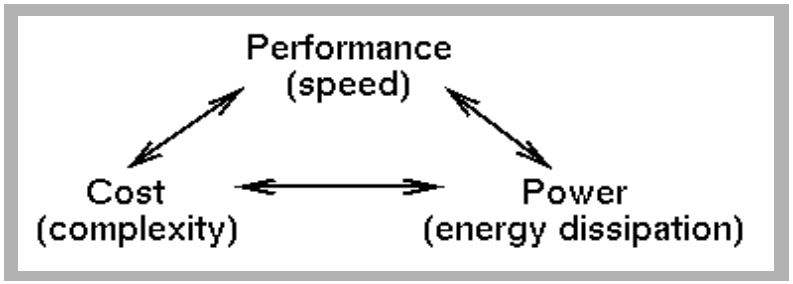
Designed to minimize power.
Single battery must last for years.

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

Design Space & Optimality

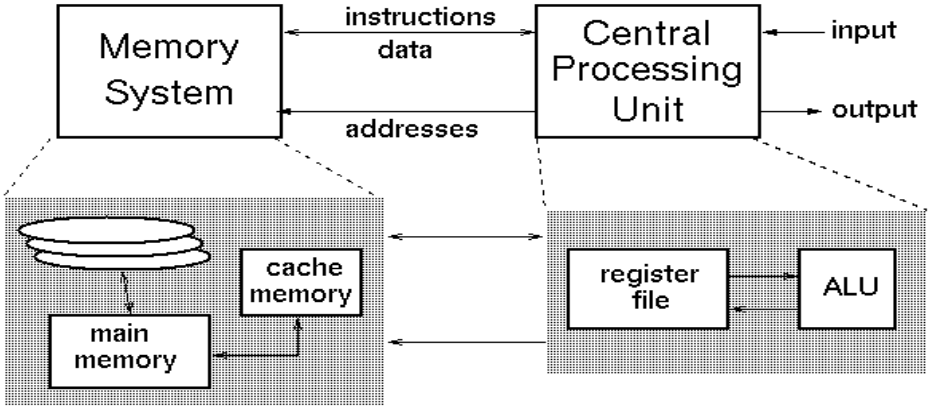


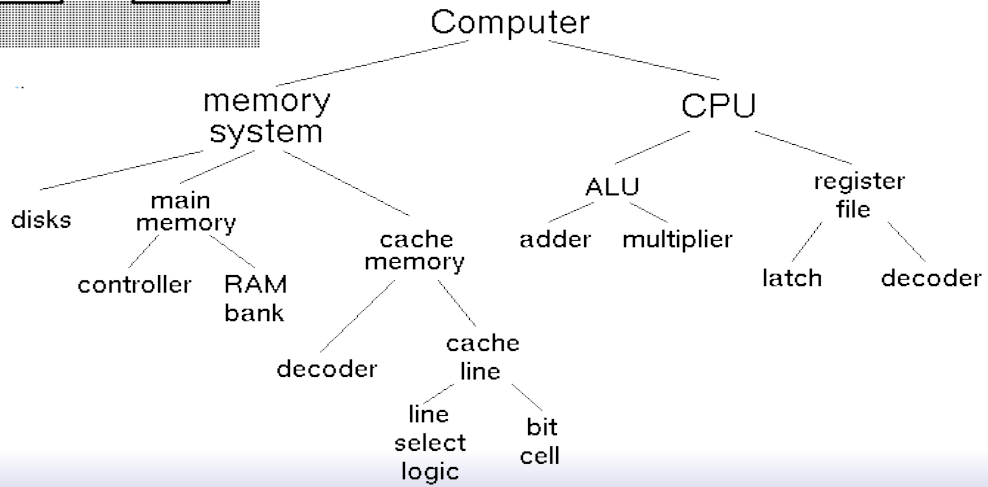
Basic Design Tradeoffs



- 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

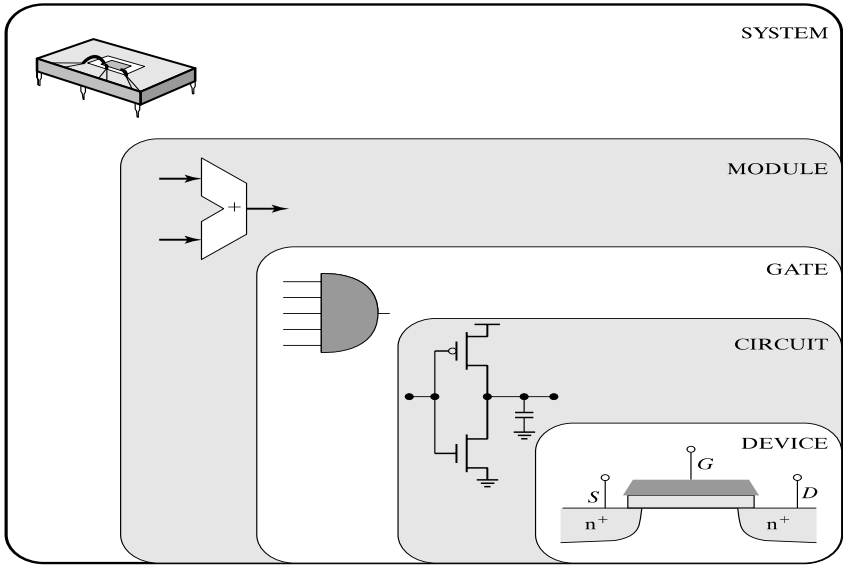
