Andes软件解决方案加速RISC-V AI与IoT应用开发

Simon TC Wang
Technical Marketing Manager
Andes Technology
August 18, 2021

Agenda

- AndeSight™ IDE
- AndeSoft™ BSP
- AI Software Stack
  - DSP and Vector programming
  - AndeSoft™ NN library
  - Use cases of NN model
- Andes Ecosystem

AndeSight™ IDE

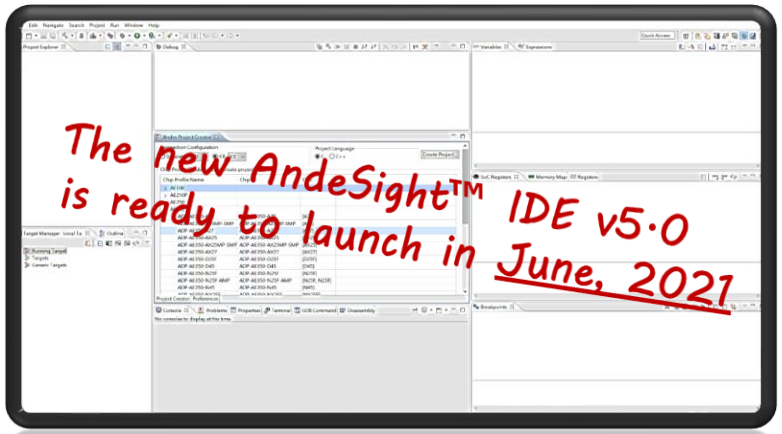
Comprehensive Development Environment

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[1]
- Peripheral drivers for AndeShape™ platform
- Near cycle-accurate simulators: AndeSim™
- Arduino support for Andes Corvette EVB

