# Machine-Level Programming I: Basics

## - History of Processors -

CENG331 - Computer Organization

Middle East Technical University

Instructors:

Murat Manguoglu (Sections 1-2)

Fall 2020

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

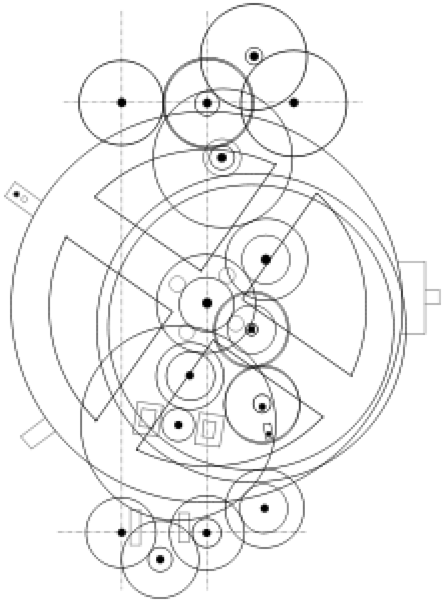
**Why worry about old ideas?**

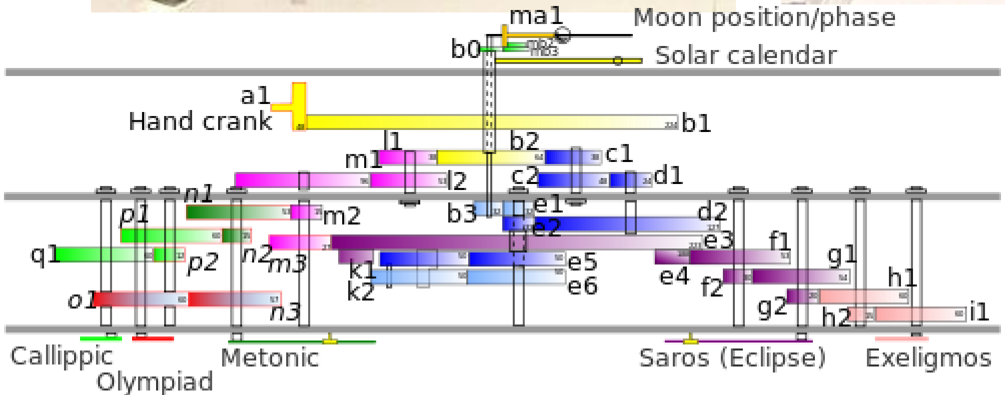
- *Die Geschichte der Wissenschaft is die Wissenschaft selbst* (The history of science is science itself) – Johann Wolfganf von Goethe 1749-1832
  - In fact, if you read any scientific paper/thesis/work, you will notice it starts with the review of the literature (i.e. History) of the earlier work
- 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

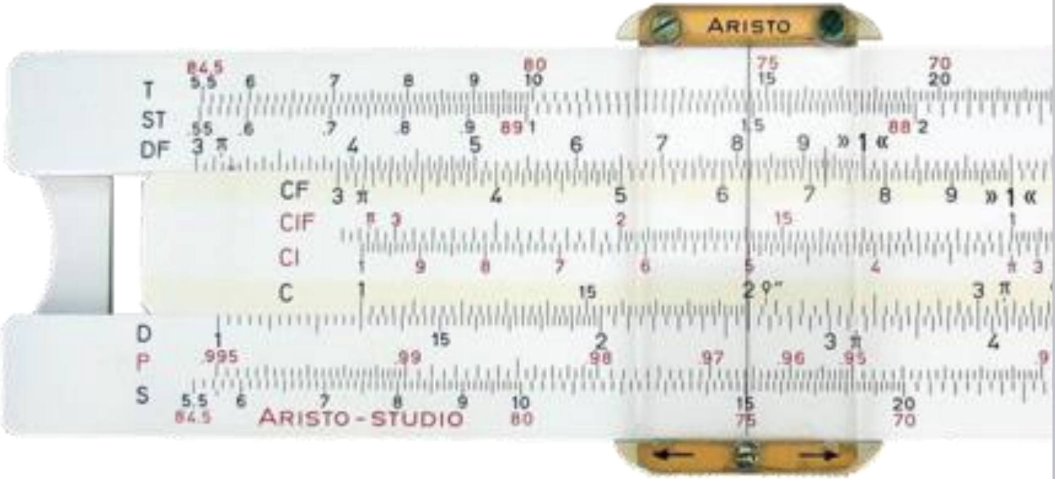
**An astronomical calendar capable of tracking the position of the sun, moon, and planets; predict eclipses, and even recreate the irregular orbit of the moon**

