Computer System

Virendra Singh
Associate Professor
Computer Architecture and Dependable Systems Lab
Department of Electrical Engineering
Indian Institute of Technology Bombay
http://www.ee.iitb.ac.in/~viren/
E-mail: viren@ee.iitb.ac.in

CP-226:Computer Architecture

About This Course

- Course Textbook
  - D.A. Patterson and J.L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, 4th edition, Elsevier/Morgan Kauffman.
  - 3rd edition OK if 4th edition not available.
- Homework
  - Couple of homework assignments, unequally weighted
- Tests
  - Periodic tests will be conducted (some are scheduled and some surprise)

About This Course

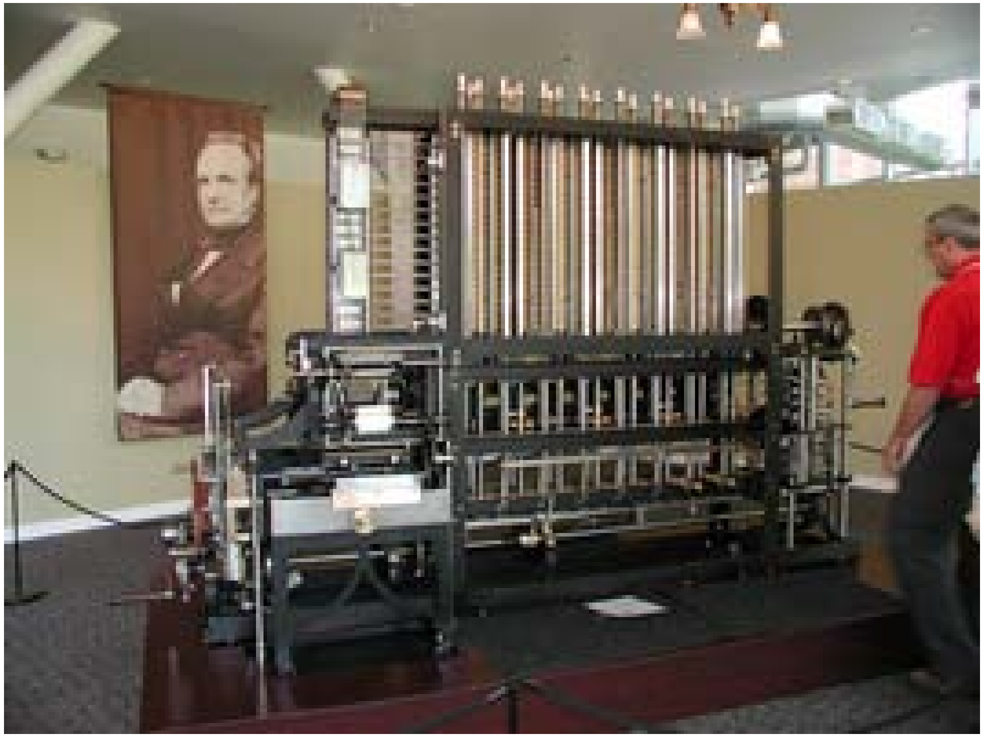
- Project
  - Implement processor for MNIT-CS13 ISA
  - Priority: working nonpipelined version
  - Bonus: pipelined version
  - Groups of 3 students
    - Form teams early
  - Must demo and submit written report

Historic Events

- 1623, 1642: Wilhelm Strickland/Blaise Pascal built a mechanical counter with carry.
- 1823-34: Charles Babbage designed difference engine. http://www.youtube.com/watch v=0anIyVGeWOI&feature=related

Babbage's Difference Engine

- Babbage Difference Engine
  - ➢ Hand-cranked mechanical computer.
  - ➢ Computed polynomial functions.
  - ➢ Designed by Charles Babbage in the early to mid 1800s.
    - ✧ Arguably the world's first computer scientist, lived 1791-1871.
  - ➢ He wasn't able to build it because he lost his funding.



- His plans survived and this working model was built.
  - Includes a working printer!

http://www.computerhistory.org/babbage/

Historic Events

- 1943-44: John Mauchly (professor) and J. Presper Eckert (graduate student) built ENIAC at U. Pennsylvania.
- 1944: Howard Aiken used "separate data and program memories" in MARK I – IV computers – *Harvard Architecture*.
- 1945-52: John von Neumann proposed a "*stored program computer*" EDVAC (Electronic Discrete Variable Automatic Computer) – *Von Neumann Architecture* – use the same memory for program and data.

Most Influential Document

- "Preliminary Discussion of the Logical Design of an Electronic Computing Instrument," 1946 report by A. W. Burks, H. H. Holdstine and J. von Neumann. Appears in *Papers of John von Neumann*, W. Aspray and A. Burks (editors), MIT Press, Cambridge, Mass., 1987, pp. 97-146.

Theory of Computing

- Alan Turing (1912-1954) gave a model of computing in 1936 – *Turing Machine*.
- Original paper: A. M. Turing, "On Computable Numbers with an Application to the *Entscheidungsproblem**," *Proc. Royal Math. Soc.*, ser. 2, vol. 42, pp. 230-265, 1936.
- Recent book: David Leavitt, *The Man Who Knew Too Much: Alan Turing and the Invention of the Computer (Great Discoveries),* W. W. Norton & Co., 2005.

