Strength of Andes RISC-V CPU IPs

- Rich experience on CPU design and market

- High quality of RISC-V CPU IPs – Risk Free
  - Andes RISC-V good at PPA performance – Power, Performance, Area
  - Professional DSP support – DSP ISA, C libraries and compiler
  - ACE & COPILOT – Generate custom instructions automatically

- RISC-V + ACE
  - Compact + High Performance + Flexible + Low Power
Highlights of RISC-V V5 CPU IPs

- Andes Custom Extension™ (ACE)
- P-extension (DSP) and V-extension (Vector)
- Caches
  - Complete cache management, controlled by CSRs
- Local memories
  - Capable of copying boot code to LM when resetting CPU
  - Access up to 32 MB local memories via V5’s AHB slave port
- Support vectored interrupt & unaligned mem access
- Comprehensive debug solutions
<table>
<thead>
<tr>
<th>Function</th>
<th>insn cycles</th>
<th>ACE cycles</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIR 128 (only CPU)</td>
<td>1600</td>
<td>128</td>
<td>12x</td>
</tr>
<tr>
<td>LMS 128 (only CPU)</td>
<td>1800</td>
<td>128</td>
<td>14x</td>
</tr>
<tr>
<td>BT DMA (only CPU)</td>
<td>bus traffic</td>
<td>background</td>
<td></td>
</tr>
<tr>
<td>ANC Hybrid Solution</td>
<td>10000</td>
<td>250</td>
<td>40x</td>
</tr>
</tbody>
</table>
# Andes V5 Processor Lineup

<table>
<thead>
<tr>
<th>Cache-Coherent 1-4 Cores</th>
<th>Linux with FPU/DSP</th>
<th>Fast/Compact with FPU/DSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>RV32</td>
<td>RV64</td>
<td></td>
</tr>
<tr>
<td>A25MP</td>
<td>AX25MP</td>
<td>N25F D25F</td>
</tr>
<tr>
<td>A25</td>
<td>AX25</td>
<td></td>
</tr>
<tr>
<td>N25F D25F</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **27-Series:**
  - MemBoost Vector Ext.
  - NX27V A27/AX27 and more.

- **45-Series:**
  - MemBoost Dual Issue.
  - N45/NX45 D45/DX45 A45/AX45 and more.

**Note:** Common features are RV*IMACN, Caches, LM, ECC, BrPred, CoDense™, PowerBrake, StackSafe™ ACE (Andes Custom Extension™); Frequencies at 28nm

**Taking RISC-V® Mainstream**
V5 25-Series Performance

<table>
<thead>
<tr>
<th>Features</th>
<th>Base</th>
<th>FP</th>
<th>Linux</th>
<th>Linux</th>
</tr>
</thead>
<tbody>
<tr>
<td>32KB I$/D$ + 256 BTB</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SP/DP FPU</td>
<td>--</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MMU and S-Mode</td>
<td>--</td>
<td>--</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>RV-P ext. draft (DSP)</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>Yes</td>
</tr>
<tr>
<td>Worst-Case Freq. (GHz)¹</td>
<td>1.2</td>
<td>1.2</td>
<td>1.2</td>
<td>1.1</td>
</tr>
</tbody>
</table>

1: 28nm SVT 9T library and high-speed memory. Frequency at 0.81v/40°C.

■ Linux support
- RISC-V MMU and S-mode
  - SV{32,39,48}, all page sizes
- 4-way 32~128-entry STLB
- 4 or 8-entry ITLB and DTLB

■ FPU (RV-F or RV-FD)
- +, −, x, x+, x−:
  - pipelined 5 cycles
- ÷, √: run in background
  - SP: 15 cycles, DP: 29 cycles

<table>
<thead>
<tr>
<th>Features</th>
<th>Base</th>
<th>FP</th>
<th>Linux</th>
<th>Linux</th>
</tr>
</thead>
<tbody>
<tr>
<td>32KB I$/D$ + 256 BTB</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SP/DP FPU</td>
<td>--</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MMU and S-Mode</td>
<td>--</td>
<td>--</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>RV-P ext. draft (DSP)</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>Yes</td>
</tr>
<tr>
<td>Worst-Case Freq. (GHz)¹</td>
<td>1.2</td>
<td>1.2</td>
<td>1.2</td>
<td>1.1</td>
</tr>
</tbody>
</table>

Coremark/MHz²: 3.58 (rv32), 3.53 (rv64)
DMIPS/MHz²: 1.96 (rv32), 2.13 (rv64)

1: 28nm SVT 9T library and high-speed memory. Frequency at 0.81v/40°C.
2: AndeSight v320 toolchain; DMIPS/ground rule uses no-inline option.