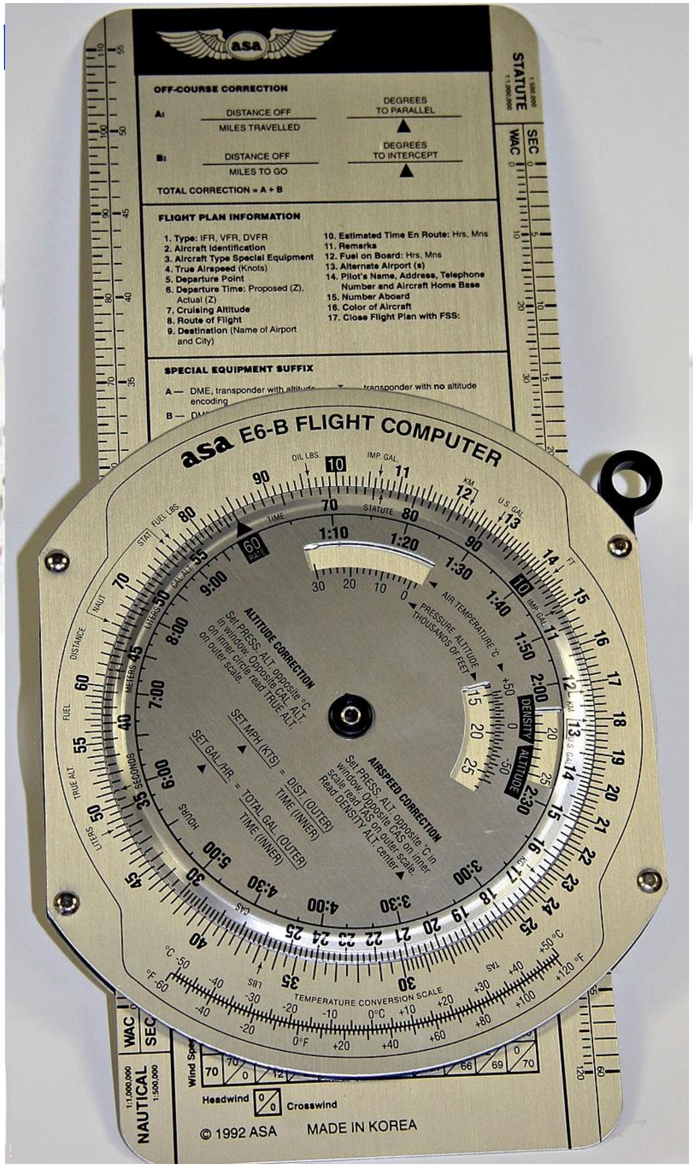




**Source:http://en.wikipedia.org/wiki/ Antikythera_mechanism**

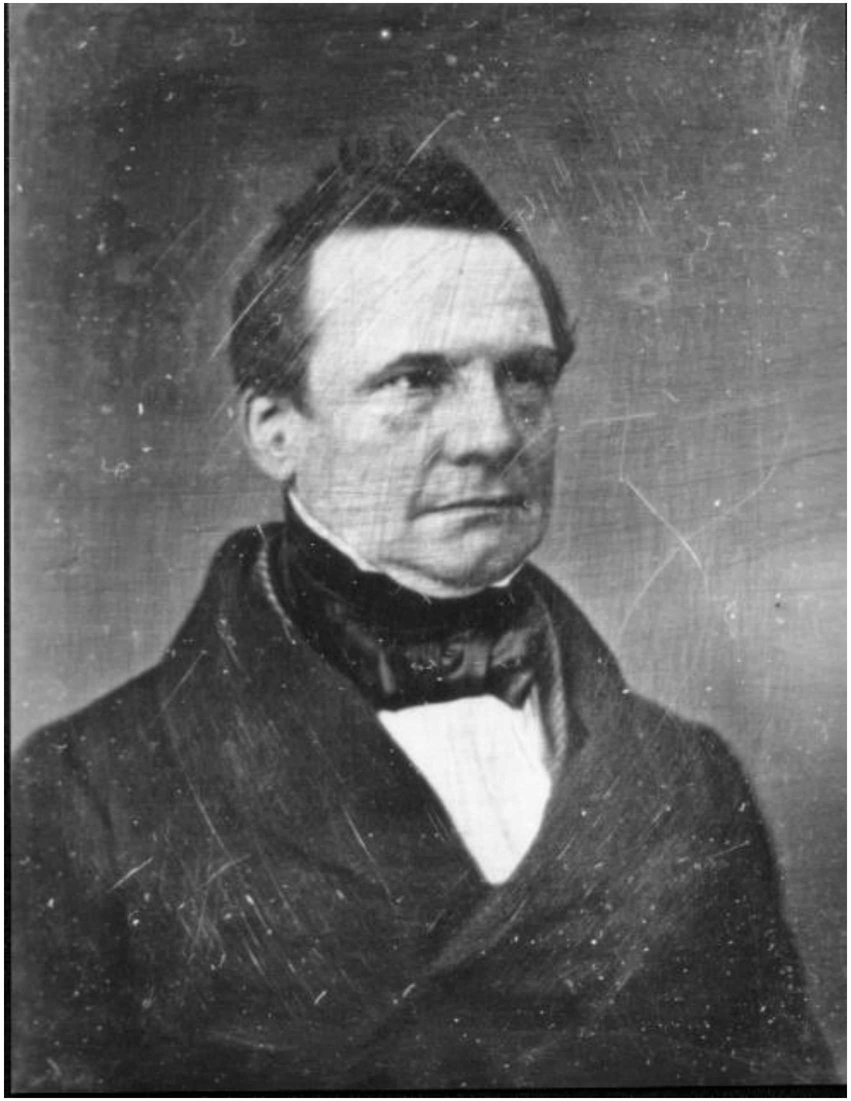
# Slide ruler
# ~1620 and still used today (but only as a backup computer on ai





# Charles Babbage 1791-1871

**Lucasian Professor of Mathematics, Cambridge University, 1827-1839**



# Charles Babbage

- ***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

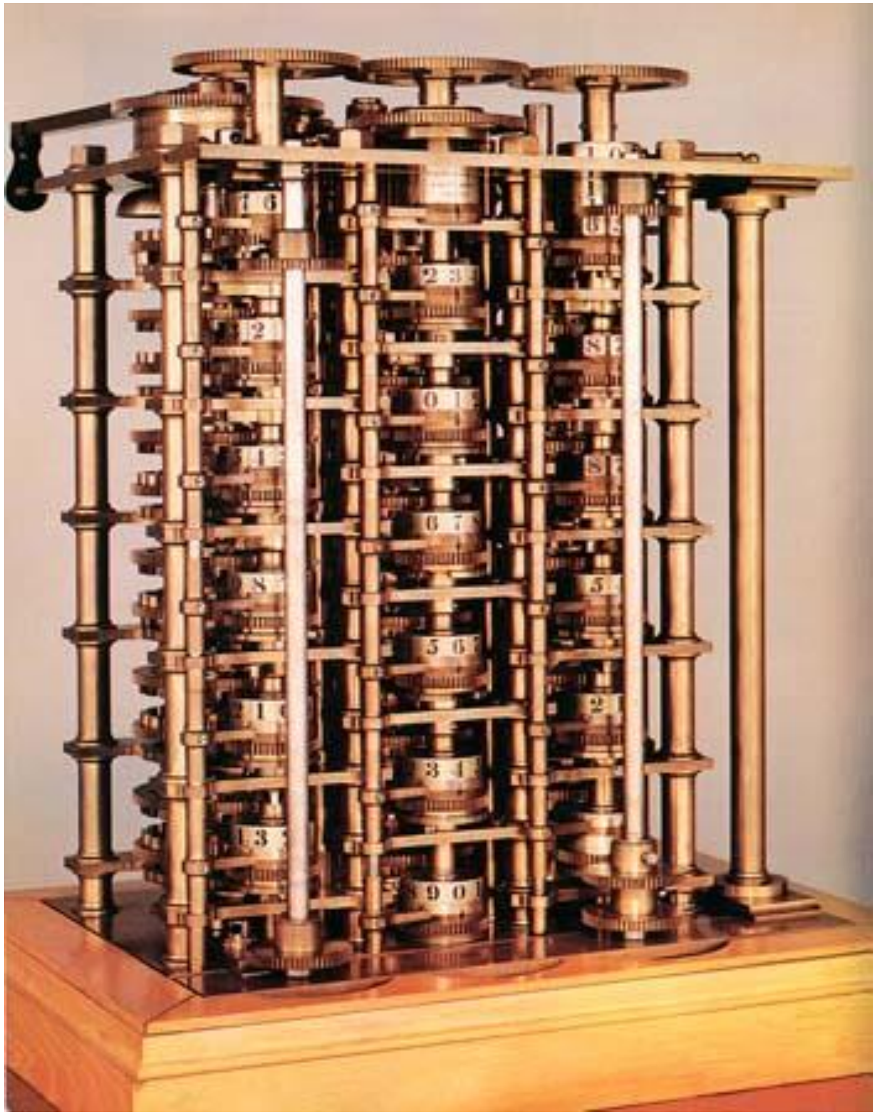
- The paper is read by Scheutz & his son in Sweden

## 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 publishe

- *conceived during a hiatus in the deve difference engine*

## Inspiration: *Jacquard Loom Machin*

- looms were controlled by punched c
  - The set of cards with fixed punc pattern of weave ⇒ *program*
  - The same set of cards could be u colored threads ⇒ *numbers*

## 1871: Babbage dies

- The machine remains unrealized.

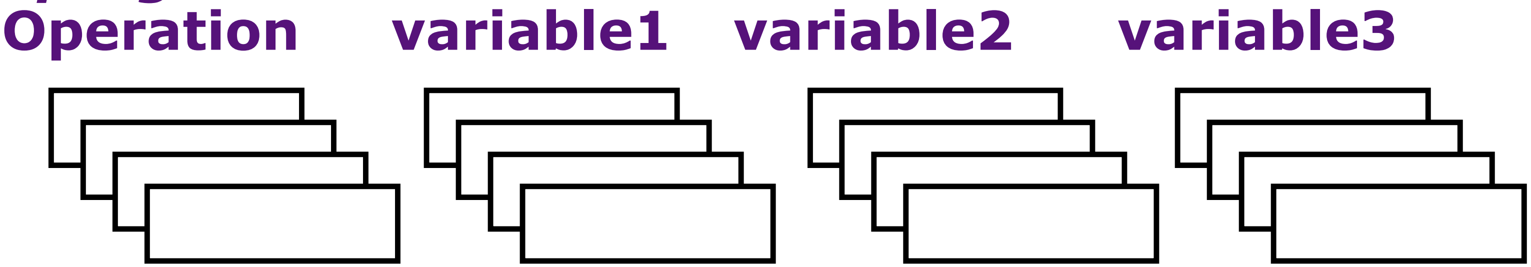
***It is not clear if the analytic engine could be built using the mechanical technology of the time***

# Analytic Engine

The first conception of a general-purpose computer

1. 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.
2. The *mill (arithmetic logic unit)* into which the quantities about to be operated upon are always brought.

