Trimaran
History and Motivation
## Terminology

**EPIC: Explicitly Parallel Instruction Computing**

<table>
<thead>
<tr>
<th>Architectural philosophy and technology:</th>
<th>RISC</th>
<th>EPIC</th>
<th>EPIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific architecture and ISA:</td>
<td>PA-RISC</td>
<td>HPL-PD</td>
<td>IA-64</td>
</tr>
<tr>
<td>Implementation:</td>
<td>PA-8500</td>
<td>–</td>
<td>Merced™</td>
</tr>
</tbody>
</table>
The Motivation for EPIC

- In 1989, we at HPL believed that within the next 10 years:
  - a high ILP processor would fit on a chip
  - superscalar complexity would be an obstacle to sustaining Moore's Law
- Achieve high levels of ILP
  - the ability to issue over eight useful operations per cycle
- Retain hardware simplicity and short cycle times even at high levels of ILP
  - avoid schemes that force hardware to make complex decisions at run-time
- True general-purpose capability
  - "scientific" computations as well as "scalar" computations, i.e., code with a high frequency of conditional branches and pointer-based memory accesses
The EPIC philosophy

- Provide the facility to design the desired record of execution (ROE) at compile-time
  - Generalize VLIW's philosophy of compile-time scheduling and resource allocation: which operations? what time? which resources? which registers?
  - Features that provide greater program (compiler) control over microarchitectural capabilities
  - Features that assist in reducing the critical path through "scalar" computations
  - Features that permit one to "play the statistics"

- Provide the ability to communicate the desired ROE to the hardware
  - Maintain run-time transparency, i.e., "obedient" hardware
  - MultiOp, adequate architectural registers, rotating registers, non-unit assumed latencies (NUAL)

- Provide the ability to freeze virtual time during execution in response to unexpected dynamic events
# Key features of HPL-PD

<table>
<thead>
<tr>
<th>Features</th>
<th>Design Record of Execution</th>
<th>Communicate Record of Execution</th>
</tr>
</thead>
<tbody>
<tr>
<td>MultiOp</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Non-unit assumed latencies (NUAL), ELRs, latency stalling</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Predication</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Compare-to-predicate</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Control speculative opcodes / exception tags</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Data speculation</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Prepared branches</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Long latency branches</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Branch prediction control</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Parallel multi-way branching</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Software pipelining branches</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Rotating registers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cache latency control</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Cache hierarchy promotion control</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>
New challenges in EPIC compilation

- Designing the desired ROE, exploiting the features of EPIC
- Managing the cache hierarchy
- The figure of merit is the schedule length, not the number of operations executed
  - Reduce the length of the critical path through the computation
  - Often, the critical path can be shortened by increasing the number of operations executed
- Statistical analysis, optimization and transformation
- Analysis of predicated code, i.e., code without a control flow graph
- Region-based compilation
- Machine description-driven ILP compilation
The Genesis of Trimaran

• Joint research partnership with the University of Illinois’ IMPACT project [1991]

• Development of Elcor [Nov. 1993]

• Leveraging of the IMPACT compiler

• Injection of compiler ideas into IMPACT

• HPL-PD architecture specification published [Feb. 1994]

• The ReaCT-ILP project at NYU proposes the Trimaran project [Feb. 1996]

• Trimaran released [Aug. 1998]
  – HP Labs
  – The University of Illinois
  – New York University
This is a point of discontinuity

- EPIC represents a new philosophy of computing
  - Explicit parallelism
  - Unprecedented programmatic control over the resources of the machine
  - Architectural features that help in engineering the desired record-of-execution and in communicating it to the processor
  - The first architectural style to consciously focus on the reduction of the critical path through the computation
  - Capable of achieving high levels of ILP on a wide spectrum of applications

- Sophisticated architectures require sophisticated usage
  - EPIC uses advanced architectural features to exploit increasingly specialized properties of the workload
  - Sophisticated compilers are crucial for the effective use of EPIC
  - Trimaran and HPL-PD provide the ability to do EPIC compiler research