ANDES TECHNOLOGY

Driving Innovations™

Taking RISC-V to Mainstream ASICs

Charlie Su, Ph.D.
CTO and SVP of R&D
2017/05/09

Biography, Dr. Charlie Hong-Men Su 蘇泓萌

- **Technical Areas:**
  - Architecture and HW/SW Interaction of Processors and SoC's
  - SoC Design for Multimedia and Networking
- **Experience:**
  - **Andes Technology,** 2005: Cofounder, CTO and SVP of R&D
  - **Faraday Technology**, 2003: Principal Architect for ARM and DSP cores
  - **Afara/Sun**, 2000: Senior Staff for Niagara T1000/T2000 processors, a 32/64-thread 8-core 64-bit Ultrasparc server-on-chip
  - **C-Cube**, 1996: Director of Architecture/Validation for leading MPEG codec's
  - **SGI/MIPS**, 1993: Architecture/Verification group for 64-bit MIPS R10K (4-way out-of-order processor) and its follow-on
  - **Intergraph**, 1991: Architecture group for Clipper C4 superscalar and C5 VLIW processors
- **Education:**
  - **PhD, Computer Science, Univ. Illinois at Urbana-Champaign (UIUC)**
  - **MS, Computer Science, National Tsing-Hua University (NTHU)**
  - **BS, Electrical Engineering, National Taiwan University (NTU)**

Taking RISC-V to Mainstream ASICs

Agenda –

- Introduction to Andes
- Highlights of Andes Processor Solutions
- AndeStar™ Architecture and V5
- New AndesCore™ NX25
- Concluding Remarks

Introduction to Andes

Overview of Andes Technology Corporation

Andes Mission

- Innovate **performance-efficient** processor IP Solutions

Andes Highlights

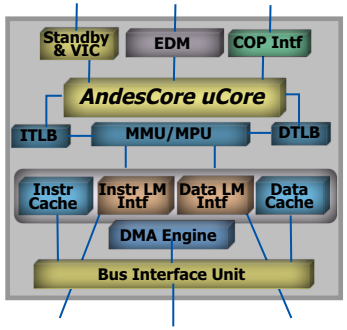
- Founded in 2005 in **Hsinchu Science Park**, Taiwan
- Core R&D team from **AMD, DEC, Intel, MIPS, nVidia, Sun**
- EETimes' Silicon 60 **Hot Startups to Watch** (2012)
- **TSMC OIP Award** "Partner of the Year" for New IP (2015)
- A founding member of **RISC-V Foundation** (2016)
- **IPO on TWSE** in March 2017

Comprehensive Processor IP Solutions

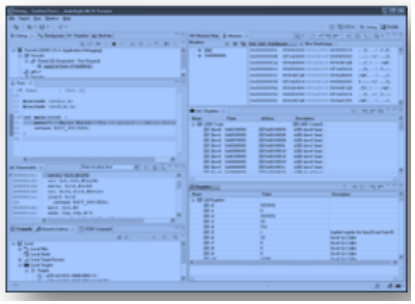
Processor Architecture
AndeStar™ (V3)

```
smw.adm  $r1,[$sp],$r5,0x0
smw.adm  $sp,[$sp],$sp,0x2
addi     $sp,$sp,-8
sethi    $r1,0x50a
lwi      $r1,[$r1+#0x98]
mov55    $r2,$r0
mov55    $r0,$r1
lwi      $r1,[$r1+#0x8]
addi     $r3,$sp,12
```

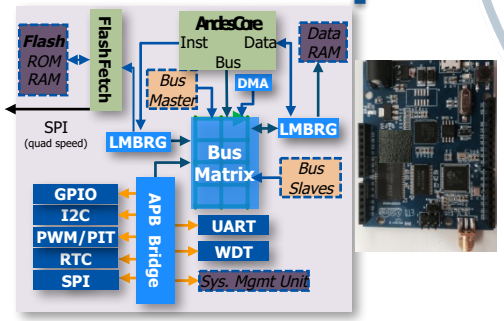
Processor IP's
AndesCore™



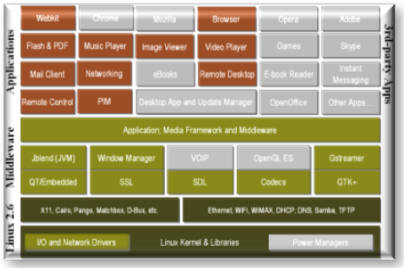
Development Tools
AndeSight™



Development Platforms
AndeShape



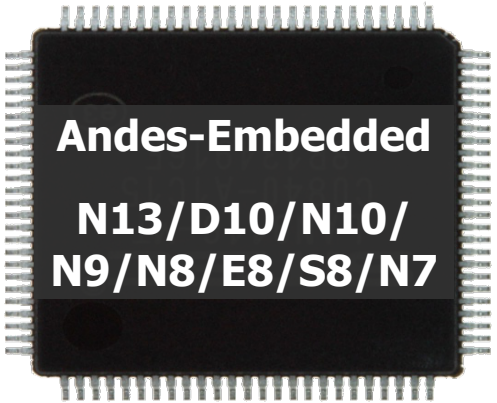
SW Stacks
AndeSoft™



Andes Embedded™

Business Status Overview

- **Over 120 commercial licensees**
  - Taiwan, China, Korea, Japan, US, Europe
  - >2B Andes-Embedded™ SoCs shipped



- **AndeSight™ IDE:**
  - >11,000 installations

- **Ecosystem**
  - >100 partners

- **Diversified applications based on Bare Metal, RTOSes, and Linux**

Executive Summary

- New-generation AndeStar V5 adopts RISC-V
  - As its architecture kernel
- AndesCores expand from 32 bits to 64 bits
  - Based on AndeStar V5 architecture
- Andes brings rich processor solutions to RISC-V

Andes is the 1st major CPU IP vendor adopting RISC-V

Introduction to Andes (Devices with Andes Embedded™)

Andes Embedded™ in SmartPhones

WiFi/Bluetooth/ GPS/FM (Combo)
- N968

Touch screen controller
- N705
- N801
- N968

NFC
- N968

Sensor Hub
- N801
- N968

eMMC controller
- N801

Andes Embedded in IOT and Sensor Fusion

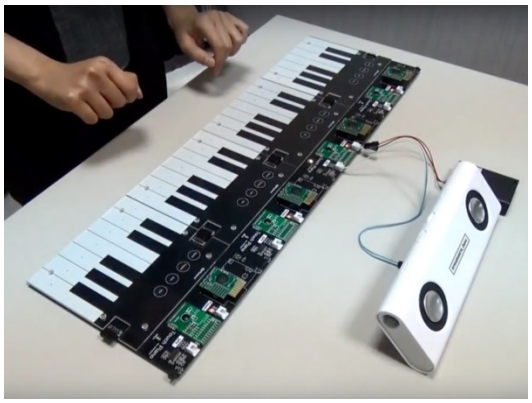
WiFi Chips for IoT

(also 802.15.4, Cat-M/NB-IOT)