Whetstone/MHz

NX25F | N25F | CM7 | CA7
---|---|---|---
1.38 | 1.56 | 1.58 | 0.50 | 1.09 | 0.54

DP | SP

Taking RISC-V® Mainstream
Andes RISC-V V5 in SoC

- Single core
- 2-8 cores
- > 30 cores
- > 100 cores
- > 1000 cores

Taking RISC-V® Mainstream
Andes RISC-V 25-Series Core Overview

- **AndeStar V5 architecture:**
  - RV32/RV64-IMACN + Andes Extensions
  - Optional FPU: SP, DP
  - Optional **DSP/SIMD: P**
  - Optional S-mode/MMU: SV32/39/48

- **5-stage pipeline, single-issue**
- **Configurable multiplier**
- **Optional branch prediction**
- **I/D caches and Local Memory**
  - Optional parity or ECC protection
  - Hit-under-miss caches
  - HW unaligned load/store accesses

- **Bus interface**
  - Master ports (AXI64*2/AHB{64,32})
  - Optional **AHB slave port** accessing LM address space
# 25-Series Features Overview

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AndeStar™</td>
<td>V5+RV32-FD-N</td>
<td>V5+RV32-FD-P-N</td>
<td>V5+RV64-FD-N</td>
<td>V5+RV64-FD-P-N</td>
<td></td>
</tr>
<tr>
<td>Pipeline &amp; GPR#</td>
<td>5-stage, 32 32-bit</td>
<td>5-stage, 32 64-bit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FPU</td>
<td>Single/Double precision (IEEE754-compliant)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSP</td>
<td>RV32-P (draft), DSP/SIMD</td>
<td>--</td>
<td>RV64-P (draft), DSP/SIMD</td>
<td>--</td>
<td></td>
</tr>
<tr>
<td>MMU</td>
<td>--</td>
<td>--</td>
<td>Sv32 virtual-memory</td>
<td>--</td>
<td></td>
</tr>
<tr>
<td>Privilege mode</td>
<td>M+U</td>
<td>M+U+S</td>
<td>M+U</td>
<td>M+U+S</td>
<td></td>
</tr>
<tr>
<td>Master Bus</td>
<td>AXI64*2, AXI64, AHB64 or AHB32</td>
<td>AXI64*2, AXI64 or AHB64</td>
<td>AXI64*2, AXI64 or AHB64</td>
<td>AXI64*2, AXI64 or AHB64</td>
<td></td>
</tr>
<tr>
<td>Slave Bus</td>
<td>AHB64 or AHB32</td>
<td>32 (A25: 32-34) bit address</td>
<td>32-64 bit address</td>
<td>AHB64</td>
<td></td>
</tr>
<tr>
<td>Branch Pred.</td>
<td>Static/Dynamic (BTB,BHT,RAS)</td>
<td>Radix2/Radix4/Radix16/Radix256/Fast</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multiplier</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memory system</td>
<td>I&amp;D Local Memory, up to 16MiB; I&amp;D cache, up to 64KiB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unique Features</td>
<td>CoDense™, StackSafe™, PowerBrake, QuickNap™, ECC/Parity, misaligned access, Andes Custom Extension™</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Debug Module</td>
<td>4-wire JTAG/2-wire Serial Debug Port; with Exception Redirection</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLIC</td>
<td>Vectored dispatch, priority-based preemption, up to 1023 sources, 255 priorities, 16 targets</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Taking RISC-V® Mainstream
<table>
<thead>
<tr>
<th>Feature</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>RISC-V User-Level Interrupt Extension</td>
<td>yes/no</td>
</tr>
<tr>
<td>RISC-V Atomic Instruction Extension</td>
<td>yes/no</td>
</tr>
<tr>
<td>RISC-V Float-Point Extension</td>
<td>none/Single-Precision/Double-Precision</td>
</tr>
<tr>
<td>Andes Cutsom Extension</td>
<td>yes/no</td>
</tr>
<tr>
<td>Privilege Architecture</td>
<td></td>
</tr>
<tr>
<td>Privilege Modes</td>
<td>Machine/Machine + User</td>
</tr>
<tr>
<td>Physical Memory Protection Entries</td>
<td>0/2/4/8/16</td>
</tr>
<tr>
<td>Performance Monitors</td>
<td>yes/no</td>
</tr>
<tr>
<td>Andes Vectored PLIC Extension</td>
<td>yes/no</td>
</tr>
<tr>
<td>Andes StackSafe Extension</td>
<td>yes/no</td>
</tr>
<tr>
<td>Andes PowerBrake Extension</td>
<td>yes/no</td>
</tr>
<tr>
<td>Bus Interface</td>
<td></td>
</tr>
<tr>
<td>Bus Type</td>
<td>ahb/axi</td>
</tr>
<tr>
<td>BIU Two-port Structure</td>
<td>yes/no</td>
</tr>
</tbody>
</table>
Configurable CPU Subsystem

