

EECS151/251A Introduction to Digital Design and ICs

Lecture 1 – Introduction

Sophia Shao



Welcome!

Tuesday and Thursday

11am-12:30pm Pacific

Mulford Hall 159 + Zoom





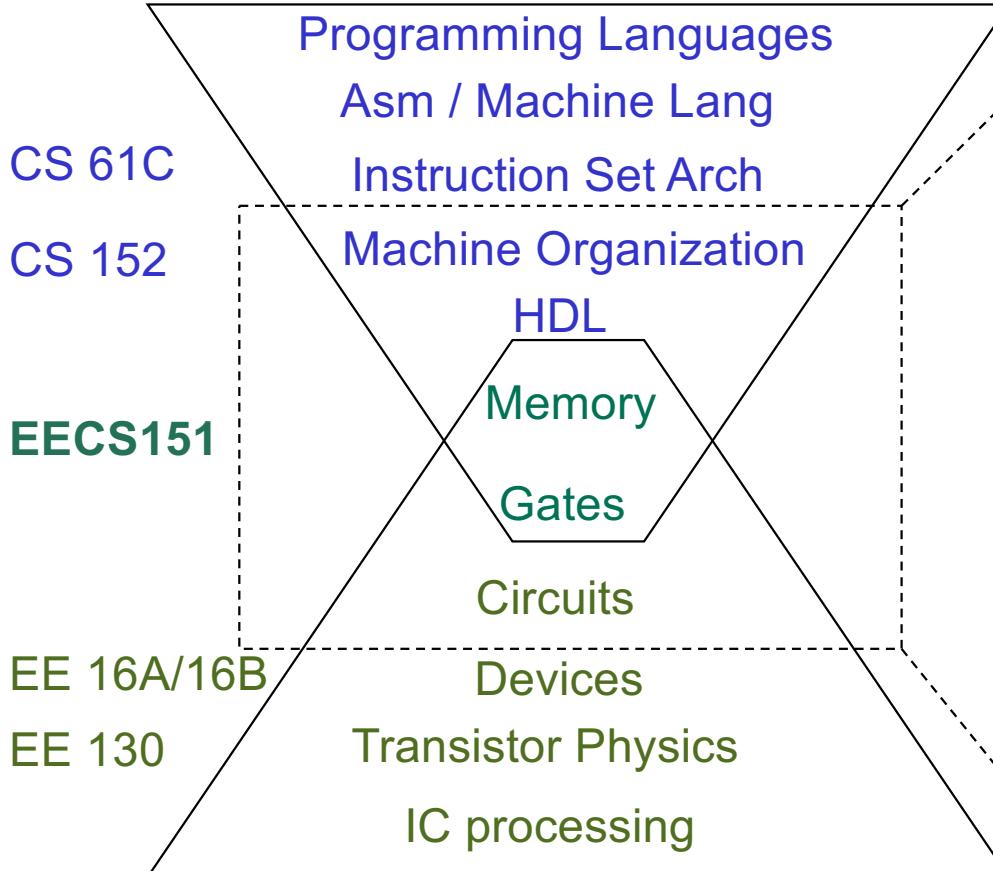
Class Goals

What This Class is All About?

- **Introduction to digital integrated circuit and system engineering**
 - Key concepts needed to be a good digital system designer
 - Discover your own creativity!
- **Learn abstractions 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**
 - *There are way more ways to mess up a chip than to get it right.*

**Digital design is not twitch.com!
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 Languages (HDL)
- Design Flow (CAD)

Prerequisites

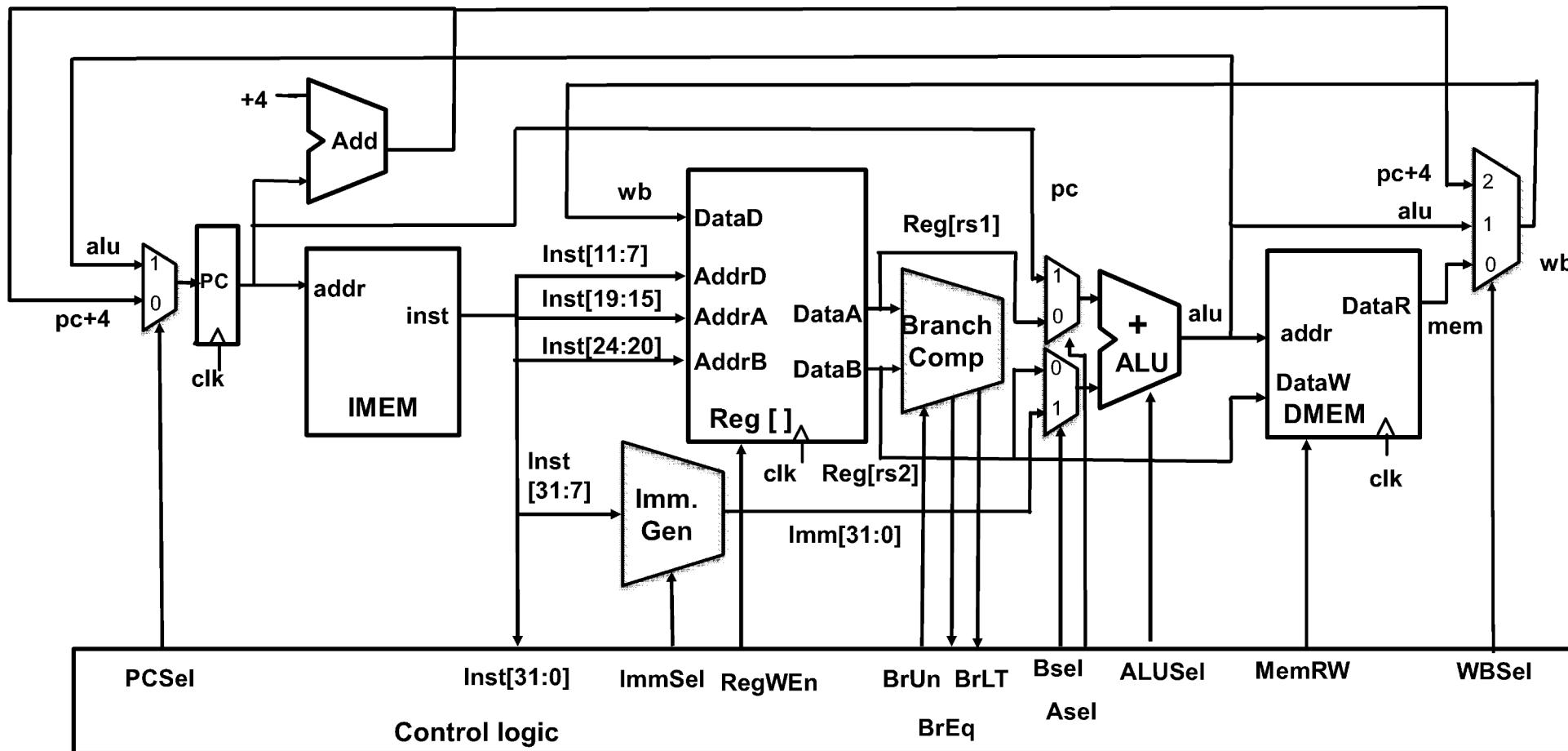
- CS61C
 - C, Boolean logic, RISC-V ISA
 - We will review combinational and sequential logic and RISC-V datapath, pipelining (and go much more in depth)
- EE16A/B
 - Digital gates, RC networks
 - We will review transistor operation and design of CMOS logic

Possible Course Sequences

- CS 61C → EECS 151 → CS152 → ...
EE 16A/B
- CS 61C → EECS 151 → CS152 → ...
EE 16A/B

- With 151/152 background should be able to take any graduate-level course in architecture/digital systems
- EECS 151 + EE 140 is a springboard into integrated circuits

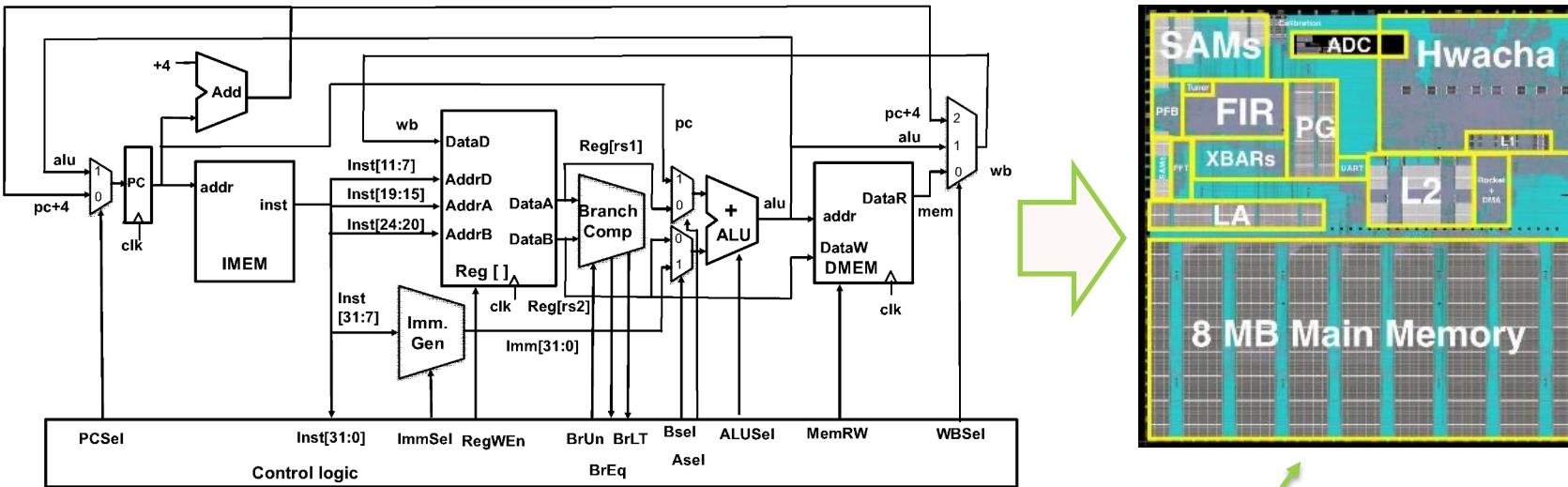
CS61C Background – RV32I



- Don't worry about details – we will rebuild it and make it work!