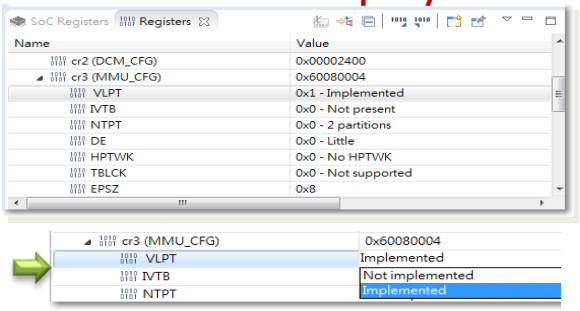


1: features in AndeSight™ v5.0

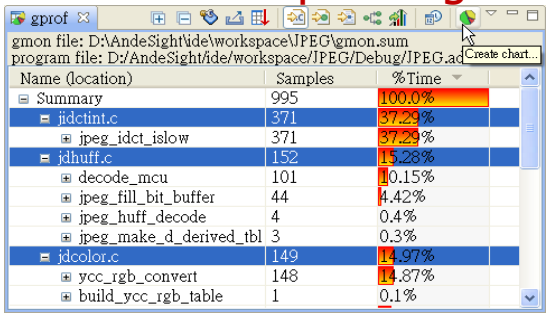
AndeSight™: Professional IDE

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

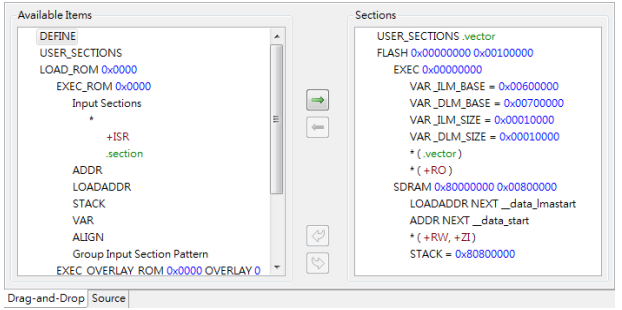
Bit-field display



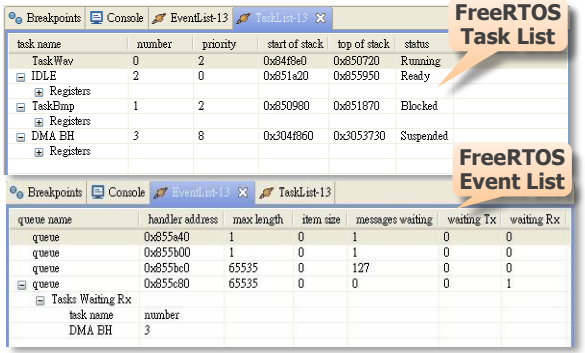
Function profiling



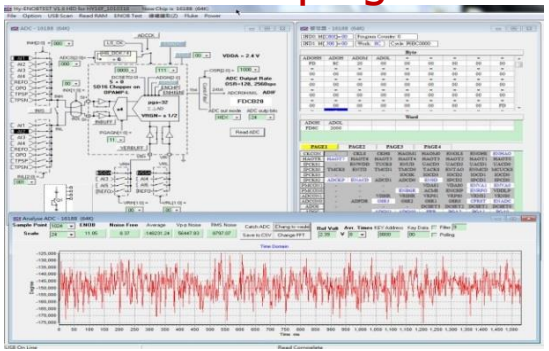
Meta linker script editor



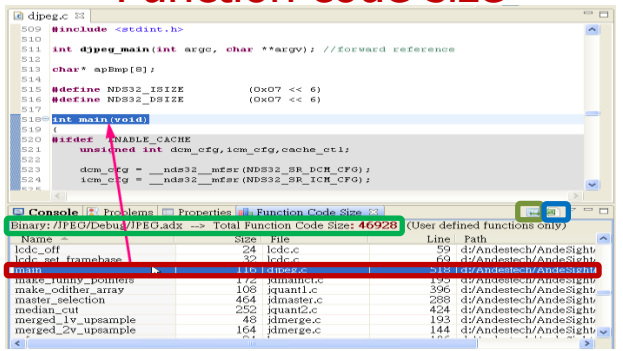
RTOS awareness



Custom plugin

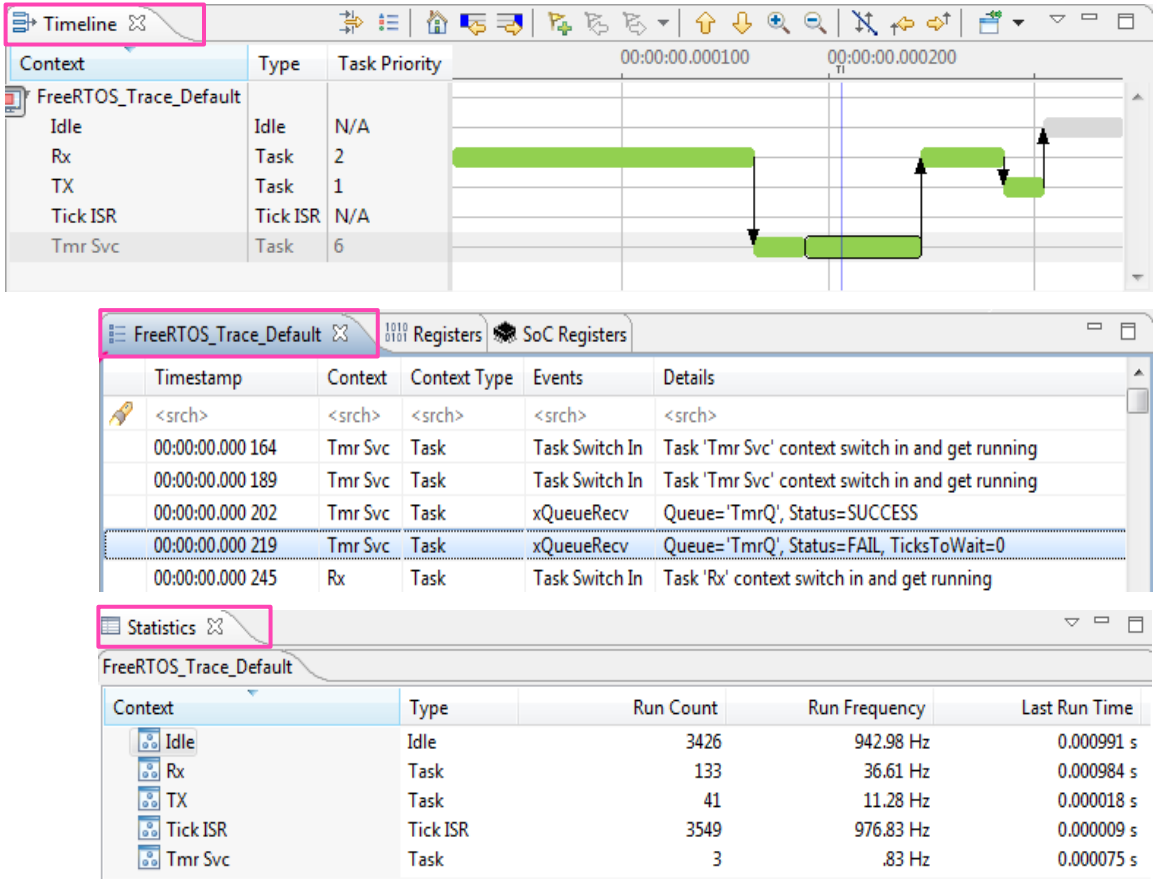


Function code size



FreeRTOS Timeline Analyzer

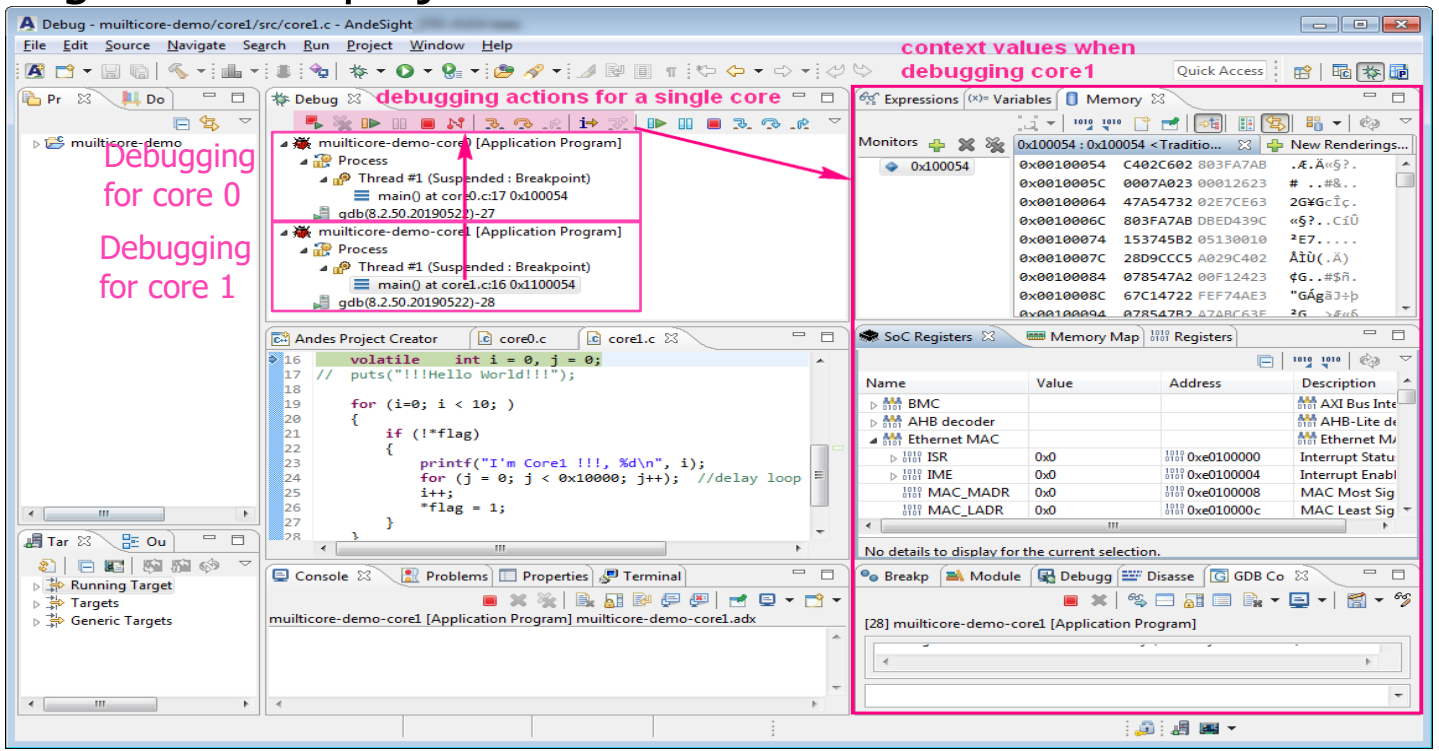
- Visualize the runtime execution behaviors of task, interrupts and events within a period of time[1]



