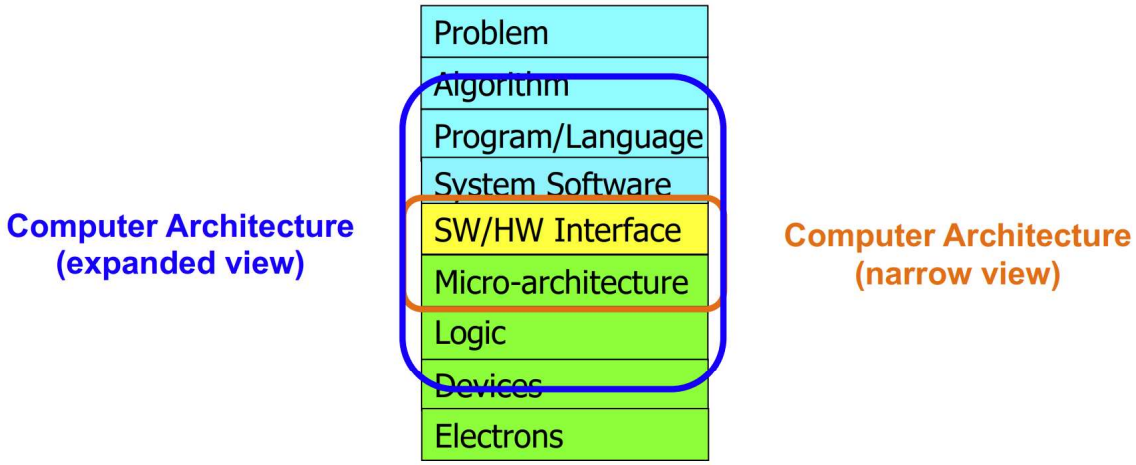
Parallel & Distributed Computing

Welcome to the class!

Introductions and discussion over the course outline.

Motivation: Why should we care about parallel and distributed computing

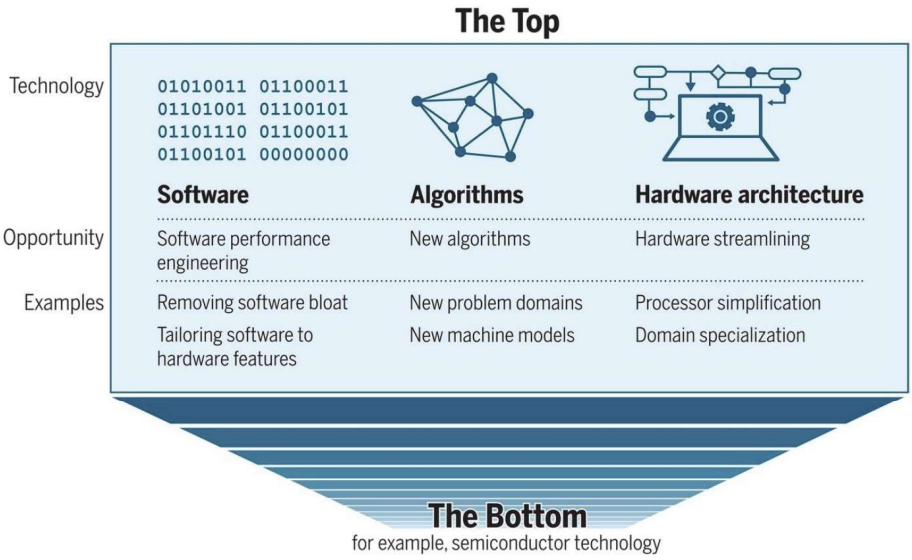
The Transformation Hierarchy



(Credit: Prof. Onur Mutlu: https://safari.ethz.ch/architecture/fall2022/lib/exe/fetch.php?media=onur-comparch-fall2022-lecture1-intro-afterlecture.pdf)

Computing System

Leiserson+, "There's plenty of room at the Top: What will drive computer performance after Moore's law?", Science, 2020



Data
Problem
Algorithm
Program/Language
Runtime System (VM, OS, MM)
ISA (Architecture)
Microarchitecture
Logic
Devices
Electrons

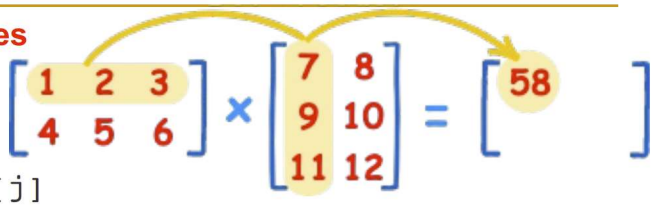
Richard Feynman, "There's Plenty of Room at the Bottom: An Invitation to Enter a New Field of Physics", a lecture given at Caltech, 1959.

