CS 152 Computer Architecture and Engineering

Lecture 1 - Introduction

Dr. George Michelogiannakis
EECS, University of California at Berkeley
CRD, Lawrence Berkeley National Laboratory

http://inst.eecs.berkeley.edu/~cs152

Pronunciation

Miheloyannakis

(optional)

What is Computer Architecture?

Application

Gap too large to bridge in one step

(but there are exceptions, e.g. magnetic compass)

Physics

In its broadest definition, computer architecture is the *design of the abstraction layers* that allow us to implement information processing applications efficiently using available manufacturing technologies.

What is Computer Architecture?

- A set of rules and methods that describe the functionality, organization and implementation of computer systems.

- Computer Architecture is the science and art of selecting and interconnecting hardware components to create computers that meet functional, performance and cost goals.

- Computer architecture acts as the intermediate between programmers and devices (e.g., VLSI).

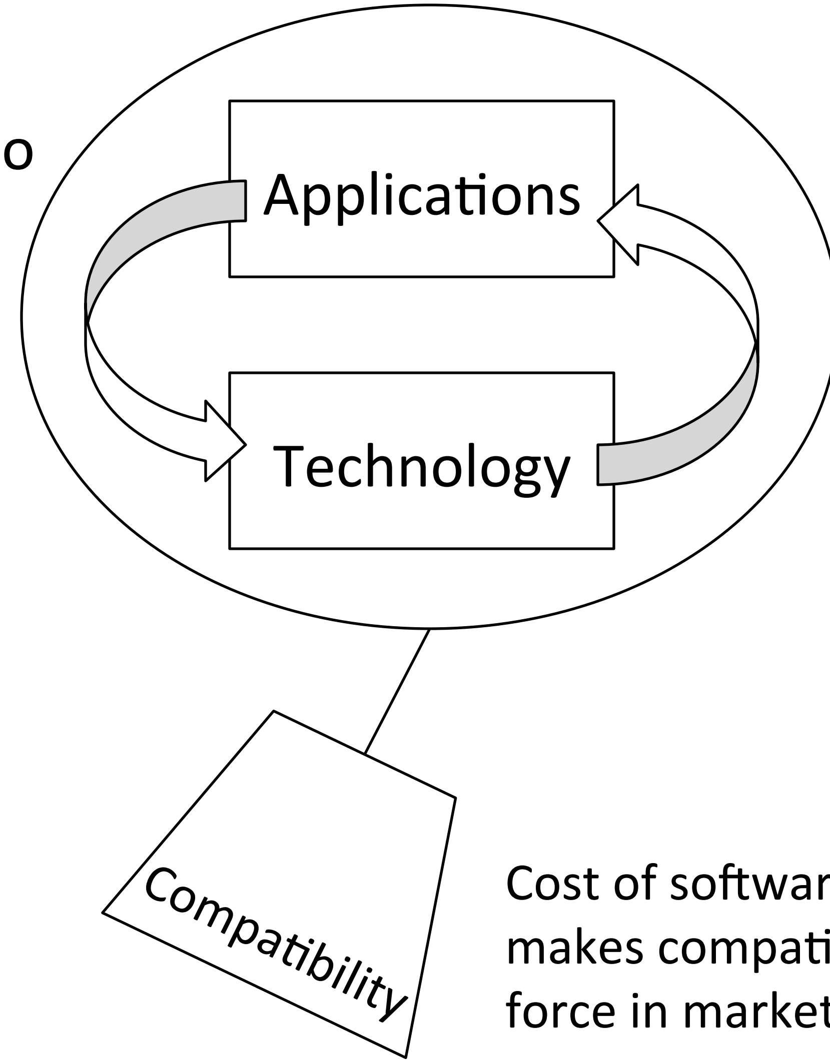
- **What are you here to learn?**

Abstraction Layers in Modern Systems

Layer	UCB EECS Courses
Application	
Algorithm	CS170
Programming Language	CS164
Operating System/Virtual Machines	CS162
Instruction Set Architecture (ISA)	CS152
Microarchitecture	CS152
Gates/Register-Transfer Level (RTL)	CS150
Circuits	EE141
Devices	EE143
Physics	

Architecture Continually Changing

Applications suggest how to improve technology, provide revenue to fund development



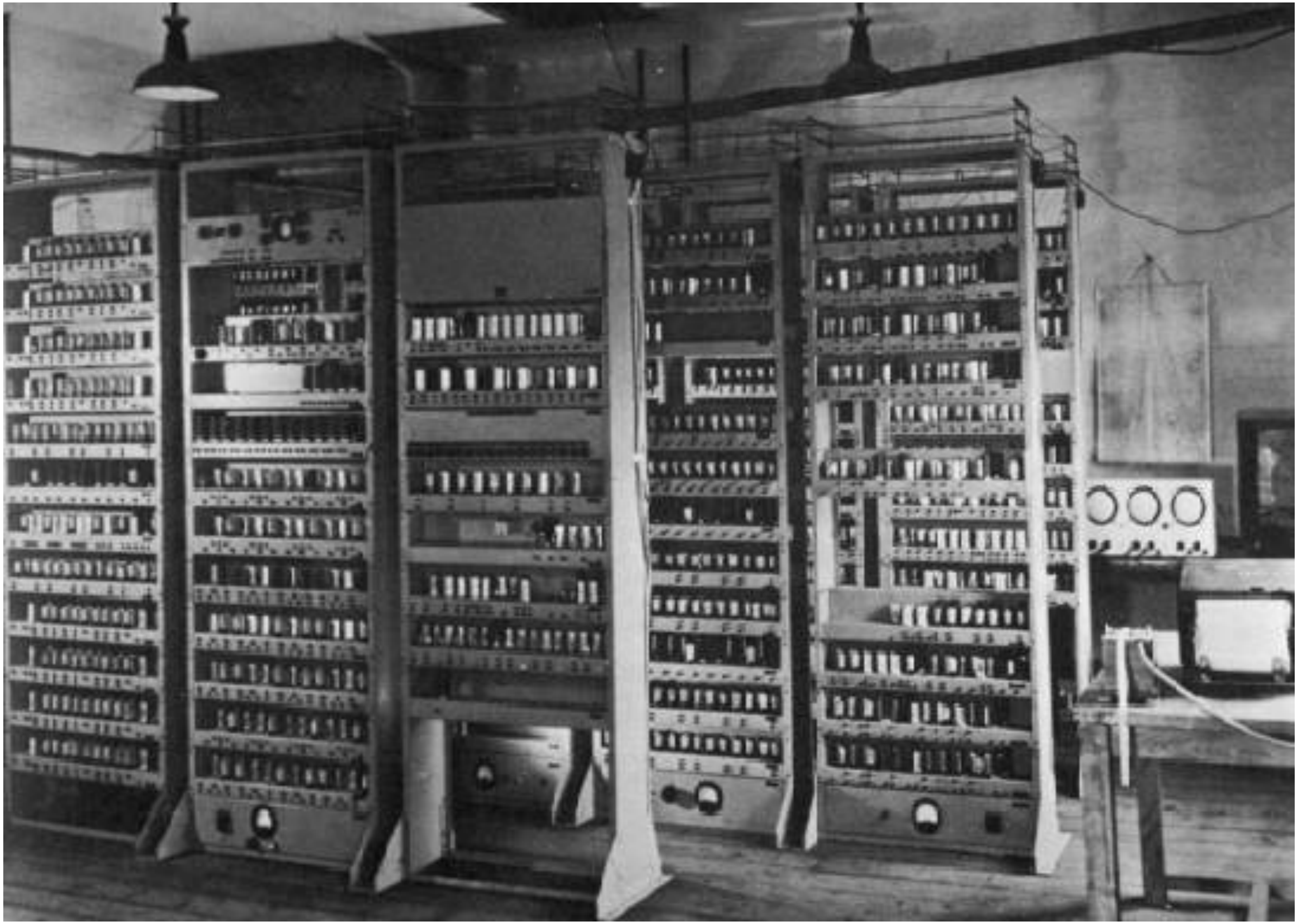
Improved technologies make new applications possible

Cost of software development makes compatibility a major force in market

Example: x86 Backwards Compatibility

- Intel's 8086 was released in 1978 with ~50 instructions
- Today, x86 has ~650 with all extensions
  - Most are rarely emitted by compilers

