MATA48 – Arquitetura de Computadores
Prof. Marcos Barreto
DCC / UFBA

AULA 03

Transição Era PC para Era pós-PC

(baseado no material)

John Hennessy and David Patterson
Stanford and UC Berkeley
13 June 2018
https://www.youtube.com/watch?v=3LVeEjsn8Ts

2019

IBM Compatibility Problem in Early 1960s

By early 1960's, *IBM had 4 incompatible lines of computers!*

701	→	7094
650	→	7074
702	→	7080
1401	→	7010

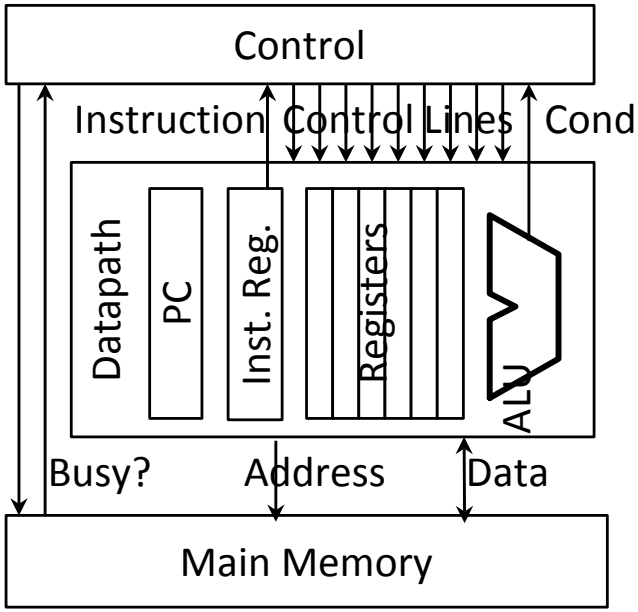
Each system had its own:

- Instruction set architecture (ISA)
- I/O system and Secondary Storage:
  magnetic tapes, drums and disks
- Assemblers, compilers, libraries,...
- Market niche: business, scientific, real time, ...

IBM System/360 – one ISA to rule them all

Control versus Datapath

- Processor designs split between *datapath*, where numbers are stored and arithmetic operations computed, and *control*, which sequences operations on datapath
- Biggest challenge for computer designers was getting control correct



- **Maurice Wilkes** invented the idea of *microprogramming* to design the control unit of a processor*

  - Logic expensive vs. ROM or RAM
  - ROM cheaper than RAM
  - ROM much faster than RAM

* "Micro-programming and the design of the control circuits in an electronic digital computer," M. Wilkes, and J. Stringer. *Mathematical Proc. of the Cambridge Philosophical Society,* Vol. 49, 1953.

Model	M30	M40	M50	M65
Datapath width	8 bits	16 bits	32 bits	64 bits
Control store size	4k x 50	4k x 52	2.75k x 85	2.75k x 87
Clock rate (ROM cycle time)	1.3 MHz (750 ns)	1.6 MHz (625 ns)	2 MHz (500 ns)	5 MHz (200 ns)
Memory capacity	8–64 KiB	16–256 KiB	64–512 KiB	128–1,024 KiB
Performance (commercial)	29,000 IPS	75,000 IPS	169,000 IPS	567,000 IPS
Performance (scientific)	10,200 IPS	40,000 IPS	133,000 IPS	563,000 IPS
Price (1964 $)	$192,000	$216,000	$460,000	$1,080,000
Price (2018 $)	$1,560,000	$1,760,000	$3,720,000	$8,720,000

Figure. Features of four models of the IBM System/360 family; IPS is instructions per second.

