

Machine-Level Programming I: Basics

- History of Processors -

CENG331 - Computer Organization

Middle East Technical University

Instructor:

Murat Manguoğlu (Section 1)

Slides 6-33 are adapted from the slides of the textbook: D. A. Patterson and J. L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, 3rd Edition

Others are adapted from slides of the textbook: <http://csapp.cs.cmu.edu/>

Computer Architecture: A Little History

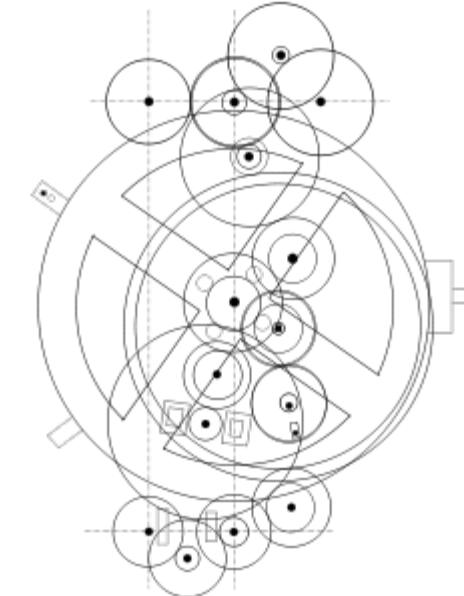
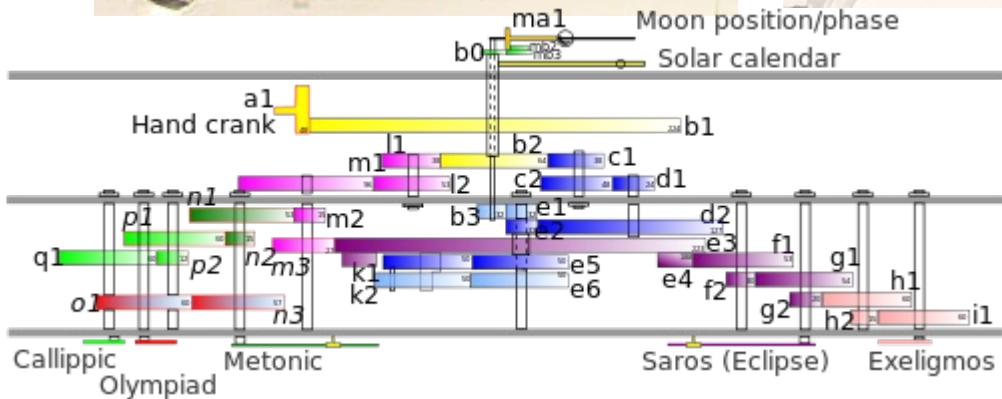
Why worry about old ideas?

- *Die Geschichte der Wissenschaft ist die Wissenschaft selbst* (The history of science is science itself) – Johann Wolfgang von Goethe
1749-1832
 - In fact, if you read any scientific paper/thesis/work, you will notice it starts with the review of the literature (i.e. History) of the earlier work
- Helps to illustrate the design process, and explains why certain decisions were taken
- Because future technologies might be as constrained as older ones
- **Those who ignore history are doomed to repeat it**
 - Every mistake made in mainframe design was also made in minicomputers, then microcomputers, where next?

Antikythera Mechanism

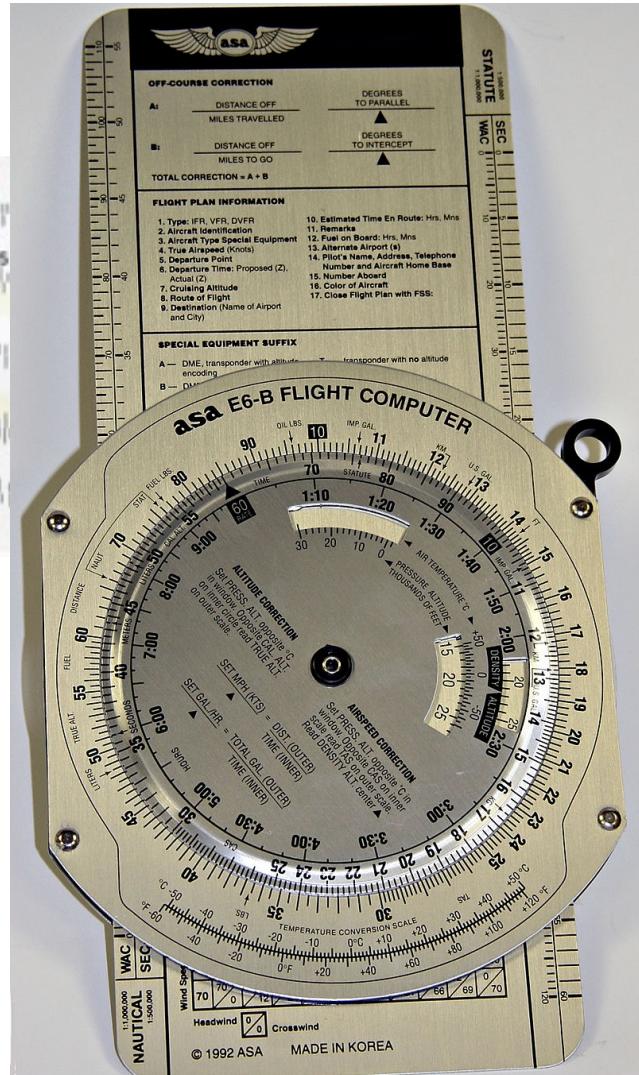
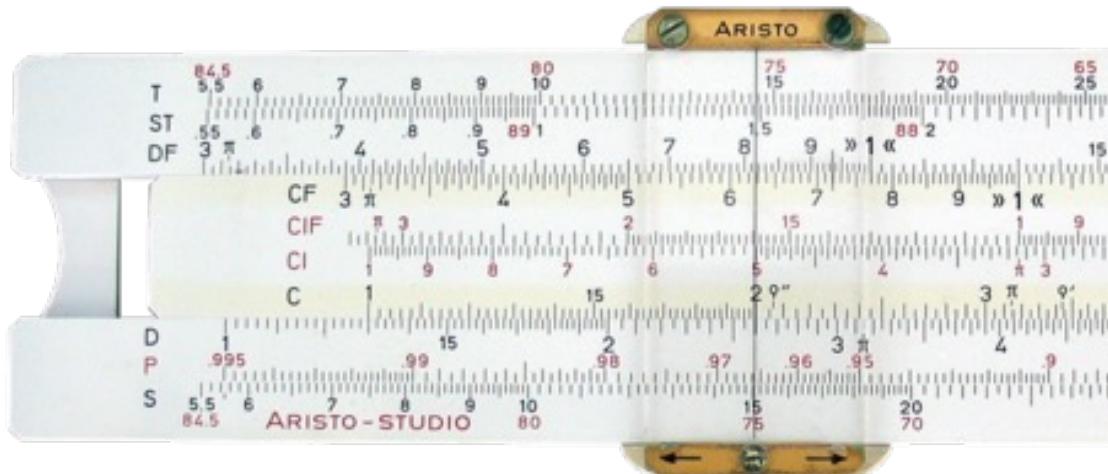
150-100 BC

An astronomical calendar capable of tracking the position of the sun, moon, and planets; predict eclipses, and even recreate the irregular orbit of the moon



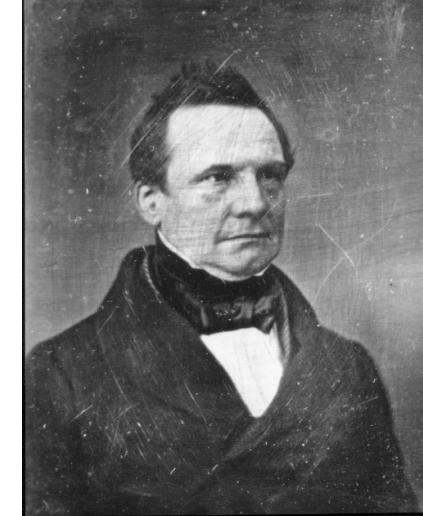
Source: http://en.wikipedia.org/wiki/Antikythera_mechanism

Slide ruler ~1620 and still used today (but only as a backup “computer” on air or sea)



Charles Babbage

Lucasian Professor of Mathematics,
Cambridge University, 1827-1839



■ *Difference Engine* 1823

■ *Analytic Engine* 1833

- The forerunner of modern digital computer!