Computing Devices Then…

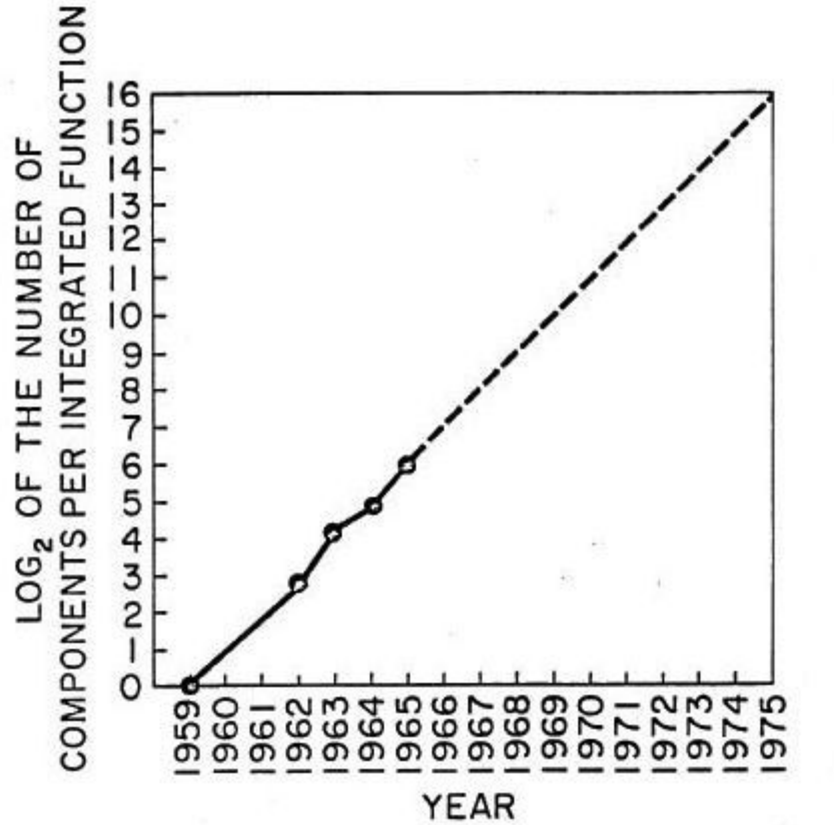


EDSAC, University of Cambridge, UK, 1949

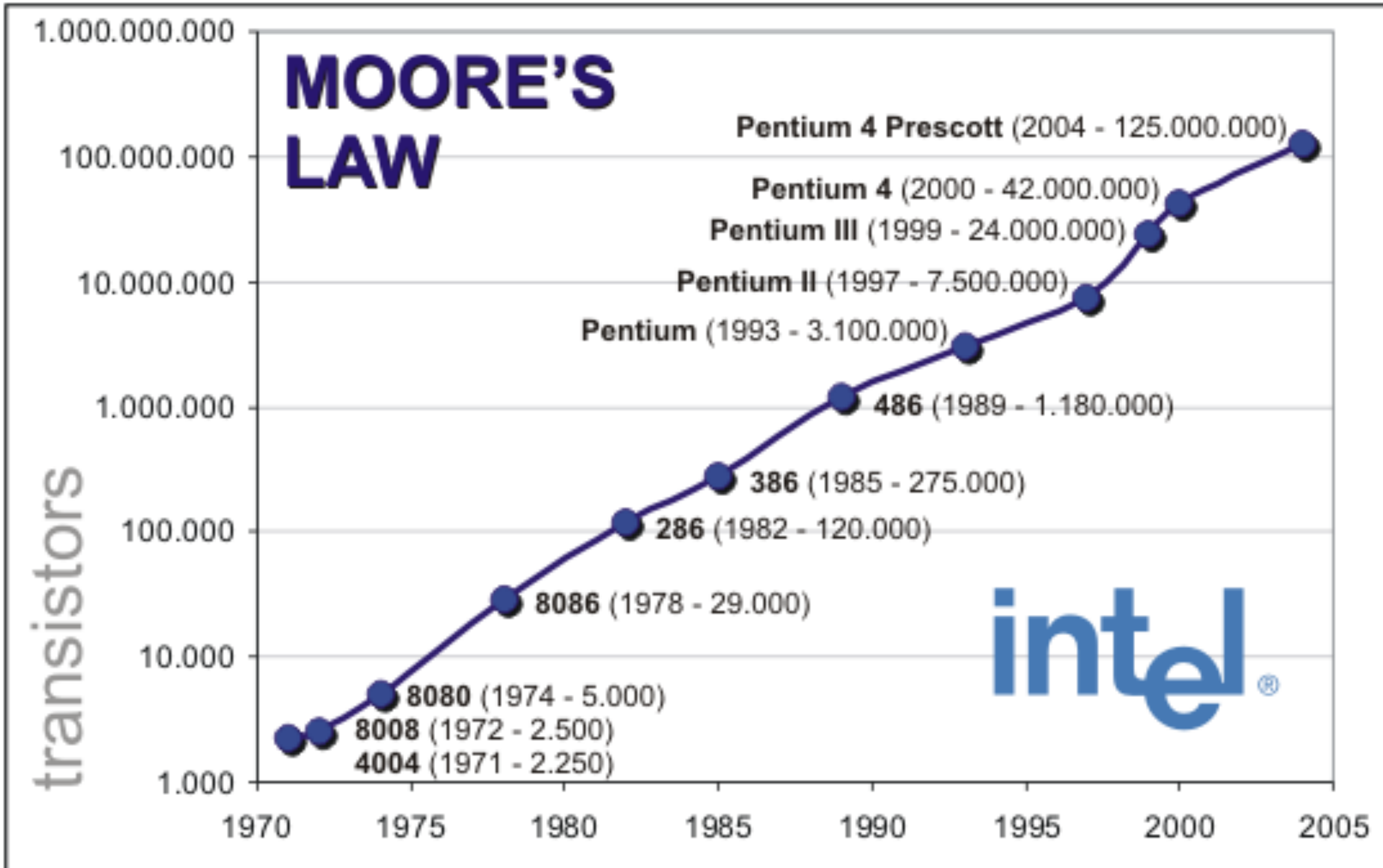
Computing Devices Now

Moore’s Law

- The observation that, over the history of computing hardware, the number of transistors in a dense integrated circuit (chip) has doubled approximately every two years.



Design Complexity

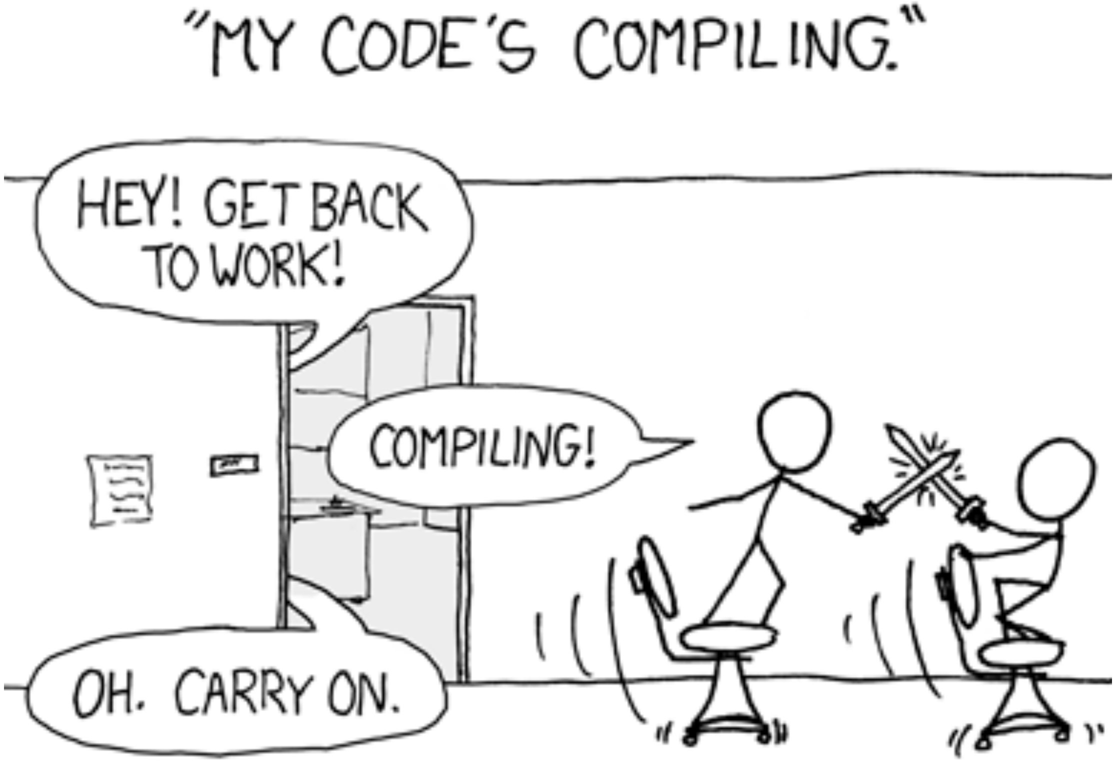


Design Capacity

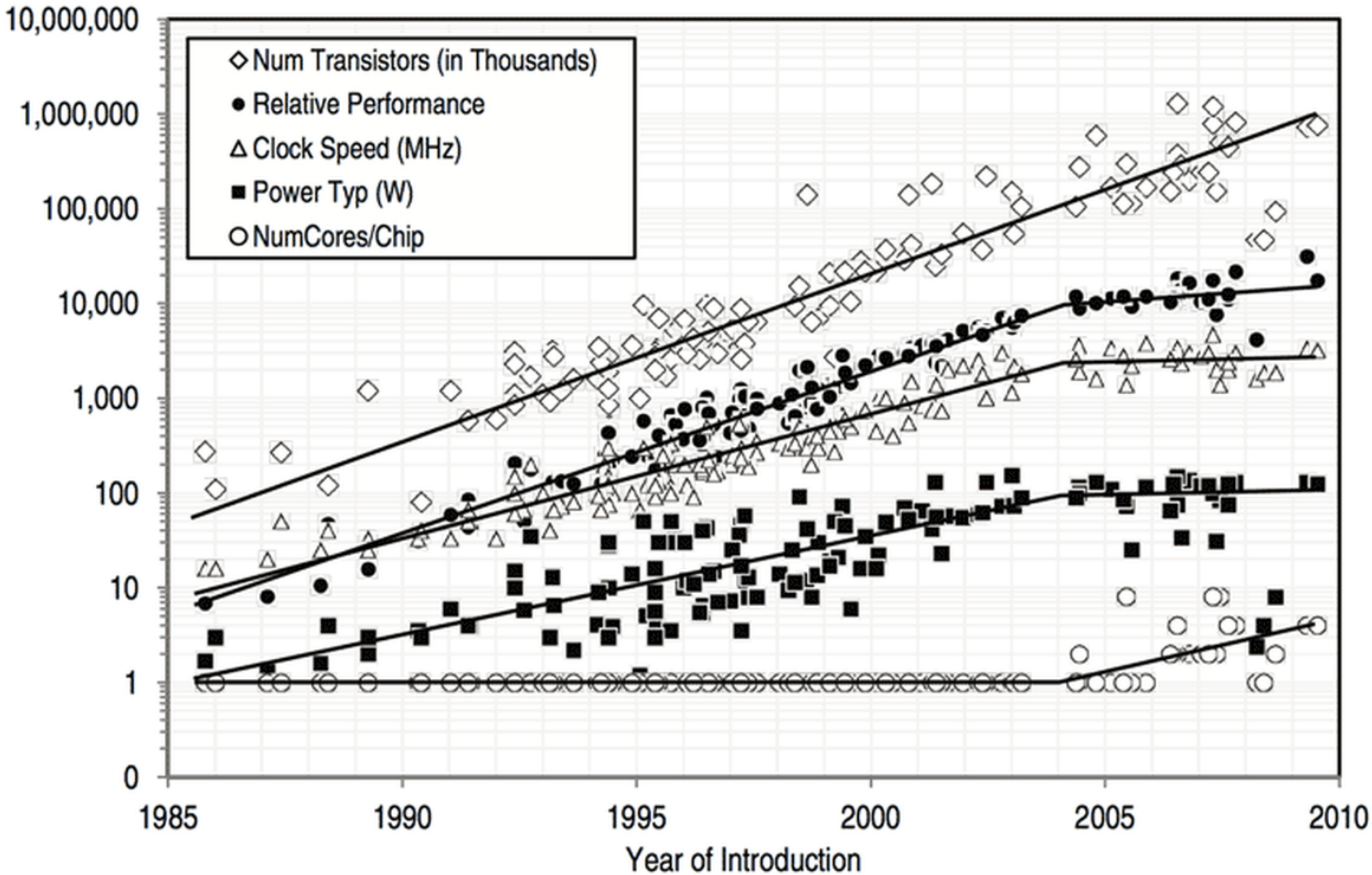
- In 1978, Intel could design a chip (8086) with 29,000 transistors
- In 2012, 2,104 million (Ivy Bridge)
- Rocket (RISC-V) which you'll be using has 75+ million transistors
- Does humanity get smarter with time?