IC Technology, Microcode, and CISC

- Logic, RAM, ROM all implemented using same transistors
- Semiconductor RAM ≈ same speed as ROM
- With Moore's Law, memory for control store could grow
- Since RAM, easier to fix microcode bugs
- Allowed more complicated ISAs (CISC)
- Minicomputer (TTL server) example:
  - -Digital Equipment Corp. (DEC)
  - -VAX ISA in 1977
- 5K x 96b microcode

5.120 words X 96 bits

Writable Control Store

- If Control Store is RAM, then could tailor "*firmware*" to application: "*Writable Control Store*"
- Microprogramming became popular in academia
  - Patterson PhD thesis*
  - SIGMICRO was for microprogramming**
- Xerox Alto (Bit Slice TTL) in 1973
  - 1st computer with Graphical User Interface & Ethernet
  - BitBlt and Ethernet controller in microcode

Chuck Thacker

* *Verification of microprograms*, David Patterson, UCLA, 1976

** "The design of a system for the synthesis of correct microprograms," David Patterson, *Proc. 8th Annual Workshop of Microprogramming*, 1975

Microprocessor Evolution

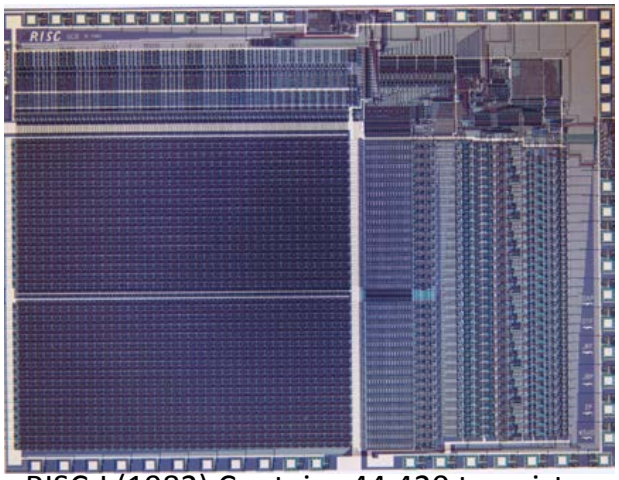
- Rapid progress in 1970s, fueled by advances in MOS technology, imitated minicomputers and mainframe ISAs
- “Microprocessor Wars”: compete by adding instructions (easy for microcode), justified given assembly language programming
- Intel iAPX 432: Most ambitious 1970s micro, started in 1975
  - 32-bit capability-based object-oriented architecture, custom OS written in Ada
  - Severe performance, complexity (multiple chips), and usability problems; announced 1981
- Intel 8086 (1978, 8MHz, 29,000 transistors)
  - “Stopgap” 16-bit processor, 52 weeks to new chip
  - ISA architected in 3 weeks (10 person weeks) assembly-compatible with 8 bit 8080
- IBM PC 1981 picks Intel **8086** for 8-bit bus (and Motorola 68000 was late)
- Estimated PC sales: 250,000
- Actual PC sales: 100,000,000 ⇒ 8086 “overnight” success
- Binary compatibility of PC software ⇒ bright future for 8086

From CISC to RISC

- Use SRAM for instruction *cache* of user-visible instructions
  - Contents of fast instruction memory change to what application needs now vs. ISA interpreter
- Use simple ISA
  - Instructions as simple as microinstructions, but not as wide
  - Compiled code only used a few CISC instructions anyways
  - Enable pipelined implementations
- Further benefit with chip integration
  - In early ‘80s, could finally fit 32-bit datapath + small caches on a single chip
- Chaitin’s register allocation scheme* benefits load-store ISAs

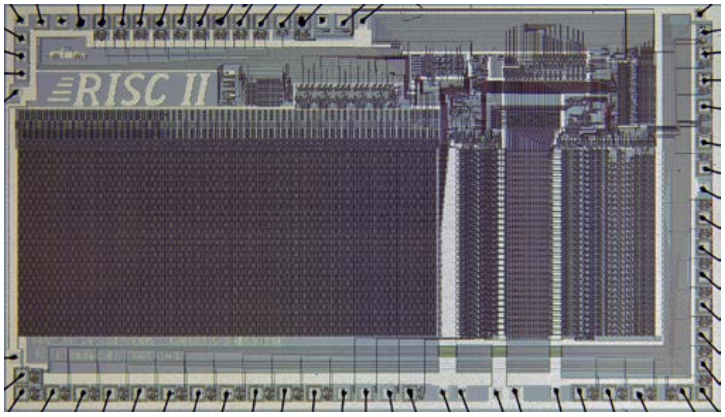
*Chaitin, Gregory J., et al. "Register allocation via coloring." *Computer languages* 6.1 (1981), 47-57.

