Comprehensive RISC-V Solutions for AIoT

Charlie Su, Ph.D.
CTO and SVP, Andes
2018/06/30

Agenda

- Introduction to Andes
- AndeStar™ V5 Architecture and AndesCore™
- AIoT Development Support for V5 Processors
- Concluding Remarks

Introduction to Andes

Introduction to Andes

- **Asia-based IPO Company**
- **13 years in the pure-play CPU IP business**
- **Before Andes adopted RISC-V technologies:**
  - AndeStar™ V1-V3 architecture
  - 10 active V3 AndesCores: 2-8 stage pipeline, single- and dual-issue
  - Upstreamed V3 (NDS32) GNU tools, OpenOCD, U-Boot, Linux, etc.
  - >140 licensees, >2.5B Andes-Embedded SoCs

Introduction to Andes

- **Rich experience in customer diversified needs:**
  - Interrupt sources: who needs >16? 2-stage core with >100.
  - Interrupt latencies: Some never ask; others care very much.
  - Efficiency: DSP+SIMD based on existing integer resources (i.e. GPR)
  - Performance: scalable vector (or SIMD) with dedicated resources
  - Need caches to be write-back and write-through
  - Loading RO-data from icache !
  - Some think GNU/LLVM the most popular; others think otherwise

Andes Activities in RISC-V Community

- **A founding member of RISC-V Foundation**
- **A major contributor to RISC-V tools**, often as maintainers
  - Compilation tools: GCC (and binutils, newlib), LLVM (and LLD)
  - Debugging tools: GDB, OpenOCD
  - U-Boot, Linux, Linux performance tools (Ftrace, Module, Perf)
- **Contributing architecture extensions too**
  - Chair of the forming P-extension (Packed SIMD/DSP) Task Group
  - Co-chair of Fast Interrupt Task Group
  - Closely watching/reviewing activities of other Task Groups
- **General promotion:**
  - Program Committee of Barcelona Workshop and Shanghai Day
  - APAC Promotion Task Group
  - Organizing a Taiwan Workshop in 2019/03 (with NTHU/ITRI)
- **Pushing RISC-V ecosystem forward with partners**
- **Taking RISC-V mainstream:**
  - To be a major application platform like x86 and ARM

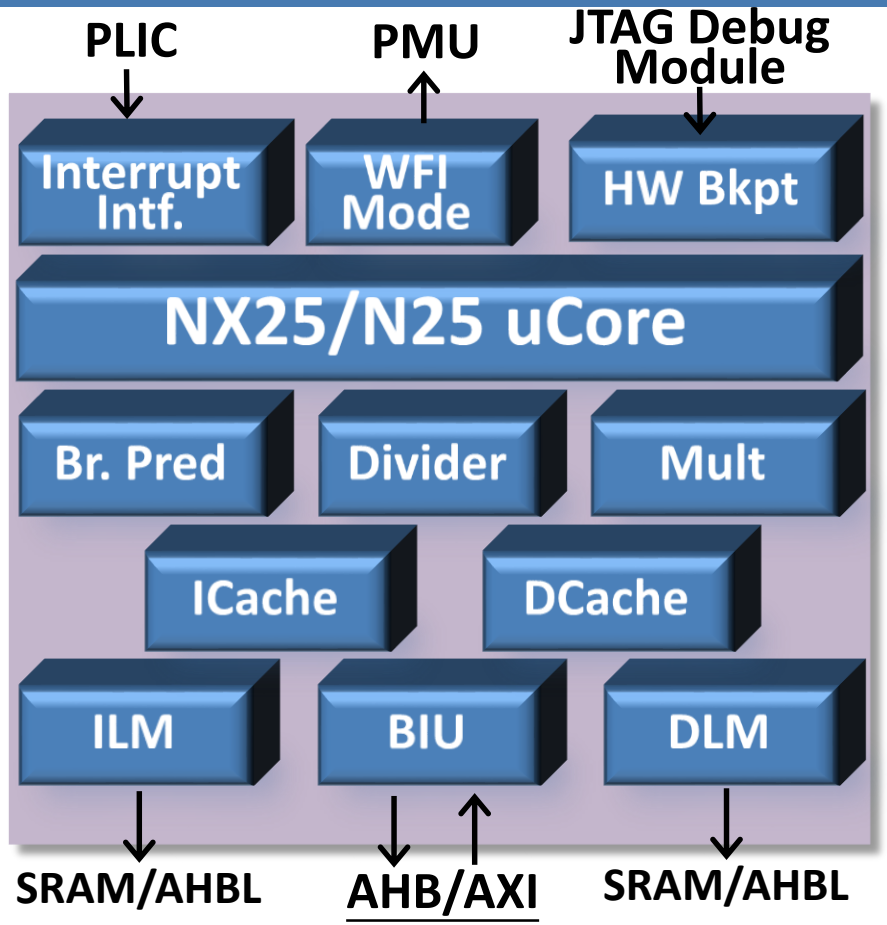
AndeStar V5 Architecture and V5 AndesCore Processors

Andes Approach to RISC-V

- **RISC-V:**
  - Concise (RV-I), Modular (RV-MACFD and more): good start
  - Extensible: understanding that one size doesn't fit all
  - Profiles: no need to be compatible from MCU to servers
- **AndeStar V5 ISA architecture:**
  - Adopt RV-IMAC as its baseline
  - Add basic performance extension instructions:
    - Path length reduction and further code size reduction (CoDense™)
  - Add DSP/SIMD ext. based on GPR → P-extension proposal
  - Provide custom-extension frameworks for DSA (or ASIP)
    - Powerful tool for SoC designers without CPU background
- **AndeStar V5 CSR extensions:**
  - Vectored PLIC with priority preemption → Fast interrupts proposal
  - Stack protection mechanism (StackSafe™)
  - Power throttling (PowerBrake)
  - Cache management in finer granularity, write-back and write-thru

V5 AndesCores: 25-series Baseline

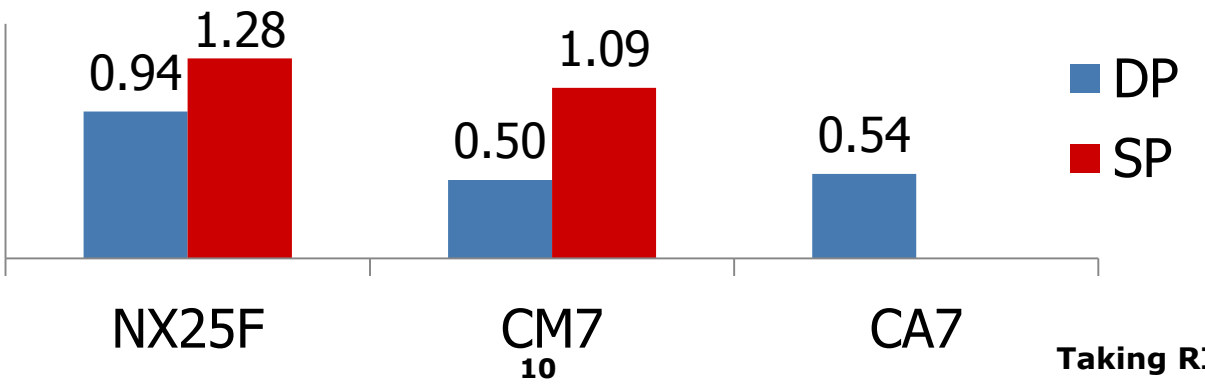
- **N25: 32-bit, NX25: 64-bit**
  - From scratch for the best PPA
  - Very configurable
- **AndeStar V5 ISA**
- **5-stage pipeline**
- **Configurable multiplier**
- **Optional branch prediction**
- **Flexible memory subsystem**
  - I/D Local Memory (LM): to 16MB
  - I/D caches: up to 64KB, 4-way
  - Optional parity or ECC
  - Hit-under-miss caches
  - load/store: unaligned accesses



- **N25 sample configurations @TSMC 28HPC RVT:**
  - Small config: 37K gates, 1.0 GHz (worst case)
  - Large config: 159K gates, 1.15GHz (worst case)
  - **Best-in-class Coremark: 3.49/MHz**

V5 AndesCores: New 25-series

- **N25/NX25:**
  - Fast-n-small for control tasks
- **N25F/NX25F: +FPU**
  - +, −, x, x+, x−: pipelined 4 cycles
  - ÷, √ : run in the background
    - 15 for SP, 29 for DP
  - FP load/store: support HP
- **A25/AX25: +FP +Linux**
  - Support RISC-V MMU and S-mode
  - 4 or 8-entry ITLB and DTLB
  - 4-way 32~128-entry Shared-TLB
- **Whetstone/MHz**:

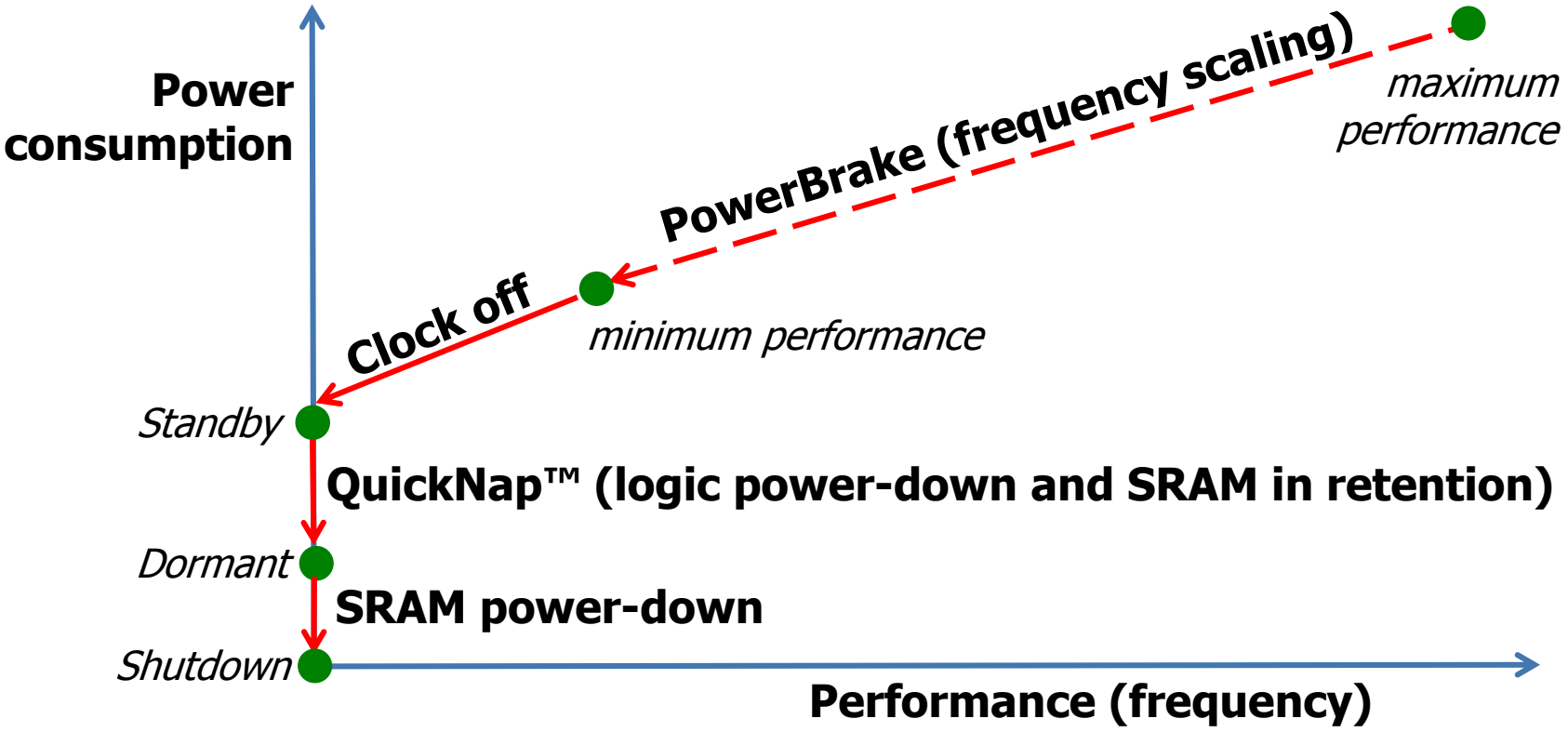


AIoT Support for V5 Processors

- Power management
- DSP SIMD for slow media processing
- ACE for DSA acceleration
- Development environment
- Protocol stack

Power Management

- **PowerBrake** to digitally adjust power (via stalling pipeline)
- **QuickNap™**: logic power-down and SRAM in retention mode
  - Put dirty bits in tag SRAM instead of flops
  - Eliminate the need to flush data cache



Andes DSP/SIMD ISA For RISC-V

- Background:
  - >150 DSP ISA in the popular V3 processors D10 and D15
  - Donated it as the basis for RV-P extension proposal (for RV32 & RV64)
  - Ported CMSIS-NN and used CIFAR-10 for image classification:
    - Single-issue D10 is ~8% faster than dual-issue CM7

Voice Codec	AMR-WB		G.729		Helix MP3
	Encode	Decode	Encode	Decode	Decode
Speedup (over no DSP ISA)	**> 4x**	**> 4x**	**> 5x**	**> 5x**	**> 2x**

- Feature highlights:
  - Efficient DSP based on existing GPRs
  - Saturation and/or rounding
  - Data types: integer (32b, 16b, 8b) and fractional (Q31, Q15, Q7)
  - 16-bit and 8-bit SIMD instructions
  - Most Sophisticated instructions: **64b += 16b x 16b + 16b x 16b**
- SW support:
  - Compiler, intrinsic functions, >200 optimized DSP library functions

ACE Features

Items	Description	
Instruction	**scalar**	single-cycle, or multi-cycle
	vector	**for** loop, or **do-while** loop
	background option	retire immediately, and continue execution in the background. Applicable to scalar and vector.
Operand	**standard**	immediate, GPR, baseline memory (thru CPU)
	custom	- ACR (ACE Register), ACM (ACE Memory) - With arbitrary width and number
	implied option	Implied operands don't appear in mnemonic
Auto Generation	- Opcode assignment: automatic by default - All required tools, and simulator (C or SystemC) - **RTL code for instruction decoding, operand mapping, dependence checking, input accesses, output updates** - Waveform control file	
Fast turnaround time !		

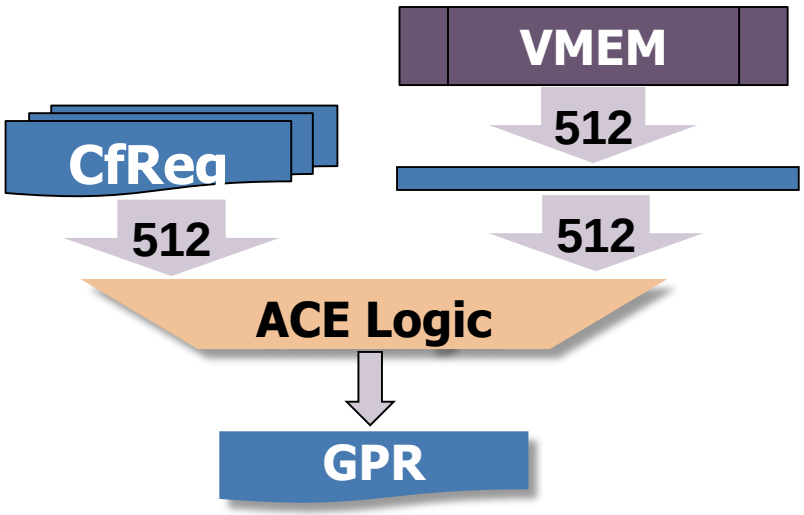
Inner Product of Vectors with 64 8-bit Data

```
reg CfReg {                 //Coef Registers
    num= 4;
    width= 512;
}
ram  VMEM {                 //data memory
     interface= sram;
     address_bits= 3;          //8 elements
     width= 512;
}
insn innerp {
  operand= {out gpr IP,
                 in CfReg C, in VMEM V};
  csim= %{       //multi-precision lib. used
    IP= 0;
    for(uint i= 0; i<64; ++i)
       IP+= ((C >>(i*8)) & 0xff) *
            ((V >>(i*8)) & 0xff);
  %};
  latency= 3;      //enable multi-cycle ctrl
};
```

innerp.ace

```
//ACE_BEGIN: innerp
assign IP= C[ 7:0]  * V[ 7:0]
         + C[15:8] * V[15:8]
           . . .
         + C[511:504] *
           V[511:504];
//ACE_END
```