Computer Architects Then

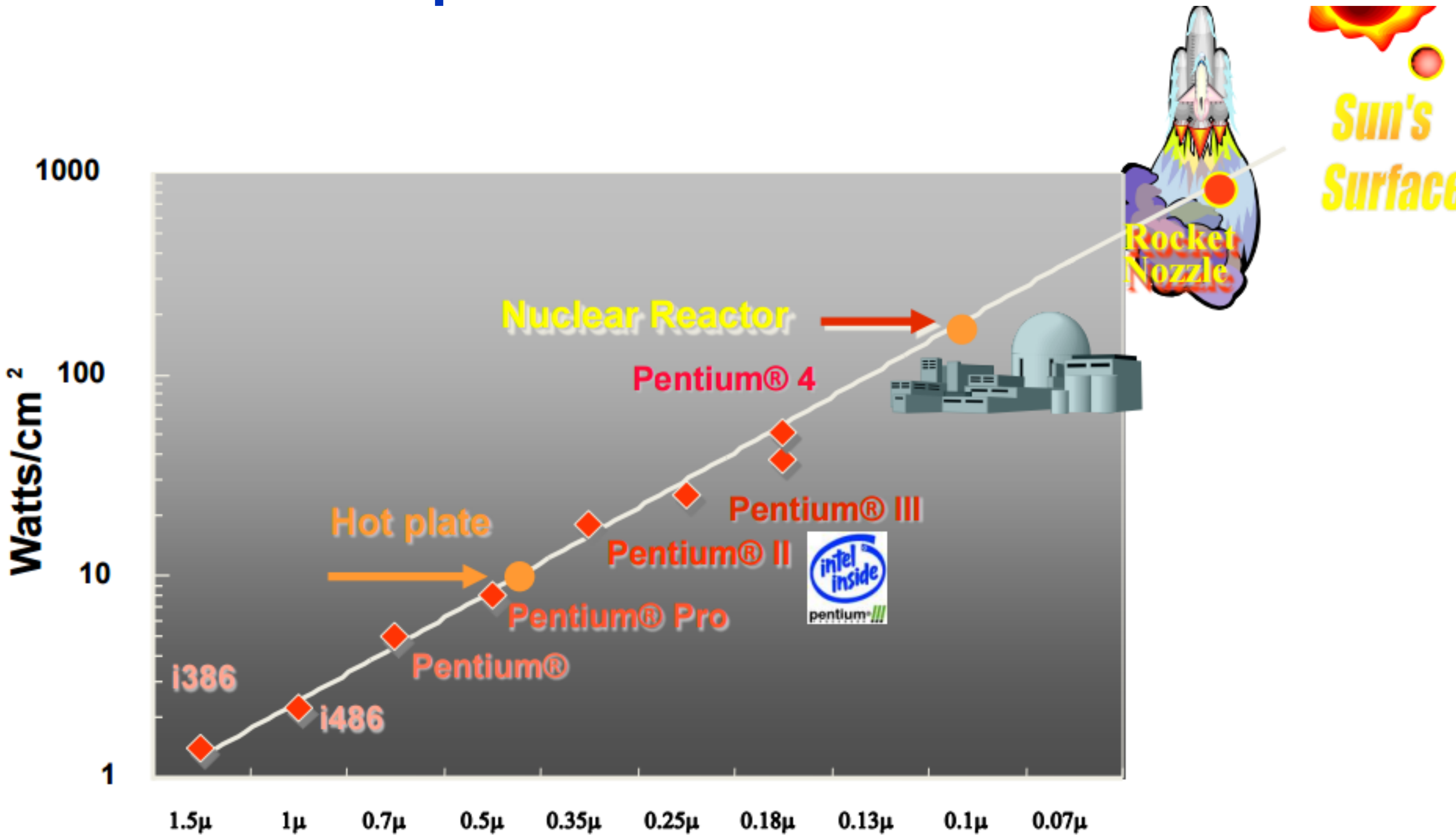
Computer Architects Now



Technology Trends

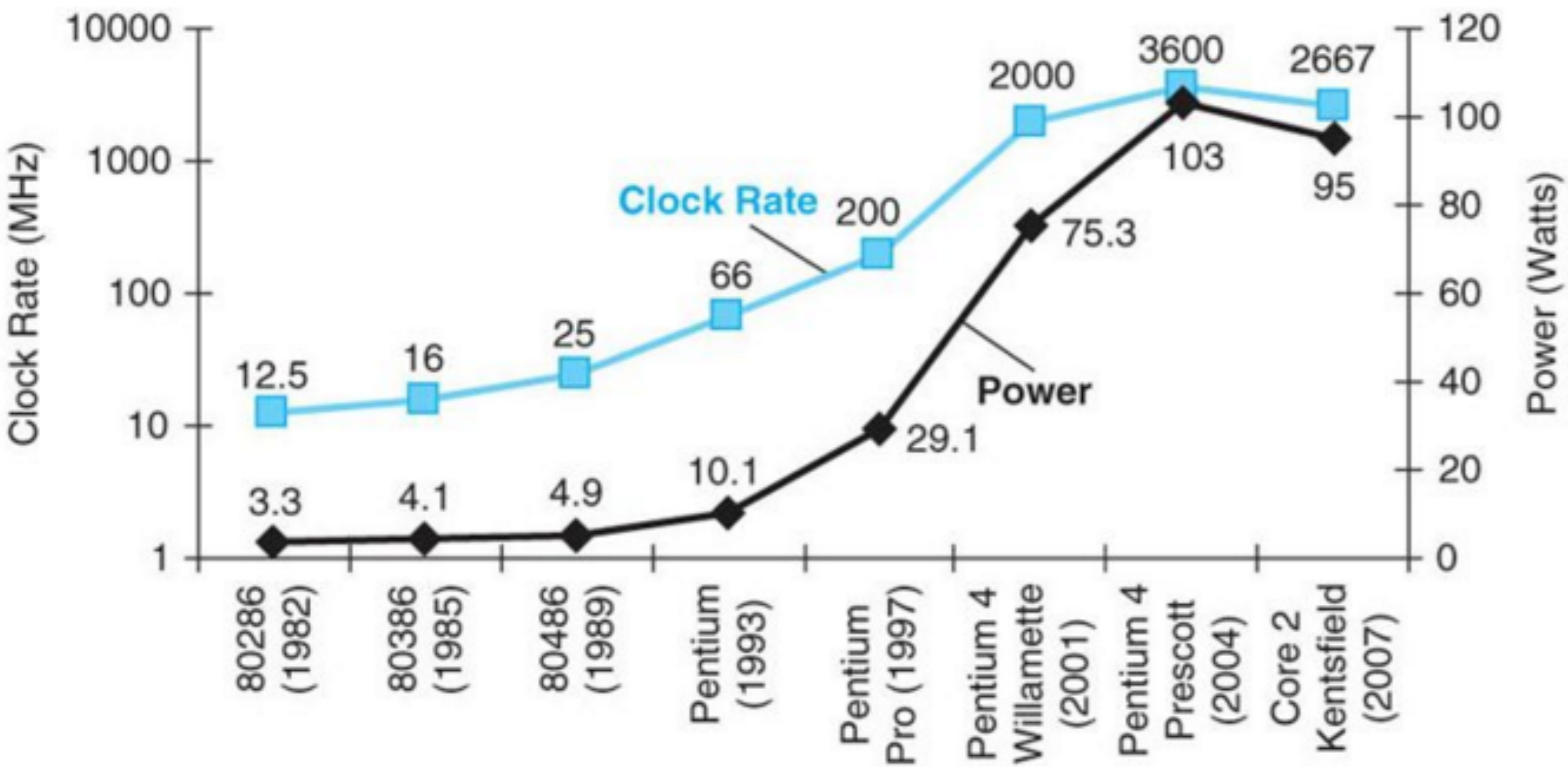


Power Dissipation



* "New Microarchitecture Challenges in the Coming Generations of CMOS Process Technologies" – Fred Pollack, Intel Corp. Micro32 conference key note - 1999.

Power Wall in Modern Processors



While at the same time chips keep getting larger.

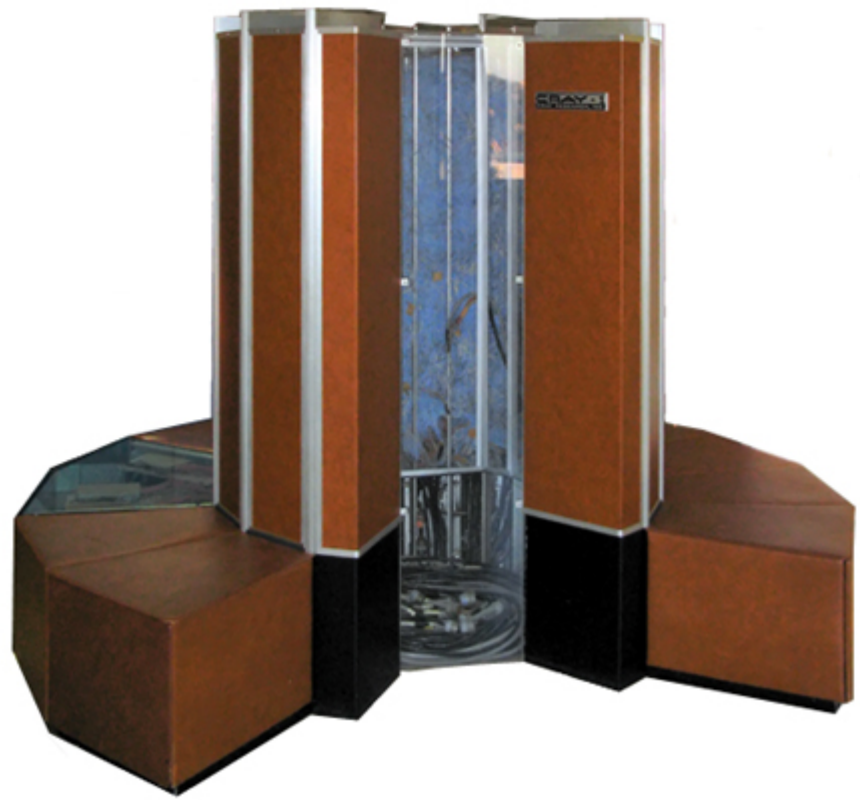
Therefore, not all of the chip can be powered on at the same time