Application

- Mathematical Tables – Astronomy
- Nautical Tables – Navy

Background

- Any continuous function can be approximated by a polynomial --- Weierstrass

Technology

- mechanical - gears, Jacquard's loom, simple calculators

Difference Engine

A machine to compute mathematical tables

Weierstrass:

- Any continuous function can be approximated by a polynomial
- Any polynomial can be computed from *difference tables*

An example

$$f(n) = n^2 + n + 41$$

$$d1(n) = f(n) - f(n-1) = 2n$$

$$d2(n) = d1(n) - d1(n-1) = 2$$

$$f(n) = f(n-1) + d1(n) = f(n-1) + (d1(n-1) + 2)$$



Weierstraß

all you need is an adder!

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

Difference Engine

1823

- Babbage's paper is published

1834

- The paper is read by Scheutz & his son in Sweden

1842

- Babbage gives up the idea of building it; he is onto Analytic Engine!

1855

- Scheutz displays his machine at the Paris World Fare
- Can compute any 6th degree polynomial
- *Speed:* 33 to 44 32-digit numbers per minute!



Now the machine is at the Smithsonian

Analytic Engine

1833: Babbage's paper was published

- *conceived during a hiatus in the development of the difference engine*

Inspiration: *Jacquard Loom Machine (1804)*

- looms were controlled by punched cards
 - The set of cards with fixed punched holes dictated the pattern of weave ⇒ *program*
 - The same set of cards could be used with different colored threads ⇒ *numbers*

1871: Babbage dies

- The machine remains unrealized.



It is not clear if the analytic engine could be built using the mechanical technology of the time

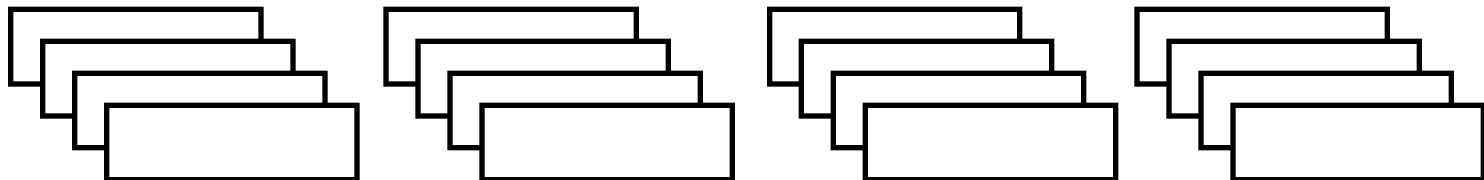
Analytic Engine

The first conception of a general-purpose computer

- The *store* in which all variables to be operated upon, as well as all those quantities which have arisen from the results of the operations are placed.
- The *mill (arithmetic logic unit)* into which the quantities about to be operated upon are always brought.

The *program*
Operation

variable1 variable2 variable3



An operation in the *mill* required feeding two punched cards and producing a new punched card for the *store*.

An operation to alter the sequence was also provided!

The first programmer

Ada Byron *aka* “Lady Lovelace” 1815-52



Ada's tutor was Babbage himself!

Babbage's Influence

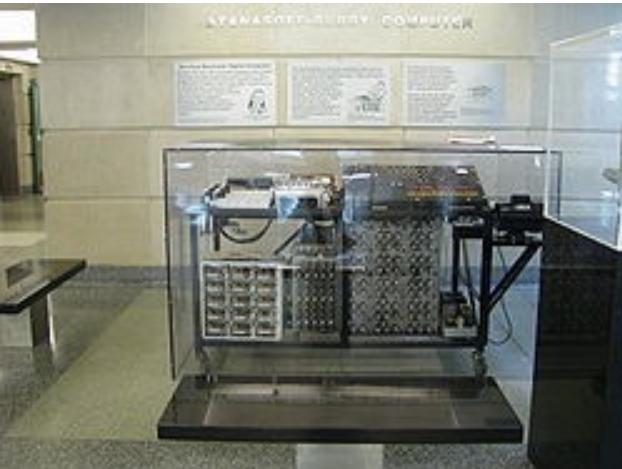
- Babbage's ideas had great influence later primarily because of
 - Luigi Menabrea, who published notes of Babbage's lectures in Italy
 - Lady Lovelace, who translated Menabrea's notes in English and thoroughly expanded them.
“... Analytic Engine weaves *algebraic patterns....*”
- In the early twentieth century - the focus shifted to analog computers but
 - Harvard Mark I built in 1944 is very close in spirit to the Analytic Engine.

Linear Equation Solver

John Atanasoff, Iowa State University

1930's:

- Atanasoff built the Linear Equation Solver.
- It had 300 tubes!
- Special-purpose binary digital calculator
- Dynamic RAM (stored values on refreshed capacitors)



Application:

- Linear and Integral differential equations

Background:

- Vannevar Bush's Differential Analyzer
--- *an analog computer*

Technology:

- Tubes and Electromechanical relays

Atanasoff decided that the correct mode of computation was using electronic binary digits.

Harvard Mark I

AIKEN - IBM AUTOMATIC SEQUENCE

■ 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)

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
 - 18,000 vacuum tubes
- Performance
 - Read in 120 cards per minute
 - Addition took 200 μ s, Division 6 ms
 - 1000 times faster than Mark I
- Not very reliable!

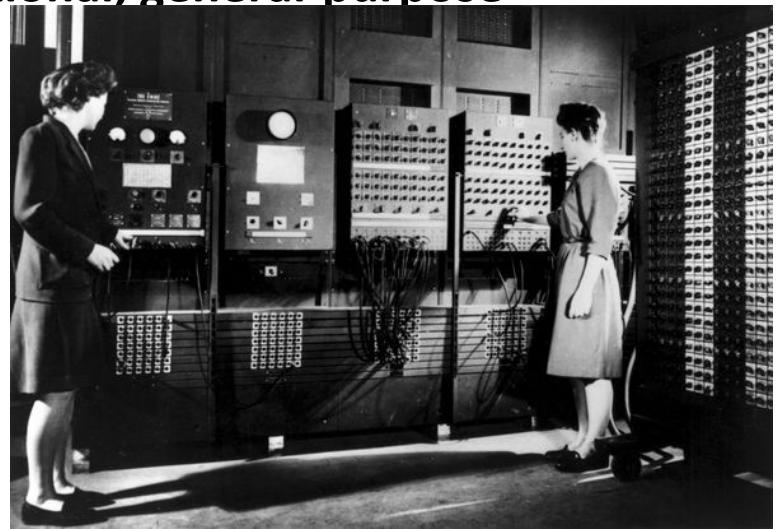


Image source: <https://www.computerhistory.org/revolution/birth-of-the-computer/4/78/317>

Application: Ballistic calculations

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

WW-2 Effort



Stored Program Computer

Program = A sequence of instructions

How to control instruction sequencing?

manual control

calculators

automatic control

external (paper tape)

Harvard Mark I , 1944

Zuse's Z1, WW2

internal

plug board

ENIAC 1946

read-only memory

ENIAC 1948

read-write memory

EDVAC 1947 (*concept*)

