

datto

Processor Architectures



Fred Mora - System Engineering, Datto

Agenda

Attack of the sabertooth sysadmin!

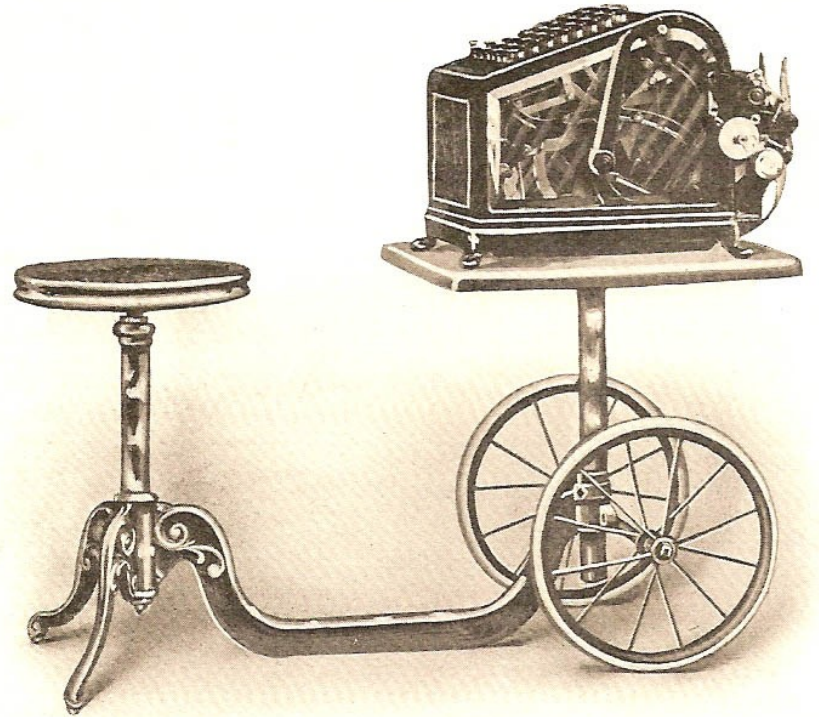
- Early on, the “Von Neumann architecture” imposed itself for digital computers
 - All-purpose CPU accessing memory
 - Memory contains program and data
- The only alternative was specialized circuitry that was very inflexible.



Cash register built in 1904 in Ohio (USA) for a Czech merchant - By Kozuch [CC BY-SA 3.0] from Wikimedia Commons

What is an instruction set architecture?

- An instruction set architecture (ISA) is, fundamentally, the “look and feel” of a CPU’s assembly language, hence the name.
- But it goes deeper than this.
- An ISA defines the instruction set of the processor, thus the programming model: registers, caching, addressing, control flow, integration to subsystems, data types, I/O, etc.



Eureka Adding Machine Carriage with Burroughs Adding Machine, 1905 - Source: Early Office Museum, officemuseum.com

Maurice Wilkes and the EDSAC

- 1947: Maurice Wilkes starts work on a digital computer at Cambridge University
- 1949: The EDSAC (Electronic Delay Storage Automatic Calculator) starts running. Uses mercury delay lines for memory (512 18-bit words)
- Notable because of its successor, EDSAC2, the first microprogrammed architecture.
- This technique allowed a rudimentary CPU to offer interesting instructions to programmers.