* *The question of decidability, posed by mathematician Hilbert.*

History Continues

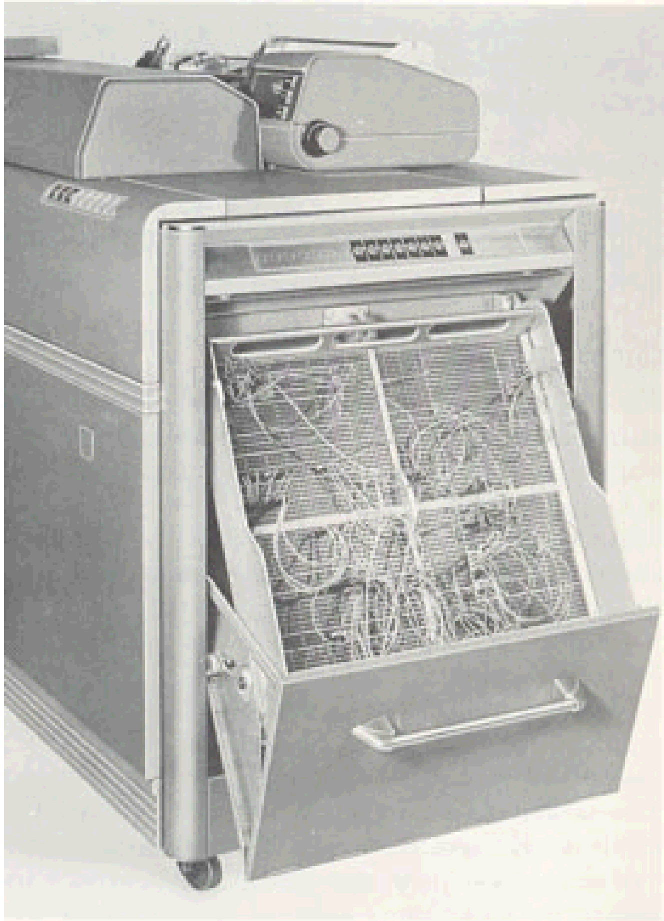
- 1946-52: Von Neumann built the IAS computer at the Institute of Advanced Studies, Princeton *– A prototype for most future computers*.
- 1947-50: Eckert-Mauchly Computer Corp. built UNIVAC I (Universal Automatic Computer), used in the 1950 census.
- 1949: Maurice Wilkes built EDSAC (Electronic Delay Storage Automatic Calculator), the first stored-program computer.

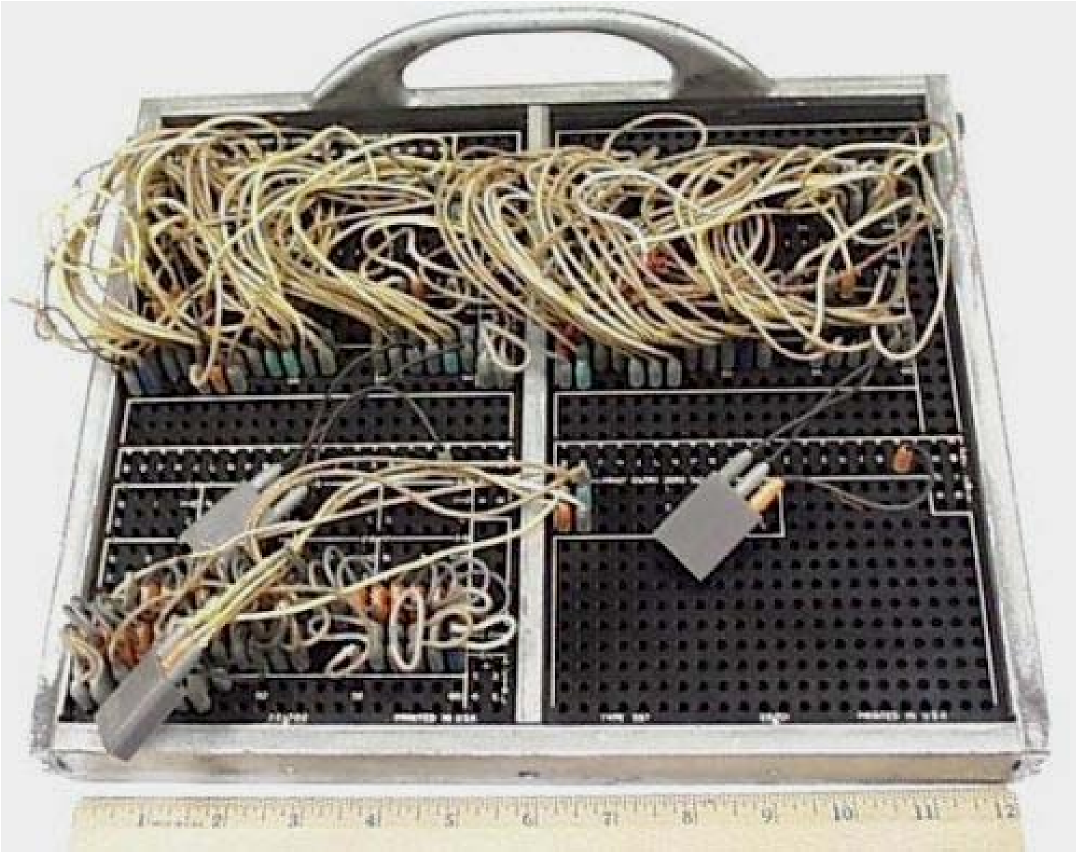
What was Computing Like?

- A data processing application involved passing decks of punched cards through electromechanical "unit record" machines.

- Repetitive sort, calculate, collate, and tabulate operations ...
  - ... were programmed with hand-wired plugboard control panels.

