Motivation

- Instruction-level parallelism improves performance by increasing the number of instructions per cycle.
ILDP – Instruction-Level Distributed Processing – emphasizing interinstruction communication with dynamic optimization and a tight interaction between hardware and low-level software
Substantial shifts in hardware technology and software applications will lead to general-purpose microarchitectures composed of small, simple, interconnected processing elements, running at very high clock frequencies.

A hidden layer of implementation-specific software -- co-designed with the hardware -- will help manage the distributed hardware resources.
During the next two or three technology generations, processor architects will face several major challenges:

- **Wire delays**
  - Short (local) and long (global) wires

- **Power consumption**
  - Dynamic power: $P = fCV^2$
  - Static power
Technology shifts (cont)

- Software issues
  - In practice many application binaries are not highly optimized
- On-chip multithreading
ILDP microarchitecture

Figure 1. An example of instruction-level distributed processing microarchitecture, consisting of an instruction fetch (IF) unit, integer units (IU), floating-point units (FU), and cache units (CU). The units communicate via point-to-point interconnections that will likely consume one or more clock cycles each.
ILDP microarchitecture

A processor consists of several distributed functional units, each fairly simple and with a very high frequency clock.

A significant part of the microarchitecture design effort will involve partitioning the processor to accommodate delays.
Clock speed

- Clock speed holds the key to increased performance
- Intel processor evolution
  - Original Pentium: 5 stages
  - Pentium Pro/II/III: 12 stages
  - Pentium 4: 20 stages
Dependence-based microarchitecture

Figure 2. Alpha 21264 clustered microarchitecture. Instructions are steered to one of two processing clusters that have duplicated register files. Communicating the results produced in one cluster to the other cluster takes a full clock cycle.
Heterogeneous ILDP

Figure 3. A heterogeneous instruction-level distributed processing chip architecture. A simple processor core is surrounded by helper engines that perform speculative tasks off the critical path and enhance overall performance.
Co-designed virtual machines

Figure 4. Supporting instruction-level distributed processing (ILDP) with a co-designed virtual machine. The virtual-machine monitor (VMM) resides in hidden memory and uses monitoring and configuration hardware to manage ILDP resources. Control can be transferred to the VMM code at any time. Dashed lines show the control transfers.
### Instruction set

**Table 1. Basic instruction types for an accumulator-based ISA.**

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Size (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R &lt;- A</td>
<td>1</td>
</tr>
<tr>
<td>A &lt;- R</td>
<td>1</td>
</tr>
<tr>
<td>A &lt;- A op R</td>
<td>2</td>
</tr>
<tr>
<td>A &lt;- A op imm</td>
<td>4</td>
</tr>
<tr>
<td>A &lt;- M(R op imm)</td>
<td>4</td>
</tr>
<tr>
<td>M(R op imm) &lt;- A</td>
<td>4</td>
</tr>
<tr>
<td>R &lt;- M(A op imm)</td>
<td>4</td>
</tr>
</tbody>
</table>

A: single accumulator

R: general-purpose register
Conclusion

- New applications and evolving hardware technology are constantly changing the mix of techniques that lead to optimal engineering solutions.

Future work

- How should developers divide functions between hardware and VMM software to optimize performance or power saving?
- Many other important and interesting research