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)

Processor IP's
AndesCore™

Development Tools
AndeSight™

Development Platforms
AndeShape™

SW Stacks
AndeSoft™

Andes Embedded™

Driving Innovations™
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

- eMMC controller
  - N801

- Sensor Hub
  - N801
  - N968
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

- Smart Plug
- Smart Lighting
- Air Purifier
- IP Cam
Andes Embedded with N7, N8, N9, N13

Mastech MS6531 IR Thermometer: ADC MCU

Nissan X-Trail: ADAS Ctrlr

Nyquest Speech Synthesizer: MCU

Toshiba SD Card: Flash Ctrlr
### AndesCore is Learning

#### Scalable Machine Learning Computers for Data Center

<table>
<thead>
<tr>
<th>Feature</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>16ff CMOS Process Node</td>
<td>16K Processors, 8192 DPU Arithmetic Units</td>
</tr>
<tr>
<td>181 Peak Tera-Ops</td>
<td>16 MB Distributed Data Memory</td>
</tr>
<tr>
<td>1.71 TB/s I/O Bandwidth</td>
<td>270 GB/s Peak Memory Bandwidth</td>
</tr>
<tr>
<td>4 Billion 16-Byte Random Access Transfers/s</td>
<td>4 Hybrid Memory Cube Interfaces</td>
</tr>
<tr>
<td>PCIe Gen3 16-lane Host interface</td>
<td>32-b Andes N9 MCU</td>
</tr>
<tr>
<td>Hardware engine for fast I/O of AES encrypted programs</td>
<td>32 Dynamic Reconfiguration Zones</td>
</tr>
</tbody>
</table>

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

AndesCore is Learning about Scalable Machine Learning Computers for Data Center.
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)

COPILOT
Custom-OPtimized Instruction developent Tools

- Extended Tools
- Extended ISS
- Extended RTL

Operands, C semantics, test-case spec
ACE script

Automated Env. For Cross Checking

Test Case Generator

- Extended RTL
- Extended ISS

Compilation

- Compiler
- Asm/Disasm
- Debugger
- IDE

CPU ISS (near-cycle accurate)

CPU RTL

Extensible Baseline Components

Executable or library

Source file

Verilog user.v
concise RTL
# ACE Features

<table>
<thead>
<tr>
<th>Items</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Instruction</strong></td>
<td></td>
</tr>
<tr>
<td>Scalar</td>
<td>- Single-cycle</td>
</tr>
<tr>
<td></td>
<td>- Multi-cycle (interruptible or non-interruptible)</td>
</tr>
<tr>
<td>Vector</td>
<td>- ‘for’ loop</td>
</tr>
<tr>
<td></td>
<td>- ‘do while’ loop</td>
</tr>
<tr>
<td>Background</td>
<td>- Issued and retired immediately from CPU pipeline, but continue the remaining execution in background</td>
</tr>
<tr>
<td><strong>Operand:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Explicit or Implied</strong></td>
<td></td>
</tr>
<tr>
<td>Standard</td>
<td>- Immediate constant</td>
</tr>
<tr>
<td></td>
<td>- GPR (up to 3R2W)</td>
</tr>
<tr>
<td></td>
<td>- Baseline memory (accessed thru CPU)</td>
</tr>
<tr>
<td>Custom</td>
<td>- ACR (ACE Register)</td>
</tr>
<tr>
<td></td>
<td>- ACM (ACE Memory)</td>
</tr>
<tr>
<td></td>
<td>- ACP (ACE Port)</td>
</tr>
<tr>
<td><strong>Auto-Generation</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Opcode selection (optional)</td>
</tr>
<tr>
<td></td>
<td>- All required tools/simulator with fast turnaround time</td>
</tr>
</tbody>
</table>
|                            | - 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

V1 → V2 → V3

CoDense™
StackSafe™
PowerBrake

Custom Ext.
DSP/FP Ext.
Security Ext.

COPilot tool
Compiler Opt.
>200 DSP Libraries
Secure RTOS

Full Feature

Baseline

RISC Kernel
AndesStar™ ISA: Next Generation

- V1
- V2
- V3
- V3m
- Next Gen.

- CoDense™
- StackSafe™
- PowerBrake

- Custom Ext.
- DSP/FP Ext.
- Security Ext.

- COPilot tool
- Compiler Opt.
- >200 DSP Libraries
- Secure RTOS

**Full Feature**

- **Baseline**
  - RISC Kernel
AndeStar™ ISA: V5

- V1
- V2
- V3
  - V3m
- V5
  - V5m

Full Feature

Baseline

RISC-V Kernel

Custom Ext. DSP/FP Ext. Security Ext.

