Andes Company Overview

Driving Innovations™
### Andes: Converting 10+ Years Investments into RISC-V Community

<table>
<thead>
<tr>
<th>Category</th>
<th>Details</th>
</tr>
</thead>
</table>
| 150+ commercial licensees                                               | • 80%+ in RISC V in ‘18  
• AIoT, Storage, Networking                                             |
| 150 people; 80% in R&D                                                  | • R&D center in Taiwan and (now) US                                     |
| Found in 2005; Public since 2017                                        | • MIPS-like 32-/64-bit RISC & DSP  
• Complete ecosystem                                                      |
| Founding member and Major Contributor to RISC-V                         | • Tool chain contribution  
• ISA extensions                                                             |
Evolution of Computing Acceleration

- Specialized peripherals
  - I/O controller, crypto engines

- Co-processors
  - Floating point (FPU)

- ISA expansion
  - Intel’s MMX, SSE, AVX, etc.

- User extensions
  - Old: ARC, Tensilica
  - New: RISC-V open opcode
# Comparison of Acceleration Methods

<table>
<thead>
<tr>
<th></th>
<th>Peripheral</th>
<th>Co-Processor</th>
<th>ISA Expansion</th>
<th>User Extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start-up latency</td>
<td>Longest</td>
<td>Long</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Resource sharing</td>
<td>None</td>
<td>Decode/Control</td>
<td>Control &amp; RF</td>
<td>Control &amp; RF</td>
</tr>
<tr>
<td>Implementation Freedom</td>
<td>A TON</td>
<td>Lots</td>
<td>None</td>
<td>Restricted</td>
</tr>
<tr>
<td>Proprietary advantage</td>
<td>Yes</td>
<td>Yes</td>
<td>None</td>
<td>Yes</td>
</tr>
<tr>
<td>Best for</td>
<td>Very heavy</td>
<td>Medium</td>
<td>Comoditized</td>
<td>Low-Medium</td>
</tr>
<tr>
<td></td>
<td>Semantic</td>
<td>Semantic</td>
<td>Computation</td>
<td>Semantic</td>
</tr>
</tbody>
</table>

**Medium Semantic Acceleration**

Long start-up Latency

---

Confidential
RISC-V ROCC Interface for ISA Extension

- Loosely coupled transaction model
  - Command queues starts in order
  - Response not guaranteed to be in-order
- Memory model same as baseline ISA
- Ideal for light-semantic operations
  - Custom ALU/data format
  - Dependent on host for resource/schedule
Andes Custom Extension (ACE) Illustrated

- RISC-V compliant 25-bit
- ACR can be arbitrarily wide
- Multiple ACMs
- Automated RISC-V to ACE signals
## Summary of ACE Capabilities

<table>
<thead>
<tr>
<th>Items</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Instructions</strong></td>
<td><strong>Items</strong></td>
</tr>
<tr>
<td>scalar</td>
<td>single-cycle, or multi-cycle</td>
</tr>
<tr>
<td>vector</td>
<td>for loop, or do-while loop</td>
</tr>
<tr>
<td>background option</td>
<td>retire immediately, and continue execution in the background. Applicable to scalar and vector.</td>
</tr>
<tr>
<td><strong>Operands</strong></td>
<td><strong>standard</strong></td>
</tr>
<tr>
<td></td>
<td><strong>custom</strong></td>
</tr>
<tr>
<td></td>
<td><strong>- Arbitrary width and number</strong></td>
</tr>
</tbody>
</table>
ACE and COPILOT Environment

Verilog user.v
Script user.ace

Concise RTL
Semantics, operands, test-case spec

Automated Env. For Cross Checking
Test Case Generator

Extended RTL
Extended ISS

COPILOT™
Custom-OPtimized Instruction deveLOpment Tools

Extended Tools
Extended ISS
Extended RTL

Compiler Asm/Disasm Debugger
CPU ISS (near-cycle accurate)
CPU RTL

Extensible Baseline Components

Executable or library
Source file

Concise RTL
Semantics, operands, test-case spec

Test Case Generator

Extended RTL
Extended ISS

COPILOT™
Custom-OPtimized Instruction deveLOpment Tools

Extended Tools
Extended ISS
Extended RTL

Compiler Asm/Disasm Debugger
CPU ISS (near-cycle accurate)
CPU RTL

Extensible Baseline Components

Executable or library
Source file
## ACE: More Capability, MUCH More Automated

<table>
<thead>
<tr>
<th>Items</th>
<th>ROCC</th>
<th>Andes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capability Low-level process signals availability</td>
<td>Yes</td>
<td>Yes ++</td>
</tr>
<tr>
<td>Assignment of opcode &amp; operands</td>
<td>Manually assigned</td>
<td>Auto assigned</td>
</tr>
<tr>
<td>RTL to decode the instructions</td>
<td>Manually developed</td>
<td>Auto-generated</td>
</tr>
<tr>
<td>RTL to check instruction dependences</td>
<td>Manually developed</td>
<td>Auto-generated</td>
</tr>
<tr>
<td>Function verification tools, patterns and flow</td>
<td>Manually developed</td>
<td>Auto-generated</td>
</tr>
<tr>
<td>Compiler/debugger for the new instructions</td>
<td>Manually developed</td>
<td>Auto-generated</td>
</tr>
<tr>
<td>Programming with the new instructions</td>
<td>Inline assembly</td>
<td>high-level intrinsic</td>
</tr>
<tr>
<td>Support for custom registers/memory ports</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>RTL to do repeated operations on new data</td>
<td>Manually developed</td>
<td>Auto-generated</td>
</tr>
</tbody>
</table>
ACE Use Case in Embedded Datapath

- RISC-V calculates pointers, kick-starts DMA, coordinates systems
- ACE instructions pass GPR pointers to ACE engines
- All signals wired automatically
- All tools generated by COPILOT