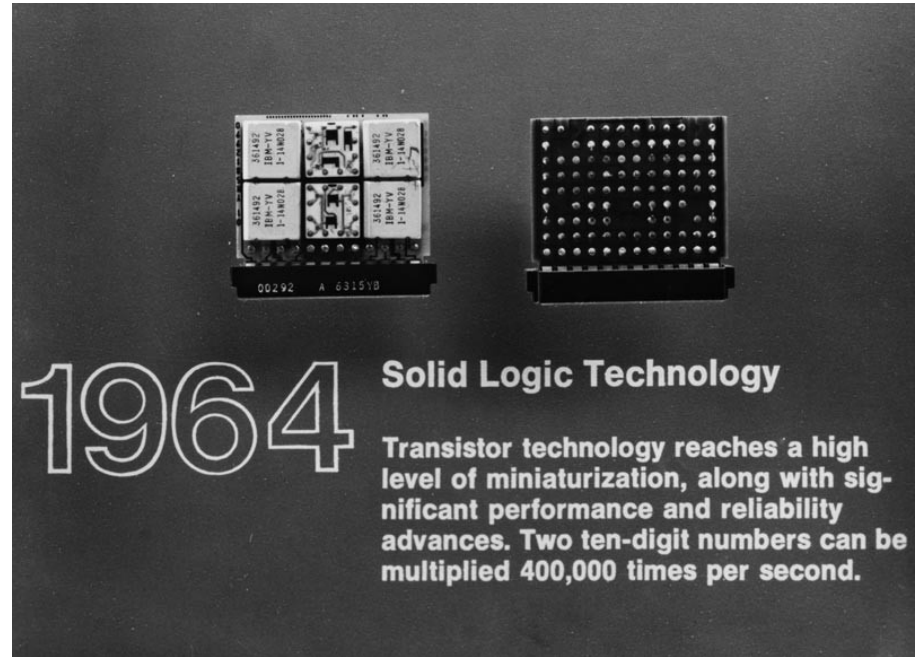
“As soon as we started programming, we found to our surprise that it wasn't as easy to get programs right as we had thought. ... I can remember the exact instant when I realized that a large part of my life from then on was going to be spent in finding mistakes in my own programs.” – Maurice Wilkes



EDSAC2 - By Unknown - University of Cambridge Computer Laboratory Archive, CC BY 2.0 uk, <https://commons.wikimedia.org/w/index.php?curid=11563773>

The Great Ancient... The IBM/360

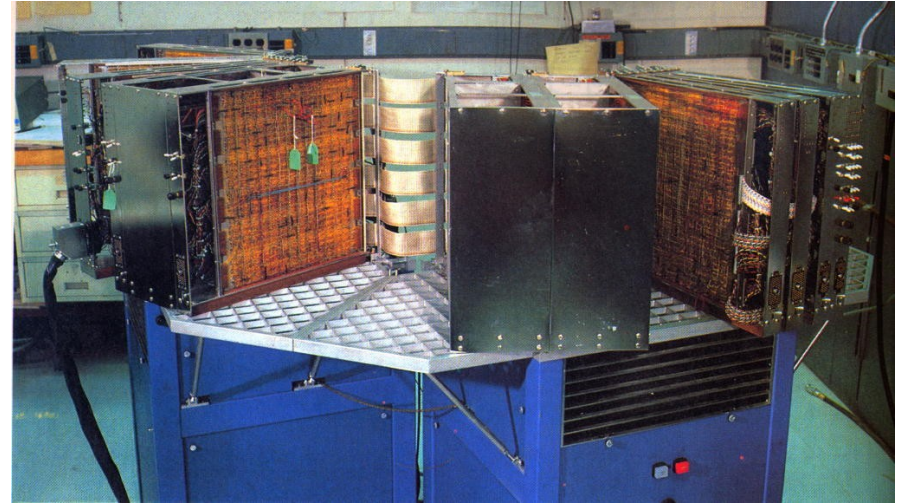
- Early computer design had very different ISAs
 - Every new model was completely different
 - Even within constructor catalog
- This wrecked havoc on software investment
- The IBM/360 architecture innovated by having one ISA across a very wide machine range
- Survives in today's zSeries mainframe
- Provided a complex instruction set
- But relatively simple hardware through microprogramming