The End of the Uniprocessor Era

Single biggest change in the history of computing systems

We Went From This

- Cray-1
- **Single** processor

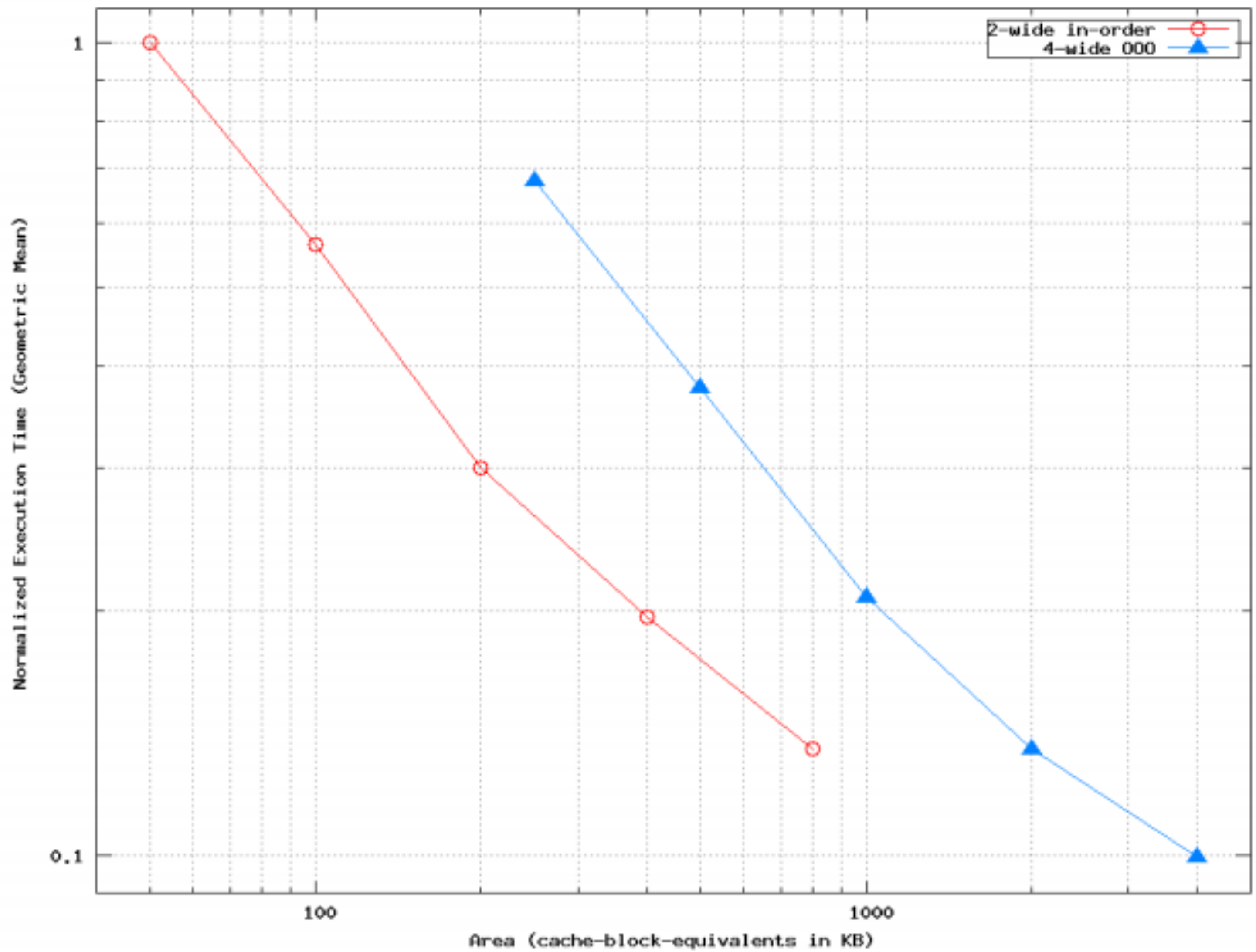


To This

- Titan, an XK7 supercomputer at Oak Ridge National Laboratory (Cray XT3) (299,008 AMD Opteron cores)

J. Huh, D.C. Burger, and S.W. Keckler. Exploring the Design Space of Future CMPs. In International Conference on Parallel Architectures and Compilation Techniques (PACT), September, 2001

Result: Simple Cores



Result: Simple Cores

YOU ARE HERE: SYSTEMS | CPU | ENTERPRISE | NEWS

Supercomputer maker Cray looking at 64-bit ARM processors for HPC

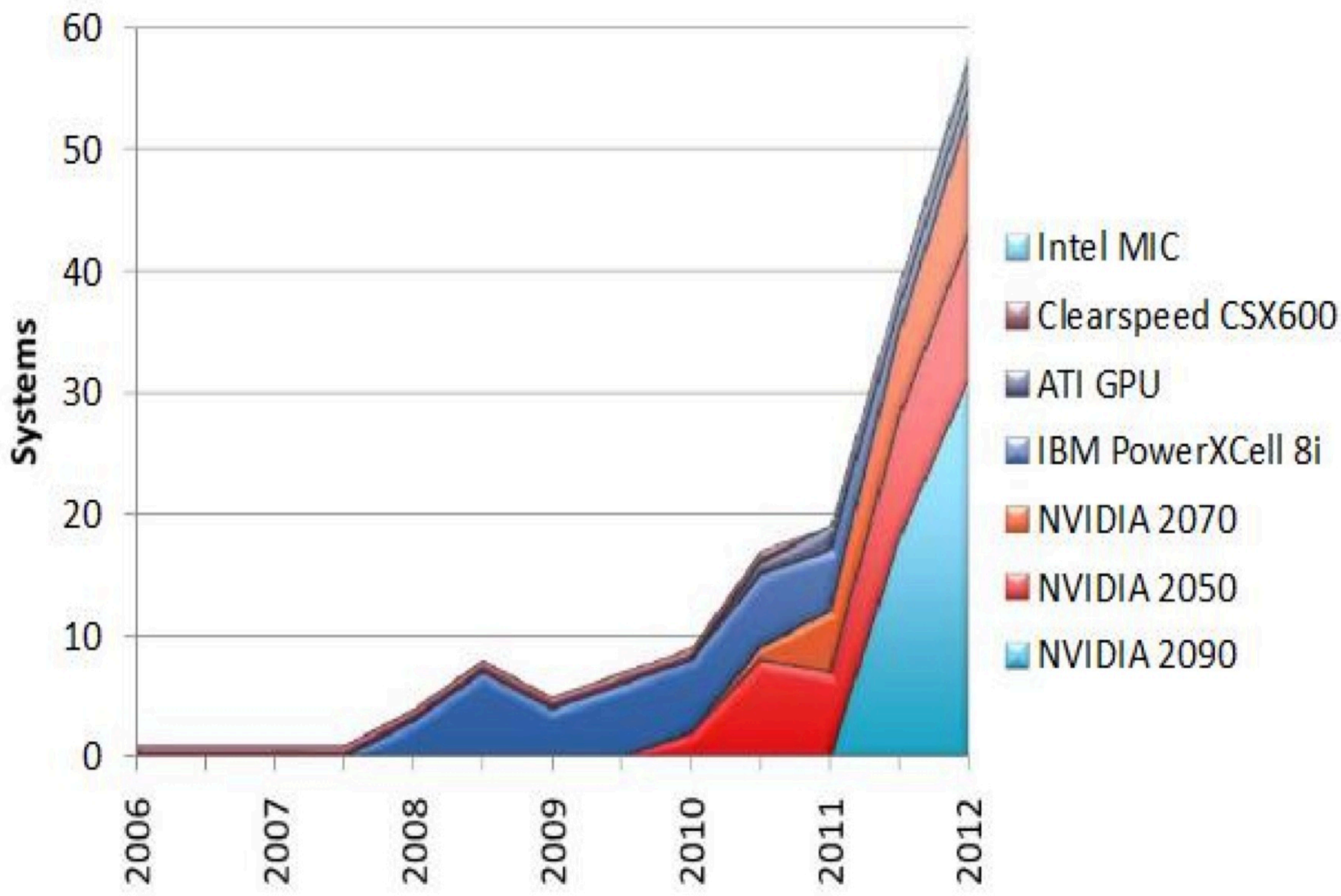
by **Mark Tyson** on 18 November 2014, 10:05

Tags: ARM (LON:ARM)
Quick Link: HEXUS.net/qaclnn
Add to My Vault:

Iconic Supercomputer maker Cray Inc. is exploring the possibilities of and evaluating *"alternative processor design points,"* according to a **press release** issued to coincide with the 2014 Supercomputing Conference in New Orleans. 64-bit ARM chips are said to be under scrutiny for inclusion in future supercomputer and data analytics systems made by the firm.

Cray was recently awarded an R&D contract by the United States Department of Energy (DOE) and it is hoped that ARM's energy efficiency leadership can be applied to Cray supercomputer designs. The DOE contract will see supercomputers made under a program called FastForward 2 which will be used in scientific research and the National Nuclear Security Administration.