- The same storage can be used to store program and data

EDSAC

1950

Maurice Wilkes

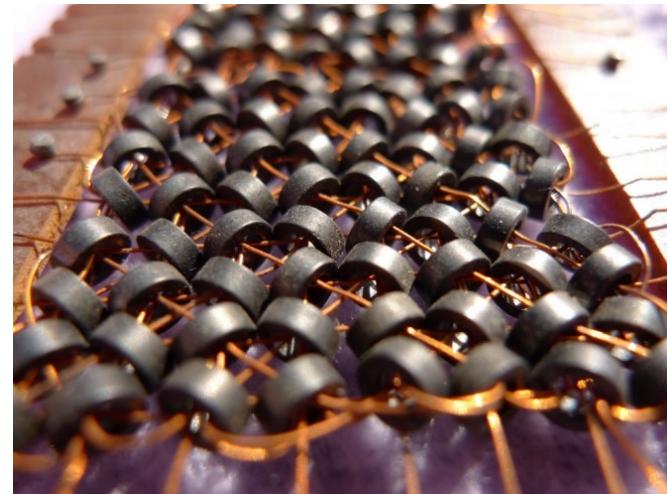
Dominant Problem: *Reliability*

Mean time between failures (MTBF)

MIT's Whirlwind with an MTBF of 20 min. was perhaps the most reliable machine !

Reasons for unreliability:

1. Vacuum Tubes
2. Storage medium
 - acoustic delay lines
 - mercury delay lines
 - Williams tubes
 - Selections



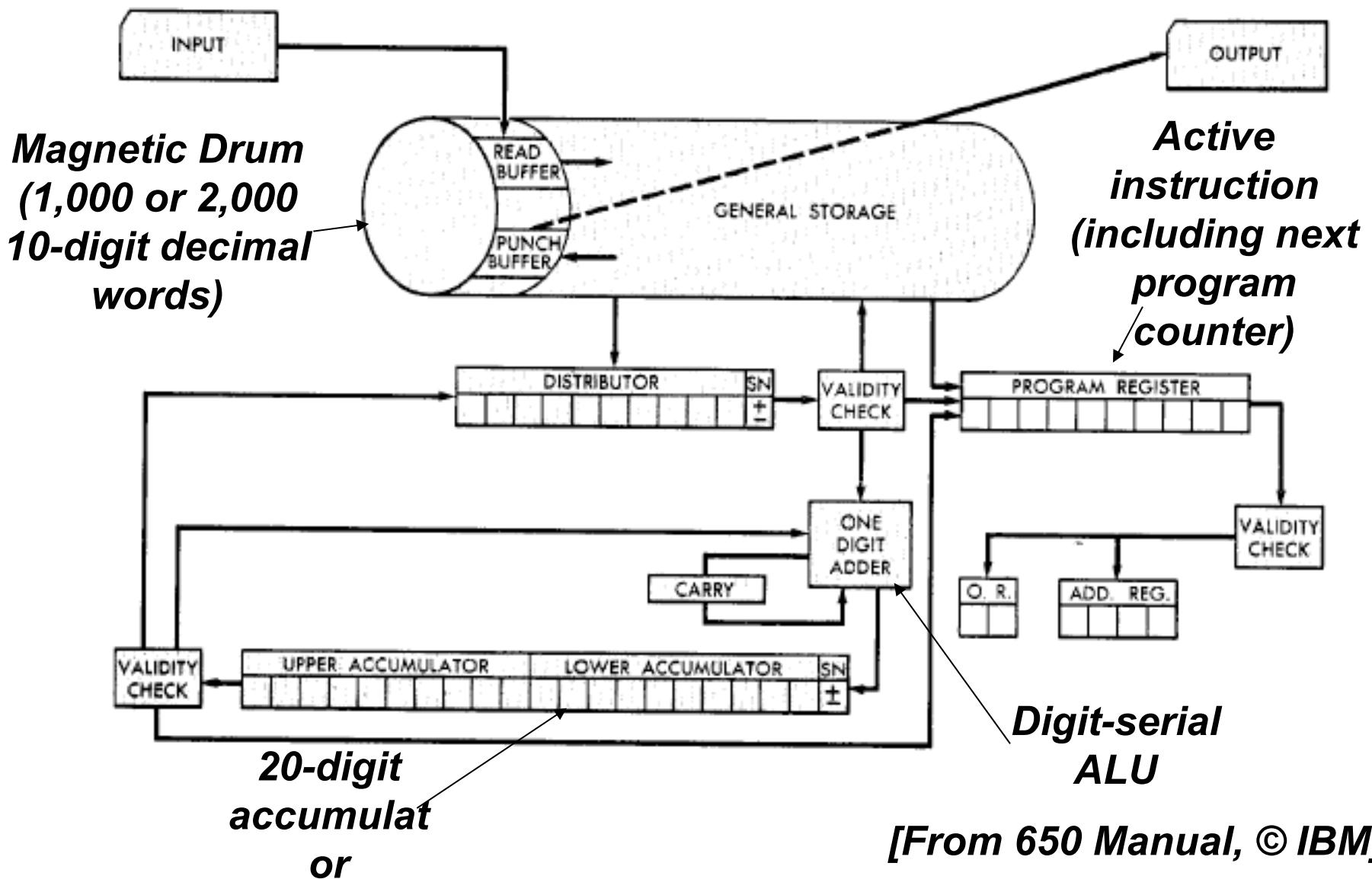
<http://www.wired.com/2010/05/0511magnetic-core-memory/>

Reliability solved by invention of **Core** memory by J. Forrester 1954 at MIT for Whirlwind project

Computers in mid 50's

- **Hardware was expensive**
- **Stores 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**

The IBM 650 (1953-4)



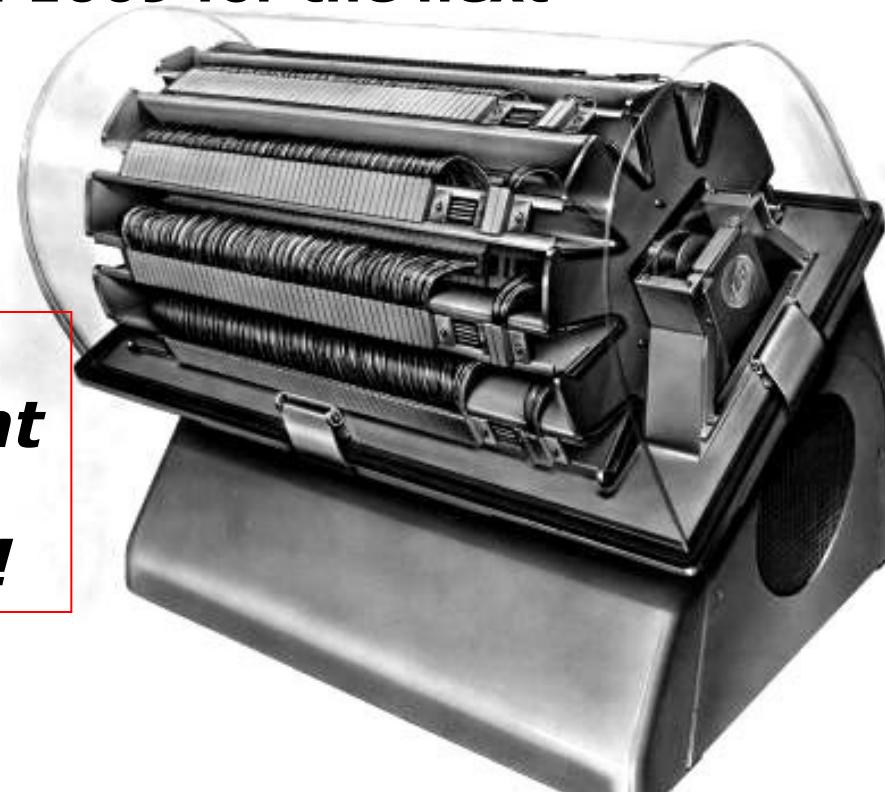
Programmer's view of the IBM 650

A drum machine with 44 instructions

Instruction: 60 1234 1009

- “Load the contents of location 1234 into the *distribution*; put it also into the *upper accumulator*; set *lower accumulator* to zero; and then go to location 1009 for the next instruction.”

