



# **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



Pure-play  
CPU IP Company



RISC-V Founding  
Premier Member



Taiwan Stock  
Exchange Listed



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<sup>+</sup>** (By end of 2020)  
accumulated Andes-  
embedded SoC shipped

**200<sup>+</sup>**  
Licensees  
Worldwide

**20K<sup>+</sup>**  
AndeSight IDE  
installation





# AndeSight™ IDE

Comprehensive Development Environment

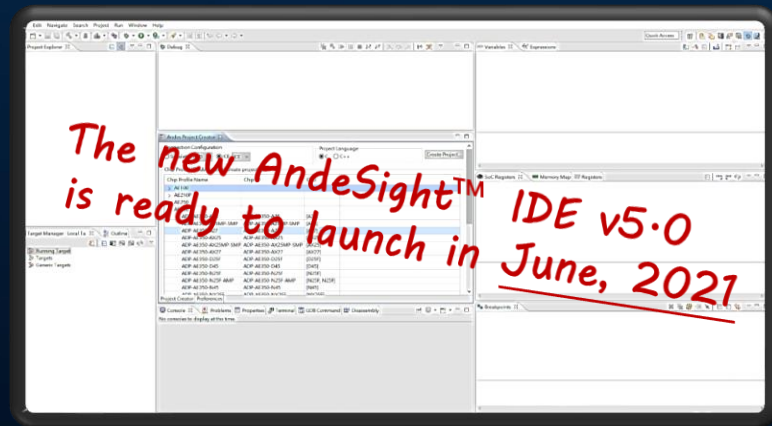


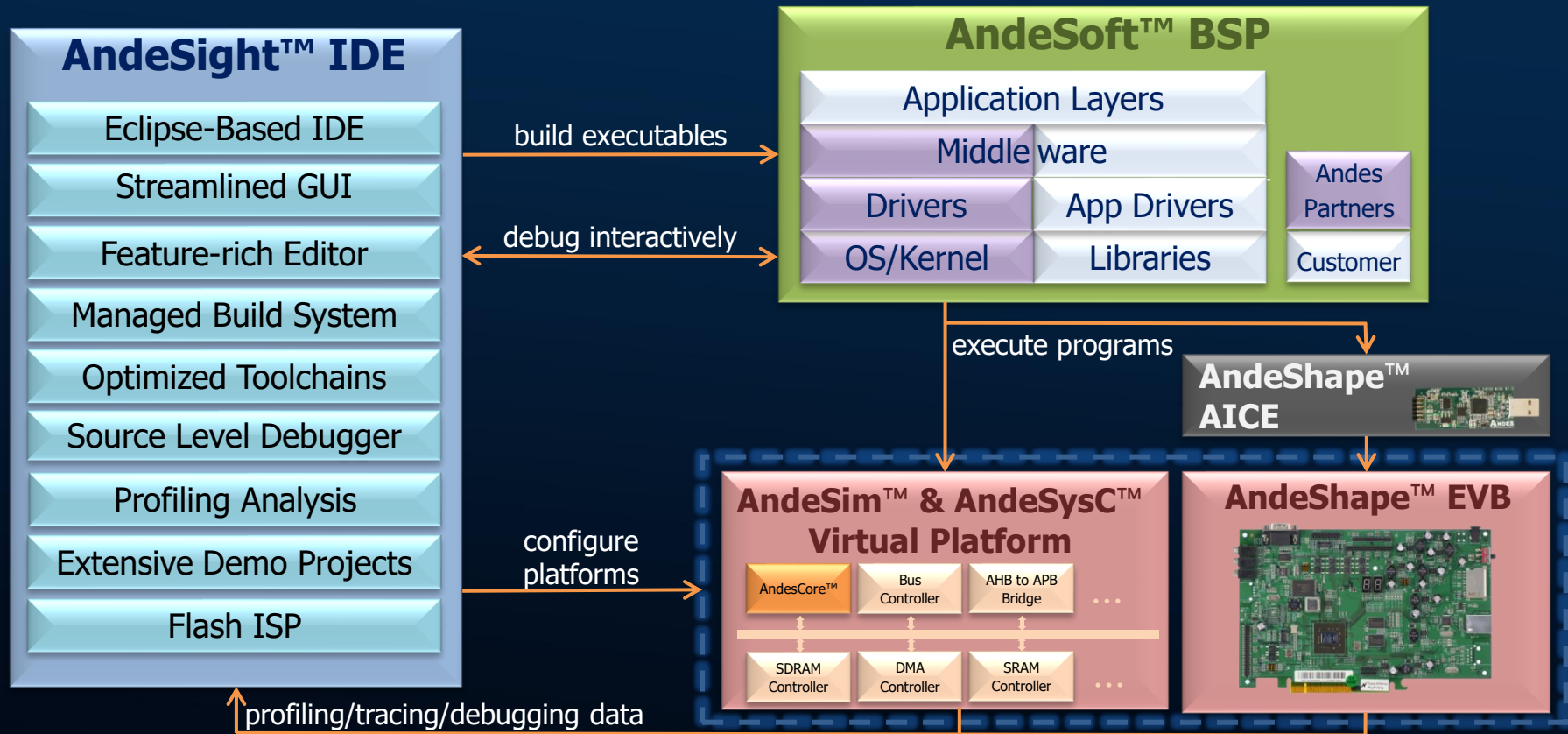
# AndeSight™ IDE for RISC-V

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<sup>1</sup>
- Peripheral drivers for AndeShape™ platform
- Near cycle-accurate simulators: AndeSim™
- Arduino support for Andes Corvette EVB









# AndeSight™: Professional IDE

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

### Bit-filed display

SoC Registers

Name	Value
cr2 (DCM_CFG)	0x00002400
cr3 (MMU_CFG)	0x60080004
VLPT	Implemented
IVTB	0x0 - Not present
NTPT	0x0 - 2 partitions
DE	0x0 - Little
HPTVWK	0x0 - No HPTVWK
TBLCK	0x0 - Not supported
EPSZ	0x8

cr3 (MMU\_CFG) 0x60080004

- VLPT Implemented
- IVTB Not implemented
- NTPT Implemented

### Function profiling

gmon file: D:\AndeSight\workspace\IPEG\gmon.sum  
program file: D:\AndeSight\workspace\IPEG\Debug\IPEG.ad

Name (location)	Samples	%Time
Summary	995	100.0%
jdcint.c	371	37.29%
jpeg_idct_islow	371	37.29%
jdhuft.c	152	15.28%
decode_mcu	101	10.15%
jpeg_fill_bit_buffer	44	4.42%
jpeg_huff_decode	4	0.4%
jpeg_make_d_derived_tbl	3	0.3%
jdcolor.c	149	14.97%
ycc_rgb_convert	148	14.87%
build_ycc_rgb_table	1	0.1%