Result: Specialization



Before That: Dennard Scaling

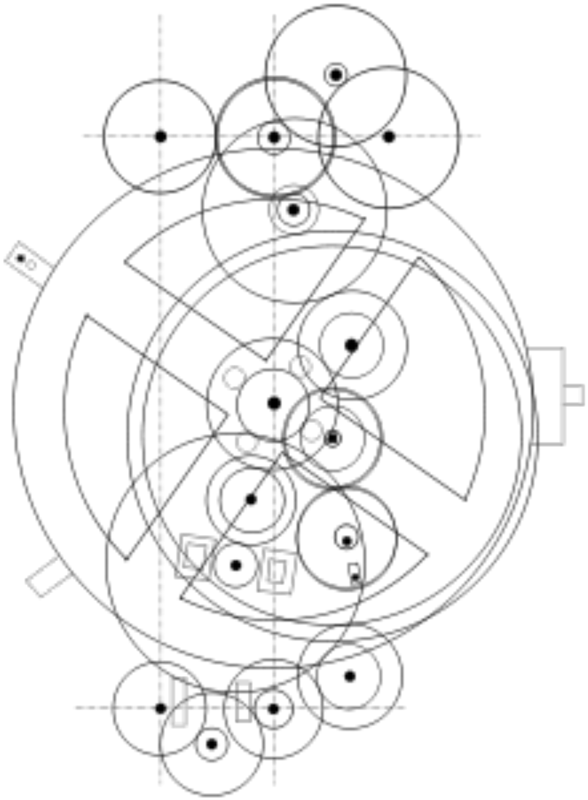
- Power = A x C x F x V^2
  - A: Activity factor
  - C: Capacitance
  - F: Frequency
  - V: Voltage
- Capacitance is related to area
  - So, as the size of the transistors shrunk, and the voltage was reduced, circuits could operate at higher frequencies at the same power
- But leakage current and threshold voltage of transistors set a lower bound for voltage
- Transistors get smaller, their power is the same -> Power density increases

Learn from the mistakes of others

A LITTLE HISTORY

Antikythera Mechanism

- Found in a Greek ship believed to have sank around 80 B.C.
- It accurately predicted lunar and solar eclipses, as well as solar, lunar and planetary positions
  - Size: 8 inches across



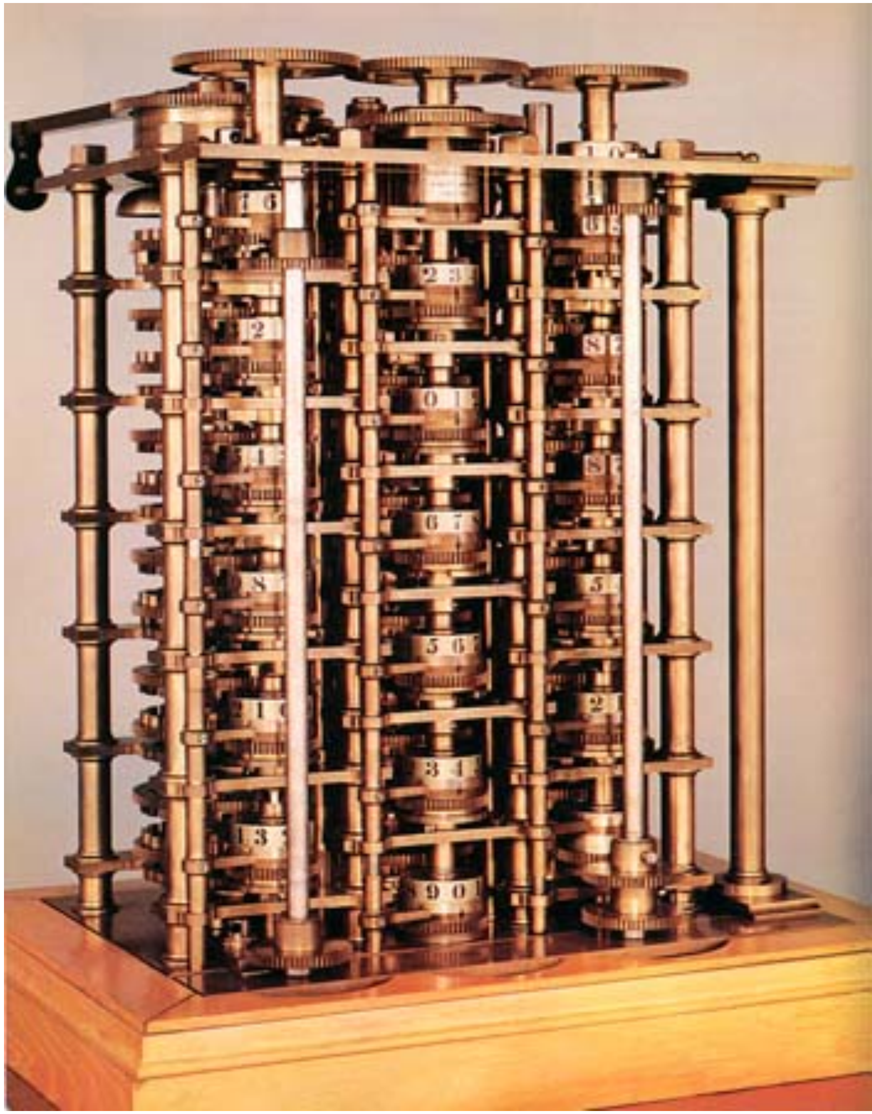
Difference Engine

1855. Can compute any 6th degree polynomial by calculating the difference between 2D matrix elements

Speed: 33 to 44 32-digit numbers per minute!

n	0	1	2	3	4
d2(n)			2	2	2
d1(n)		2	4	6	8
f(n)	41	43	47	53	61

Now the machine is at the Smithsonian



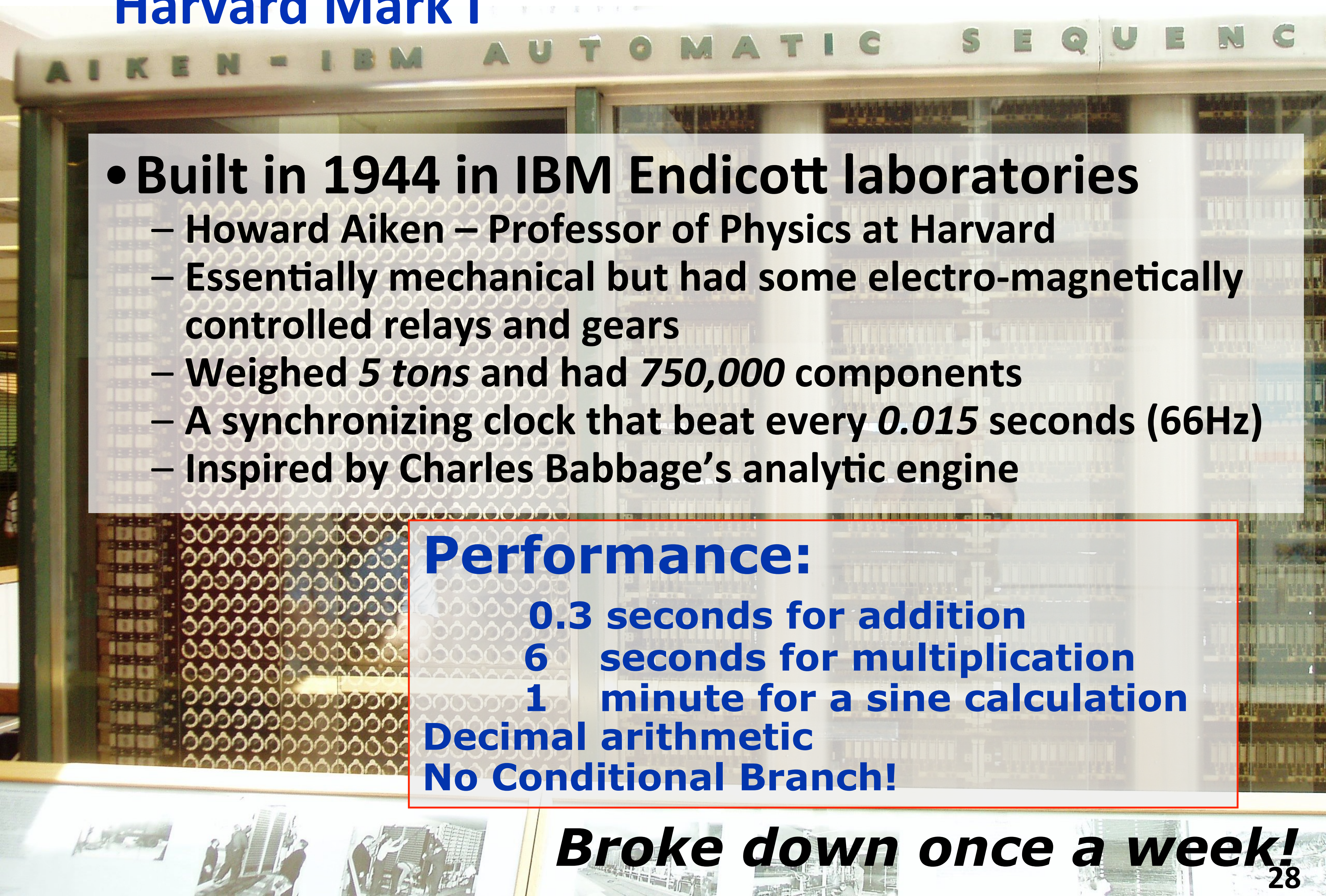
Harvard Mark I

- **Built in 1944 in IBM Endicott laboratories**
  - **Howard Aiken – Professor of Physics at Harvard**
  - **Essentially mechanical but had some electro-magnetically controlled relays and gears**
  - **Weighed *5 tons* and had *750,000* components**
  - **A synchronizing clock that beat every *0.015* seconds (66Hz)**
  - **Inspired by Charles Babbage's analytic engine**

Performance:

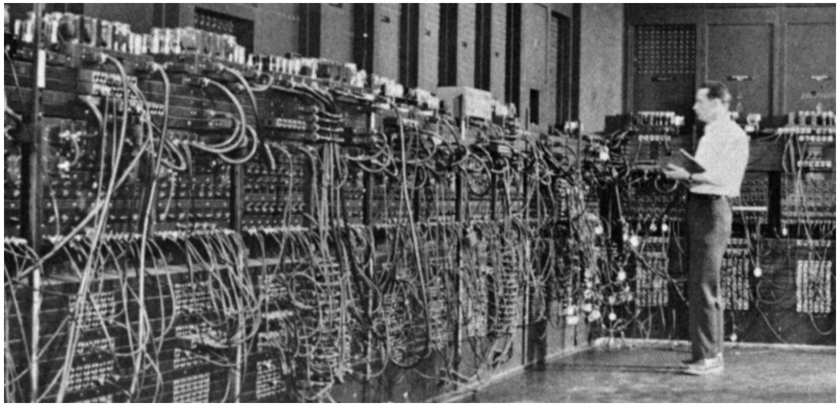
0.3 seconds for addition
6 seconds for multiplication
1 minute for a sine calculation
Decimal arithmetic
No Conditional Branch!

Broke down once a week!