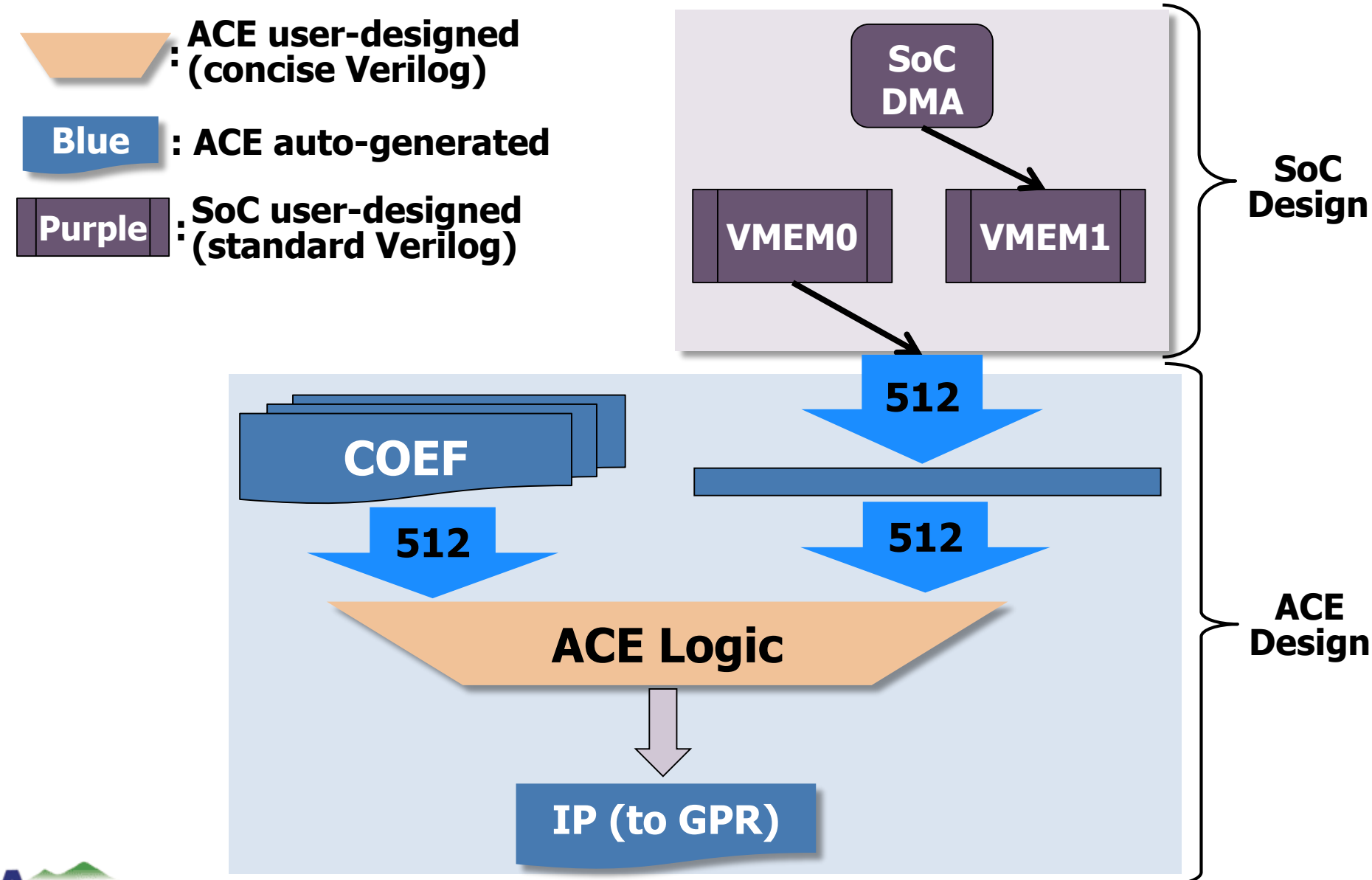
innerp.v



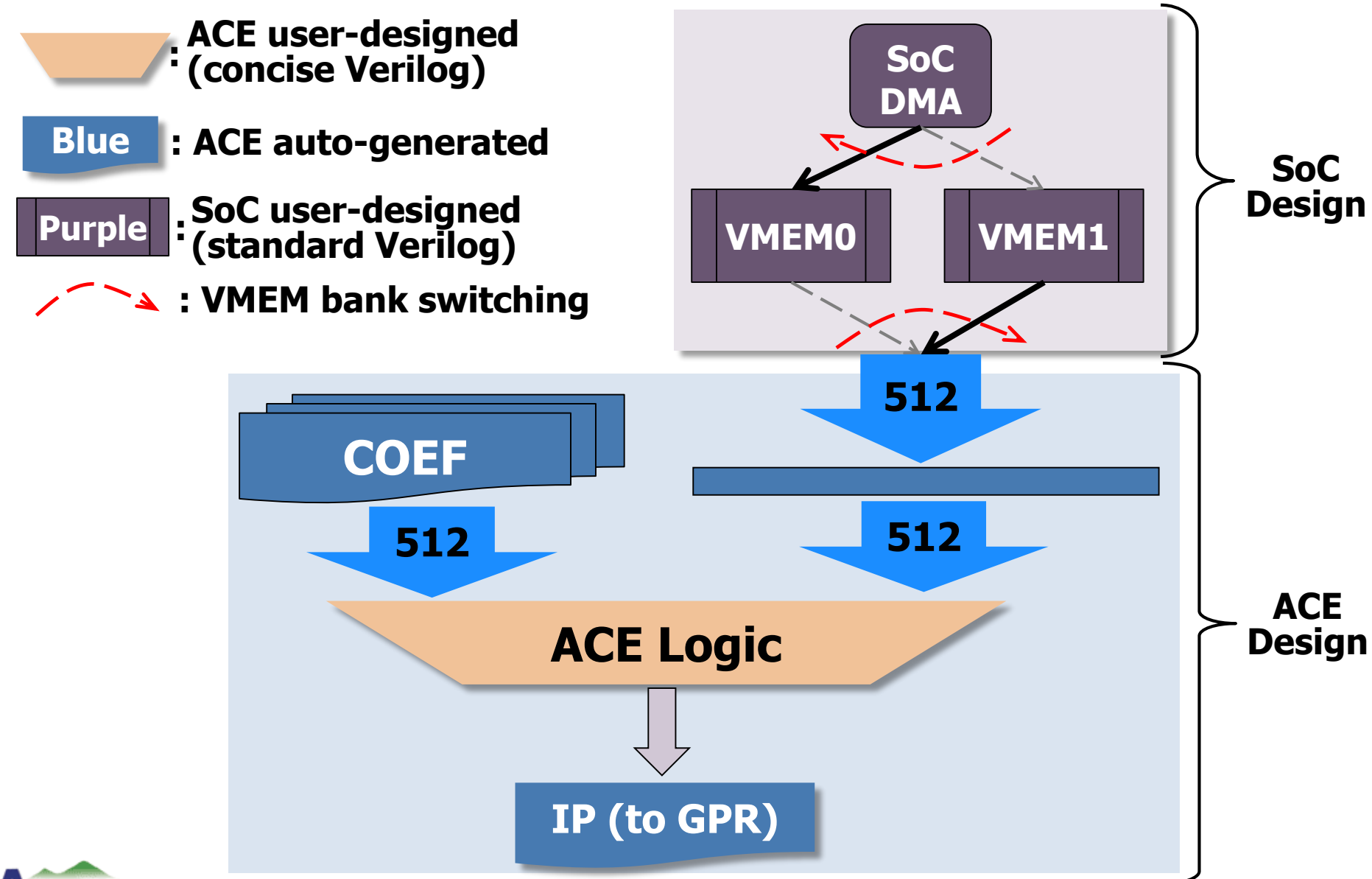
Speedup: 85x

Intrinsic: long **ace_innerp**(CfReg_t, VMEM_t);

Double Buffering for Compute and DMA

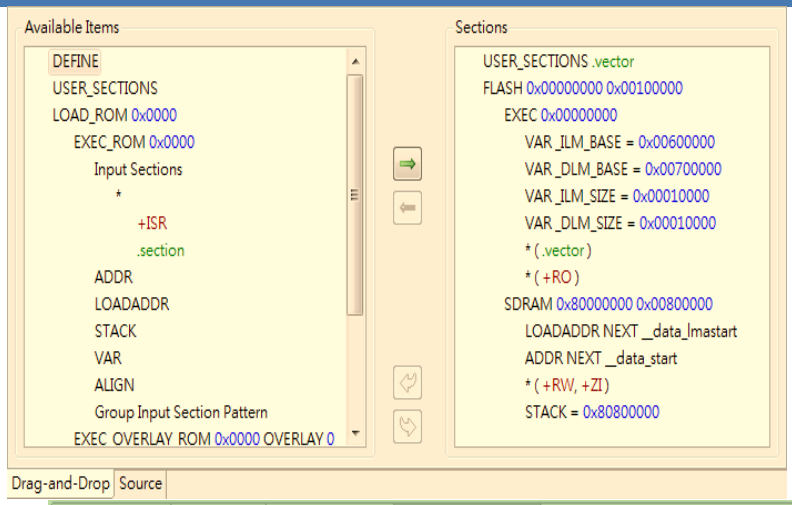


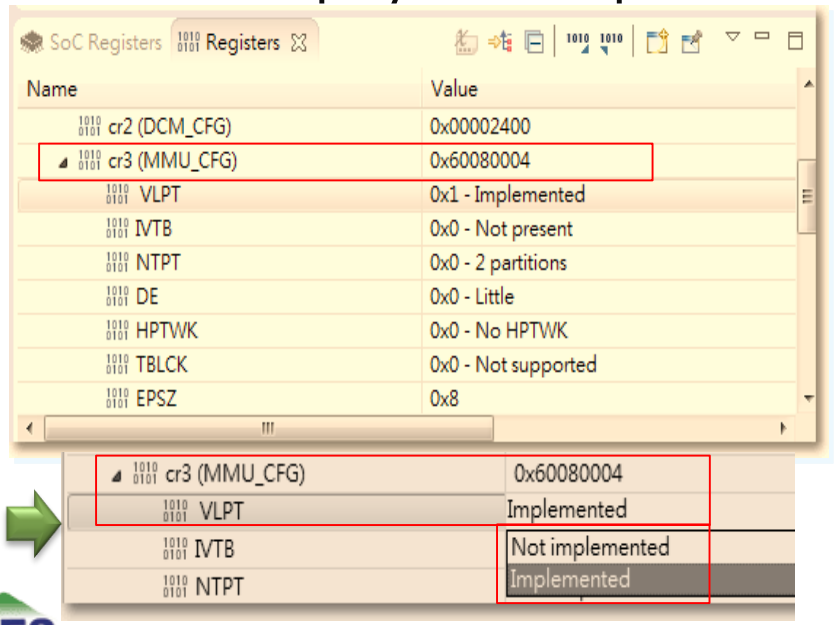
Double Buffering for Compute and DMA

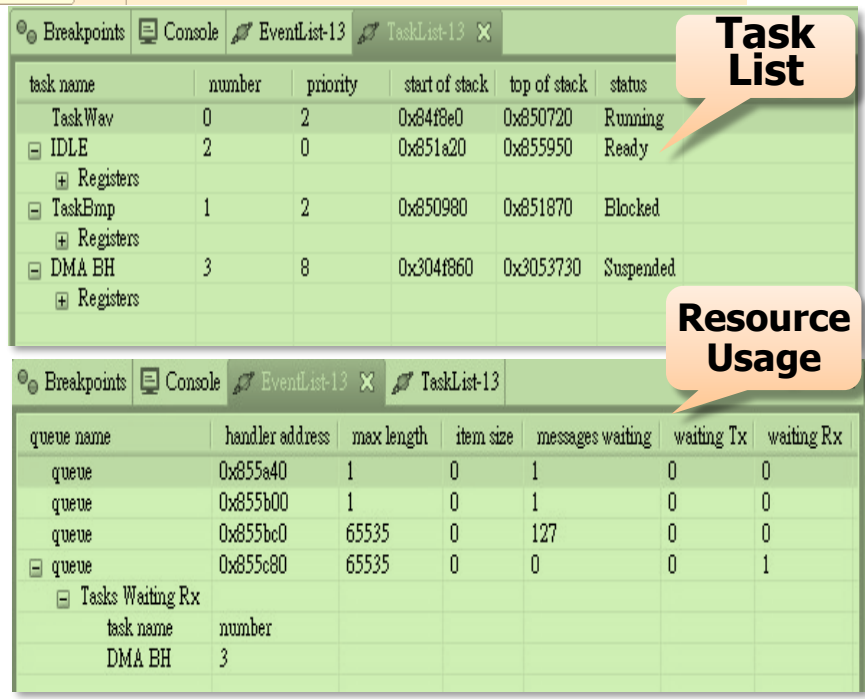


AndeSight™: Professional IDE

- **Eclipse-based**
- **Project Setup:**
  - Meta Linker Script Editor
  - Flash ISP configured through GUI
- **Debug Support:**
  - Script-Based RTOS Awareness
  - Virtual Hosting
  - Register Bitfield Viewing/Update
  - Break-n-Display on Exceptions