At the end of EECS 151

- Should be able to build a complex digital system



Berkeley chip
of IEEE Journal of Solid-State Circuits



Administrivia

EECS151/251A Fall 2022 Teaching Staff



Ella Schwarz
(FPGA Lab)



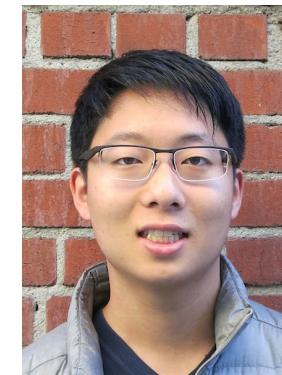
Erik Anderson
(Discussion,
ASIC Lab)



Hansung Kim
(Discussion,
ASIC Lab)



Jennifer Zhou
(FPGA Lab)



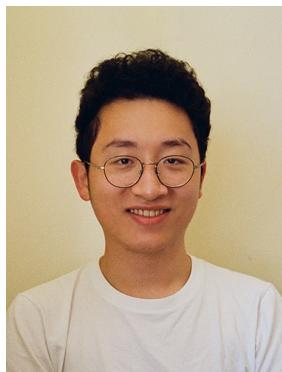
Paul Kwon
(Discussion,
FPGA Lab)



Professor
Sophia Shao



Raghav Gupta
(FPGA Lab)



Richard Yan
(ASIC Lab)



Roger Hsiao
(Discussion,
ASIC Lab)



Simon Guo
(FPGA Lab)



Yikuan Chen
(FPGA Lab,
Admin)



Chengyi Zhang
(Reader)

Course Information

- Basic Source of Information, class website:

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

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

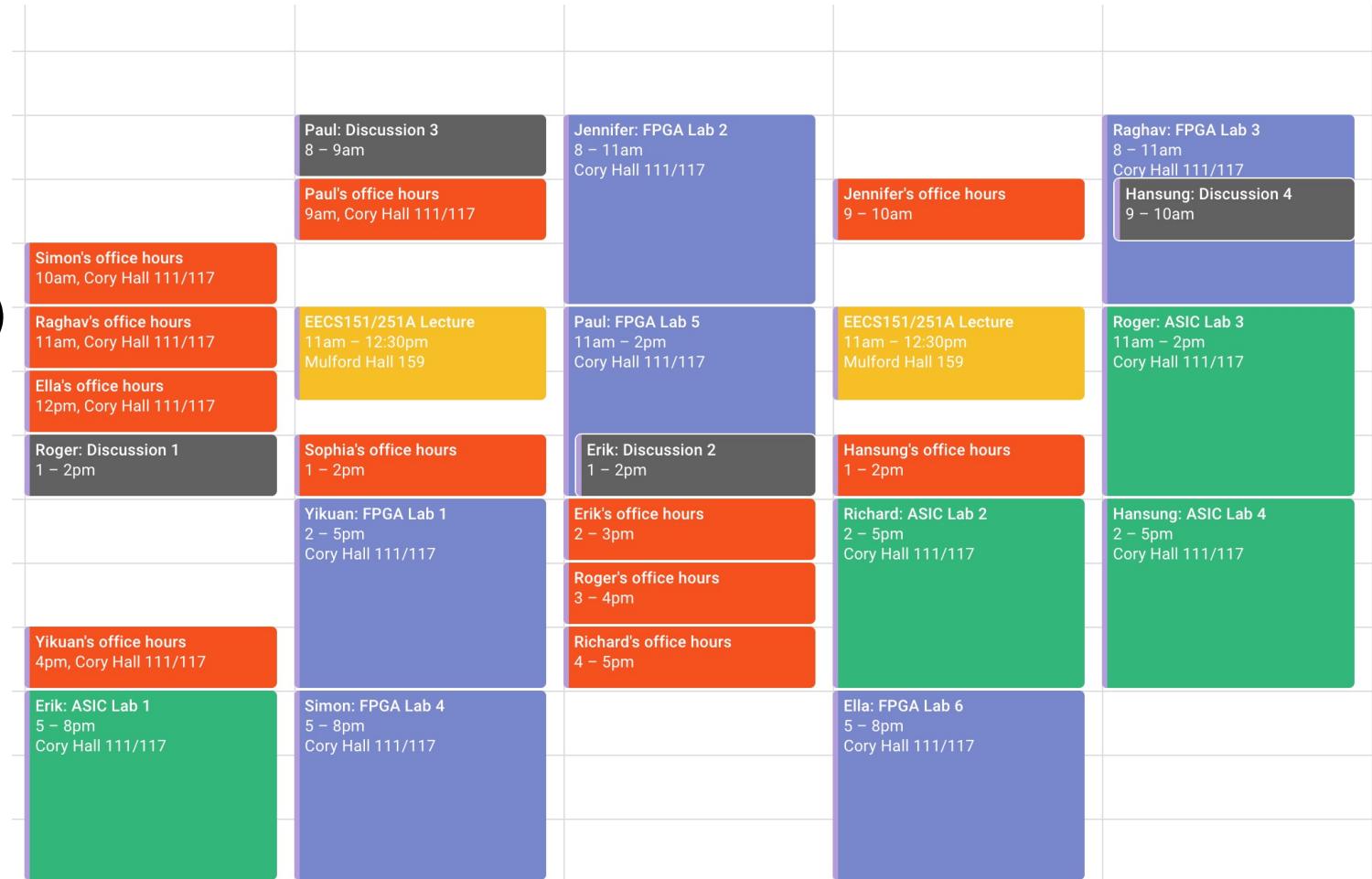


Print only what you need: Save a tree!

Class Organization

- Lectures
- Discussion sessions
- Office hours
- Problem Sets (~1 per week)
- Labs – FPGA or ASIC
- Design project
- 1 Midterm + 1 Final

MON TUE WED THU FRI



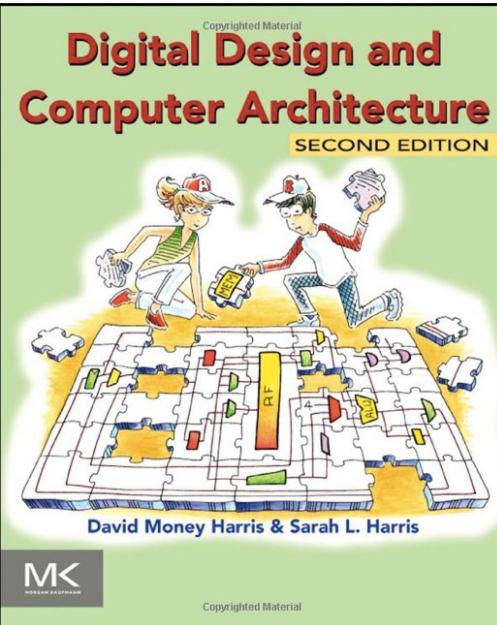
Lectures

- Slides available on website before the lecture
- Lectures are recorded!
 - If you can, come to lectures!
 - We like interactive lectures!
 - Recording will be posted after the lecture.



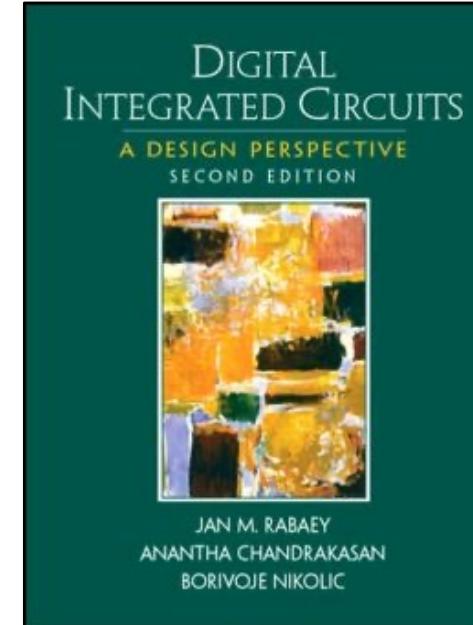
Class Textbooks

No Required Book this semester

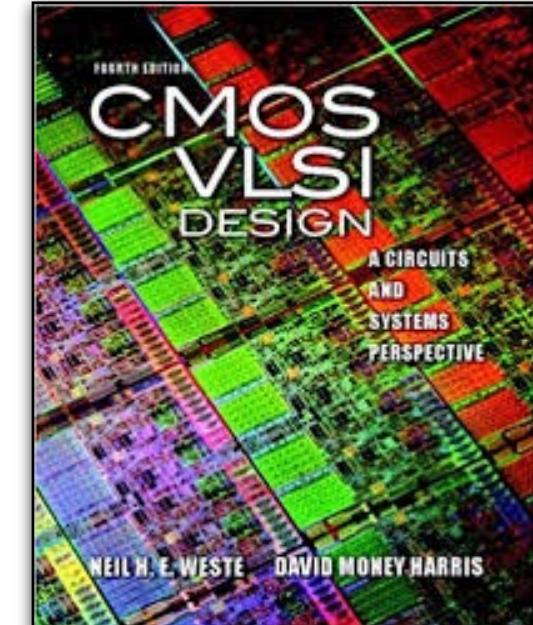


Recommended
(previously required)

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



Recommended
(previously required)



Useful

Discussion Sessions

- 4 Discussions in total
 - We just added a new one on Fridays.
 - Feel free to go to any of them.
- Start next Friday!
- Review of important concepts from lecture
- Help with problem sets



Problem Sets

- Approximately 10 over the course of the semester (one per week)
- Posted on Thursday, due on Friday 11:59pm, 8 days later
- Essential to understanding of the material
- Ok to discuss with colleagues but need to turn in your own work / write-up
- Late turn-in:
 - 7 slip days for lectures + 14 slip days for labs.
 - 20% point deduction per day after slip days, except with documented medical excuse
- Solutions posted the week after due date



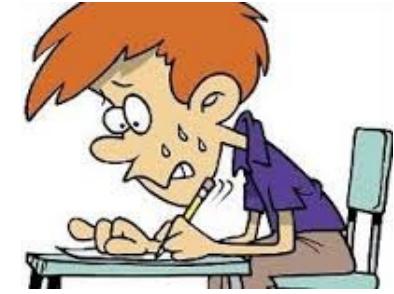
Labs

- Choose either FPGA or ASIC
- 6 FPGA / 6 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 RRR week
- 6 FPGA Labs and 4 ASIC Labs
 - Stick to one of the labs.
 - Know the Lab TA and fellow students.
- **Labs start next week!**



Midterm and Final

- 1 Midterm + 1 Final
- Review session in advance
 - Midterm:
 - Late October, 7-9pm (tentative)
 - Final:
 - Wed, 12/14, 8-11am Pacific
- No alternative midterm/final.
 - We may allow students to take the final right after the official slot.



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

“Clobber Policy”

- The clobber policy allows you to:
 - Override your Midterm score with the score on the final if you perform better on the final;
 - Note that the reverse is not true - you must take the entire final exam, regardless of your Midterm score.