### Meta linker script editor

Available Items

```
DEFINE
USER_SECTIONS
LOAD_ROM 0x0000
EXEC_ROM 0x0000
Input Sections
+ISR
.section
ADDR
LOADADDR
STACK
VAR
ALIGN
Group Input Section Pattern
EXEC OVERLAY ROM 0x0000 OVERLAY 0
```

Sections

```
USER_SECTIONS_vector
FLASH 0x00000000 0x00100000
EXEC 0x00000000
VAR_ILM_BASE = 0x00600000
VAR_ILM_SIZE = 0x00010000
VAR_ILM_SIZE = 0x00010000
VAR_ILM_SIZE = 0x00010000
* (+vector)
* (+RO)
SDRAM 0x80000000 0x00800000
LOADADDR NEXT _data_mlastart
ADDR NEXT _data_start
* (+RW, +ZI)
STACK = 0x80800000
```

### RTOS awareness

FreeRTOS Task List

task name	number	priority	start of stack	top of stack	status
TaskWav	0	2	0x84480	0x850720	Running
IDLE	2	0	0x851a20	0x855930	Ready
TaskEmp	1	2	0x850980	0x851870	Blocked
DMA BH	3	8	0x304f60	0x305370	Suspended

FreeRTOS Event List

queue name	handler address	max length	item size	messages waiting	waiting Tx	waiting Rx
queue	0x855e40	1	0	1	0	0
queue	0x855b00	1	0	1	0	0
queue	0x855b00	65535	0	127	0	0
queue	0x855e80	65535	0	0	0	1

### Custom plugin

Custom plugin window showing a circuit diagram and a waveform.

### Function code size

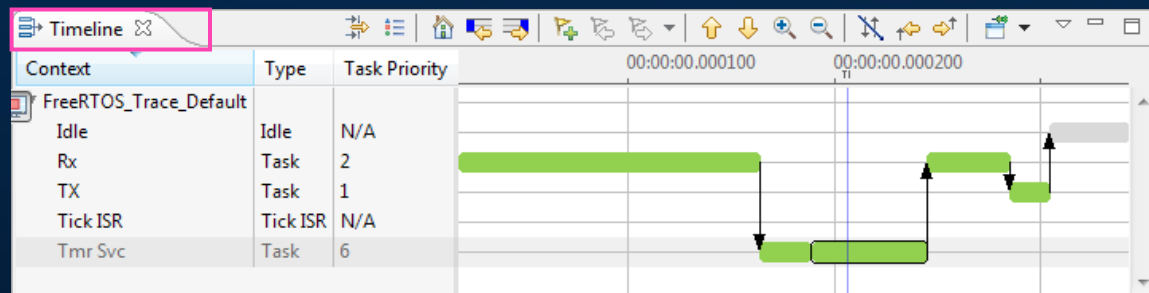
Function Code Size

Name	Size	File	Line	Path
ldc_off	24	ldc.c	59	d:\AndeSight\AndeSight\workspace\IPEG\src\ldc.c
ldc_set framebuffer	32	ldc.c	69	d:\AndeSight\AndeSight\workspace\IPEG\src\ldc.c
main	176	main.c	105	d:\AndeSight\AndeSight\workspace\IPEG\src\main.c
make_runny_pointers	108	quant.c	396	d:\AndeSight\AndeSight\workspace\IPEG\src\quant.c
master_selection	464	master.c	288	d:\AndeSight\AndeSight\workspace\IPEG\src\master.c
median_cut	252	quant.c	424	d:\AndeSight\AndeSight\workspace\IPEG\src\quant.c
merged_lv_upsample	193	ldc.c	193	d:\AndeSight\AndeSight\workspace\IPEG\src\ldc.c
merged_2v_upsample	164	ldc.c	144	d:\AndeSight\AndeSight\workspace\IPEG\src\ldc.c



# FreeRTOS Timeline Analyzer

- Visualize the runtime execution behaviors of task, interrupts and events within a period of time<sup>1</sup>



The screenshot displays the 'FreeRTOS\_Trace\_Default' window. It shows a list of events with columns for Timestamp, Context, Context Type, Events, and Details. The events include task switches and queue operations.

Timestamp	Context	Context Type	Events	Details
<srch>	<srch>	<srch>	<srch>	<srch>
00:00:00.000 164	Tmr Svc	Task	Task Switch In	Task 'Tmr Svc' context switch in and get running
00:00:00.000 189	Tmr Svc	Task	Task Switch In	Task 'Tmr Svc' context switch in and get running
00:00:00.000 202	Tmr Svc	Task	xQueueRecv	Queue='TmrQ', Status=SUCCESS
00:00:00.000 219	Tmr Svc	Task	xQueueRecv	Queue='TmrQ', Status=FAIL, TicksToWait=0
00:00:00.000 245	Rx	Task	Task Switch In	Task 'Rx' context switch in and get running

The screenshot displays the 'Statistics' window of the FreeRTOS Trace Analyzer. It shows a table with columns for Context, Type, Run Count, Run Frequency, and Last Run Time. The statistics are for the 'FreeRTOS\_Trace\_Default' trace.

Context	Type	Run Count	Run Frequency	Last Run Time
Idle	Idle	3426	942.98 Hz	0.000991 s
Rx	Task	133	36.61 Hz	0.000984 s
TX	Task	41	11.28 Hz	0.000018 s
Tick ISR	Tick ISR	3549	976.83 Hz	0.000009 s
Tmr Svc	Task	3	.83 Hz	0.000075 s

