CS250P: Computer Systems Architecture Introduction

Sang-Woo Jun

Fall 2022

UCI

About Me

❑ Sang-Woo Jun
   o Assistant Professor, UC Irvine
   o Ph.D. (2018) @ MIT

❑ Research Interests
   o Systems architecture
   o Accelerators
   o NVM storage
   o Applications!
      • Graphs, Bioinformatics, Machine learning…

❑ Some Nice Papers
   o (ISCA, VLDB, FAST, FPGA, MICRO, …)

❑ Some Nice Media Coverage
   o Engadget, The Next Platform, …

Why should we learn about computer architecture?

As a software developer

As a hardware architect

Why should software engineers learn about architecture?

"People who are really serious about software should make their own hardware."

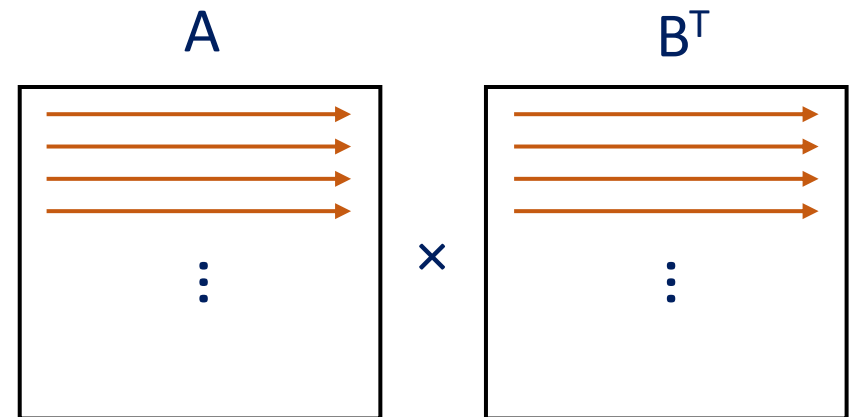– Alan Kay

Computer architecture effects example 1

- ❑ Multiplying two 2048 x 2048 matrices
  - ○ 16 MiB, doesn't fit in any cache
- ❑ Machine: Intel i5-7400 @ 3.00GHz
- ❑ Time to transpose B is also counted

```
for (i=0 to N)
  for (j=0 to N)
    for (k=0 to N)
      C[i][j] += A[i][k] * B[k][j];
```
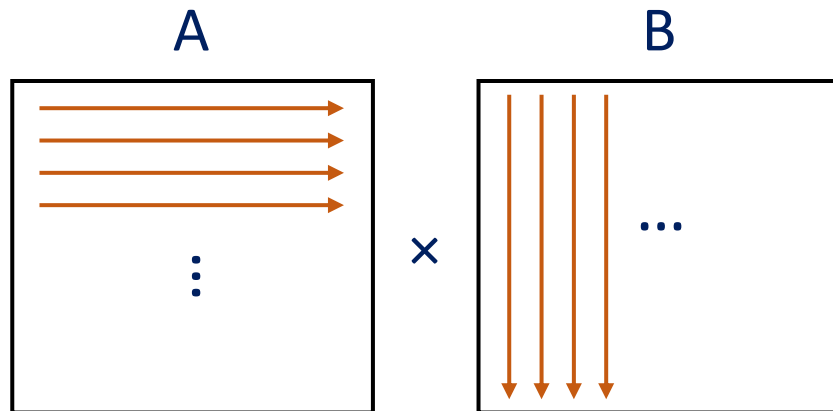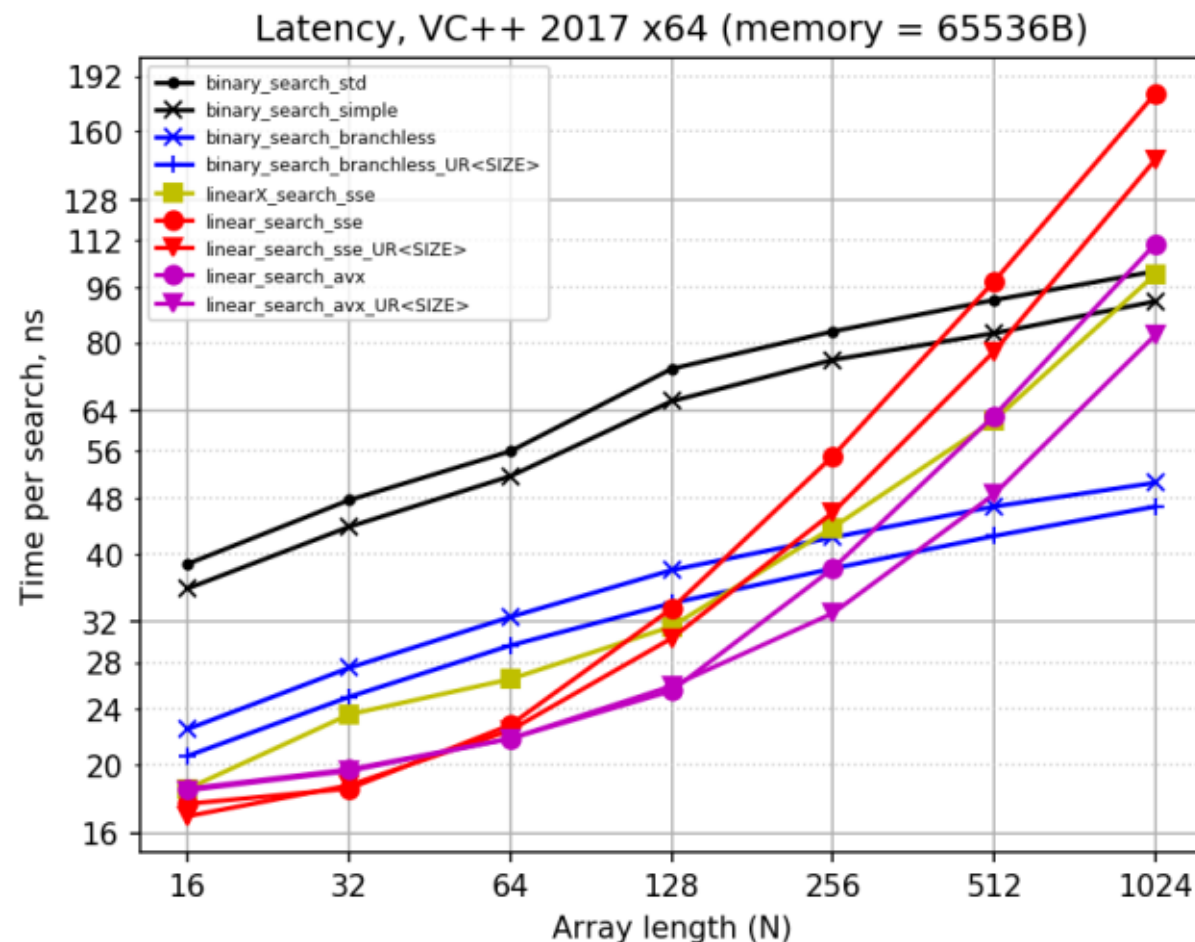