Image source: https://science.sciencemag.org/content/368/6495/eaam9744

Software & Hardware Optimizations

Multiplying Two 4096-by-4096 Matrices

```
for i in xrange(4096):
 for j in xrange(4096):
  for k in xrange(4096):
   C[i][j] += A[i][k] * B[k][j]
```



Implementation	Running time (s)	Absolute speedup
Python	25,552.48	1x
Java	2,372.68	11x
C	542.67	47x
Parallel loops	69.80	366x
Parallel divide and conquer	3.80	6,727x
plus vectorization	1.10	23,224x
plus AVX intrinsics	0.41	62,806x

Leiserson+, "There's plenty of room at the Top: What will drive computer performance after Moore's law?", Science, 2020

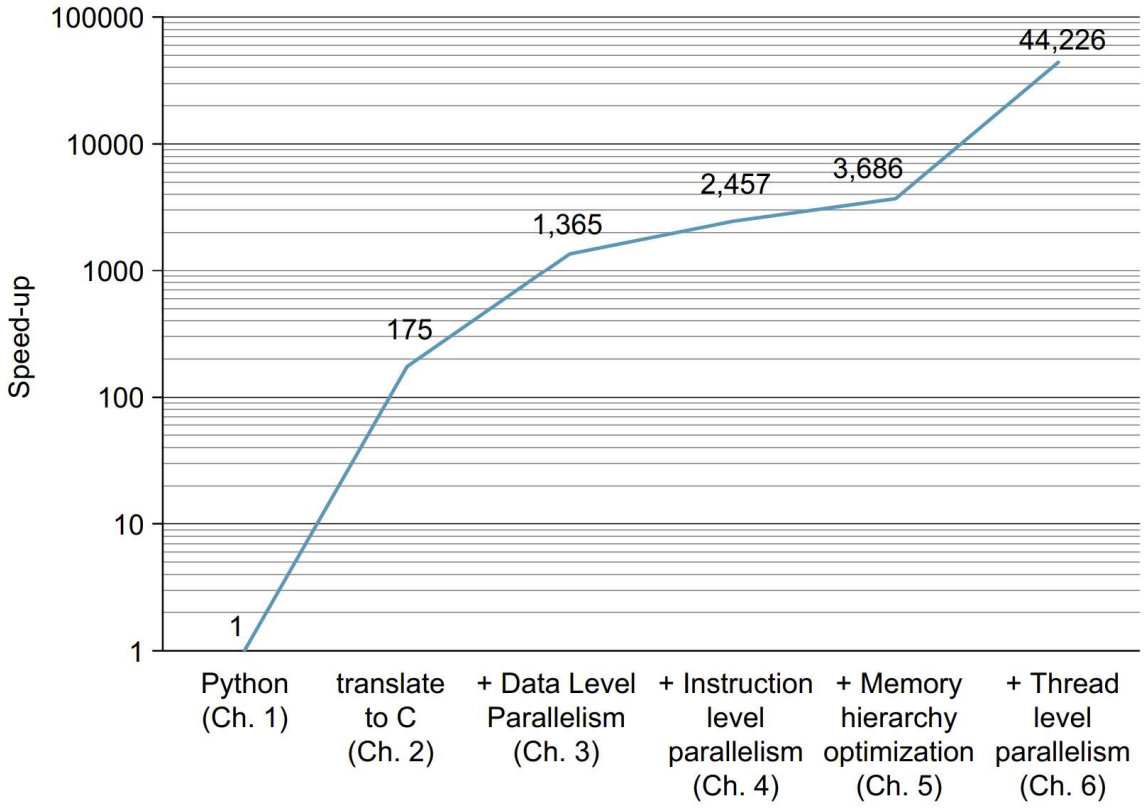


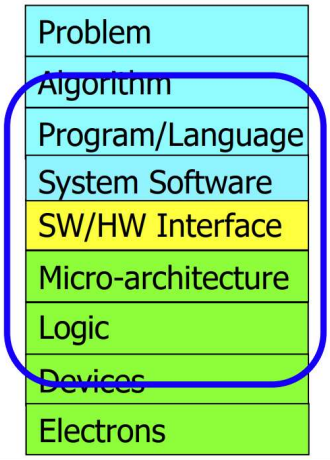
FIGURE 1.20 Optimizations of matrix multiply program in Python in the next five chapters of this book.

Axiom

To achieve the highest energy efficiency and performance:

we must take the expanded view

of computer architecture



Co-design across the hierarchy: Algorithms to devices

Specialize as much as possible within the design goals

Plateaued performance from a single core (Not just absolute performance, perf per watt, perf per $)