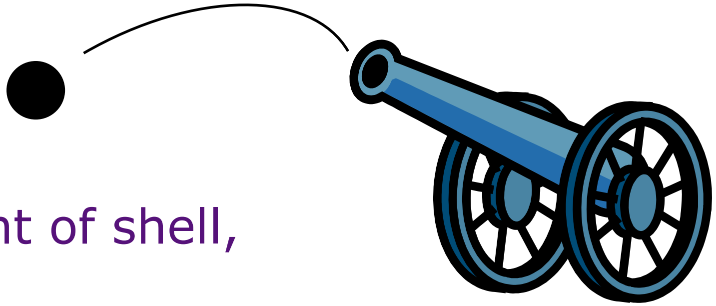
Electronic Numerical Integrator and Computer (ENIAC)

- Inspired by Atanasoff and Berry, Eckert and Mauchly designed and built ENIAC (1943-45) at the University of Pennsylvania
- The first, completely electronic, operational, general-purpose analytical calculator!
  - 30 tons, 72 square meters, 200KW
- Performance
  - Read in 120 cards per minute
  - Addition took 200 μs, Division 6 ms
  - 1000 times faster than Mark I
- Not very reliable!



WW-2 Effort

Application: Ballistic calculations



angle = f (location, tail wind, cross wind,
air density, temperature, weight of shell,
propellant charge, ...)

Computers in mid 50's

- Hardware was expensive
- Store instructions were small (1000 words)

  ⇒ No resident system software!
- Memory access time was 10 to 50 times slower than the processor cycle

  ⇒ Instruction execution time was totally dominated by the *memory reference time*.
- The *ability to design complex control circuits* to execute an instruction was the central design concern as opposed to *the speed* of decoding or an ALU operation
- Programmer's view of the machine was inseparable from the actual hardware implementation
- MTBF 20 minutes was state of the art

Compatibility Problem at IBM

By early 60's, *IBM had 4 incompatible lines of computers!*

701	→	7094
650	→	7074
702	→	7080
1401	→	7010

Each system had its own

- Instruction set
- I/O system and Secondary Storage:
  magnetic tapes, drums and disks
- assemblers, compilers, libraries,...
- market niche
  business, scientific, real time, ...

⇒ *IBM 360*

IBM 360 : Design Premises

Amdahl, Blaauw and Brooks, 1964

- The design must lend itself to *growth and successor machines*
- General method for connecting I/O devices
- Total performance - answers per month rather than bits per microsecond $\Rightarrow$ *programming aids*
- Machine must be capable of *supervising itself* without manual intervention
- Built-in *hardware fault checking* and locating aids to reduce down time
- Simple to assemble systems with redundant I/O devices, memories etc. for *fault tolerance*
- Some problems required floating-point larger than 36 bits

IBM 360: *A General-Purpose Register (GPR) Machine*

- Processor State
  - 16 General-Purpose 32-bit Registers
    - » *may be used as index and base register*
    - » *Register 0 has some special properties*
  - 4 Floating Point 64-bit Registers
  - A Program Status Word (PSW)
    - » *PC, Condition codes, Control flags*
- A 32-bit machine with 24-bit addresses
  - But no instruction contains a 24-bit address!
- Data Formats
  - 8-bit bytes, 16-bit half-words, 32-bit words, 64-bit double-words

The IBM 360 is why bytes are 8-bits long today!

IBM 360: Initial Implementations

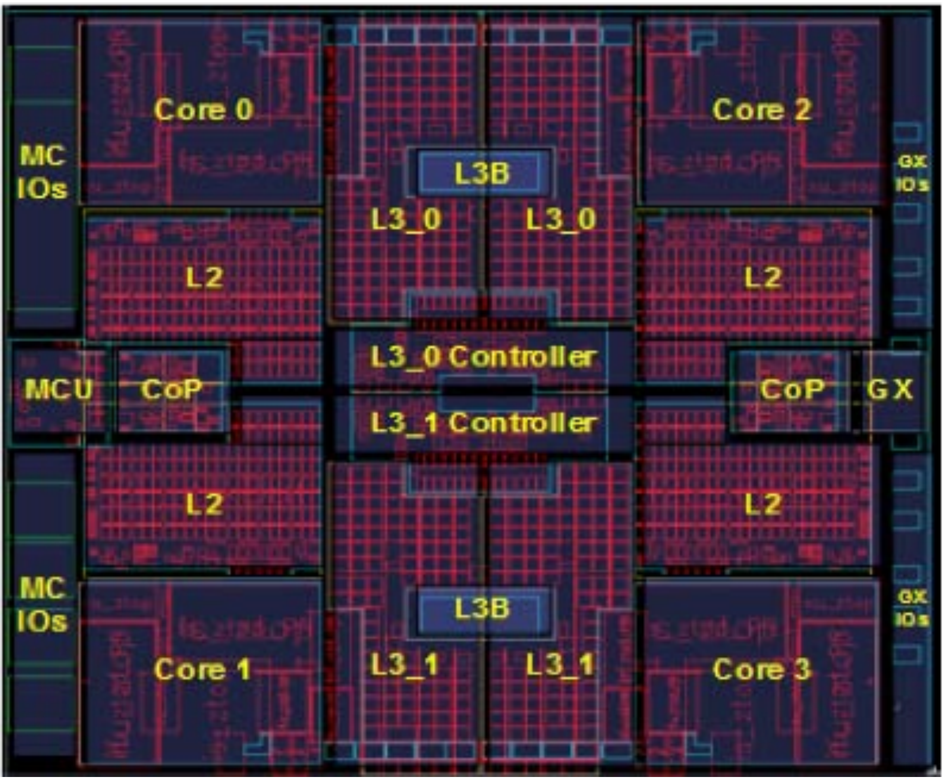
	Model 30	. . .	*Model 70*
Storage	8K - 64 KB		256K - 512 KB
Datapath	8-bit		64-bit
Circuit Delay	30 nsec/level		5 nsec/level
Local Store	Main Store		Transistor Registers
Control Store	Read only 1μsec	Conventional circuits	

IBM 360 instruction set architecture (ISA) completely hid the underlying technological differences between various models.

Milestone: The first true ISA designed as portable hardware-software interface!

With minor modifications it still survives today!

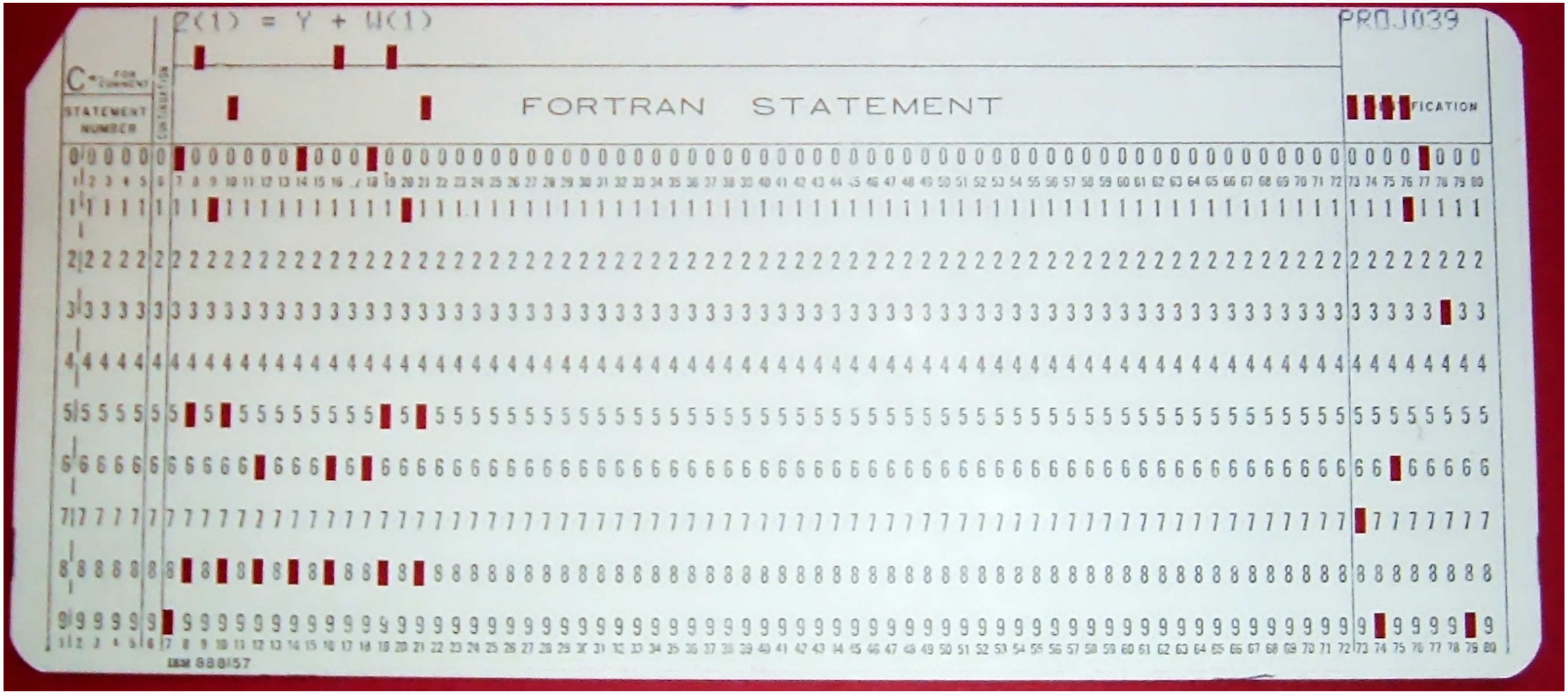
IBM 360: 47 years later…
The zSeries z11 Microprocessor



[IBM, HotChips, 2010]

- 5.2 GHz in IBM 45nm PD-SOI CMOS technology
- 1.4 billion transistors in 512 mm^2
- 64-bit virtual addressing
  - original S/360 was 24-bit, and S/370 was 31-bit extension
- Quad-core design
- Three-issue out-of-order superscalar pipeline
- Out-of-order memory accesses
- Redundant datapaths
  - every instruction performed in two parallel datapaths and results compared
- 64KB L1 I-cache, 128KB L1 D-cache on-chip
- 1.5MB private L2 unified cache per core, on-chip
- On-Chip 24MB eDRAM L3 cache
- Scales to 96-core multiprocessor with 768MB of shared L4 eDRAM

