CISC, RISC, and DSP Microprocessors

Douglas L. Jones

ECE 497

Spring 2000

Outline

- Microprocessors circa 1984
- RISC vs. CISC
- Microprocessors circa 1999
- Perspective: why did things evolve so?
- The future of embedded microprocessors (?)

Goals for this Lecture

- Understand key differences between various microprocessor types
- Understand why they're that way

General-Purpose Microprocessor circa 1984: Intel 8088

- ~100,000 transistors
- Clock speed: ~ 5 MHz
- Address space: 20 bits
- Bus width: 8 bits
- 100+ instructions
- 2-35 cycles per instruction
- Microcoded architecture

- Many addressing modes
- Relatively inexpensive

Apparent Trends

- Larger address space
- Higher clock speed
- More transistors
- More instructions
- More arithmetic capability
- More memory management support

- Wider buses for high-performance processors
- High-end processors more expensive

DSP Microprocessors circa 1984: TMS32010

- Clock speed: 20 MHz
- Word/bus width: 16 bits
- Address space: 8, 12 bits
- ~50,000 transistors
- ~ 35 instructions
- 4-cycle execute of most instructions

- Harvard architecture: separate program and data memory, buses
- 16x16 hardware multiplier
- Double-length accumulator with saturation
- A few special DSP instructions
- Relatively expensive

Apparent Trends

- Higher clock rates
- Fewer cycles/instruction
- Somewhat expanded address spaces
- More specialized DSP instructions
- Lower cost

- Meanwhile, there was RISC ...

RISC Processors circa 1984

- Academic research topic
- 12-16 instructions
- Single-cycle execute
- No microcode!

Arguments Advanced for RISC

- Small, heavily optimized instruction set executable in single short cycle
- All instructions same size
- No microcode = faster execution
- Extra speed more than offsets increased code size, reduced functionality
- Better compiler target

- (Simple enough for academic designs, class projects!)

Arguments Advanced for CISC

- Fewer instructions per task
- Shorter programs
- Hardware implementation of complex instructions faster than software
- Extra addressing modes help compiler

The RISC vs CISC Controversy

- Lots of argument
- Hundreds of papers
- Hottest topic in computer architecture
- In mid to late ‘80s, many RISC uPs introduced: MIPS, SPARC (Sun), MC88000, PowerPC, I960 (Intel), PA-RISC
- For a time, RISC looked tough to beat ...

CISC Processors circa 1999

- Clock Speed: ~400 MHz
- Several million transistors
- 32-bit address space or more
- 32-bit external buses, 128-bit internally
- ~ 100 instructions
- Superscalar CPU
- Judiciously microcoded

- On-chip cache
- Very complex memory hierarchy
- Single-cycle execute of most instructions!
- 32-bit floating-point ALU on board!
- Multimedia extensions
- Harvard architecture (internally)!

- Very expensive (100s of dollars)
- 10s of Watts power consumption

RISC Architectures circa 1999

- The same!!

DSP Microprocessors circa 1999

- Clock speed: 100-200 MHz
- 16-bit (fixed point) or 32-bit (floating point) buses and word sizes
- 16-24 bit address space
- Some on-chip memory
- Single-cycle execution of most instructions

- Harvard architecture
- Lots of special DSP instructions
- 50mW to 2 W power consumption
- Cheap!

Questions

- *If CISC and RISC have adopted all the distinguishing features of early DSP microprocessors and more, why didn't they take over the DSP embedded market, too?*
- Answer: because of the "and more"

- Current high-volume DSP applications (e.g., hard disk drive controllers, cell phones) require low cost, low power
- DSP uPs stripped of all but the most essential features for DSP applications
- Most quoted numbers for DSP uPs not MIPS, but MIPS/$$, MIPS/mW

- Market needs in DSP embedded systems are sufficiently different that no single architectural family can compete in both DSP and general-purpose uP market

Questions

- *Why did RISC become commercial in mid/late ‘80s?*
- *Why did CISC survive?*
- *Why have RISC/CISC converged?*

Myth and Reality in RISC vs CISC

- **Myth:** CISC designs were inferior
- **Reality:** CISC designers made good design tradeoffs at each technology point

- **Myth:** Can’t achieve high performance, single-cycle execution with CISC
- **Reality:** They did it!