The *program*



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*....”

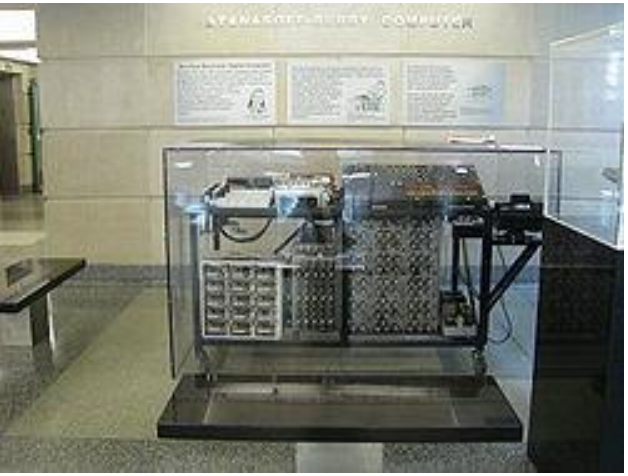
- **In the early twentieth century - the focus shifted to analog computers but**
  - *Harvard Mark I built in 1944 is very close in spirit to the Analytic Engine.*

# 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*

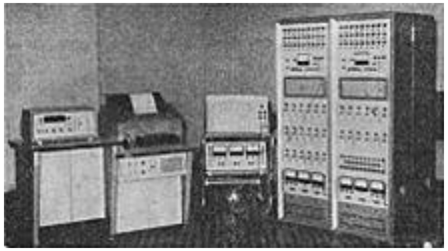
## *Technology:*

- Tubes and Electromechanical relays

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

# Electronic analog computers

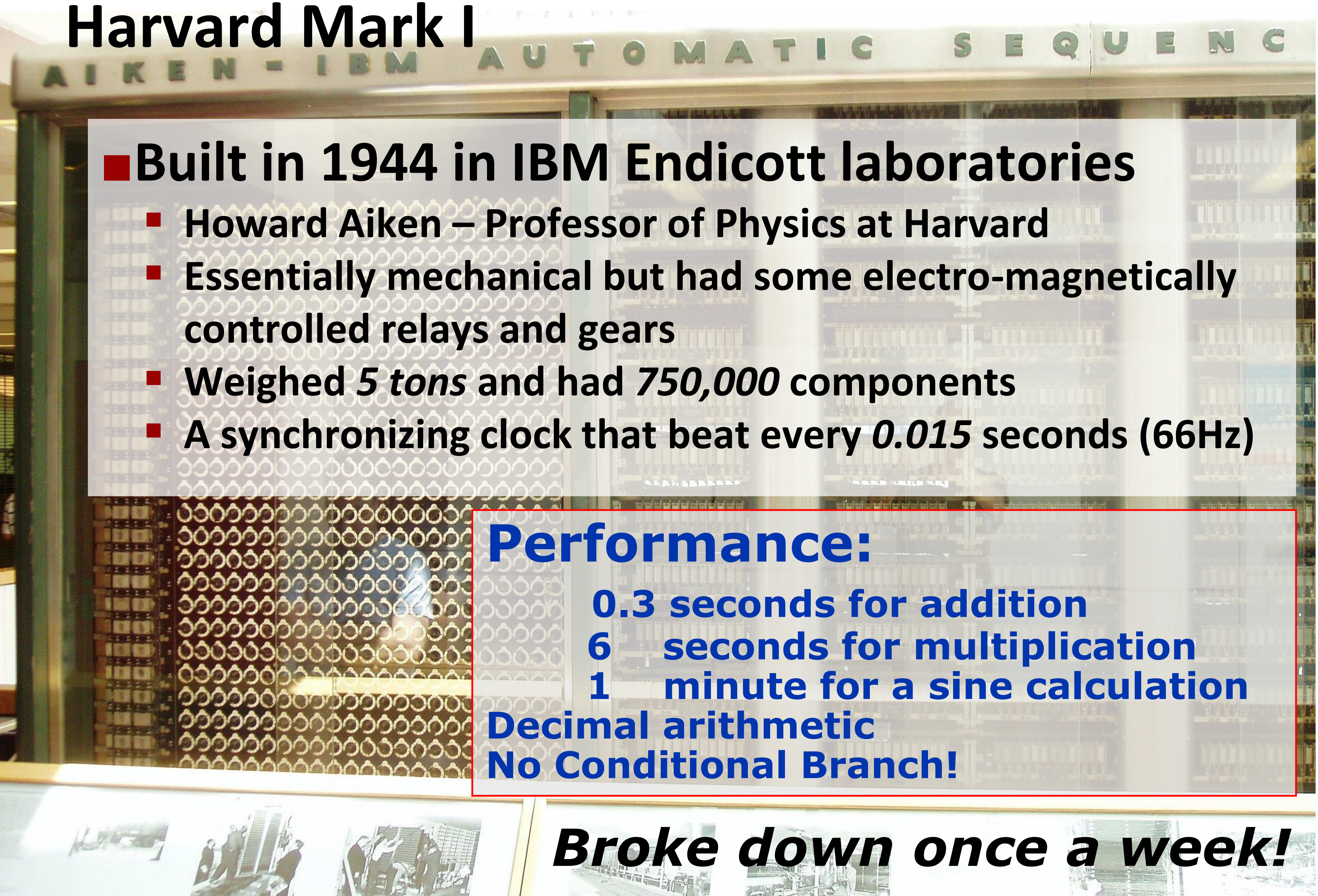
1960 Newmark analogue computer, made up of five units. This computer was used to solve differential equations



ELWAT , Poland, 1967

AKAT-1 , Poland, 1959

# 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)

**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, ... )**

# Electronic Discrete Variable Automatic Computer (EDVAC)

- **ENIAC's programming system was external**
  - Sequences of instructions were executed independently of the results of the calculation
  - Human intervention required to take instructions "out of order"
- **Eckert, Mauchly, John von Neumann and others designed EDVAC (1944) to solve this problem**
  - Solution was the *stored program computer*

    ⇒ *"program can be manipulated as data"*
- ***First Draft of a report on EDVAC* was published in 1945, but just had von Neumann's signature!**
  - In 1973 the court of Minneapolis attributed the honor of *inventing the computer* to John Atanasoff