1: EVB target only

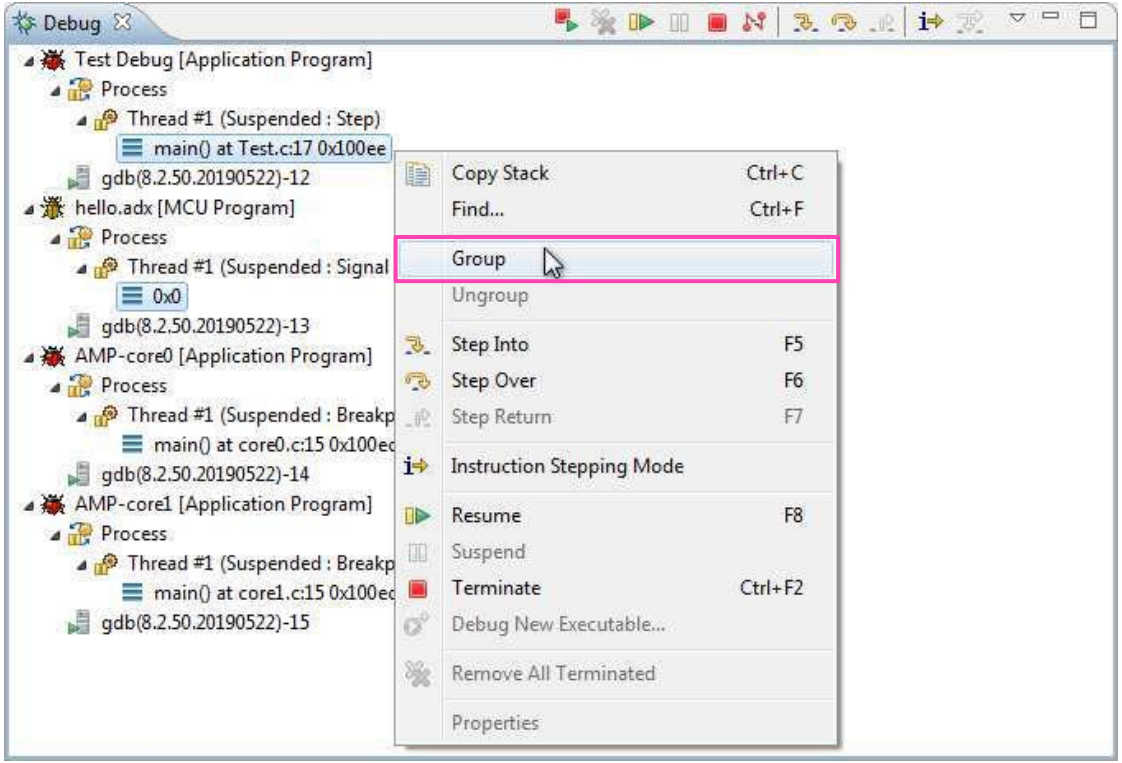
Multicore Development Support

- Build and debug the multicore software with separate configurations by simply creating multicore projects



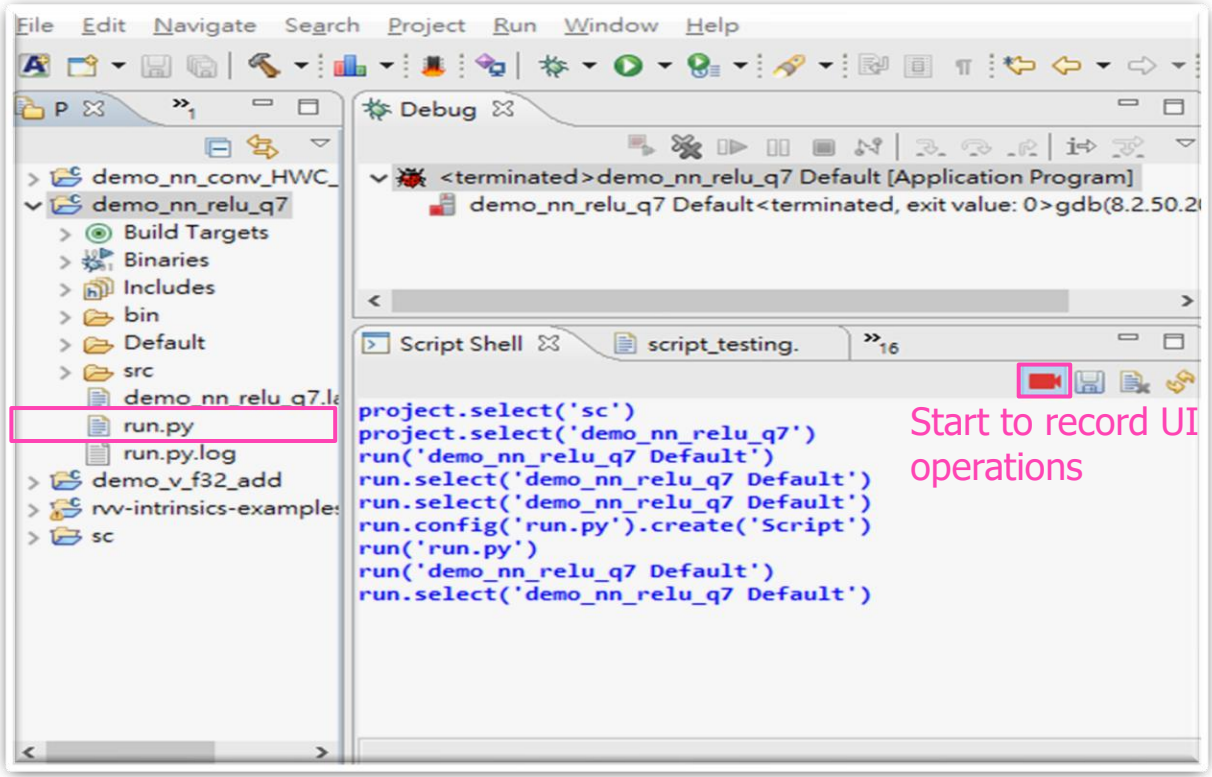
Core Grouping

- 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
  - …



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™, Qemu**[1]
- 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

aws qualified device Amazon FreeRTOS

- Open source port on Andes: **FreeRTOS (RV32/64 UP), Zephyr (RV32/64 UP/SMP)**
- Commercial port on Andes[2]: **Azure RTOS ThreadX, LiteOS, RT-Thread, SylixOS**
- **RISC-V ready**[3]**:** 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/v5.4 (LTS), UP/SMP, LTP verified, device drivers and advanced features: **strace**, **ftrace**, **Perf**, **SMU**, **power throttling**, **suspend-to-RAM, CPU hotplug, HIGHMEM** and **kernel module**
- **U-Boot, U-Boot-SPL, OpenSBI** and **BBL**

1: available upon request
2: supported by commercial vendors and not included in Andes products. Users need to license from the commercial vendors.
3: known and available RISC-V architecture port but lack of Andes platform features. Not included in Andes products.

AndeSoft™: Bare Metal

- Rich startup demo projects for Andes-specific features

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

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

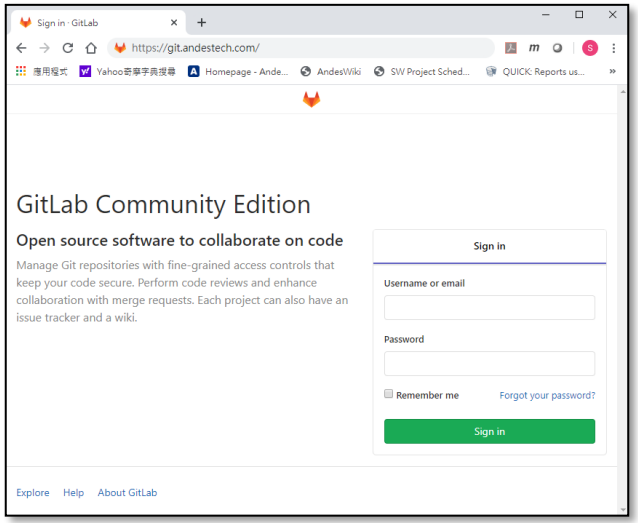
AndeSoft™: Linux

- Linux Distribution and Build System*

  *: available upon request

fedora debian yocto PROJECT OpenWrt Wireless Freedom BuildRoot Making Embedded Linux Easy

