Chapter 2

Other Instruction Sets

The Intel x86 ISA

- Evolution with backward compatibility
- 4004 (1971): Intel invents the microprocessor
- 8080 (1974): 8-bit microprocessor
  - Accumulator, plus 3 index-register pairs
- 8086 (1978): 16-bit extension to 8080
  - Complex instruction set (CISC)
- 8087 (1980): floating-point coprocessor
  - Adds 60 floating point instructions, 80-bit FP registers
- 80286 (1982): 24-bit addresses, MMU
  - Segmented memory mapping and protection
- 80386 (1985): 32-bit extension (now IA-32)
  - Additional addressing modes and operations
  - Paged memory mapping as well as segments
### The Intel x86 ISA

- **Further evolution…**
  - **i486 (1989):** pipelined, on-chip caches and FPU
    - Compatible competitors: AMD, Cyrix, …
  - **Pentium (1993):** superscalar, 64-bit datapath
    - Later versions added MMX (Multi-Media eXtension) instructions
    - The infamous FDIV bug (read text book)
  - **Pentium Pro (1995), Pentium II (1997):**
    - New microarchitecture (see Colwell - *The Pentium Chronicles*)
    - 57 new instructions
    - Primarily for multimedia applications (SIMD)
  - **Pentium III (1999):**
    - Added SSE (Streaming SIMD Extensions) and associated registers
    - Four 32-bit floating point operations in parallel
    - Useful in speech recognition, video encoding/decoding

- **And further…**
  - **Pentium 4 (2001) – Itanium:**
    - New microarchitecture - IA-64 (RISC-like) architecture
    - Added SSE2 instructions
      - Explicitly Parallel Instruction Computing (EPIC)
    - 128-bit bundle with three instructions and a template
    - 128 general purpose registers and 128 floating point registers
    - Done by a partnership between HP and Intel
    - Able to run both UNIX and Microsoft Windows
  - **AMD64 (2003):** extended architecture to 64 bits
    - Increases address space to 64 bits
    - Widens all registers to 64 bits
    - Other changes (AMD64)
  - **EM64T – Extended Memory 64 Technology (2004):**
    - AMD64 adopted by Intel (with refinements)
    - Added SSE3 instructions
The Intel x86 ISA

- Further yet …
  - Intel Core (2006)
    - Added SSE4 instructions, virtual machine support
  - AMD64 (announced 2007): SSE5 instructions
    - Intel declined to follow, instead…
  - Advanced Vector Extension (announced 2008)
    - Longer SSE registers, more instructions
- If Intel didn’t extend with compatibility, its competitors would!
- Technical elegance ≠ market success

The Intel x86 ISA Conclusions

- Conclusion (Hennessy and Patterson):
  - “This history illustrates the impact of the “golden handcuffs” of compatibility”.
  - “Adding new features as someone might add clothing to a packed bag”.
  - “An architecture that is difficult to explain and impossible to love”.
### Implementing IA-32

- Complex instruction set makes implementation difficult:
  - Hardware translates instructions to simpler micro-operations:
    - Simple instructions: 1 – 1.
    - Complex instructions: 1 – many.
  - Microengine similar to RISC.
- Comparable performance to RISC:
  - Compilers avoid complex instructions.

### ARM & MIPS Similarities

- ARM: the most popular embedded core.
- Similar basic set of instructions to MIPS.

<table>
<thead>
<tr>
<th></th>
<th>ARM</th>
<th>MIPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date announced</td>
<td>1985</td>
<td>1985</td>
</tr>
<tr>
<td>Instruction size</td>
<td>32 bits</td>
<td>32 bits</td>
</tr>
<tr>
<td>Address space</td>
<td>32-bit flat</td>
<td>32-bit flat</td>
</tr>
<tr>
<td>Data alignment</td>
<td>Aligned</td>
<td>Aligned</td>
</tr>
<tr>
<td>Data addressing modes</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>Registers</td>
<td>15 × 32-bit</td>
<td>31 × 32-bit</td>
</tr>
<tr>
<td>Input/output</td>
<td>Memory mapped</td>
<td>Memory mapped</td>
</tr>
</tbody>
</table>
ARM v8 Instructions

- In moving to 64-bit, ARM did a complete overhaul.
- ARM changes from v7 (v8 resembles MIPS):
  - No conditional execution field;
  - Immediate field is a 12-bit constant;
  - Dropped load/store multiple;
  - PC is no longer a General Purpose Register (GPR);
  - GPR set expanded to 32;
  - Addressing modes work for all word sizes;
  - Divide instruction;
  - Branch if equal/branch if not equal instructions.

Fallacies

- Backward compatibility \( \Rightarrow \) instruction set doesn’t change
  - But they do accrete more instructions.
### Top 10 80x86 Instructions

<table>
<thead>
<tr>
<th>Rank</th>
<th>Instruction</th>
<th>Integer Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>load</td>
<td>22%</td>
</tr>
<tr>
<td>2</td>
<td>conditional branch</td>
<td>20%</td>
</tr>
<tr>
<td>3</td>
<td>compare</td>
<td>16%</td>
</tr>
<tr>
<td>4</td>
<td>store</td>
<td>12%</td>
</tr>
<tr>
<td>5</td>
<td>add</td>
<td>8%</td>
</tr>
<tr>
<td>6</td>
<td>and</td>
<td>6%</td>
</tr>
<tr>
<td>7</td>
<td>sub</td>
<td>5%</td>
</tr>
<tr>
<td>8</td>
<td>move register-register</td>
<td>4%</td>
</tr>
<tr>
<td>9</td>
<td>call</td>
<td>1%</td>
</tr>
<tr>
<td>10</td>
<td>return</td>
<td>1%</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>96%</td>
</tr>
</tbody>
</table>

### 80x86 Instruction Length Distribution
Fallacies

- Powerful instruction $\Rightarrow$ higher performance
  - Fewer instructions required.
  - But complex instructions are hard to implement
    - May slow down all instructions, including simple ones.
  - Compilers are good at making fast code from simple instructions.
- Use assembly code for high performance
  - Modern compilers are better at dealing with modern processors.
  - More lines of code $\Rightarrow$ more errors and less productivity.

Concluding Remarks

- Design principles:
  1. Simplicity favors regularity;
  2. Smaller is faster;
  3. Make the common case fast;
- MIPS: typical of RISC ISAs