## Intel Processors

### Processor Specifications

<table>
<thead>
<tr>
<th>Processor</th>
<th>Date</th>
<th>f (MHz)</th>
<th>Trans.</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>4004</td>
<td>4/71</td>
<td>0.108</td>
<td>2300</td>
<td>First µP</td>
</tr>
<tr>
<td>8008</td>
<td>4/72</td>
<td>0.108</td>
<td>3500</td>
<td>First 8-bit µP</td>
</tr>
<tr>
<td>8080</td>
<td>4/74</td>
<td>2</td>
<td>6000</td>
<td>Popular 8-bit</td>
</tr>
<tr>
<td>8086</td>
<td>6/78</td>
<td>5-10</td>
<td>29k</td>
<td>First 16-bit µP; 20-bit addressing</td>
</tr>
<tr>
<td>8088</td>
<td>6/79</td>
<td>5-8</td>
<td>29k</td>
<td>Simpler; IBM PC</td>
</tr>
<tr>
<td>80286</td>
<td>2/82</td>
<td>8-12</td>
<td>134k</td>
<td>Protected mode, 24-bit addressing</td>
</tr>
<tr>
<td>80386</td>
<td>10/85</td>
<td>16-33</td>
<td>275k</td>
<td>32-bit (IA-32)</td>
</tr>
<tr>
<td>80486</td>
<td>4/89</td>
<td>25-100</td>
<td>1.2M</td>
<td>Pipelined (5-stage); cache</td>
</tr>
<tr>
<td>Pentium</td>
<td>3/93</td>
<td>60-233</td>
<td>3.1M</td>
<td>Superscalar, dual pipeline</td>
</tr>
<tr>
<td>PentiumPro</td>
<td>3/95</td>
<td>150-200</td>
<td>5.5M</td>
<td>Out-of-order; L2 cache</td>
</tr>
<tr>
<td>Pentium II</td>
<td>5/97</td>
<td>233-400</td>
<td>7.5M</td>
<td>MMX (SIMD instructions)</td>
</tr>
<tr>
<td>Pentium III</td>
<td>3/99</td>
<td>450-1200</td>
<td>9.5-26M</td>
<td>SSE (incl. SIMD-FP); 10-stage pipeline</td>
</tr>
<tr>
<td>Pentium 4</td>
<td>12/00</td>
<td>1300-2200</td>
<td>42M</td>
<td>SSE2 (128-bit); TC; 20-stage pipeline</td>
</tr>
</tbody>
</table>

### Market Introduction

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4004</td>
<td>4-bit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8008</td>
<td>8-bit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8080</td>
<td>Base registers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8086</td>
<td>32-bit extension (IA-32)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8088</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80286</td>
<td>286</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80386</td>
<td>386</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pentium</td>
<td>486</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pentium Pro</td>
<td>368</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pentium II</td>
<td>385</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pentium III</td>
<td>385</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pentium 4</td>
<td>486</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Current High-End Processors

Source: Microprocessor Report, © Cahners 2006

- = IA-32
- = IA-64
Current IA-32 Processors

Manufacturing Cost

Estimated Manufacturing Cost
(Source: In-Sat)

Current IA-32 Processors

Prices

Source: Microprocessor Report, © Cahners 2006

Outline

- Historical limitations of IA-32
- How some Pentium designs have worked around the main limitations
  - PentiumPro: Achieving superscalar out-of-order execution on a CISC
  - Pentium4: Achieving 2GHz clock frequency

Legacy IA-32 Features

- Very small number of registers, partly dedicated or specialised
- Natively 16-bit, extended to 32 in successive steps requiring backward compatibility (e.g., 3 modes for address generation)
- Highly variable instruction length and encoding (1 to 17 bytes in original IA-32, prefixes, postfixes, etc.)
- CISC instruction set
Registers (I)

- Very small number of general purpose registers (approx. 4 integer plus 8 FP—not shown, versus 32+32 typ. RISC)

**General Registers + PC + Flags**

<table>
<thead>
<tr>
<th>Register</th>
<th>31</th>
<th>15</th>
<th>8</th>
<th>7</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAX</td>
<td>AX</td>
<td>AH</td>
<td>AL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ECX</td>
<td>CK</td>
<td>CH</td>
<td>CL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EDX</td>
<td>DX</td>
<td>DH</td>
<td>DL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EBX</td>
<td>BX</td>
<td>BH</td>
<td>BL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ESP</td>
<td>SP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ESI</td>
<td>SI</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EDI</td>
<td>DI</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EIP</td>
<td>IP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EFLAGS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Segment Registers**
  - CS: Code segment ptr
  - SS: Stack segment ptr
  - DS: Data segment ptr
  - ES: Extra data segm. ptr
  - FS: Data segment ptr 2
  - GS: Data segment ptr 3