# 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**

# Technology Issues

**ENIAC** ⇒ **EDVAC**

**18,000 tubes** **4,000 tubes**

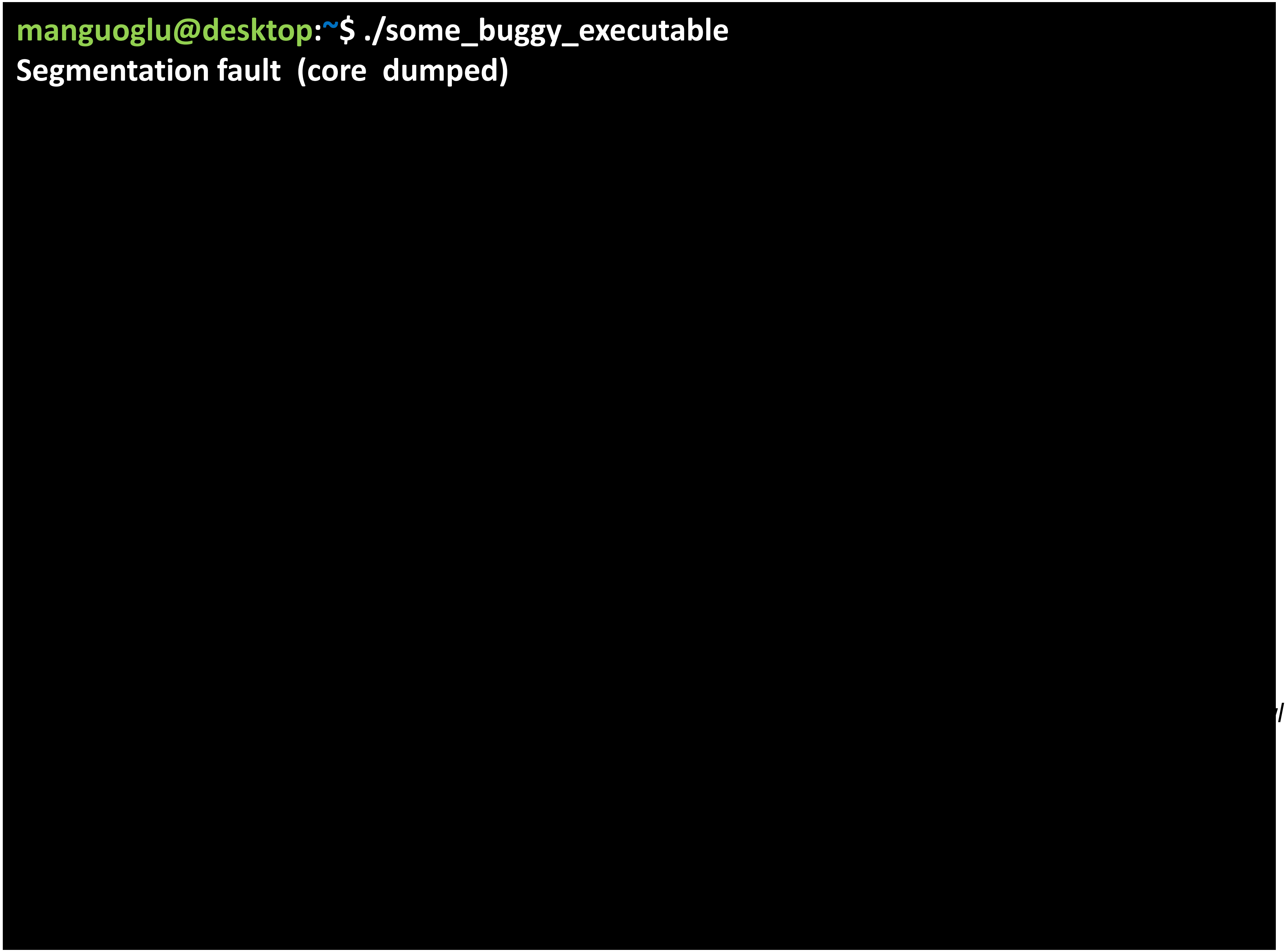
**20 10-digit numbers** **2000 word storage**

**mercury delay lines**

***ENIAC had many asynchronous parallel units but only one was active at a time***

**BINAC : Two processors that checked each other for reliability.**

***Didn't work well because processors never agreed***



```
manguoglu@desktop:~$ ./some_buggy_executable
Segmentation fault  (core  dumped)
```

# Commercial Activity: 1948-52

## IBM’s SSEC (follow on from Harvard Mark I)

*Selective Sequence Electronic Calculator*

- 150 word store.
- Instructions, constraints, and tables of data were read from paper tapes.
- 66 Tape reading stations!
- Tapes could be glued together to form a loop!
- Data could be output in one phase of computation and read in the next phase of computation.

# And then there was IBM 701

**IBM 701 -- 30 machines were sold in 1953-54**
**used CRTs as main memory, 72 tubes of 32x32b each**

**IBM 650 -- a cheaper, drum based machine,**
**more than 120 were sold in 1954**
**and there were orders for 750 more!**

***Users stopped building their own machines.***

**Why was IBM late getting into computer technology?**

***IBM was making too much money!***
**Even without computers, IBM revenues were doubling every 4 to 5 years in 40’s and 50’s.**

# Computers in mid 50's

- **Hardware was expensive**
- **Stores were small (1000 words)**

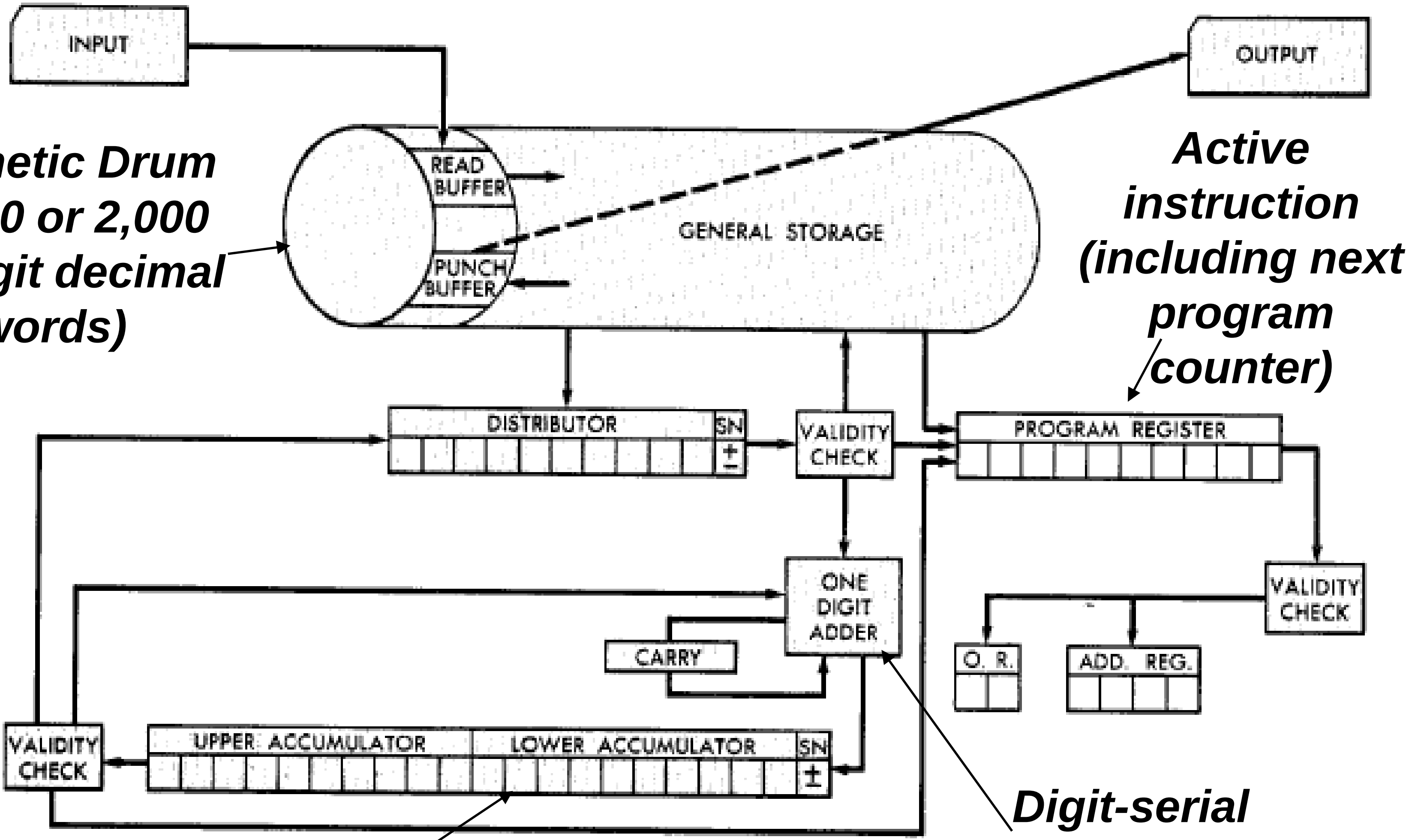
  ⇒ No resident system software!