Source: IBM archives

1974: IBM's Project 801

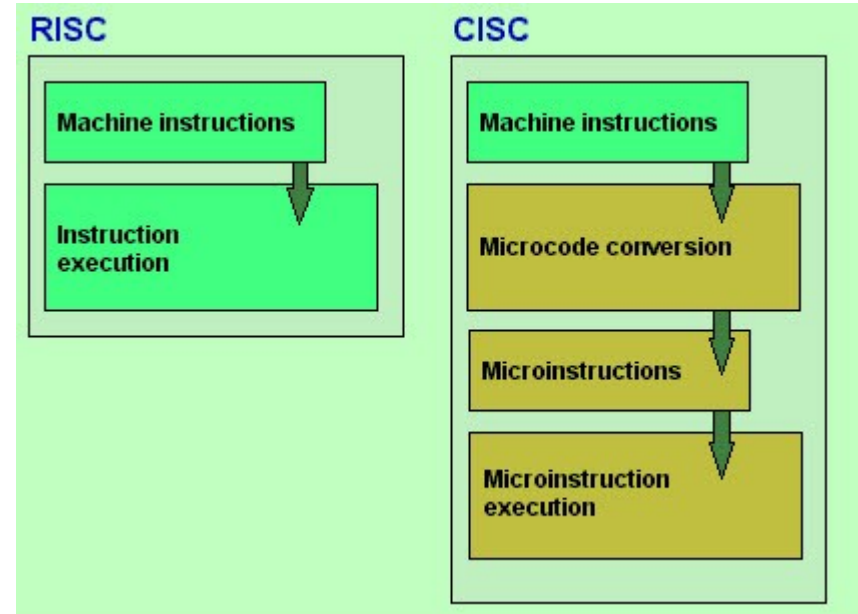
- Needed a fast processor for telecom, e.g. phone switches.
- Estimates was 10 MIPS. A big mainframe delivered 2 MIPS.
- Project started under John Cocke in Bldg 801 of TJ Watson Research Center, hence the name
- Resounding success, delivered 15 MIPS
- Programmer-hostile instruction set meant the chip needed optimizing compilers.
- Gave birth to first RISC implementation.
- The project became the basis for the current line of IBM POWER processors, starting in 1990 with the RISC System/6000.



IBM 801 RISC processor - Internal prototype. Source: Forth Institute of Computer Science, RISC history, Pr. Katavenis, Dept. of CS, Univ. of Crete.

RISC vs. CISC

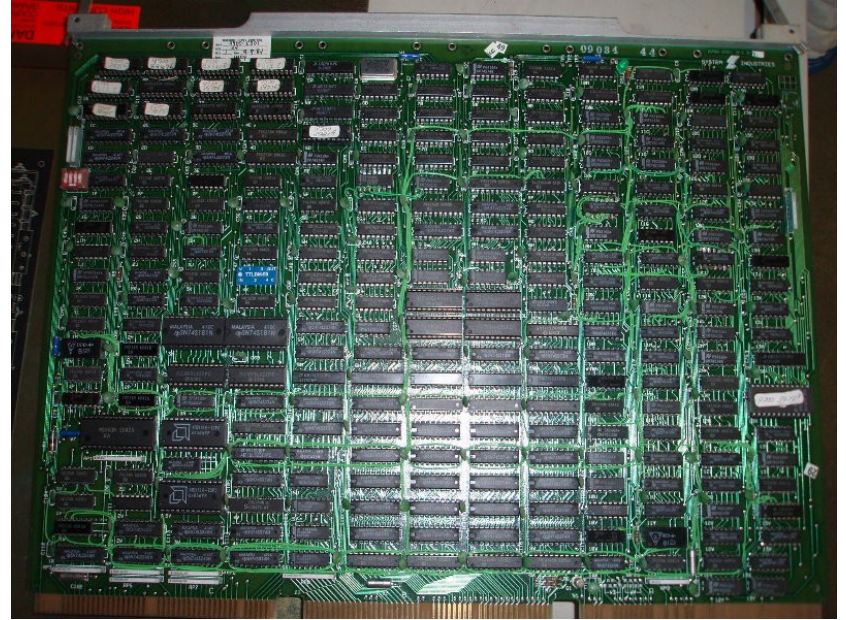
- John L. Hennessy's landmark paper, VLSI Processor Architecture (1984):
<https://pdfs.semanticscholar.org/ee02/e249bfdbc94a41acd9041d5ee9eadf77b169.pdf>
- RISC reason for speed: instructions are executing quickly, very few clock cycles
- CISC processors need about 6-8x more clock ticks per instruction than RISC
- RISC assembly is 2x the code path length (twice as many instructions for the same compilation on average)
- So RISC still has a 3-4x advantage
- And, since RISC is simpler, their clock frequency can scale up more easily.