63.19 seconds

VS

10.39 seconds
(6x performance!)

Computer architecture effects example 2

❑ Binary search vs. branchless binary search vs. linear search

o Where does this difference come from, and how do I exploit this?

o Architecture, assembly knowledge!

Latency, VC++ 2017 x64 (memory = 65536B)

Computer architecture effects example 3


```
int result[P];

// Each of P parallel workers processes 1/P-th of the data;
// the p-th worker records its partial count in result[p]
for (int p = 0; p < P; ++p )
    pool.run( [&,p] {
        result[p] = 0;
        int chunkSize = DIM/P + 1;
        int myStart = p * chunkSize;
        int myEnd = min( myStart+chunkSize, DIM );
        for( int i = myStart; i < myEnd; ++i )
            for( int j = 0; j < DIM; ++j )
                if( matrix[i*DIM + j] % 2 != 0 )
                    ++result[p]; } );
pool.join();                    // Wait for all tasks to complete

odds = 0;                       // combine the results
for( int p = 0; p < P; ++p )
    odds += result[p];
```

matrix —→ DIM DIM

Speedup for Example 1
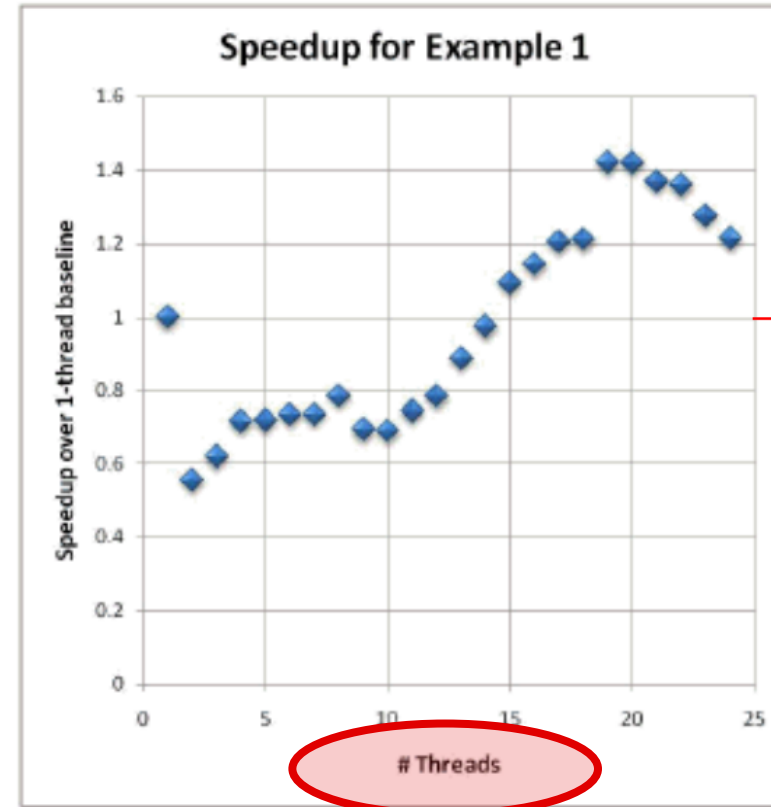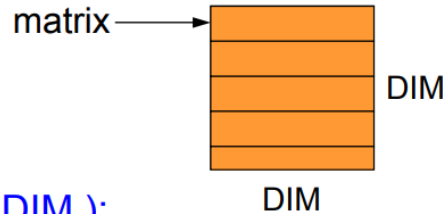
Faster than 1 core

Slower than 1 core

Threads

REALLY BAD scalability! Why?

Computer architecture effects example 4

```
for (target in stream):
    entities[target].string.append(char);
```

