6 VLIW Architectures

- Very Long Instruction Word
- EPIC: Explicit Parallel Instruction-set Computer
- CISC $\rightarrow$ RISC $\rightarrow$ EPIC
- 6.1 Basic principles
- 6.2 Overview of proposed and commercial VLIW architectures
- 6.3 Case study: The Trace 200 family

Basic Structure of VLIW Architecture

Basic principle of VLIW

- Controlled by very long instruction words
  - comprising a control field for each of the execution units
- Length of instruction depends on
  - the number of execution units (5-30 EU)
  - the code lengths required for controlling each EU (16-32 bits)
- Disadvantages: on average only some of the control fields will actually be used
  - waste memory space and memory bandwidth
  - e.g. Fortran code is 3 times larger for VLIW (Trace processor)

Instruction word format: Trace 7/200

VLIW: Static Scheduling of instructions/

- Instruction scheduling done entirely by [software] compiler
- Lesser Hardware complexity translate to
  - increase the clock rate
  - raise the degree of parallelism (more EU); [Can this be utilized?]
- Higher Software (compiler) complexity
  - compiler needs to aware hardware detail
    - number of EU, their latencies, repetition rates, memory load-use delay, and so on
    - cache misses: compiler has to take into account worst-case delay value
  - this hardware dependency restricts the use of the same compiler for a family of VLIW processors
6.3 Case study: Trace 200

- 256-bit VLIW words
- Capable of executing 7 instructions/cycle
  - 4 integer operations
  - 2 FP
  - 1 Conditional branch
- Found that every 5th to 8th operation on average is a conditional branch
- Use sophisticated branching scheme: multi-way branching capability
  - executing multi-paths
  - assign priority code corresponds to its relative order

Trace 7/200

- 256-bit VLIW words
- Capable of executing 7 instructions/cycle
  - 4 integer operations
  - 2 FP
  - 1 Conditional branch
- Found that every 5th to 8th operation on average is a conditional branch
- Use sophisticated branching scheme: multi-way branching capability
  - executing multi-paths
  - assign priority code corresponds to its relative order

Trace 28/200: storing long instructions

- 1024 bit per instruction
- A number of 32-bit fields maybe empty
- Storing scheme to save space
  - 32-bit mask indicating each sub-field is empty or not
  - Followed by all sub-fields that are not empty
  - Resulting still 3 time larger memory space required to store Fortran code (vs. VAX object code)
  - Very complex hardware for cache fill and refill
- Performance data is Impressive indeed!

Trace: Performance data

<table>
<thead>
<tr>
<th>Trace</th>
<th>Instruction issue rate (MIPS)</th>
<th>Compiled Linpack (full precision) (MFLOPS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace 7200</td>
<td>130</td>
<td>6</td>
</tr>
<tr>
<td>Trace 14200</td>
<td>130</td>
<td>10</td>
</tr>
<tr>
<td>DEC 3700</td>
<td>45</td>
<td>0.97</td>
</tr>
<tr>
<td>Crusoe</td>
<td>8</td>
<td>24</td>
</tr>
</tbody>
</table>

Transmeta: Crusoe

- Full x86-compatible: by dynamic code translation
- High performance: 700MHz in mobile platforms

Crusoe

- "Remarkably low power consumption"
  - "A full day of web browsing on a single battery charge"
  - (333–400MHz TM3120 in production now, running Linux! TM5400 runs Windows, production mid 2000)
Intel: Itanium (VLIW) {EPIC} Processor

- 895 MHz production frequency
  - EPIC enables up to 20 operations per clock
- 4 MB high speed on-cartridge L3 cache
- Over 200M transistors
  - 25M in CPU, 200M in L3 cache
- 2.1 Gb/s multi-drop system bus
  - Enhanced Defer Mechanism enables high scalability through improved bus efficiency
- Extensive reliability and availability
  - ECC, parity protection, enhanced MCA
- Excellent functionality on initial silicon
  - No architectural or ISA changes required
  - MP functionality on track to plan

Tuning / testing for production this year

Copyright Intel