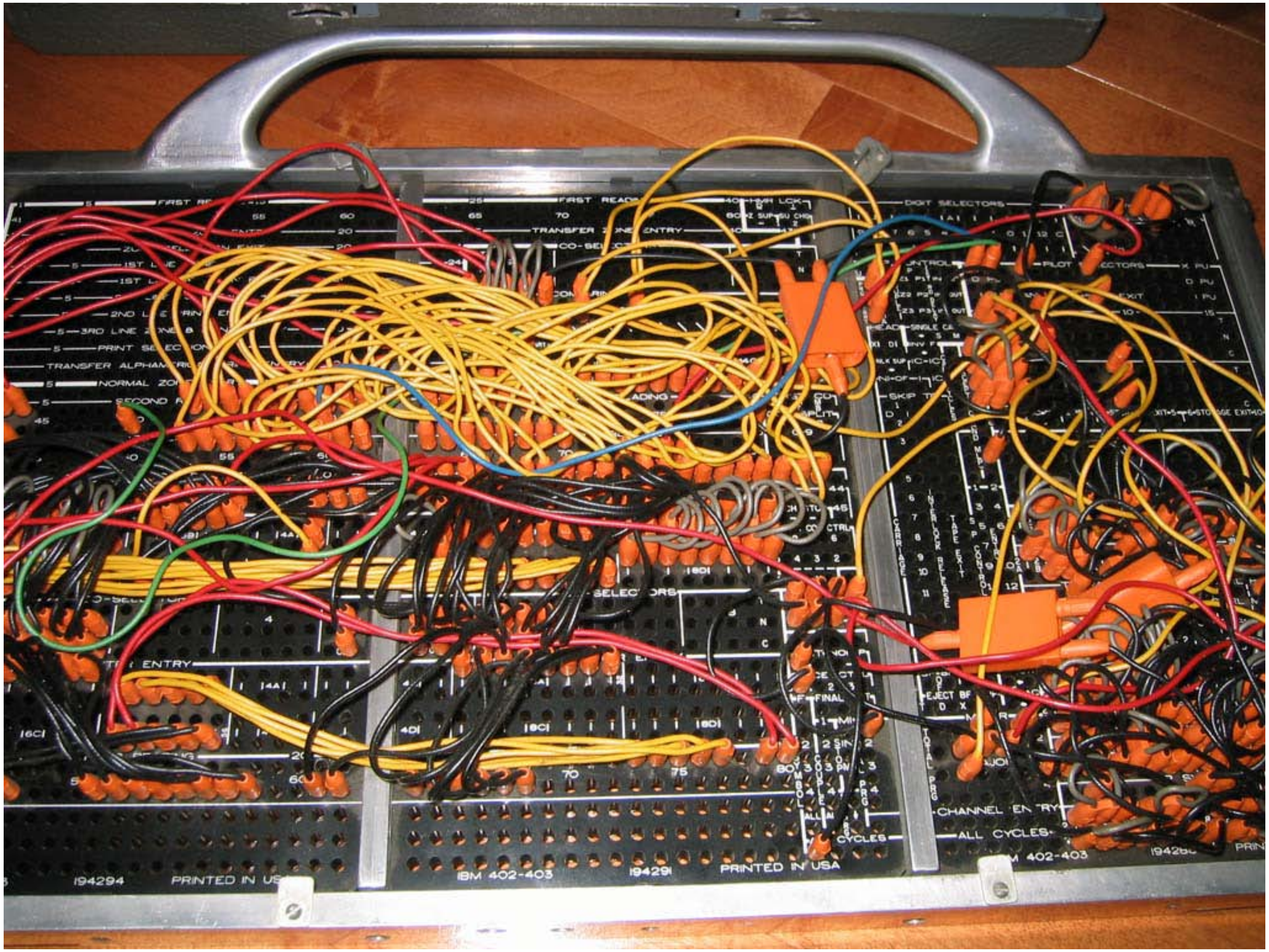
Plugboard Control Panel





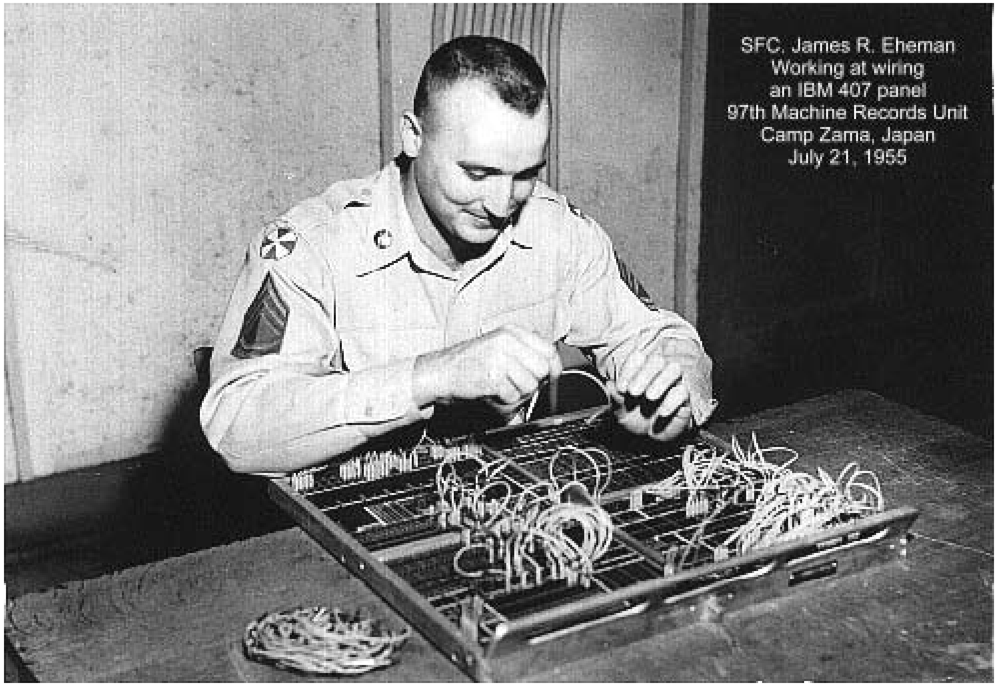
IBM 407 Accounting Machine (1949)

Plugboard Control Panel



Programming a Plugboard

- "Programming" was hand-wiring plugboards.



"Hmm, should I pass this parameter by value or by reference?"

Programming a Plugboard

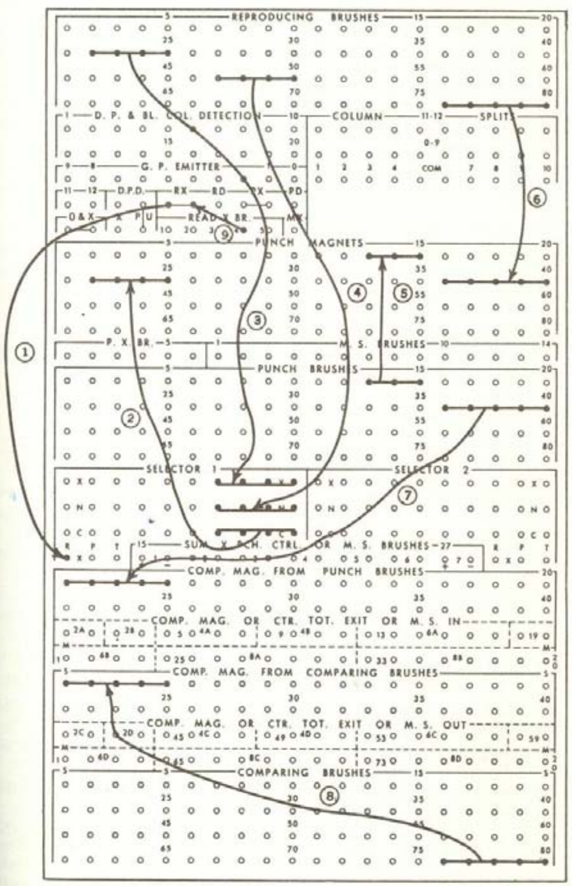


Figure 3.33 Plugboard wiring for IBM 514

- Plugboard wiring diagram
  - It doesn't look too complicated, does it?