Berkeley & Stanford RISC Chips

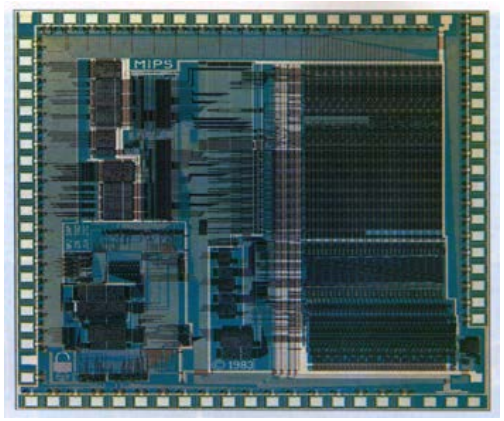


Fitzpatrick, Daniel, John Foderaro, Manolis Katevenis, Howard Landman, David Patterson, James Peek, Zvi Peshkess, Carlo Séquin, Robert Sherburne, and Korbin Van Dyke. "A RISCy approach to VLSI." *ACM SIGARCH Computer Architecture News* 10, no. 1 (1982):

RISC-I (1982) Contains 44,420 transistors, fabbed in 5 µm NMOS, with a die area of 77 mm^2, ran at 1 MHz



RISC-II (1983) contains 40,760 transistors, was fabbed in 3 µm NMOS, ran at 3 MHz, and the size is 60 mm^2



Hennessy, John, Norman Jouppi, Steven Przybylski, Christopher Rowen, Thomas Gross, Forest Baskett, and John Gill. "MIPS: A microprocessor architecture." In *ACM SIGMICRO Newsletter*, vol. 13, no. 4, (1982).

Stanford MIPS (1983) contains 25,000 transistors, was fabbed in 3 µm & 4 µm NMOS, ran at 4 MHz (3 µm), and size is 50 mm^2 (4 µm) (Microprocessor without Interlocked Pipeline Stages)

"Iron Law" of Processor Performance: How RISC can win

$$\frac{\text{Time}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} * \frac{\text{Clock cycles}}{\text{Instruction}} * \frac{\text{Time}}{\text{Clock cycle}}$$

- CISC executes fewer instructions per program
  (≈ 3/4X instructions),
  but many more clock cycles per instruction
  (≈ 6X CPI)
  ⇒ RISC ≈ 4X faster than CISC

"Performance from architecture: comparing a RISC and a CISC with similar hardware organization," Dileep Bhandarkar and Douglas Clark, *Proc. Symposium, ASPLOS*, 1991.