Storage Devices Also Progressed

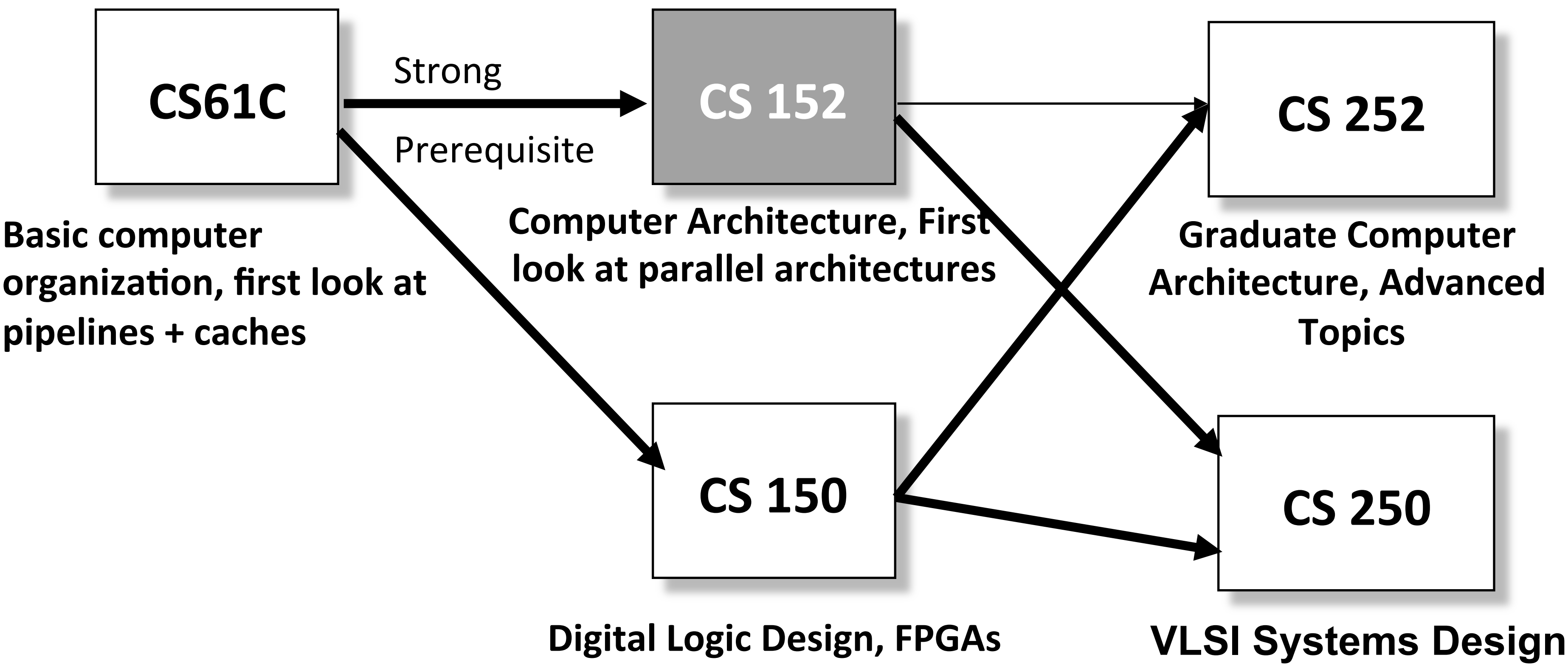


Magnetic Storage Devices

7.25 MB

LOGISTICS

Related Courses



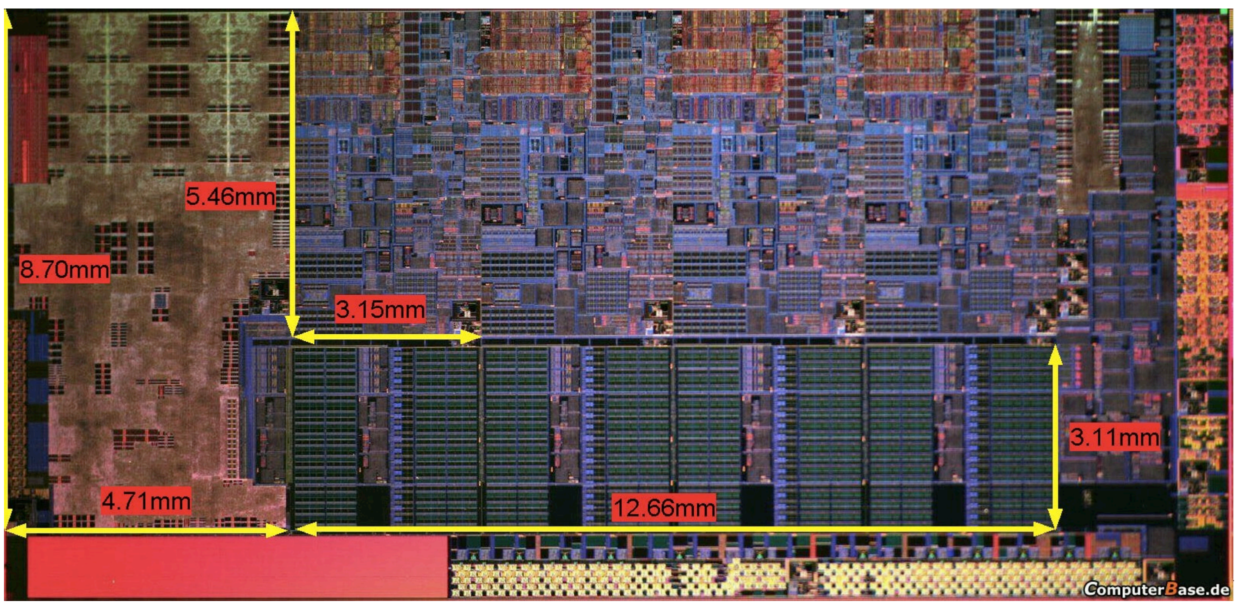
CS61C vs CS152 vs CS252

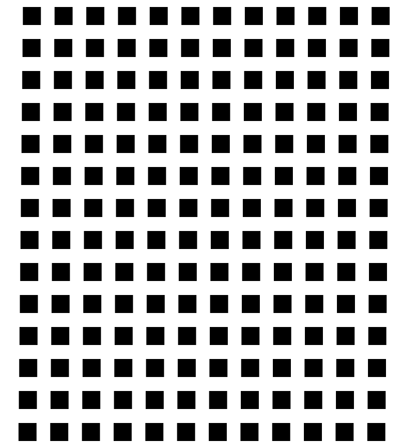
- CS152 focuses on interaction of software and hardware
  - more architecture and less digital engineering
  - more useful for OS developers, compiler writers, performance programmers
- Much of the material you'll learn this term was previously in CS252
  - Some of the current CS61C was in CS252 over 20 years ago!
  - Maybe every 10 years, shift CS252->CS152->CS61C?
- CS152 begins where CS61C left off (with overlap)
- CS252 delves into more detail and has a research project

CS152 Executive Summary

The processor you built in CS61C

What you'll understand and experiment with in CS152





Plus, the technology behind chip-scale multiprocessors (CMPs) and graphics processing units (GPUs)

CS152 Structure and Syllabus

Five modules

1. Simple machine design (ISAs, microprogramming, unpipelined machines, Iron Law, simple pipelines)
2. Memory hierarchy (DRAM, caches, optimizations) plus virtual memory systems, exceptions, interrupts
3. Complex pipelining (score-boarding, out-of-order issue)
4. Explicitly parallel processors (vector machines, VLIW machines, multithreaded machines)
5. Multiprocessor architectures (memory models, cache coherence, synchronization)

CS152 Administrivia

Instructor: George Michelogiannakis, **mihelog@eecs**

Office Hours: After lectures, Wednesdays 11-12:30pm 341A Soda

T. A.: Colin Schmidt, **colins@eecs**

Office Hours: Tuesday 2-4pm 651 Soda

Lectures: M/W, 9-10:30AM, 306 Soda

Section: Th 2PM-4PM, 9 105 Latimer

Text: *Computer Architecture: A Quantitative Approach,*

Hennessey and Patterson, 5th Edition (2012)

Readings assigned from this edition, some readings available in older editions –see web page.

Web page: **http://inst.eecs.berkeley.edu/~cs152**

Lectures available online

Piazzza: **http://piazza.com/berkeley/spring2016/cs152**

CS152 Course Components

- 15% Problem sets (one per module)
  - Intended to help you learn the material. Feel free to discuss with other students and instructors, but must turn in your own solutions. Grading based mostly on effort, but quizzes assume that you have worked through all problems. Solutions released after PSs handed in
- 45% Quizzes (one per module)
  - In-class, closed-book, no calculators, no smartphones, no laptops,...
  - Based on lectures, readings, problem sets, and labs
- 40% Labs (one per module)
  - Labs use advanced processor and system simulators
  - Directed plus open-ended sections to each lab

- Sections will review each of the above
- Check the website for deadlines!
- Sign up for Piazza!

CS152 Labs

- Each lab has directed plus open-ended assignments
- Directed portion (2/7) is intended to ensure students learn main concepts behind lab
  - Each student must perform own lab and hand in their own lab report
- Open-ended assignment (5/7) is to allow you to show your creativity
  - Roughly a "mini-project"
    - E.g., try an architectural idea and measure potential, or try to improve a design. Negative results OK (if explainable!)
  - Students can work individually or in groups of two
  - Group open-ended lab reports must be handed in separately (but state who you worked with)
  - Students can work in different groups for different assignments
- Lab reports must be readable English summaries
- Two free two-day extensions per student
- You may have to learn scripting languages