Data Processing

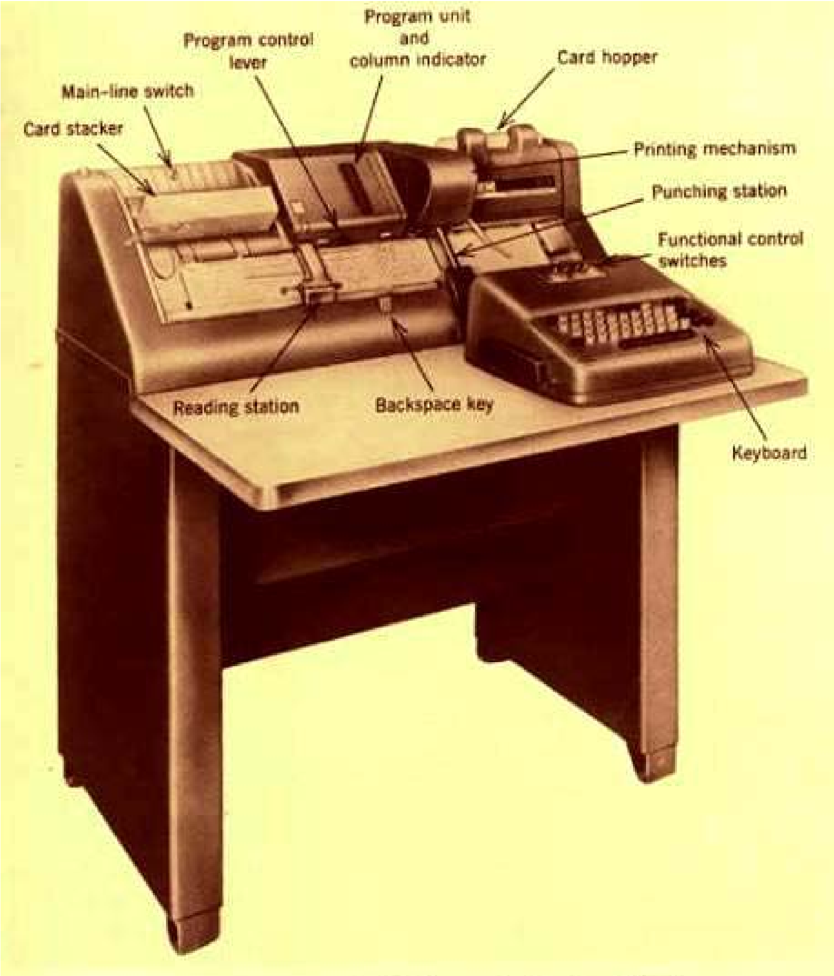


FIGURE 4.11 IBM 026 Keypunch (Courtesy IBM).

- Cards were punched manually at a keypunch machine.
  - Or they were punched automatically by unit-record equipment under program control.

Data Processing

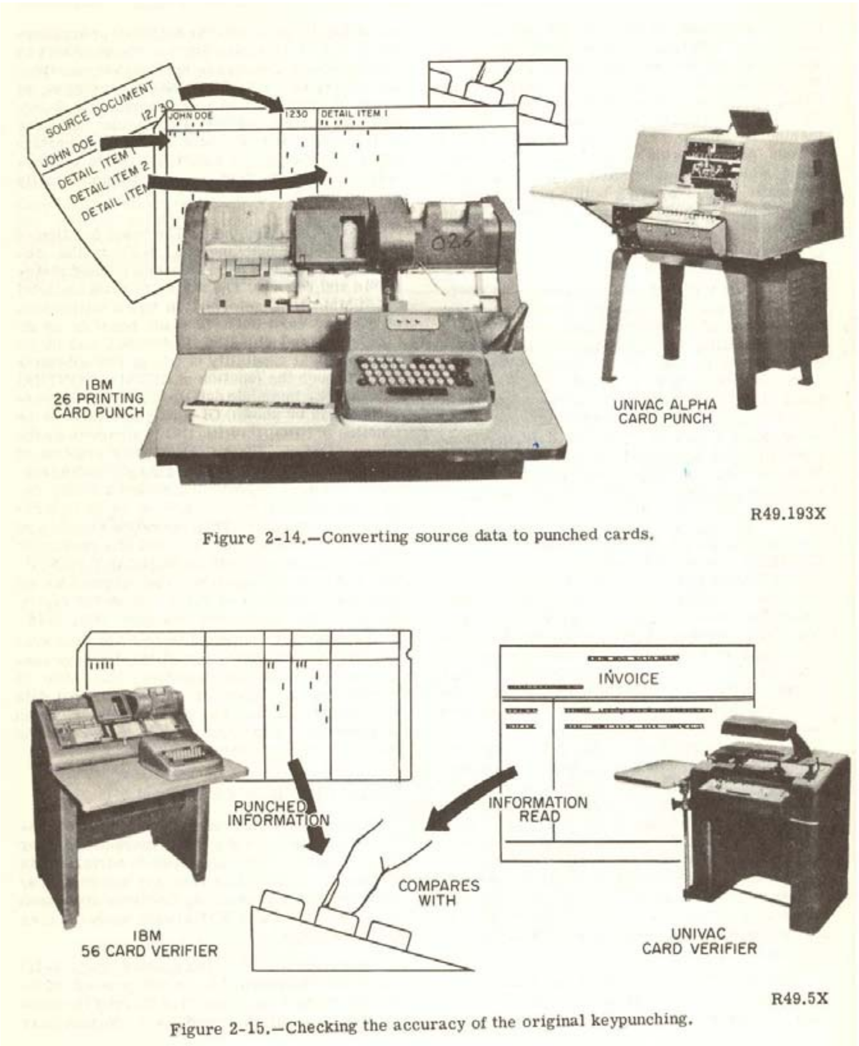


Figure 2-14.—Converting source data to punched cards.