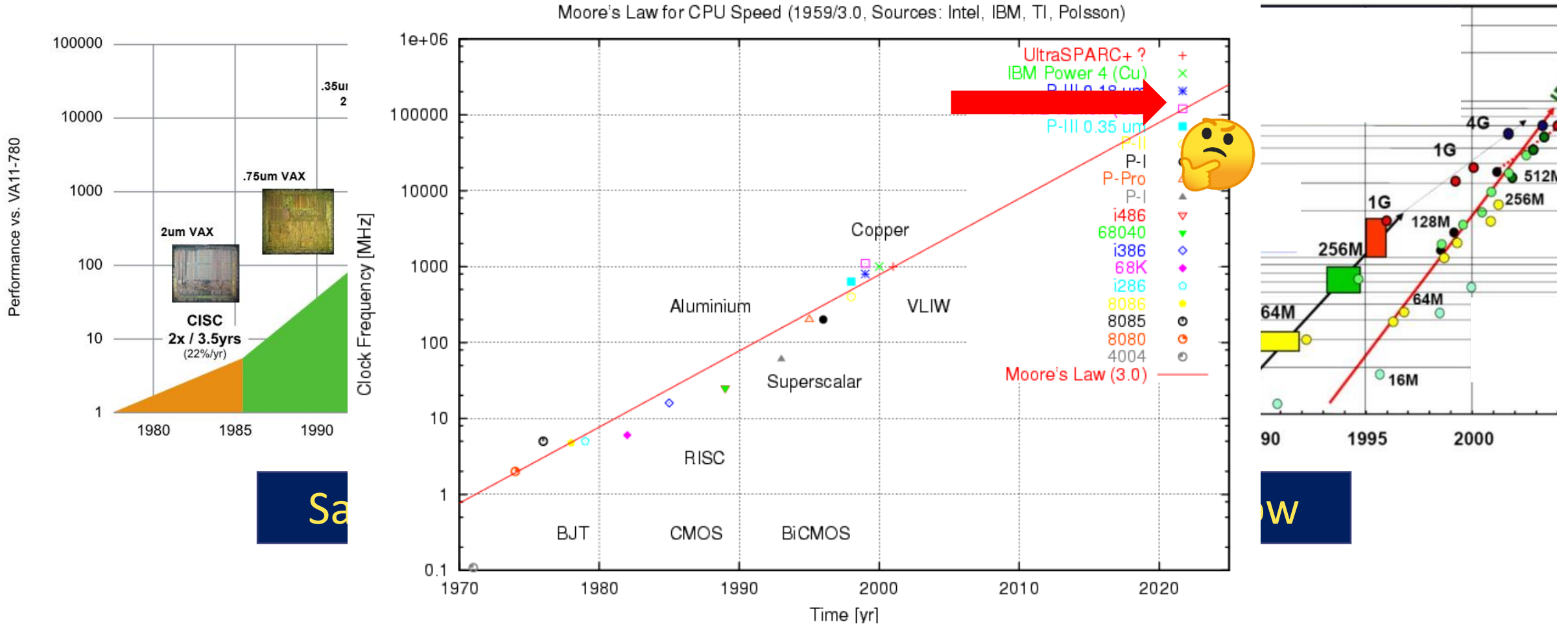When entities.size < (1<<16): 1 GB/s

When entities.size > (1<<20): 200 MB/s

Why??

Why do we need computer architects?
-- The simpler past

Now: The end of Moore's law and performance scaling

40 Years of Processor Performance

16nm Zynq RFSoC

55nm Radeon HD4870

180nm MIPS IP Core

.35um Alpha 21264

.75um VAX

2um VAX

CISC
2x / 3.5yrs
(22%/yr)

RISC
2x / 1.5yrs
(52%/yr)

End of Dennard Scaling Multicore
2x / 3.5yrs
(23%/yr)

Amdahl's Law
2x / 6yrs
(12%/yr)

End of the Line?
2x / 20yrs
(3%/yr)

Performance vs. VA11-780

Source: John Hennessy and David Patterson, Computer Architecture: A Quantitative Approach, 6/e 2018

