Accelerating RISC-V AI and IoT Development with Andes Software Solutions

Simon TC Wang
Technical Marketing Manager
Andes Technology
April 26, 2021
Agenda

• AndeSight™ IDE
• AndeSoft™ BSP
• AI Software Stack
  • DSP and Vector programming
  • NN library and real NN model use cases
Andes at A Glance

Who We Are

CPU
Pure-play
CPU IP Company

RISC-V Founding
Premier Member

Taiwan Stock
Exchange Listed

Hammer
Major Open-Source
Contributor/Maintainer

Director of the Board
Vice Chair of TSC
Running Task Groups
RISC-V Ambassador

Quick Facts

16 years old
company

80%
R&D
employees

7B+
(By end of 2020)
accumulated Andes-
embedded SoC shipped

200+
Licensees
Worldwide

20K+
AndeSight IDE
installation

Taking RISC-V® Mainstream
AndeSight™ IDE
Comprehensive Development Environment
AndeSight™ IDE for RISC-V

Accelerate your RISC-V software developments with the comprehensive development environment!

AndeSight™ IDE with AndeSoft™ BSP inside

- User-friendly and easy-to-use IDE
- Highly-optimized GCC/LLVM toolchains for outstanding performance and compact memory footprint
- Abundant demos for boosting your development
- Rich RTOSes and Linux (LTP verified)
- Optimized compute library: DSP, Vector
- Peripheral drivers for AndeShape™ platform
- Near cycle-accurate simulators: AndeSim™
- Arduino support for Andes Corvette EVB

The new AndeSight™ IDE v5.0 is ready to launch in June, 2021
SoC SW Development Environment

AndeSight™ IDE
- Eclipse-Based IDE
- Streamlined GUI
- Feature-rich Editor
- Managed Build System
- Optimized Toolchains
- Source Level Debugger
- Profiling Analysis
- Extensive Demo Projects
- Flash ISP

AndeSoft™ BSP
- Application Layers
- Middleware
- Drivers
- App Drivers
- OS/Kernel
- Libraries

Andes Partners
Customer

AndeShape™ AICE
- Managed Build System
- Extensive Demo Projects
- Optimized Toolchains
- Streamlined GUI
- Feature-rich Editor
- Source Level Debugger
- Profiling Analysis

AndeSim™ & AndeSysC™ Virtual Platform
- AndesCore™
- Bus Controller
- AHB to APB Bridge
- SDRAM Controller
- DMA Controller
- SRAM Controller

Andes Partners
Customer

AndeShape™ EVB

configure platforms
build executables
dump interactively
execute programs

profiling/tracing/debugging data

AndesCore™ Bus Controller AHB to APB Bridge
SDRAM Controller DMA Controller SRAM Controller

Taking RISC-V® Mainstream

Subject to change without notice
Copyright © 2021 Andes Technology
AndeSight™: Professional IDE

- Eclipse-based IDE, enriched from 16-year continuous development

- Bit-filed display
- Function profiling
- Meta linker script editor

RTOS awareness
- Custom plugin
- Function code size

Taking RISC-V® Mainstream
FreeRTOS Timeline Analyzer

- Visualize the runtime execution behaviors of task, interrupts and events within a period of time\(^1\)

1: EVB target only
Build and debug the multicore software with separate configurations by simply creating multicore projects.
Core Grouping

- Grouping Cores for Efficient Debugging
  - Debug commands can be sent to a specific set of cores at the same time.
AndeSight™ Scripting

- Record and replay UI operations with Python script
  - Test automation
  - Issue reproduction
  - ...