Latency

Throughput

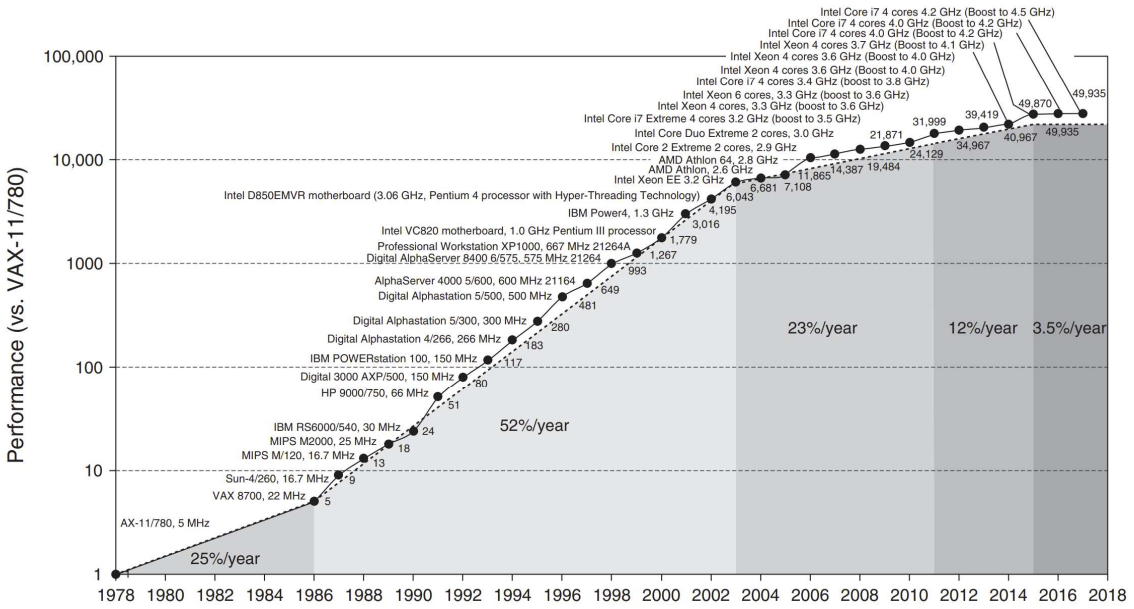


FIGURE 1.17 Growth in processor performance since the mid-1980s. This chart plots performance relative to the VAX 11/780 as measured by the SPECint benchmarks (see Section 1.11). Prior to the mid-1980s, processor performance growth was largely technology-driven and averaged about 25% per year. The increase in growth to about 52% since then is attributable to more advanced architectural and organizational ideas. The higher annual performance improvement of 52% since the mid-1980s meant performance was about a factor of seven larger in 2002 than it would have been had it stayed at 25%. Since 2002, the limits of power, available instruction-level parallelism, and long memory latency have slowed uniprocessor performance recently, to about 3.5% per year.