Good programmers optimized the placement of instructions on the drum to reduce latency!



Variety of Instruction Formats

- ***Zero address format: Stack***
- ***One address formats: Accumulator machines***
Accumulator is always other source and destination operand
- ***Two address formats: the destination is same as one of the operand sources***

(Reg \times Reg) to Reg		$R_I \leftarrow (R_I) + (R_J)$
(Reg \times Mem) to Reg		$R_I \leftarrow (R_I) + M[x]$
- ***Three address formats: One destination and up to two operand sources per instruction***

(Reg \times Reg) to Reg		$R_I \leftarrow (R_J) + (R_K)$
(Reg \times Mem) to Reg		$R_I \leftarrow (R_J) + M[x]$

Stack Machines (Mostly) Died by 1980

1. Stack programs are not smaller if short (Register) addresses are permitted.
2. Modern compilers can manage fast register space better than the stack discipline.

GPR's and caches are better than stacks

Early language-directed architectures often did not take into account the role of compilers!

B5000, B6700, HP 3000, ICL 2900, Symbolics 3600

Some would claim that an echo of this mistake is visible in the SPARC architecture register windows

Stacks post-1980

- **Inmos Transputers (1985-2000)**
 - Designed to support many parallel processes in Occam language
 - Fixed-height stack design simplified implementation
 - Stack trashed on context swap (fast context switches)
 - Inmos T800 was world's fastest microprocessor in late 80's
- **Forth machines**
 - Direct support for Forth execution in small embedded real-time environments
 - Several manufacturers (Rockwell, Patriot Scientific)
- **Java Virtual Machine**
 - Designed for software emulation, not direct hardware execution
 - Sun PicoJava implementation + others
- **Intel x87 floating-point unit**
 - Severely broken stack model for FP arithmetic
 - Deprecated in Pentium-4, replaced with SSE2 FP registers

Electronic analog computers



1960 Newmark analogue computer, made up of five units. This computer was used to solve differential equations



ELWAT , Poland, 1967



AKAT-1 , Poland, 1959

Software Developments

Libraries of numerical routines

- Floating point operations
- Transcendental functions
- Matrix manipulation,
equation solvers, . . .

Machines required experienced operators

⇒ Most users could not be expected to understand these programs, much less write them

⇒ Machines had to be sold with a lot of resident software

1955

*High level Languages - Fortran 1956
Operating Systems -*

- Assemblers, Loaders, Linkers, Compilers
- Accounting programs to keep track of usage and charges

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: 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!

Intel x86 Processors

- “Mostly” Dominate laptop/desktop/server market
- Design principle
 - Backwards compatible up until 8086, introduced in 1978
 - Added more features as time goes on
- Complex instruction set computer (CISC)
 - Many different instructions with many different formats
 - But, only small subset encountered with Linux programs
 - Hard to match performance of Reduced Instruction Set Computers (RISC – Examples: IBM Power, Arm, ...)
 - But, Intel seemed to be doing just that!
 - In terms of speed. Less so for low power.

x86 Clones: Advanced Micro Devices (AMD)

■ Historically

- AMD has followed Intel, but sometimes Intel followed AMD too.
- A little bit slower, a lot cheaper

■ Then

- Recruited top circuit designers from Digital Equipment Corp. and other downward trending companies
- Built Opteron: tough competitor to Pentium 4
- Developed x86-64, their own extension to 64 bits

■ Recent Years

- Intel got its act together
 - Still leads the world in semiconductor technology
- AMD has fallen behind
 - Relies on external semiconductor manufacturer

■ More recently (than the book)

- Intel is falling behind
- Apple and AMD , and Nvidia are starting to lead the market

Intel and AMD's 64-Bit History

■ 2001: Intel Attempts Radical Shift from IA32 to IA64

- Totally different architecture (Itanium)
- Executes IA32 code only as legacy
- Performance disappointing

■ 2003: AMD Steps in with Evolutionary Solution

- x86-64 (now called “AMD64”)

■ Intel Felt Obligated to Focus on IA64

- ~~Hard to admit mistake or that AMD is better, IA64 is actually not a bad architecture, latest one was produced in 2017 (Itanium 9760 – 8 cores 2.66Ghz 32MB Cache)~~

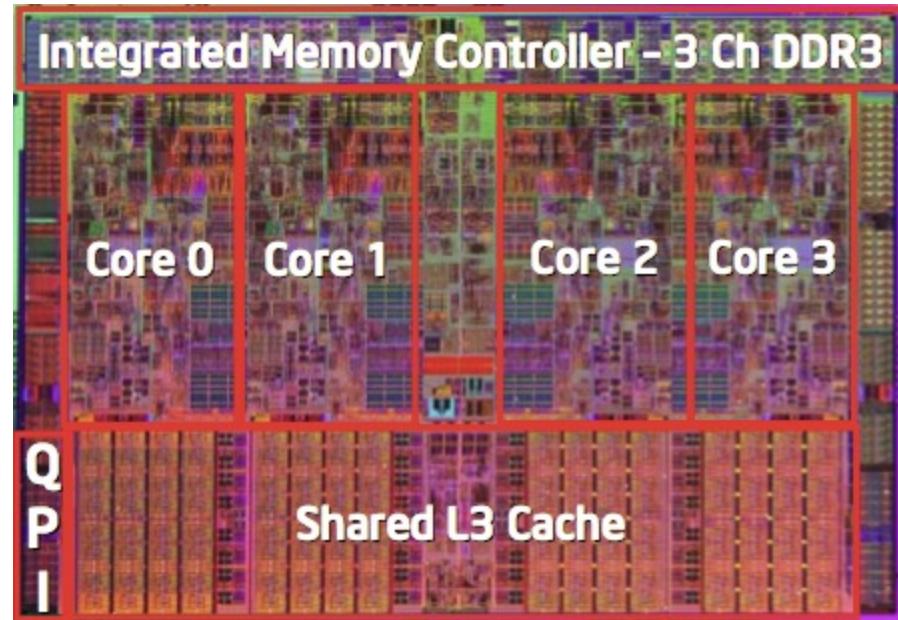
■ 2004: Intel Announces EM64T extension to IA32

- Extended Memory 64-bit Technology
- Almost identical to x86-64!

x86 Processors, cont.

■ Machine Evolution

■ 386	1985	0.3M
■ Pentium	1993	3.1M
■ Pentium/MMX	1997	4.5M
■ PentiumPro	1995	6.5M
■ Pentium III	1999	8.2M
■ Pentium 4	2001	42M
■ Core 2 Duo	2006	291M
■ Core i7	2008	731M
■ AMD epyc	2017	19.2B



■ Added Features

- Instructions to support multimedia operations
- Instructions to enable more efficient conditional operations
- Transition from 32 bits to 64 bits
- More cores

Intel (x86)

■ Core i9-9980XE (2018)

- 18 cores
- 36 threads
- 3.0-4.4 Ghz
- 24.75 MB Cache
- 14nm manufacturing tech.

