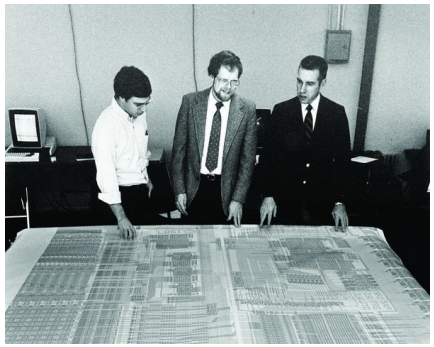
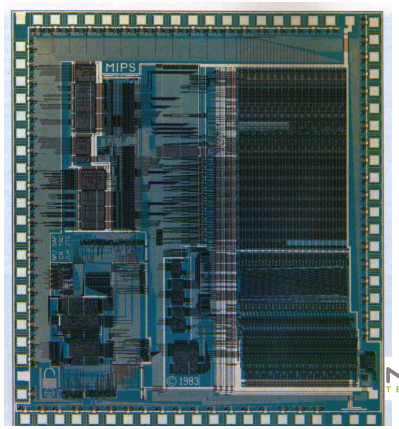
Midterm I: May 3rd

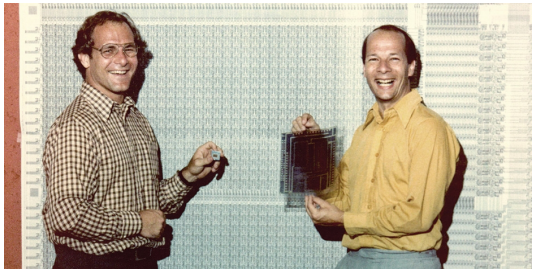
Lessons from HW1?

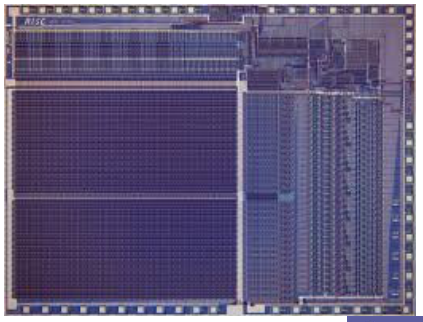
- Windows does a lot in the background (updater, virus scanner, etc). Under load not so.
- A lot of disc accesses and cache misses while idle.
- Page faults happen in clusters. A page fault on one process then it happens on others at the same time.
- There are a lot of context switches that occur.
- Not a lot of threaded parallelism
- most CPU usage from the idle process -> accounting only
- windows telemetry uses your CPU, network and your data
- windows defender caused the disc usage to go up to 100% and stay there
- background "bloatware" took more CPU than the foreground process in use
- chrome was eating a ton of CPU time; launched 10 different processes to run a single video
- 4K streaming consumed far more CPU time than 1080 video

RISC v CISC









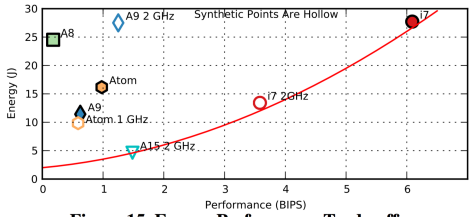
John Cooke (1975)

Why did CISC happen?

- Marketing: More instructions, more better
- Code density
  - Memory was expensive & slow (and still is)
- Support HLL (High Level Language) *debatable!
- Expose cool tricks to the user

What & Why is RISC?

- Fixed length instructions -> simplified decoding logic and instruction encoding
- Small number of instructions -> less effort for the compiler*, speed (faster), less hardware to support
- More easily utilize ILP
  - e.g. ADD [BX + DX], AX ; [BX + DX] = [BX + DX] + AX ; LOAD t0, [BX + DX], ADD t0, AX, t0, STORE [BX + DX], t0
  - Simpler instructions are easier to pipeline
- By not supporting a lot of instructions can make better use of silicon area elsewhere (definitely true in 80s and 90s)
- Not microprogrammed (there is "PAL" code).
- Easier to design
  - Less prone to errors
- General purpose registers



RISC-V

SPECIFICATIONS

- User-Level ISA Specification v2.1
- Draft Privileged ISA Specification v1.9.1

SOFTWARE TOOLS

- RISC-V Tools
  - GCC
  - GDB
  - LLVM
  - Clang
  - Verification Suite
- Linux
- Yocto
- FreeBSD

SOFTWARE IMPLEMENTATIONS

- Spike (ISA Simulator)
- QEMU
- RISCVEMU
- ANGEL (JavaScript ISA Simulator)