Course Information

- For interactions between faculty, GSIs and fellow students – we are using Ed Discussion
For fastest response **post your questions on Ed Discussion.**



(Follow the link on the course website to register)

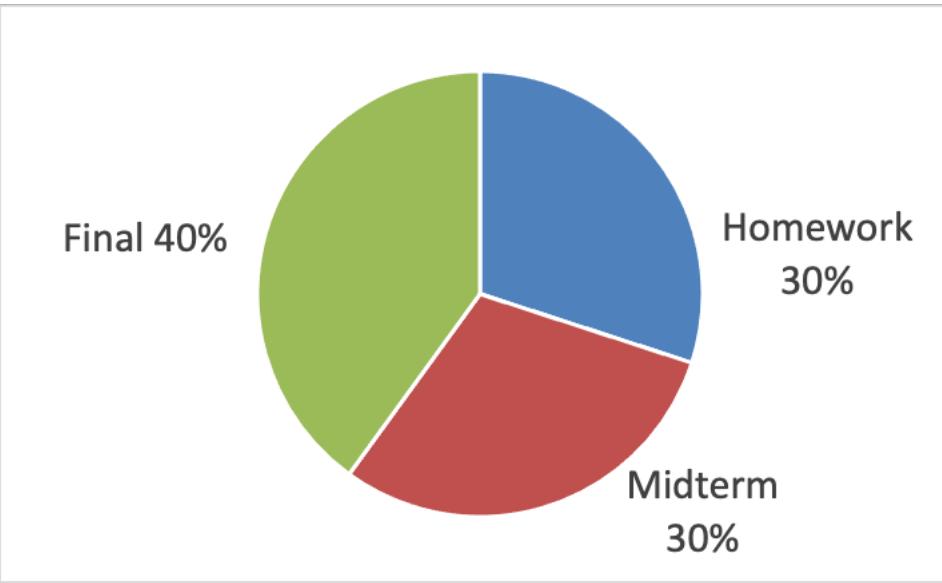
<https://inst.eecs.berkeley.edu/~eecs151/>

Honor Code

- 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.
- 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.

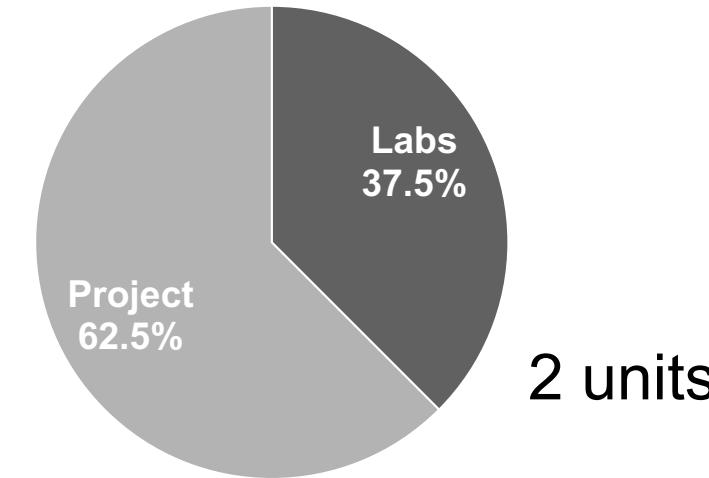
Grading Breakdown

Lecture



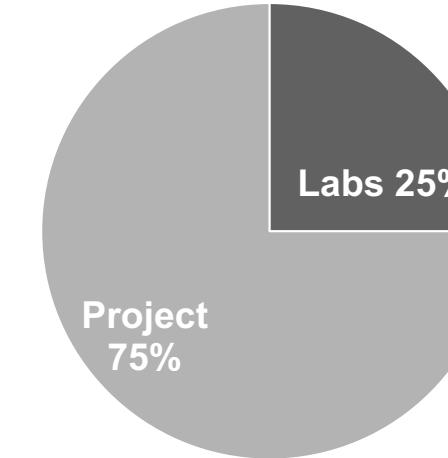
3 units

LA - ASIC



2 units

LB - FPGA



2 units

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”.
 - Slides will be online before the lecture. Study 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.

Tips on How to Get a Good Grade

- 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 10+ 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

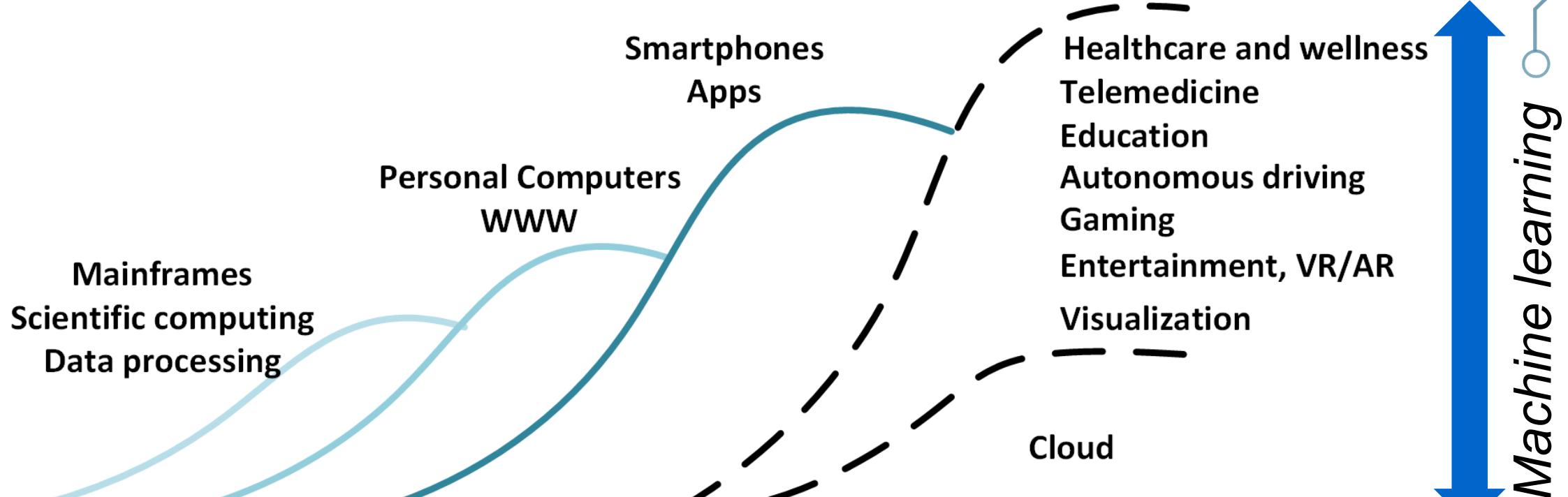
- Labs start next week.
- Discussion starts next Friday.
 - New Discussion 4 added!
- HW 1 assigned next Thursday.
- Register on Ed as soon as possible.
- Register for your EECS151 class account at inst.eecs.berkeley.edu/webacct



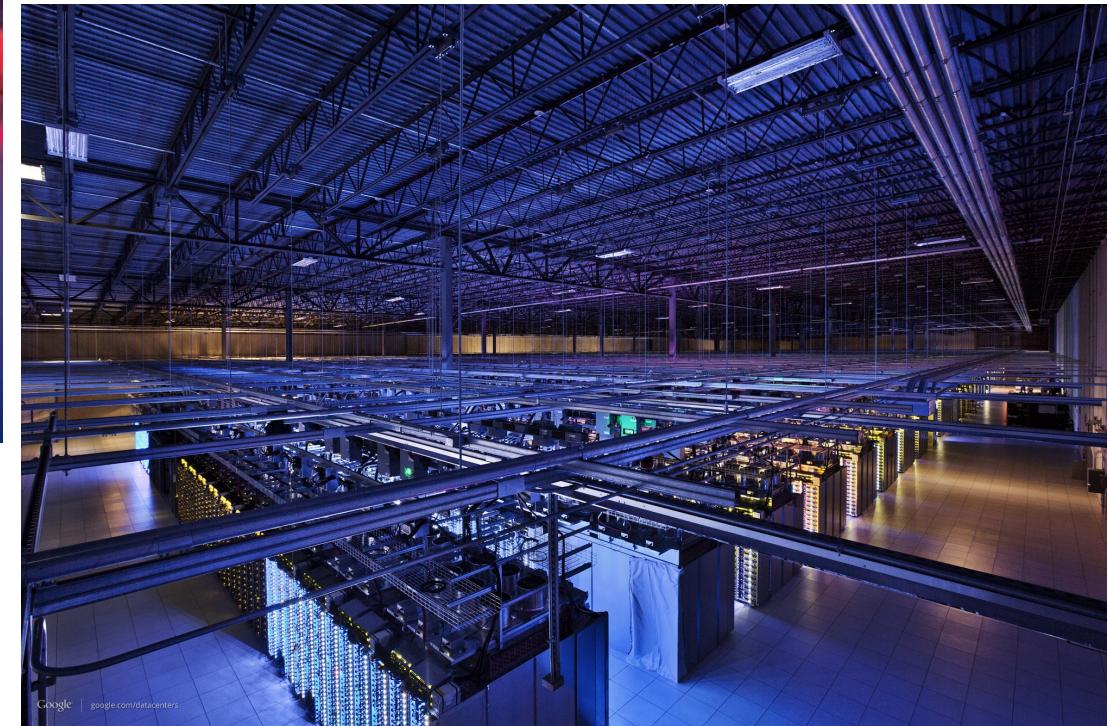
Digital Integrated Circuits and Systems

Past, Present and Future

Diversifying Applications



Remember: My Other Computer ...