Start to record UI operations

Python script:
```python
run.py
project.select('sc')
project.select('demo_nn_relu_q7')
run('demo_nn_relu_q7 Default')
run.select('demo_nn_relu_q7 Default')
run.config('run.py').create('Script')
run('run.py')
run('demo_nn_relu_q7 Default')
run.select('demo_nn_relu_q7 Default')
```
AndeSoft™ BSP
Application Building Blocks Inside of AndeSight™ IDE
AndeSoft™ BSP: Application Building Blocks

**Fundamental**

- Compiler/Toolchain are contributed to and supported officially by GNU/LLVM communities
- Optimized MCUlib, newlib and glibc
- Optimized low-level compute libraries for NN, DSP and vector processing: libnn, libdsp, libvec
- Fast and near cycle-accurate simulators: AndeSim™, AndeSysC™, Qemu¹
- Debugging: GDB and ICEman (speed-optimized OpenOCD)
- Concise linker script and its tools, Linker Scattering-and-Gathering (LdSaG)
- Bare-metal drivers and demo programs to demo AndesCore™ features

**Real-Time Operating Systems**

- Open source port on Andes: Zephyr, FreeRTOS
- Commercial port on Andes: Azure RTOS ThreadX, LiteOS, RT-Thread, SylixOS
- RISC-V ready: VxWorks, µC/OS-[II/III], MyNewt, embOS, RTEMS, Nuttx, seL4, uC3/Compact, AliOS Things, TencentOS Tiny, HarmonyOS, Nucleus

**Linux, Middleware and SW Framework**

- Linux kernel v4.17 and LTS v5.4, device drivers and advanced features: strace, ftrace, Perf, SMU, power throttling, suspend to RAM, HIGHMEM and kernel module
- U-Boot, OpenSBI and BBL

¹: available upon request
AndeSoft™: Bare Metal

- Rich **startup demo** projects for Andes-specific features

<table>
<thead>
<tr>
<th>Categories</th>
<th>Startup demo</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Interrupt</strong></td>
<td>PLIC, CLIC</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>MMU, PMP, PMA, cache, cache lock, ECC, bus matrix slave port</td>
</tr>
<tr>
<td><strong>Power Management</strong></td>
<td>PowerBrake, hibernate, WFI CPU standby/resume</td>
</tr>
<tr>
<td><strong>Programming</strong></td>
<td>DSP, printf UART redirect, C++ programming</td>
</tr>
<tr>
<td><strong>Misc</strong></td>
<td>StackSafe™, performance monitor, SMP</td>
</tr>
</tbody>
</table>

- **AMSI (Andes MCU Software Interface) driver APIs**
  - DMA, Flash, GPIO, I2C, PWM, RTC, SPI, UART and WDT
**AndesSoft™: Linux**

### Linux kernel
- Mainline Linux compatible
- V4.17 and LTS v5.4, RV[32|64]GC, SMP
- Cache coherence and non-coherence support
- Linux Test Projects (LTP) verified
- Device drivers for AndeShape™ AE350 platform

### Kernel features
- **strace/ftrace** to debug Linux applications easily
- **Perf** to evaluate the system bottleneck
- **Power Management**
  - Suspend-to-RAM: suspended by sysfs and wakeup by RTC and UART interrupt
  - PowerBrake: power throttling controlled by sysfs
- **Kernel module** all RV32/RV64 relocation type
- **HIGHMEM** support >1GB memory for RV32
- **CPU hotplug**

### U-Boot, U-Boot-spl, OpenSBI and BBL
AndeSoft™: Linux

- Linux Distribution and Build System*

- Andes GitLab Service for Linux Development Packages
  - A public web service: git.andestech.com
  - Under Andes Maintenance Program

*: available upon request
AI Software Stack
DSP, Vector and NN Programming
AI Software Stack and Ecosystem

AI applications

Signal and data processing
pre-/post-processing for audio, voice, video, image, ...