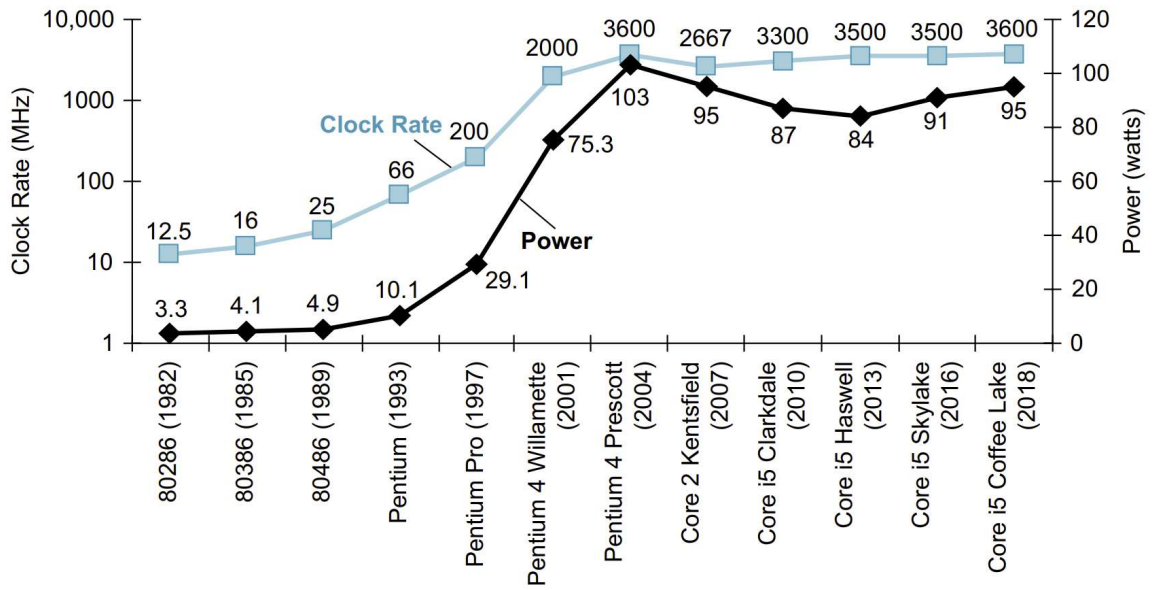
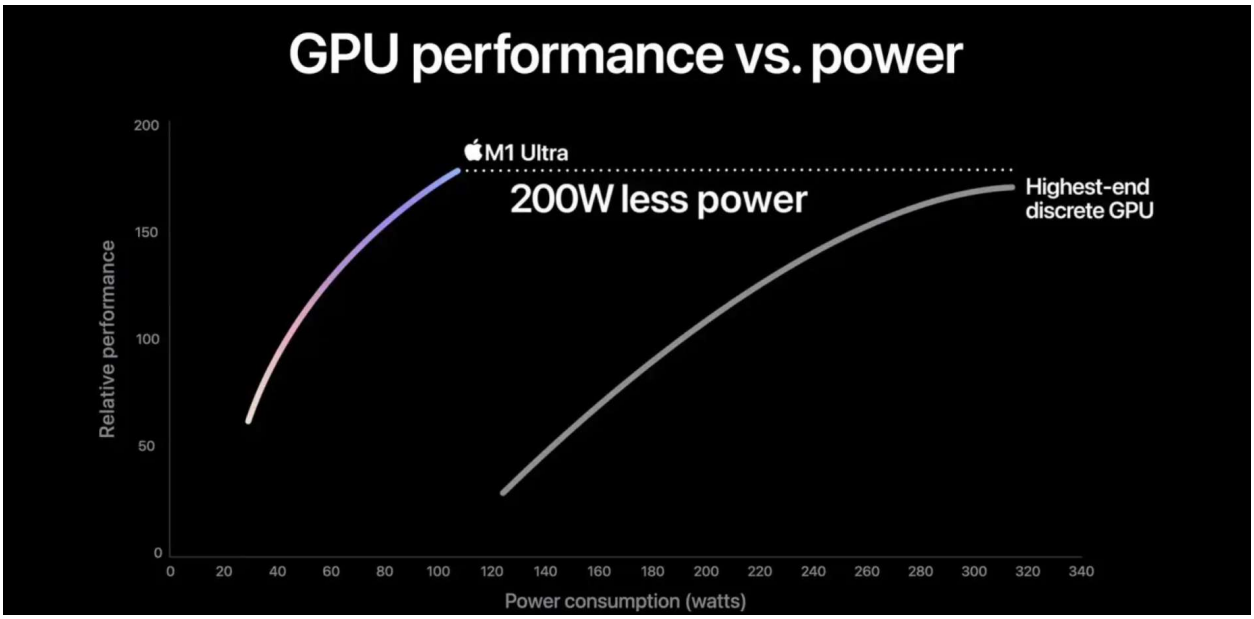
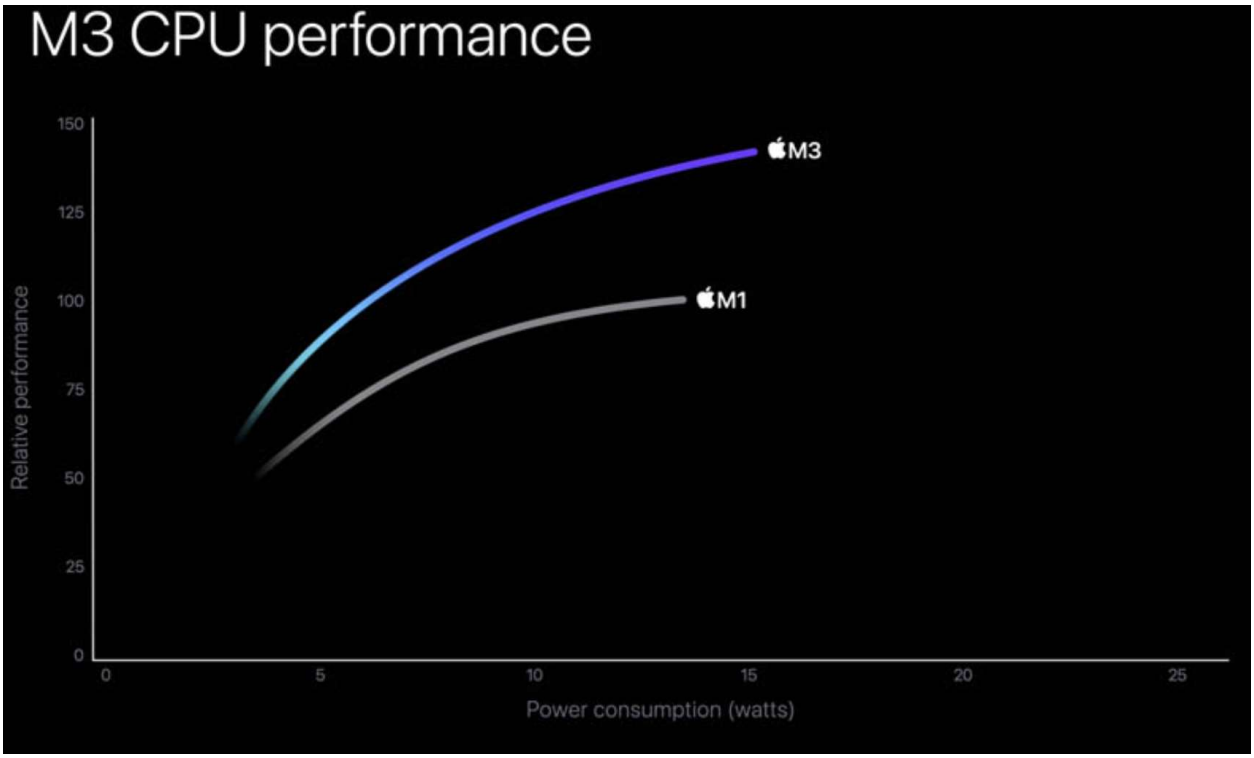