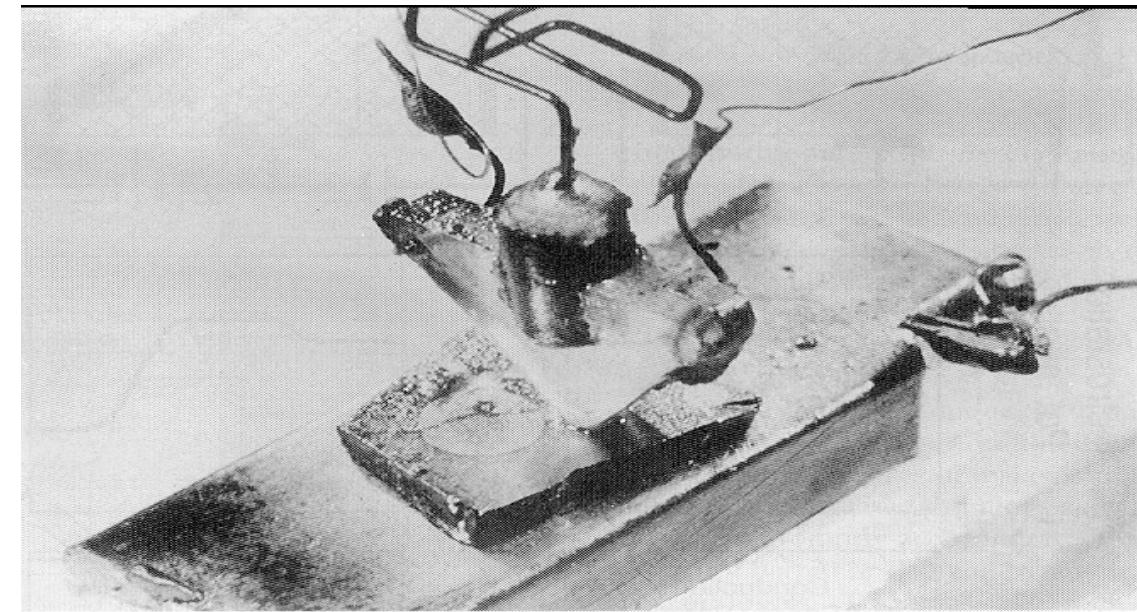
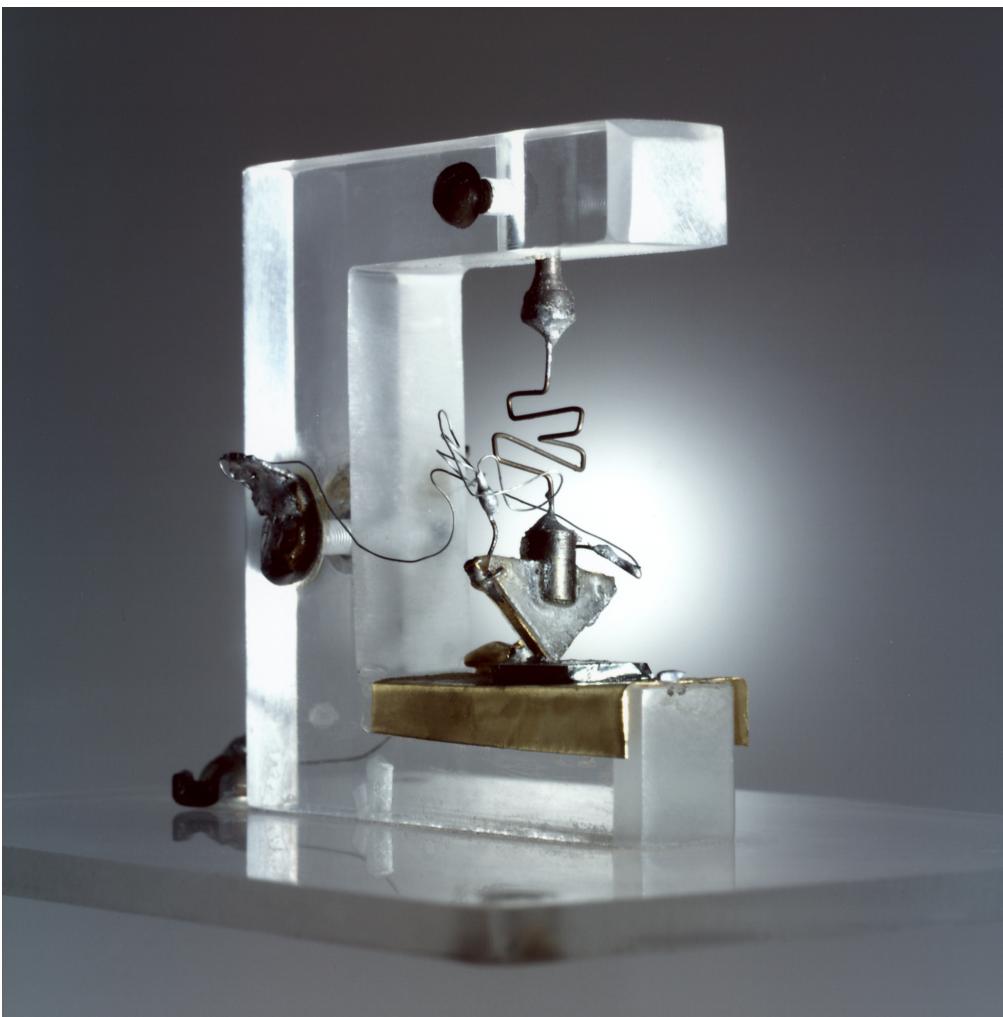


- It is being customized as well!



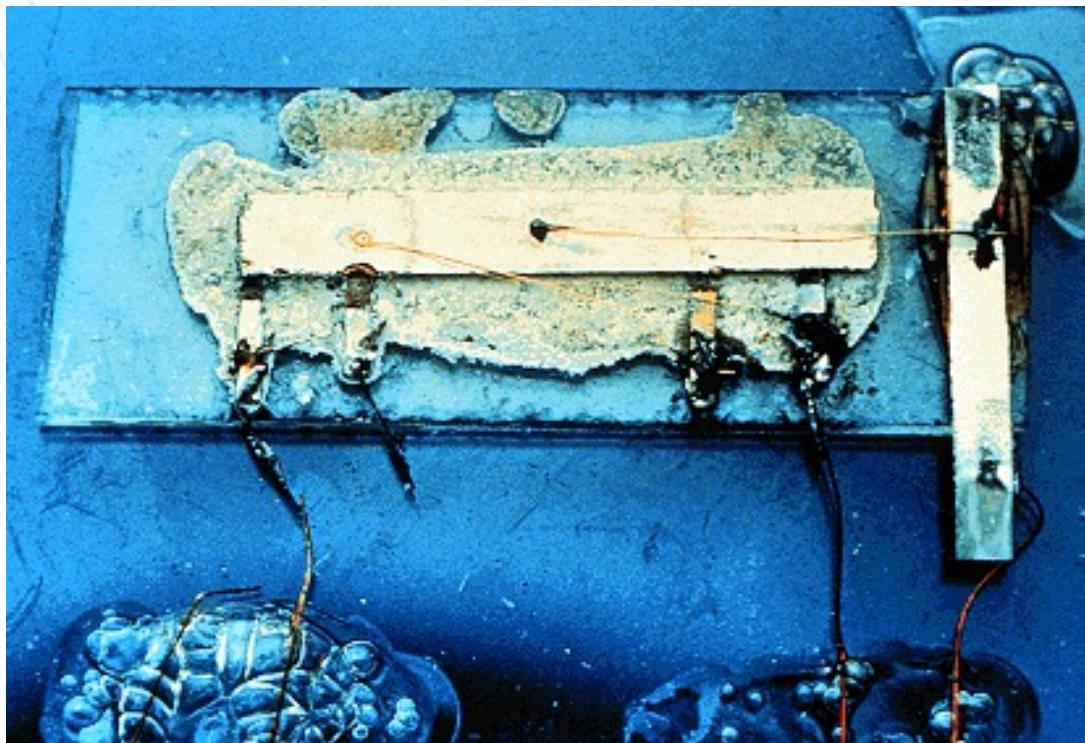
How Did All This Arise?

The Transistor Revolution



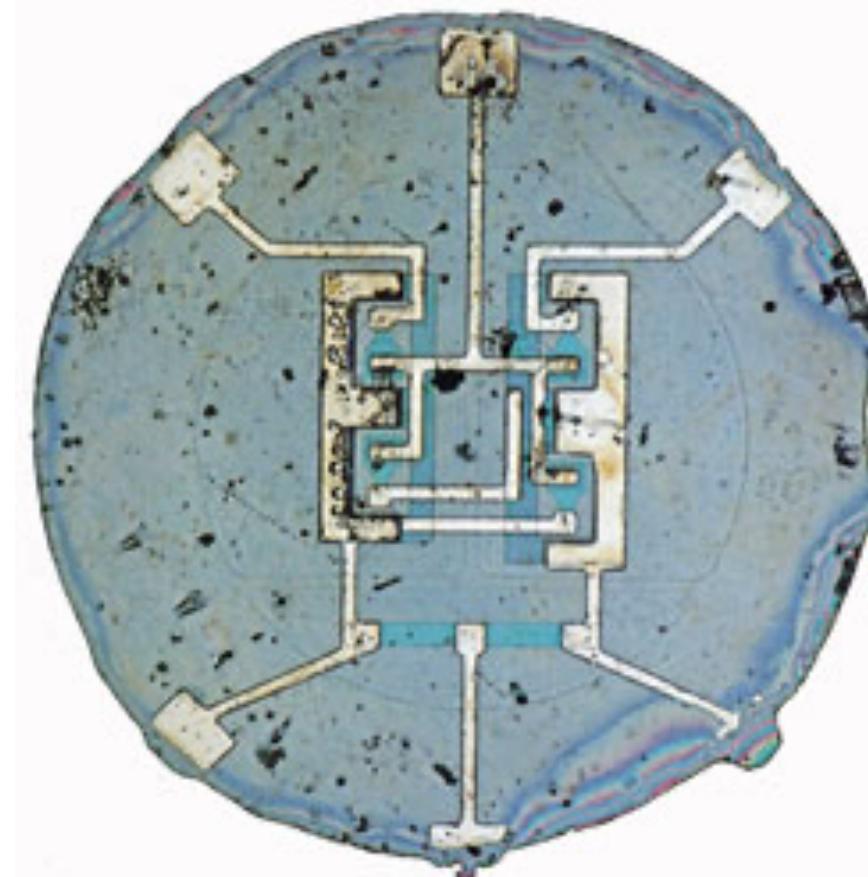
First transistor
Bell Labs, Dec 1947

First Integrated Circuits (1958-59)



Jack Kilby, Texas Instruments

Bob Noyce, Fairchild



Moore's Law

- In 1965, Gordon Moore noted that the number of transistors on a chip doubled every 12 months.
- He made a prediction that semiconductor technology will double its effectiveness every ~~12~~ months

~~18~~

24

“The complexity for minimum component **costs** has increased at a rate of roughly a factor of two per year. Certainly over the short term, this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years. That means by 1975, the number of components per integrated circuit for minimum cost will be 65,000.”

Gordon Moore, Cramming more Components onto Integrated Circuits, (1965).