Sensor Fusion chips are used in Notebook PCs from Acer, Asus, HP, Lenovo

Andes Embedded with N7, N8, N9, N13

Mastech MS6531 IR Thermometer: ADC MCU



Nyquest Speech Synthesizer: MCU

Nisan X-Trail: ADAS Ctlr

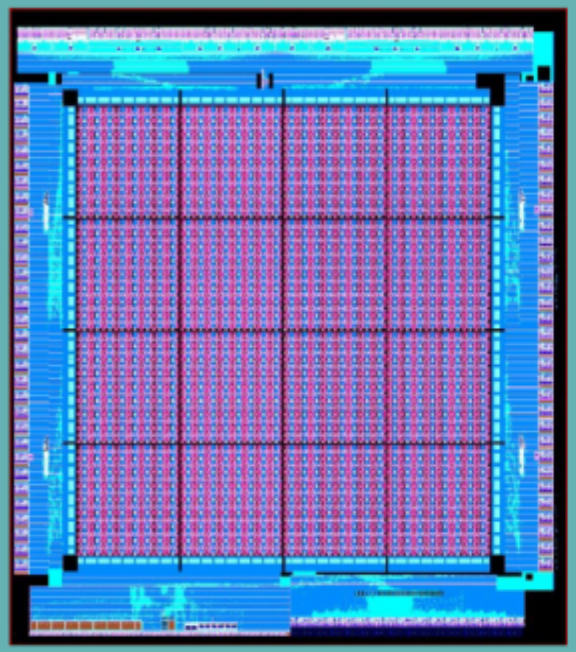
Toshiba SD Card: Flash Ctlr

AndesCore is Learning

Waue Computing

Dataflow Processing Unit (DPU) Architecture

Scalable Machine Learning Computers for Data Center



16ff CMOS Process Node	16K Processors, 8192 DPU Arithmetic Units	Self-timed, MPP Synchronization
181 Peak Tera-Ops	16 MB Distributed Data Memory	8 MB Distributed Instruction Memory
1.71 TB/s I/O Bandwidth	270 GB/s Peak Memory Bandwidth	2048 Outstanding Memory Requests
4 Billion 16-Byte Random Access Transfers/s	4 Hybrid Memory Cube Interfaces	2 DDR4 Interfaces
PCIe Gen3 16-lane Host interface	32-b Andes N9 MCU	1 MB program store for paging
Hardware engine for fa[illegible] of AES encrypted progr	32 Dynamic Reconfiguration Zones	Variable fabric dimensions (user programmable at boot)

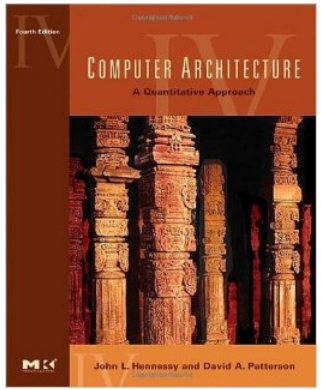
AndesCore as System Ctl Processor to control their 16,384 tiny processors

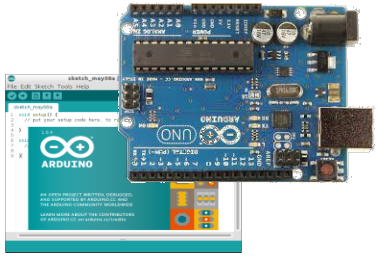
Highlights of
Andes Processor Solutions

Supporting to Reach 2-Billion Units

Andes Products and Open Source

- **Several Andes products are built on open source and prior art:**
  - AndeSight IDE: Eclipse, GCC, LLVM, GDB/OpenOCD
  - AndeSoft Stack: FreeRTOS/OpenRTOS, eCos, Contiki, Linux, middleware
  - AndeShape Andino boards: Arduino-compatible
  - AndeStar ISA: RISC architectures





- **Our innovations started, not stopped, here**

AndesCore™ N/D/E/S Series

Novel

Novel processors with high efficiency, PowerBrake, StackSafe™, CoDense™ *(N7, N8, N9, N10, N13, N15)*

Dsp

DSP-capable processors with cost-efficient pipeline *(D10, D15)*

Extensible

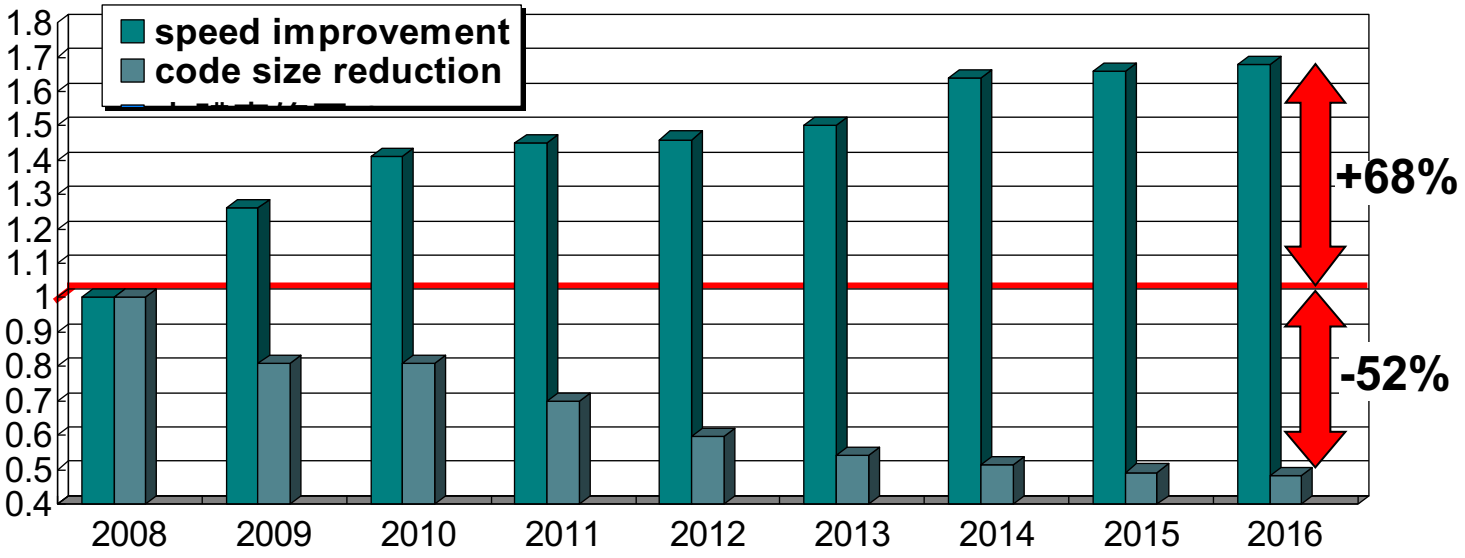
Extensible processors for application-specific acceleration and code security *(E8)*