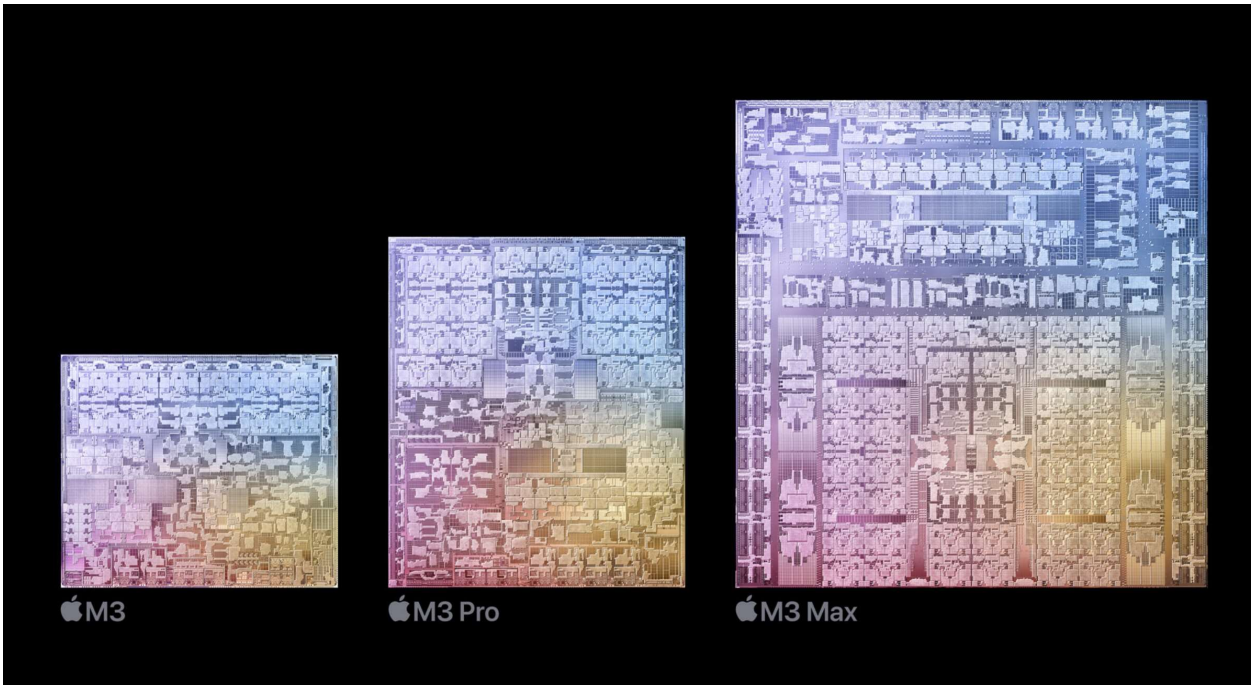
FIGURE 1.16 Clock rate and power for Intel x86 microprocessors over nine generations and 36 years. The Pentium 4 made a dramatic jump in clock rate and power but less so in performance. The Prescott thermal problems led to the abandonment of the Pentium 4 line. The Core 2 line reverts to a simpler pipeline with lower clock rates and multiple processors per chip. The Core i5 pipelines follow in its footsteps.

There is so much any single “computer” can do, no matter how big/expensive

For fault-tolerance (If one fails, someone else is there to take over)