Moore's Law - 1965

Transistors

Per Die

10^{10}

10^9

10^8

10^7

10^6

10^5

10^4

10^3

10^2

10^1

10^0

1960

1965

1970

1975

1980

1985

1990

1995

2000

2005

2010

33

Graph from S.Chou, ISSCC'2005

EECS151 L01 INTRODUCTION

Shao Fall 2022 © UCB

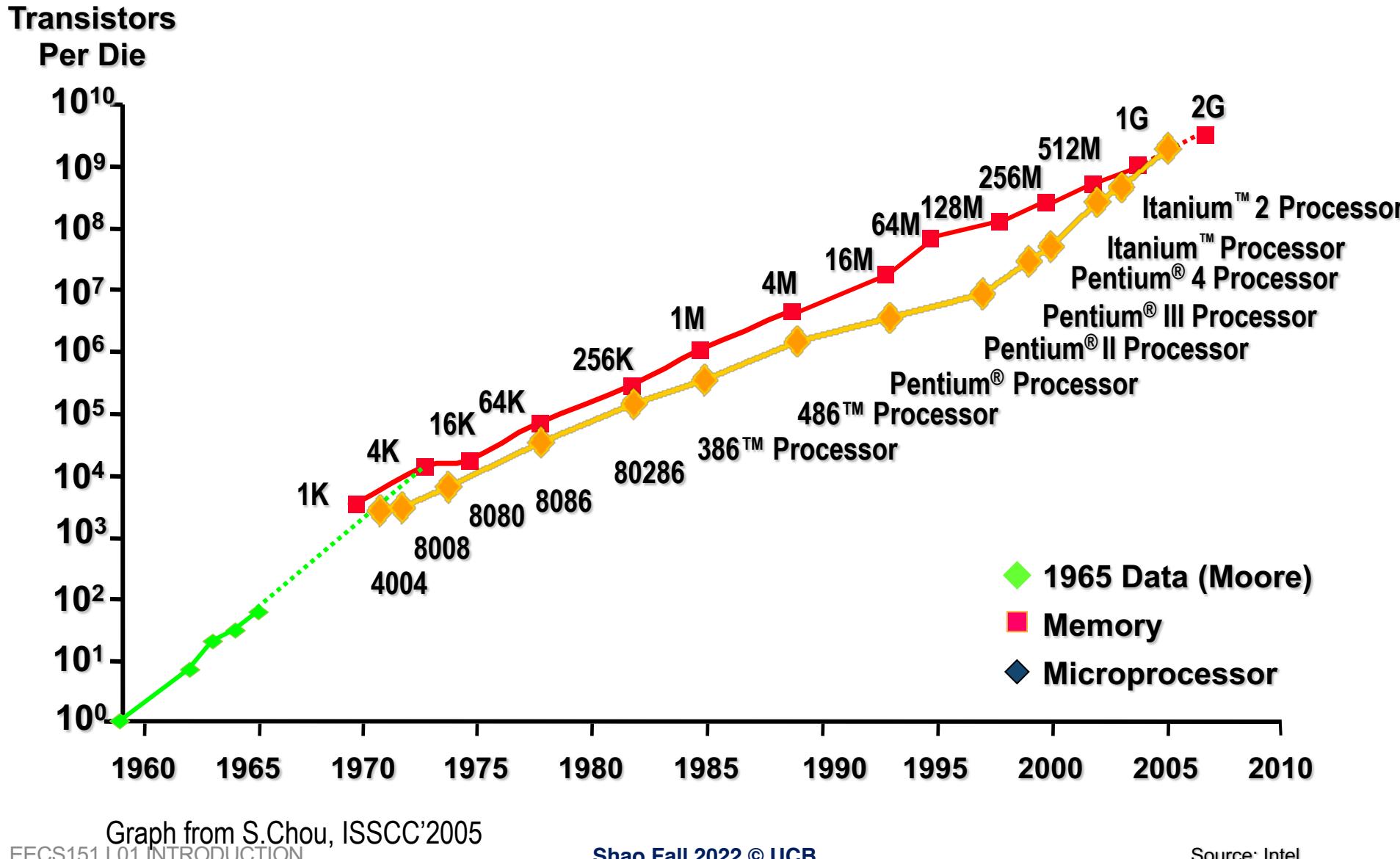
Source: Intel

◆ 1965 Data (Moore)

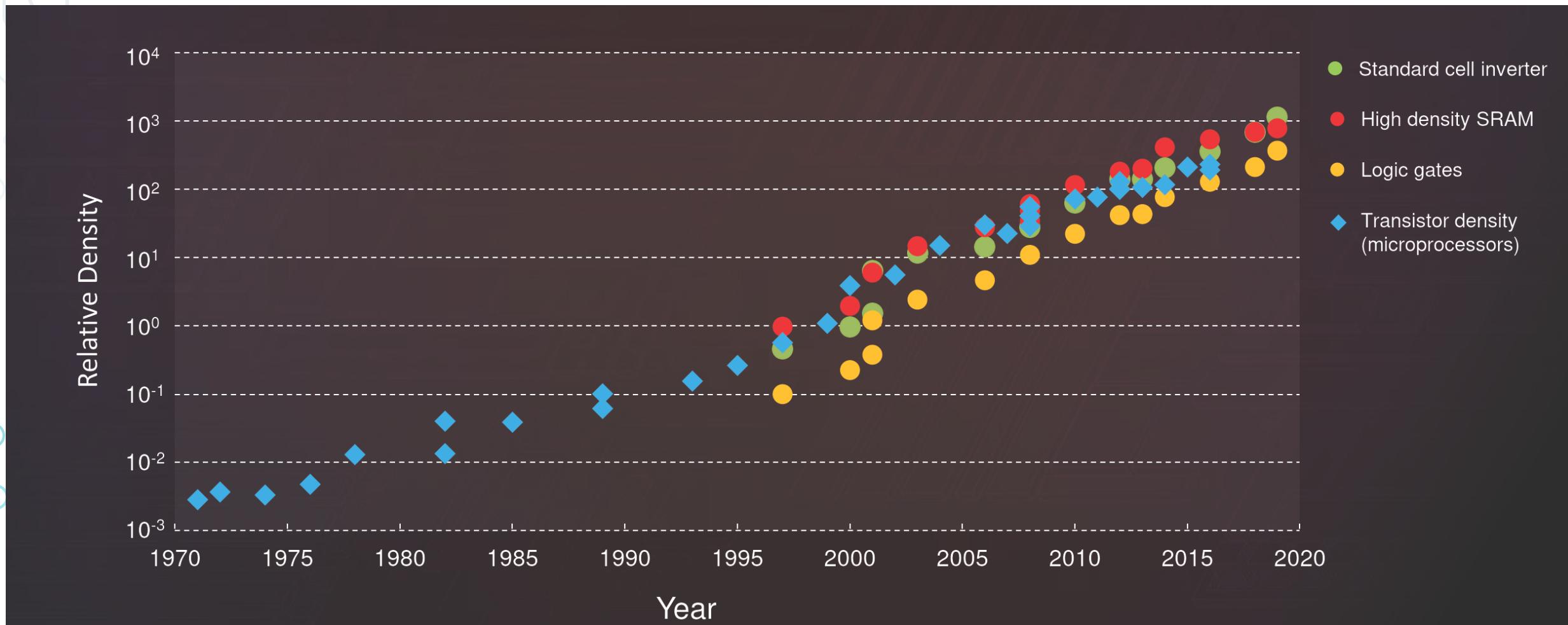
"Reduced **cost is one of the big attractions of integrated electronics, and the cost advantage continues to increase as the technology evolves toward the production of larger and larger circuit functions on a single semiconductor substrate."**

Electronics, Volume 38, Number 8, April 19, 1965

Moore's Law - 2005

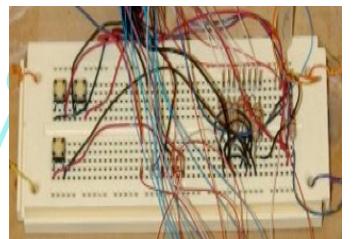


Transistor Counts

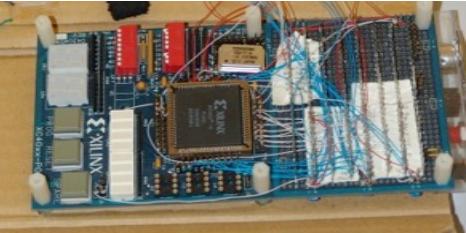


H.-S. P. Wong, HotChips, August 2019.

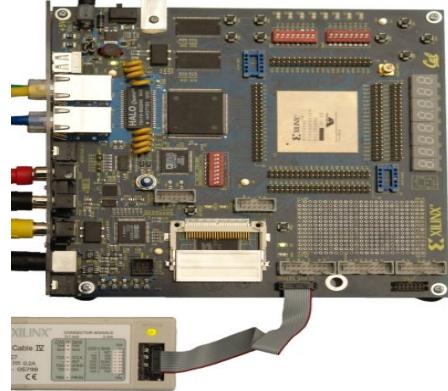
CS150/EECS151 Project Complexity



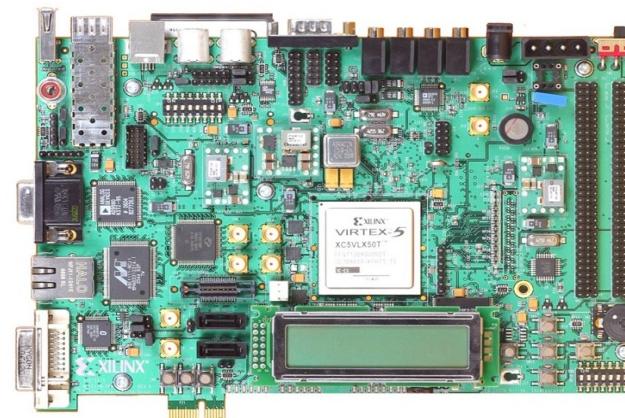
1980 Pong game
10's of logic gates



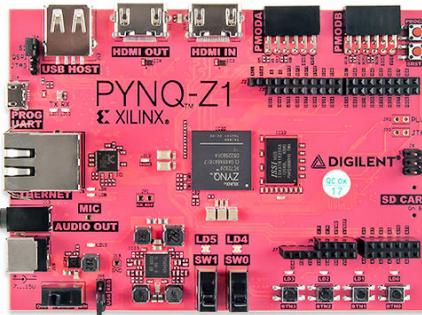
1995 MIDI synthesizer
1000's of logic gates