Secure

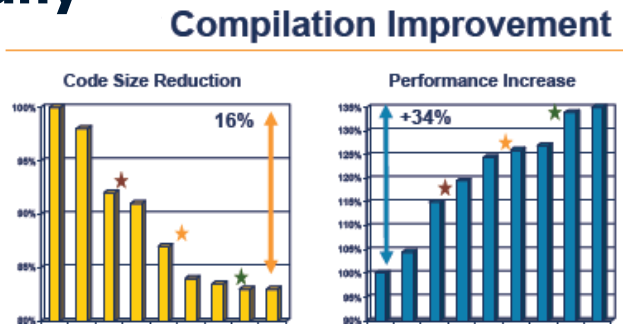
Security processors for best protection *(S8)*

Advancement in Compiler Optimizations

- **Andes compiler improvement measured by EEMBC:**

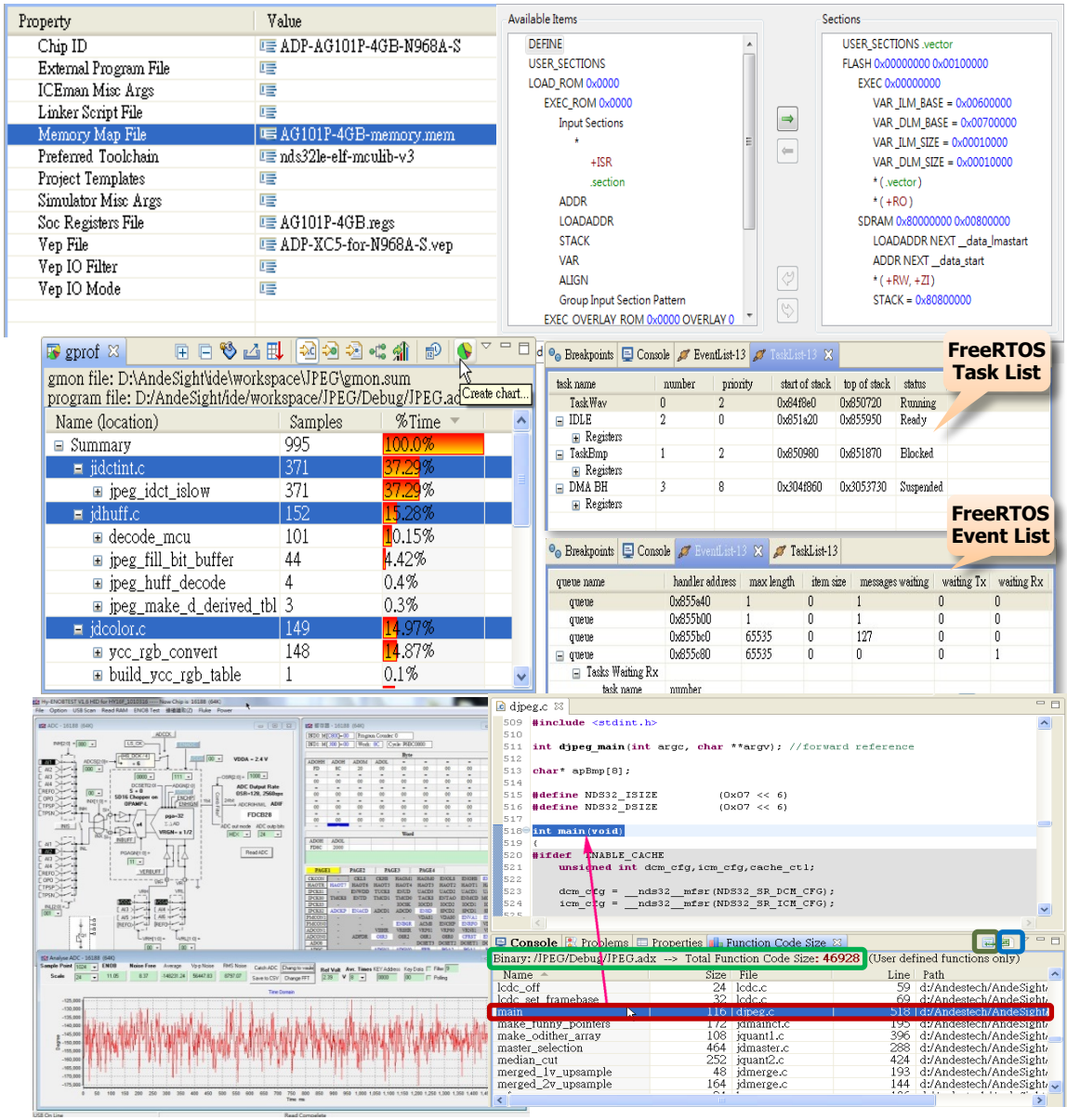


- **9-year overall improvement vs. A-Company**
  - Speedup: Andes +68% verses +34%
  - Code size: Andes -52% verses -16%
- **Today, Andes has about**
  - 40% higher performance efficiency
  - 20% smaller code size



AndeSight IDE: Rich Features

- **Project Setup:**
  - Linker Script Editor
  - Flash ISP
- **Debug Support:**
  - RTOS-Awareness
  - Registers w/ Bitfield View
- **Program Analysis**
  - Function Profiling
  - Code Coverage
  - Performance Meter
  - Function Code Size
  - (Static) Stack Size
- **Custom Plugin Intf**



Andes Custom Extension™ (ACE)

Verilog user.v

concise RTL

operands,
C semantics,
test-case spec

ACE script

COPILOT

Custom-OPtimized Instruction deveLOpment Tools

Automated Env. For Cross Checking

Test Case Generator

Extended RTL

Extended ISS

Extended Tools + Compiler Asm/Disasm Debugger IDE

Extended ISS + CPU ISS (near-cycle accurate)

Extended RTL + CPU RTL

Extensible Baseline Components

Executable or library

Source file

ACE Features

Items	Description	
Instruction	**Scalar**	- Single-cycle - Multi-cycle (interruptible or non-interruptible)
	Vector	- 'for' loop - 'do while' loop
	Background	- Issued and retired immediately from CPU pipeline, but continue the remaining execution in background
Operand: **Explicit or Implied**	**Standard**	- Immediate constant - GPR (up to 3R2W) - Baseline memory (accessed thru CPU)
	Custom	- ACR (ACE Register) - ACM (ACE Memory) - ACP (ACE Port)
Auto-Generation	- Opcode selection (optional) - All required tools/simulator with fast turnaround time - RTL for instruction decoding, operand mapping and accesses, dependence checking, and result gathering.	

