Taking RISC-V® Mainstream

From Edge to Cloud

Charlie Su, Ph.D.
CTO and EVP
Andes Technology

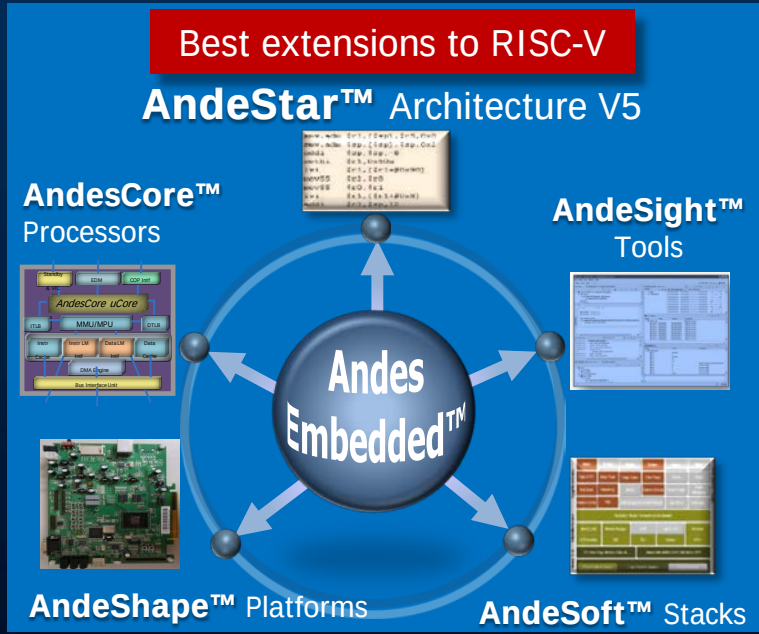
RISC-V Day Tokyo, 2019/09/30

Andes Corporate Overview

>1 Bn Annual Run Rate of Andes-Embedded SoC	• >180 commercial licensees
Silicon Valley Tie	• Core R&D from AMD, DEC, Intel, MIPS, nVidia, and Sun
14-Year CPU IP Company	• IPO in 2017
RISC-V Founding Member and Major Contributor	• Chairing Task Groups • Contributing to GNU, LLVM, uBoot, Linux, and others

Benefits of Andes V5 Architecture

- **V5: RISC-V + Andes Extensions**
- **V5 processors offer**
  - 20% higher per-MHz performance
  - 12% smaller code size
  - Rich HW features for embedded systems
    - Full cache management
    - StackSafe™ stack protection
    - QuickNap™/PowerBrake power management
    - Misaligned loads/stores like ARM and x86
  - Flexible configurations
  - Powerful tools for custom extensions



V5 Adoption: From Edge to Cloud

- **Applications:**
  - ADAS
  - AIoT at the edge
  - Blockchain
  - Communication: BT, WiFi, 5G
  - Datacenter AI accelerator
  - FPGA
  - MCU
  - Multimedia: A/V, AR/VR
  - SSD: enterprise/consumer
  - Security
- **50% use AI**
- **40nm to 7nm**