2000-2010 eTV tuner
10K's logic gates



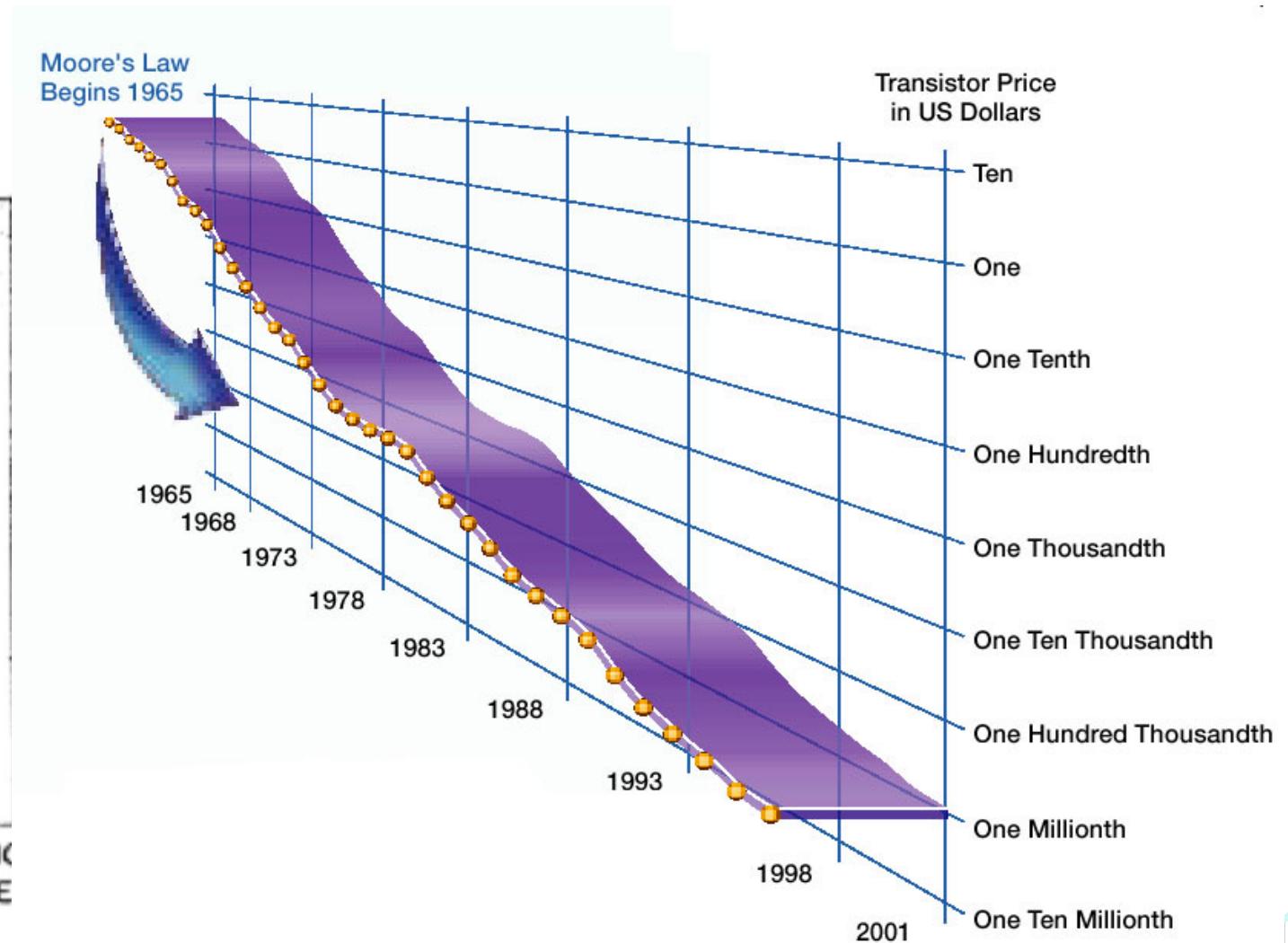
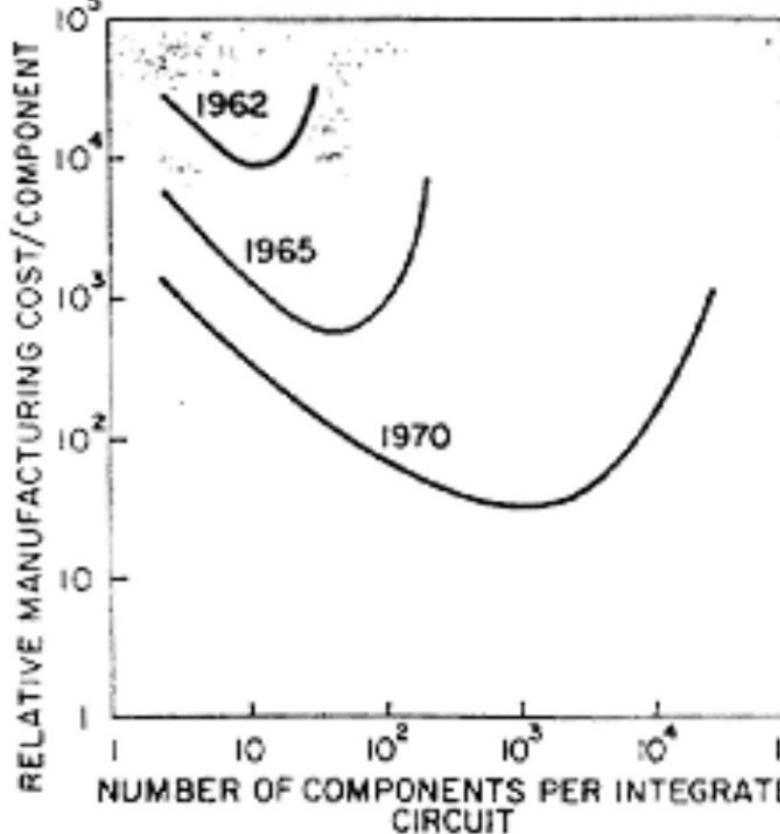
2010-2017 MIPS CPU or BYO
1M logic gates



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

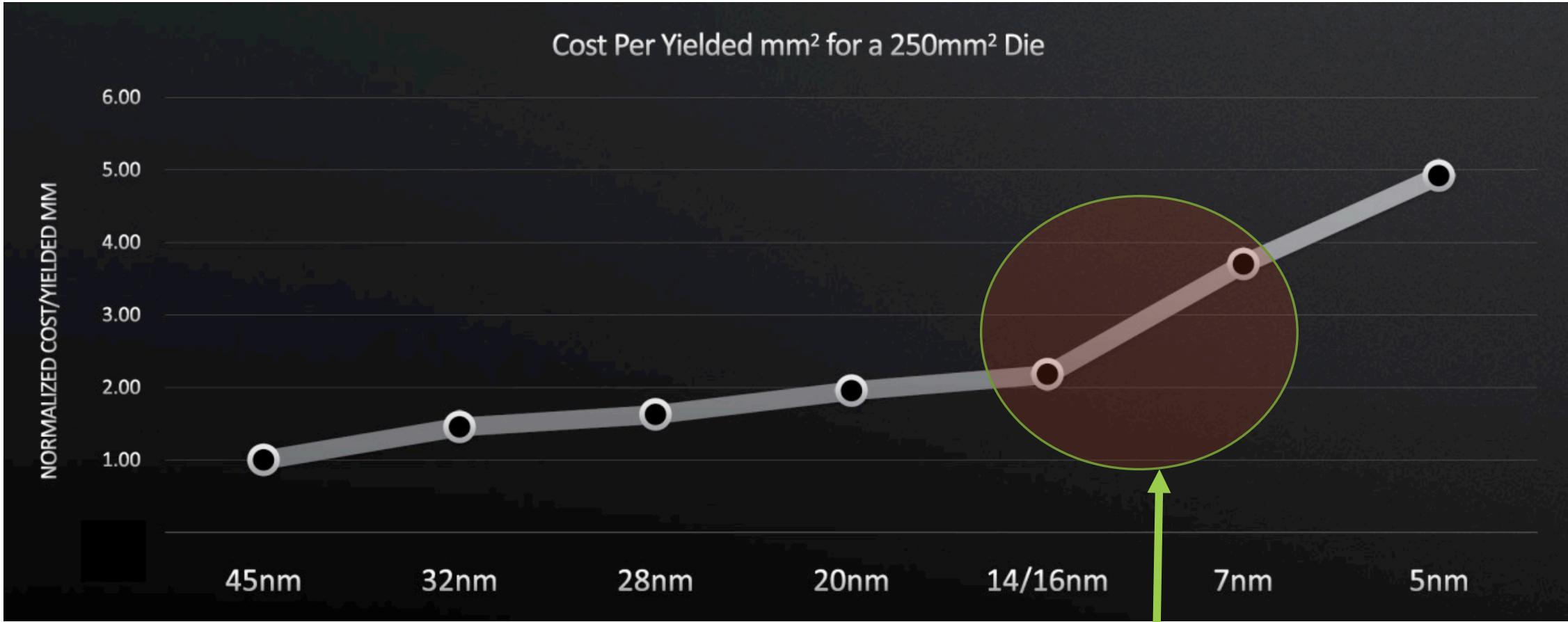
The Key: Cost

Moore's 1965 paper:



Moore's Law ends when cost reduction stops

Recent Cost Trend



L. Su, HotChips, August 2019.

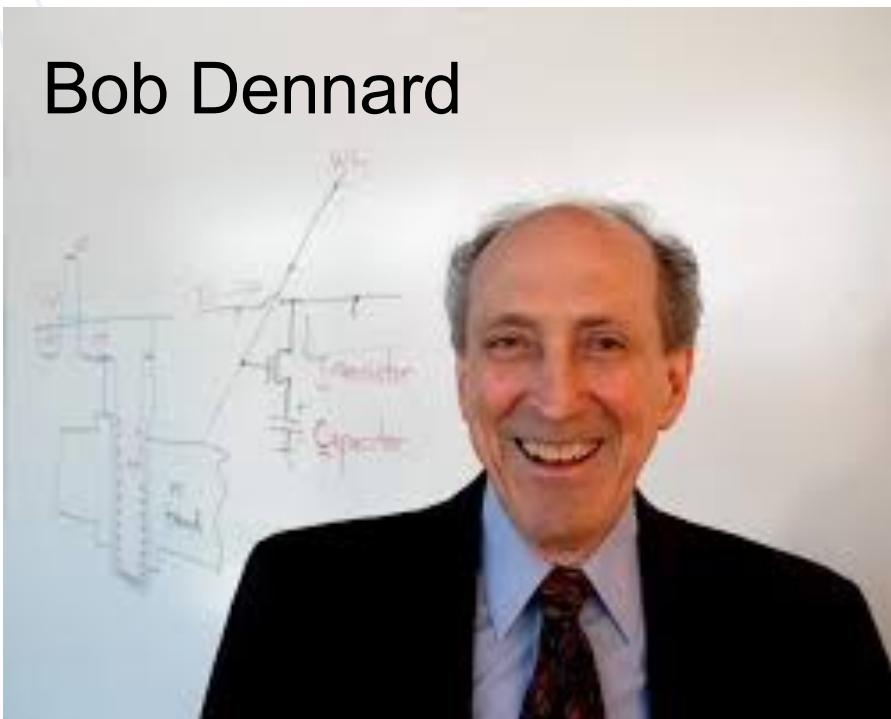
Cost nearly doubled!



The Other Trends Power!