Hierarchy & Design Representation

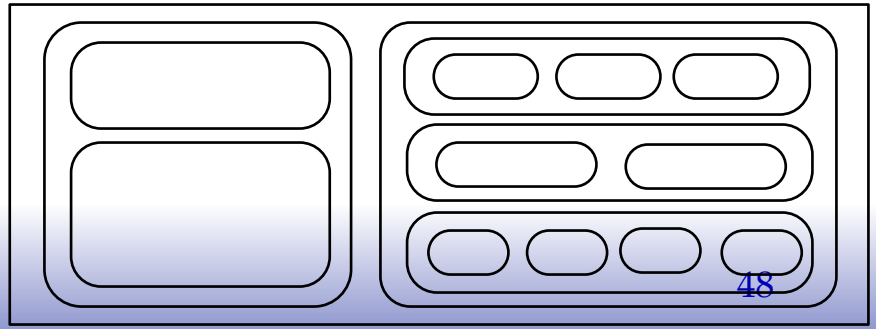




Hierarchy in Designs – Complexity Control

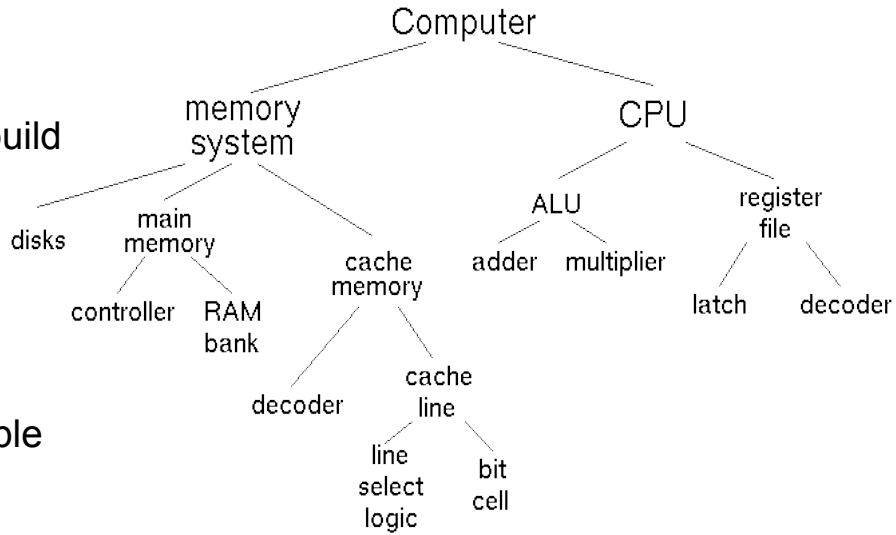
- Design Abstraction
  - Hide details and reduce number of things to handle at any time
- Modular design
  - Divide and conquer
  - Simplifies implementation and debugging





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