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

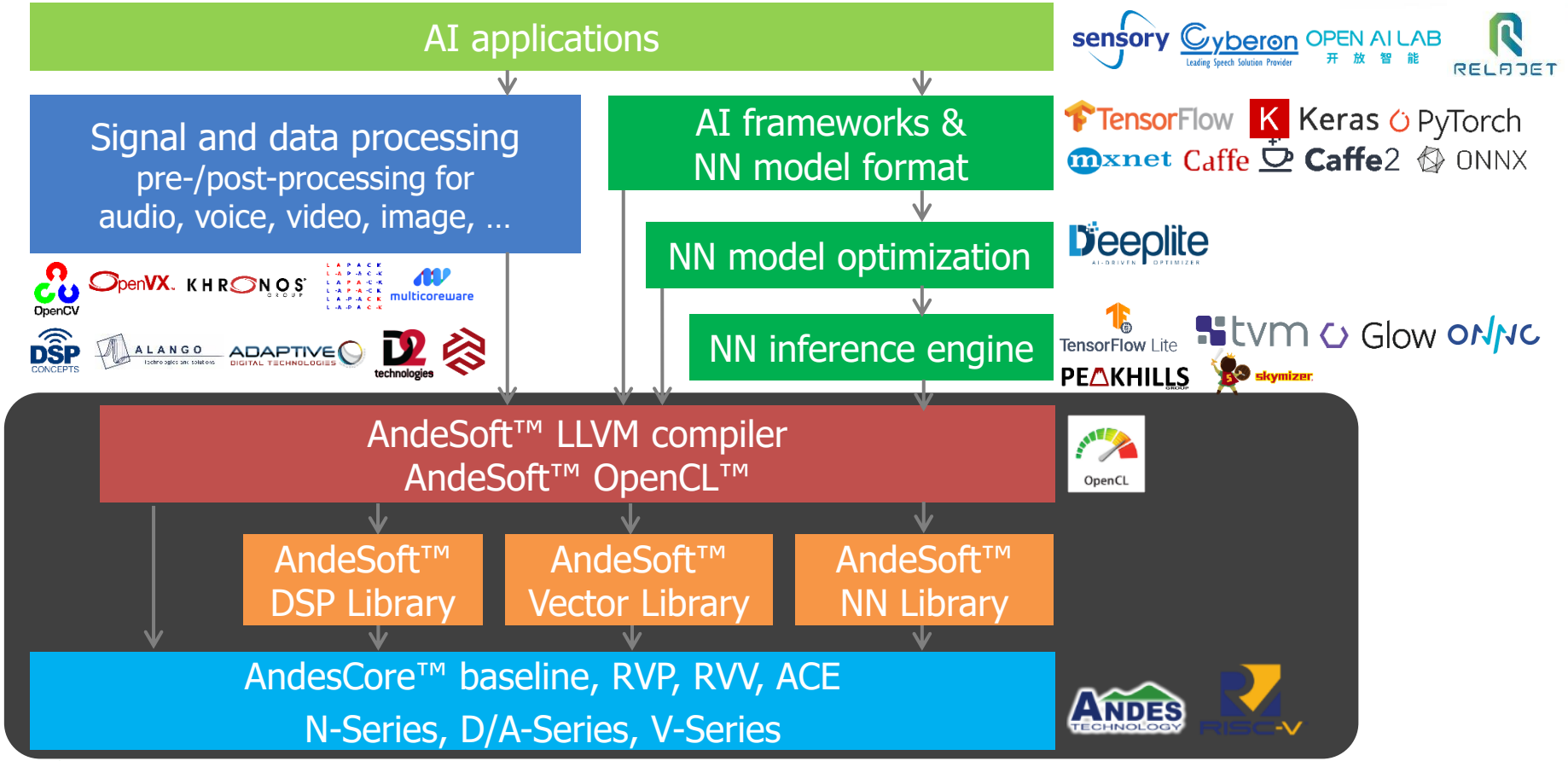


ANDES TECHNOLOGY GitLab

AI Software Stack

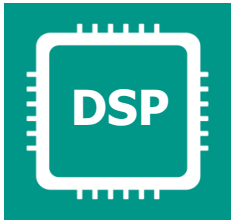
DSP, Vector and NN Programming

AI Software Stack and Ecosystem

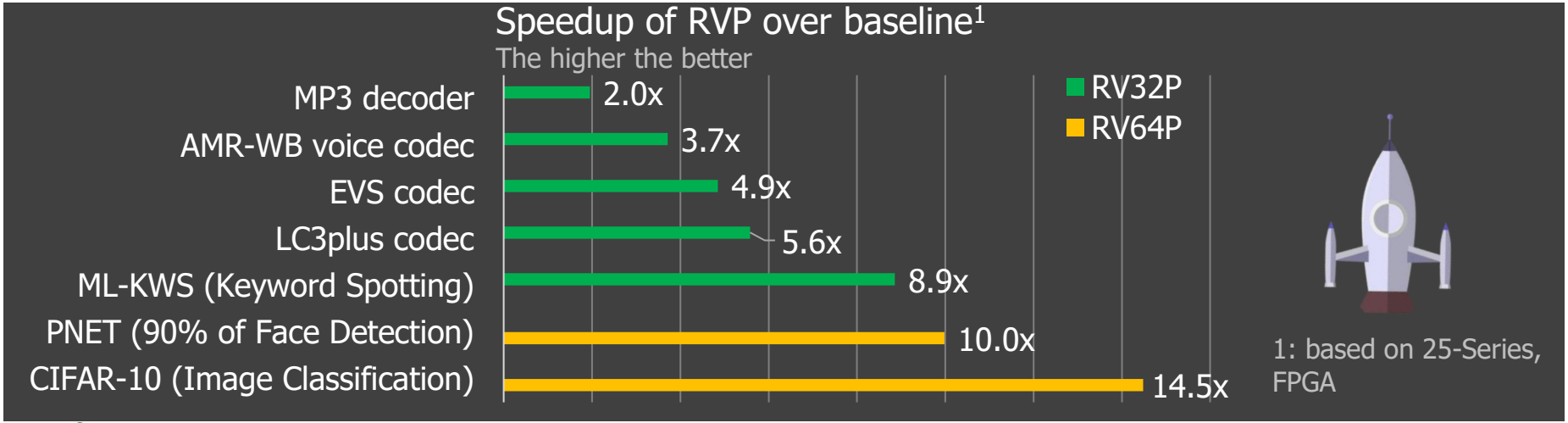


RISC-V DSP/SIMD Software Solutions

First implementation of RISC-V DSP/SIMD extension (RVP)

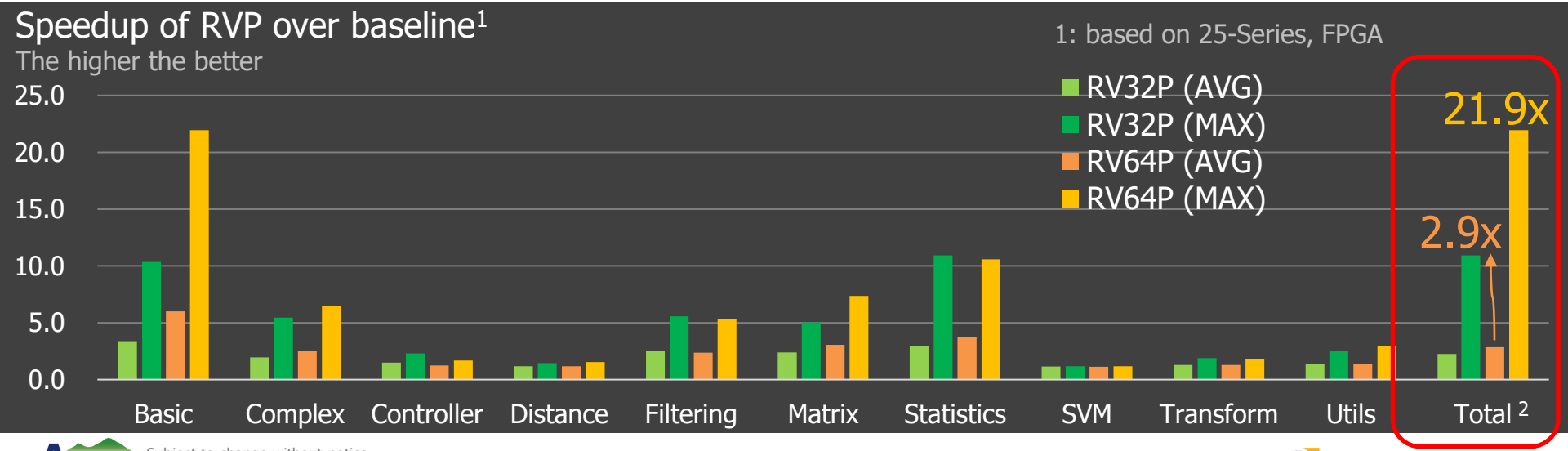


- Compiler automatically generates partial RVP instructions to facilitate development
- DSP intrinsic functions: as C-like functions without writing error-prone assembly
- AndeSoft™ DSP library
- Increase power efficiency to your DSP applications: 2.0x-5.6x speedup for codecs, and 8.9x-14.5x speedup for NN models