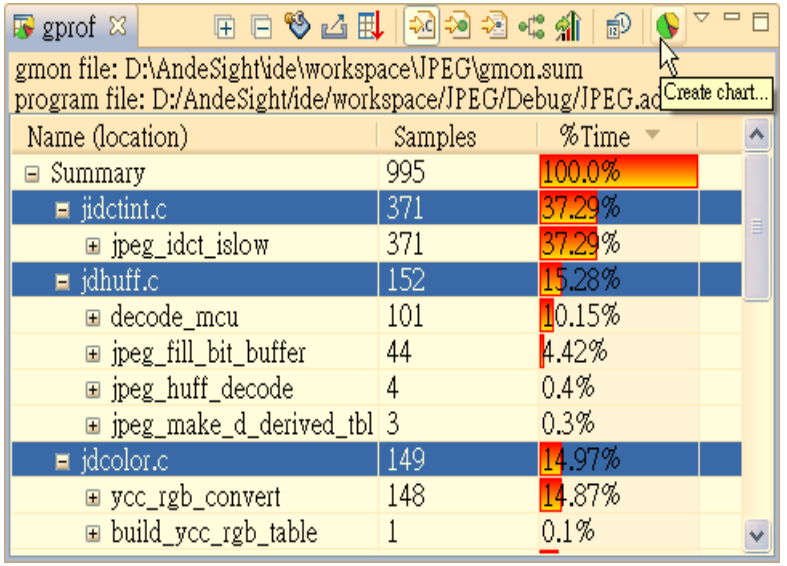


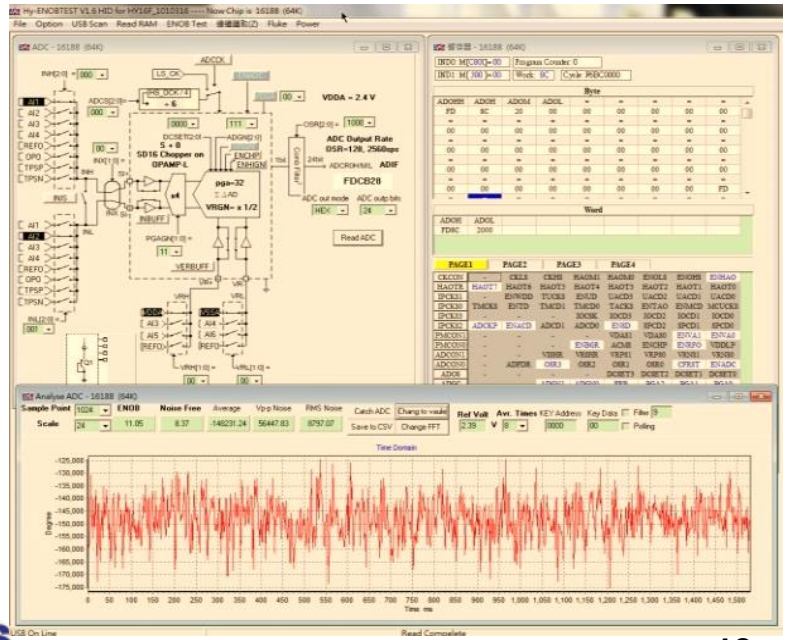


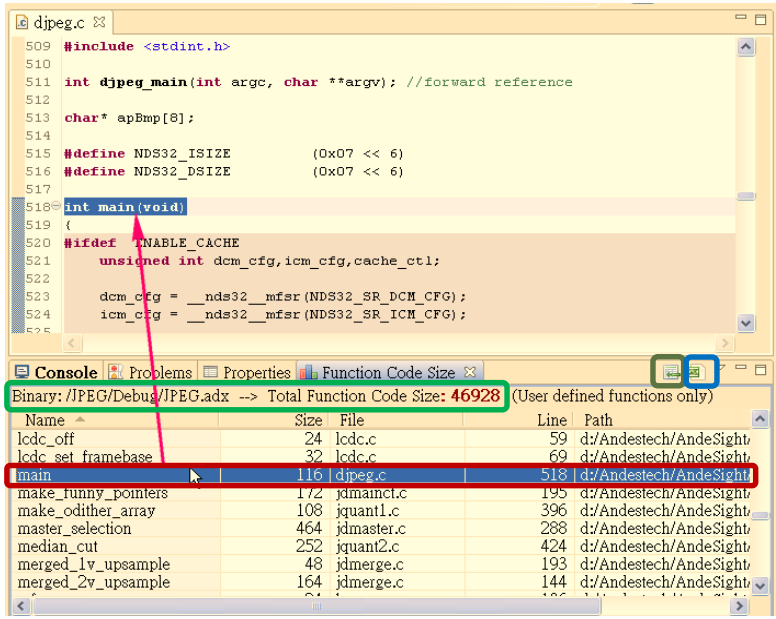


AndeSight™: Professional IDE

- **Program Analysis**
  - Function Profiling
  - Performance Meter
  - Code Coverage
  - Function Code Size
  - (Static) Stack Size
- **Debug/Analysis for Arduino**
- **Custom Plugin Intf**





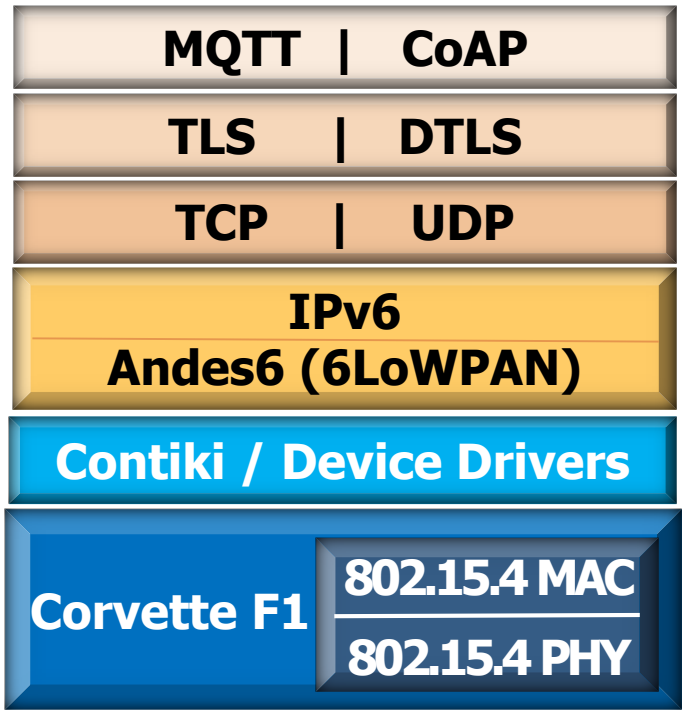


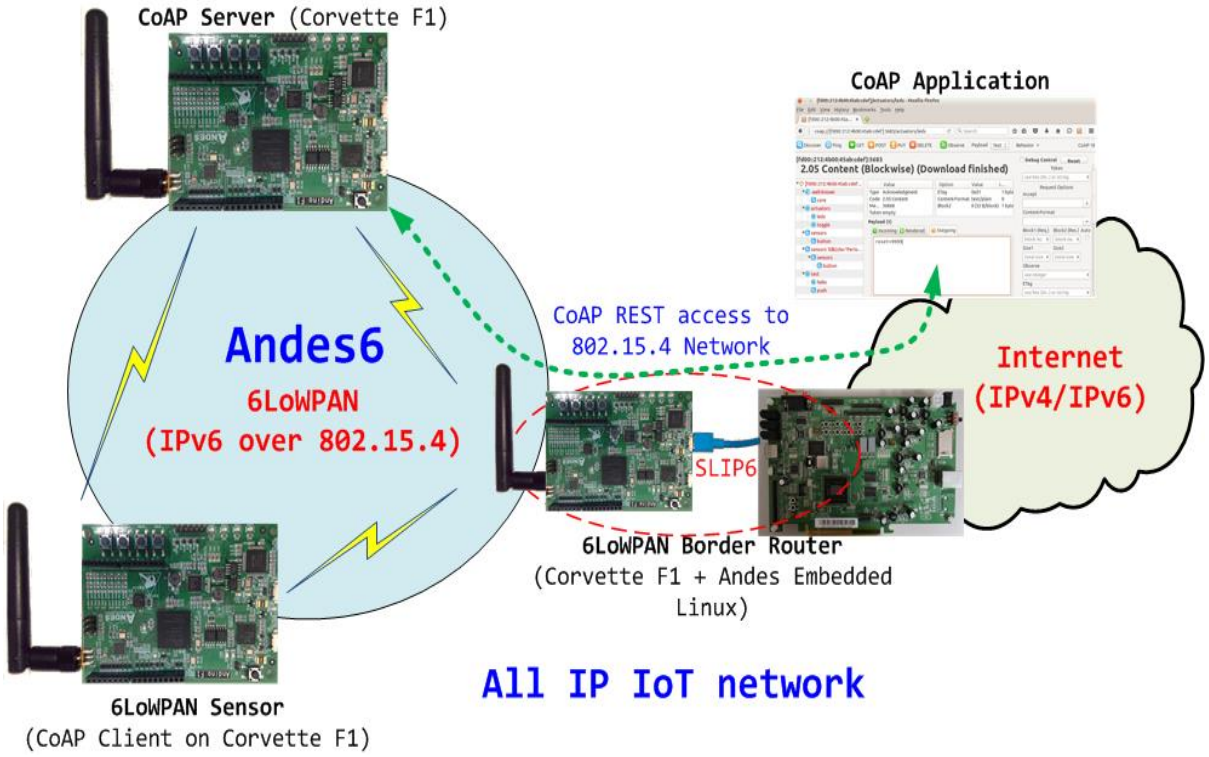
Development Environment

- **AndeShape™ Development Boards**
  - Full-Featured ADP-XC7
  - Compact Corvette-F1 (Arduino-compatible)
    - With 802.15.4 and ICE on board
- **Qemu Virtual Board**
  - AX25 with AE350 SoC platform
  - Booted U-Boot and Linux
  - Used by openSUSE project for UEFI
- **AndeSoft™ SW Stack**
  - Bare metal projects for Andes-enhanced features
  - RTOS'es: FreeRTOS, ThreadX, Contiki, more
  - IoT Stack talking to the Cloud (next pages)
- **Special Support from 3rd Parties**
  - Imperas fast simulator
  - Trace32 debugger
  - UltraSoC trace support