Figure 2-15.—Checking the accuracy of the original keypunching.

- Cards were re-keyed on a verifier to ensure accuracy.
  - Good cards were notched at the top right edge.
  - Bad cards were notched at the top edge above each erroneous column.

CADSL

Data Processing

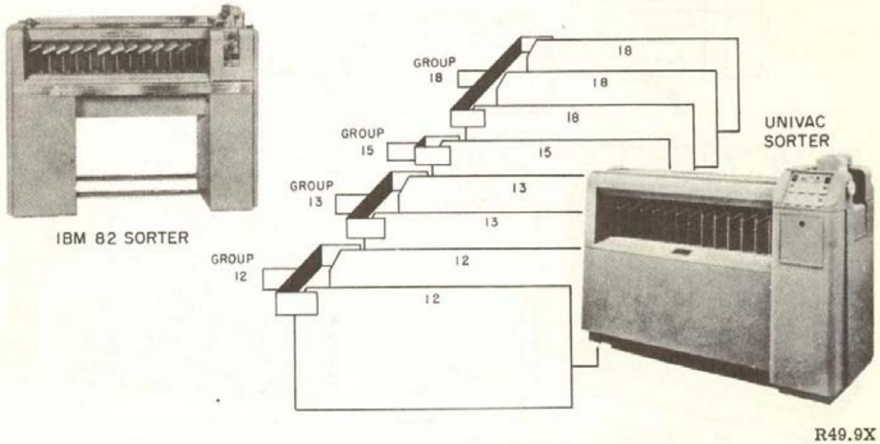


Figure 2-16.—Grouped cards in a definite sequence.

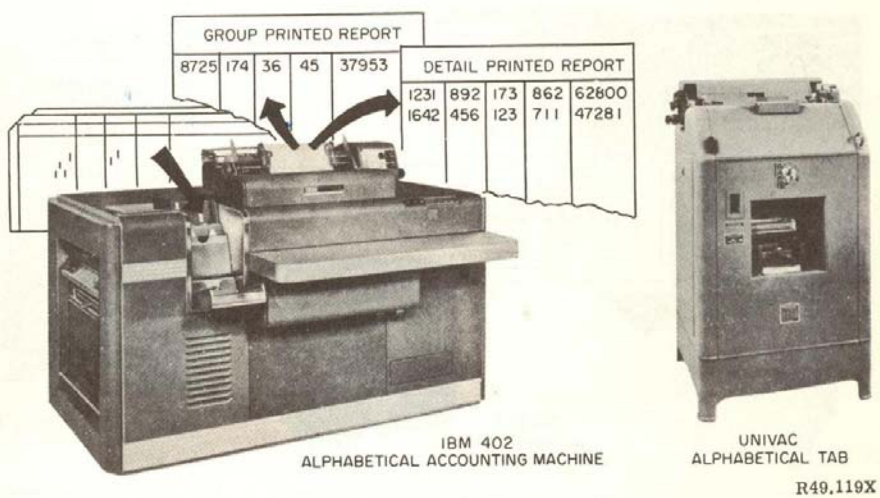


Figure 2-17.—End of the line processing.

- A sorter sorted cards one column at a time.
  - You had to run decks of cards multiple times through a sorter.
- Accounting machines performed arithmetic on card fields and printed reports.

Running a Data Processing Application ...

- ... meant passing decks of cards through a sequence of unit-record machines.
  - Each machine was programmed via its plugboard to perform its task for the application.
  - Each machine had little or no memory.
  - The punched cards stored the data records
  - The data records moved as the cards moved.

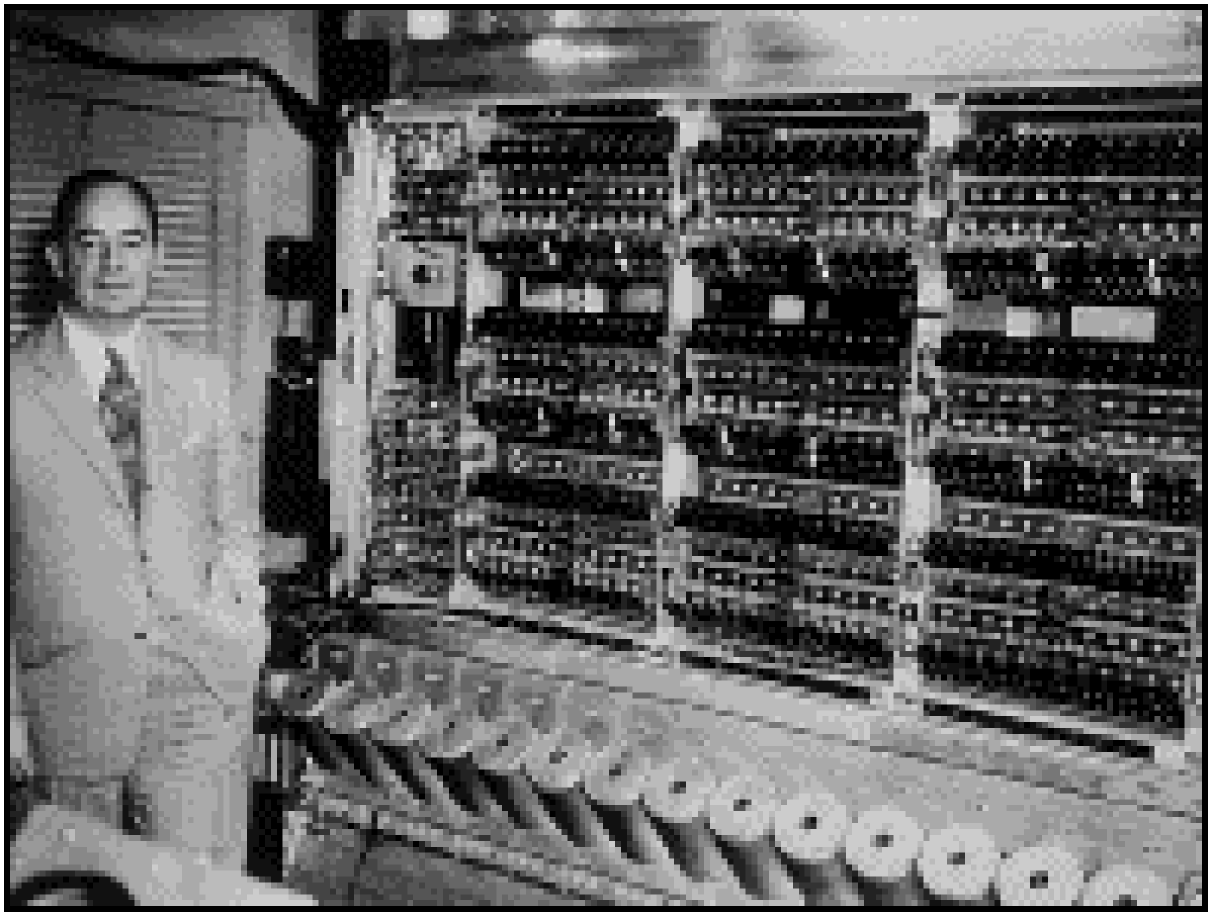
An entire work culture evolved around punched cards!