- **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)



***Magnetic Drum (1,000 or 2,000 10-digit decimal words)***

***Active instruction (including next program counter)***

***Digit-serial ALU***

***20-digit accumulat or***

***[From 650 Manual, © IBM]***

# 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!***

# Evolution of Addressing Modes

1. Single accumulator, absolute address

   LOAD x

2. Single accumulator, index registers

   LOAD x, IX

3. Indirection

   LOAD (x)

4. Multiple accumulators, index registers, indirection

   LOAD R, IX, x

   or LOAD R, IX, (x) the meaning?

   R ← M[M[x] + (IX)]

   or R ← M[M[x + (IX)]]

5. Indirect through registers

   LOAD $R_I$, $(R_J)$

6. The works

   LOAD $R_I$, $R_J$, $(R_K)$ $R_J$ = index, $R_K$ = base addr

# Variety of Instruction Formats

- ***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 × Reg) to Reg $R_I \leftarrow (R_I) + (R_J)$

  (Reg × Mem) to Reg $R_I \leftarrow (R_I) + M[x]$

  - *x* can be specified directly or via a register
  - effective address calculation for *x* could include indexing, indirection, ...

- ***Three address formats:* One destination and up to two operand sources per instruction**

  (Reg x Reg) to Reg $R_I \leftarrow (R_J) + (R_K)$

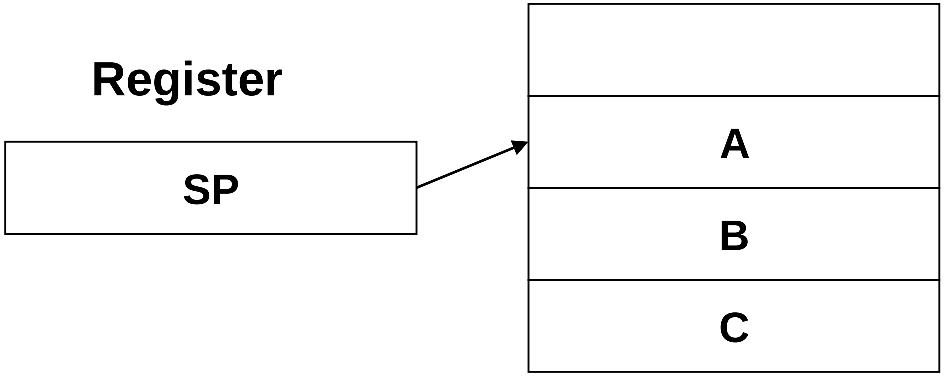
  (Reg x Mem) to Reg $R_I \leftarrow (R_J) + M[x]$

# Zero Address Formats

- **Operands on a stack**

  | | |
  |---|---|
  | add | M[sp-1] ← M[sp] + M[sp-1] |
  | load | M[sp] ← M[M[sp]] |

  - Stack can be in registers or in memory (usually top of stack cached in registers)



  - Example: Burrough Corporation B5000

# 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

# Software Developments

**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**

## Libraries of numerical routines

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

**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, ...**

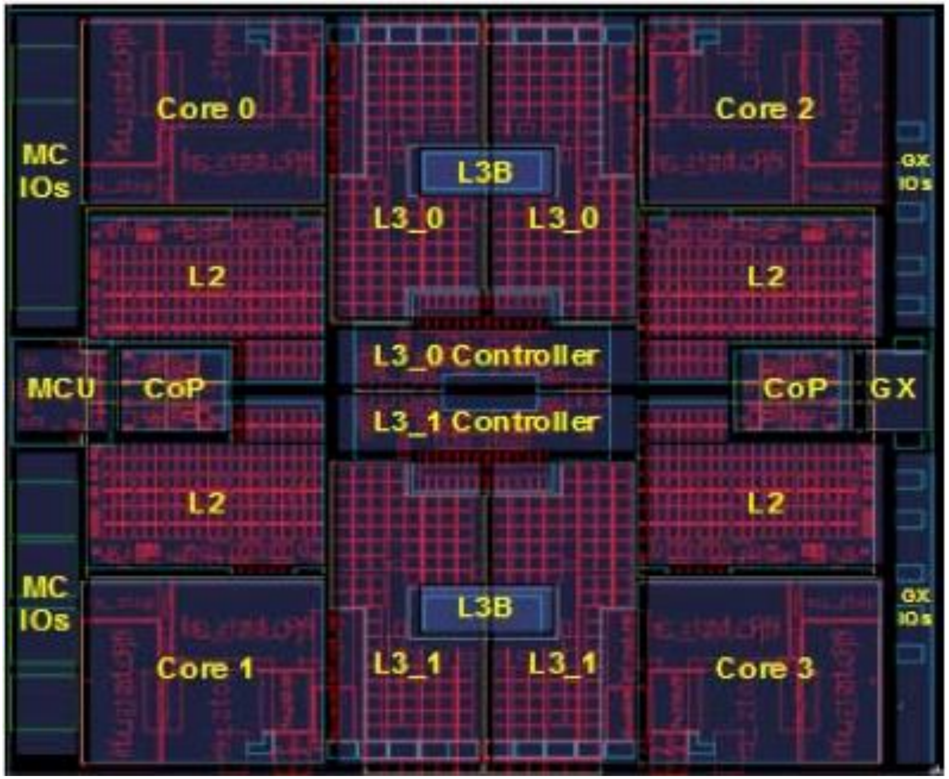
⇒ ***IBM 360***

# 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!***

# 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**

# Intel x86 Processors

- **Dominate laptop/desktop/server market**

- **Evolutionary design**
  - 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 has done 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
    - Leads the world in semiconductor technology
  - AMD has fallen behind
    - Relies on external semiconductor manufacturer

# 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 that bad, 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!
- **All but low-end x86 processors support x86-64**
  - But, lots of code still runs in 32-bit mode

