Intel Micro-Architecture (Haswell i7)

Goal: Fast overview of one of Intel's main processors

Highlights:
- Superscalar
- Speculative Execution
- Register Renaming
- 14-deep pipeline

A bit of x86 instruction set

More details:
- Class textbook
- Ars Technica website

X86 Milestones – Evolution of the instruction set

Some relevant steps (not all):
- 1974: 8080 8-bit, 2MHz, 6k transistors
- 1978: 8086 16-bit, 5-10MHz, 29k transistors
- 1980: 8087 floating point coprocessor
- 1982: 80286 16-bit, 6-12.5MHz, 134k transistors, 24-bit address space
- 1985: 80386 32-bit, 16-33MHz, 256k transistors, 256B code cache
- 1989: 80486 32-bit, 25MHz, 1.2M transistors, 8KB L1, 5-stage pipe
- 1992: Pentium 32-bit, 60-66MHz, 3.3M transistors, 16KB L1, L2, branch predict, superscalar (CPI=0.5).
- 1995: Pentium Pro, 32-bit, 200MHz, 5.5M transistors, CPI=1/3, 12-stage pipeline, out-of-order execute, predicated instructions, 4-bit branch history.
- 1996: Pentium MMX, 150-233MHz, 4.5M transistors, SIMD (single instruction multiple data) instructions.
- 2000: Pentium 4, 1.3-3.0GHz, 42M transistors, 20-deep pipeline, symmetric multithreading
- 2006: Core 2 Duo, 64-bit, 1.0-2.3GHz, 291M transistors, 14-stage pipeline, multi-core
- 2008: Nehalem/i7, 1.73-3.46 GHz, 2.6B transistors, quad/octo-core, SMT, shared L3
- 2013: Haswell/i7 ...

Backwards compatible
X86 Operands

16x64-bit registers plus special-purpose registers (Flag, segments, etc).
2-operand instructions:

<table>
<thead>
<tr>
<th>Source/Destination operand Type</th>
<th>Second source operand</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>

Multiple data memory addressing modes:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Indirect</td>
<td>Mem[Reg[$id]]</td>
</tr>
<tr>
<td>Base + 8/32/64-bit displace</td>
<td>Mem[Reg[$id]+displace]</td>
</tr>
<tr>
<td>Base + scaled index</td>
<td>Mem[Reg[$id]+Reg[$id2]*2^{scale}], scale=0..3</td>
</tr>
<tr>
<td>Base + scaled + 8/32/64 displace</td>
<td>Mem[Reg[$id]+Reg[$id2]*2^{scale}+displace]</td>
</tr>
</tbody>
</table>

Instructions

Data movement: Move, push, pop
Arithmetic & logic: test, integer, decimal math, etc.
Control Flow: branches, jumps, calls, returns
  Condition codes
    Instructions set flags in a register (zero, negative, etc)
    Branches test flags from previous instruction
String instructions: string move, compare (legacy from 8080, not much used)

Streaming SIMD (MMX, SSE)
  Single instruction, multiple data (i.e. 4x8-bit adds simultaneously)
  Intended for multi-media
**Instruction Encoding**

Range from 1-byte to 17-byte!
- Opcode says bitwidth of 8-bit/32-bit...
- May have extra byte to indicate addressing mode
- Extra byte for scaled index mode.

<table>
<thead>
<tr>
<th>Stack Operation (PUSH)</th>
<th>Branch (JE)</th>
<th>SW/LW (MOV)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>JALR (CALL)</th>
<th>ADDI (ADD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALL [Offset]</td>
<td>ADD [AddMode] Immediate</td>
</tr>
</tbody>
</table>

Set condition code with AND (TEST)

<table>
<thead>
<tr>
<th>TEST [AddMode] Immediate</th>
</tr>
</thead>
</table>

**Instruction Decoding**

X86 instructions are essentially pseudo-instructions, converted to multiple RISC-like micro-ops

CPU decodes X86 into micro-ops at runtime

![Diagram](image-url)
Instruction Scheduling

Haswell

56 µop Decode Queue

192 Entry Reorder Buffer (ROB)

168 Integer Registers

168 AVX Registers

48 Entry Branch Order Buffer

72 Entry Load Buffer

42 Entry Store Buffer

60 Entry Unified Scheduler

Picture from David Kanter, RealWorldTech.com

ALUs

60-entry Unified Reservation Station

Integer ALU/Shift

Integer ALU/UELA

Load Store Address

Select Store Data

Integer ALU/Shift

Integer ALU/Shift

Integer ALU/Shift

Branch

Fused Div/FP Mul

Fused Div/FP Add

Vector Int ALU

Vector Int ALU

Vector Logical

Vector Logical

Vector Shifts

Divide

Branch

Picture from Anand Lal Shimpi, anandtech.com
Parallelism

CPU is 4-way superscalar, ~14 pipeline stages (P4 had 20!)
Superscalar picks from 96-Instruction window.
Register renaming to 168 registers.
Each chip has 4 cores
Symmetric Multithreading (2-way per core)
On-chip GPU

Cache Organization

All 64 byte block, write-back.

(Per core) Split L1 Caches
32KB, 8-way Set Associative Instruction Cache
Can fetch 16B/cycle.
32KB, 8-way Set Associative Data Cache
4-cycle latency. 64B/cycle loads, 32B/cycle stores.

(Per core) L2 Unified Cache
256KB, 8-way Set Associative
12-cycle latency

(Per-chip) L3 Unified Cache
6-8MB, ?? Set Associative
36-cycle latency

On-package L4 for some versions w/high-performance GPU.
14-Stage Pipeline

(Nehalem Diagram, Haswell similar)

Multiple Members of Single CPU Family

Picture from Hiroshige Goto
Layout (Annotated)

Picture from Hiroshige Goto

Intel Haswell Wafer

Haswell die w/pin as a reference

Picture from Intel Free Press