Running Into the Power Wall

1000

Nuclear Reactor

Pentium® 4

Hot plate

Pentium® III

Pentium® II

intel inside pentium ///

Pentium® Pro

Pentium®

i486

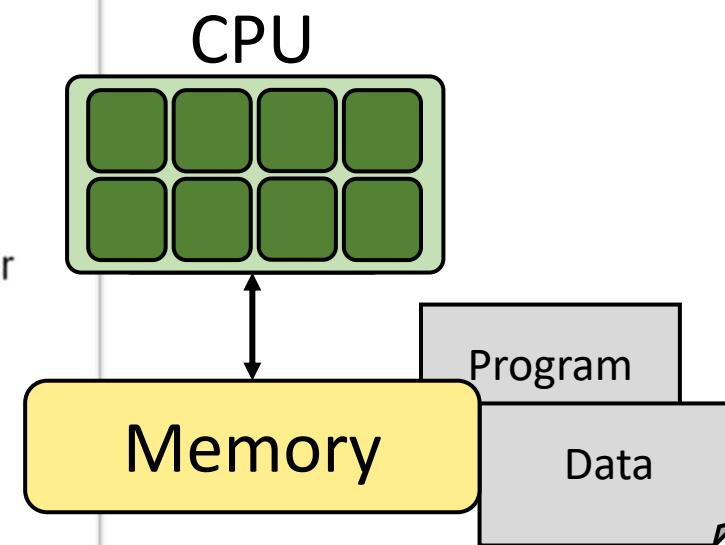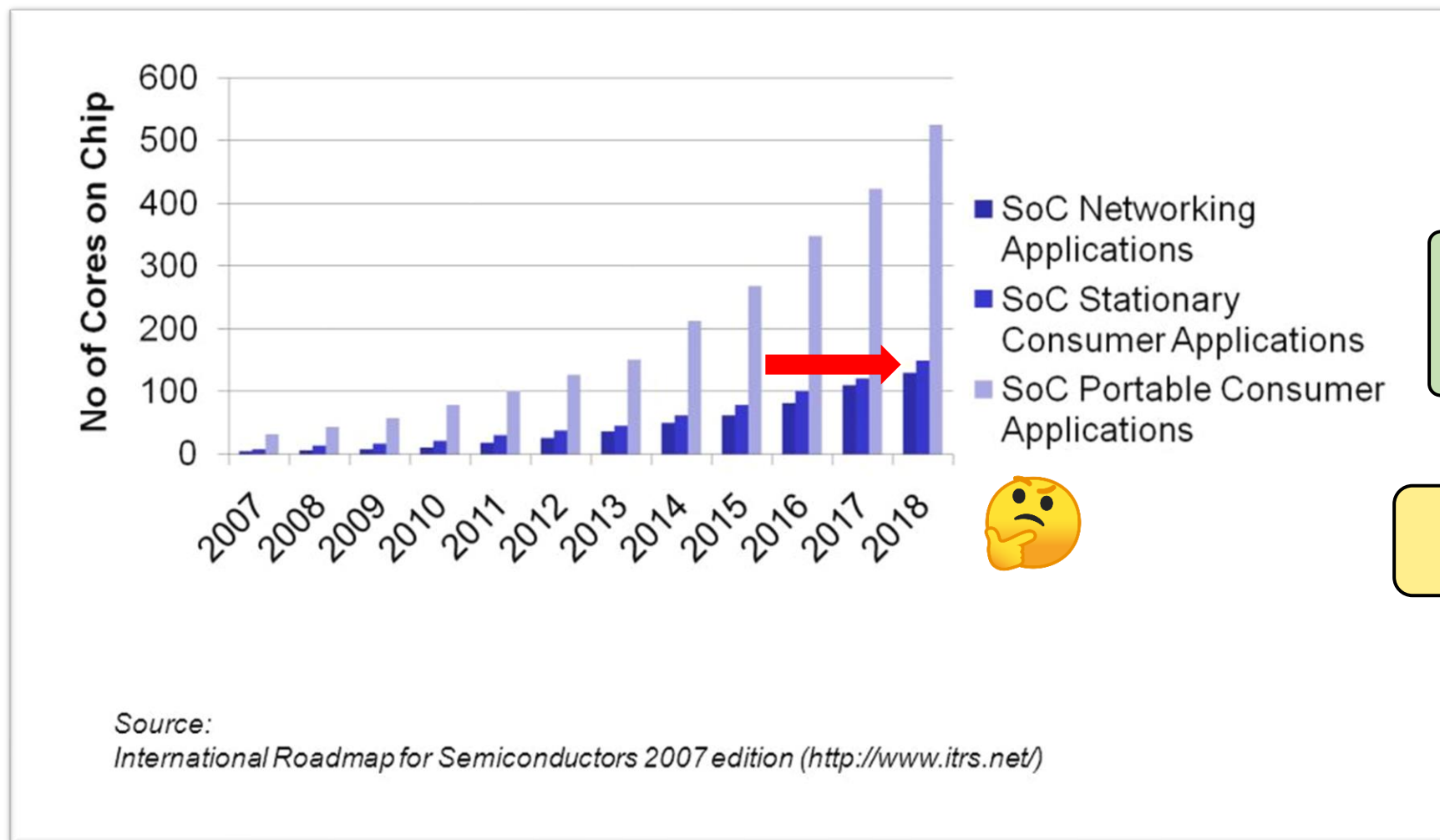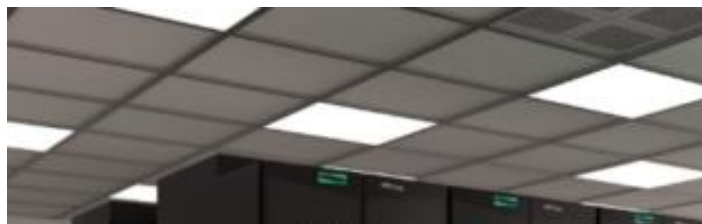| 1μ | 0.7μ | 0.5μ | 0.35μ | 0.25μ | 0.18μ | 0.13μ | 0.1μ | 0.07μ | 0.007 μ |
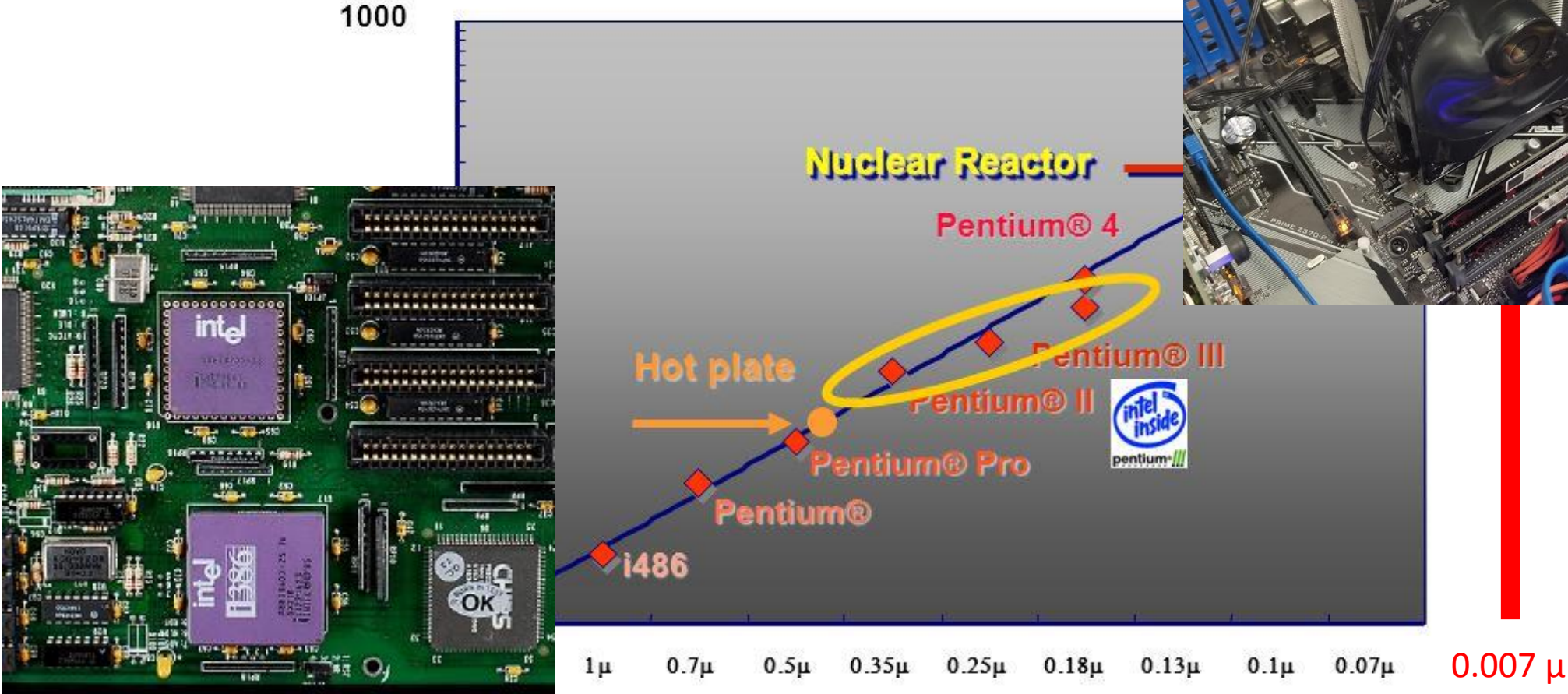
* "New Microarchitecture Challenges in the Coming Generations of CMOS Process Technologies" –
Fred Pollack, Intel Corp. Micro32 conference key note - 1999.

Crisis Averted With Manycores?