- JTAG Debug Xport
- Debug Module
- PLIC
- 25-series BIU
- Memory
- Data Memory
- AXI64*2/AHB64,32
- AHB64,32

Taking RISC-V® Mainstream
Easy SoC Integration

AE350 Platform

Taking RISC-V® Mainstream
Speedup with DSP ISA on 25-Series

- Real world speed up, using DSP extension

<table>
<thead>
<tr>
<th></th>
<th>RV64P</th>
<th>RV32P</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR (Image Classification)</td>
<td>14x</td>
<td></td>
</tr>
<tr>
<td>PNET (90% of Face detection)</td>
<td>7.57x</td>
<td></td>
</tr>
<tr>
<td>Keyword Spotting (voice)</td>
<td>5.36x</td>
<td></td>
</tr>
<tr>
<td>AMR voice codec</td>
<td>3.67x</td>
<td></td>
</tr>
<tr>
<td>MP3 decode</td>
<td>1.95x</td>
<td></td>
</tr>
</tbody>
</table>

• Taking RISC-V® Mainstream

- 64 bit
  - RV64P
  - PNET (90% of Face detection)
  - Keyword Spotting (voice)
  - CIFAR (Image Classification)

- 32 bit
  - RV32P
  - AMR voice codec
  - MP3 decode

- Speedup with DSP ISA on 25-Series
DSP Support

- **DSP ISA**
  - The basis of RISC-V P-extension draft that Andes contributed.
  - 300 instructions derived and evolved from real use cases (over decades)
    - Support 32 bits and 64 bits
    - Support saturation and rounding
    - Cover SIMD, partial SIMD, bit manipulation and etc.

- **DSP intrinsic functions**
  - Users can use as C-like functions without bothering to program in assembly

- **DSP library**
  - >200 functions in 8 categories (basic, complex, controller, filtering, matrix, statistics, transform, utils)

- **Some DSP instructions are auto-generated by compiler to facilitate development**

- **Compatible with CMSIS-DSP library API**
  - By including an API wrapper header file
  - Microcontroller Software Interface Standard (CMSIS)
### DSP Library Comparison with CPU A

**● RV32-P: Speedups over CPU A (with 3% larger code size)**

<table>
<thead>
<tr>
<th>Q</th>
<th>Speedup</th>
<th>Basic</th>
<th>Cmplx</th>
<th>Ctrl</th>
<th>Filter</th>
<th>Matrix</th>
<th>Ststcs</th>
<th>Xform</th>
<th>Utils</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVG</td>
<td>1.80</td>
<td>1.26</td>
<td>1.73</td>
<td>1.31</td>
<td>1.19</td>
<td>2.20</td>
<td>1.08</td>
<td>1.40</td>
<td></td>
<td>1.50</td>
</tr>
<tr>
<td>MAX</td>
<td>6.94</td>
<td>1.80</td>
<td>2.17</td>
<td>2.63</td>
<td>1.77</td>
<td>6.75</td>
<td>1.31</td>
<td>2.77</td>
<td></td>
<td>6.94</td>
</tr>
<tr>
<td>F32 AVG</td>
<td>1.31</td>
<td>1.33</td>
<td>2.31</td>
<td>1.08</td>
<td>1.42</td>
<td>1.23</td>
<td>1.14</td>
<td>1.24</td>
<td></td>
<td>1.38</td>
</tr>
<tr>
<td>MAX</td>
<td>1.42</td>
<td>1.64</td>
<td>2.55</td>
<td>2.09</td>
<td>1.78</td>
<td>1.35</td>
<td>1.39</td>
<td>2.05</td>
<td></td>
<td>2.55</td>
</tr>
</tbody>
</table>