- **Accumulator**
- **Count reg**: string, loop
- **Data reg**: multiply, divide
- **Base addr reg**: Base pointer
- **Stack pointer**
- **Base ptr (base stack reg)**
- **Index reg, string src ptr**
- **Index reg, string dest ptr**
- **Instruction ptr (PC)**
- **Condition codes**

Registers (II)

- Small number of registers makes spilling more frequent
- Advanced compiler techniques (e.g., loop unrolling, Lesson 10) increase register pressure
- Partial specialization of the registers makes effective compiler scheduling difficult

Memory Addressing (I)

**Real Mode (8086)**

- Logical address = Segment + Offset
- Physical address = Logical address

Memory Addressing (II)

**Protected Mode (80286)**

- Logical address = Segment + Offset
- Physical address = Logical address + Segment offset
Memory Addressing (III)

- Protected Mode (80386, 80486, and Pentium)

Addressing Modes (I)

- Absolute
- Register indirect → [reg]
  - 16-bit registers: BX, SI, DI
  - 32-bit registers: EAX, ECX, EDX, EBX, ESI, EDI
- Displacement → [reg + displacement]
  - 16-bit registers: BP, BX, SI, DI
  - 32-bit registers: EAX, ECX, EDX, EBX, ESI, EDI
  - Displacement on 8, 16, or 32 bits
- Indexed → [base reg + reg]
  - 16-bit registers: BX+SI, BX+DI, BP+SI, BP+DI

Addressing Modes (II)

- Indexed with displacement
  → [base reg + reg + displacement]
  - Same registers as in mode indexed
- Scaled indexed → [base reg + 2^scale x reg]
  - Only in 32-bit mode
  - Scale is 0, 1, 2, or 3
  - Index register can be any of the basic registers (except ESP)
  - Base register can be any of the basic registers
- Scaled indexed with displacement
  → [base reg + 2^scale x reg + displacement]

Address Segment

- For every indirect addressing (e.g., [reg]) the appropriate segment would be needed
- Default:
  - References to instructions (IP) use CS (code segment register)
  - References to stack (BP or SP) use SS (stack segment register)
  - All other references use DS (data segment register)
- A one-byte instruction prefix can modify the default
Instructions—IA-32 is not the same architecture since the mid-80’s

- Classic CISC set derived from extended accumulator architecture
- Improved orthogonality in the 32-bit extensions (80386)
- Added FP capabilities previously on a coprocessor (80486)
- Added MultiMedia Extensions MMX as SIMD (single-instruction multiple-data) integer instructions (Pentium II)
- Added Streaming SIMD Extension SSE, most notably consisting of SIMD FP instructions (Pentium III)
- Added SSE2, essentially extension of MMX+SSE to 128 bits (Pentium 4)

Operand Types

- Not a Load/Store architecture

<table>
<thead>
<tr>
<th>Source 1 = Destination</th>
<th>Source 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register</td>
<td>Register</td>
</tr>
<tr>
<td>Register</td>
<td>Immediate</td>
</tr>
<tr>
<td>Register</td>
<td>Memory</td>
</tr>
<tr>
<td>Memory</td>
<td>Register</td>
</tr>
<tr>
<td>Memory</td>
<td>Immediate</td>
</tr>
</tbody>
</table>

Immediate values can be on 8, 16, or 32 bits

Instruction Examples

JE addr
if equal(CC) then IP ← addr (IP-128 ≤ addr ≤ IP+128)

J MP addr
IP ← addr

CALL addr,seg
SP ← SP-2; Mem[SS:SP] ← IP+5
SP ← SP-2; Mem[SS:SP] ← CS
IP ← addr; CS ← seg

MOVW BX,[DI +45]
BX ← Mem[DS:DI+45]

PUSH SI
SP ← SP-2
Mem[SS:SP] ← SI

POP DI
DI ← Mem[SS:SP]
SP ← SP+2

ADD AX,#6765
AX ← AX+6765

TEST DX,#42
set CC flags with (DX and 42)

MOVSB
DI ← DI+1
SI ← SI+1

Instruction Encoding

- One instruction coded on 1 to 17 bytes in original IA-32
- Several types of modifiers/prefixes
- Two combinations of constants of variable length
  - Immediate and Displacement
  - 8, 16, and 32-bit
- Opcode “lost” and only moderately orthogonal
### Examples of Instruction Encoding