Andes IoT Stack

- **Andes6 connectivity components**
  - Contiki RTOS for OS services
  - An implementation of 6LoWPAN (IPv6 over 802.15.4)
  - Commercial (e.g. InsideSecure) or open source TLS for security
- **Connecting to the Clouds (Microsoft Azure, Acer BYOC)**





Concluding Remarks

Concluding Remarks

- **RISC-V is emerging as a major application platform**
- **AIoT is an emerging application**
- **Andes offers comprehensive RISC-V solutions**
  - **V5 processors:**
    - N25/NX25: Fast-n-small cores for control tasks
    - N25F/NX25F: FP cores for computation tasks such as AI and GPS
    - A25/AX25: Application Processors with high performance efficiency
  - **ACE for AI or DSA Acceleration**
    - A separate option available for all V5 cores
  - **Rich development tools and SW stacks**:
    - from Andes and partners
- **Experience and focus is very important for RISC-V**

Andes is your best RISC-V partner !

Thank You !!





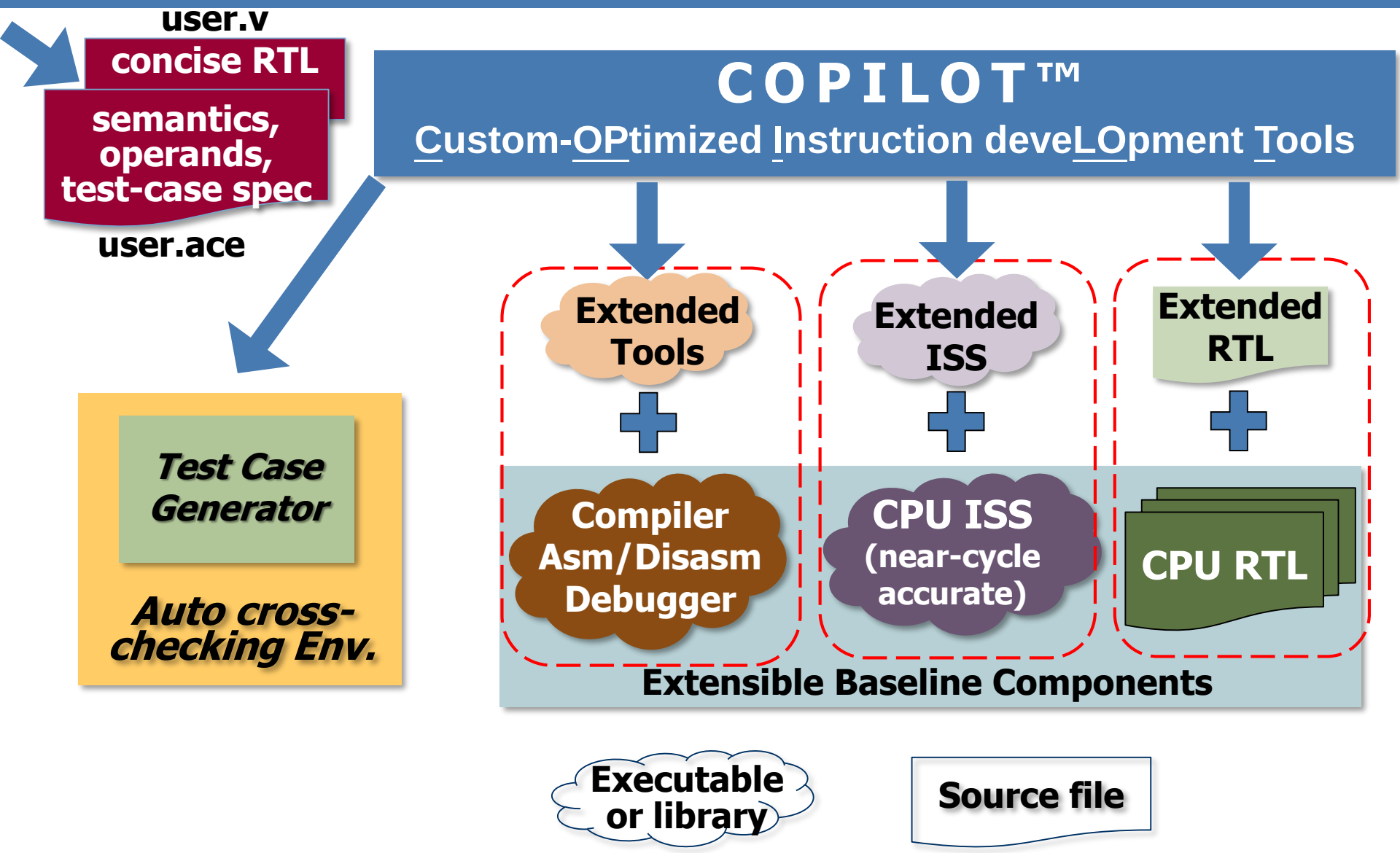




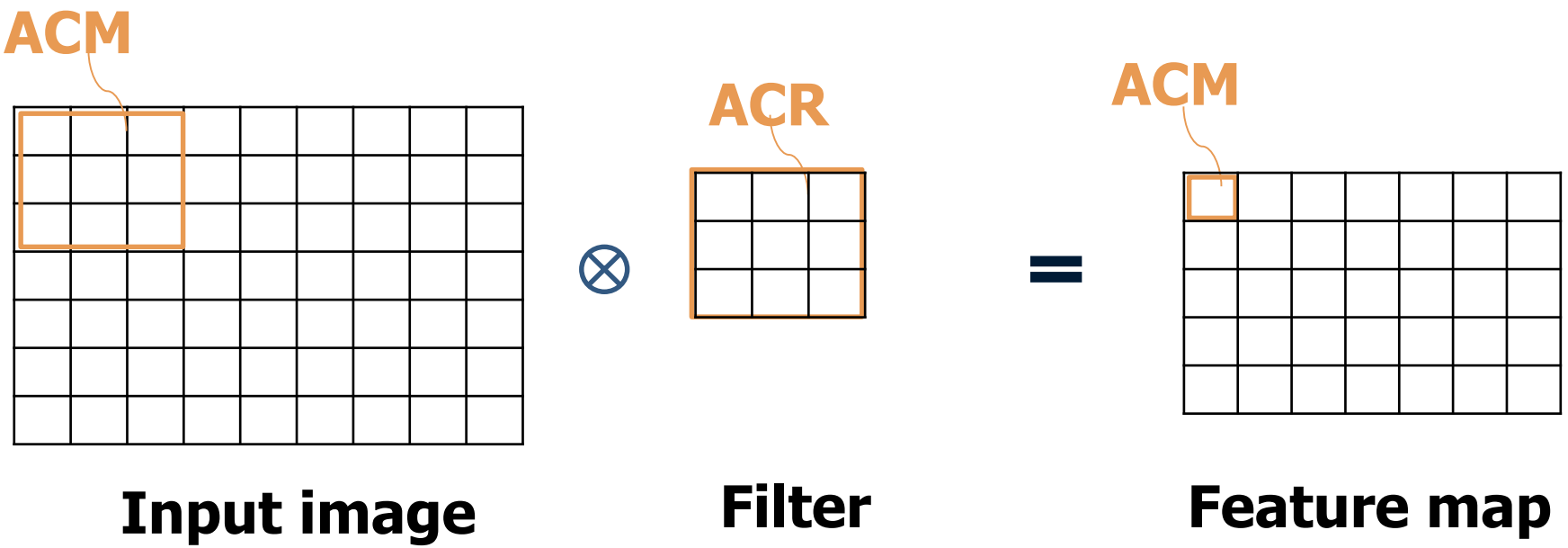



Additional Material

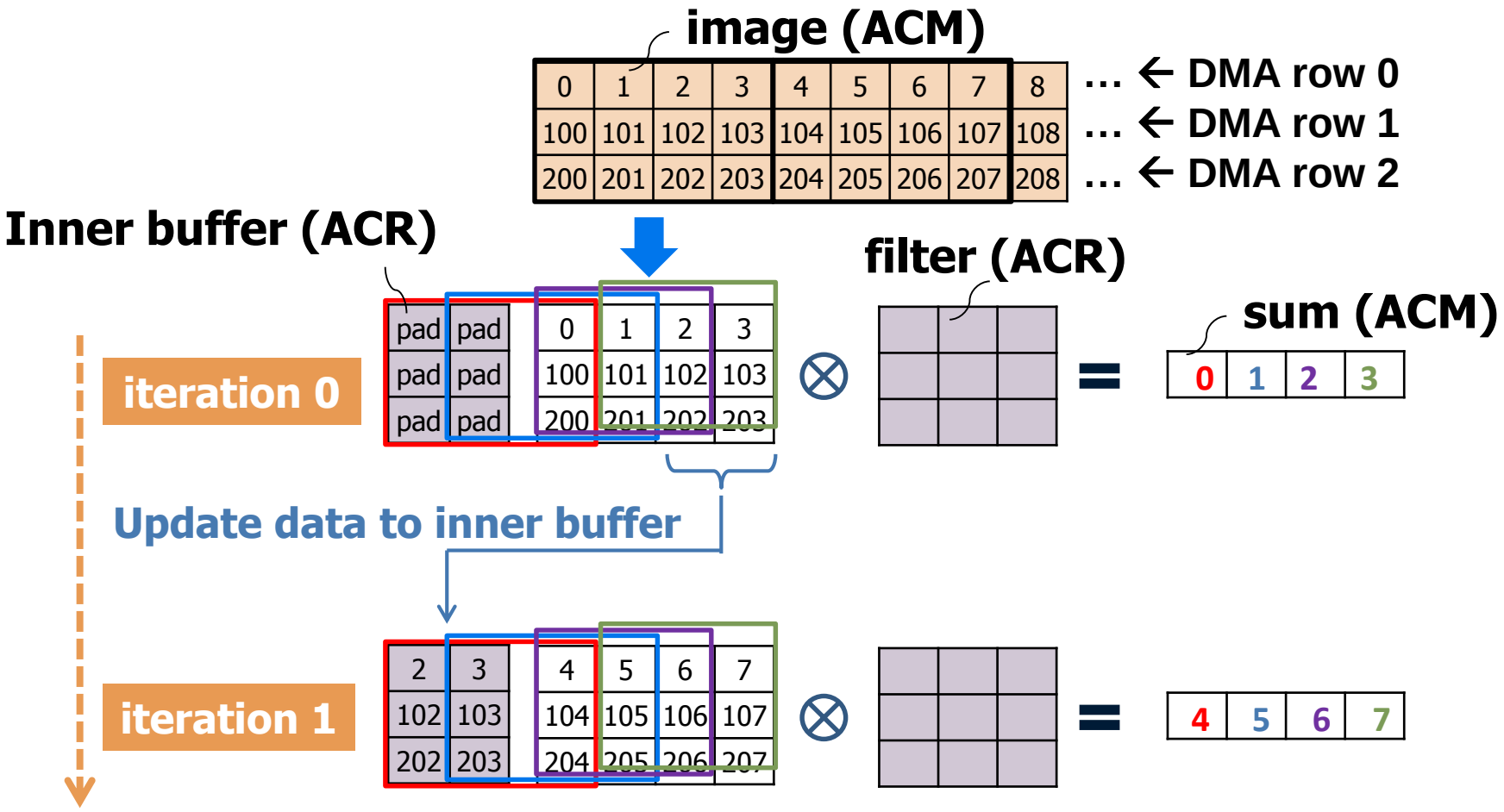
ACE Framework for DSA



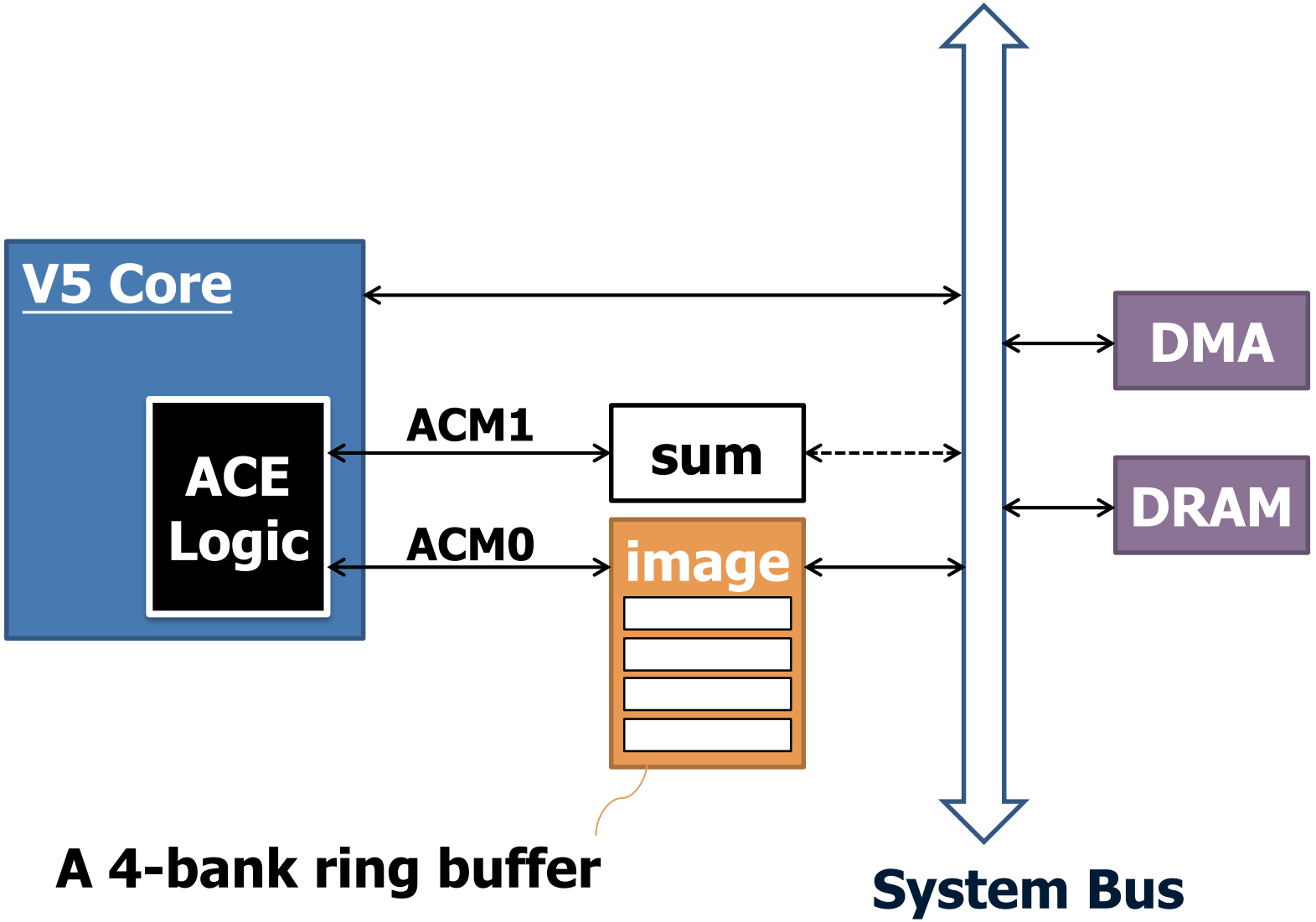
ACE Instruction: Matrix Convolution



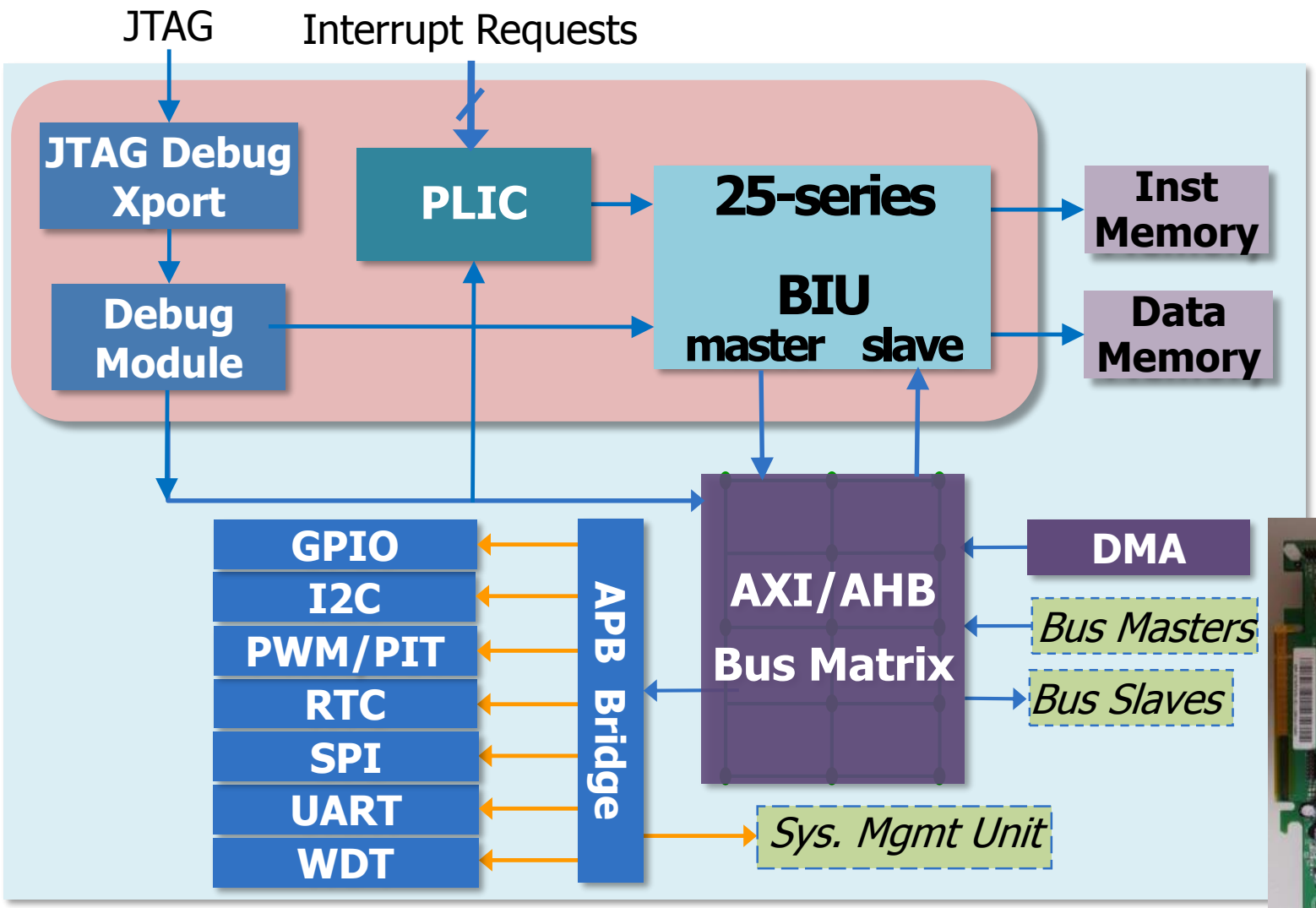
Operation Flow



System Block Diagram

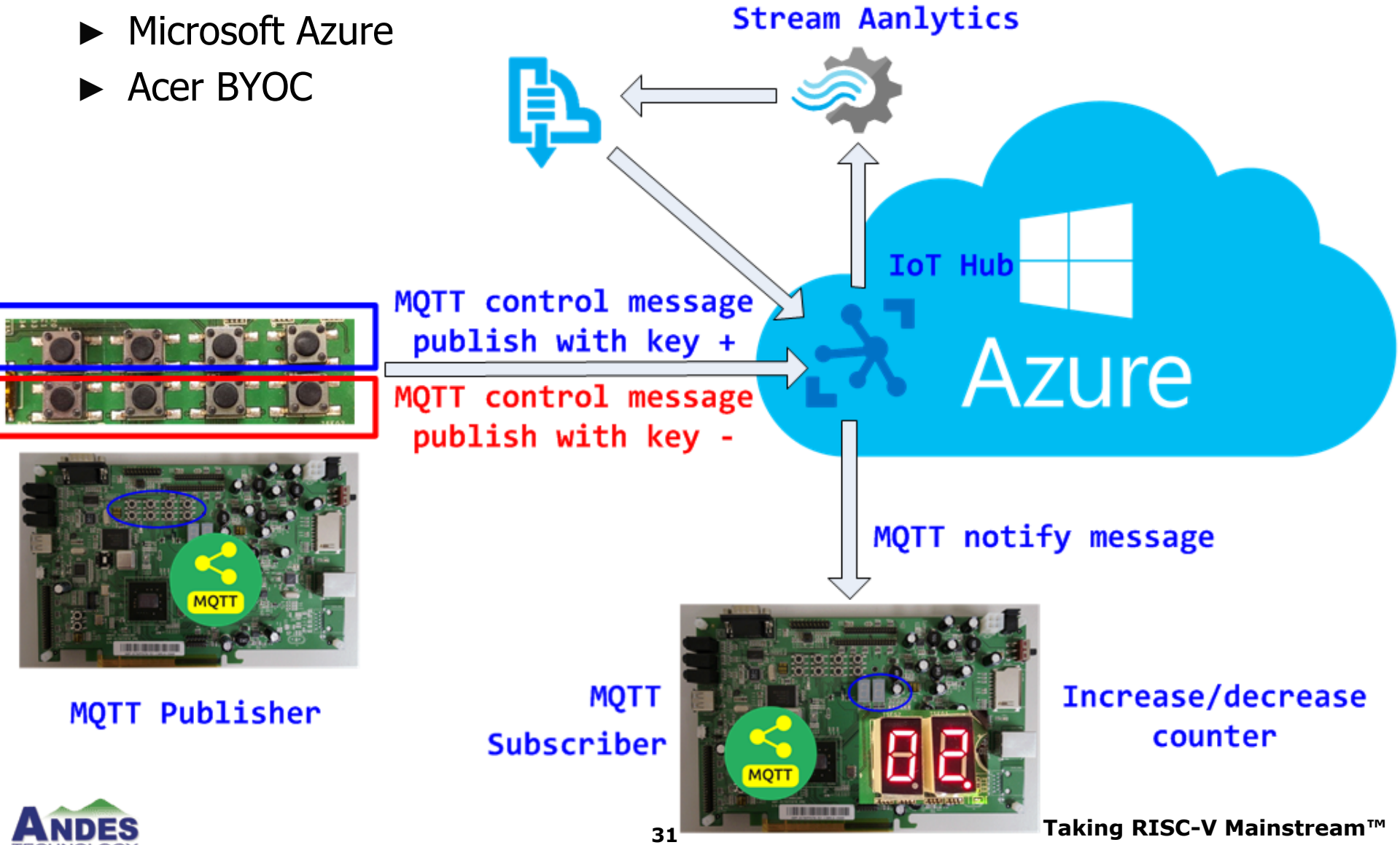