1: EVB target only



# Multicore Development Support

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

Debugging for core 0

Debugging for core 1

debugging actions for a single core

context values when debugging core1

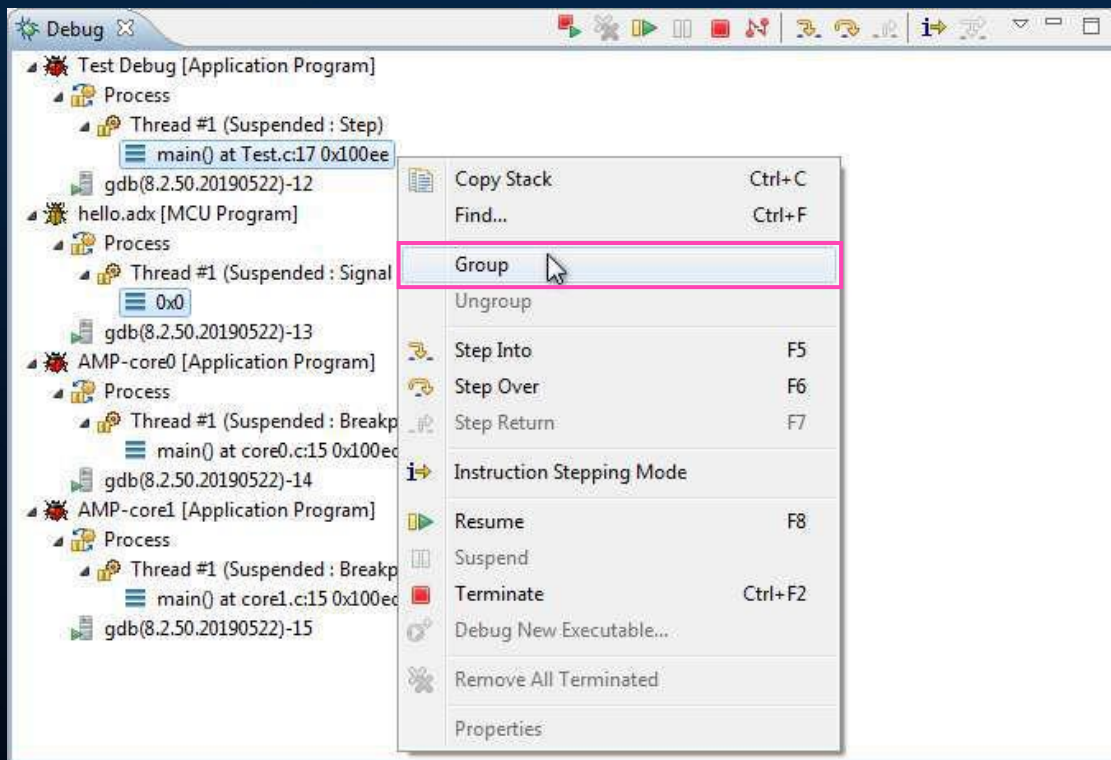
```
16 volatile int i = 0, j = 0;
17 // puts("!!!Hello World!!!");
18
19 for (i=0; i < 10; )
20 {
21     if (!*flag)
22     {
23         printf("I'm Core1 !!!, %d\n", i);
24         for (j = 0; j < 0x10000; j++); //delay loop
25         i++;
26         *flag = 1;
27     }
28 }
```

Name	Value	Address	Description
BMC			AXI Bus Inte
AHB decoder			AHB-Lite de
Ethernet MAC			Ethernet M
ISR	0x0	0xe0100000	Interrupt Statu
IME	0x0	0xe0100004	Interrupt Enabl
MAC_MADR	0x0	0xe0100008	MAC Most Sig
MAC_LADR	0x0	0xe010000c	MAC Least Sig

# Core Grouping

## ■ Grouping Cores for Efficient Debugging

- Debug commands can be sent to a specific set of cores at the same time.

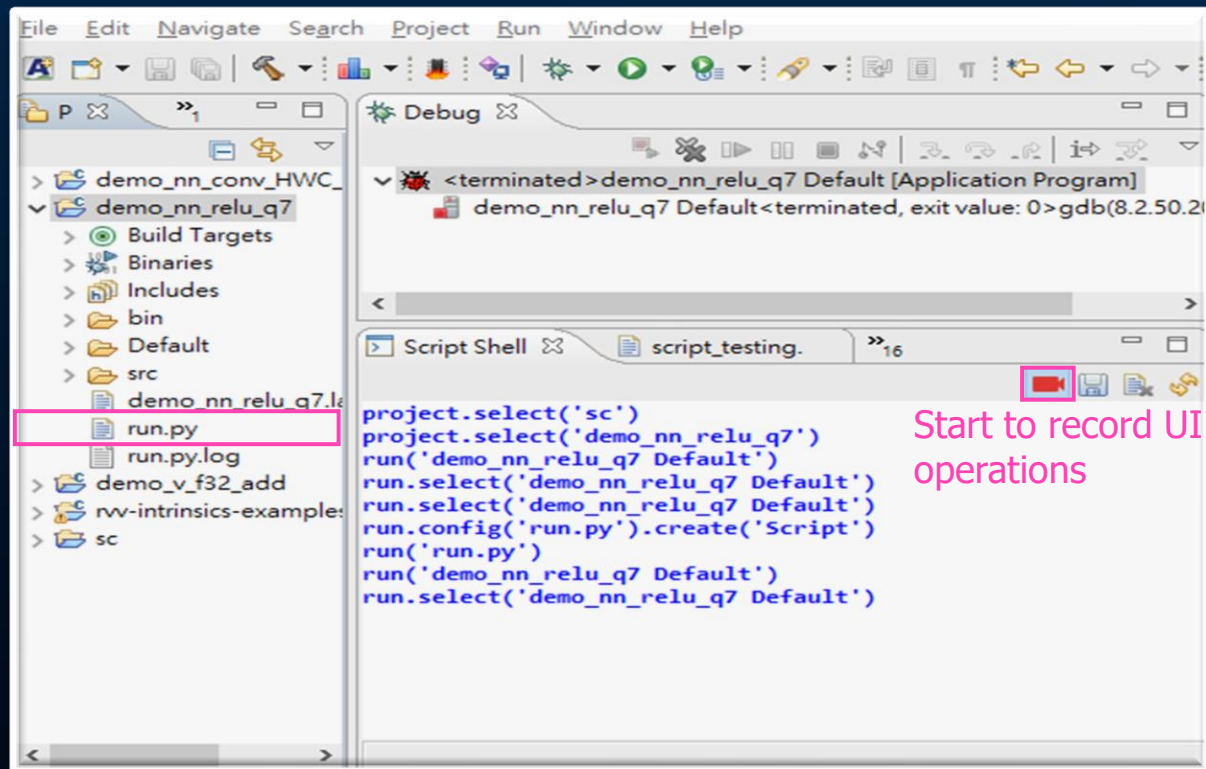




# AndeSight™ Scripting

## ■ Record and replay UI operations with Python script

- Test automation
- Issue reproduction
- ...