AI frameworks & NN model format

NN model optimization

NN inference engine

AndeSoft™ LLVM compiler
AndeSoft™ OpenCL™

AndeSoft™ DSP Library
AndeSoft™ Vector Library
AndeSoft™ NN Library

AndesCore™ baseline, RVP, RVV, ACE
N-Series, D/A-Series, V-Series
First implementation of RISC-V DSP/SIMD extension (RVP)

- **DSP intrinsic functions**: as C-like functions without writing error-prone assembly
- **AndeSoft™ DSP library**: ~300 functions in 10 categories
  - Basic, complex, controller, distance, filtering, matrix, statistic, SVM, transform, utils
- Increase power efficiency to your DSP applications: 2.0x-5.6x speedup for codecs, and 8.9x-14.5x speedup for NN models

---

**Speedup of RVP over baseline**

<table>
<thead>
<tr>
<th>Application</th>
<th>RV32P Speedup</th>
<th>RV64P Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP3 decoder</td>
<td>2.0x</td>
<td></td>
</tr>
<tr>
<td>AMR-WB voice codec</td>
<td>3.7x</td>
<td></td>
</tr>
<tr>
<td>EVS codec</td>
<td>4.9x</td>
<td></td>
</tr>
<tr>
<td>LC3plus codec</td>
<td>5.6x</td>
<td></td>
</tr>
<tr>
<td>ML-KWS (Keyword Spotting)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PNET (90% of Face Detection)</td>
<td>8.9x</td>
<td></td>
</tr>
<tr>
<td>CIFAR-10 (Image Classification)</td>
<td>10.0x</td>
<td>14.5x</td>
</tr>
</tbody>
</table>

---

1: based on 25-Series, FPGA
AndesSoft™ DSP Library

- Optimized low-level DSP functions for RISC-V Baseline and RVP processors
- Compatible with CMSIS-DSP API
- Compiler automatically generates partial RVP instructions to facilitate development
- Boost signal processing performance
- Speedup 2.9x in average and 21.9x in maximum

Speedup of RVP over baseline\(^1\)
The higher the better

<table>
<thead>
<tr>
<th>Category</th>
<th>RV32P (AVG)</th>
<th>RV32P (MAX)</th>
<th>RV64P (AVG)</th>
<th>RV64P (MAX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic</td>
<td>25.0</td>
<td>25.0</td>
<td>25.0</td>
<td>25.0</td>
</tr>
<tr>
<td>Complex</td>
<td>25.0</td>
<td>25.0</td>
<td>25.0</td>
<td>25.0</td>
</tr>
<tr>
<td>Controller</td>
<td>25.0</td>
<td>25.0</td>
<td>25.0</td>
<td>25.0</td>
</tr>
<tr>
<td>Distance</td>
<td>25.0</td>
<td>25.0</td>
<td>25.0</td>
<td>25.0</td>
</tr>
<tr>
<td>Filtering</td>
<td>25.0</td>
<td>25.0</td>
<td>25.0</td>
<td>25.0</td>
</tr>
<tr>
<td>Matrix</td>
<td>25.0</td>
<td>25.0</td>
<td>25.0</td>
<td>25.0</td>
</tr>
<tr>
<td>Statistics</td>
<td>25.0</td>
<td>25.0</td>
<td>25.0</td>
<td>25.0</td>
</tr>
<tr>
<td>SVM</td>
<td>25.0</td>
<td>25.0</td>
<td>25.0</td>
<td>25.0</td>
</tr>
<tr>
<td>Transform</td>
<td>25.0</td>
<td>25.0</td>
<td>25.0</td>
<td>25.0</td>
</tr>
<tr>
<td>Utils</td>
<td>25.0</td>
<td>25.0</td>
<td>25.0</td>
<td>25.0</td>
</tr>
<tr>
<td>Total(^2)</td>
<td>21.9x</td>
<td>21.9x</td>
<td>2.9x</td>
<td>2.9x</td>
</tr>
</tbody>
</table>