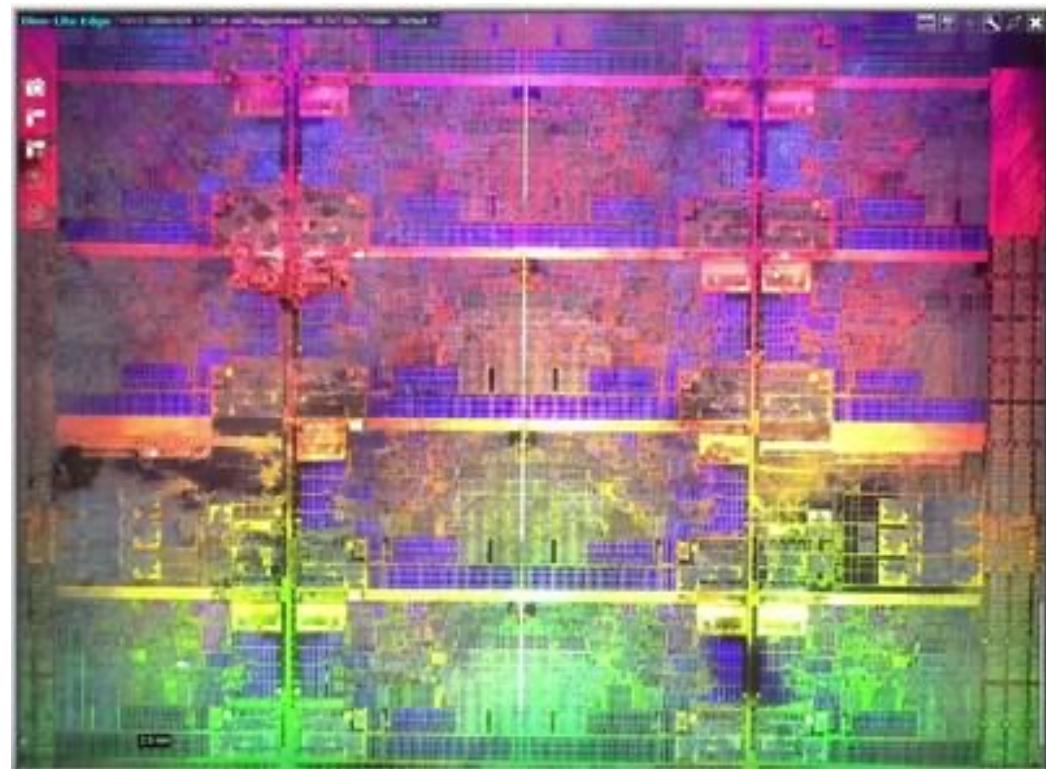
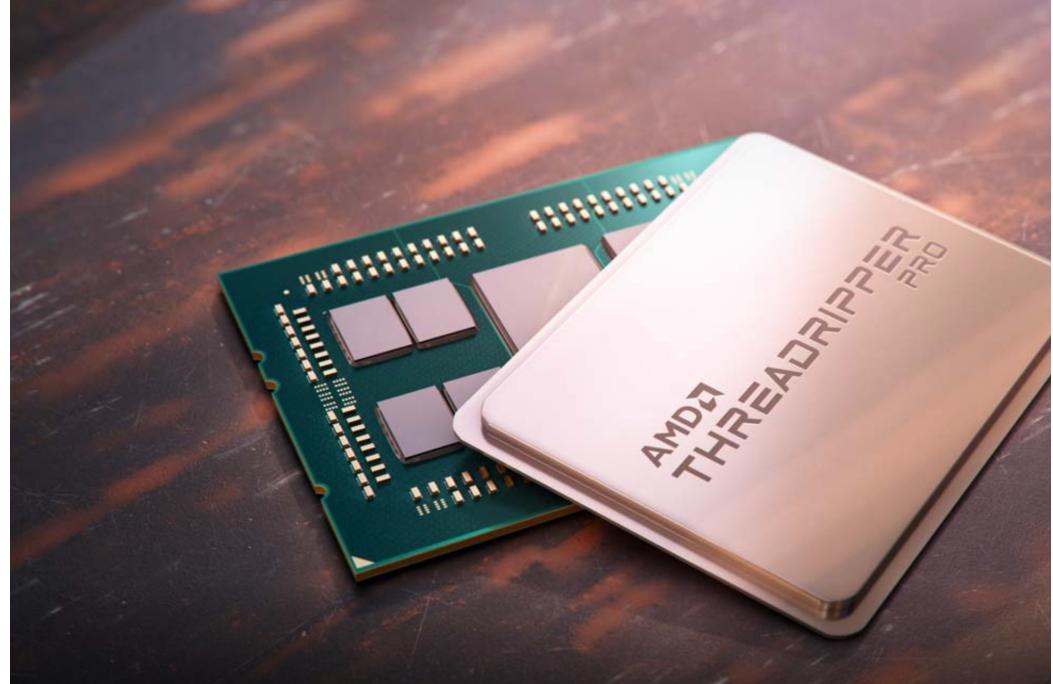


Image source: [https://hwbot.org/newsflash/5002_video_intel_core_i9_7980xe_die_extraction_with_de8auer_\(part_2\)](https://hwbot.org/newsflash/5002_video_intel_core_i9_7980xe_die_extraction_with_de8auer_(part_2))

AMD (x86)

- **Ryzen Threadripper PRO 5995WX**

- 64 Cores (128 threads)
- 4MB L1, 32MB L2, 256 MB L3
- 2.7-4.5 Ghz
- 7 nm manufacturing

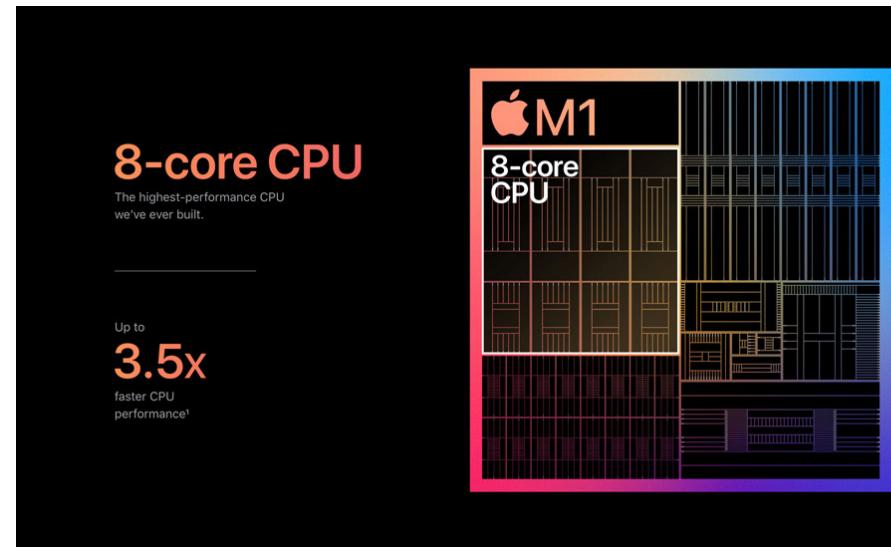


Apple (ARM: a RISC architecture)

■ M1 (2020) → M2(2022)

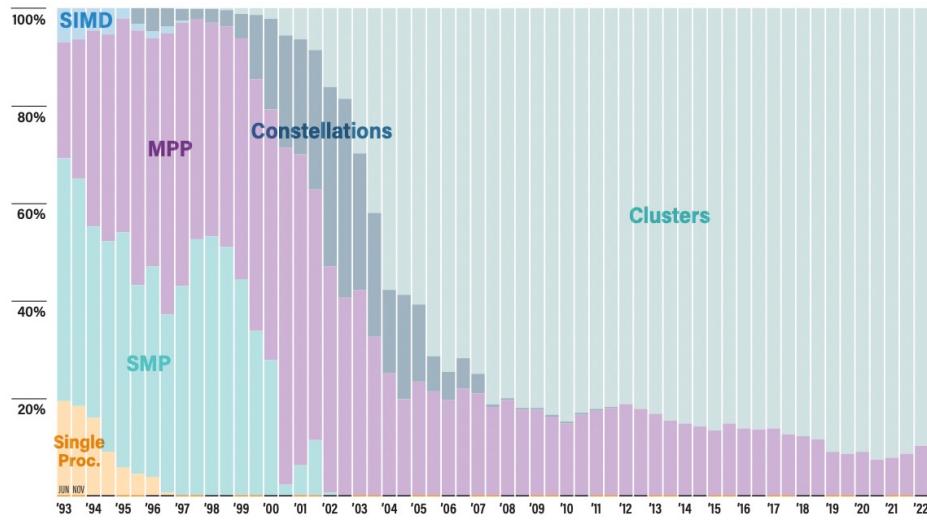
- 8 (4 high perf. + 4 eff.)
- Cache
 - 192KB L1-I + 128KB L1-D (perf. cores)
 - 12MB → 16 MB L2 (perf. cores)
 - 8 MB L3
- 3.2 GHz → 3.5 GHz
- Integrated GPU and Neural Engine
- **5nm*** manufacturing tech.

