Microprocessor Evolution: 4004 to Pentium Pro

Krste Asanovic
Laboratory for Computer Science
Massachusetts Institute of Technology
First Microprocessor
Intel 4004, 1971

- 4-bit accumulator architecture
- 8\,\mu m pMOS
- 2,300 transistors
- 3 x 4 mm$^2$
- 750kHz clock
- 8-16 cycles/inst.
Microprocessors in the Seventies

Initial target was embedded control
• First micro, 4-bit 4004 from Intel, designed for a desktop printing calculator

Constrained by what could fit on single chip
• Single accumulator architectures

8-bit micros used in hobbyist personal computers
• Micral, Altair, TRS-80, Apple-II

Little impact on conventional computer market until VISICALC spreadsheet for Apple-II (6502, 1MHz)
• First “killer” business application for personal computers
DRAM in the Seventies

Dramatic progress in MOSFET memory technology

1970, Intel introduces first DRAM (1Kbit 1103)

1979, Fujitsu introduces 64Kbit DRAM

=> By mid-Seventies, obvious that PCs would soon have > 64KBytes physical memory
Microprocessor Evolution

Rapid progress in size and speed through 70s
- Fueled by advances in MOSFET technology and expanding markets

Intel i432
- Most ambitious seventies’ micro; started in 1975 - released 1981
- 32-bit capability-based object-oriented architecture
- Instructions variable number of bits long
- Severe performance, complexity, and usability problems

Intel 8086 (1978, 8MHz, 29,000 transistors)
- “Stopgap” 16-bit processor, architected in 10 weeks
- Extended accumulator architecture, assembly-compatible with 8080
- 20-bit addressing through segmented addressing scheme

Motorola 68000 (1979, 8MHz, 68,000 transistors)
- Heavily microcoded (and nanocoded)
- 32-bit general purpose register architecture (24 address pins)
- 8 address registers, 8 data registers
# Intel 8086

<table>
<thead>
<tr>
<th>Class</th>
<th>Register</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data:</td>
<td>AX,BX</td>
<td>“general” purpose</td>
</tr>
<tr>
<td></td>
<td>CX</td>
<td>string and loop ops only</td>
</tr>
<tr>
<td></td>
<td>DX</td>
<td>mult/div and I/O only</td>
</tr>
<tr>
<td>Address:</td>
<td>SP</td>
<td>stack pointer</td>
</tr>
<tr>
<td></td>
<td>BP</td>
<td>base pointer (can also use BX)</td>
</tr>
<tr>
<td></td>
<td>SI,DI</td>
<td>index registers</td>
</tr>
<tr>
<td>Segment:</td>
<td>CS</td>
<td>code segment</td>
</tr>
<tr>
<td></td>
<td>SS</td>
<td>stack segment</td>
</tr>
<tr>
<td></td>
<td>DS</td>
<td>data segment</td>
</tr>
<tr>
<td></td>
<td>ES</td>
<td>extra segment</td>
</tr>
<tr>
<td>Control:</td>
<td>IP</td>
<td>instruction pointer (lower 16 bit of PC)</td>
</tr>
<tr>
<td></td>
<td>FLAGS</td>
<td>C, Z, N, B, P, V and 3 control bits</td>
</tr>
</tbody>
</table>

- Typical format $R \leftarrow R \text{ op } M[X]$, many addressing modes
- Not a GPR organization!
IBM PC, 1981

Hardware
- Team from IBM building PC prototypes in 1979
- Motorola 68000 chosen initially, but 68000 was late
- IBM builds “stopgap” prototypes using 8088 boards from Display Writer word processor
- 8088 is 8-bit bus version of 8086 => allows cheaper system
- Estimated sales of 250,000
- 100,000,000s sold

Software
- Microsoft negotiates to provide OS for IBM. Later buys and modifies QDOS from Seattle Computer Products.

Open System
- Standard processor, Intel 8088
- Standard interfaces
- Standard OS, MS-DOS
- IBM permits cloning and third-party software
The Eighties: Microprocessor Revolution

Personal computer market emerges

- Huge business and consumer market for spreadsheets, word processing and games
- Based on inexpensive 8-bit and 16-bit micros: Zilog Z80, Mostek 6502, Intel 8088/86, …

Minicomputers replaced by workstations

- Distributed network computing and high-performance graphics for scientific and engineering applications (Sun, Apollo, HP,…)
- Based on powerful 32-bit microprocessors with virtual memory, caches, pipelined execution, hardware floating-point

Massively Parallel Processors (MPPs) appear

- Use many cheap micros to approach supercomputer performance (Sequent, Intel, Parsytec)
The Nineties

Distinction between workstation and PC disappears

Parallel microprocessor-based SMPs take over low-end server and supercomputer market

MPPs have limited success in supercomputing market

High-end mainframes and vector supercomputers survive “killer micro” onslaught

64-bit addressing becomes essential at high-end
  • In 2001, 4GB DRAM costs <$5,000

CISC ISA (x86) thrives!
Reduced ISA Diversity in Nineties

Few major companies in general-purpose market
- Intel x86 (CISC)
- IBM 390 (CISC)
- Sun SPARC, SGI MIPS, HP PA-RISC (all RISCs)
- IBM/Apple/Motorola introduce PowerPC (another RISC)
- Digital introduces Alpha (another RISC)

Software costs make ISA change prohibitively expensive
- 64-bit addressing extensions added to RISC instruction sets
- Short vector multimedia extensions added to all ISAs, but without compiler support

=> Focus on microarchitecture (superscalar, out-of-order)

CISC x86 thrives!
- RISCs (SPARC, MIPS, Alpha, PowerPC) fail to make significant inroads into desktop market, but important in server and technical computing markets

“RISC advantage” shrinks with superscalar out-of-order execution
Intel Pentium Pro, (1995)