Von Neumann Bottleneck

- Von Neumann architecture uses the same memory for instructions (program) and data.
- The time spent in memory accesses can limit the performance. This phenomenon is referred to as *von Neumann bottleneck*.
- To avoid the bottleneck, later architectures restrict most operands to registers (temporary storage in processor).

Ref.: D. E. Comer, *Essentials of Computer Architecture*, Upper Saddle River, NJ: Pearson Prentice-Hall, 2005, p. 87.

John von Neumann (1903-1957)



Second Generation Computers

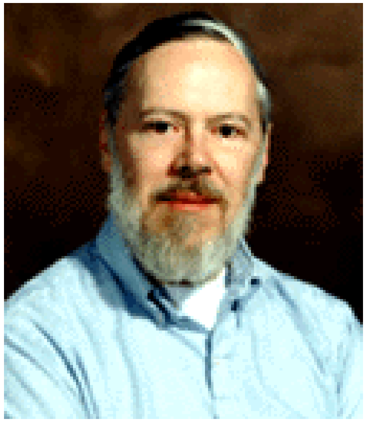
- 1955 to 1964
- Transistor replaced vacuum tubes
- Magnetic core memories
- Floating-point arithmetic
- High-level languages used: ALGOL, COBOL and FORTRAN
- System software: compilers, subroutine libraries, batch processing
- Example: IBM 7094

Third Generation Computers

- Beyond 1965
- Integrated circuit (IC) technology
- Semiconductor memories
- Memory hierarchy, virtual memories and caches
- Time-sharing
- Parallel processing and pipelining
- Microprogramming
- Examples: IBM 360 and 370, CYBER, ILLIAC IV, DEC PDP and VAX, Amdahl 470