Dennard Scaling (1974)

Bob Dennard



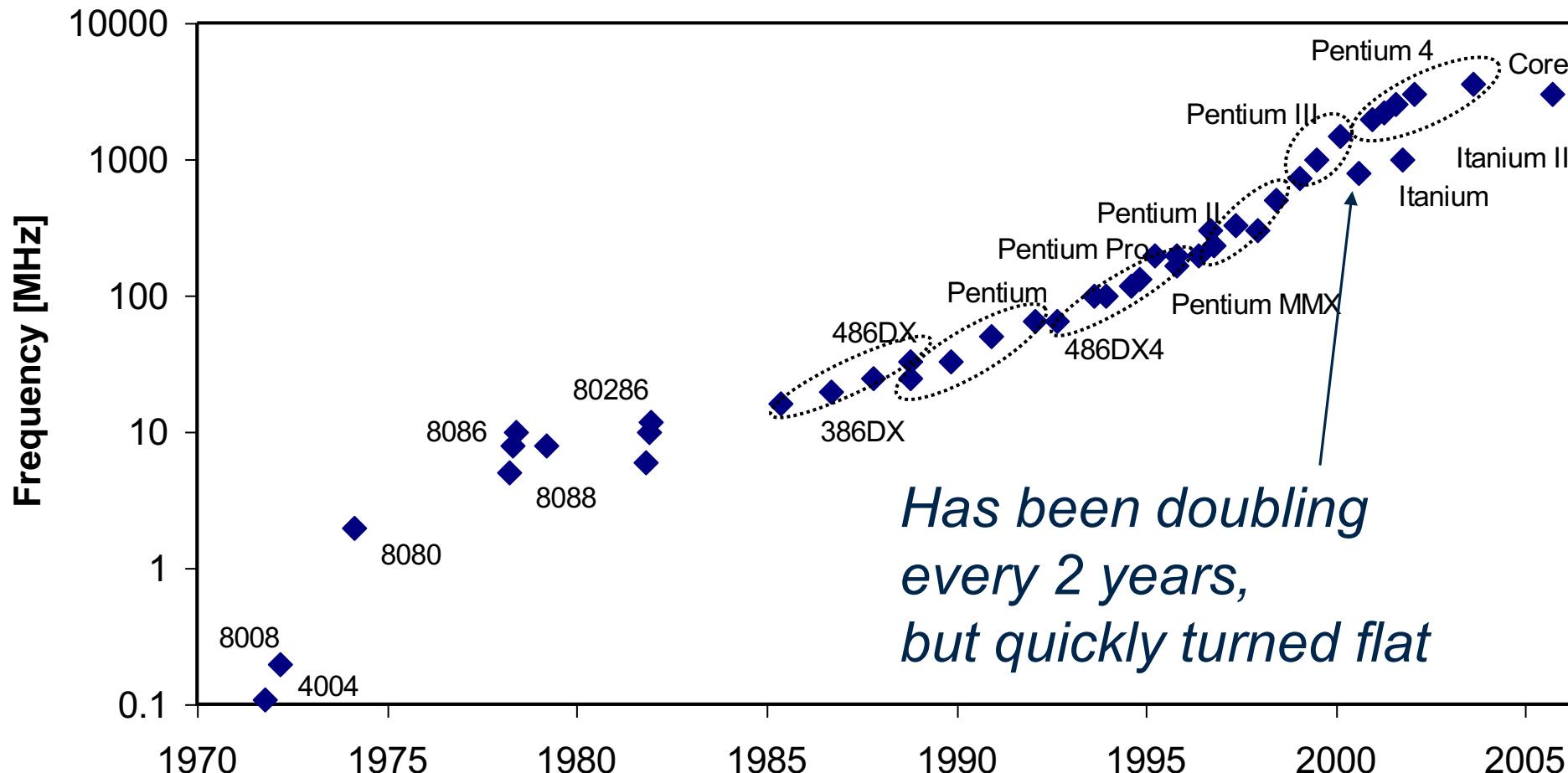
- Voltages (and currents) should be scaled proportionally to the dimensions of the transistor
- And, in theory, power *density* constant!

Device or Circuit Parameter	Scaling Factor
Device dimension t_{ox}, L, W	$1/k$
Doping concentration N_a	k
Voltage V	$1/k$
Current I	$1/k$
Capacitance eA/t	$1/k$
Delay time per circuit VC/I	$1/k$
Power dissipation per circuit VI	$1/k^2$
Power density VI/A	1

R.H. Dennard, F. Gaenslen, H.-N. Yu, L. Rideout, E. Bassous, A. LeBlanc, Andre
"Design of ion-implanted MOSFET's with very small physical dimensions," IEEE
Journal of Solid State Circuits. SC-9 (5), 1974.