- **JE PC+3657**
  - `JE` cond displ

- **CALLF**
  - `CALLF` offset segment number

- **MOV BX,[DI+45]**
  - `MOV` r postbyte displ

- **PUSH SI**
  - `PUSH` r

- **ADD AX,#6765**
  - `ADD` reg w constant

- **SHL BX,1**
  - `SHL` w

- **TEST DX,#42**
  - `TEST` w postbyte immediate

### 1995: PentiumPro (P6)
**A Superscalar IA-32 CISC?**

- How to adapt the superscalar ideas to fit such an irregular architecture?
  - Complexity of decoding is huge
  - Parallel decoding of instructions is tough due to an encoding strongly variable in length
  - Instructions mix memory operations with computations
  - Too few registers

### PentiumPro Microarchitecture: Out-of-order CISC Execution

- Decoder from IA-32 to P6 uops: A “CISC-to-RISC converter”...

- Classic RISC out-of-order engine

### PentiumPro In-Order Section

- Converts every IA-32 instruction into one or more internal RISC-like 118-bit instructions (micro-operations or uops); on average 1 instruction = 1.5-2.0 uops
- Three decoders and a sequencer work in parallel to perform the conversion
  - Two highest priority simple decoders intercept register-register operations (1 instr. → 1 uop)
  - A low priority general decoder handles all other basic operations (1 instr. → 4 uops)
  - A sequencer is used by the general decoder for very complex operations (1 instr. → several groups of 4 uops)
- Reorder Buffer (ROB) implements renaming and commits uops in program order to the Real Register File (RRF)
PentiumPro Out-of-order Section

- Superscalar very similar to the general model studied in previous lessons
- Up to 20 uops wait in the Reservation Stations until the operands are all available
- A maximum of 5 uops can be issued per cycle: a generic calculation (int or FP), a simple integer (no shift, mul, nor div), a load, a store address, and a store data
- A Memory Reorder Buffer (MOB) reorders memory accesses and waits for D-cache availability

PentiumPro Package

- L2 cache
- Processor

2000: Pentium 4
Processors for the Multi-GHz Era

- Isn’t Pentium dead in favour of IA-64 and Itanium? Clearly not...
- How to modify Pentium III to achieve way less than 1ns of cycle time?
  - Pipeline expansion? (Pentium III: 10 stages)
  - Wire propagation time becomes very tangible compared to computation
  - uop decoding very heavy
Evolution of Pentium Pipeline: From 5 to 20 Stages

- Pentium: Prefetch, Decode, Execute, Write-back
- Pentium Pro: Fetch, Decode, Decode, Rename, ROB, Id/Sld, Dispatch, Execute
- Pentium 4: TC Tcl IP, TC Fetch, Drive, Alloc, Rename, Queue, Schedule

First microarchitecture with dedicated stages for wire delays: key to very high frequencies

Some Pentium 4 Characteristics

- **Trace caches** to memorize approx. 12,000 recent uops—sort of L0 cache to avoid IA-32 instruction decoding from the main loop
- Approx. 126 uops can be in-flight at one time (3 times more than Pentium III)
- **Data speculation** to execute a potentially dependent load before a store: if the dependence was real, the load is squashed and replayed
- P4 ALUs can perform simple operations in half clock-cycle, to sustain throughput
  - Two dependent operations can be scheduled in the same cycle

AMD Hammer x86-64: Extension of IA-32 to 64 bits (I)

- Extension of IA-32 to 64 bits (I)
- Legacy Byte Order for Instructions
- x86, 64 Instruction Byte Order
- New x86-64 Instruction Byte Order

AMD Hammer x86-64: Extension of IA-32 to 64 bits (II)

- 64K, 3-way, Predecoded Level 1 Instruction Cache
Conclusions

- IA-32 is the oldest important ISA around
- It is not absolutely fixed but constantly evolving with many new add-ons (MMX, SSE, SSE2, CMOV, etc.)
- Intel has managed to continue pushing the performance by adapting to its CISC nature the techniques developed to speed-up newer RISC processors
  - Similar work has been done by some competitors—notably by AMD with Athlon, for a few months the fastest IA-32 processor on the market
- Is IA-32 really dying? Is it evolving toward 64 bits?!

References and Where to Learn More

- References:
  - COD, Sections 3.12, 4.9, 5.7, 6.9, and 7.6
- Where to learn more:
  - P. Glaskowsky, Pentium 4 (Partially) Previewed, Microprocessor Report, 28th August 2000
  - S. Leibson, AMD Drops 64-bit Hammer on x86, Microprocessor Report, 4th September 2000

All papers available at http://iap.epfl.ch/courses/archord2/