# AndeSoft™ BSP

Application Building Blocks Inside of AndeSight™ IDE

## 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<sup>1</sup>**
- 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 AndeCore™ 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**



# AndeSoft™: Bare Metal

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

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

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



## ■ 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\*

fedora

debian

yocto  
PROJECT

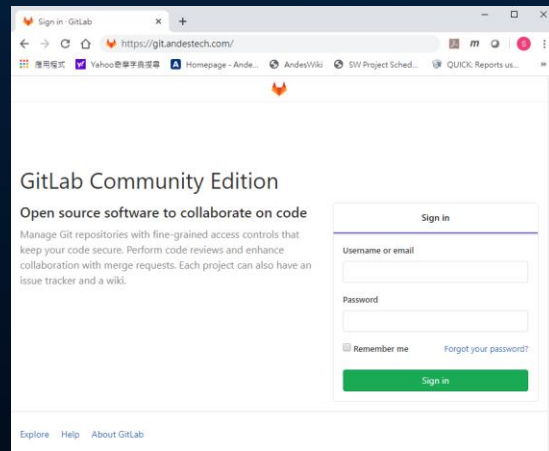
OpenWrt  
Wireless Freedom

BuildRoot  
Making Embedded Linux Easy

\*: available upon request

## ■ Andes GitLab Service for Linux Development Packages

- A public web service: [git.andestech.com](https://git.andestech.com)
- Under Andes Maintenance Program