Source:
International Roadmap for Semiconductors 2007 edition (http://www.itrs.net/)

CPU

Memory

Program

Data

The State of C

Department of Energy requested e... ...ntial power consumption of San Francisco

...Research nuclear reactor

| Heavy use of GPUs | ...aded programming |

1,000,000,000,000,000,000 floating p...

Using 2016 technolog...
Using 2019 technology, ...
Using 2022 technology, 20 MW
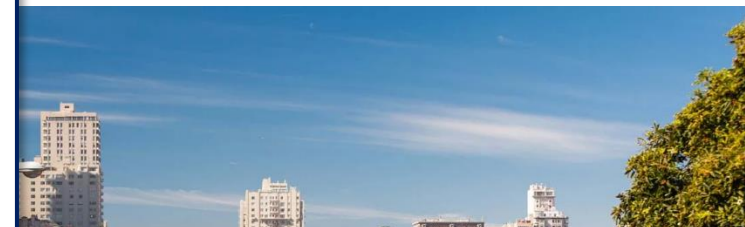
158 MW

Image: TheNextPlatform

(Calculated from "Electricity Consumption by County", California energy commission)

Also, scaling size is becoming more difficult!

❑ Processor fabrication technology has always reduced in size
  o As of 2022, transitioning from 5 nm to 3 nm

Q: Is sub-3nm even feasible?

Q: What does 3 nm even mean?

Image source: Intel

Number of Semiconductor Manufacturers with a Cutting Edge Logic Fab

Year 2000

Year 2008

Year 2022

Image source: WikiChip

Only three players left?!

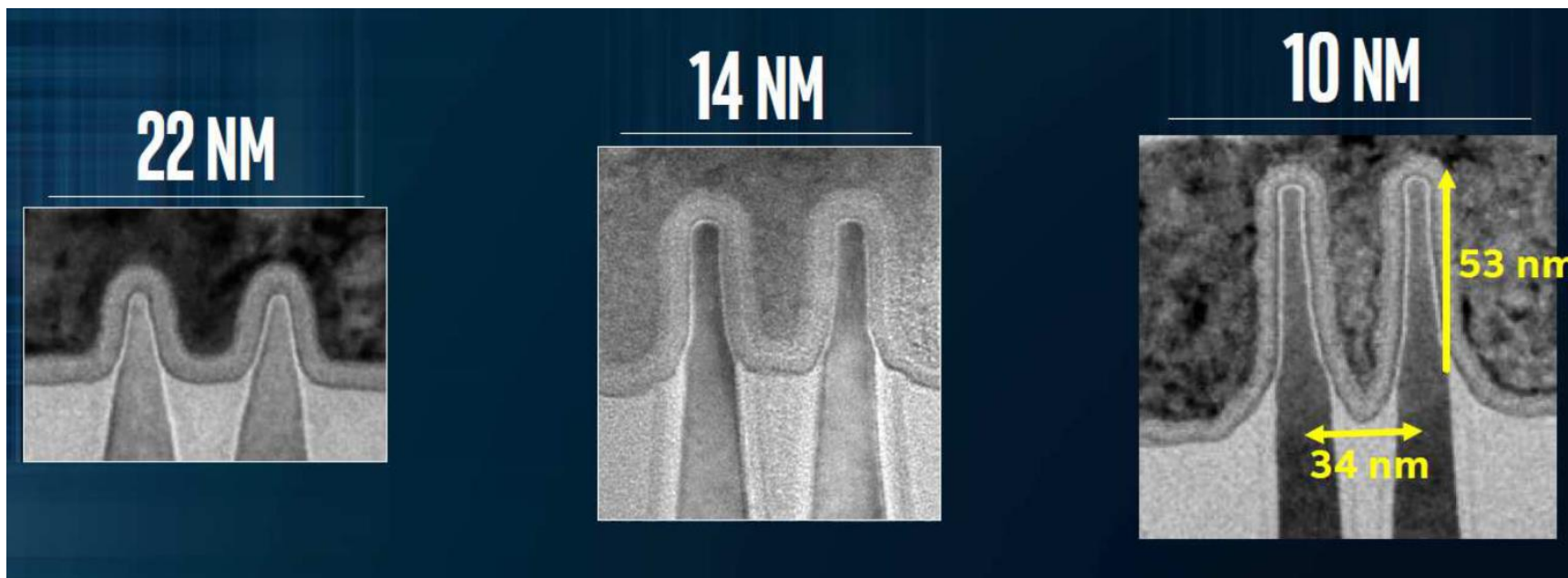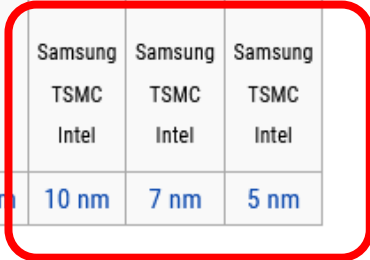| 180 nm	130 nm	90 nm	65 nm	45 nm/40 nm	32 nm/28 nm	22 nm/20 nm	16 nm/14 nm	10 nm	7 nm	5 nm
SilTerra										
X-FAB										
Dongbu HiTek										
ADI	ADI									
Atmel	Atmel									
Rohm	Rohm									
Sanyo	Sanyo									
Mitsubishi	Mitsubishi									
ON	ON									
Hitachi	Hitachi									
Cypress	Cypress	Cypress								
Sony	Sony	Sony								
Infineon	Infineon	Infineon								
Sharp	Sharp	Sharp								
Freescale	Freescale	Freescale								
Renesas (NEC)	Renesas	Renesas	Renesas	Renesas						
Toshiba	Toshiba	Toshiba	Toshiba	Toshiba						
Fujitsu	Fujitsu	Fujitsu	Fujitsu	Fujitsu						
TI	TI	TI	TI	TI						
Panasonic	Panasonic	Panasonic	Panasonic	Panasonic	Panasonic					
STMicroelectronics	STM	STM	STM	STM	STM					
HLMC	HLMC		HLMC	HLMC	HLMC					
UMC	UMC	UMC	UMC	UMC	UMC		UMC			
IBM	IBM	IBM	IBM	IBM	IBM	IBM				
SMIC	SMIC	SMIC	SMIC	SMIC	SMIC		SMIC			
AMD	AMD	AMD	GlobalFoundries	GF	GF	GF	GF			
Samsung	Samsung	Samsung	Samsung	Samsung	Samsung	Samsung	Samsung	Samsung	Samsung	Samsung
TSMC	TSMC	TSMC	TSMC	TSMC	TSMC	TSMC	TSMC	TSMC	TSMC	TSMC
Intel	Intel	Intel	Intel	Intel	Intel	Intel	Intel	Intel	Intel	Intel

We can't keep doing what we used to

❑ Limited number of transistors, limited clock speed
- How to make the ABSOLUTE BEST of these resources?