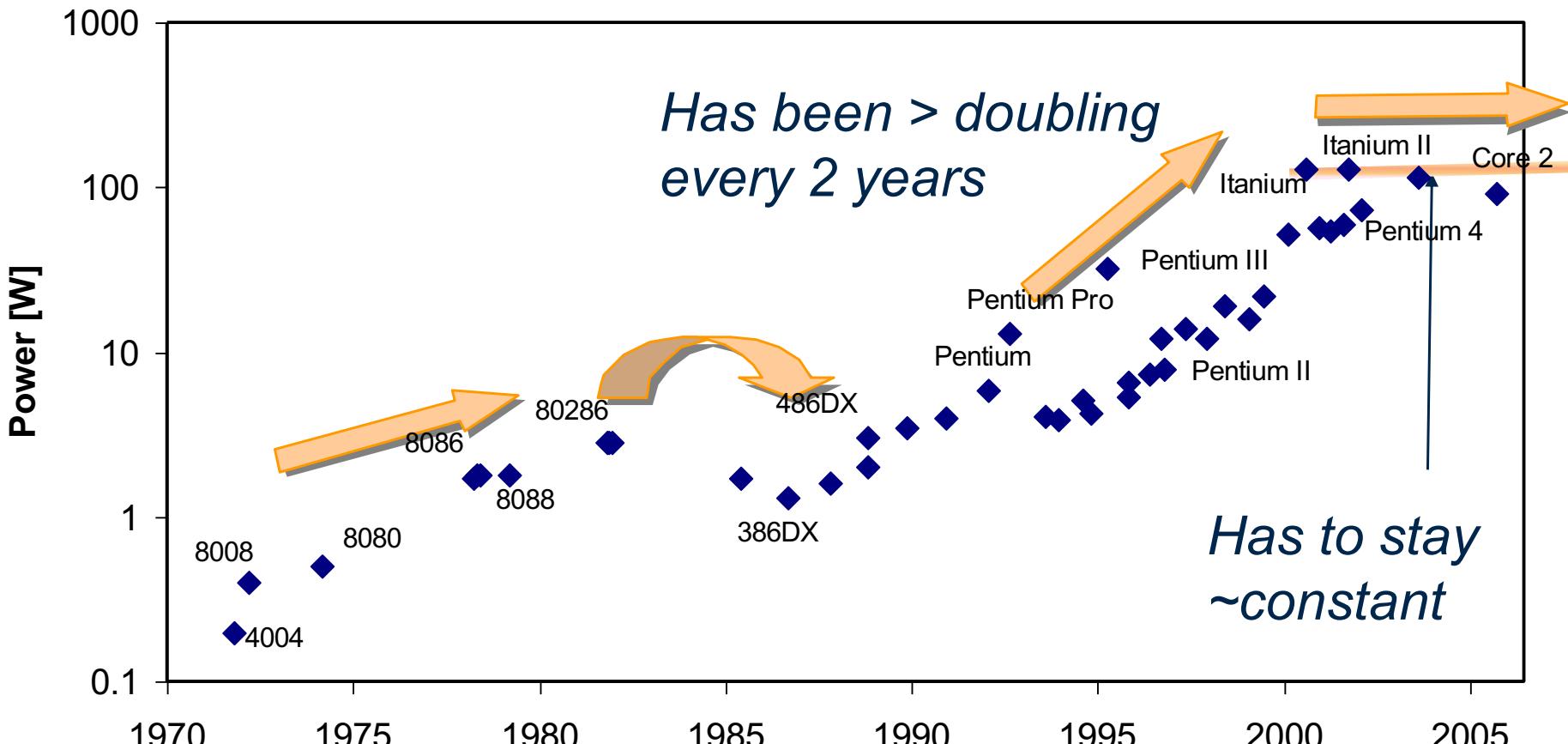
Frequency

Frequency Trends in Intel's Microprocessors

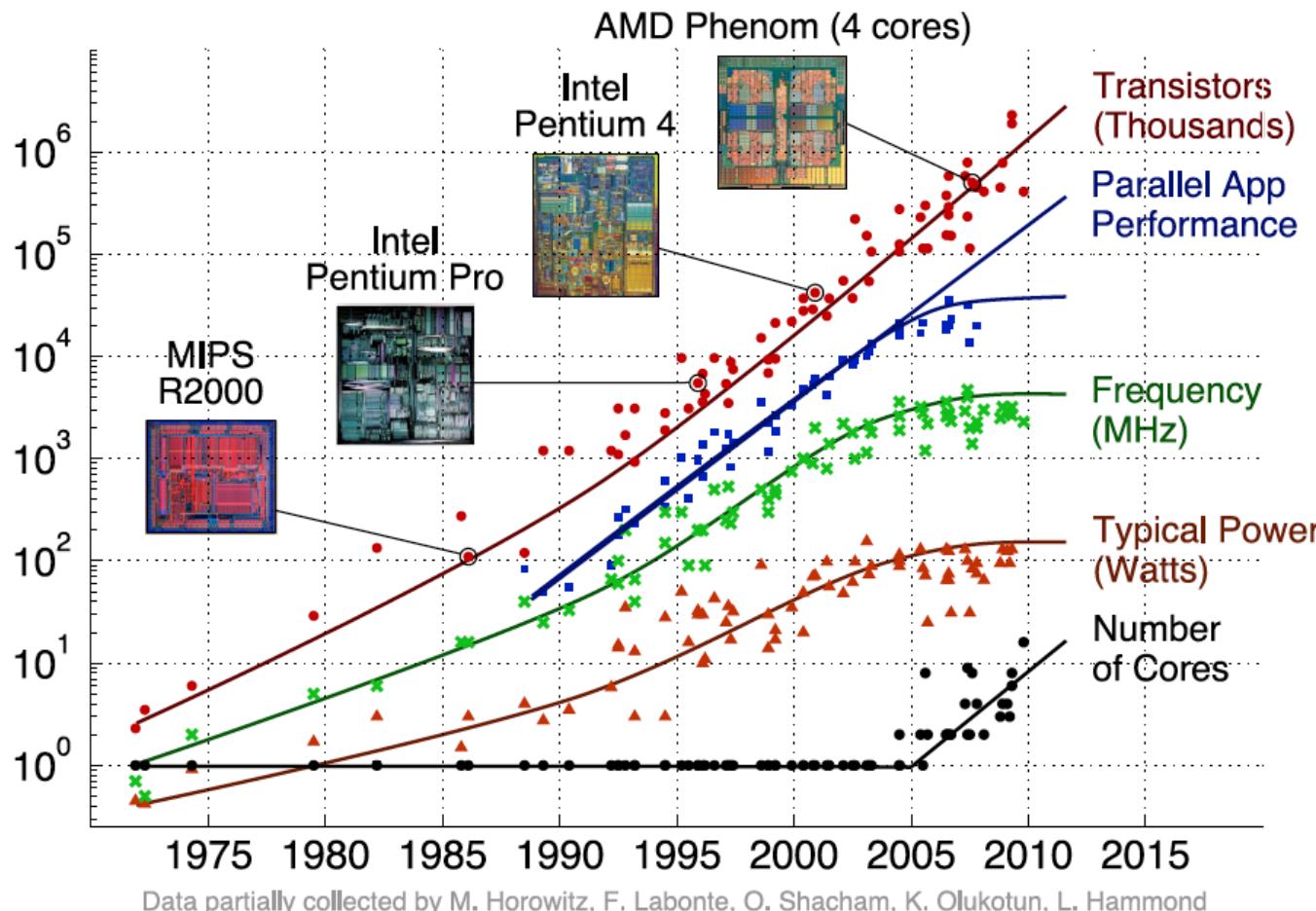


Power Dissipation

Power Trends in Intel's Microprocessors



Power and Performance Trends

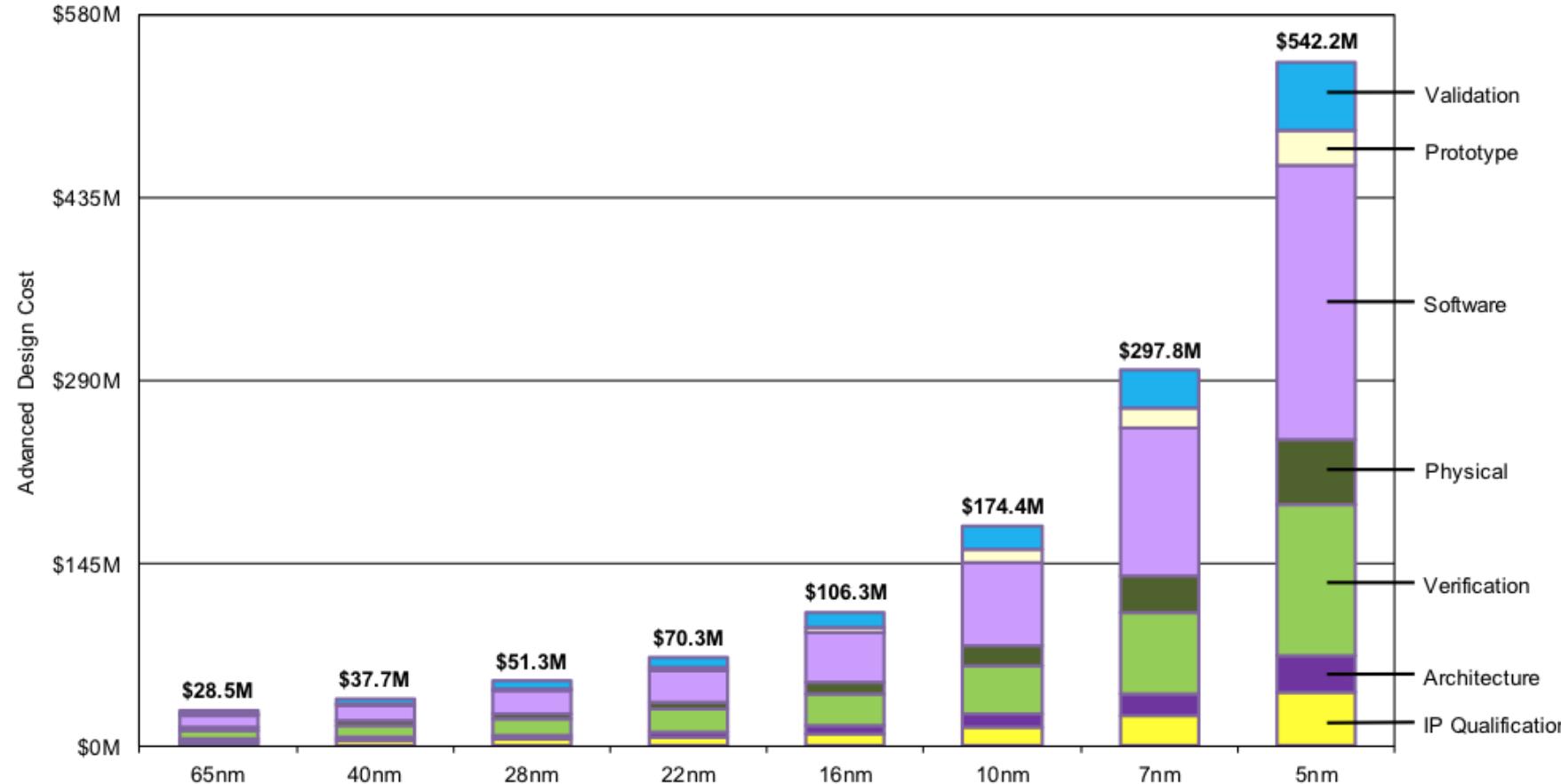


- What do we do next?



The Other Demon: Complexity and Design Costs

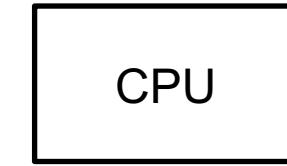
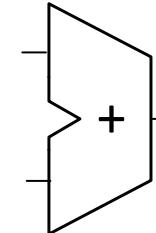
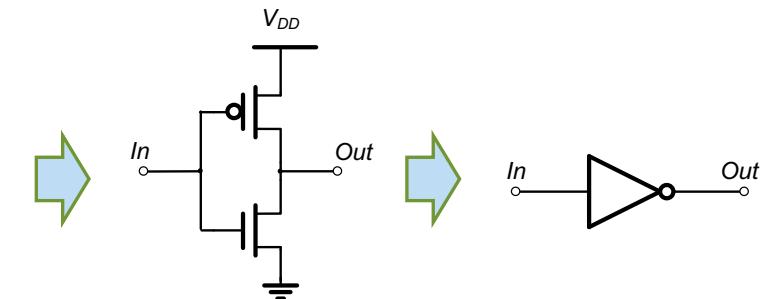
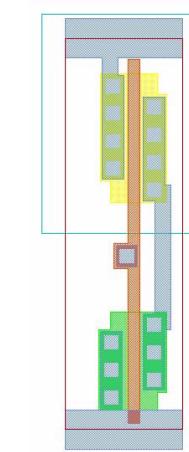
Cost Of Developing New Products



- These are non-recurring (NRE) costs, need to be amortized over the lifetime of a product
- We will attempt to dismantle this...

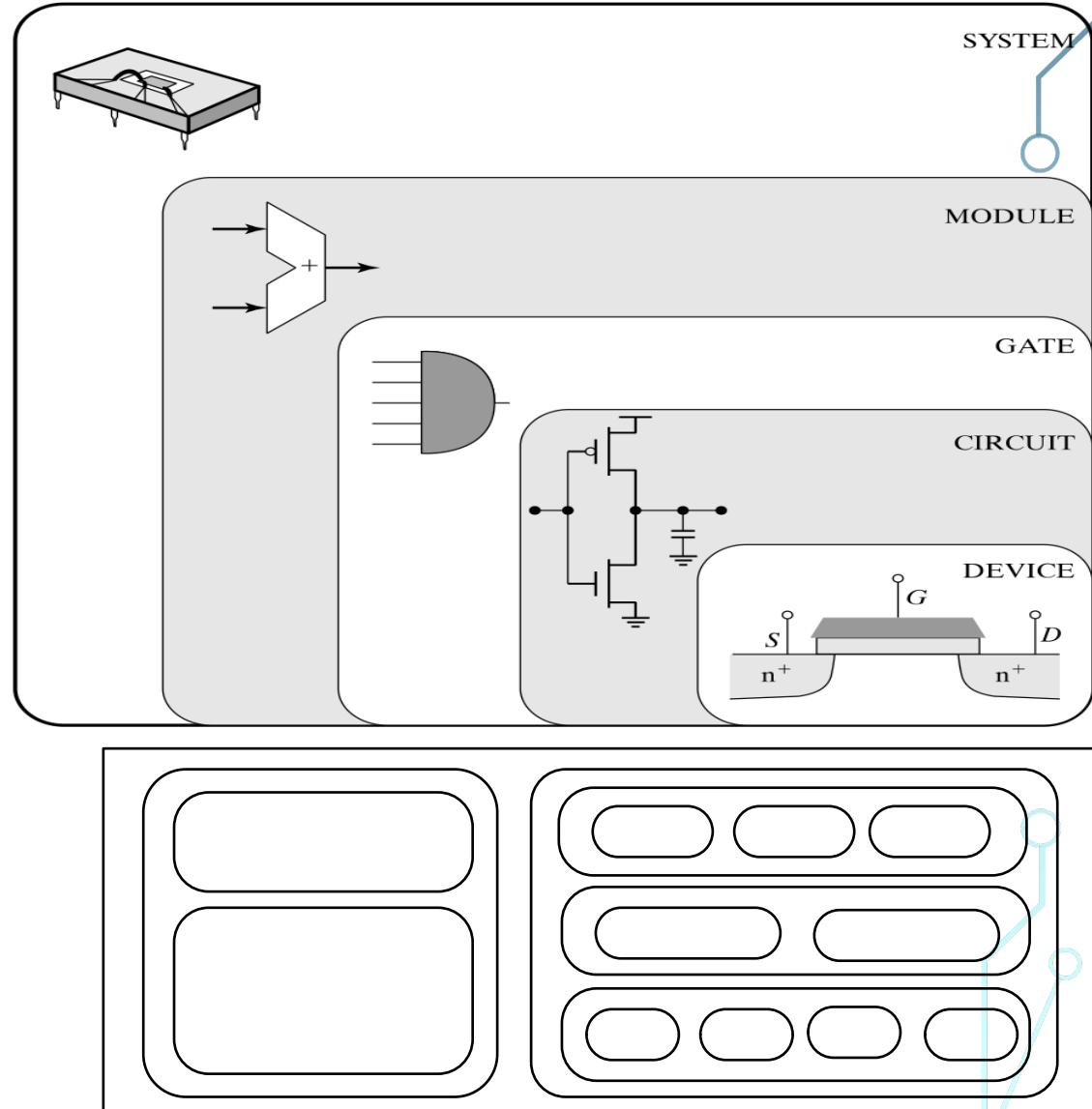
Abstraction

- How to design a Pong game
 - Hand layout
 - Gate-level design (semi custom)
 - Synthesis, place and route
 - HLS, HDL
 - “Computer, design a pong game”



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



Digital Design: What's it all about?

- Given a functional description and performance, cost, & power constraints, create 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 a bit an art as well as 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