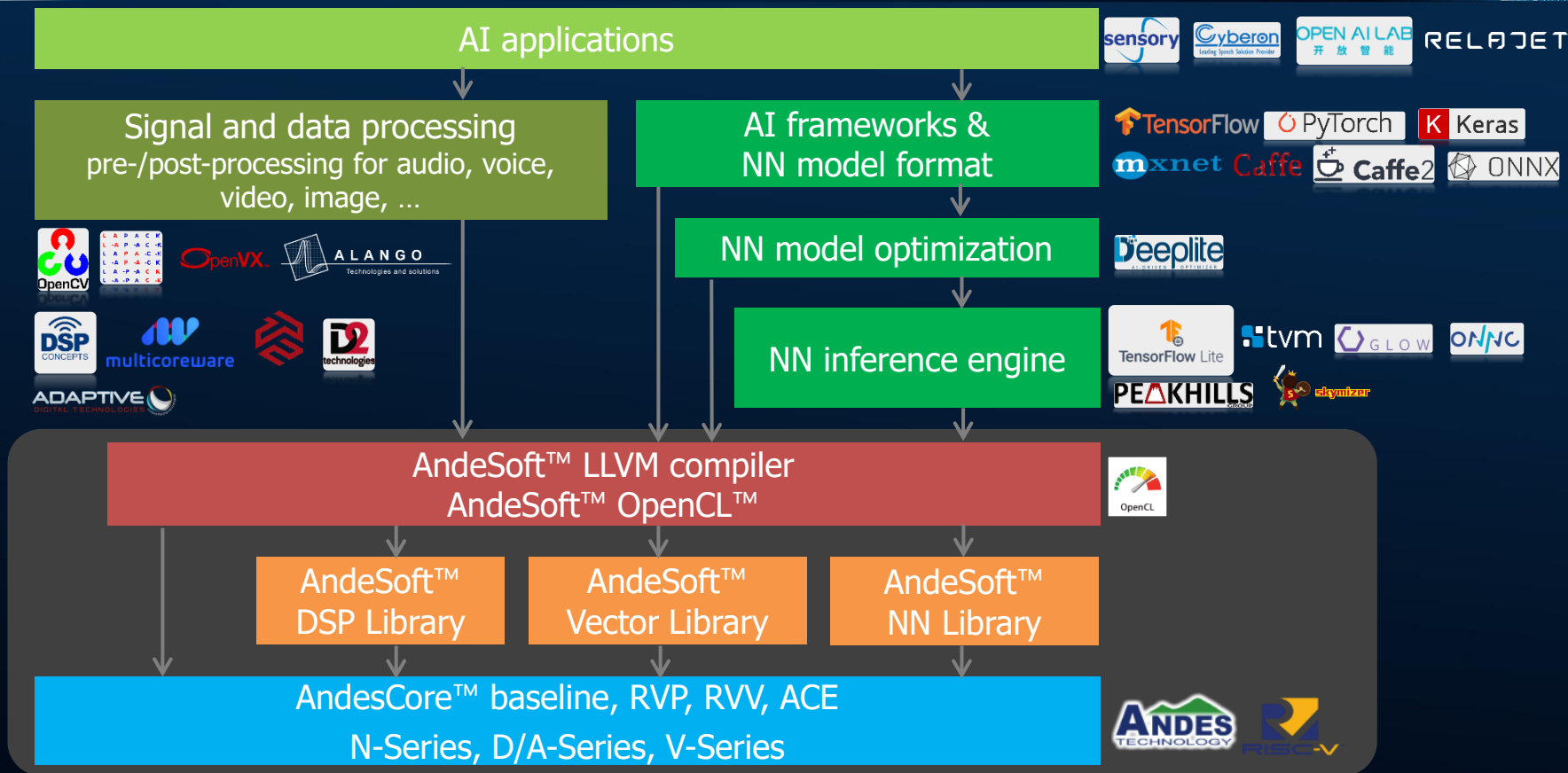
GitLab



# AI Software Stack

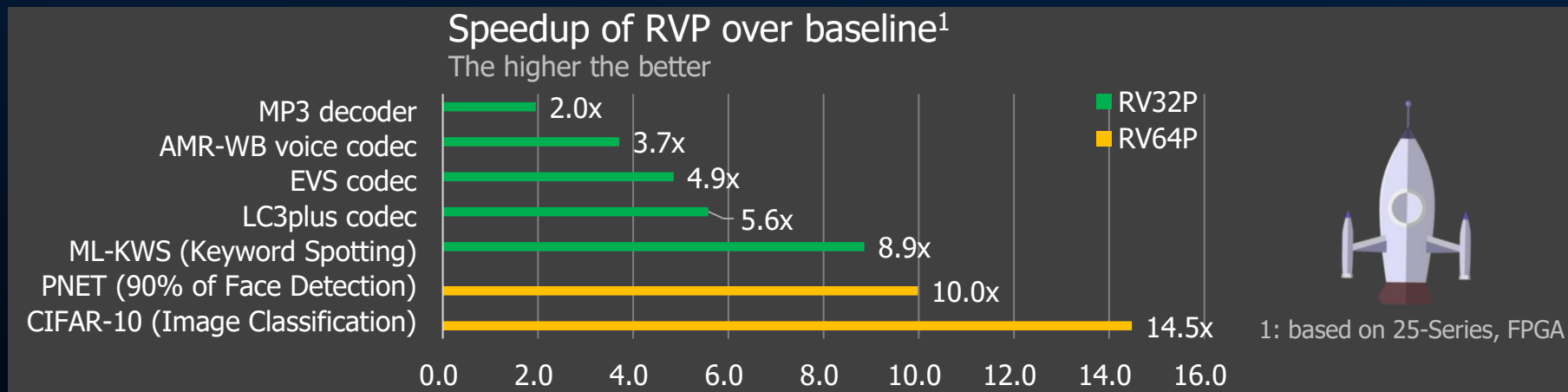
## DSP, Vector and NN Programming

# AI Software Stack and Ecosystem



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

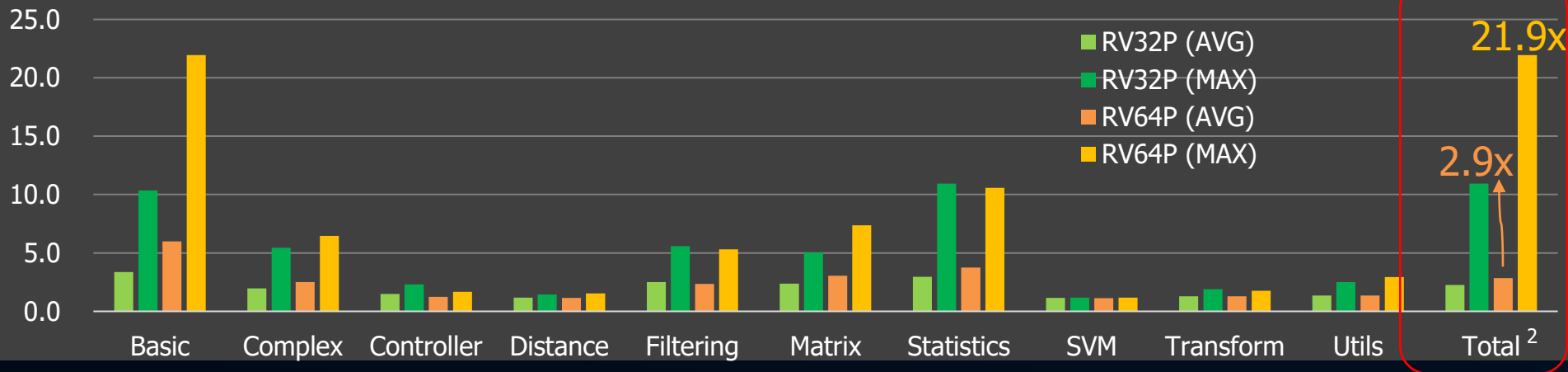
- **DSP intrinsic functions**: as C-like functions without writing error-prone assembly
- **AndeSoft™ DSP library**: ~300 functions in 10 categories
  - Basic, complex, controller, distance, filtering, matrix, statistic, SVM, transform, utils
- Increase power efficiency to your DSP applications: **2.0x-5.6x**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



- 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<sup>1</sup>

The higher the better







# RISC-V Vector Software Solutions

## ■ Standard software solutions in AndeSight™ IDE

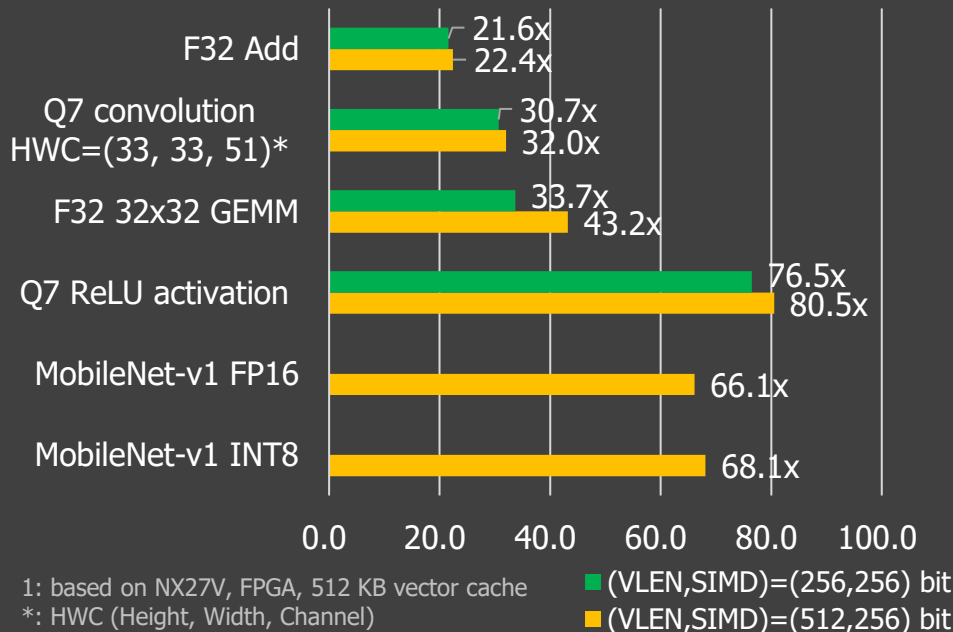
- AndeSim™: new cycle-accurate simulator
- Toolchains: **intrinsic functions**, as C-like functions without writing error-prone assembly
- AndeSoft™ Vector library:
  - Optimized for RISC-V **baseline** and **RVV** processors
  - > 100 functions in 5 categories: basic, filtering, image, matrix, and transform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

## ■ Advanced software solutions

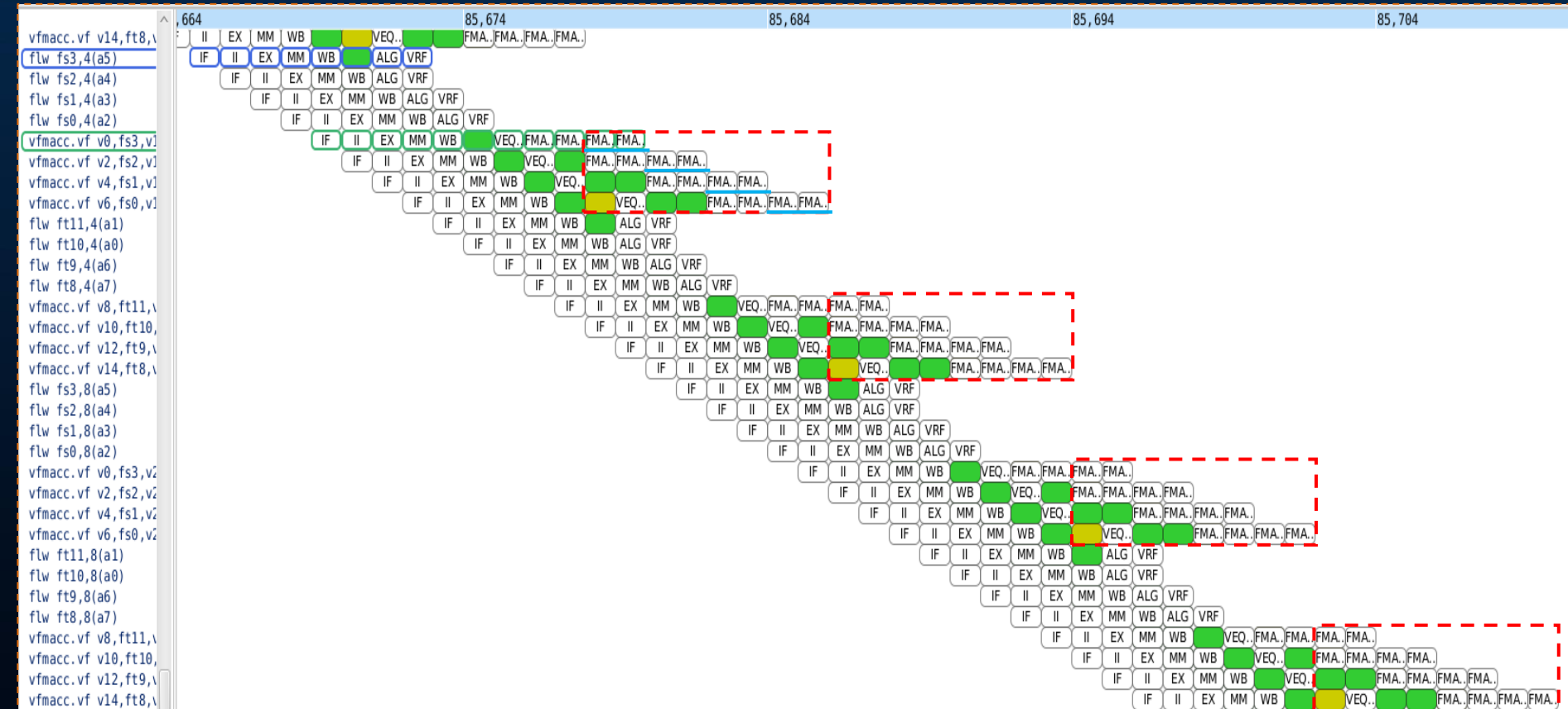
- AndeSysC™: SystemC support for AndeSim™
- AndesClarity™: GUI-based processor pipeline visualizer and analyzer
- AndeSoft™ NN library: optimized neural network functions