Source: Edgex blog

The VAX

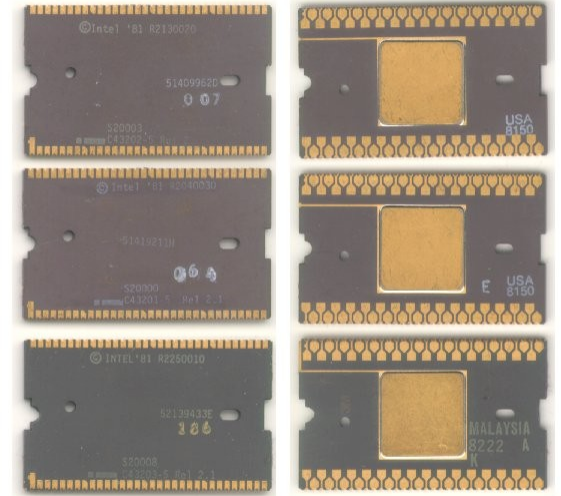
- 1977: DEC VAX 11
- Built from standard TTL and RAM ICs
- Becomes the standard minicomputer
- Lots of very complex instructions to please assembly programmers
- But under the hood, simple hardware (by today's standards)
 - 99 different controllable execution blocks
 - 4096 99-bit words of microcode
- Floating point, multiplication, substring match, polynomial computation, queue management!



VAX11/750 motherboard. Note the TTL circuitry. Source: rogerwilco.org

The iAPX 432 failure (1975-1986)

- 1975: After the 8080 success, Intel starts a new ISA project known as the 8800 processor
- Announced in 1981 as the iAPX 432
- High-level ISA of the future, meant to be programmed in the new standard, future-proof language, ADA – selected by the DoD as COBOL's successor. No assembly please.
- Bloated ISA led to a 90,000 transistor design
- Which had to be split in 2 costly chips
- Interchip bandwidth limits led to low performance
- 1982: The 80286 was 4x faster at same clock speed
- Only one system builder (British company High Integrity Systems) ever adopted the iAPX 423 chipset
- Abandoned after abysmal sales in 1986
- Morale: Good design intents aren't enough
- It's sad to see the slapped-together 80286 was much more successful.



iAPX 432 chipset. Source: David Patterson, Google.

The 8086

- In 1976, Intel worries that the 8800 will take too long
- Gets Gordon Moore to develop a new processor, the 8086, compatible with 8-bit line, with a shitty ISA tackled in 3 weeks, just to hold them over until the 8800.
- The 8086 performance lagged way behind competitors such as the Motorola 68000 (in Mac, Atari STs, Amiga)
- Then of course some IBM engineer in Boca Raton decided to use the 8088 for their new model 5150 aka PC.
- Reasoning: If needed, IBM could acquire that small Intel shop, whereas Motorola was too big.
- This gave birth to the PC architecture, the Lubyanka of CPU architecture.



IBM 5150 Personal Computer motherboard. The big chip on the lower right is the 8088. Source: http://gunkies.org/wiki/IBM_5150

Speeding up the 8086

- Due to complexity and heating, clock speed is limited to 3-4 GHz.. POWER8: 4.2 GHz (although POWER9 has a lower clock speed currently)
- More cores rather than more speed
- To increase performance, the chip is internally looking more like a RISC processor
- 8086 instructions are decoded into microcode

VLIW or the return of the Good Idea Fairy

- One fine day at Intel (1994): [Why spend all that time decoding CISC assembly instructions into RISC microcode?](#)
- [Why not directly assemble microcode?](#)
- [Make it non-8086 compatible so that pesky AMD can't copy it.](#)
- HP: [Hey, our PA-RISC architecture is getting long in the teeth, mind if we join?](#)
- [And screw scheduling and branch prediction, that's the compiler's job! Say what? Alan Turing says nope? Who cares!](#)
- Thus was born the IA-64 architecture a.k.a EPIC.
- 1999: Folks, meet the Itanium, our one and future CPU.
- First CPU released in 2001, never gained traction
- Expensive failure due to problems writing good compiler while the 8086 speed kept improving.