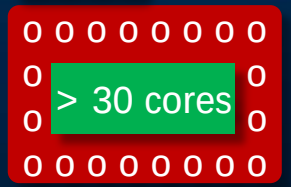
IOT Edge

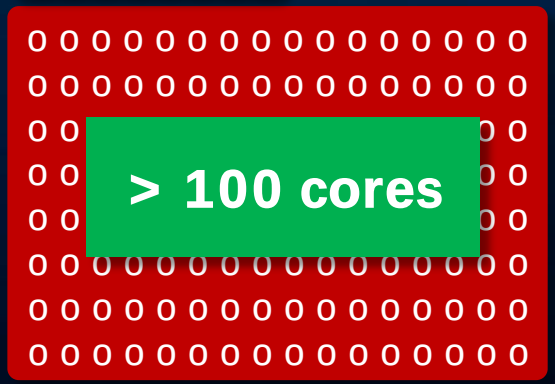
Consumer

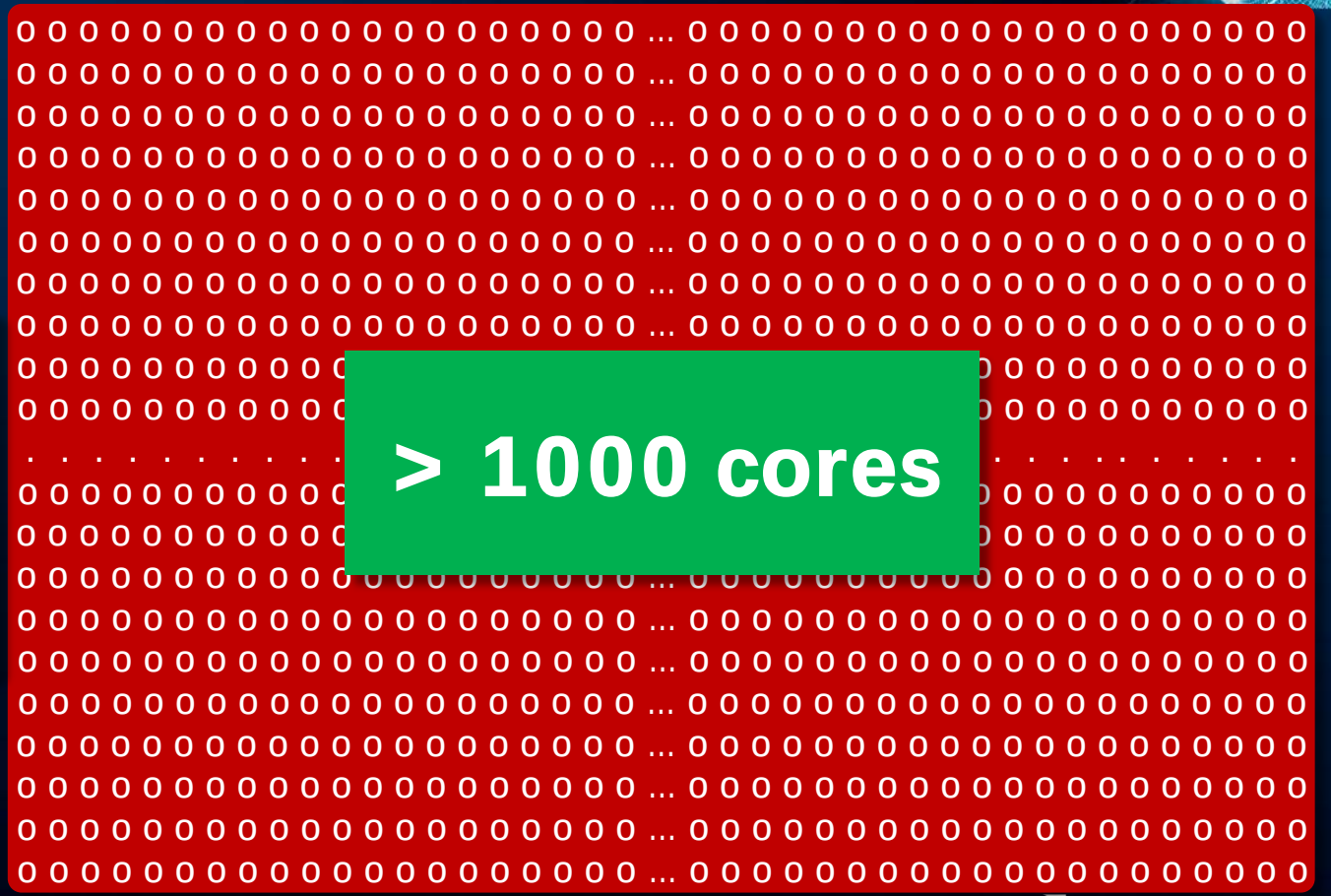
Cloud

V5 Adoption: Cores Per SoC

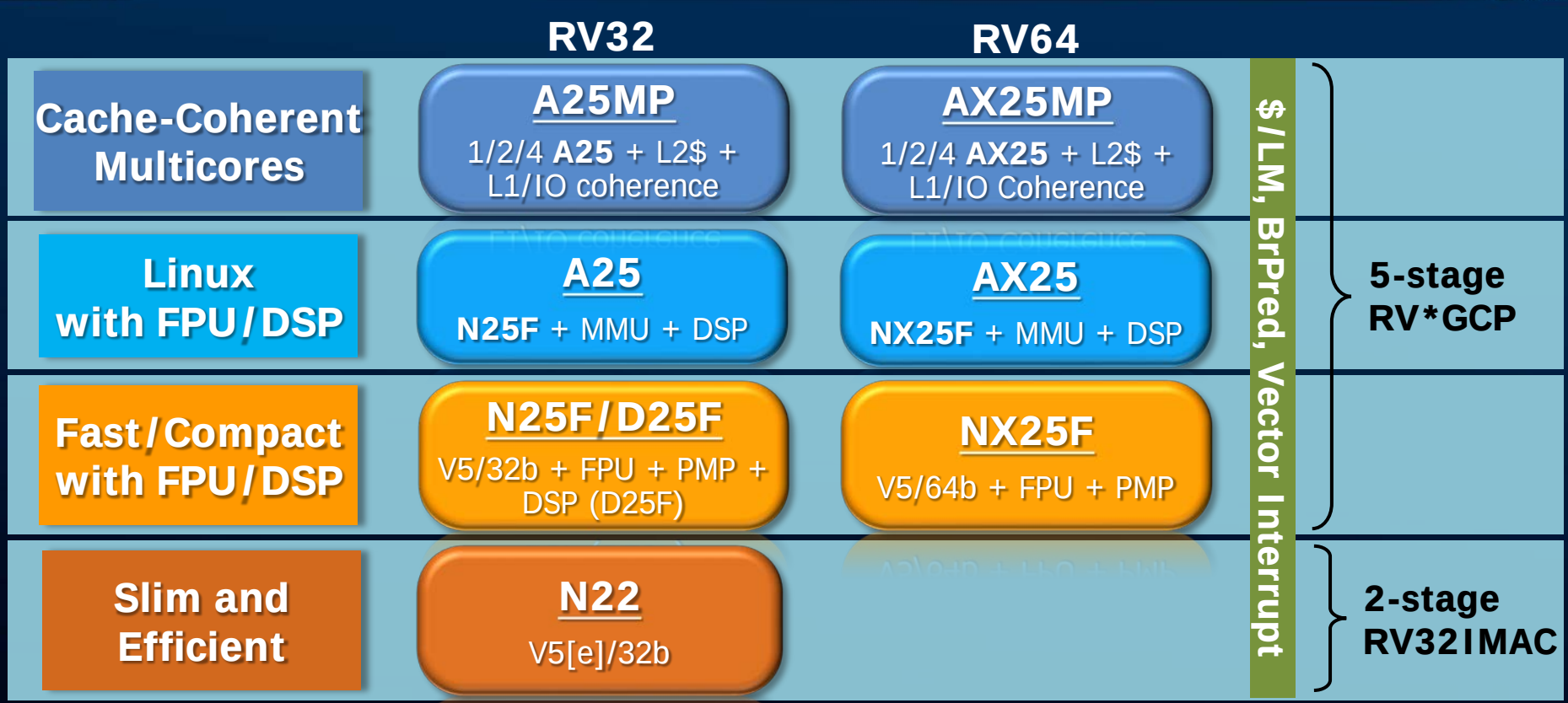
2-8 cores







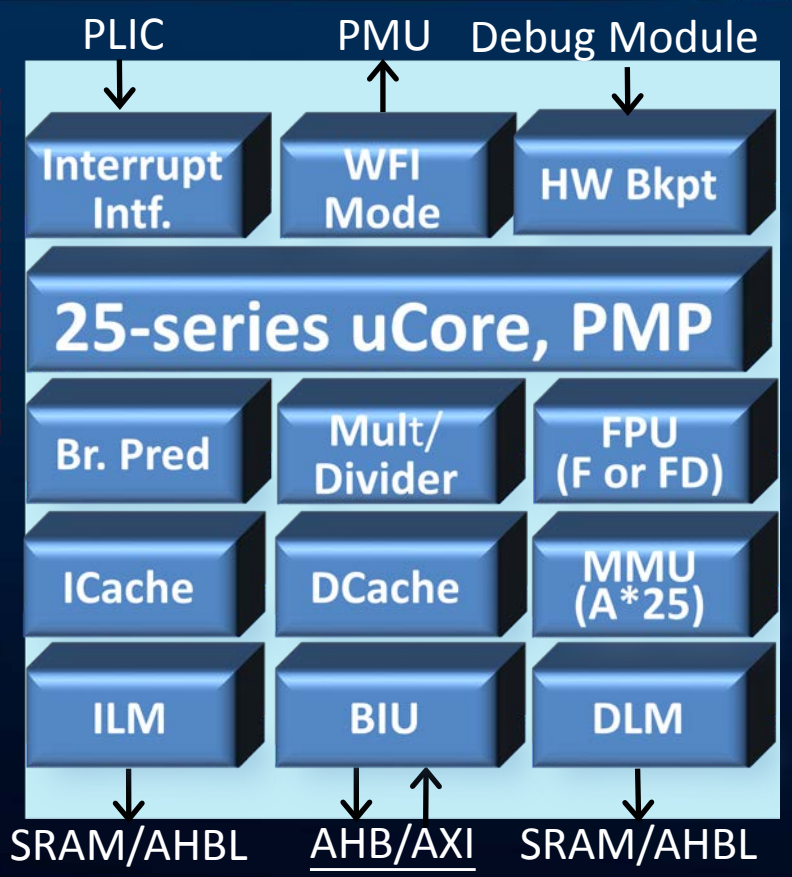
Andes V5 Processor Lineup



25-Series: Performance

Features	Base	FP	Linux	Linux
32KB I$/D$ + 256 BTB	Yes	Yes	Yes	Yes
SP/DP FPU	--	Yes	Yes	Yes
MMU and S-Mode	--	--	Yes	Yes
DSP/SIMD (RV-P ext.[1])	--	--	--	Yes
Worst-Case Freq. (GHz)[2]	1.2	1.2	1.2	1.1