### Speedup of RVV over baseline<sup>1</sup>

The higher the better



# AndesClarity™ Processor Pipeline Analyzer



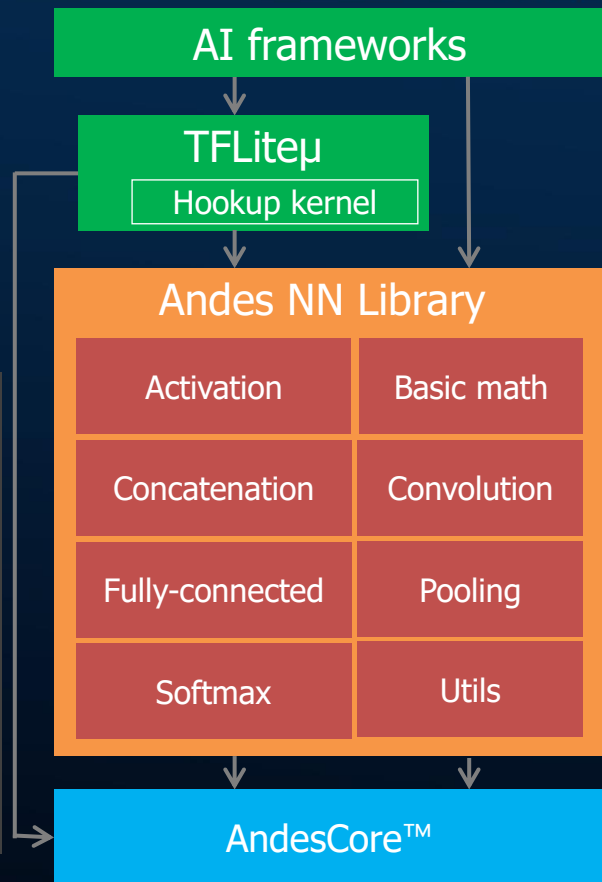
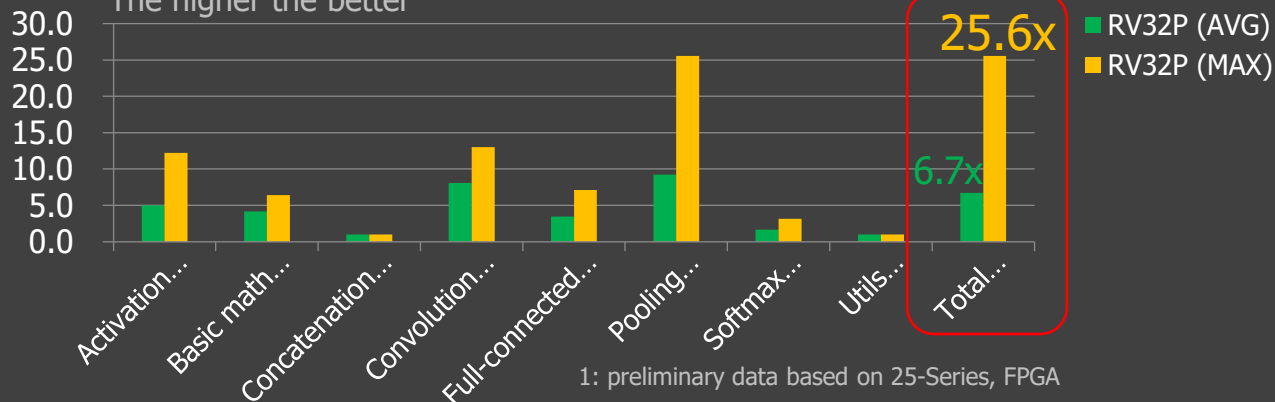