❑ Timely example: Apple M1 Processor
- Claims to outperform everyone (per Apple)
- How?
  - "8-wide decoder" […] "16 execution units (per core)"
  - "(Estimated) 630-deep out-of-order"
  - "Unified memory architecture"
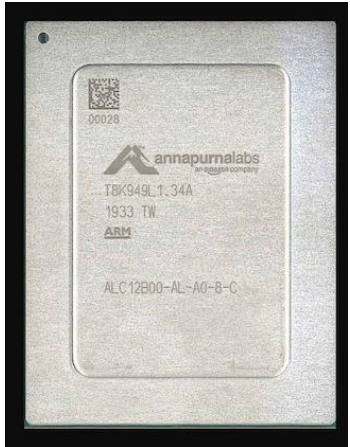 - Hardware/software optimized for each other
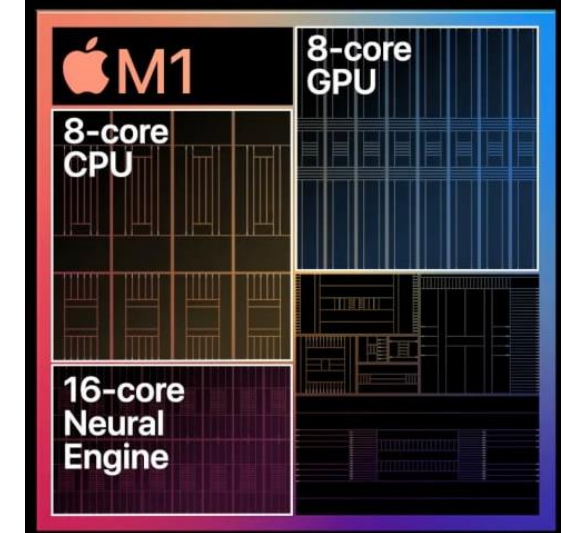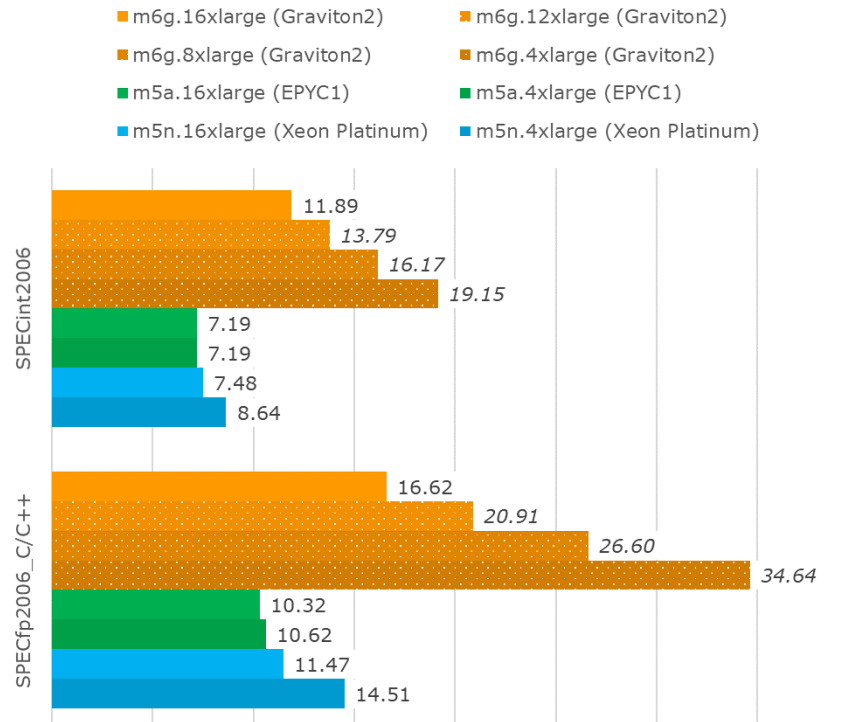

What do these mean?

Not just apple! (Amazon, Microsoft, EU, …)

Image source: Apple

We can't keep doing what we used to

AWS Graviton 2:

64-Core ARM

European Processor Accelerator (EPAC):

4-Core RISC-V +

Variable Precision Accelerator +

Stencil and Tensor Accelerator

Amazon EC2 Throughput Per Dollar

- m6g.16xlarge (Graviton2)
- m6g.12xlarge (Graviton2)
- m6g.8xlarge (Graviton2)
- m6g.4xlarge (Graviton2)
- m5a.16xlarge (EPYC1)
- m5a.4xlarge (EPYC1)
- m5n.16xlarge (Xeon Platinum)
- m5n.4xlarge (Xeon Platinum)

SPECint2006
- 11.89
- 13.79
- 16.17
- 19.15
- 7.19
- 7.19
- 7.48
- 8.64

SPECfp2006_C/C++
- 16.62
- 20.91
- 26.60
- 34.64
- 10.32
- 10.62
- 11.47
- 14.51

Image source: Anandtech, "Amazon's Arm-based Graviton2 Against AMD and Intel: Comparing Cloud Compute"
Image source: TheNextPlatform, "Europe Inches Closer to Native RISC-V Reality"

No better time to be an architect!

"There are Turing Awards waiting to be picked up
if people would just work on these things."
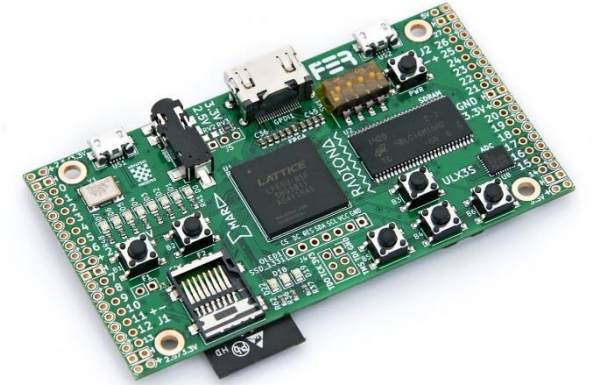—David Patterson, 2018

And on that note…

Welcome to CS 250P!