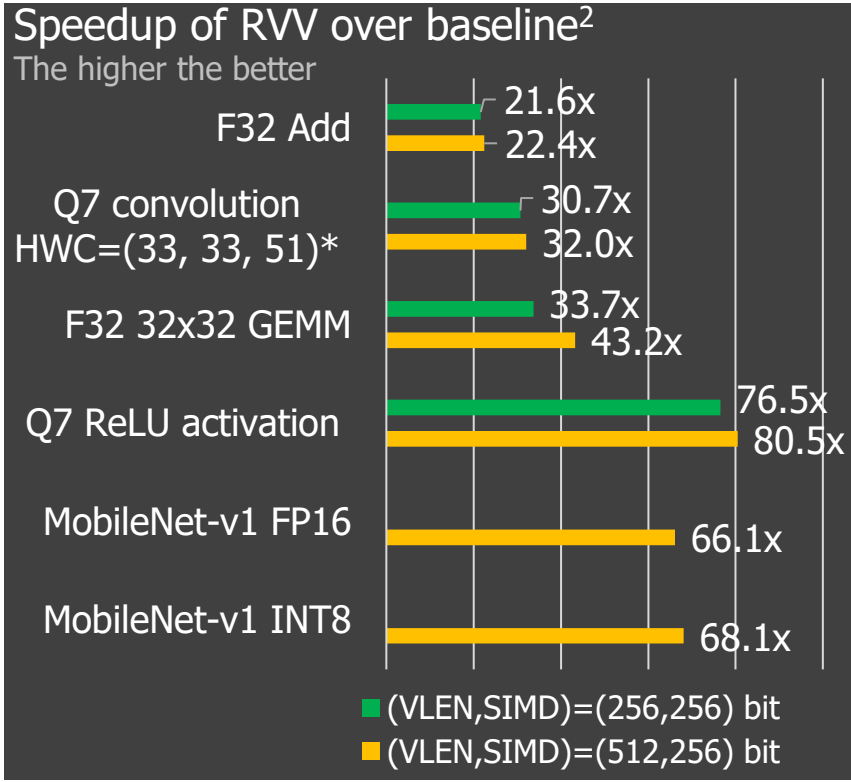
AndeSoft™ DSP Library

- Optimized low-level DSP functions for RISC-V Baseline and RVP processors
- ~300 functions in 10 categories
  - Basic, complex, controller, distance, filtering, matrix, statistic, SVM, transform, utils
- Compatible with CMSIS-DSP API
- Boost signal processing performance
- Speedup 2.9x in average and 21.9x in maximum



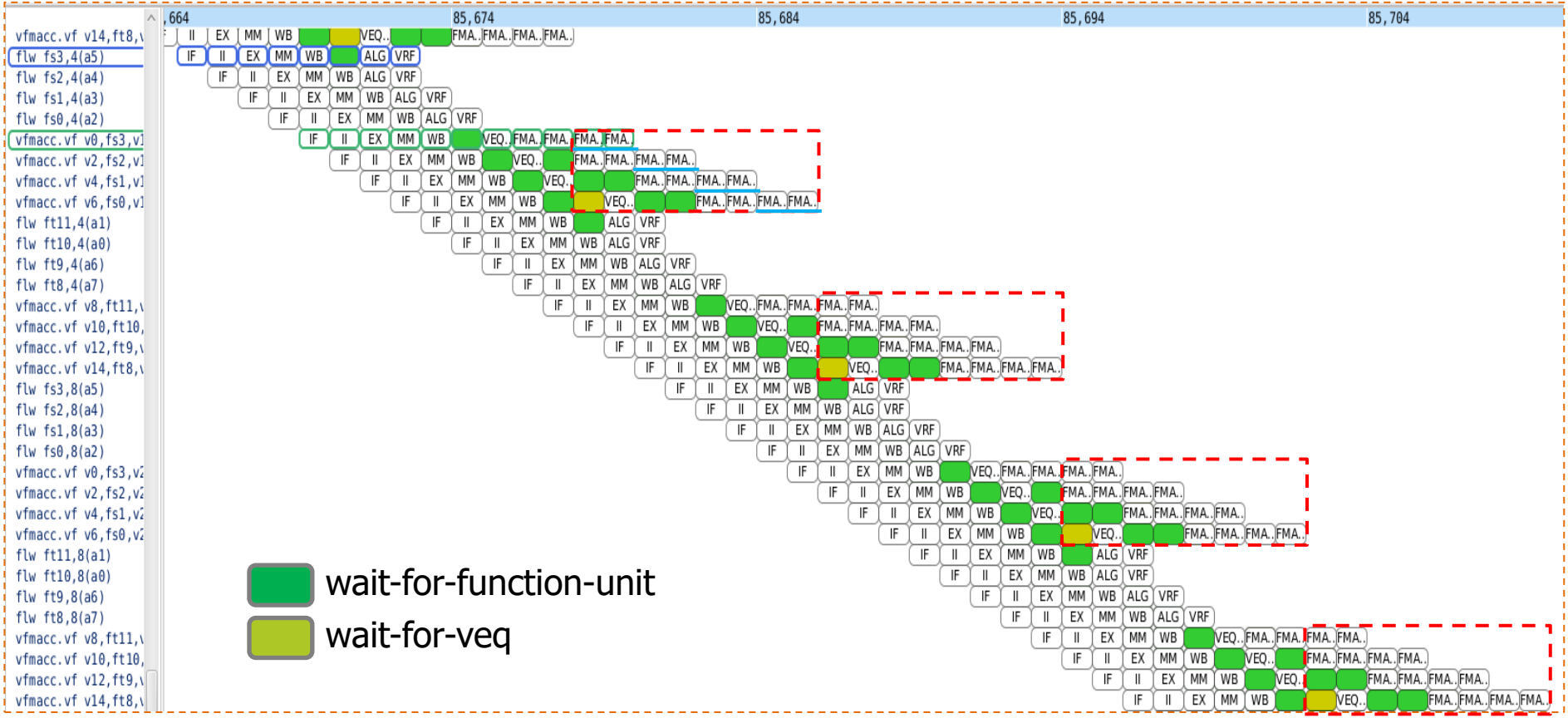
RISC-V Vector Software Solutions

- Standard software solutions in AndeSight™ IDE
  - AndeSim™: near 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 CPU
    - > 100 functions in 5 categories: basic, filtering, image, matrix, and transform
    - Compatible with NE10 library APIs
- Advanced software solutions[1]
  - AndeSysC™: SystemC support for AndeSim™
  - AndesClarity™: GUI-based processor pipeline visualizer and analyzer
  - AndeSoft™ NN library: optimized neural network functions