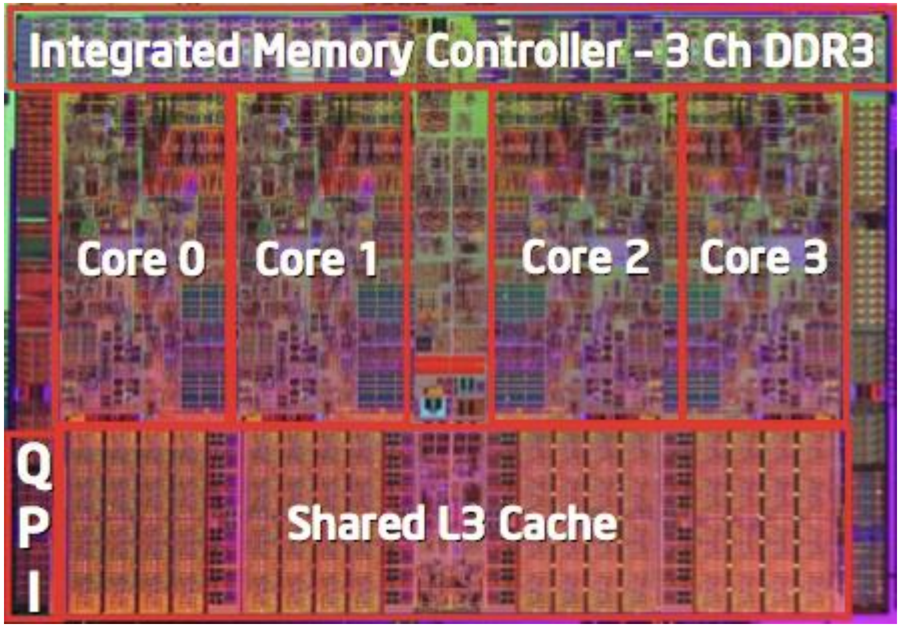
# x86 Evolution: Milestones

<table>
<tr><th><em>Name</em></th><th><em>Date</em></th><th><em>Transistors</em></th><th><em>MHz</em></th></tr>
<tr><td><strong>8086</strong></td><td><strong>1978</strong></td><td><strong>29K</strong></td><td><strong>5-10</strong></td></tr>
<tr><td colspan="4">▪ First 16-bit Intel processor.  Basis for IBM PC &amp; DOS<br>▪ 1MB address space</td></tr>
<tr><td><strong>386</strong></td><td><strong>1985</strong></td><td><strong>275K</strong></td><td><strong>16-33</strong></td></tr>
<tr><td colspan="4">▪ First 32 bit Intel processor , referred to as IA32<br>▪ Added “flat addressing”, capable of running Unix</td></tr>
<tr><td><strong>Pentium 4E</strong></td><td><strong>2004</strong></td><td><strong>125M</strong></td><td><strong>2800-3800</strong></td></tr>
<tr><td colspan="4">▪ First 64-bit Intel x86 processor, referred to as x86-64</td></tr>
<tr><td><strong>Core 2</strong></td><td><strong>2006</strong></td><td><strong>291M</strong></td><td><strong>1060-3500</strong></td></tr>
<tr><td colspan="4">▪ First multi-core Intel processor</td></tr>
<tr><td><strong>Core i7</strong></td><td><strong>2008</strong></td><td><strong>731M</strong></td><td><strong>1700-3900</strong></td></tr>
<tr><td colspan="4">▪ Four cores</td></tr>
</table>

# 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 |
| Core i7 Skylake | 2015 | 1.9B |
| 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 – Desktop state-of-the-art

- **Core i9-9980XE**
  - 18 cores
  - 36 threads
  - 3.0-4.4 Ghz
  - 24.75 MB Cache

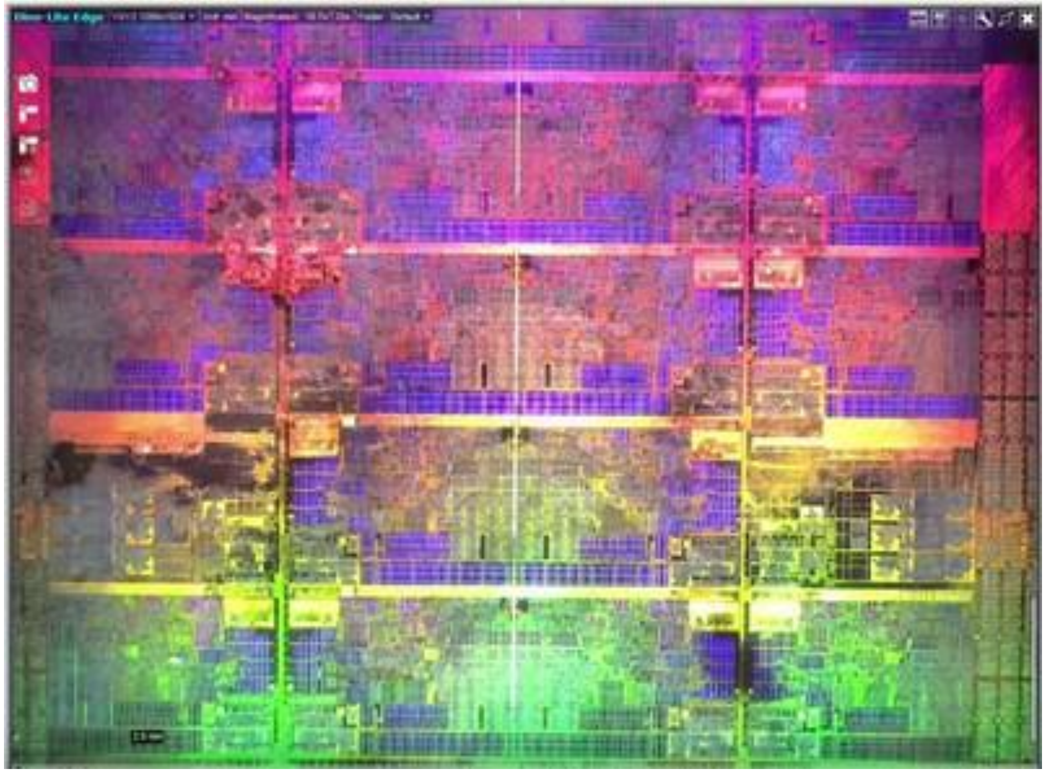


Image source: https://hwbot.org/newsflash/5002_video_intel_core_i9_7980xe_die_extraction_with_de8auer_(part_2)

# AMD –Server state-of-the-art

- **AMD epyc 7742**
  - 64 cores
  - 128 threads
  - 256 MB Cache
  - 2.25-3.4 GHz

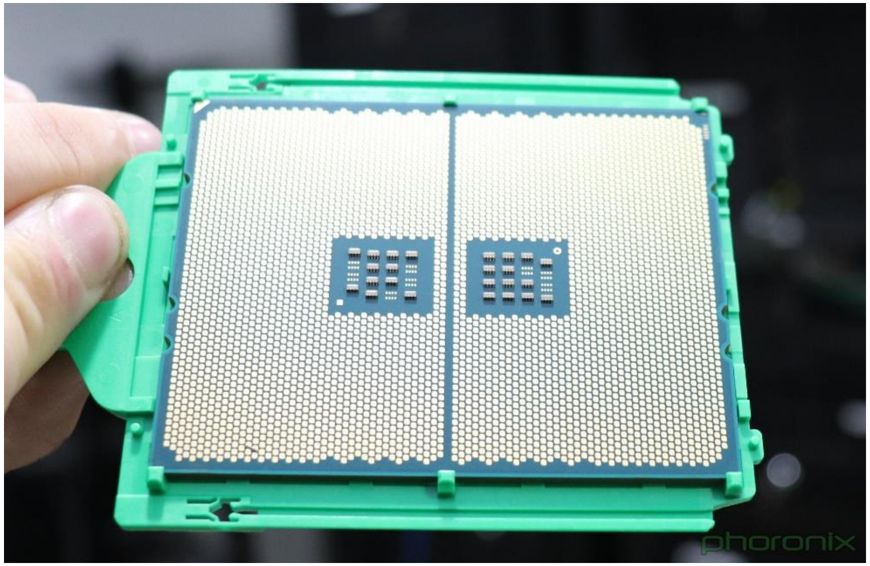
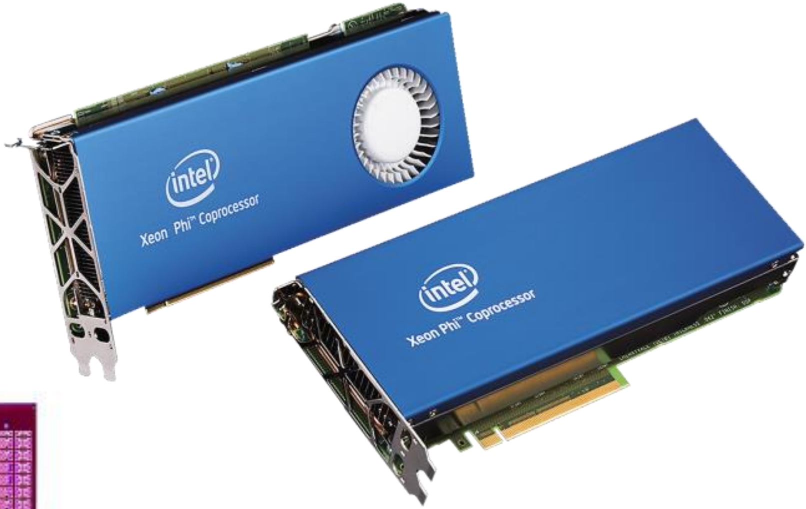