Pre-integrated Platform



Andes IoT Enablement: To the Cloud

- **Andes is cloud-ready for**
  - Microsoft Azure
  - Acer BYOC



AndeSoft™: Bare Metal, RTOS, and Linux

- **Bare metal support:**
  - Fast interrupt handling
- **Latest RTOS'es:**
  - FreeRTOS v10.0, 32-bit and 64-bit
  - ThreadX v5.7, 32-bit and the 1st RISC-V 64-bit port
  - AliOS Things, Huawei LiteOS, and RT-Thread
- **V3 Linux is upstreamed to 4.17 !!**
- **V5/64-bit Linux ready with advanced tools**
  - uBoot (mainlined, maintainer) : universal bootloader, to enable UEFI and OpenSUSE
  - Ftrace (upstreamd to 4.17) : Kernel tracing tool
  - Module (upstreamd to 4.17) : enable dynamic loading of kernel modules
  - Perf (patchset sent) : System-wide profiling

ThreadX RV32 to RV64

- Focus on ThreadX standard port assembly files
- Abstract assembly operation for RV32/RV64
  - Load/Store instructions
  - Registers width
  - Data type size
- RV32/RV64 assembly porting parts
  - Context save/restore handling
  - Offset value for TCB member accessing

RV32/RV64 assembly porting

Abstract assembly operation definition

```
#if __riscv_xlen == 64
#define STORE    sd
#define LOAD     ld
#define REGWORDS 2
#else
#define STORE    sw
#define LOAD     lw
#define REGWORDS 1
#endif
```

Offset of TCB member

```
LOAD t1, _tx_thread_current_ptr