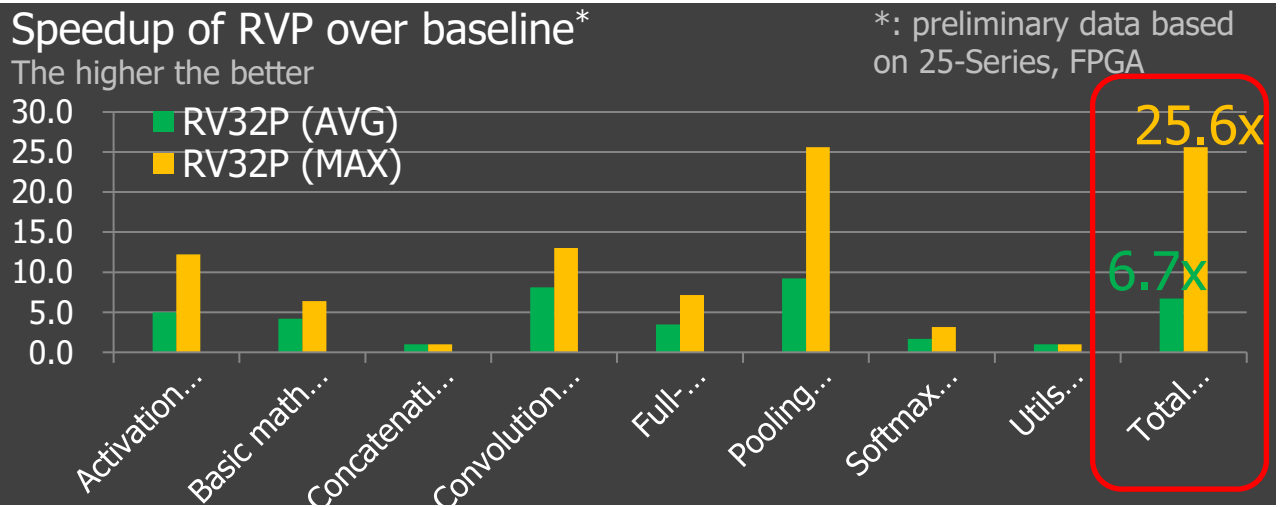
1: separate product packages needed additional licenses
2: based on NX27V, FPGA, 512 KB vector cache
*: HWC (Height, Width, Channel)

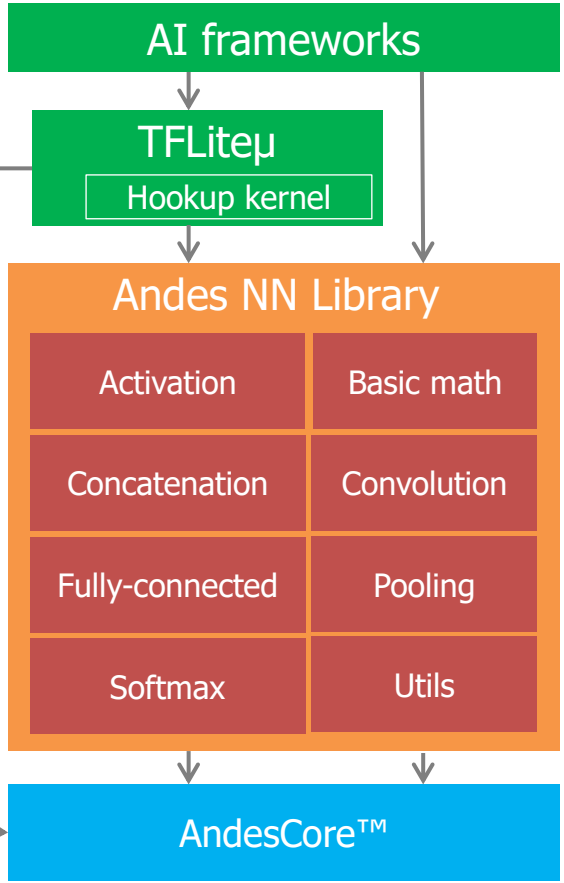
AndesClarity™ Processor Pipeline Analyzer



AndeSoft™ NN Library[1]

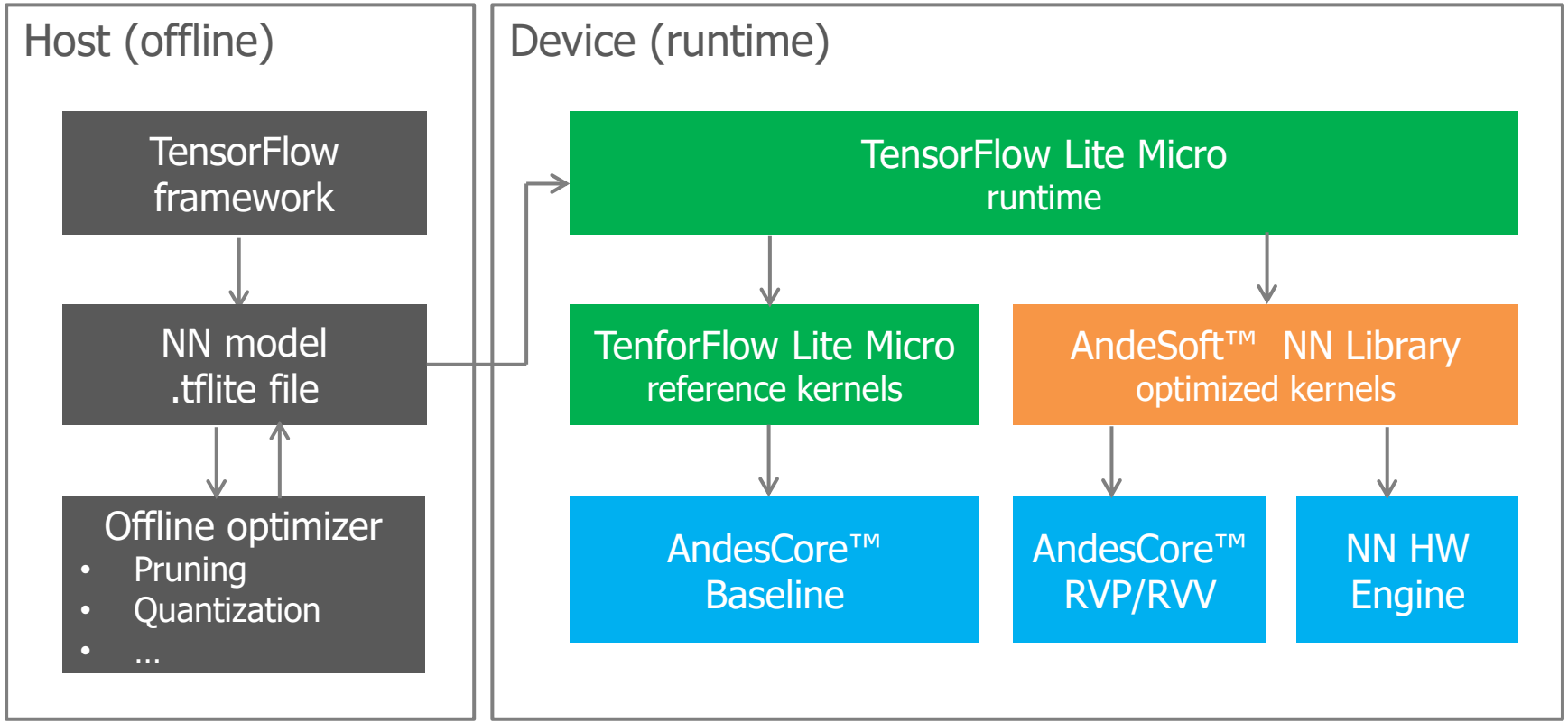
- Optimized neural network functions for Pure-C, RVP, and RVV processors
- Boost NN performance by using SIMD and vector instructions
- >120 functions in 8 categories
- Compatible with CMSIS-NN APIs