Maturity and Stability

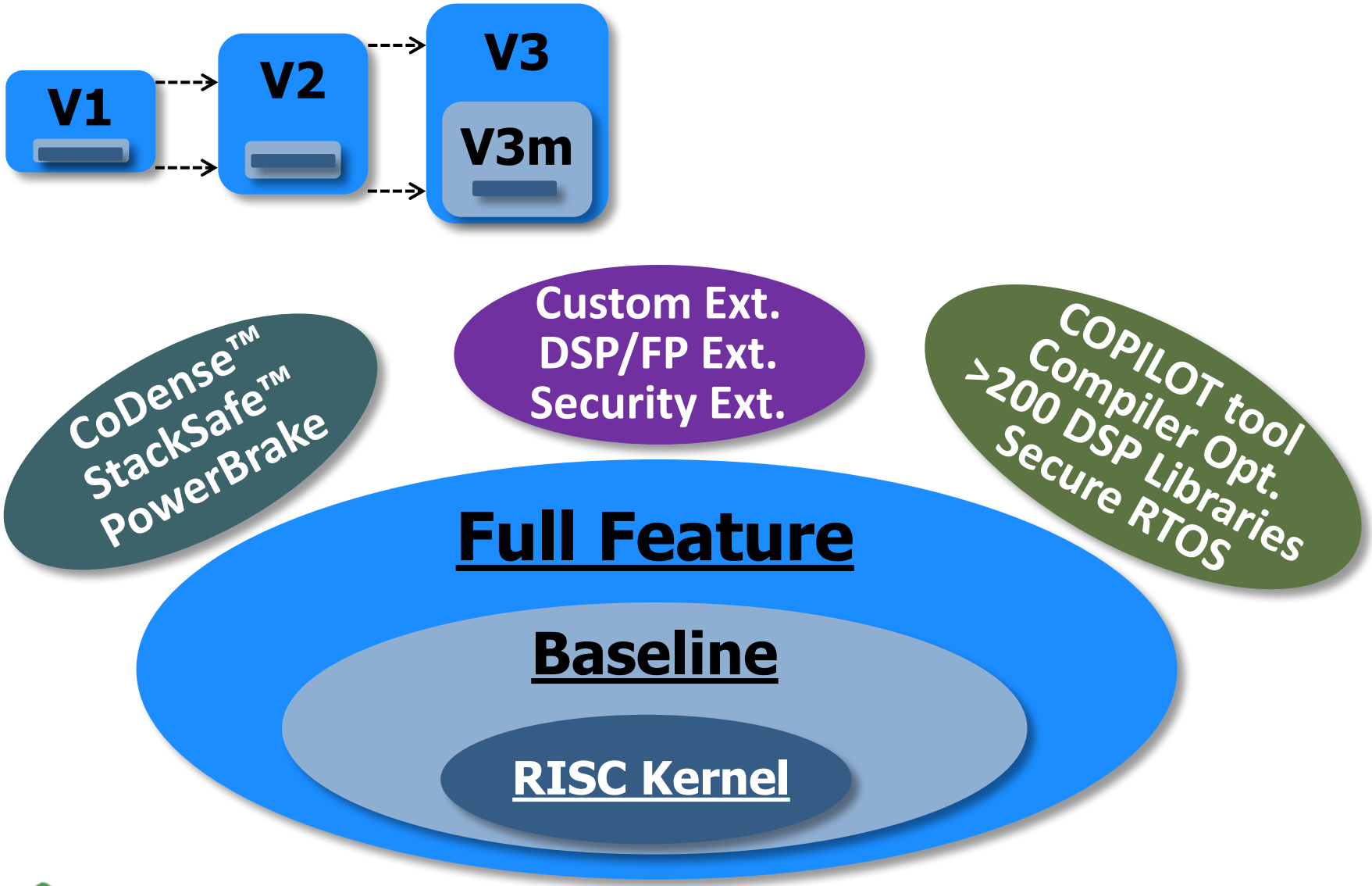
- Silicon-proven and mass production records
- It's about maturity and stability
  - ➔Most users don't want to upgrade w/o clear benefits
  - Open-source strategy: adopt a stable version in a managed pace
- Product verification
  - Heavy simulation and model checking for RTL design
    - ➔ Going thru standard EDA tool flow
  - Compiler test suites: open-source, commercial and in-house
  - Debugger/ICE test suites
  - AndeSight IDE: commercial GUI testing tools
  - Linux: LTP and more
- It's also about long-term commitment to IP business