- **Myth:** RISC is a better compiler target
- **Reality:** About the same

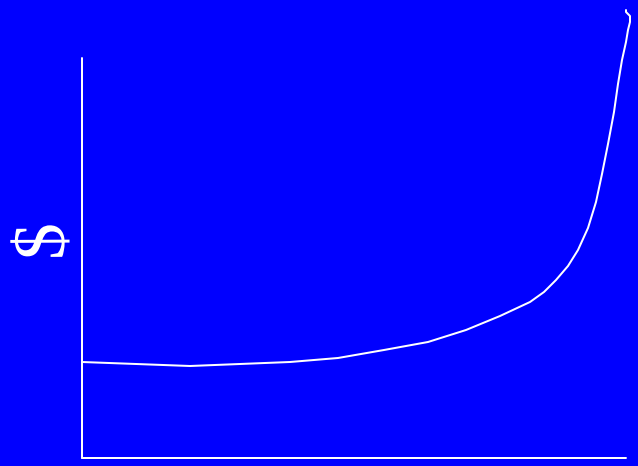
- **Myth:** Commercial RISC chips are really "RISC"
- **Reality:** All commercial RISC chips have a relatively large, complex instruction set

The Real Answer (Opinion!)

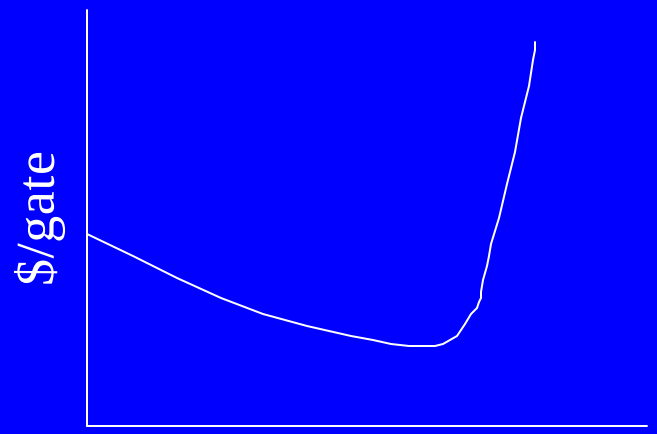
- The main factor driving general-purpose microprocessor design has been the peculiar economics of semiconductor manufacturing

Economics of IC Manufacturing

Cost per chip



Cost per transistor



Implications

- These curves STRONGLY favor designs near the knee of the curve!!!
- All microprocessors of a given generation have roughly the same number of transistors
- Key design tradeoff: what to do with X transistors?

RISC vs. CISC: 500k transistors

- For a few years in the late ‘80s, designers had a choice:
  - CISC CPU and no on-chip cache
  - RISC CPU and on-chip cache
- On-chip cache was probably a slightly better choice, giving RISC 2-3 years of modest advantage

- It wasn’t about RISC at all, it was about the on-chip cache!

RISC vs. CISC: 2M transistors

- Now possible to have both CISC and on-chip cache
- CISC recovers parity, maybe even advantage
- RISC chips become more CISC-like

Even More Transistors ...

- Then more transistors became available than single CISC CPU and reasonable cache could use … what now?
  - Multi-processor chips?
  - Superscalar?
  - VLIW?

Convergence: 5M transistors

- Superscalar won. But …
  - It's really hard to pipeline and schedule superscalar computations when instruction cycles, wordlengths differ, and when there are 100s of different instructions
  - Compilers used only a small subset of instructions
  - Nobody coded in assembly anymore

- This pushed CISC designs to be more RISC-like
- Hence, convergence!

What's Next: 50M transistors

- Approaching limits of superscalability
- Economics of IC manufacturing say more transistors per microprocessor chip
  - VLIW?
  - Symmetric multi-processor on a chip?
  - Heterogeneous multi-processor on a chip?
- The latter is emerging (DSP+ARM, Pentium + MMX)

Question

- *Are there other embedded microprocessor types besides DSP waiting to emerge?*
- AI has been tried and failed
- Video game uPs (“Emotion Engine”)
- Need application area with unique enough requirements that it’s not well served by existing families

- Great opportunity for enterprising chip designer!

Future of DSP Microprocessors

- DSP market and applications remain sufficiently unique to sustain an independent class of microprocessors
- Expect ever-lower-power devices, higher performance within power, cost constraints
- Fixed point will always be with us!
- As will floating point!

- Compilers may finally become useful for embedded DSP development, but user must remain informed, in control when needed
- Hybrid DSP/general purpose uP systems may become the norm