1: separate product packages needed additional licenses

Inference Flow with TensorFlow Lite Micro[1]



1: available upon request

AndeSoft™ NN Library API

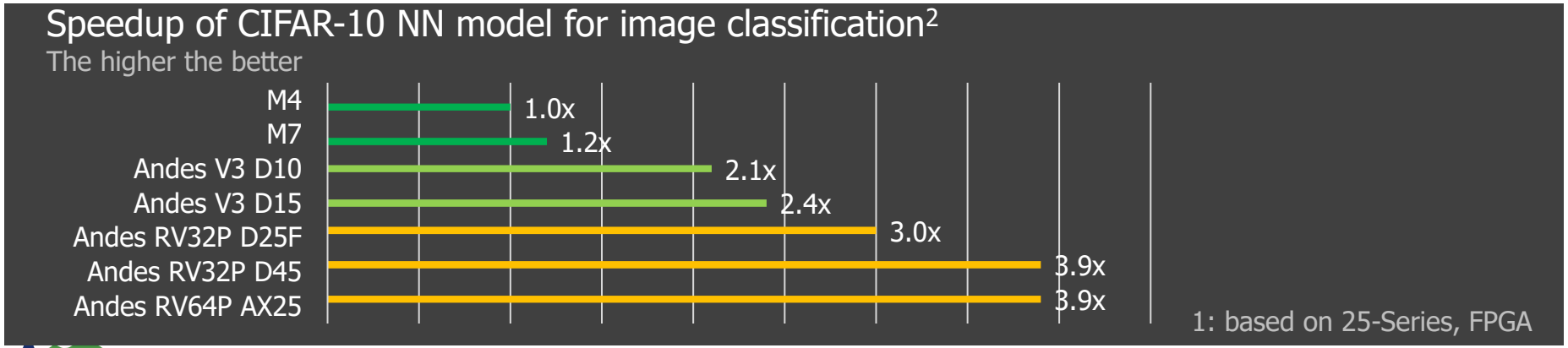
Categories	Operators	NN Library API[1]
Activation	Sigmoid Tanh	riscv_nn_activate_[s8|s16]
	ReLU Leaky ReLU	riscv_nn_relu_[s8|s16] riscv_nn_leaky_relu_s8
Basic	Element-wise	riscv_nn_ew_[add|mul]_s8
Concatenation	Concatenation	riscv_nn_concate_s8_[w|x|y|z]
Convolution	Convolution 2D Pointwise convolution Depthwise convolution	riscv_nn_conv_HWC_[IN]_[OUT]_[WEIGHT]_[sft|sym|asym][2] riscv_nn_conv_1x1_HWC_[IN]_[OUT]_[WEIGHT]_[sft|sym|asym][2] riscv_nn_conv_dw_HWC_[IN]_[OUT]_[WEIGHT]_[sft|sym|asym][2]
Fully-connected	Fully-connected	riscv_nn_fc_[IN]_[OUT]_[WEIGHT]_[sft|sym|asym][2]
Pooling	Maximum pooling Average pooling	riscv_nn_avepool_HWC_s8 riscv_nn_maxpool_HWC_s8
Softmax	Softmax	riscv_nn_softmax_[u8|s8|s16]
Utils	Reshape	riscv_nn_reshape_s8

1: including but not limited to the list
2: supporting data types of input, output and weight would be either of u8, s8 or s16

Use Case: CIFAR-10 NN Model

CIFAR-10	Operators	NN Library APIs
Layer 1	Convolution Activation (ReLU)	riscv_nn_conv_HWC_s8_s8_s8_RGB_sft_bias_fast riscv_nn_relu_s8
	Pooling (maxpool)	riscv_nn_maxpool_HWC_s8
Layer 2	Convolution Activation (ReLU)	riscv_nn_conv_HWC_s8_s8_s8_sft_bias_fast riscv_nn_relu_s8
	Pooling (maxpool)	riscv_nn_maxpool_HWC_s8
Layer 3	Convolution Activation (ReLU)	riscv_nn_conv_HWC_s8_s8_s8_sft_bias_fast riscv_nn_relu_s8
	Pooling (maxpool)	riscv_nn_maxpool_HWC_s8
Layer 4	Fully-connected	riscv_nn_fc_s8_s8_s8_sft_bias_fast
Layer 5	Softmax	riscv_nn_softmax_s8_fast

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



Use Case: MobileNet-v1 NN Model

Inference latency of MobileNet-v1 NN model for image classification

The lower the better

MobileNet-v1	Data Type	ISA	VLEN (bit)	SIMD (bit)	Normalized latency[3] (ms @1GHz, 1 core)
CA9 (Xilinx PYNQ)[1]	FP32	NEON		128	703.4
CA53 (Raspberry Pi-3B)[1]		NEON		128	438.5
CA72 (Firefly RK3399)[1]		NEON		128	210.3
CA73 (Kirin 970)[1]		NEON		128	292.4
Andes NX27V[2]	FP16	RVV	512	256	62.3

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw / s1	3 × 3 × 32 dw	112 × 112 × 32
Conv / s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw / s2	3 × 3 × 64 dw	112 × 112 × 64
Conv / s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw / s1	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw / s2	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 256	28 × 28 × 128
Conv dw / s1	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw / s2	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 512	14 × 14 × 256
5× Conv dw / s1	3 × 3 × 512 dw	14 × 14 × 512
5× Conv / s1	1 × 1 × 512 × 512	14 × 14 × 512
Conv dw / s2	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw / s2	3 × 3 × 1024 dw	7 × 7 × 1024
Conv / s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Avg Pool / s1	Pool 7 × 7	7 × 7 × 1024
FC / s1	1024 × 1000	1 × 1 × 1024
Softmax / s1	Classifier	1 × 1 × 1000

Table 2. Resource Per Layer Type

Type	Mult-Adds	Parameters
Conv 1 × 1	94.86%	74.59%
Conv DW 3 × 3	3.06%	1.06%
Conv 3 × 3	1.19%	0.02%
Fully Connected	0.18%	24.33%