CoDense™ StackSafe™ PowerBrake

COPILLOT tool Compiler Opt. >200 DSP Libraries Secure RTOS

V5m+ more Andes Ext.
RV64IMAC + Andes Ext.
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
  - Andes Sixty-four and Thirty-two Architecture

- 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
  - **Andes** Sixty-four and **Thirty-two Architecture**

- **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, compiler, and debug toolchain
  - 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

Platform Interrupt Controller

LMBRG

Flash/ROM/RAM

NX25

Inst

Data

BIU

AXI/AHB Bus Matrix

DMA

Bus Masters

Bus Slaves

APB Bridge

WDT

Sys. Mgmt Unit

GPIO

I2C

PWM/PIT

RTC

SPI

UART

AXI/AHB IP

APB IP

CPU Subsystem

Customers’ or Partners’ IP’s

Driving Innovations™
RTOS Awareness Debugging: FreeRTOS

Click to show register list

Task List

Event List

queue name

"TmrQ"

"Tmr Svc"

32
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
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

File madd32.ace: ACE definition script
- **insn**: instruction name, “madd32”
- **op(operand)**: 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: madd32
assign acc_out = acc_in
    + dat[15:0] * coef[15:0]
    + dat[31:16] * coef[31:16];
//ACE_END```
AndesCore is Learning

Scalable **Machine Learning Computers** for Data Center

<table>
<thead>
<tr>
<th>Feature</th>
<th>Specification</th>
<th>Feature</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>16ff CMOS Process Node</td>
<td>16K Processors, 8192 DPU Arithmetic Units</td>
<td>Self-timed, MPP Synchronization</td>
<td></td>
</tr>
<tr>
<td>181 Peak Tera-Ops</td>
<td>16 MB Distributed Data Memory</td>
<td>8 MB Distributed Instruction Memory</td>
<td></td>
</tr>
<tr>
<td>1.71 TB/s I/O Bandwidth</td>
<td>270 GB/s Peak Memory Bandwidth</td>
<td>2048 Outstanding Memory Requests</td>
<td></td>
</tr>
<tr>
<td>4 Billion 16-Byte Random Access Transfers/s</td>
<td>4 Hybrid Memory Cube Interfaces</td>
<td>2 DDR4 Interfaces</td>
<td></td>
</tr>
<tr>
<td>PCIe Gen3 16-lane Host interface</td>
<td></td>
<td>1 MB program store for paging</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Variable fabric dimensions</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(user programmable at boot)</td>
<td></td>
</tr>
</tbody>
</table>

**AndesCore as System Ctrl Processor**
### RTOS Awareness Debugging: FreeRTOS

#### Task List

<table>
<thead>
<tr>
<th>task name</th>
<th>number</th>
<th>priority</th>
<th>start of stack</th>
<th>top of stack</th>
<th>status</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;IDLE&quot;</td>
<td>3</td>
<td>0</td>
<td>0x208438 &lt;uxIdleTaskStack.2447&gt;</td>
<td>0x208af0 ...</td>
<td>Running</td>
</tr>
<tr>
<td>&quot;Task 2&quot;</td>
<td>2</td>
<td>2</td>
<td>0x200cf8 &lt;ucHeap+2272&gt;</td>
<td>0x2013d0 ...</td>
<td>Delayed</td>
</tr>
<tr>
<td>&quot;Task 1&quot;</td>
<td>1</td>
<td>1</td>
<td>0x200430 &lt;ucHeap+24&gt;</td>
<td>0x200b08 ...</td>
<td>Delayed</td>
</tr>
<tr>
<td>&quot;Tmr Svc&quot;</td>
<td>4</td>
<td>6</td>
<td>0x208c38 &lt;uxTimerTaskStack.2454&gt;</td>
<td>0x209ac0 ...</td>
<td>Suspended</td>
</tr>
</tbody>
</table>

#### Event List

<table>
<thead>
<tr>
<th>queue name</th>
<th>handler address</th>
<th>max length</th>
<th>item size</th>
<th>messages waiting</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;TmrQ&quot;</td>
<td>0x200378</td>
<td>5</td>
<td>32</td>
<td>0</td>
</tr>
</tbody>
</table>

---

**Click to show register list**
### RTOS Awareness Debugging: FreeRTOS

<table>
<thead>
<tr>
<th>task name</th>
<th>number</th>
<th>priority</th>
<th>start of stack</th>
<th>top of stack</th>
<th>status</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;IDLE&quot;</td>
<td>3</td>
<td>0</td>
<td>0x208438 &lt;uxIdleTaskStack.2447&gt;</td>
<td>0x208af0 &lt;...</td>
<td>Running</td>
</tr>
<tr>
<td>&quot;Task 2&quot;</td>
<td>2</td>
<td>2</td>
<td>0x200cf8 &lt;ucHeap+2272&gt;</td>
<td>0x2013d0 ...</td>
<td>Delayed</td>
</tr>
<tr>
<td>&quot;Task 1&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;Tmr Svc&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Task List**

**Queue Name**

- "TmrQ"

**Click to show register list**