THE GOOD IDEA FAIRY



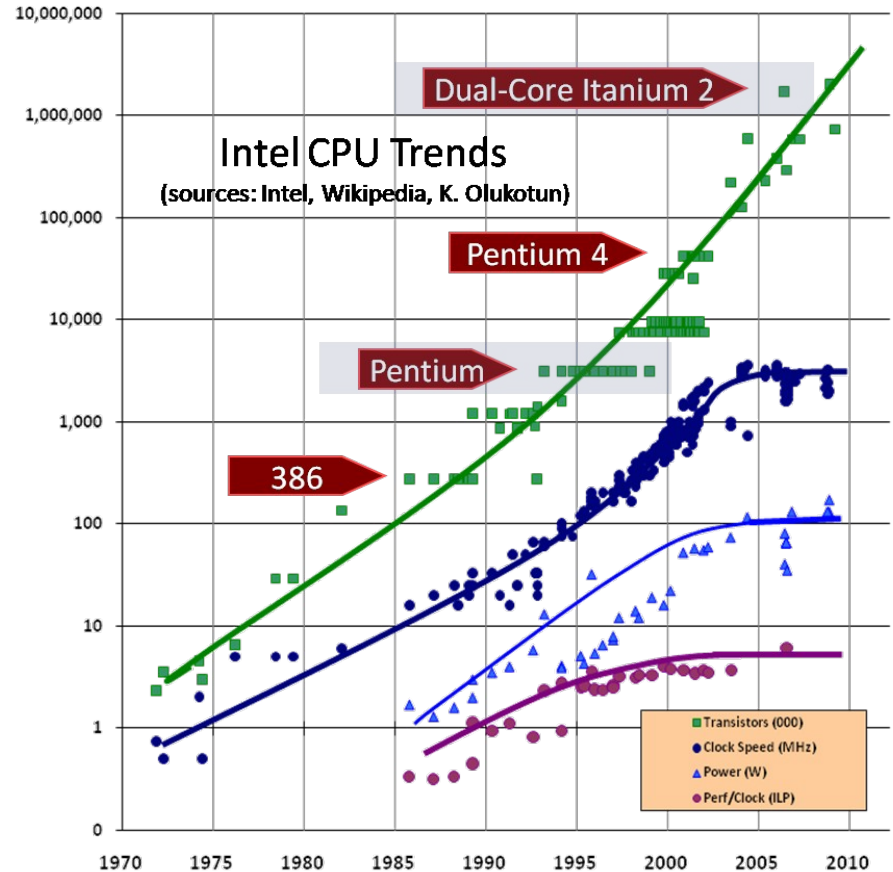
HAS COME FOR A VISIT

Why aren't cores getting faster

- Today's processor attempt to offer more performance by offering more cores
- But for desktop users as well as for many workloads, parallelism is a pipe dream
- Scripting languages are quick and fun, but they give you slow, interpreted, single-thread programs
- So why aren't cores getting faster?

Moore's Law is mostly dead

- Moore originally predicted that the number of transistor per chip would double at constant cost every 18 to 24 months
- It had a good run, but it has hit hard limits with modern gigascale chips.
- Even with upcoming Extreme Ultra-Violet technology, transistors cannot keep getting smaller for much longer without getting down to a few molecules. Then what? 3D circuits? Silicon alternatives?
- Moore's Law became Moore's guideline, then Moore's suggestion.
- Higher gate count was an "easy" and reliable way to increase performance
- Smaller transistors mean faster switching
- But larger chips mean signal propagation and clock sync become an issue
- Also, Dennard's Scaling Law stopped working in the mid-2000s



Source: <http://www.gotw.ca/publications/concurrency-ddj.htm>

Dennard's Law of Transistor Scaling

- Robert H. Dennard is the inventor of dynamic RAM.
- He observed in 1974 that smaller transistors consume less power since power dissipation is proportional to speed and gate area
- When density double, clock speed increases by 40% at equal power consumption.
- But smaller transistors started leaking current at around 90 nm, thus increasing power consumption dramatically
- Higher consumption means more power to dissipate
- This in turn limits clock speed
- Heroic technological efforts to limit current leak led to a clock speed increase of about 30% in 10 years.