❑ We will learn:
- o How modern processors are designed to achieve high performance
- o under which restrictions, and
- o actually get hands-on experience with hardware design
- o using a sequence of gently guided labs.

Course mechanics

- ❑ Lectures: TuTh 8:00- 9:20p @ EH 1200
- ❑ Discussions: Fri 7:00- 7:50p @ DBH 1100
  - o May not always have lectures, but myself or at least one TA will be there for questions (Esmerald Aliaj, Jiyoung Ahn)

- ❑ Grading: Homework: 50%, midterm exam: 25%, final exam: 25% (all grades curved).
- ❑ Labs?
  - o Will use a high-level hardware-description language (Bluespec)
  - o By the end of the class, you will have a highly efficient CPU design that actually runs on metal! (FPGA)

What this class does and doesn't do

❑ It doesn't do
- o Bit-level control signal management
  - (Not how modern processors are designed!)
- o Details of the Intel x86 architecture
  - Very complicated and cluttered with backwards compatibility from the 70s
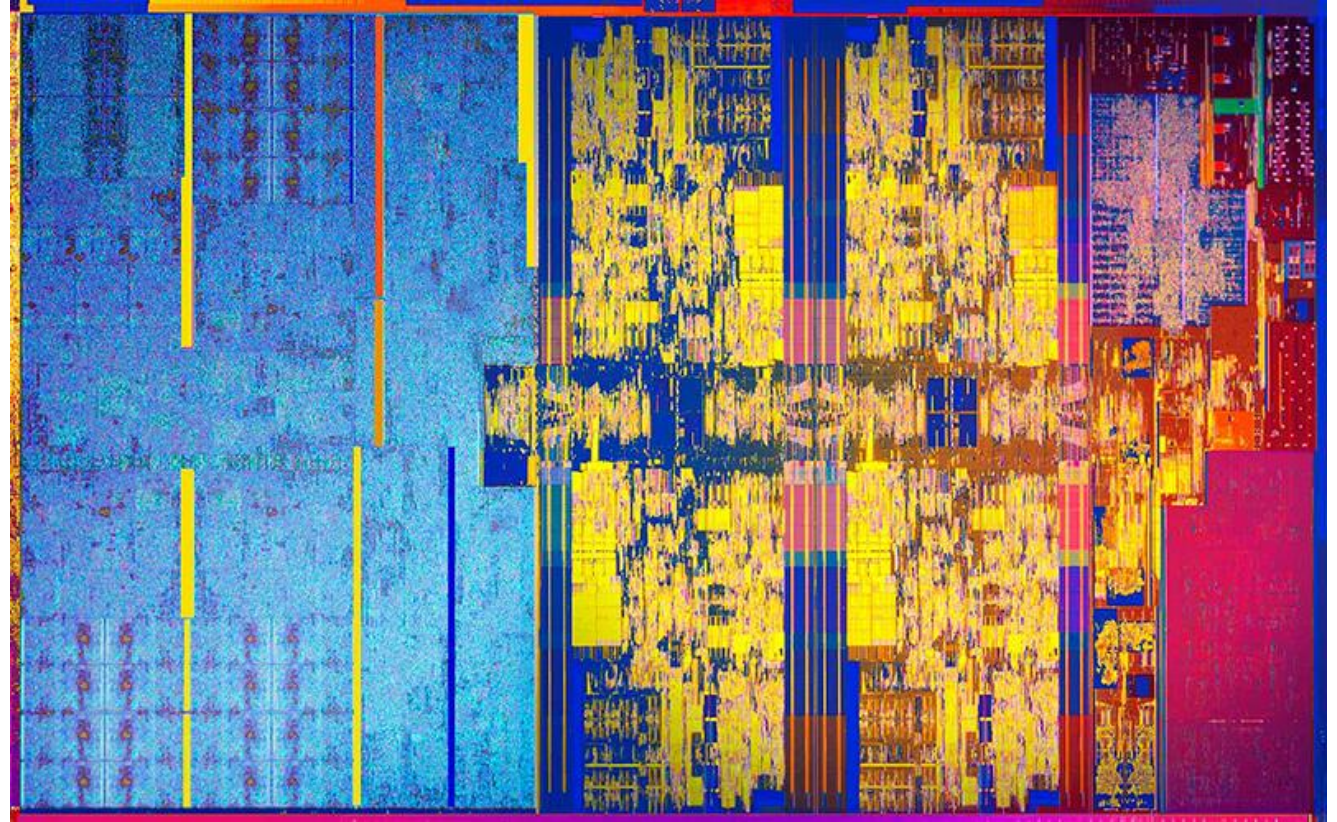 - But will introduce parts of it!