C Programming Language and UNIX Operating System

1972

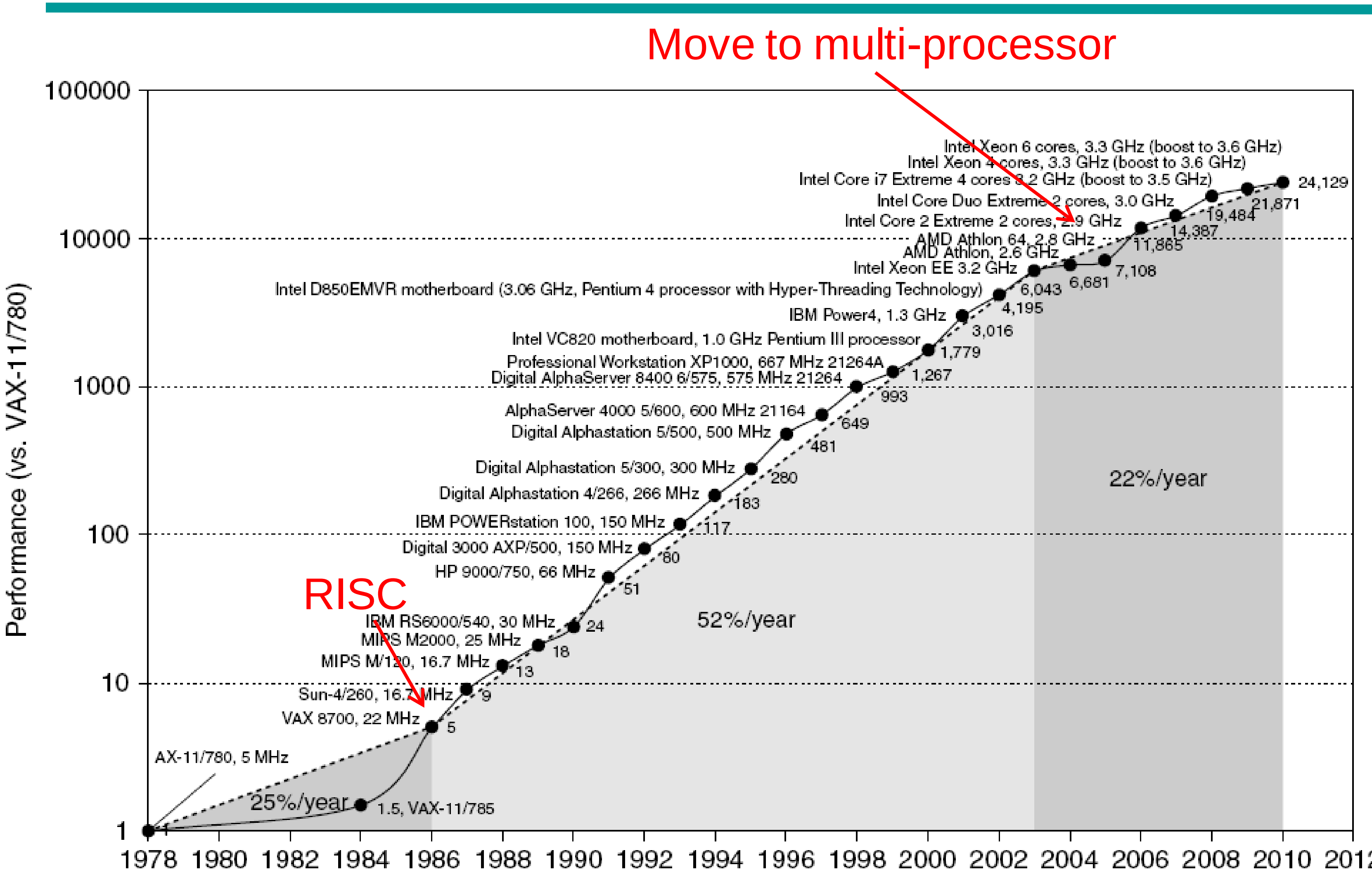


Now

The Current Generation

- Personal computers
- Laptops and Palmtops
- Networking and wireless
- SOC and MEMS technology
- And the future!
    - Biological computing
    - Molecular computing
    - Nanotechnology
    - Optical computing
    - Quantum computing

Single Processor Performance

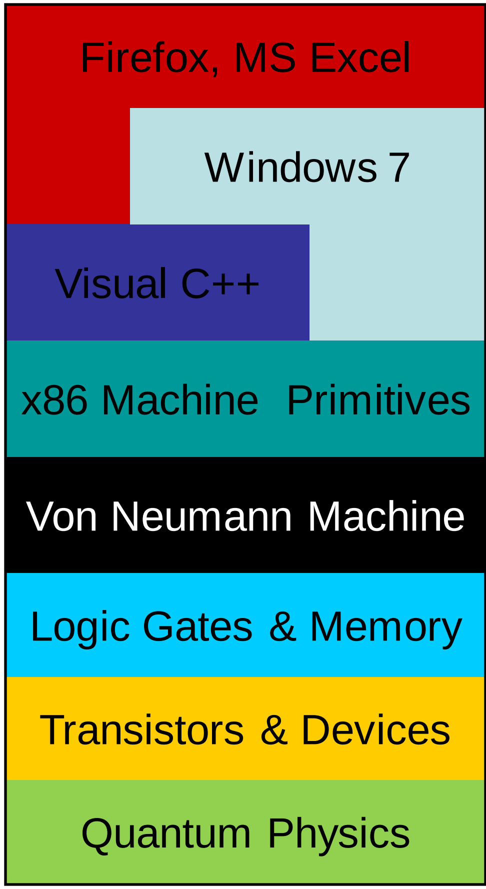


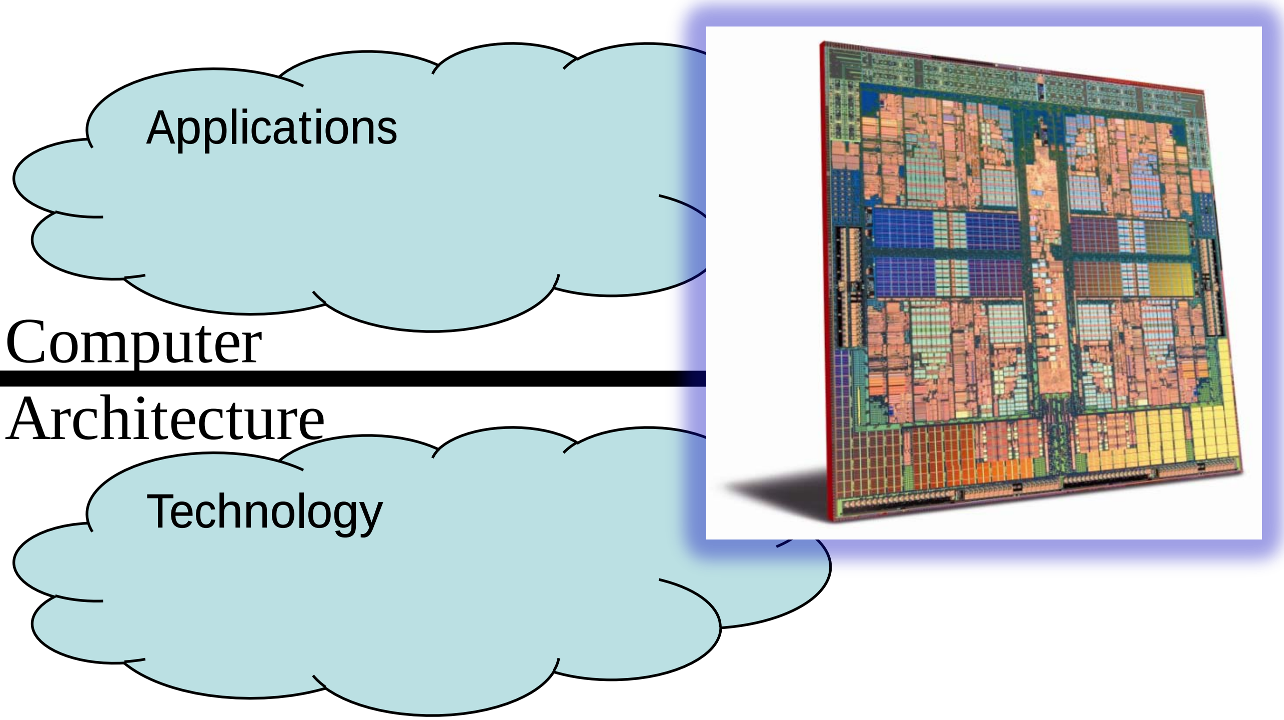
CADSL

Computer Architecture's Changing Definition

- 1950s to 1960s:
  Computer Architecture Course = Computer Arithmetic
- 1970s to mid 1980s:
  Computer Architecture Course = Instruction Set Design, especially ISA appropriate for compilers
- 1990s onwards:
  Computer Architecture Course = Design of CPU (Processor Microarchitecture), memory system, I/O system, Multiprocessors