“TANSTAAFL” (Robert A. Heinlein)

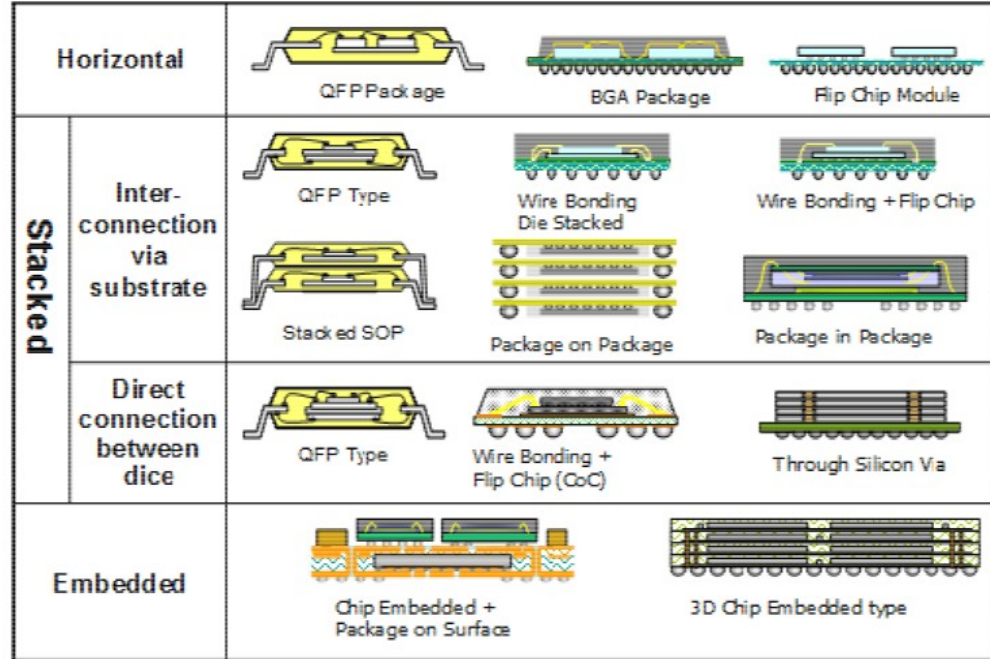
- In “The Moon is a Harsh Mistress”, Heinlein popularized the expression “Tanstaaf!” as an equivalent of “duh” - There ain’t no such thing as a free lunch. There is always a hidden cost.
- Programmers enjoyed their “free lunch” in the ever-faster clock speed years
- Modern programming focused on high-productivity practices that require lots of computing power: Interpreted languages, late binding, dynamic typing, mono-threading.
- Example: Bash, Python, Ruby. Great languages, complete dogs.
- We are now saddled with a performance wall and no faster CPU to pick up the slack...
- ... while we accumulate layers upon layers of slow, single-threaded, interpreted code.
- The next revolution will be finding efficient ways to use parallelism and JIT compilation in interpreted languages.

Current lithography tech

- Lithography is the process that etches silicon wafers
- Current state of the art is 7 nm - TSMC just (April 2018) announced mass production at 7 nm
- Enabler is EUV scanners from ASML (Connecticut)
- They cheat: pitch between transistors is 24 to 40 nm
- 5-nm demo is planned for 2020
- At this scale, a DRAM memory bit cell holds only a few electrons

Consumer electronics: SiP

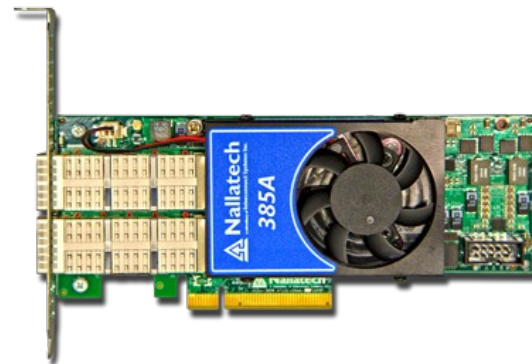
- Systems in Packages is a fancy term for stacking chips
- Even at today's density, chips need to be very densely packed to satisfy consumer needs
- Do not confuse with Session Initiation Protocol (there are only 17500 Three-Letter Acronyms after all)



Source: International Technology Roadmap for Semiconductors 2.0 - 2015 Edition - Executive Report

The future?

- High volume is not necessarily Intel
- Open source RISC-V architecture gaining traction
 - Lots of very small cores
 - Great for AI and realtime image processing
 - Single ISA for very different kind of processing
- Modern workloads run on:
 - graphics processors (with CUDA)
 - FPGA accelerator (with OpenCL) - Intel supplies an OpenCL toolkit for Altera FPGAs.



Nallatech 385A, one of several FPGA OpenCL-compatible boards

Questions?