❑ It does do
- o Constructive computer architecture experience
  - Always working with a functional computer architecture design
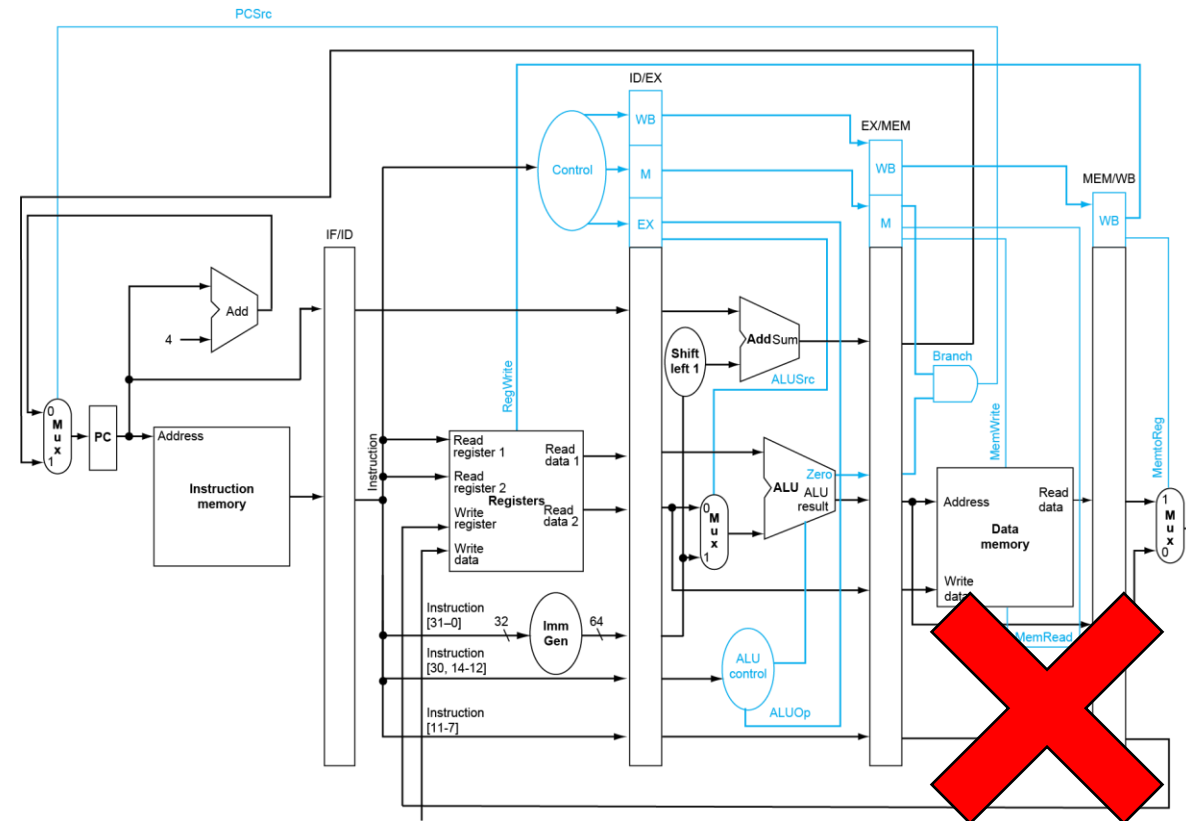
Times have changed…
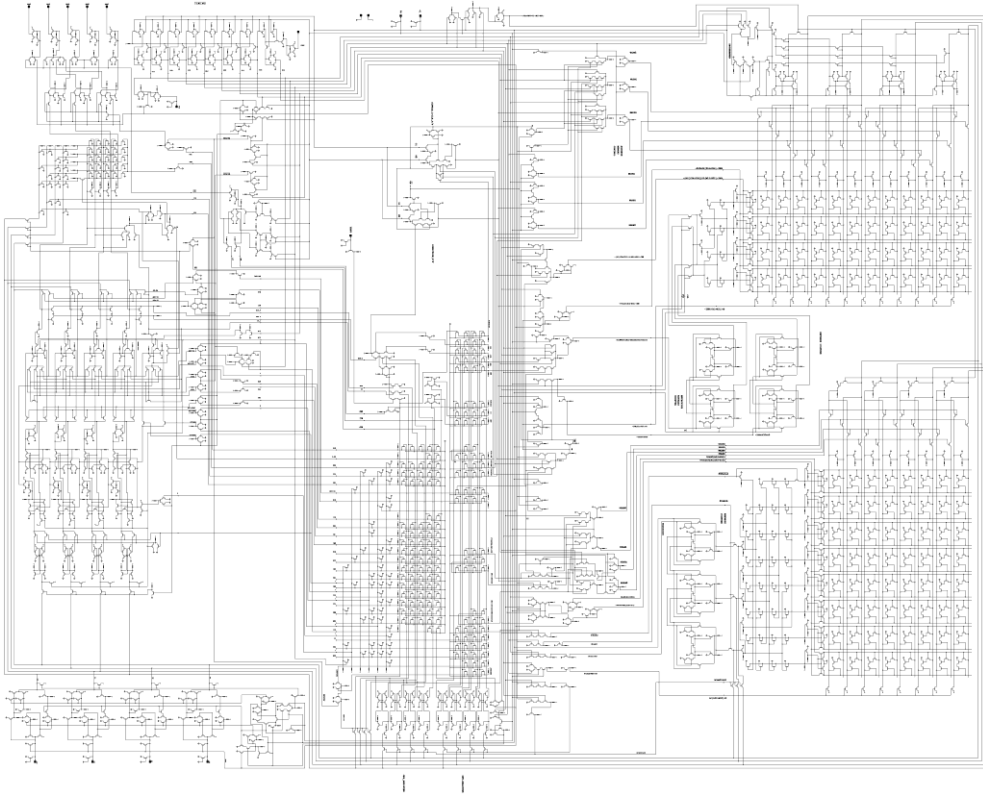
(1971) 2,250 transistors!
Intel 4004 Schematics drawn by Lajos Kintli and Fred Huettig
for the Intel 4004 50th anniversary project

(2020) +1 Billion transistors!
Intel Core-i7 die (Source: Intel)

We will use modern tools

❑ RISC-V
   o Open-source Instruction-Set Architecture (ISA) based on what was learned in the past decades

❑ Bluespec
   o A high-level hardware-description language

❑ Also, x86/ARM
   o De-facto standards in the modern world

Some important ideas in computer architecture

- ❑ Pipelining
- ❑ Caches and their design
- ❑ Branch prediction
- ❑ Virtual memory and privileges
- ❑ Superscalar
- ❑ Simultaneous multithreading
- ❑ Speculative execution
- ❑ Out-of-Order Execution
- ❑ Vector operations
- ❑ Accelerators

How far can we go in CS250P?

Course outline

❑ Part 1: The Hardware-Software Interface
  o What is a 'good' processor?
  o Assembly programming and conventions

❑ Part 2: Recap of digital design
  o Combinational and sequential circuits
  o How their restrictions influence processor design

❑ Part 3: Computer Architecture
  o Simple and pipelined processors
  o Out-of-order and explicitly parallel architectures
  o Caches and the memory hierarchy

❑ Part 4: Computer Systems
  o Operating systems, Virtual memory