1: based on 25-Series, FPGA

2: Speedup 2.9x in average and 21.9x in maximum
RISC-V Vector Software Solutions

### Standard software solutions in AndeSight™ IDE
- AndeSim™: new cycle-accurate simulator
- Toolchains: intrinsic functions, as C-like functions without writing error-prone assembly
- AndeSoft™ Vector library:
  - Optimized for RISC-V baseline and RVV processors
  - > 100 functions in 5 categories: basic, filtering, image, matrix, and transform
  - Compatible with NE10 library APIs

### Advanced software solutions
- AndeSysC™: SystemC support for AndeSim™
- AndesClarity™: GUI-based processor pipeline visualizer and analyzer
- AndeSoft™ NN library: optimized neural network functions

#### Speedup of RVV over baseline

<table>
<thead>
<tr>
<th>Operation</th>
<th>baseline</th>
<th>RVV 256x256</th>
<th>RVV 512x256</th>
</tr>
</thead>
<tbody>
<tr>
<td>F32 Add</td>
<td></td>
<td>21.6x</td>
<td>22.4x</td>
</tr>
<tr>
<td>Q7 convolution</td>
<td></td>
<td>30.7x</td>
<td>32.0x</td>
</tr>
<tr>
<td>HWC=(33, 33, 51)*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F32 32x32 GEMM</td>
<td></td>
<td>33.7x</td>
<td>43.2x</td>
</tr>
<tr>
<td>Q7 ReLU activation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MobileNet-v1 FP16</td>
<td></td>
<td>66.1x</td>
<td></td>
</tr>
<tr>
<td>MobileNet-v1 INT8</td>
<td></td>
<td>68.1x</td>
<td></td>
</tr>
</tbody>
</table>

1: based on NX27V, FPGA, 512 KB vector cache
*: HWC (Height, Width, Channel)

(VLEN,SIMD)=(256,256) bit
(VLEN,SIMD)=(512,256) bit

The higher the better
AndesSoft™ NN Library

- Optimized neural network functions for Pure-C, RVP, and RVV processors
- Boost NN performance by using SIMD and vector instructions
- >120 functions in 8 categories
- Compatible with CMSIS-NN APIs

Andes NN Library
- Activation
- Basic math
- Concatenation
- Convolution
- Fully-connected
- Pooling
- Softmax
- Utils

AI frameworks
- TFLiteµ
- Hookup kernel

Speedup of RVP over baseline\(^1\)
The higher the better

1: preliminary data based on 25-Series, FPGA

AndesCore™
Inference Flow with TensorFlow Lite Micro

Host (offline)
- TensorFlow framework
- NN model .tflite file
- Offline optimizer
  - Pruning
  - Quantization
  - ...

Device (runtime)
- TensorFlow Lite Micro runtime
- TensorFlow Lite Micro reference kernels
- AndeSoft™ NN Library
  - optimized kernels
- AndesCore™
  - Baseline
- AndesCore™
  - RVP/RVV
- NN HW Engine
<table>
<thead>
<tr>
<th>Categories</th>
<th>Operators</th>
<th>NN Library API¹</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Activation</strong></td>
<td>Sigmoid</td>
<td>riscv_nn_activate_[q7</td>
</tr>
<tr>
<td></td>
<td>Tanh</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ReLU</td>
<td>riscv_nn_relu_[q7</td>
</tr>
<tr>
<td></td>
<td>Leaky ReLU</td>
<td>riscv_nn_leaky_relu_q7</td>
</tr>
<tr>
<td><strong>Basic</strong></td>
<td>Element-wise</td>
<td>riscv_nn_ew_add_s8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>riscv_nn_ew_mul_s8</td>
</tr>
<tr>
<td><strong>Concatenation</strong></td>
<td>Concatenation</td>
<td>riscv_nn_concate_s8_[w</td>
</tr>
<tr>
<td><strong>Convolution</strong></td>
<td>Convolution</td>
<td>riscv_nn_conv_HWC_[q7</td>
</tr>
<tr>
<td></td>
<td>Pointwise convolution</td>
<td>riscv_nn_conv_1x1_HWC_[q7</td>
</tr>
<tr>
<td></td>
<td>Depthwise convolution</td>
<td>riscv_nn_conv_dw_HWC_[q7</td>
</tr>
<tr>
<td><strong>Fully-connected</strong></td>
<td>Fully-connected</td>
<td>riscv_nn_fc_[q7</td>
</tr>
<tr>
<td><strong>Pooling</strong></td>
<td>Maximum pooling</td>
<td>riscv_nn_maxpool_[q7</td>
</tr>
<tr>
<td></td>
<td>Average pooling</td>
<td>riscv_nn_avepool_[q7</td>
</tr>
<tr>
<td><strong>Softmax</strong></td>
<td>Softmax</td>
<td>riscv_nn_softmax_[q7</td>
</tr>
<tr>
<td><strong>Utils</strong></td>
<td>Reshape</td>
<td>riscv_nn_reshape_s8</td>
</tr>
</tbody>
</table>