/* SP = _tx_thread_current_ptr
-> tx_thread_stack_ptr;  */
LOAD sp, 8*REGWORDS(t1)
...
...
```

```
typedef struct TX_THREAD_STRUCT
{
  ULONG tx_thread_id;
  ULONG tx_thread_run_count;
  VOID  *tx_thread_stack_ptr;
  ...
  ...
} TX_THREAD;
```

Context Save

```
addi  sp, sp, -128*REGWORDS

STORE x1, 0x70*REGWORDS(sp)
STORE x7, 0x44*REGWORDS(sp)
STORE x8, 0x30*REGWORDS(sp)
...
...
```

Context Restore

```
...
...
LOAD x1, 0x70*REGWORDS(sp)
LOAD x7, 0x44*REGWORDS(sp)

addi sp, sp, 128*REGWORDS
mret
```

Linux Contribution

- V5 u-boot
  - merge to mainline and Rick is risc-v maintainer
  - https://github.com/u-boot/u-boot/blob/master/MAINTAINERS
  - Used by Open SUSE to port EFUI
- V5 Linux tools
- V3 Linux
  - Merged into 4.17
- V3 uClibc-ng
  - Upstream and merge to mainline

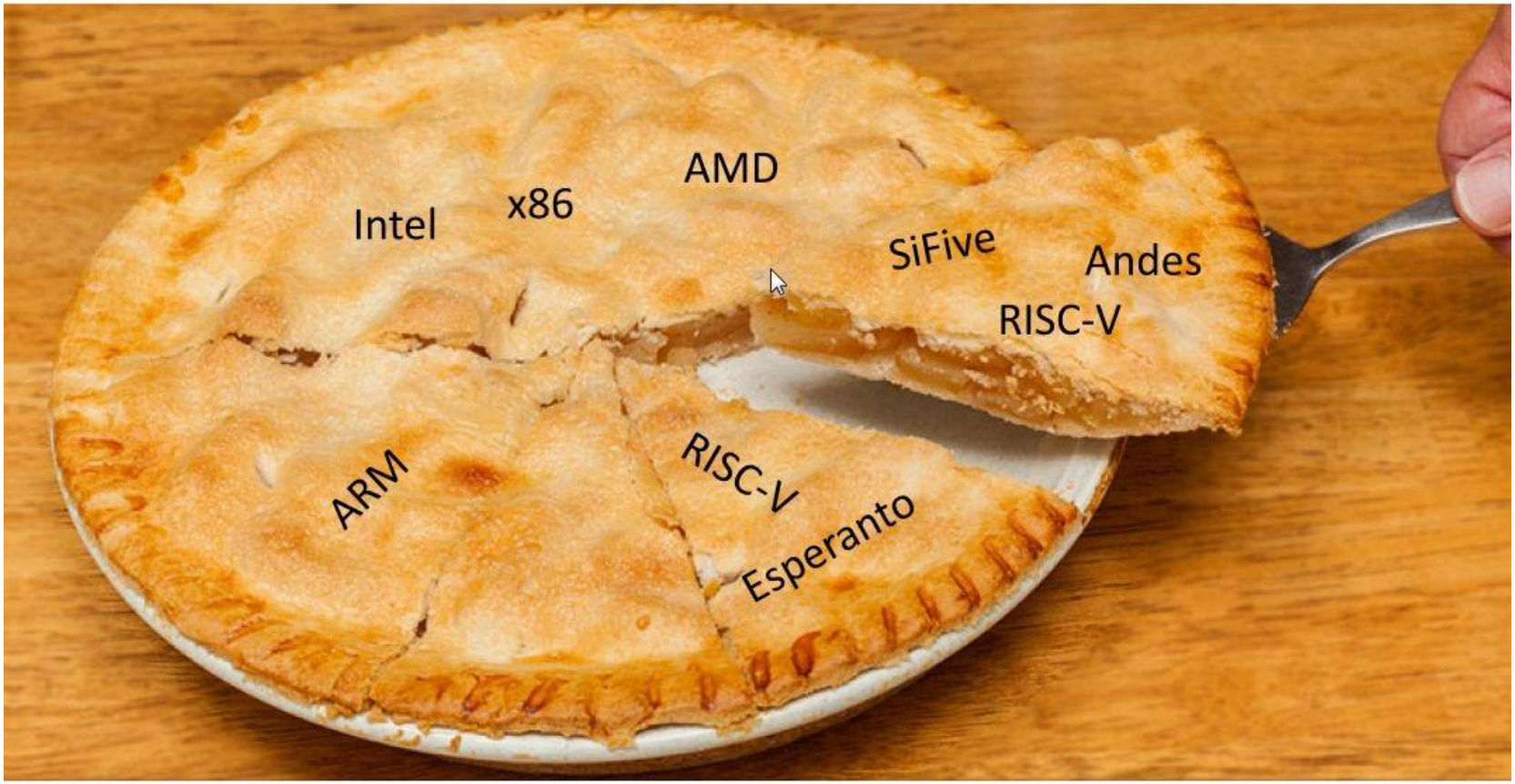
Additional Material

Concluding Remarks

- **RISC-V is emerging as a major application platform**
- **Andes offers comprehensive RISC-V solutions**
  - **V5 CPU cores:**
    - N25/NX25: fast-n-small core for control tasks
    - N25F/NX25F: FP cores for computation tasks and AI
    - A25/AX25: Application processors
  - **ACE for Domain-Specific Acceleration**
    - A separate option available for all V5 cores
  - **Strong tools and SW support**
- **Let us work with you for the next SoC projects**