CISC vs. RISC Today

PC Era

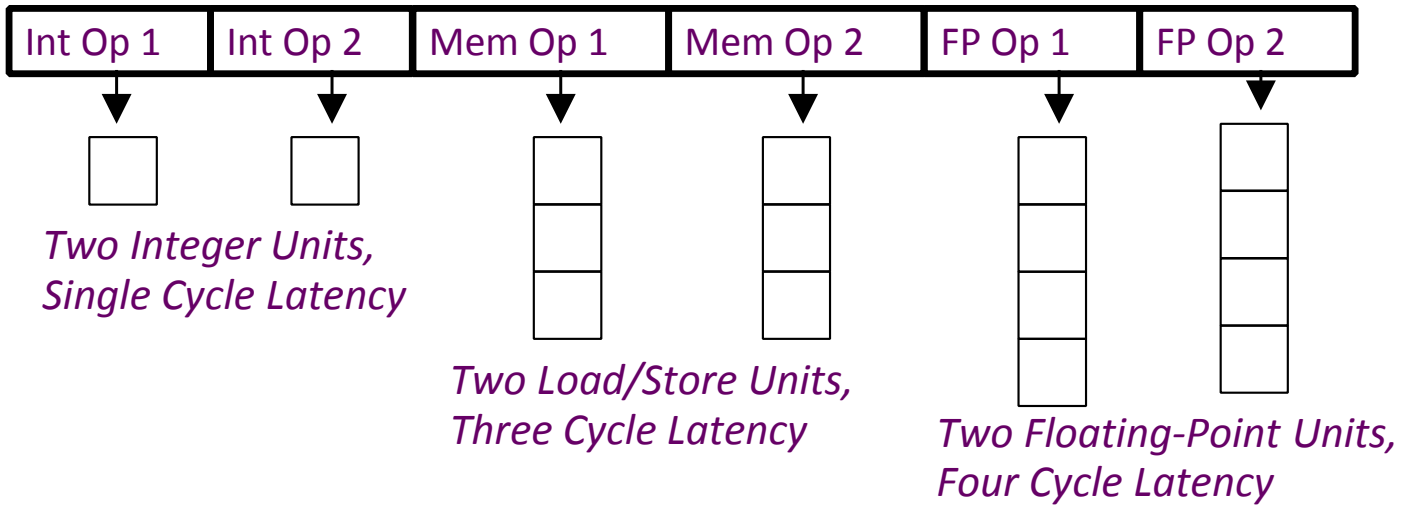
- Hardware translates x86 instructions into internal RISC instructions
- Then use any RISC technique inside MPU
- > 350M / year !
- x86 ISA eventually dominates servers as well as desktops

PostPC Era: Client/Cloud

- IP in SoC vs. MPU
- Value die area, energy as much as performance
- > 20B total / year in 2017
  - x86 in PCs peaks in 2011, now decline ~8% / year (2016 < 2007)
  - x86 servers ⇒ Cloud ~10M servers total* (0.05% of 20B)
- 99% Processors today are RISC

*"A Decade of Mobile Computing", Vijay Reddi, 7/21/17, *Computer Architecture Today*

VLIW: Very Long Instruction Word (Josh Fisher)



- Multiple operations packed into one instruction (like a wide microinstruction)
- Each operation slot is for a fixed function
- Constant operation latencies are specified
- Architecture requires guarantee of:
  - Parallelism within an instruction ⇒ no cross-operation RAW check
  - No data use before data ready ⇒ no data interlocks

From RISC to Intel/HP Itanium, EPIC IA-64

- EPIC is Intel's name for their VLIW architecture
  - "Explicitly Parallel Instruction Computing"
  - A binary object-code-compatible VLIW
  - Developed jointly with HP starting 1994
- IA-64 was Intel's chosen 64b ISA successor to 32b x86
  - IA-64 = Intel Architecture 64-bit
  - AMD wouldn't be able to make IA-64, unlike x86, so had to make 64-bit x86
- First chip late (2001 vs 1997), but eventually delivered (2002)
- Many companies gave up RISC for Itanium since it was widely believed to be inevitable (Microsoft, SGI, Hitachi, Bull, …)

VLIW Issues and an "EPIC Failure"

- Compiler couldn't handle complex dependencies in integer code (pointers)
- Code size explosion
- Unpredictable branches
- Variable memory latency (unpredictable cache misses)

  -Out of Order techniques dealt with cache latencies
- Out of Order subsumed VLIW benefits
- "*The Itanium approach...was supposed to be so terrific –until it turned out that the wished-for compilers were basically impossible to write*."

  - Donald Knuth, Stanford
- Pundits noted delays and under performance of Itanium product ridiculed by the chip industry

Itanimum ⇒ "Itanic" (like infamous ship *Titanic*)

Summary Part I: Consensus on ISAs Today

- Not CISC: no new general-purpose CISC ISA in 30 years
- Not VLIW: no new general-purpose VLIW ISA in 15 years.
  VLIW has failed in general-purpose computing arena
  - Complex VLIW architectures close to in-order superscalar in complexity, no real advantage on large complex apps
  - Although VLIWs successful in embedded DSP market

    (Simpler VLIWs, easier branches, no caches, smaller programs)
- RISC! Widely agreed (still) that RISC principles are best for general purpose ISA!