*2021: IBM announced 2nm manufacturing technology. Diameter of an atom is ~0.1-0.5 nm

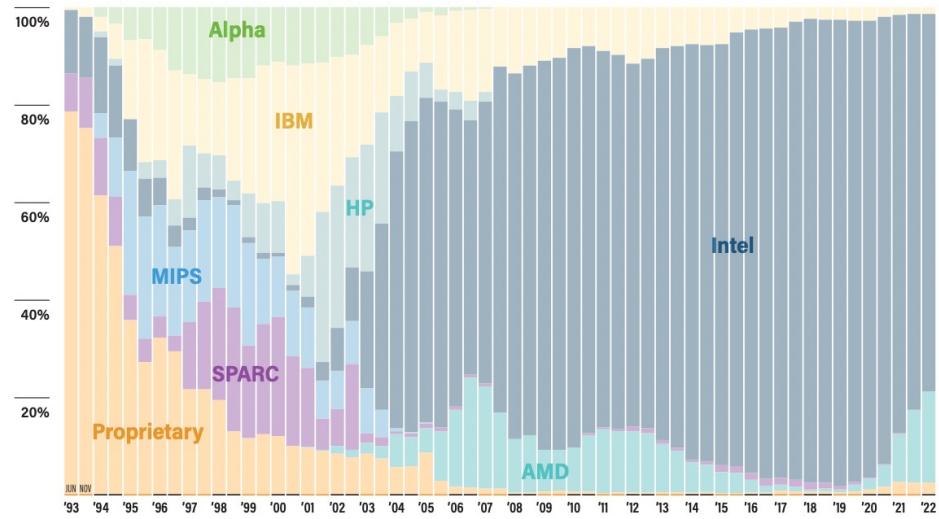


Most Powerful Computers

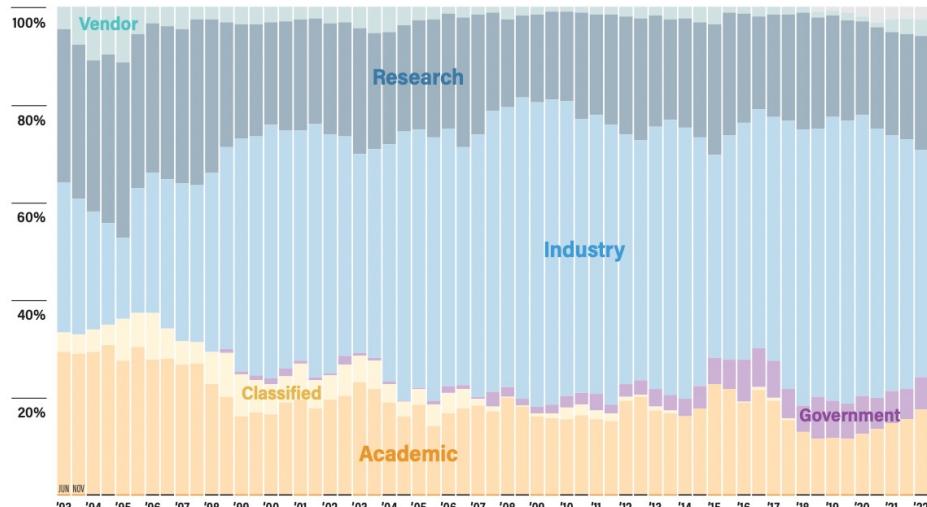
Architectures



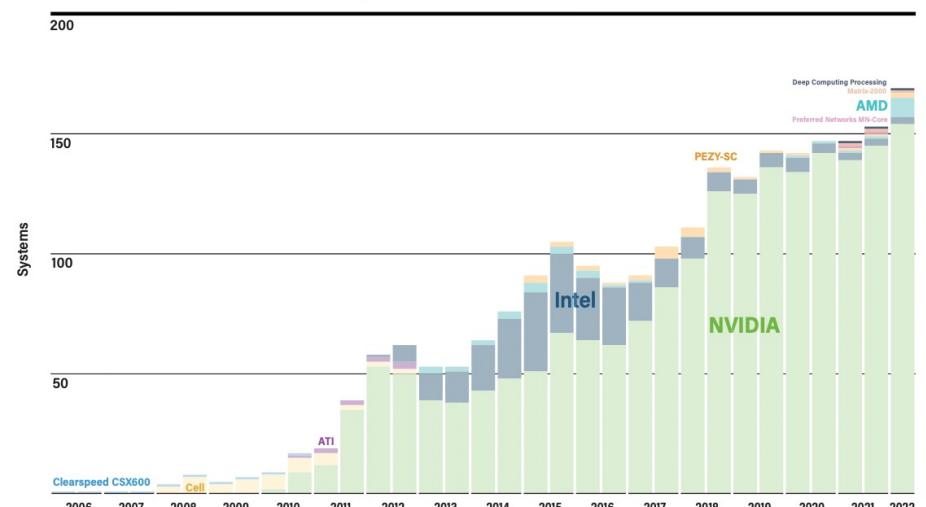
Chip Technology



Installation Type



Accelerators/Co-processors



NATIONAL BESTSELLER

INSIDE INTEL

UPDATED
EDITION

"A TRULY
FASCINATING
READ...the first
unauthorized
history of this
highly secretive
company."
—BARRON'S