- During decode, translate complex x86 instructions into RISC-like micro-operations (uops)
  - e.g., “R ← R op Mem” translates into
    - `load T, Mem`  # Load from Mem into temp reg
    - `R ← R op T`  # Operate using value in temp
- Execute uops using speculative out-of-order superscalar engine with register renaming
- Pentium Pro family architecture (P6 family) used on Pentium-II and Pentium-III processors
Intel Pentium Pro (1995)

- External Bus
- L2 Cache

- Bus Interface
- Instruction Cache and Fetch Unit
- Instruction Decoder
- Register Alias Table

- Branch Target Buffer
- Micro-Instruction Sequencer
- Reservation Station

- Memory Reorder Buffer
- Data Cache
- Memory Interface Unit
- Address Generation Unit
- Integer Unit
- Floating-Point Unit
- Reorder Buffer and Retirement Register File

- x86 CISC macro instructions

- Internal RISC-like micro-ops
**P6 Instruction Fetch & Decode**

- 8KB I-cache, 4-way s.a., 32-byte lines, virtual index, physical tag
- I-TLB: 32+4 entry, fully assoc.
- PC from branch predictor

- 16-byte aligned fetch of 16 bytes

- Fetch Buffer (holds x86 insts.)

- Simple Decoder
- Simple Decoder
- Complex Decoder

- 1 uop
- 1 uop
- 1-4 uops

- I-TLB has 32 entries for 4KB pages plus 4 entries for 4MB pages

- uop Buffer (6 entries)
P6 uops

- Each uop has fixed format of around 118 bits
  - opcode, two sources, and destination
  - sources and destination fields are 32-bits wide to hold immediate or operand

- Simple decoders can only handle simple x86 instructions that map to one uop

- Complex decoder can handle x86 translations of up to 4 uops

- Complicated x86 instructions handled by microcode engine that generates uop sequence

- Intel data shows average of 1.2-1.7 uops per x86 instruction on SPEC95 benchmarks, 1.4-2.0 on MS Office applications
P6 Reorder Buffer and Renaming

- **uop Buffer (6 entries)**
- **Allocate ROB, RAT, RS entries**
- **Reorder Buffer (ROB)**
  - Data
  - Status
  - 40 entries in ROB
- **Register Alias Table (RAT)**
  - EAX
  - EBX
  - ECX
  - EDX
  - ESI
  - EDI
  - ESP
  - EBP

Values move from ROB to architectural register file (RRF) when committed.
P6 Reservation Stations and Execution Units

- Reservation Station (20 entries)
  - Renamed uops (3/cycle)
  - Dispatch up to 5 uops/cycle
  - Stores only leave MOB when uop commits

- Memory Reorder Buffer (MOB)
  - 1 store
  - 1 load

- D-TLB
  - 8KB D-cache, 4-way s.a., 32-byte lines, divided into 4 interleaved banks
  - D-TLB has 64 entries for 4KB pages fully assoc., plus 8 entries for 4MB pages, 4-way s.a.

- ROB (40 entries)
  - Load data

- Execution Units
  - Store Data
  - Store Addr.
  - Load Addr.
  - Int. ALU
  - Int. ALU
  - FP ALU
P6 Retirement

• After uop writes back to ROB with no outstanding exceptions or mispredicts, becomes eligible for retirement
• Data written to RRF from ROB
• ROB entry freed, RAT updated
• uops retired in order, up to 3 per cycle
• Have to check and report exceptions at valid x86 instruction fault points
  – complex instructions (e.g., string move) may generate thousands of uops
P6 Branch Penalties

Branch mispredict penalty
P6 Branch Target Buffer (BTB)

- 512 entries, 4-way set-associative
- Holds branch target, plus two-level BHT for taken/not-taken
- Unconditional jumps not held in BTB
- One cycle bubble on correctly predicted taken branches (no penalty if correctly predicted not-taken)
Two-Level Branch Predictor

*Pentium Pro uses the result from the last two branches to select one of the four sets of BHT bits (~90-95% correct)*

- Fetch PC
- 2-bit global branch history shift register
- Shift in Taken/¬Taken results of each branch
- Taken/¬Taken?
P6 Static Branch Prediction

- If a branch misses in BTB, then static prediction performed
- Backwards branch predicted taken, forwards branch predicted not-taken
P6 Branch Penalties

BTB predicted taken penalty

Fetch buffer

uop buffer

Reservation Station

ROB

Branch resolved

Decode and predict branch that missed in BTB (backwards taken, forwards not-taken)
P6 System

PCI Bus

Memory controller

AGP Bus

AGP Graphics Card

Glueless SMP to 4 procs., split-transaction

Frontside bus

DRAM

CPU

L1 I$ L1 D$

CPU

L1 I$ L1 D$

CPU

L1 I$ L1 D$

CPU

L1 I$ L1 D$

L2 $

L2 $

L2 $

L2 $

Backside bus
Pentium-III Die Photo

Programmable Interrupt Control

External and Backside Bus Logic

Packed FP Datapaths

Page Miss Handler

Instruction Fetch Unit:
16KB 4-way s.a. I-cache

Instruction Decoders:
3 x86 insts/cycle

Integer Datapaths

Floating-Point Datapaths

Microinstruction Sequencer

Memory Order Buffer

Memory Interface Unit (convert floats to/from memory format)

MMX Datapaths

Register Alias Table

Allocate entries (ROB, MOB, RS)

Reservation Station

Branch Address Calc

Reorder Buffer (40-entry physical regfile + architect. regfile)
Pentium Pro vs MIPS R10000

Estimates of 30% hit for CISC versus RISC

– compare with original “RISC Advantage” of 2.6

“RISC Advantage” decreased because size of out-of-order core largely independent of original ISA