**● RV32-P: Speedups over CPU A (with 32% smaller code size)**

<table>
<thead>
<tr>
<th>Q</th>
<th>Speedup</th>
<th>Basic</th>
<th>Cmplx</th>
<th>Ctrl</th>
<th>Filter</th>
<th>Matrix</th>
<th>Ststcs</th>
<th>Xform</th>
<th>Utils</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVG</td>
<td>1.45</td>
<td>1.11</td>
<td>1.54</td>
<td>1.28</td>
<td>1.03</td>
<td>1.93</td>
<td>1.07</td>
<td>1.30</td>
<td></td>
<td>1.34</td>
</tr>
<tr>
<td>MAX</td>
<td>5.15</td>
<td>1.59</td>
<td>1.85</td>
<td>2.63</td>
<td>1.56</td>
<td>5.29</td>
<td>1.31</td>
<td>2.77</td>
<td></td>
<td>5.29</td>
</tr>
<tr>
<td>F32 AVG</td>
<td>1.01</td>
<td>1.13</td>
<td>1.74</td>
<td>1.03</td>
<td>1.16</td>
<td>1.12</td>
<td>1.13</td>
<td>1.02</td>
<td></td>
<td>1.17</td>
</tr>
<tr>
<td>MAX</td>
<td>1.35</td>
<td>1.48</td>
<td>2.11</td>
<td>2.09</td>
<td>1.55</td>
<td>1.22</td>
<td>1.39</td>
<td>2.05</td>
<td></td>
<td>2.11</td>
</tr>
</tbody>
</table>
## DSP Instruction Examples

<table>
<thead>
<tr>
<th>Types</th>
<th>Instruction Operations</th>
<th>Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIMD</td>
<td>Four 8x8 multiplications: 16= 8x8; 16= 8x8; 16= 8x8; 16= 8x8</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Two 16x16 multiplications: 32= 16x16; 32= 16x16</td>
<td></td>
</tr>
<tr>
<td>Partial SIMD</td>
<td>Four 8x8 multiplications with 32b accumulation: 32= 32 + 8x8 + 8x8 + 8x8 + 8x8; 32= 32 + 8x8 + 8x8 + 8x8 + 8x8 (2(^{nd}) op: RV64 only)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Two 16x16 multiplications with 32b accumulation: 32= 32 + 16x16 + 16x16; 32= 32 + 16x16 + 16x16 (2(^{nd}) op: RV64 only)</td>
<td></td>
</tr>
<tr>
<td>RV64 Only</td>
<td>Two 32x32 multiplications with 64b accumulation: 64= 64 + 32x32 + 32x32</td>
<td>3</td>
</tr>
</tbody>
</table>
## D25F vs. CPU A

<table>
<thead>
<tr>
<th>Features</th>
<th>D25F</th>
<th>CPU A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom Instruction</td>
<td>Andes Custom Extension™</td>
<td>No</td>
</tr>
<tr>
<td>Pipeline stages</td>
<td>5 stages</td>
<td>3 stages</td>
</tr>
<tr>
<td>Floating point</td>
<td>SP, DP, HP conv. at LD/ST, Background $\div \sqrt{}$</td>
<td>SP only</td>
</tr>
<tr>
<td>DSP Extensions</td>
<td>SIMD-instructions with 8/16/32-bit element size</td>
<td>8/16-bit SIMD arithmetic</td>
</tr>
<tr>
<td></td>
<td>Complex DSP instructions operating on 16/32/64-bit data</td>
<td></td>
</tr>
<tr>
<td>I/D Local Memory</td>
<td>4KB~16MB</td>
<td>Yes</td>
</tr>
<tr>
<td>L1 I/D Cache</td>
<td>8KB~64KB</td>
<td>No</td>
</tr>
<tr>
<td>SRAM Error Protection</td>
<td>ECC or Parity</td>
<td>No</td>
</tr>
<tr>
<td>Bus Interface</td>
<td>AHB32, AHB64, or AXI64</td>
<td>AHB Lite, APB</td>
</tr>
<tr>
<td>I/D Local Memory DMA</td>
<td>With AHB slave port</td>
<td>No</td>
</tr>
<tr>
<td>Pre-integrated Platform</td>
<td>AXI-based platform</td>
<td>No</td>
</tr>
<tr>
<td>Additional Features</td>
<td>CoDense™ code size reduction, StackSafe™ stack protection, PowerBrake &amp; QuickNap™ power management</td>
<td>-</td>
</tr>
<tr>
<td>DMIPS/MHz</td>
<td>1.96</td>
<td>1.25</td>
</tr>
<tr>
<td>CoreMark/MHz</td>
<td>3.58</td>
<td>3.42</td>
</tr>
</tbody>
</table>