**Image source: https://www.phoronix.com/image-viewer.php?id=amd-epyc-7502-7742&image=amd_rome_4_lrg**

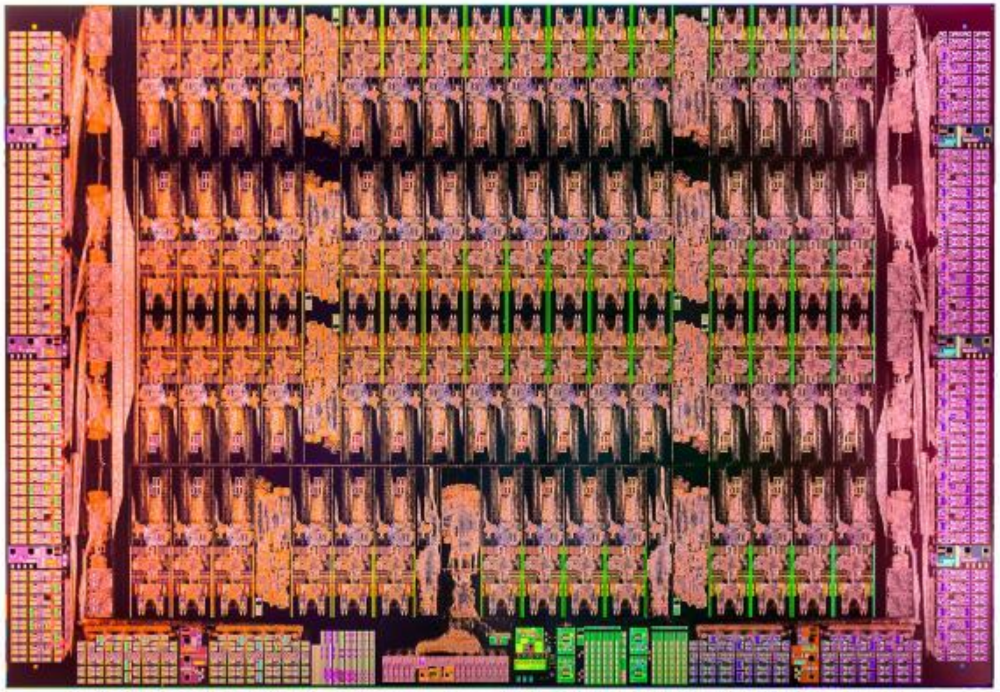
# Manycore (vs Multicore) processors

http://www.intel.com/content/www/us/en/processors/xeon/xeon-phi-detail.html

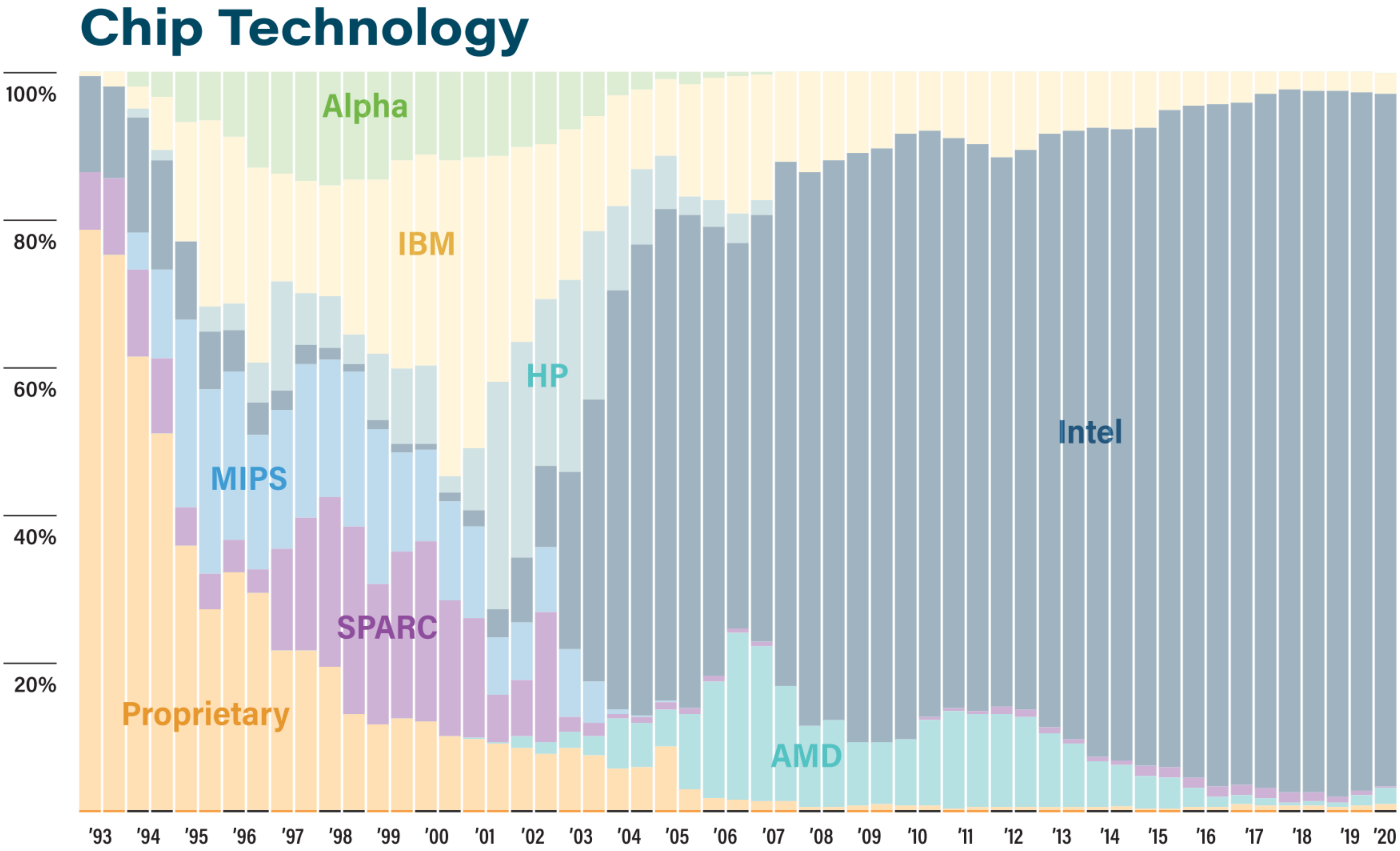


Upto 72 cores

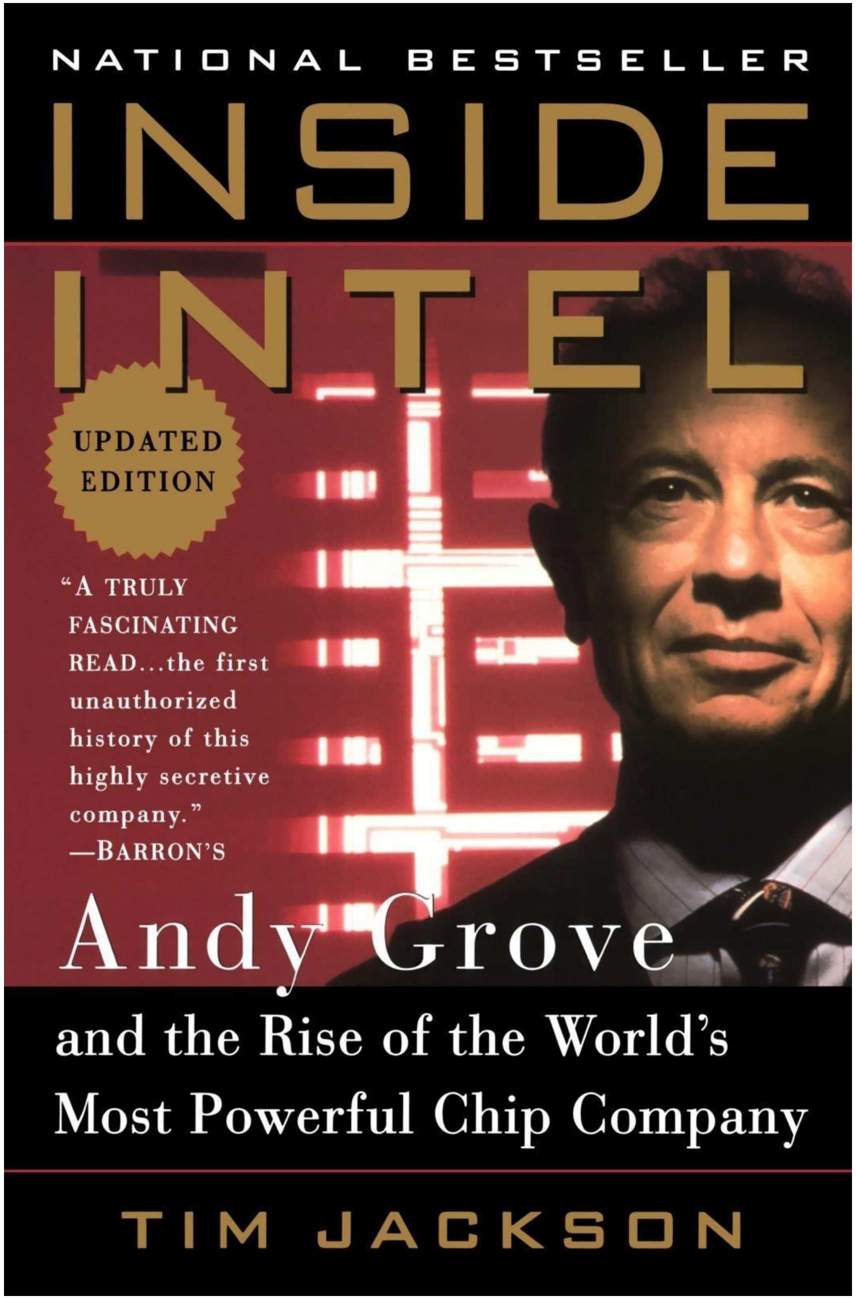