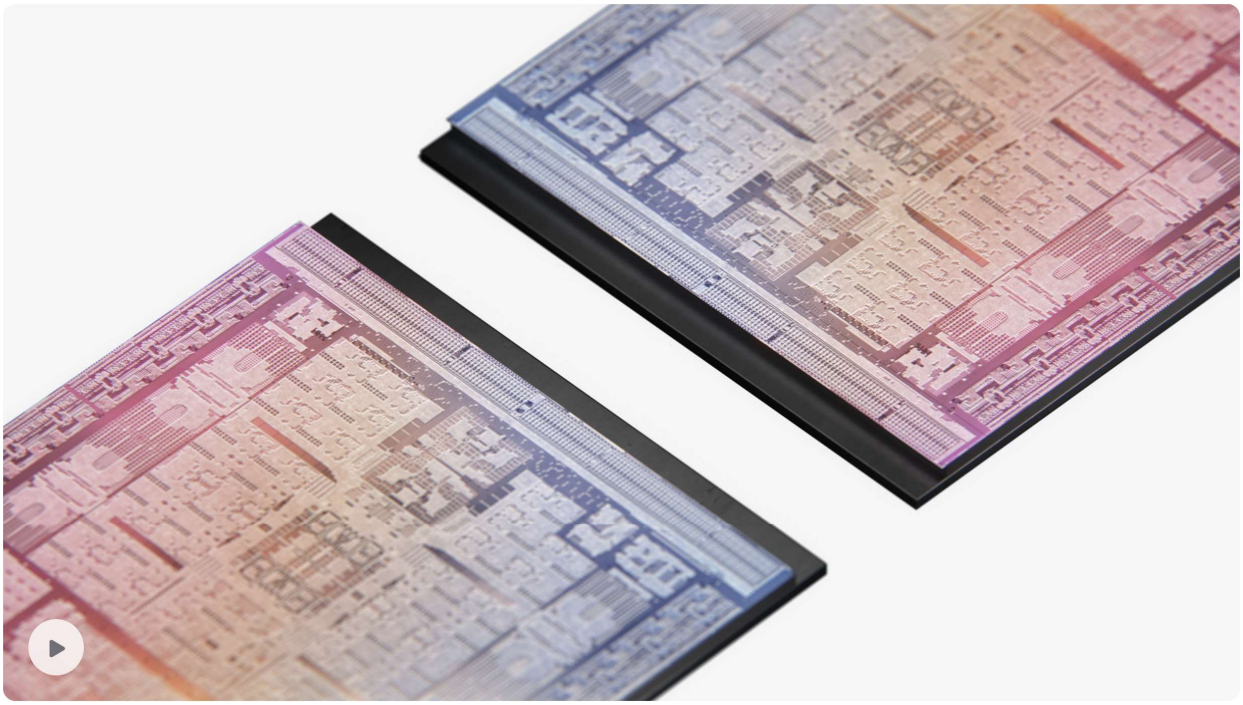


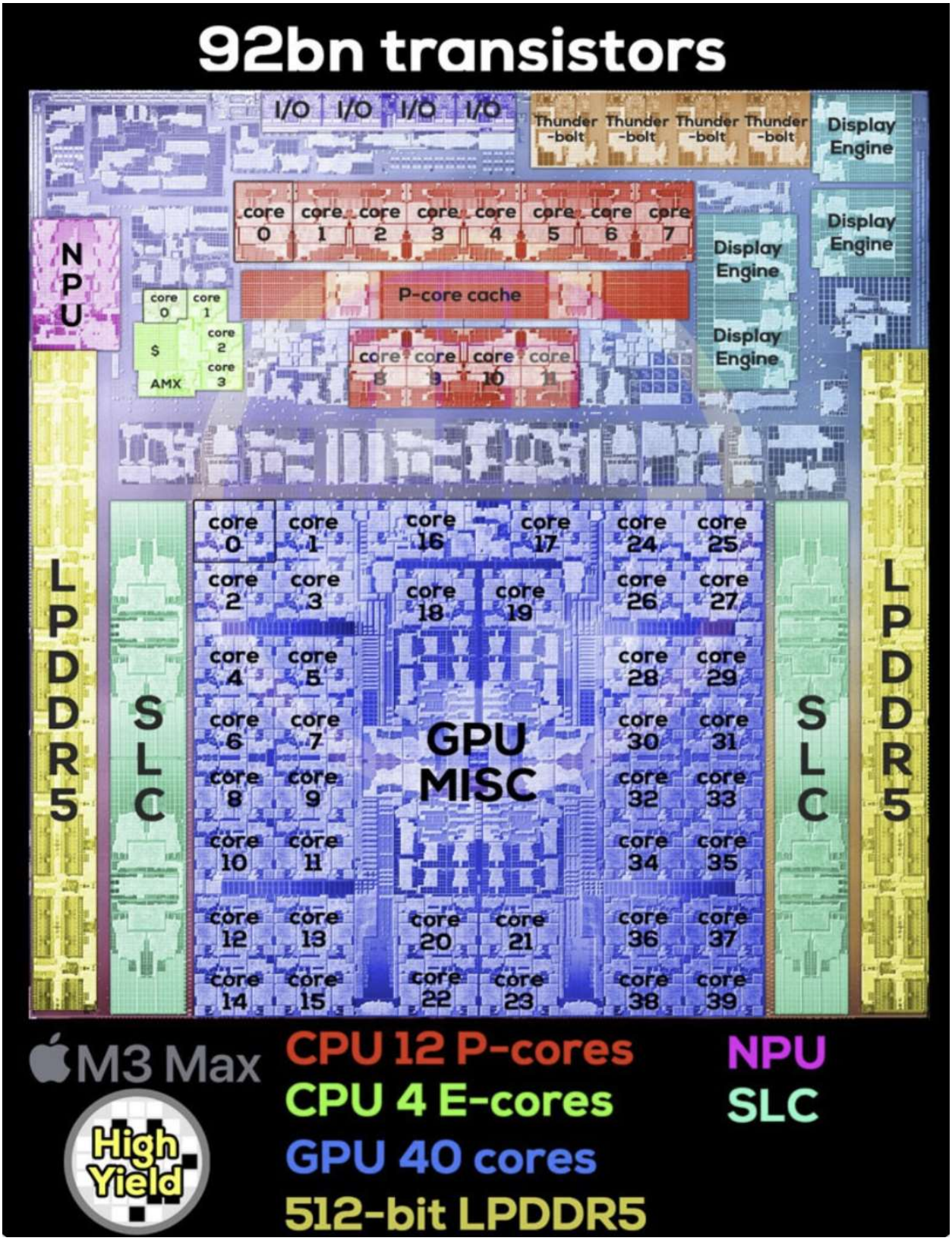

M3
M3 Pro
M3 Max

Apple M3
Apple M3 Pro
Apple M3 Max

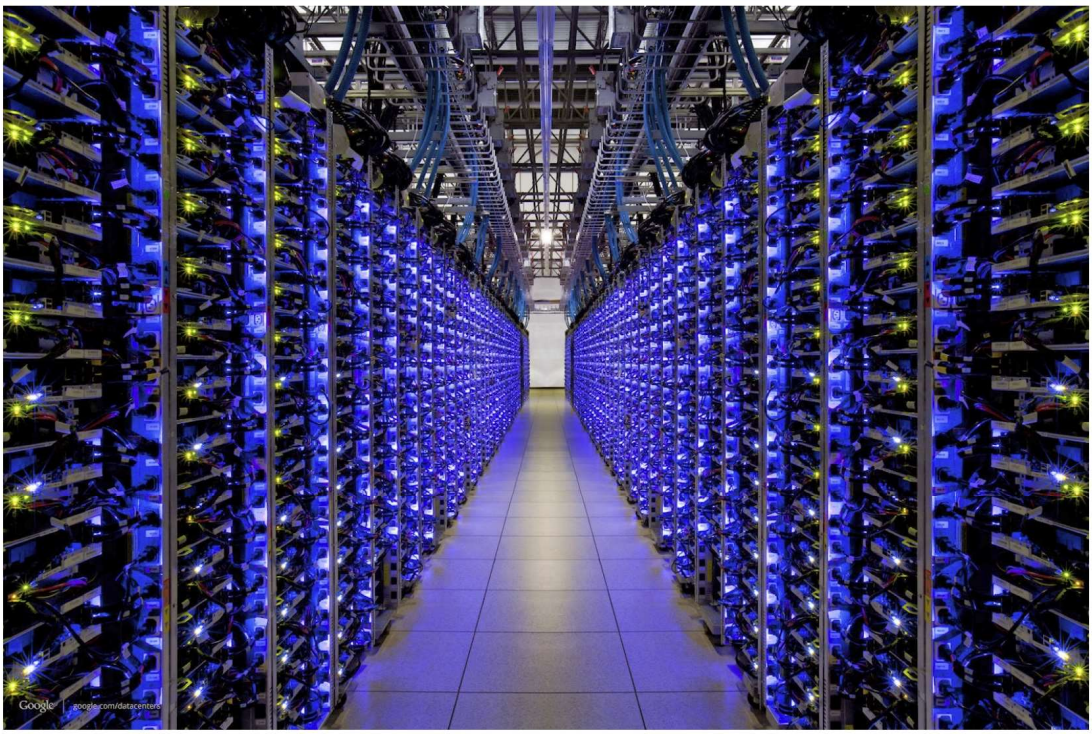
General information	
Launched	**M3, Pro and Max**: Oct 30, 2023
Designed by	Apple Inc.
Common manufacturer(s)	TSMC
Performance	
Max. CPU clock rate	**Performance cores**: 4.05 GHz[1] **Efficiency cores**: 2.75 GHz
Cache	
L1 cache	**Performance cores** 192+128 KiB per core **Efficiency cores** 128+64 KiB per core
L2 cache	**Performance cores** **M3 and M3 Pro:** 16 MiB **M3 Max:** 32 MiB **Efficiency cores** **M3, M3 Pro, M3 Max:** 4 MiB
Architecture and classification	
Application	**M3**: Desktop (iMac), Notebook (MacBook Pro) **M3 Pro**: Notebook (MacBook Pro) **M3 Max**: Notebook (MacBook Pro)

Technology node	3 nm
Instruction set	ARMv8.6-A[2]
Physical specifications	
Transistors	**M3**: 25 billion **M3 Pro**: 37 billion **M3 Max**: 92 billion
Cores	**M3**: 8 (4× high-performance + 4× high-efficiency) **M3 Pro**: 11–12 (5–6× high-performance + 6× high-efficiency) **M3 Max**: 14–16 (10–12× high-performance + 4× high-efficiency)
Memory (RAM)	**M3**: 8–24 GB (128-bit) **M3 Pro**: 18–36 GB (192-bit) **M3 Max**: 36–128 GB (384–512-bit)[3]