AndeStar Architecture

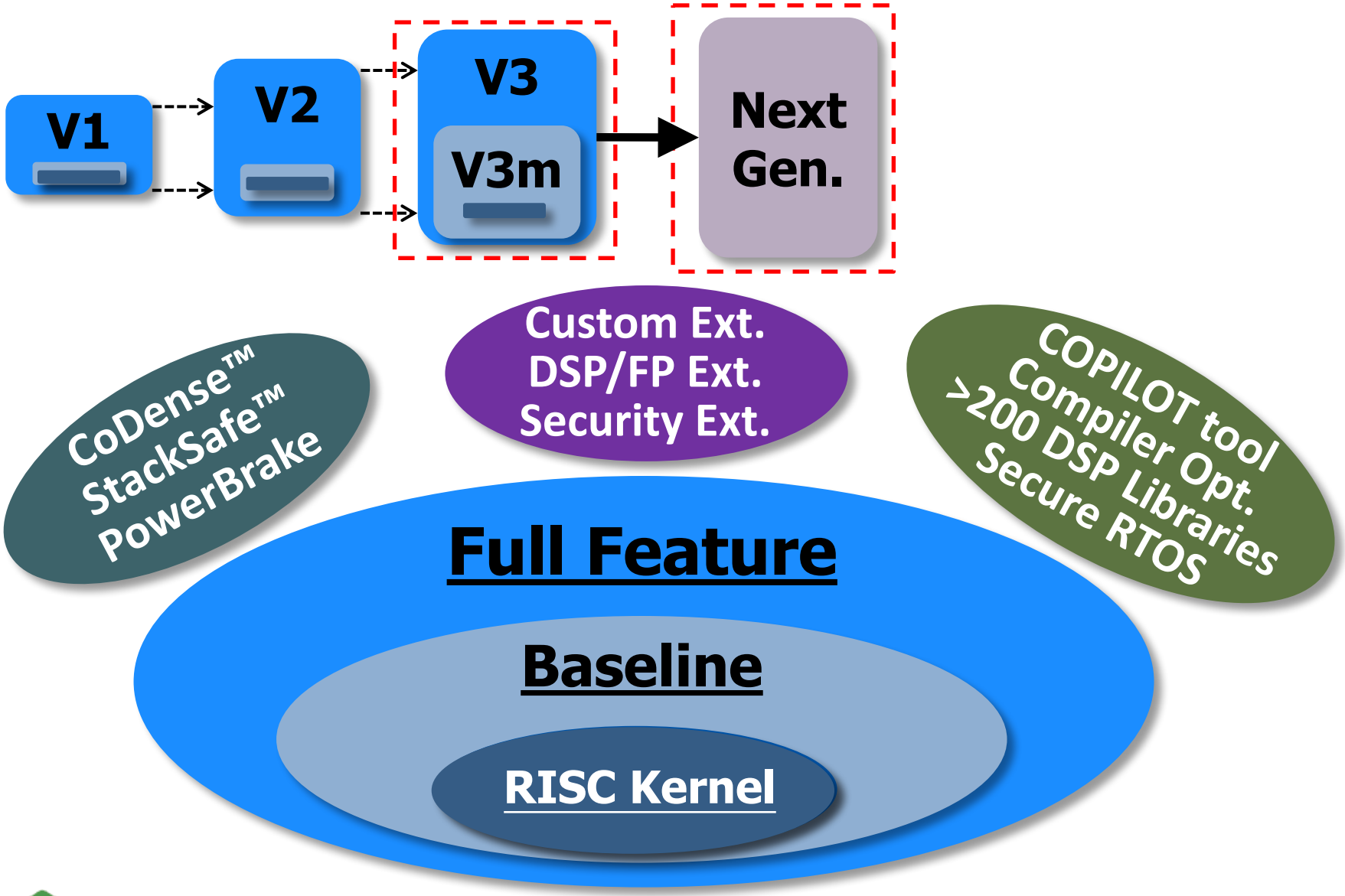
A Closer Look at AndeStar

- **Andes Sixteen and Thirty-two Architecture**
  - 16-bit and 32-bit instructions
- **RISC architecture**
- **Characteristics of AndeStar's "RISC kernel"**
  - Intermixable 16-bit and 32-bit instructions
  - 16- and 32- GPR configurations
  - No delayed branch, no predicated execution, no condition code
  - $PC isn't a GPR, and $r0 isn't hardwired to 0
  - Basic ALU, loads/stores, branches
- **Instructions with longer immediate**
- **Load/store with additional addressing modes**
- **Patented load/store multiple words**
- **More**
  - **Characteristics of major commercial RISC architectures are all different**
  - **RISC-V is most similar to AndeStar except $r0 is 0**