# AndeSoft™ NN Library

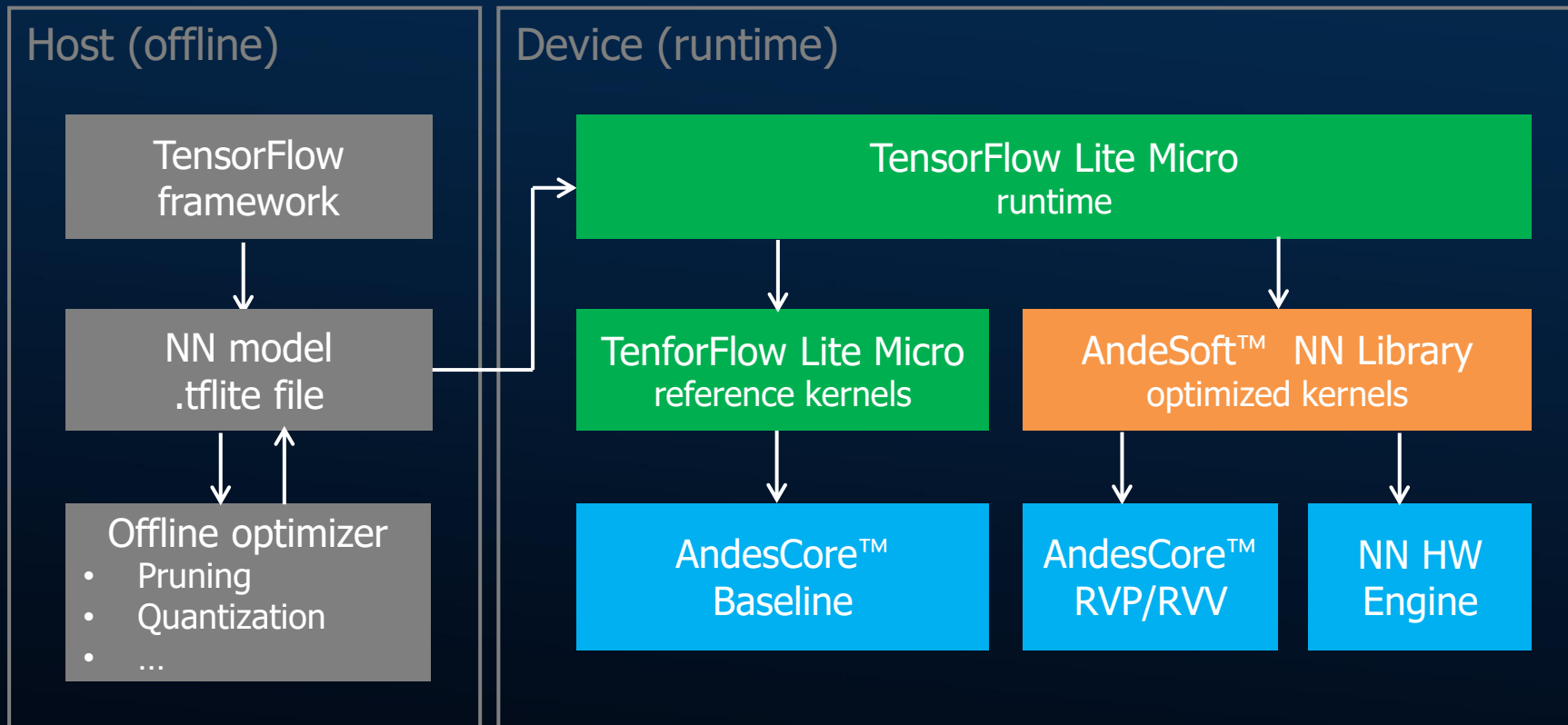
- Optimized neural network functions for **Pure-C**, **RVP**, and **RVV** processors
- Boost NN performance by using SIMD and vector instructions
- >120 functions in 8 categories
- Compatible with CMSIS-NN APIs

## Speedup of RVP over baseline<sup>1</sup>

The higher the better



# Inference Flow with TensorFlow Lite Micro





# AndeSoft™ NN Library API



Categories	Operators	NN Library API <sup>1</sup>
<b>Activation</b>	Sigmoid	riscv_nn_activate_[q7 q15]
	Tanh	
	ReLU	riscv_nn_relu_[q7 q15]
	Leaky ReLU	riscv_nn_leaky_relu_q7
<b>Basic</b>	Element-wise	riscv_nn_ew_add_s8 riscv_nn_ew_mul_s8
<b>Concatenation</b>	Concatenation	riscv_nn_concate_s8_[w x y z]
<b>Convolution</b>	Convolution	riscv_nn_conv_HWC_[q7 u8 s8 q15]
	Pointwise convolution	riscv_nn_conv_1x1_HWC_[q7 u8 s8]
	Depthwise convolution	riscv_nn_conv_dw_HWC_[q7 u8 s8]
<b>Fully-connected</b>	Fully-connected	riscv_nn_fc_[q7 u8 s8 q15]
<b>Pooling</b>	Maximum pooling	riscv_nn_maxpool_[q7 s8]_HWC
	Average pooling	riscv_nn_avepool_[q7 s8]_HWC
<b>Softmax</b>	Softmax	riscv_nn_softmax_[q7 u8 s8 q15]
<b>Utils</b>	Reshape	riscv_nn_reshape_s8

1: including but not limited to the list



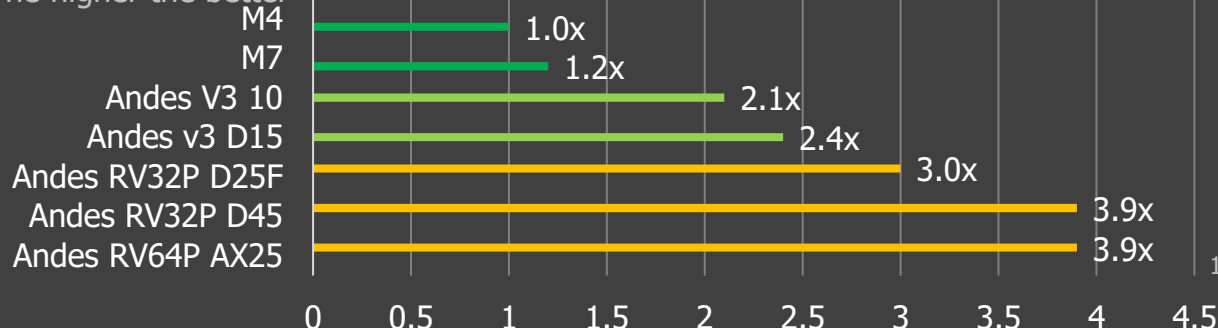
# Use Case: CIFAR-10 NN Model

CIFAR-10	Operators	NN Library APIs
layer 1	Convolution	riscv_nn_conv_HWC_q7_RGB_fast
	Activation (ReLU)	riscv_nn_relu_q7
	Pooling (maxpool)	riscv_nn_maxpool_q7_HWC
layer 2	Convolution	riscv_nn_conv_HWC_q7_fast
	Activation (ReLU)	riscv_nn_relu_q7
	Pooling (maxpool)	riscv_nn_maxpool_q7_HWC
layer 3	Convolution	riscv_nn_conv_HWC_q7_fast
	Activation (ReLU)	riscv_nn_relu_q7
	Pooling (maxpool)	riscv_nn_maxpool_q7_HWC
layer 4	Fully-connected	riscv_nn_fc_q7_fast
layer 5	Softmax	riscv_nn_softmax_q7

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

## Speedup of CIFAR-10 NN model for image classification<sup>2</sup>

The higher the better



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

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

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

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
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
Avrg 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%