Note: N25F use AndeSight v3.1.0; DMIPS/MHz follows the ground rule with no-in-line option.
AndeShape™ and Comprehensive Kits

- **AndeShape™ Development Boards**
- **Debugging Hardware**
  - AICE-MINI+, AICE-MICRO
- **Near-Cycle Accurate Simulator**
- **Qemu Virtual Board**
- **AndeSoft™ Software Stack**
  - Bare metal demo projects
  - RTOS’es: FreeRTOS, ThreadX, Contiki, more
  - Linux: RV32/RV64, UP and SMP
- **Rich Support from 3rd Parties**
  - IAR, Imperas, Lauterbach, Segger, UltraSoC, etc.
- Configurable L2 cache size of 0KiB, 128KiB, 256KiB, 512KiB, 1MiB and 2MiB

AndesCore™ A*25 Multi-Core Processor

- PLIC I/F
- Trace Port (x4)
- L2 CSR
- Bus Master Interface: AXI-128/AXI-64

IO LM Slave Port x4 (AHB-32/64)
IO Coherence Slave Port (AXI-64)
Debug I/F (e.g., to Debug Xport)

L1-to-L2 64b

Cache Coherence (MESI)/L2 Cache Controller

Core 0
ILM | DLM
I$ | D$

Core 1
ILM | DLM
I$ | D$

Core 2
ILM | DLM
I$ | D$

Core 3
ILM | DLM
I$ | D$

Core 0
Core 1
Core 2
Core 3

M

S

Taking RISC-V® Mainstream
### A25MP Multi-Core Benchmark on Linux

<table>
<thead>
<tr>
<th>Processors</th>
<th>Single-core</th>
<th>Dual-core</th>
<th>Quad-core</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoreMark</td>
<td>3.50</td>
<td>6.92</td>
<td>13.84</td>
</tr>
</tbody>
</table>

### AX25MP Multi-Core Benchmark on Linux

<table>
<thead>
<tr>
<th>Processors</th>
<th>Single-core</th>
<th>Dual-core</th>
<th>Quad-core</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoreMark</td>
<td>3.45</td>
<td>6.90</td>
<td>13.80</td>
</tr>
</tbody>
</table>
Andes Custom Extension™ (ACE) Framework

- C code
- Verilog
- Attributes

- scalar/vector
- background
- wide operands

COPILOT
Custom-OPtimized Instruction deVeLOpment Tools

- Extended Tools
- Extended ISS
- Extended RTL

Automated Env. For Cross Checking
Test Case Generator

Extended ISS
Extended RTL

Compiler Asm/Disasm Debugger

IDE

CPU ISS (near-cycle accurate)

CPU RTL

Extensible Baseline Components

Taking RISC-V® Mainstream
Inner Product of Vectors with 64 8-bit Data

```plaintext
reg CfReg {
    num = 4;
    width = 512;
}

ram VMEM {
    interface = sram;
    address_bits = 3;  //8 elements
    width = 512;
}

insn ip64B {
    operand = {out gpr IP, in CfReg C, in VMEM V};
    csim = %{
        IP = 0;
        for (uint i = 0; i < 64; ++i)
            IP += ((C >> (i*8)) & 0xff) * ((V >> (i*8)) & 0xff);
    };
    latency = 3;  //enable multi-cycle ctrl
};
```

//ACE_BEGIN: ip64B

//ACE_END

Intrinsic: long ace_ip64B(CfReg_t, VMEM_t);

Speedup: 85x
Custom Port (ACP) for Direct HW Engine Control

**port command** {  
  // a 90-bit output port to  
  // all 4 HW engines,  
  width = 90;  
  // including a valid bit and  
  // a HW engine ID field  
  io_type = out;  
}

// 4 HW engines

**port ready** {  
  // 4 ready signals  
  num = 4;  
  io_type = in;  
}

**port results** {  
  // 4 256-bit input ports  
  num = 4;  
  width = 256;  
  io_type = in;  
}

**App. code sequence:**
prepare command (say, thru ACR);  
send command;  
do other useful work;  
wait for results to be ready;  
get results;
Design for Energy Efficiency