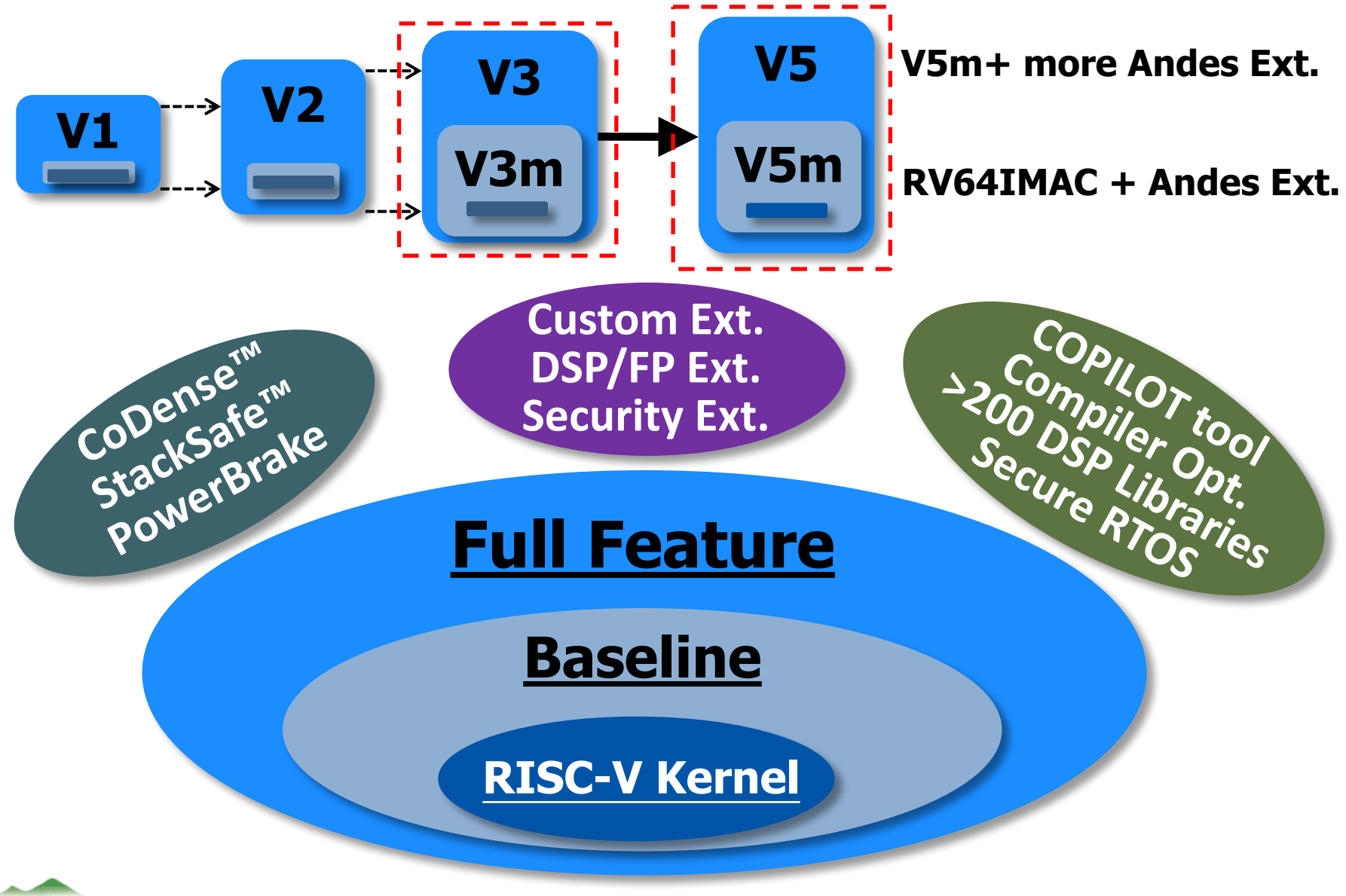
AndeStar™ ISA: V1 to V3



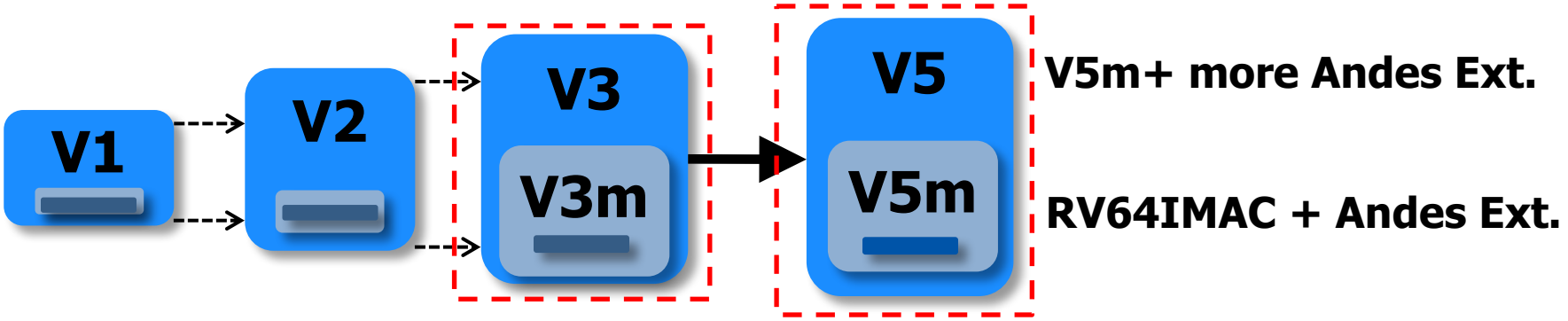
AndeStar™ ISA: Next Generation



AndeStar™ ISA: V5

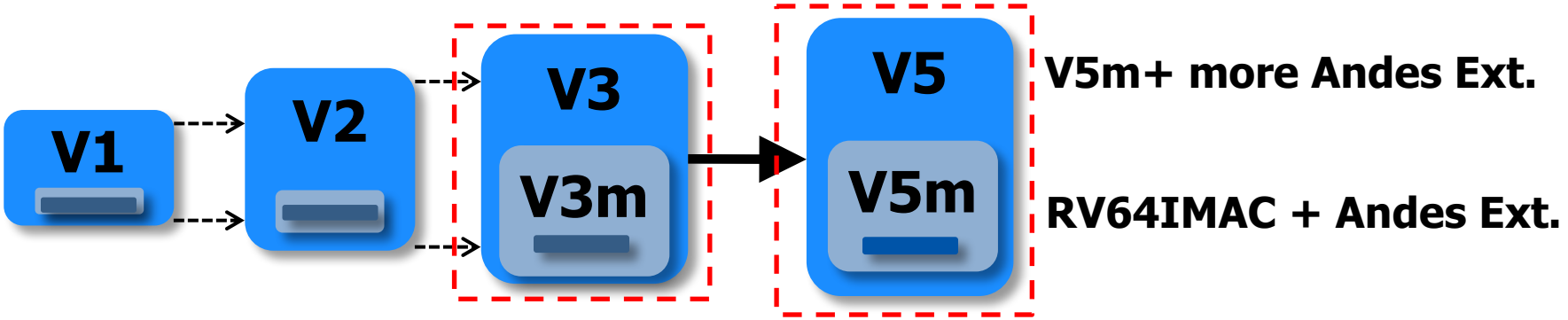


Adopting RISC-V as Natural ISA Evolution



- **AndeStar embraces RISC-V as its subset**
  - Common directions: compact kernel, modularity, extensibility, 64 bits
  - Good momentum behind the RISC-V ecosystem
  - ➔ **Ande**s **S**ixty-four and **T**hirty-two **Ar**chitecture
- **Bringing AndeStar strength to RISC-V through V5**
  - Architecture beyond the kernel for diversified requirements
  - Efficient processor pipeline for leading PPA
  - Platform IP support to help speed up SoC construction
  - AndeSight IDE, and compiler/library optimizations
  - RTOS and Linux support, and middleware (such as IoT stacks)
  - Commercial-grade verification for all products
  - Professional supporting infrastructure

Adopting RISC-V as Natural ISA Evolution



- **AndeStar embraces RISC-V as its subset**
  - Common directions: compact kernel, modularity, extensibility, 64 bits
  - Good momentum behind the RISC-V ecosystem
  - ➔ **Ande**s **S**ixty-four and **T**hirty-two **Ar**chitecture
- **Bringing AndeStar strength to RISC-V through V5**
  - Architecture beyond the kernel for diversified requirements
  - Efficient processor pipeline for leading PPA
  - Platform IP support to help speed up SoC construction
  - Andes Ha[illegible] compiler optimization
  - RTOS and Linux support, and middleware (such as IoT stacks)
  - Commercial-grade verification for all products
  - Professional supporting infrastructure

A Complete Package !