Goal

- **Andes Mission : Taking RISC-V Maintream**
  - Help making RISC-V the new-generation application platform for computing devices

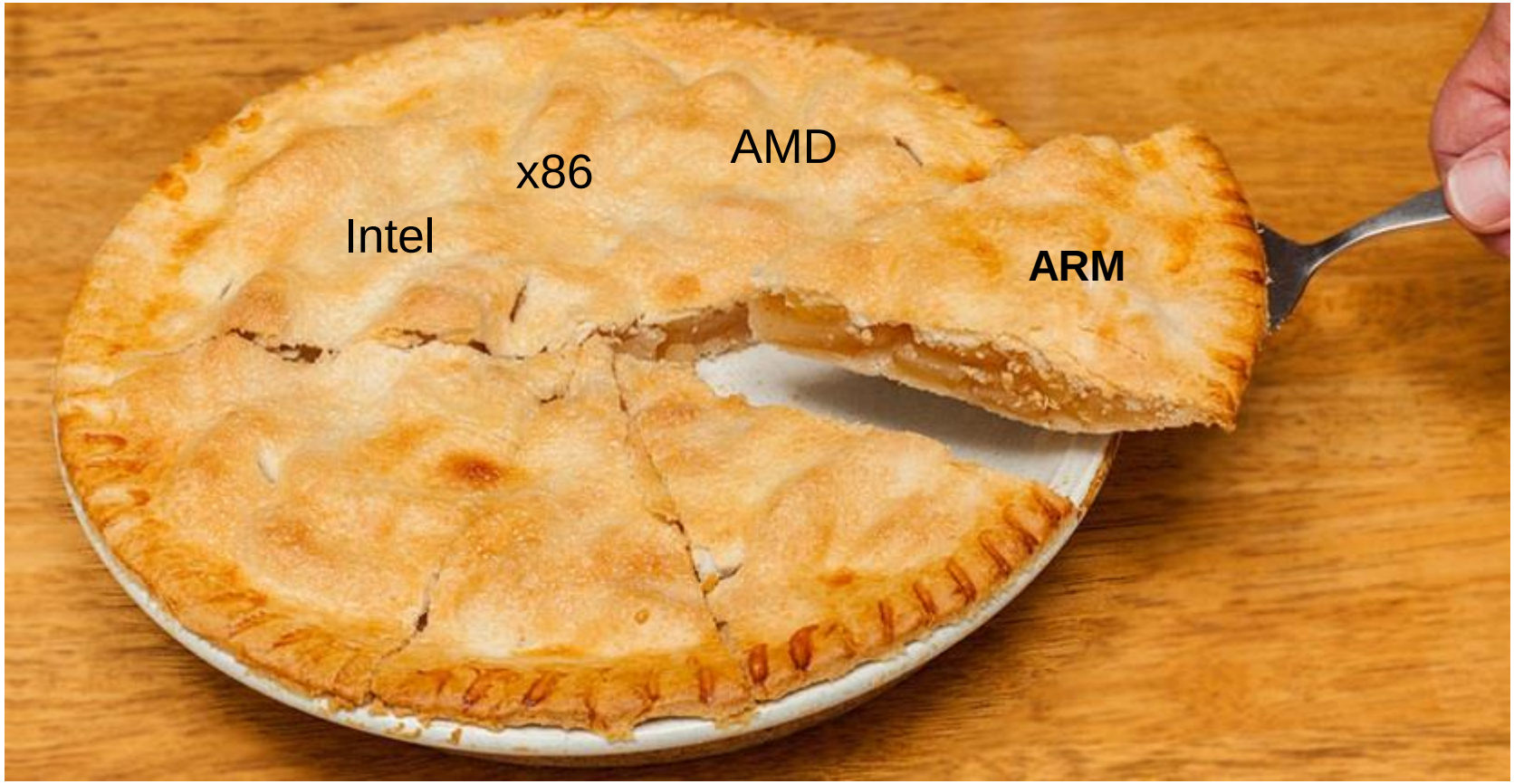


Performance Efficiency For Low Power

- **Efficient pipeline:**
  - General performance: >20% higher (e.g. 3.49 CM/MHz)
  - Hit-under-miss caches: optimize performance with minor additional logic
    - Continue execution when a miss fill is ongoing
  - Half-precision FP load/store to reduce memory footprint and cache misses
  - HW support for misaligned accesses:
    - Good for porting existing SW
    - Without it, >100 cycles are needed in the exception handler
- **Caches for low power:**
  - Only single-port SRAMs used to reduce its area and power
  - Designed for fast logic power-down and wakeup
- **RTL design for high clock-gating synthesis (98%)**

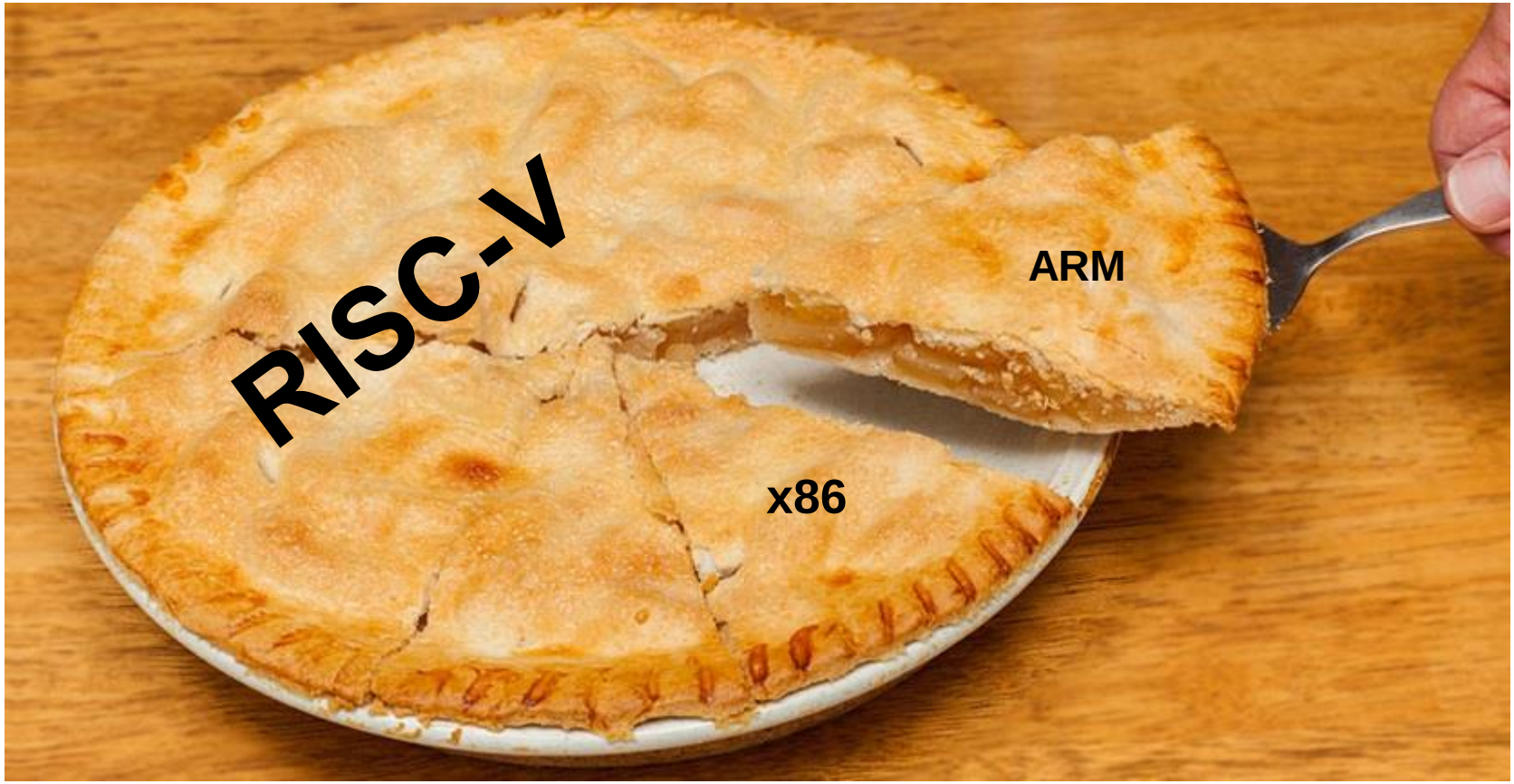
Goal

- **Andes Mission : Taking RISC-V Mainstream**
- Andes RISC-V solution: Much more than "good enough"
  - Don't be OpenRISC of the 2010s.



Goal

- **Andes Mission : Taking RISC-V Mainstream**
- Andes RISC-V solution: Much more than "good enough"
  - Don't be OpenRISC of the 2010s.



IoT Platform from Express Logic

- **Industrial grade connectivity For AndesCores**