(XeonPhi co-processors are no longer produced)



# Most Powerful Computers - which processor ?



source: www.top500.org



NATIONAL BESTSELLER

# INSIDE INTEL

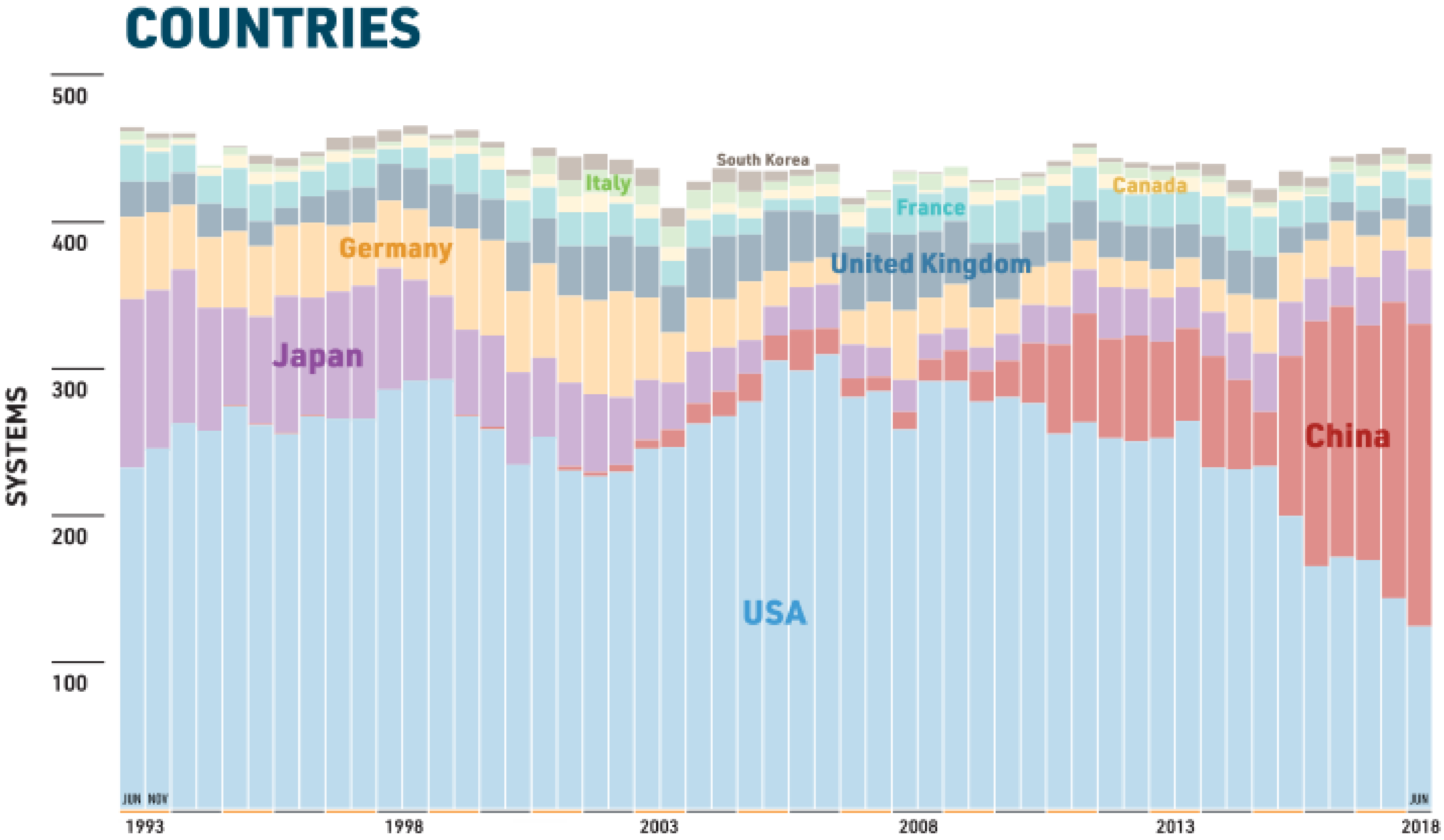
UPDATED EDITION

"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?



source: www.top500.org

# 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!*