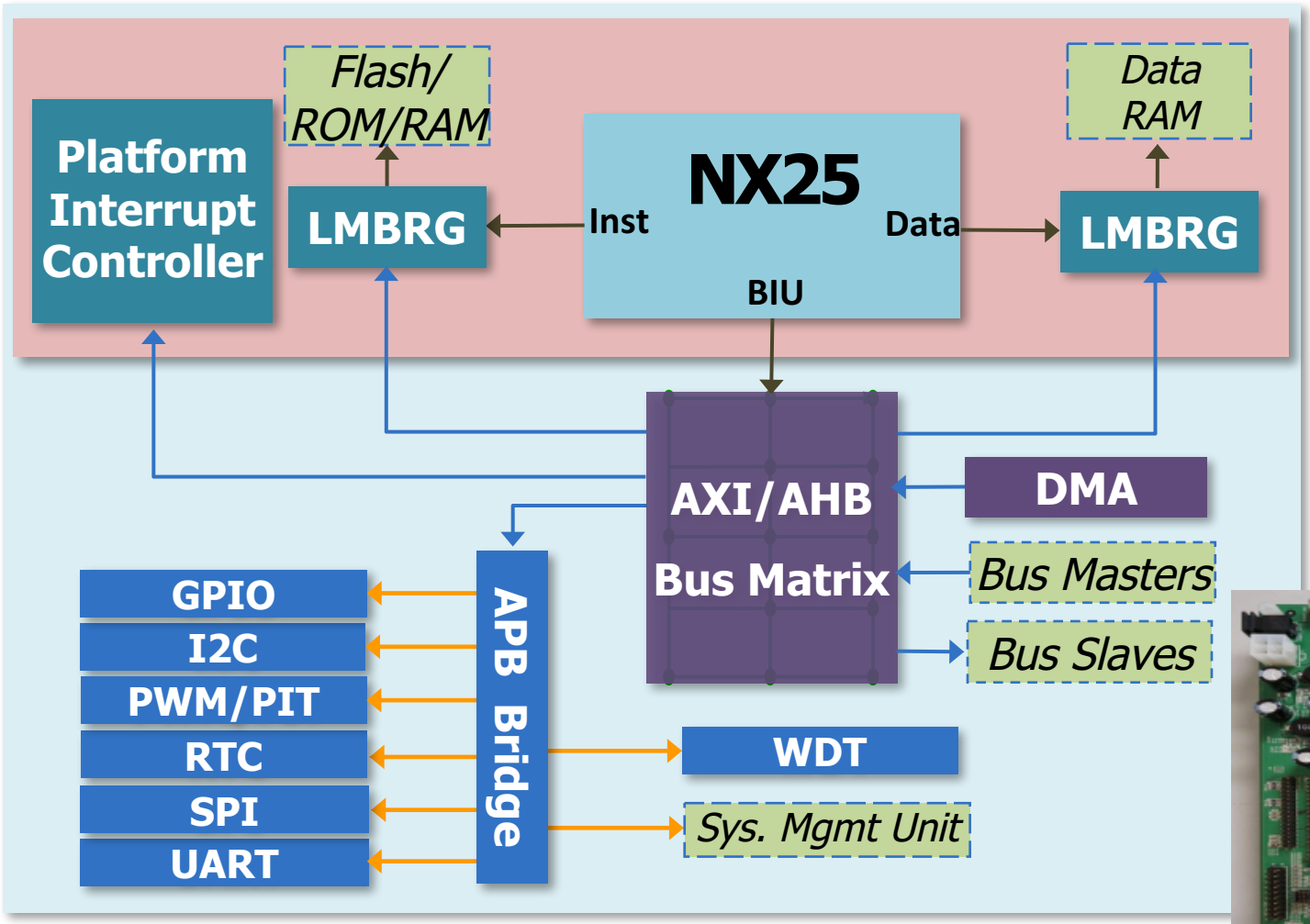
New AndesCore NX25

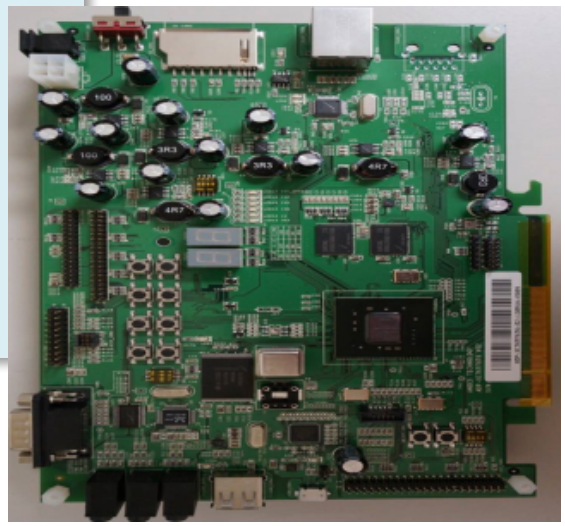
AndesCore NX25

- **AndeStar V5m 64-bit architecture**
- **5-stage pipeline**
- **Dynamic branch prediction**
- **Local Memory (LM) and caches**
  - With parity and ECC error protection
- **AXI64/AHB64 with >32 address bits**
- **JTAG debug module**
- **CoDense™, StackSafe™ and PowerBrake**
- **PLIC (Platform Interrupt Controller):**
  - Up to 1023 sources, 255 priority levels, and 16 targets
  - Efficient interrupt nesting with priority-based preemption
  - SW interrupt generation
- **28nm (RVT)**:>1 GHz (worst case), 67 K gates, 17 uW/MHz

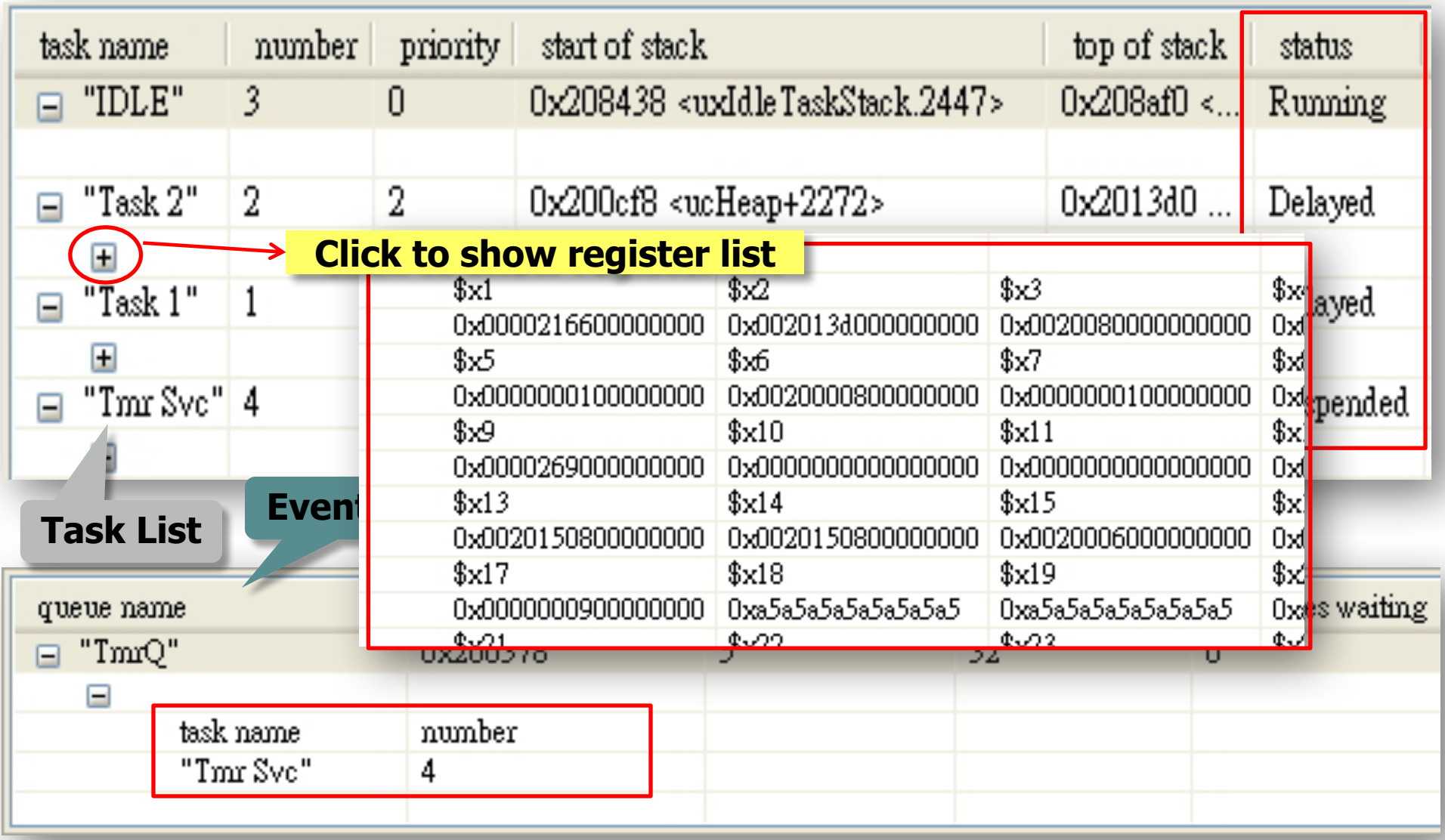
Note: Information are subject to change without notice.