知乎 @月露



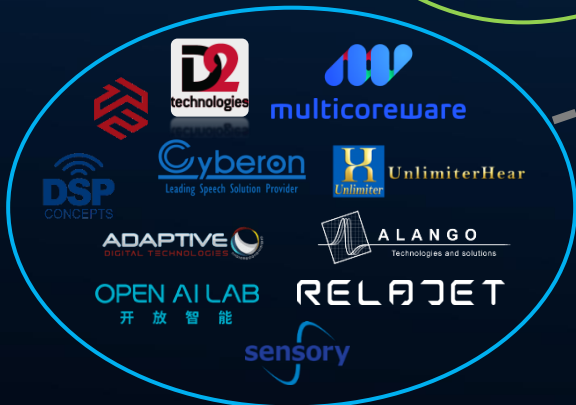
# Andes Ecosystem

# Andes Partners and Ecosystem

AI tools, SW and IP



Security



DSP, Audio and Vision

Development tools

RTOS



# IAR Embedded Workbench for RISC-V

- Complete build and debug toolchain for RISC-V
- Support **all series of Andes RISC-V CPU<sup>1</sup>**
- Functional Safety Certified
  - IEC 61508
  - ISO 26262
  - EN 50128
  - IEC 62304



## IDE

Embedded Workbench



## Debugger

I-Jet ICE

C-SPY tool



## Toolchain

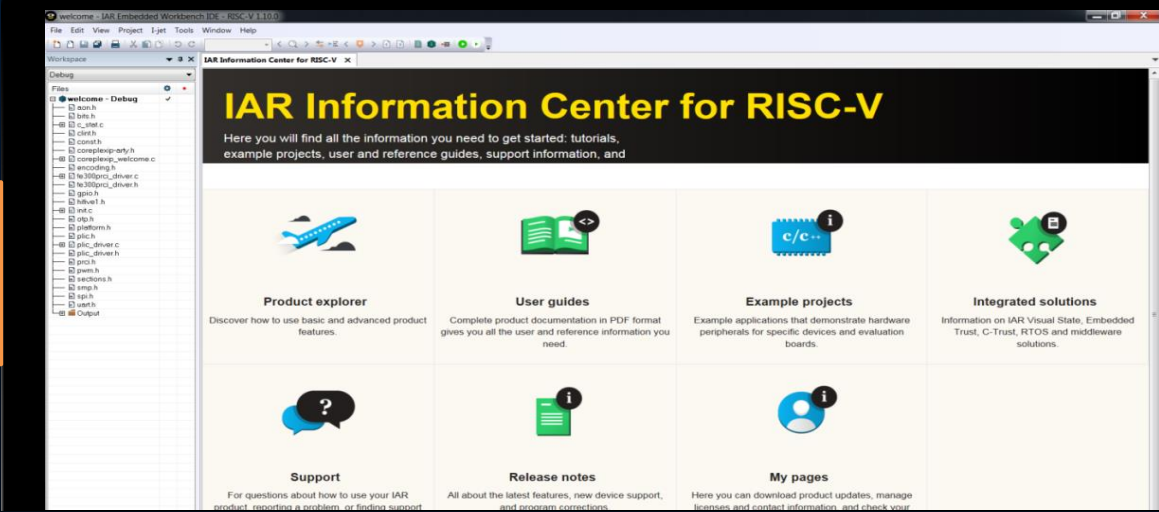
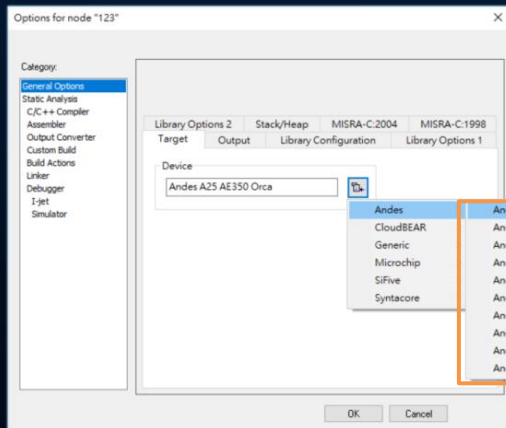
Compiler

C-STAT

Functional Safety

1: AndesCore™ 45-Series will be available in the mid-year of 2021

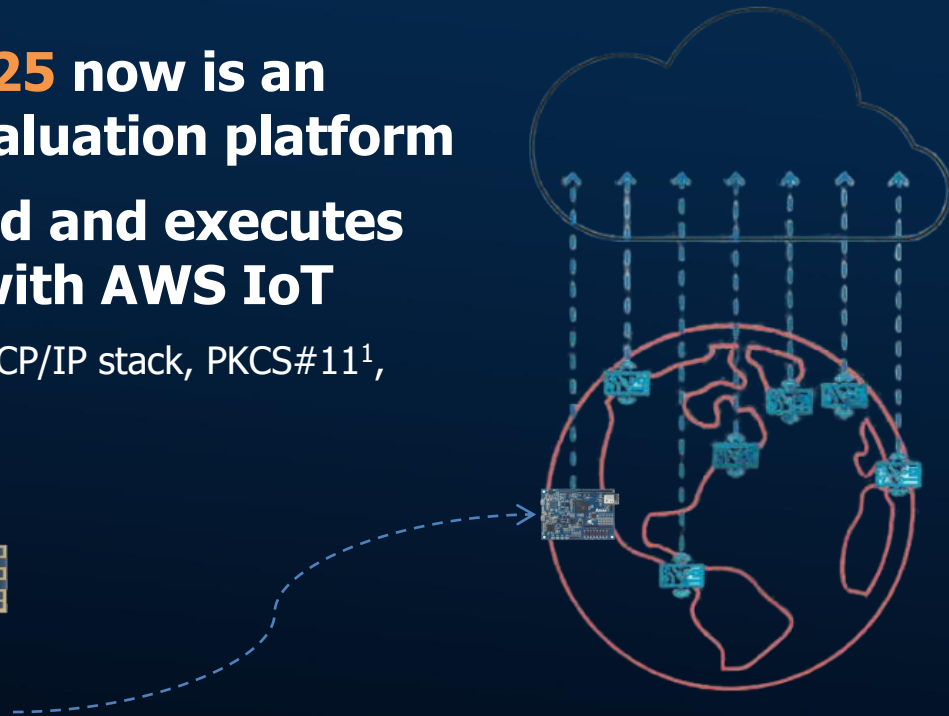
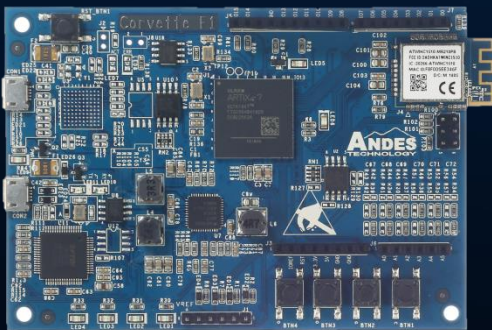
2: Part of Andes V5 RISC-V extensions are supported in the mid-year of 2021



Subject to change without notice  
Copyright© 2021 Andes Technology

# 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<sup>1</sup>, TLS<sup>2</sup>, MQTT<sup>3</sup>



qualified  
device  
Amazon FreeRTOS



- 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







- Previous version for evaluation
- Three-month time limit



**Thank you**