- Clock Gating
- RTL Designs
- Microarchitecture
  - Performance
  - Efficient Designs
- PowerBrake
  - Clock Throttling
  - w. Duty Ratio
- QuickNap™
  - Cache-Intact Fast Core
  - Power Down, Wakeup

Performance Critical Tasks

- Fully Operational

- Core Logic Power Off

- Logic and SRAM off

- Instantaneous Power

- Periodic Services

- Idle Loop

- Time
AndeSight™: Professional IDE

- Eclipse-based, enriched by 15-year effort

### FreeRTOS Task List

<table>
<thead>
<tr>
<th>Task Name</th>
<th>x.priority</th>
<th>start of stack</th>
<th>top of stack</th>
<th>state</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main</td>
<td>3</td>
<td>0x0800200</td>
<td>0x0800700</td>
<td>Running</td>
</tr>
<tr>
<td>Task1</td>
<td>3</td>
<td>0x0800800</td>
<td>0x0800D00</td>
<td>Blocked</td>
</tr>
<tr>
<td>Task2</td>
<td>2</td>
<td>0x0801000</td>
<td>0x0801500</td>
<td>Ready</td>
</tr>
</tbody>
</table>

### FreeRTOS Event List

<table>
<thead>
<tr>
<th>Event Name</th>
<th>heap_used</th>
<th>stack_used</th>
<th>running</th>
<th>waiting_for</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMA DE</td>
<td>0x080000</td>
<td>0x080050</td>
<td>120</td>
<td>0</td>
</tr>
<tr>
<td>Tasks Waiting For</td>
<td>0x080010</td>
<td>0x080060</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Taking RISC-V® Mainstream
AndeSoft™: Bare Metal Support

- **Bare metal**
  - Rich startup demo projects for Andes-specific features
    - PLIC, CLIC
    - MMU, PMP, cache, ECC, bus matrix slave port
    - PowerBrake, hibernate, WFI CPU standby/resume
    - StackSafe™, performance monitor
    - DSP, printf UART redirect, C++ programming
  - AMSI (Andes MCU Software Interface) driver APIs
    - UART, GPIO, RTC, PWM, QSPI, I2C and WDT
  - Easy to use and catch up
AndeSoft™: RTOS Support

RTOS
- FreeRTOS (open source): 32/64 bits
- ThreadX (from Express Logic): 32/64 bits
- RISC-V ready: Zephyr, RT-Thread, SylixOS, μC/OS-[II/III], MyNewt, LiteOS, AliOS Things

FreeRTOS v10.0
- FreeRTOS test suite verified
- Support AE350 (AXI/AHB) platform
- Tickless idle
  - Reduce power consumption by stopping periodic tick interrupt in the idle mode
  - Based on RISC-V standard mtime/mtimecmp
- RTOS-awareness debugging
  - AndeSight™
  - Lauterbach’s Trace32®
AndeSoft™: Linux Support

- **Linux distribution:**
  - Fedora port ready
  - OpenWRT port for networking

- **Linux kernel tools**
  - strace/ftrace for developers to debug
  - Perf to evaluate the bottleneck of the whole system
  - Power management
    - Suspend2ram: suspended by sysfs and wakeup by RTC and UART interrupt
    - PowerBrake: power throttling mechanism controlled by sysfs
  - Kernel module support all relocation types for RV32 and RV64

- **Development tools:**
  - Linux awareness debugging
    - Lauterbach Trace32®
Open Source Contributions

- Major contributor, some as maintainer
- **GCC/Binutils**
  - RV32IE
  - Interrupt attribute
  - ELF attribute support
- **LLVM/LLD**
  - RV[32|64]IMAFDC code gen
  - Hard-float calling convention.
- **Debugging tools:** GDB, OpenOCD
- **Linux**
  - ftrace, Perf, kernel module, non-coherent support
  - Fedora port
- **u-Boot**

```
[root@fedora-riscv ~]# cat /proc/cpuinfo
hart : 0
isa : rv64i2p0m2p0a2p0f2p0d2p0c2p0x5-0p0
mmu : sv39

[root@fedora-riscv ~]# uname -a
Linux fedora-riscv 4.17.0-00250-gd63b2bc-dirty #4 PREEMPT
[root@fedora-riscv ~]
```
Why Andes for RISC-V

Your Trusted RISC-V CPU Vendor

- Leading Commercial RISC-V CPU company
- CPU IPO company with 15-year history
- 350+ customers worldwide
- + 5 Bn shipping record
- Profession FAE team and support system
- Better code size and performance, mature tool
Thank You