Pre-integrated Platform Based on NX25





RTOS Awareness Debugging: FreeRTOS



Concluding Remarks

Concluding Remarks

- **Open source**
  - A platform for technology advancement and coopetition
- **Open source spirit** (reflecting J.F. Kennedy)
  - Ask not what the community can do for you
  - Ask what you can do for the community
- **Andes has been actively contributing to the RISC-V toolchain:**
  - Ported/integrated GCC testing framework and testsuites
  - Fixed 100+ failures due to GCC testsuite regressions
  - Contributed 20+ bug fixes to official RISC-V toolchain
    - 2nd major contributor
    - Serving as co-maintainer

Concluding Remarks

- **RISC-V has a great start**
  - Open, compact, modular, extensible, 64-bit
- **Advance Beyond Free** (ISA) **into Risk-Free** (SoC)
- **The current environment for RISC-V is**
  - Good for CPU experts
  - Not for majority of SoC design teams, who demand
    - Faster time-to-market
    - High stability, quality and comprehensive support
    - Lower total cost of product development/maintenance
- **Need experienced IP vendors to bring it to the mass**
- **Andes aims to address it with AndeStar V5**
  - Started with an efficient 64-bit implementation in NX25
  - Bring V3 features and new features to V5 processors

Thank You !!

Taking RISC-V to Mainstream ASICs

– *With AndeStar™ V5*

www.andestech.com

knect.me

Abstract

- Andes is the Taiwan-based CPU IP company with about 2-billion Andes-Embedded SoCs shipped for diversified applications from wireless connectivity, touch controllers, storage, video codec, IoT, to deep learning and datacenter routers. As a Founding Member, Andes would like to help bringing RISC-V to those markets with the infrastructure we developed in the past 12 years.

ACE: A Half-Page Example

madd32.ace

```
insn madd32 {
   operand= {io gpr acc,
             in gpr dat, in gpr coef};
   csim= %{
      acc+=(dat & 0xffff)* (coef & 0xffff)
         + (dat >>16)     * (coef >>16);
   %};
   latency= "1";
};
```

madd32.v

```
//ACE_BEGIN: madd32
 assign acc_out = acc_in
        + dat[15:0] *coef[15:0]
        + dat[31:16]*coef[31:16];
//ACE_END
```

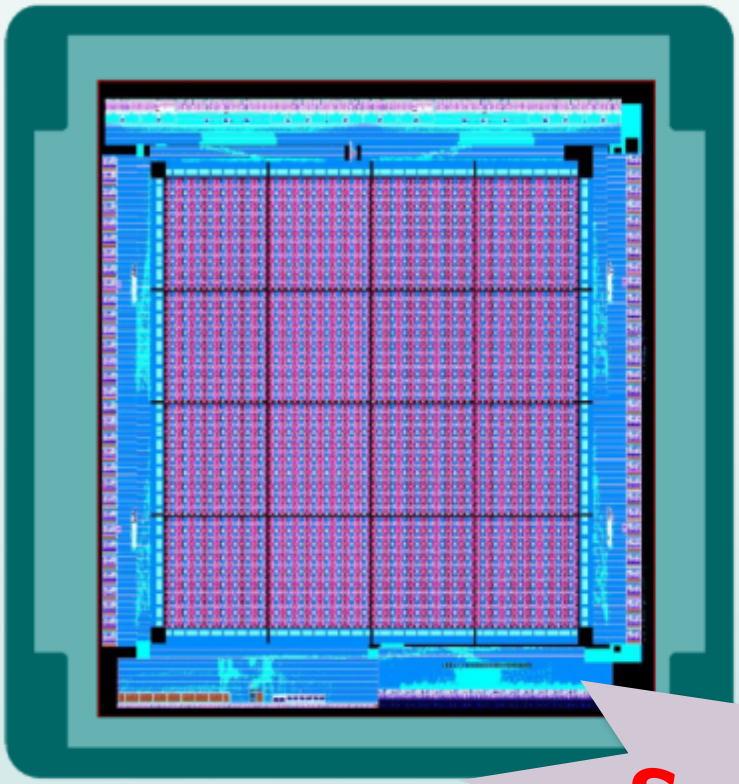
- **File madd32.ace: ACE definition script**
  - **insn**: instruction name, "madd32"
  - **op(erand)**: operand names and attributes (in/out/io gpr, imm, etc.)
  - **csim**: instruction semantics in C for instruction set simulator
  - **latency**: estimated cycles spent on instruction execution; default is 1.
- **File madd32.v: concise Verilog RTL**
  - **//ACE_BEGIN . . . //ACE_END**: instruction-specific Verilog logic

AndesCore is Learning

Wave Computing

Dataflow Processing Unit (DPU) Architecture

Scalable Machine Learning Computers for Data Center



16ff CMOS Process Node	16K Processors, 8192 DPU Arithmetic Units	Self-timed, MPP Synchronization
181 Peak Tera-Ops	16 MB Distributed Data Memory	8 MB Distributed Instruction Memory
1.71 TB/s I/O Bandwidth	270 GB/s Peak Memory Bandwidth	2048 Outstanding Memory Requests
4 Billion 16-Byte Random Access Transfers/s	4 Hybrid Memory Cube Interfaces	2 DDR4 Interfaces
PCIe Gen3 16-lane Host interface	32-b Andes N9 MCU	1 MB program store for paging
Hardware engine for fa[illegible] of AES encrypted progr	32 Dynamic Reconfiguration Zones	Variable fabric dimensions (user programmable at boot)

AndesCore as System Ctl Processor

RTOS Awareness Debugging: FreeRTOS

task name	number	priority	start of stack	top of stack	status
"IDLE"	3	0	0x208438 <uxIdleTaskStack.2447>	0x208af0 <...	Running
"Task 2"	2	2	0x200cf8 <ucHeap+2272>	0x2013d0 ...	Delayed
"Task 1"	1	1	0x200430 <ucHeap+24>	0x200b08 ...	Delayed
"Tmr Svc"	4	6	0x208c38 <uxTimerTaskStack.2454>	0x209ac0 <...	Suspended

Click to show register list

Task List

Event List

queue name	handler address	max length	item size	messages waiting
"TmrQ"	0x200378	5	32	0

task name	number
"Tmr Svc"	4

RTOS Awareness Debugging: FreeRTOS