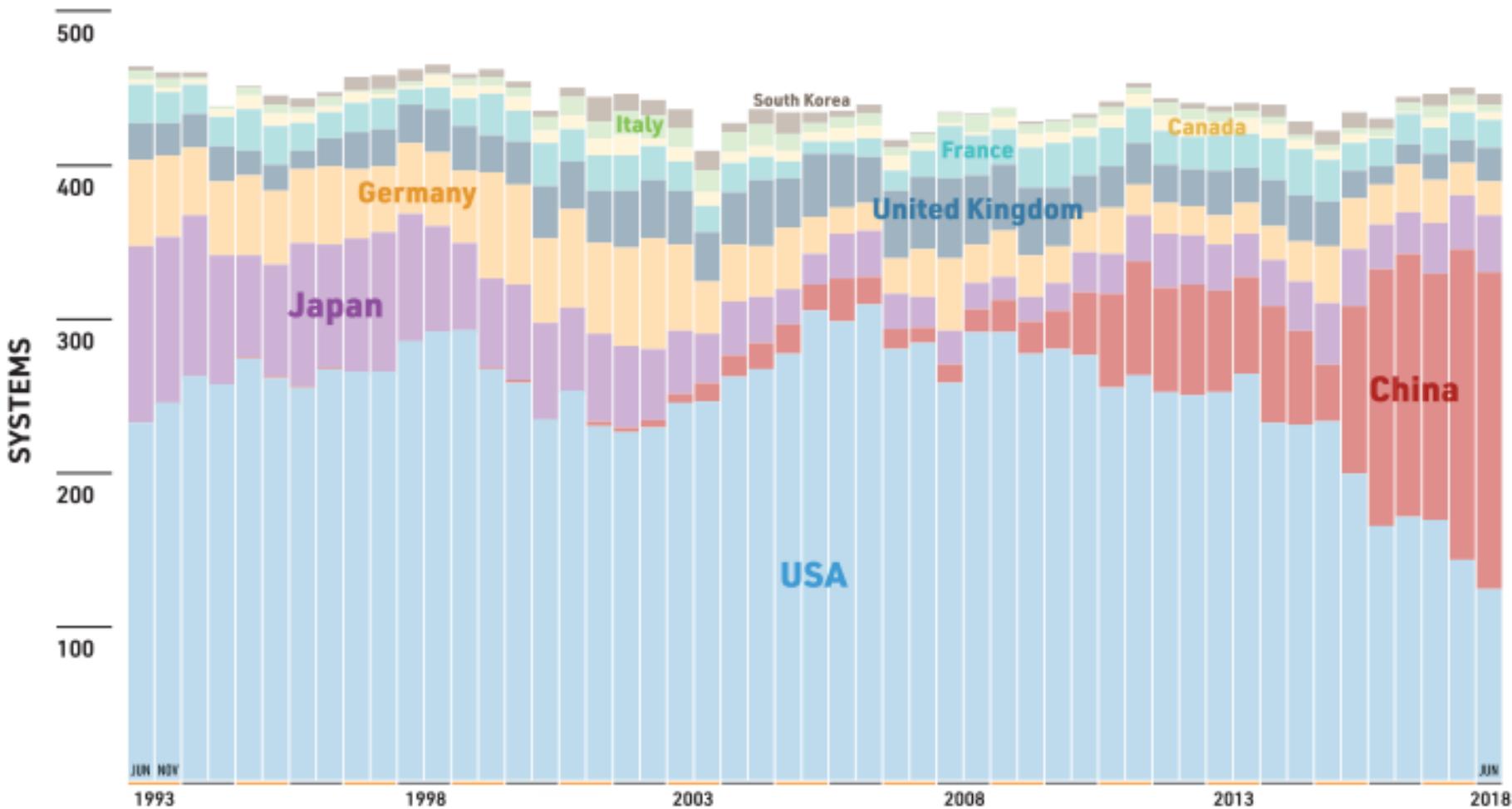


Andy Grove
and the Rise of the World's
Most Powerful Chip Company

TIM JACKSON

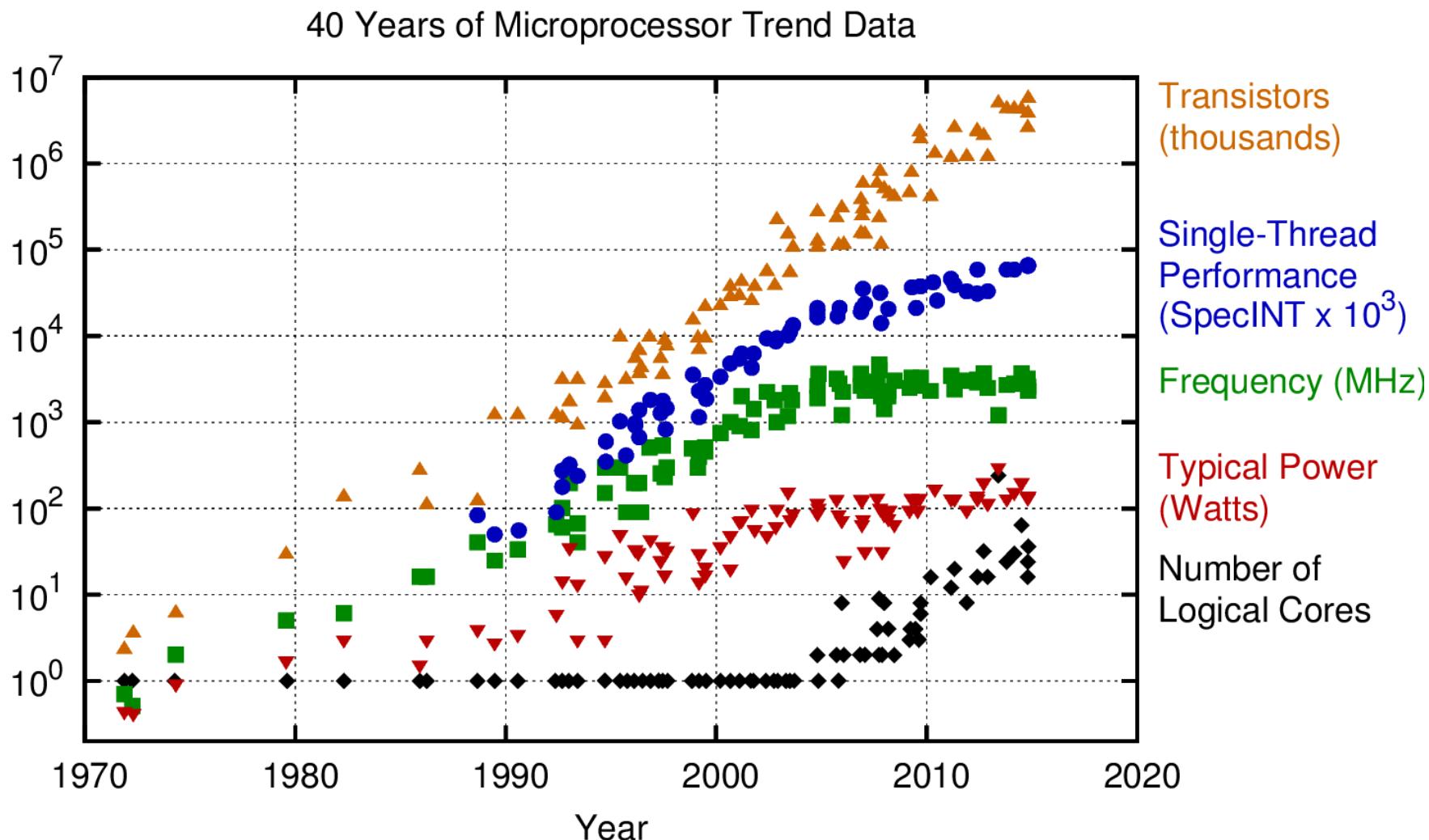
Most Powerful Computers – where?

COUNTRIES



source: www.top500.org

Trends – Moore's Law



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2015 by K. Rupp

Image source: <https://www.karlrupp.net/wp-content/uploads/2015/06/40-years-processor-trend.png>

FLOP/s as a metric

High Performance Computing (HPC) units are:

Flop: floating point operation, usually double precision unless noted

Flop/s: floating point operations per second

Bytes: size of data (a double precision floating point number is 8)

Typical sizes are millions, billions, trillions...

Mega	$Mflop/s = 10^6 \text{ flop/sec}$	$Mbyte = 2^{20} = 1048576 \sim 10^6 \text{ bytes}$
Giga	$Gflop/s = 10^9 \text{ flop/sec}$	$Gbyte = 2^{30} \sim 10^9 \text{ bytes}$
Tera	$Tflop/s = 10^{12} \text{ flop/sec}$	$Tbyte = 2^{40} \sim 10^{12} \text{ bytes}$
Peta	$Pflop/s = 10^{15} \text{ flop/sec}$	$Pbyte = 2^{50} \sim 10^{15} \text{ bytes}$
Exa	$Eflop/s = 10^{18} \text{ flop/sec}$	$Ebyte = 2^{60} \sim 10^{18} \text{ bytes}$
Zetta	$Zflop/s = 10^{21} \text{ flop/sec}$	$Zbyte = 2^{70} \sim 10^{21} \text{ bytes}$
Yotta	$Yflop/s = 10^{24} \text{ flop/sec}$	$Ybyte = 2^{80} \sim 10^{24} \text{ bytes}$