GPU(s)	Apple-designed integrated graphics **M3**: 8–10 core GPU **M3 Pro**: 14–18 core GPU **M3 Max**: 30–40 core GPU
Products, models, variants	
Variant(s)	Apple A17[4]
History	
Predecessor(s)	Apple M2




92bn transistors
I/O I/O I/O I/O
Thunder -bolt
Thunder -bolt
Thunder -bolt
Thunder -bolt
Display Engine
Display Engine
Display Engine
Display Engine
core 0 core 1 core 2 core 3 core 4 core 5 core 6 core 7
P-core cache
NPU
core 0
core 1
$
core 2
core 3
AMX
core 8 core 9 core 10 core 11
LPDDR5
SLC
GPU MISC
core 0 core 1 core 16 core 17 core 24 core 25
core 2 core 3 core 18 core 19 core 26 core 27
core 4 core 5 core 28 core 29
core 6 core 7 core 30 core 31
core 8 core 9 core 32 core 33
core 10 core 11 core 34 core 35
core 12 core 13 core 20 core 21 core 36 core 37
core 14 core 15 core 22 core 23 core 38 core 39
SLC
LPDDR5
M3 Max
High Yield
CPU 12 P-cores
CPU 4 E-cores
GPU 40 cores
512-bit LPDDR5
NPU
SLC

Different Platforms, Different Goals



Source: http://datacentervoice.com/wp-content/uploads/2015/10/data-center.jpg


FRONTIER
OAK RIDGE
National Laboratory
U.S. DEPARTMENT OF
ENERGY
Hewlett Packard
Enterprise
AMD