¹: including but not limited to the list
# Use Case: CIFAR-10 NN Model

<table>
<thead>
<tr>
<th>CIFAR-10</th>
<th>Operators</th>
<th>NN Library APIs</th>
</tr>
</thead>
<tbody>
<tr>
<td>layer 1</td>
<td>Convolution</td>
<td>riscv_nn_conv_HWC_q7_RGB_fast</td>
</tr>
<tr>
<td></td>
<td>Activation (ReLU)</td>
<td>riscv_nn_relu_q7</td>
</tr>
<tr>
<td></td>
<td>Pooling (maxpool)</td>
<td>riscv_nn_maxpool_q7_HWC</td>
</tr>
<tr>
<td>layer 2</td>
<td>Convolution</td>
<td>riscv_nn_conv_HWC_q7_fast</td>
</tr>
<tr>
<td></td>
<td>Activation (ReLU)</td>
<td>riscv_nn_relu_q7</td>
</tr>
<tr>
<td></td>
<td>Pooling (maxpool)</td>
<td>riscv_nn_maxpool_q7_HWC</td>
</tr>
<tr>
<td>layer 3</td>
<td>Convolution</td>
<td>riscv_nn_conv_HWC_q7_fast</td>
</tr>
<tr>
<td></td>
<td>Activation (ReLU)</td>
<td>riscv_nn_relu_q7</td>
</tr>
<tr>
<td></td>
<td>Pooling (maxpool)</td>
<td>riscv_nn_maxpool_q7_HWC</td>
</tr>
<tr>
<td>layer 4</td>
<td>Fully-connected</td>
<td>riscv_nn_fc_q7_fast</td>
</tr>
<tr>
<td>layer 5</td>
<td>Softmax</td>
<td>riscv_nn_softmax_q7</td>
</tr>
</tbody>
</table>

- ✓ **Performance boost with AndeSoft™ NN library**
- ✓ **Increase power efficiency**
- ✓ **Lower response time**

---

### Speedup of CIFAR-10 NN model for image classification

The higher the better

<table>
<thead>
<tr>
<th>Andes RV64P AX25</th>
<th>3.9x</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andes RV32P D45</td>
<td>3.0x</td>
</tr>
<tr>
<td>Andes RV32P D25F</td>
<td>2.4x</td>
</tr>
<tr>
<td>Andes v3 D15</td>
<td>2.1x</td>
</tr>
<tr>
<td>Andes V3 10</td>
<td>1.2x</td>
</tr>
<tr>
<td>M7</td>
<td>1.0x</td>
</tr>
<tr>
<td>M4</td>
<td></td>
</tr>
</tbody>
</table>

1: based on 25-Series, FPGA
## Use Case: MobileNet-v1 NN Model

Inference latency of MobileNet-v1 NN model for object detection
The lower the better