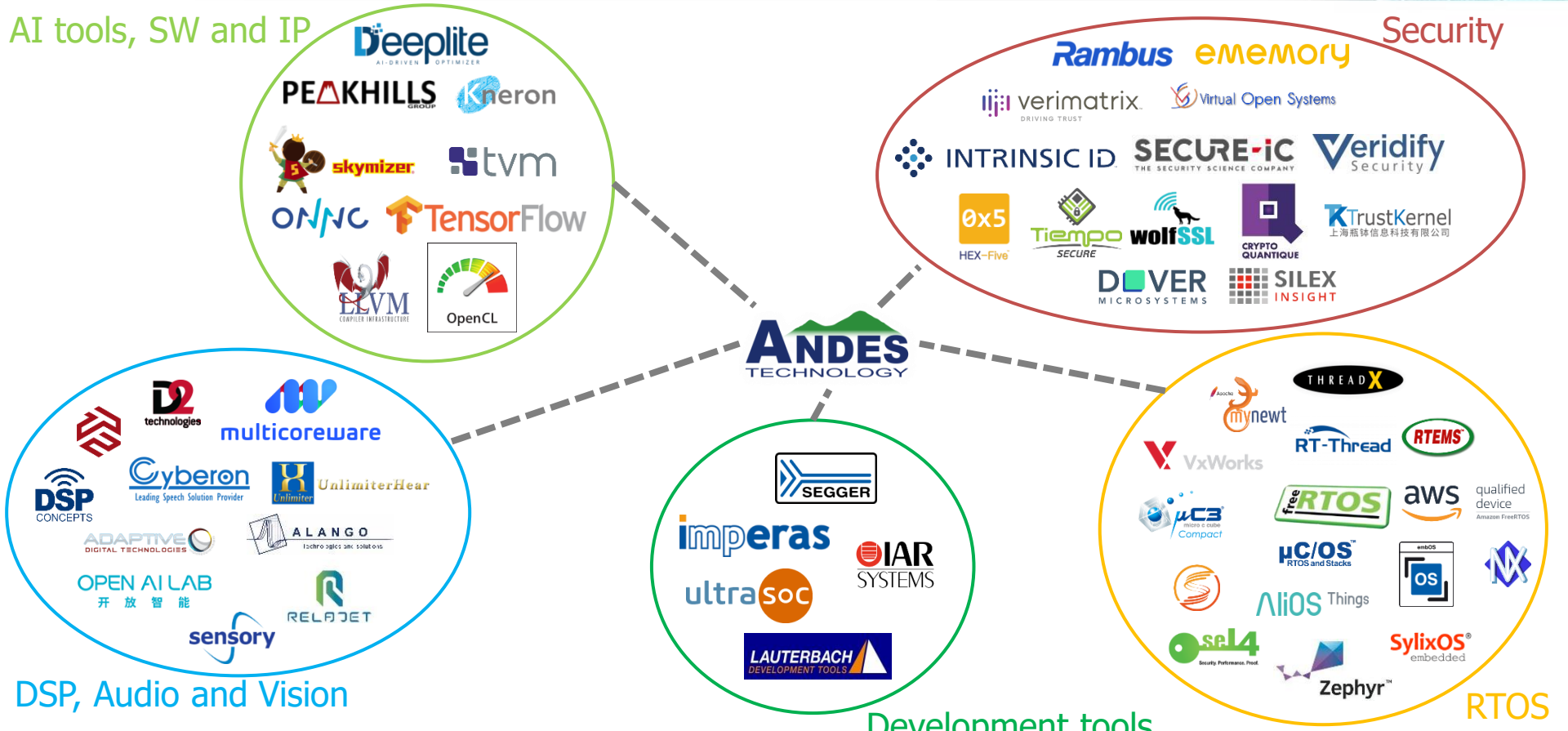
1: TVM, https://github.com/apache/tvm/wiki/Benchmark#arm-cpu

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

Andes Ecosystem

Andes Partners and Ecosystem



Cyberon DSpotter

- Cyberon DSpotter: Always-listening Voice Trigger/Command
  - DNN-based modeling: higher noise robustness
  - Phoneme-based approach: no need to collect training data
  - Low resource requirement
  - Friendly tool for rapid command customization
  - Global language support: 40+ languages available

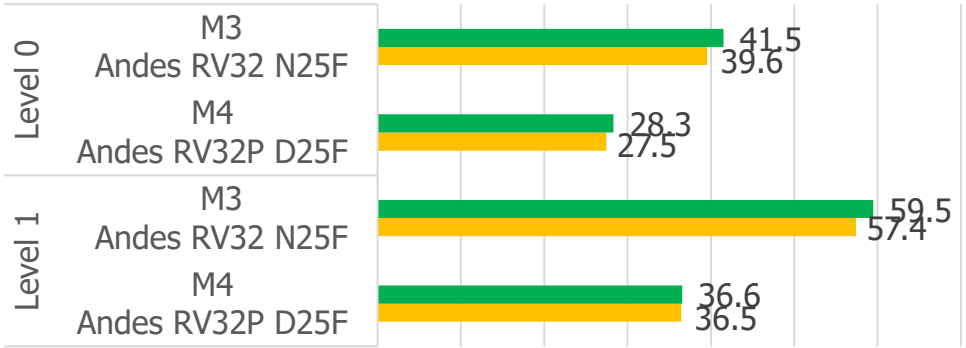
Cyberon
Leading Speech Solution Provider

Model	Level 0	Level 1
Hit rate	98.4% (SNR 10dB) 86.5% (SNR 5dB)	98.4% (SNR 10dB) 90.4% (SNR 5dB)
Code size	30 KB	30 KB
Data size	95KB + 64B x N	155KB + 64B x N
RAM	40KB + 128B x N	45 KB + 128B x N

N: number of commands

DSpotter computation (MCPS)[1]

The lower the better

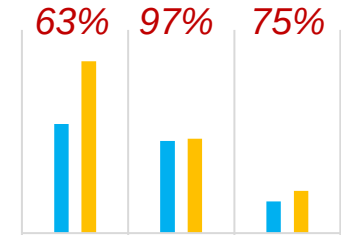


1: "M" data is based on real chip platform; Andes data is based on FPGA wit h similar configurations

Tiny ONNC | Tiny and Faster on RISC-V

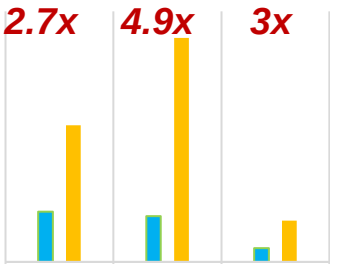
- **Tiny ONNC** exports popular PyTorch models into a function within *Andes NN Library (libnn)* function calls
- **Tiny** – the transformed *Tiny MLPerf* benchmark uses only 1/8 flash and 63% SRAM, comparing *ONNC* with *TensorflowLite for Microcontroller (TFLM)*
- **Faster** – on *Andes D25F platform*, the generated benchmark is 4.9x faster than TFLM on ARM Cortex-M4

SRAM (byte)



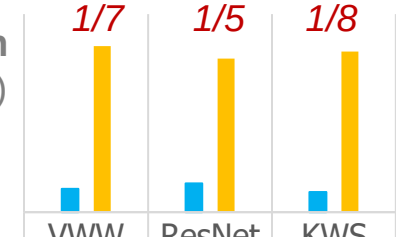
	VWW	ResNet	KWS
Tiny PyTorch	64516	54548	18818
TLFM	101492	55792	25072

Time (ms)



	VWW	ResNet	KWS
TP+D25	80	73	22
TFLM+CM4	218	358	66

Flash (byte)



	VWW	ResNet	KWS
Tiny PyTorch	5886	7264	5086
TFLM	41904	38770	40486

PYTORCH × skymizer × ANDES TECHNOLOGY

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!

ANDES
TECHNOLOGY