FRONTIER - HPE CRAY EX235A, AMD OPTIMIZED 3RD GENERATION EPYC 64C 2GHZ, AMD INSTINCT MI250X, SLINGSHOT-11

Site:	DOE/SC/Oak Ridge National Laboratory
System URL:	https://www.olcf.ornl.gov/frontier/
Manufacturer:	HPE
Cores:	8,699,904
Processor:	AMD Optimized 3rd Generation EPYC 64C 2GHz
Interconnect:	Slingshot-11
Installation Year:	2021
Performance	
Linpack Performance (Rmax)	1,194.00 PFlop/s
Theoretical Peak (Rpeak)	1,679.82 PFlop/s
Nmax	24,219,648
HPCG [TFlop/s]	14,054.0
Power Consumption	
Power:	22,703.00 kW (Submitted)
Power Measurement Level:	1
Software	
Operating System:	HPE Cray OS

Challenge:

How to program all this computing stuff that has so much diversity?

Programmers cant passively sit idle like in old days!

How to measure performance? What is performance?:

Airplane	Passenger capacity	Cruising range (miles)	Cruising speed (m.p.h.)	Passenger throughput (passengers × m.p.h.)
Boeing 737	240	3000	564	135,360
BAC/Sud Concorde	132	4000	1350	178,200
Boeing 777-200LR	301	9395	554	166,761
Airbus A380-800	853	8477	587	500,711

FIGURE 1.14 The capacity, range, and speed for a number of commercial airplanes. The last column shows the rate at which the airplane transports passengers, which is the capacity times the cruising speed (ignoring range and takeoff and landing times).

(Speed of sound: ~ 767 miles per hour. Concorde rough Mach 2)

The Classic CPU Performance Equation

We can now write this basic performance equation in terms of **instruction count** (the number of instructions executed by the program), CPI, and clock cycle time:

$$\text{CPU time} = \text{Instruction count} \times \text{CPI} \times \text{Clock cycle time}$$

or, since the clock rate is the inverse of clock cycle time:

$$\text{CPU time} = \frac{\text{Instruction count} \times \text{CPI}}{\text{Clock rate}}$$

What we want? More "powerful" computers.

Why: To solve problems that matter (harder and harder problems. Ex: Billions of parameters, Bigger worlds in gaming, More users, more requests per second …)

Performance per watt:

"Replacing large inefficient processors with many smaller, efficient processors can deliver better performance per joule both in the large and in the small, but only if software can efficiently use them."

Uni processor to multi-core movement:

"It emphasizes that understanding the underlying hardware trends and learning to adapt software to them is where innovation and technical advances will occur in the years ahead."

Two broad ways to parallelize: One where the end programmer needs to activiely find parallelism in the workload (and the underlying system such as runtime and OS needs to map it to the available hardware). The second is where the programmer does not do much effort to find and expose parallelism and only the compilers/runtime/OS/processor do the hard work of finding different kinds of parallelisms.

Asymptotic complexity vs reducing the constants:

(a) Find the “best” available algorithm
(b) Then try to reduce the constants

As engineers we need to see when to use which alogirhtms. At times we might be using a “slower” algorithm

Example: Insertion sort vs quick sort when number of elements are small

Example 2: Summing up an array using one processor vs two processors

Aspects of fault tolerance: A broken processor should not impact program. Or usually a program should be able to be agnostic of the number of processing elements present. But there are challenges. We can get thrashing if we blindly start creating threads.

task-level parallelism or process-level parallelism

Utilizing multiple processors by running independent programs simultaneously.

parallel processing program

A single program that runs on multiple processors simultaneously.

Metrics:

Greater throughput

Lower response time (execution time)

Cluster

A set of computers connected over a local area network that function as a single large multiprocessor.

In addition, clusters can serve equally demanding applications outside the sciences, such as search engines, Web servers, email servers, and databases.

Traditional discussion of clock speed and CPI:

SMP:

processors are often called cores in a multicore chip. The number of cores is expected to increase with Moore's Law. These multicores are almost always Shared Memory Processors (SMPs), as they usually share a single physical address space.

shared memory multiprocessor (SMP)

A parallel processor with a single physical address space.

"The state of technology today means that programmers who care about performance must become parallel programmers, for sequential code now means slow code."

"The tall **challenge** facing the industry is to create hardware and software that will make it easy to write **correct** parallel processing programs that will execute efficiently in performance and energy as the number of cores per chip scales.

		Software	
		Sequential	**Concurrent**
Hardware	Serial	Matrix Multiply written in MatLab running on an Intel Pentium 4	Windows Vista Operating System running on an Intel Pentium 4
	Parallel	Matrix Multiply written in MATLAB running on an Intel Core i7	Windows Vista Operating System running on an Intel Core i7

FIGURE 6.1 Hardware/software categorization and examples of application perspective on concurrency versus hardware perspective on parallelism.

"You might guess that the only challenge of the parallel revolution is figuring out how to make naturally sequential software have high performance on parallel hardware, but it is also to make concurrent programs have high performance on multiprocessors as the number of processors increases."

Check Yourself

True or false: To benefit from a multiprocessor, an application must be concurrent

Challenges in distributed systems:

Head to the blog: [The joy and pain of distributed systems]