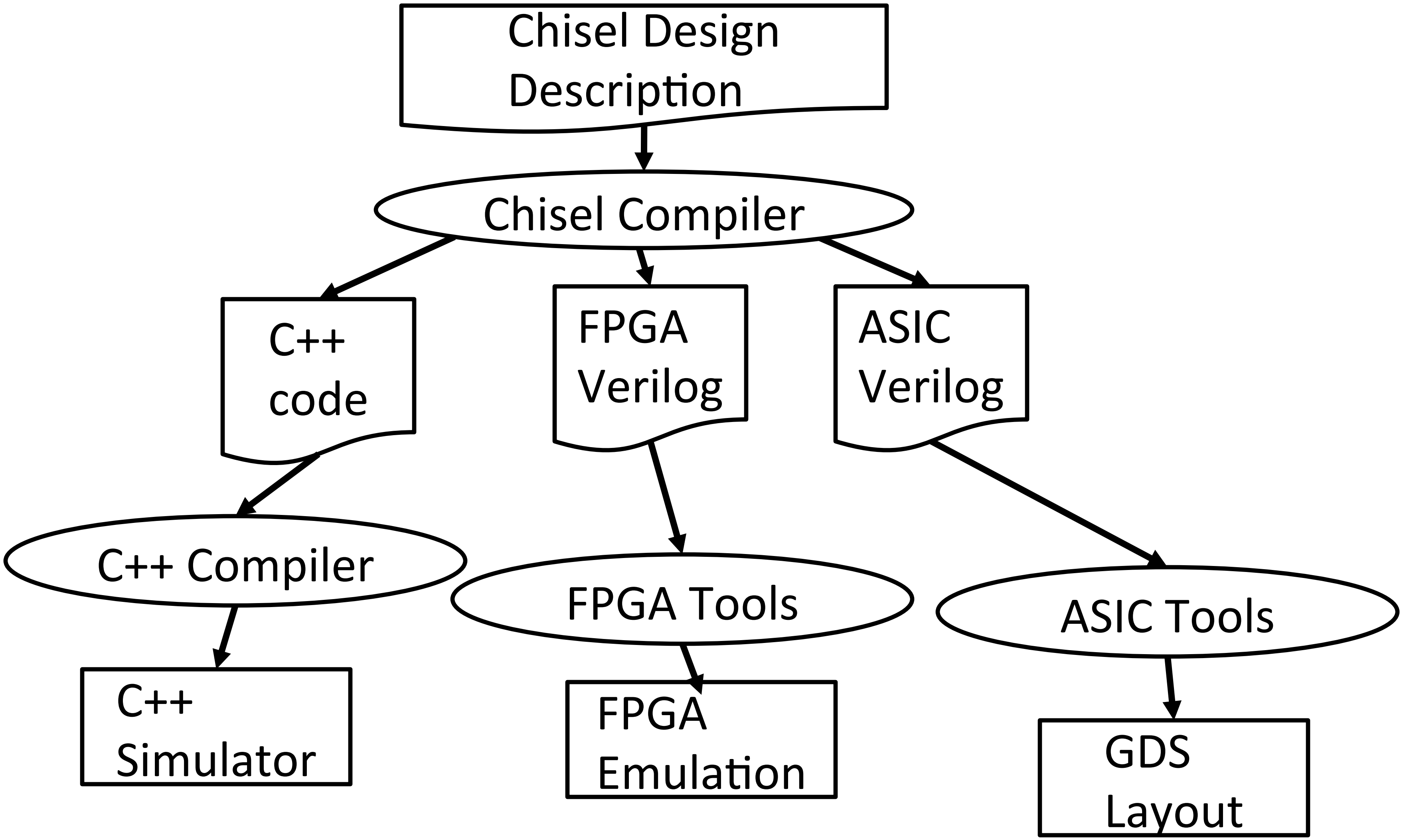
RISC-V ISA

- RISC-V is a new simple, clean, extensible ISA we developed at Berkeley for education and research
  - RISC-I/II, first Berkeley RISC implementations
  - Berkeley research machines SOAR/SPUR considered RISC-III/IV
- Both of the dominant ISAs (x86 and ARM) are too complex to use for teaching
- RISC-V ISA manual available on web page
  - See "resources" on class website
- Full GCC-based tool chain available

Chisel simulators

- Chisel is a new hardware description language we developed at Berkeley based on Scala
  - ***C***onstructing ***H***ardware ***i***n a ***S***cala ***E***mbedded ***L***anguage
- Labs will use RISC-V processor simulators derived from Chisel processor designs
  - Gives you much more detailed information than other simulators
  - Can map to FPGA or real chip layout
- You need to learn some minimal Chisel in CS152, but we'll make Chisel RTL source available so you can see all the details of our processors
- Can do lab projects based on modifying the Chisel RTL code if desired

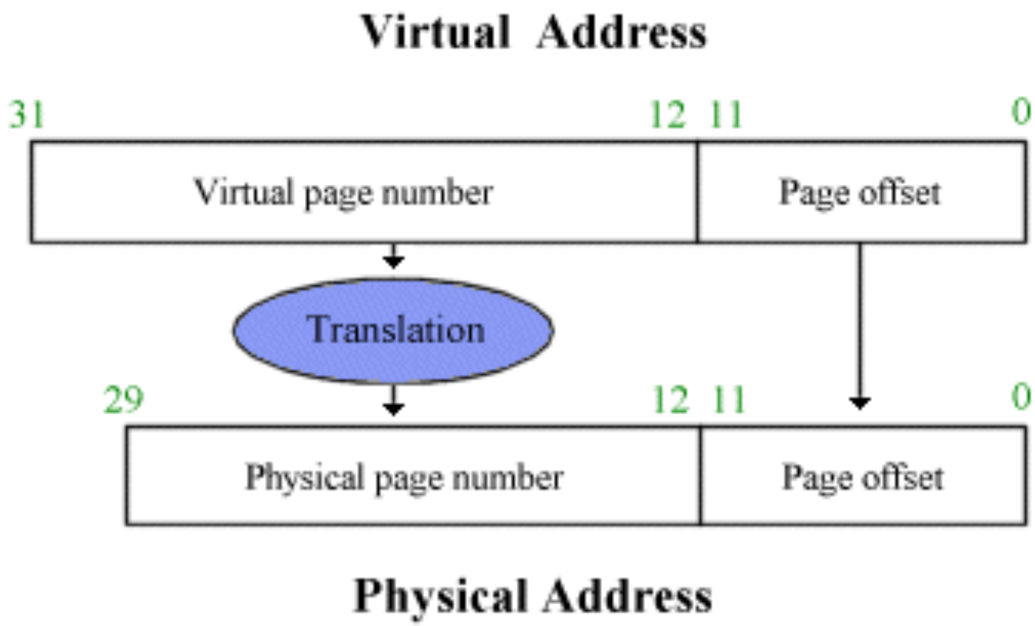
Chisel Design Flow



FAMILIARITY QUIZ

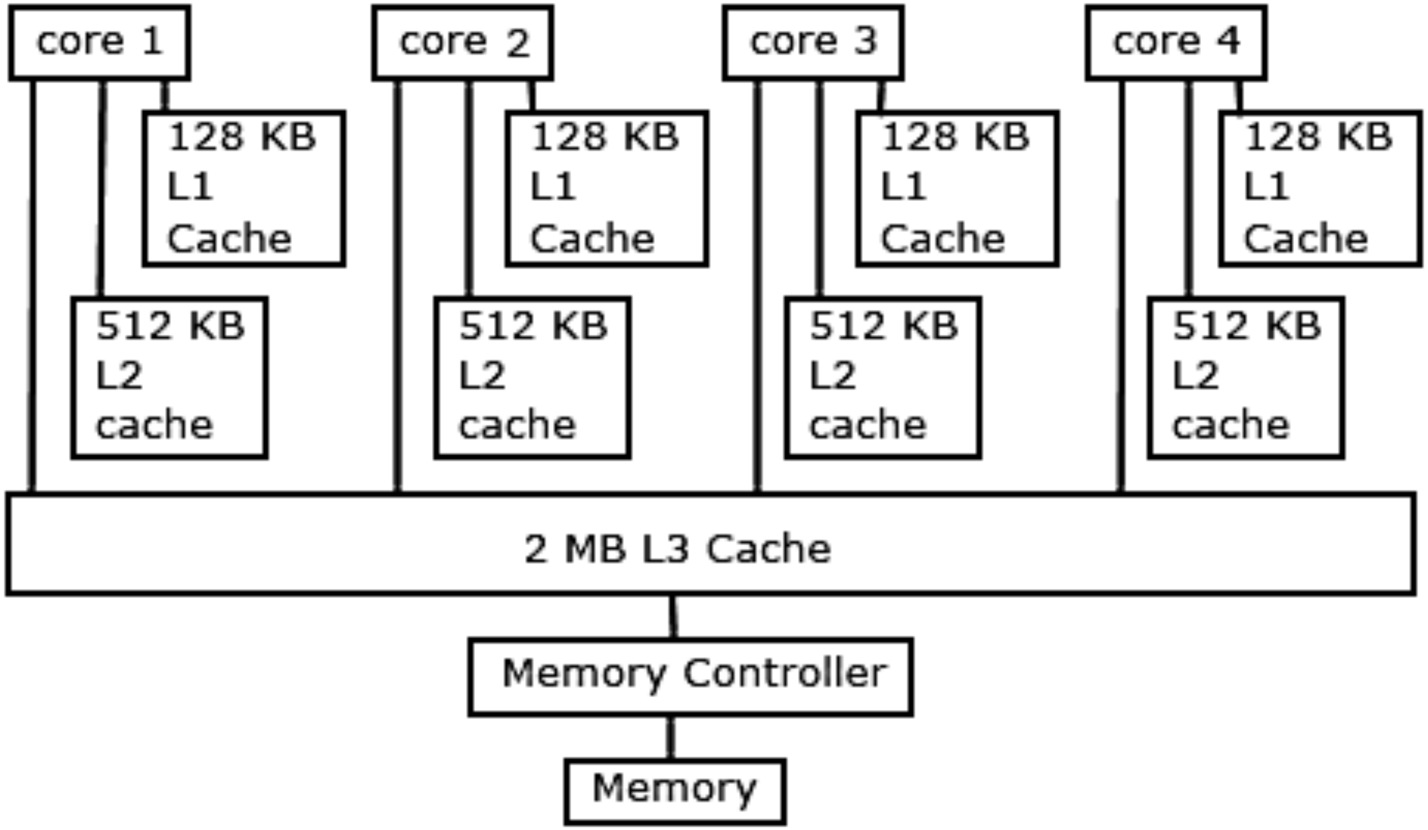
Pipelined Processor

Virtual Addresses



Mapping from a virtual to a physical address

Caches


core 1
core 2
core 3
core 4
128 KB L1 Cache
128 KB L1 Cache
128 KB L1 Cache
128 KB L1 Cache
512 KB L2 cache
512 KB L2 cache
512 KB L2 cache
512 KB L2 cache
2 MB L3 Cache
Memory Controller
Memory

Birds Cache (hoard) too!

- Same idea. Bring valuable objects close
- Acorn Woodpeckers store their food in holes drilled in trees

In Conclusion

- Computer Architecture >> ISAs and RTL
- CS152 is about interaction of hardware and software, and design of appropriate abstraction layers
- Computer architecture is shaped by technology and applications
  - History provides lessons for the future
- Computer Science at the crossroads from sequential to parallel computing
  - Salvation requires innovation in many fields, including computer architecture
- Read Chapter 1 & Appendix A for next time!

Acknowledgements

- These slides contain material developed and copyright by:
  - Arvind (MIT)
  - Krste Asanovic (MIT/UCB)
  - Joel Emer (Intel/MIT)
  - James Hoe (CMU)
  - John Kubiatowicz (UCB)
  - David Patterson (UCB)
  - Various websites and papers

- MIT material derived from course 6.823
- UCB material derived from course CS252