Current fastest (public) machine ~ 442 Pflop/s

Up-to-date list at www.top500.org

The Top500 List

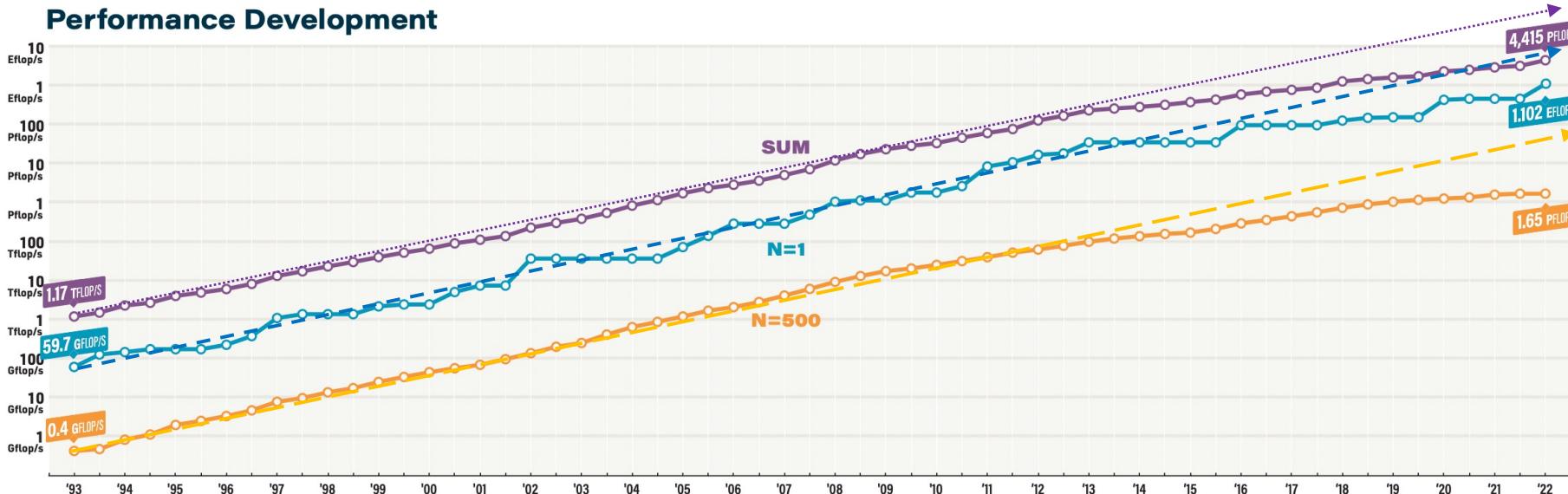
- Listing the 500 most powerful computers in the world
- Yardstick: Rmax of Linpack
 - Solve $Ax=b$, **dense** problem, matrix is random
 - Dominated by **dense** matrix-matrix multiply
- Update twice a year:
 - ISC'xy in June in Germany
 - SCxy in November in the U.S.
- All information available from the TOP500 web site at:
www.top500.org
- Green500: **<https://www.top500.org/green500/>** (most energy efficient)
- HPCG: **<https://www.top500.org/hpcg/>** (instead of Linpack, uses Conjugate Gradient algorithm for solving sparse linear systems)

Most Powerful Computers



JUNE 2022	SYSTEM	SPECS	SITE	COUNTRY	CORES	R _{MAX} PFLOP/S	POWER MW
1	Frontier	HPE Cray EX235a, AMD Opt 3rd Gen EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-10	DOE/SC/ORNL	USA	8,730,112	1,102.0	21.3
2	Fugaku	Fujitsu A64FX (48C, 2.2GHz), Tofu Interconnect D	RIKEN R-CCS	Japan	7,630,848	442.0	29.9
3	LUMI	HPE Cray EX235a, AMD Opt 3rd Gen EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-10	EuroHPC/CSC	Finland	1,268,736	151.9	2.94
4	Summit	IBM POWER9 (22C, 3.07GHz), NVIDIA Volta GV100 (80C), Dual-Rail Mellanox EDR Infiniband	DOE/SC/ORNL	USA	2,414,592	148.6	10.1
5	Sierra	IBM POWER9 (22C, 3.1GHz), NVIDIA Tesla V100 (80C), Dual-Rail Mellanox EDR Infiniband	DOE/NNSA/LLNL	USA	1,572,480	94.6	7.44

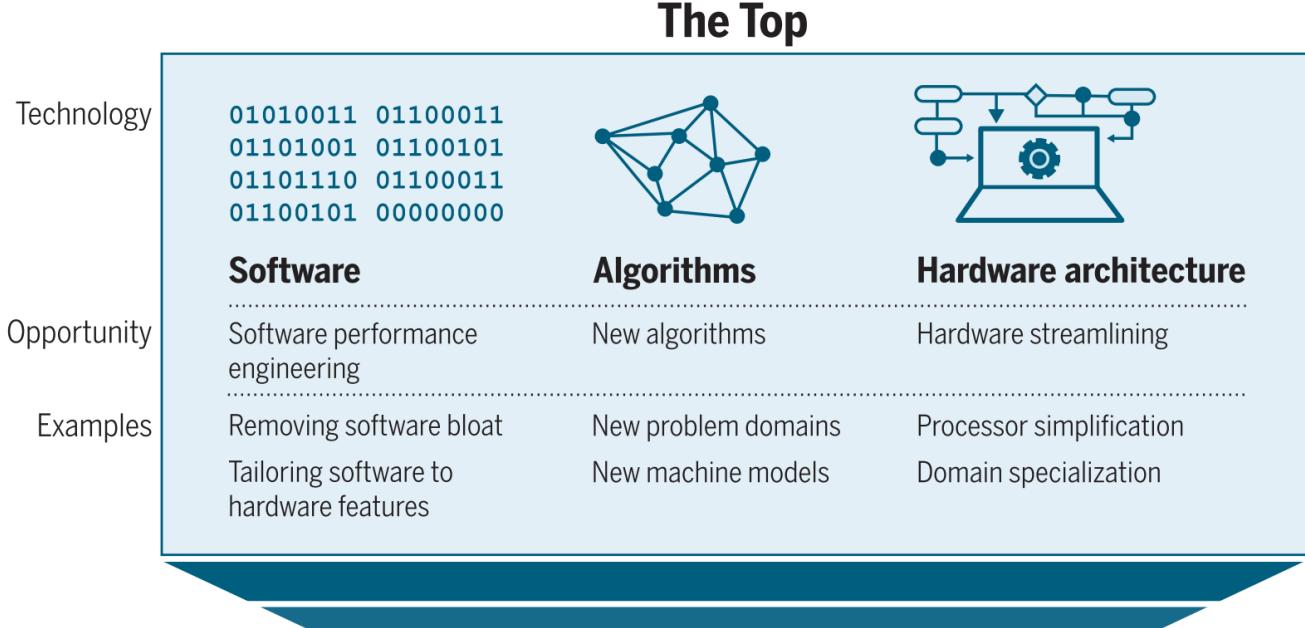
Performance Development



source: www.top500.org

Moore's law might be already failing!

But there is more room for improvement in the “post-Moore era”¹:



Performance gains after Moore's law ends. In the post-Moore era, improvements in computing power will increasingly come from technologies at the “Top” of the computing stack, not from those at the “Bottom”, reversing the historical trend.

¹ Leiserson, C. E., Thompson, N. C., Emer, J. S., Kuszmaul, B. C., Lampson, B. W., Sanchez, D., & Schardl, T. B. (2020). There's plenty of room at the Top: What will drive computer performance after Moore's law?. *Science*, 368(6495).

Our Coverage

■ IA32

- The traditional x86

■ x86-64

- The standard
- \$gcc hello.c
- \$gcc -m64 hello.c

■ Presentation

- 3rd edition covers x86-64
- 2nd International Edition covers IA32 + x86-64
- 2nd edition covers only IA32
- We will only cover x86-64

Thank you!