<table>
<thead>
<tr>
<th>MobileNet-v1</th>
<th>Data Type</th>
<th>ISA</th>
<th>VLEN (bit)</th>
<th>SIMD (bit)</th>
<th>Real chip latency (ms)</th>
<th>Normalized latency&lt;sup&gt;3&lt;/sup&gt; (ms @1GHz, 1 core)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA9 (Xilinx PYNQ)&lt;sup&gt;1&lt;/sup&gt;</td>
<td>FP32</td>
<td>NEON</td>
<td>128721.4</td>
<td>650MHz, 2 cores</td>
<td>703.4</td>
<td></td>
</tr>
<tr>
<td>CA53 (Raspberry Pi-3B)&lt;sup&gt;1&lt;/sup&gt;</td>
<td>NEON</td>
<td>128121.8</td>
<td>1.2GHz, 4 cores</td>
<td>438.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CA72 (Firefly RK3399)&lt;sup&gt;1&lt;/sup&gt;</td>
<td>NEON</td>
<td>12877.9</td>
<td>1.8Hz, 2 cores</td>
<td>210.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CA73 (Kirin 970)&lt;sup&gt;1&lt;/sup&gt;</td>
<td>NEON</td>
<td>12841.3</td>
<td>2.36GHz, 4 cores</td>
<td>292.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Andes NX27V&lt;sup&gt;2&lt;/sup&gt;</td>
<td>FP16</td>
<td>RVV</td>
<td>512</td>
<td>256</td>
<td>62.3</td>
<td></td>
</tr>
<tr>
<td>Andes A27L2&lt;sup&gt;2&lt;/sup&gt;</td>
<td>INT8</td>
<td>RVP</td>
<td>32</td>
<td></td>
<td>564.5</td>
<td></td>
</tr>
<tr>
<td>Andes AX27L2&lt;sup&gt;2&lt;/sup&gt;</td>
<td>INT8</td>
<td>RVP</td>
<td>64</td>
<td></td>
<td>358.5</td>
<td></td>
</tr>
</tbody>
</table>

1: TVM, https://github.com/apache/tvm/wiki/Benchmark#arm-cpu
2: AndeSoft™ NN Library, PyTorchCV imagenet-1k "MobileNet x1.0", https://pypi.org/project/pytorchcv/  
preliminary data based on FPGA and scaling to 1.0 GHz. Real SoC performance will depend on memory subsystem  
3: Scale to the same frequency; estimated performance 4 cores="3"*1 core, the magic number "3" is from experience
IAR Embedded Workbench for RISC-V

- Complete build and debug toolchain for RISC-V
- Support all series of Andes RISC-V CPU
- Functional Safety Certified
  - IEC 61508
  - ISO 26262
  - EN 50128
  - IEC 62304

1: AndesCore™ 45-Series will be available in the mid-year of 2021
2: Part of Andes V5 RISC-V extensions are supported in the mid-year of 2021
AWS Device Qualification Program for FreeRTOS

- AndeShape™ Corvette-F1 N25 now is an AWS FreeRTOS-qualified evaluation platform
- The FreeRTOS port is verified and executes smoothly and consistently with AWS IoT
  - FreeRTOS kernel, Wi-Fi management, TCP/IP stack, PKCS#11\(^1\), TLS\(^2\), MQTT\(^3\)

1: Public Key Cryptography Standards (PKCS) #11 refers to the cryptographic token interface
2: Transport Layer Security
3: Message Queue Telemetry Transport
Summary

- **AndeSight™ IDE**
  - User-friendly and easy-to-use IDE
  - Accelerate your RISC-V software developments with the comprehensive development environment

- **AndeSoft™ BSP**
  - Highly-optimized toolchains for better performance and smaller memory footprint
  - Well-integrated building blocks to reduce time-to-market

- **AI Software Stack**
  - Boost the performance for RISC-V AI and IoT applications with optimized AndeSoft™ DSP, Vector and NN libraries
AndeSight™ IDE Free Download

- Previous version for evaluation
- Three-month time limit
Thank you