Computer Architecture

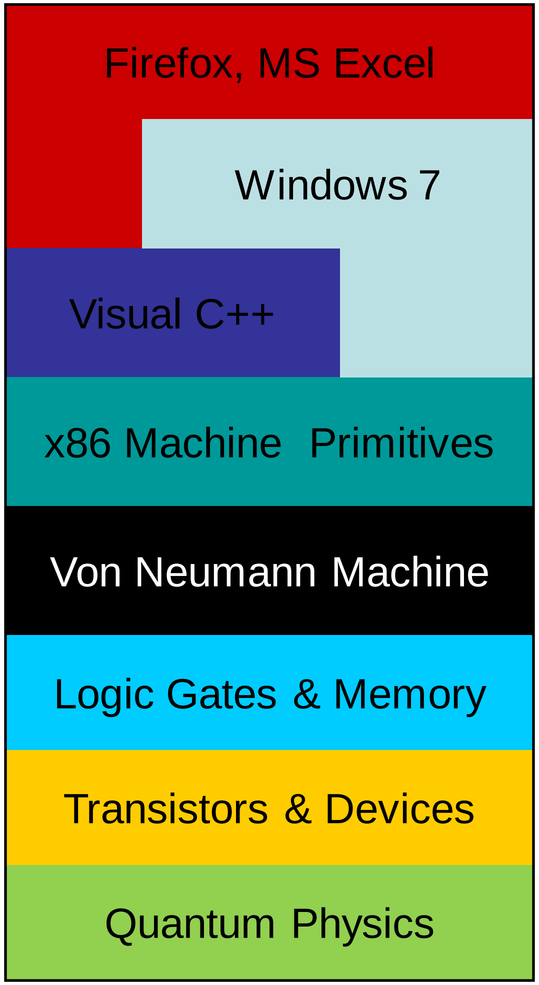




- Rely on *abstraction layers* to manage complexity

What's the Big Deal?

- Tower of abstraction
- Complex interfaces implemented by layers below
- Abstraction hides detail
- Hundreds of engineers build one product
- Complexity unmanageable otherwise



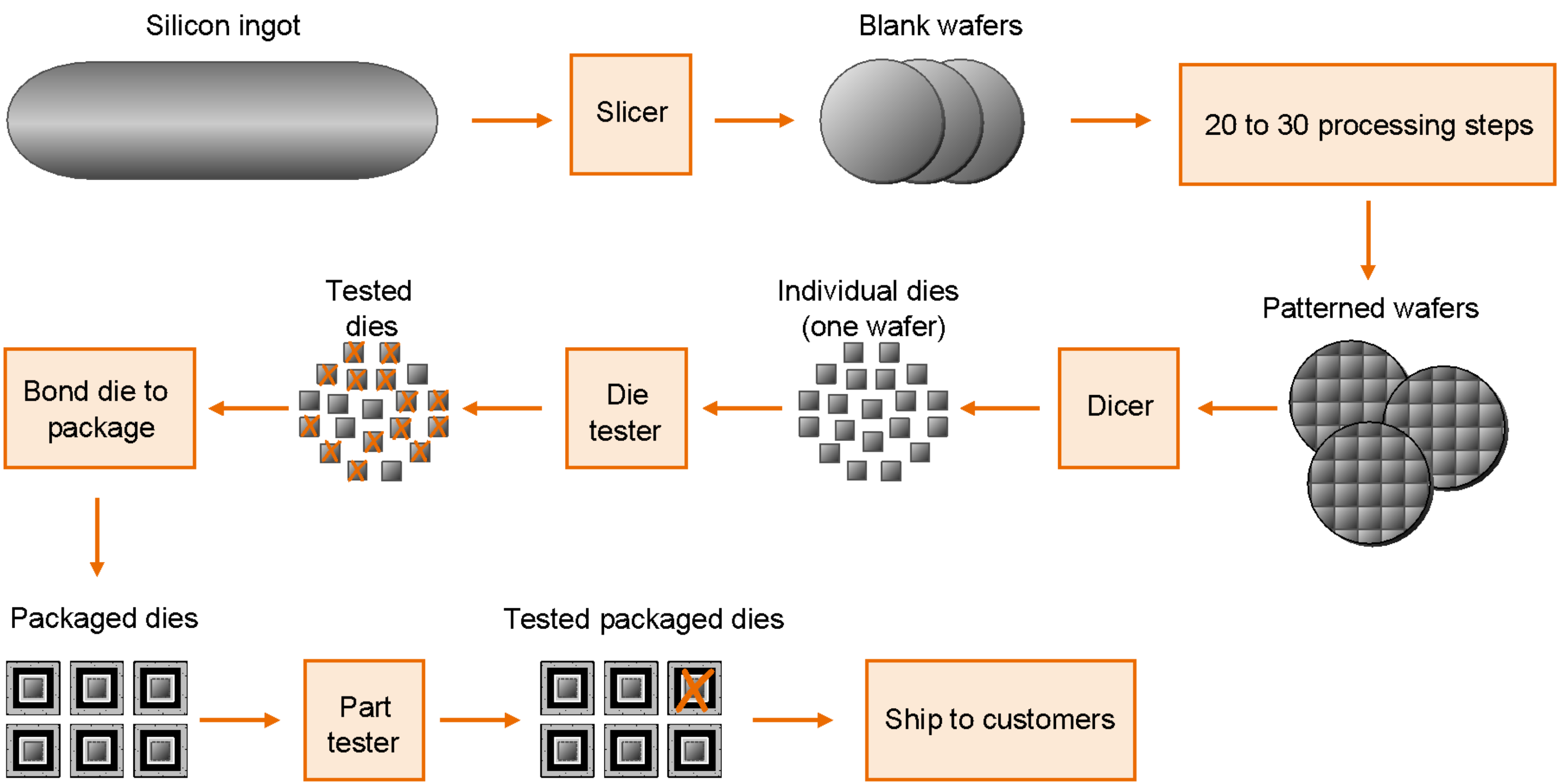
Bottom Line

- Designers must know BOTH software and hardware
- Both contribute to layers of abstraction
- IC costs and performance
- Compilers and Operating Systems

Building Computer Chips

- Complex multi-step process
  - ✓ Slice silicon ingots into wafers
  - ✓ Process wafers into patterned wafers
  - ✓ Dice patterned wafers into dies
  - ✓ Test dies, select good dies
  - ✓ Bond to package
  - ✓ Test parts
  - ✓ Ship to customers and make money

Building Computer Chips



Thank You