Coremark/MHz[3]	3.57 (rv32), 3.53 (rv64)			
DMIPS/MHz (no-inline)[3]	1.97 (rv32), 2.13 (rv64)			
Whetstone/MHz (SP)[3]	1.58 (rv32), 1.56 (rv64)			
Whetstone/MHz (DP)[3]	1.30 (rv32), 1.38 (rv64)			

1: **RV-P extension draft**
2: **28nm** SVT 9T library and high-speed memory. Frequency at 0.81v/-40ºc.
3: **AndeSight v320** toolchain; DMIPS/ground rule uses no-inline option.



First RISC-V Cores with RV-P

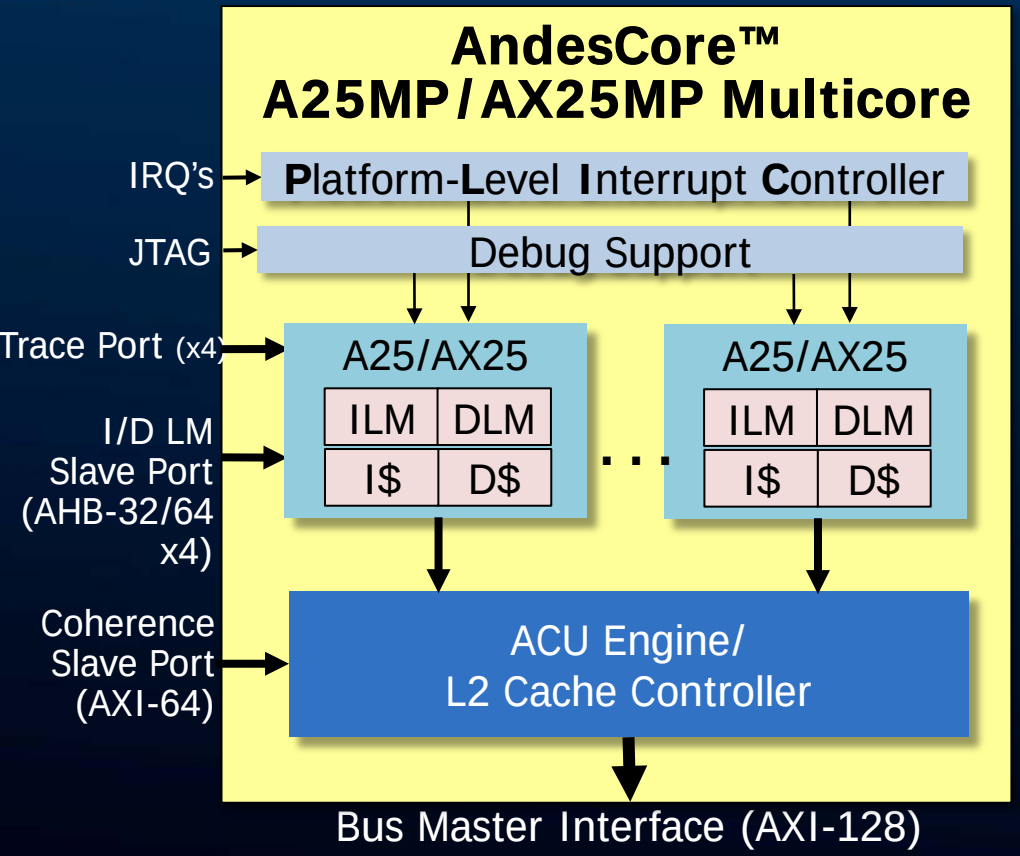
- **P-Extension ISA for efficient SIMD/DSP:**
  - 32= 32 + 8x8 + 8x8 + 8x8 + 8x8

  RV32 performs one set of the above; RV64 performs 2 independent sets

- **Speedups on 32-bit V5 cores**
  - MP3 decoder: **2.0**x
  - AMR voice codec: **3.7**x
  - Keyword Spotting (Tensorflow model): **5.2**x
  - >200 DSP libraries: **5.2**x and **1.8**x max./avg. speedups

- **Speedups on 64-bit V5 cores:**
  - Image classification (CIFAR10): **11**x
  - Face detection (P-net): **7.6**x
  - >200 DSP libraries: **11**x and **2.5**x max./avg. speedups

A(X)25MP: Cache-Coherent Multicore

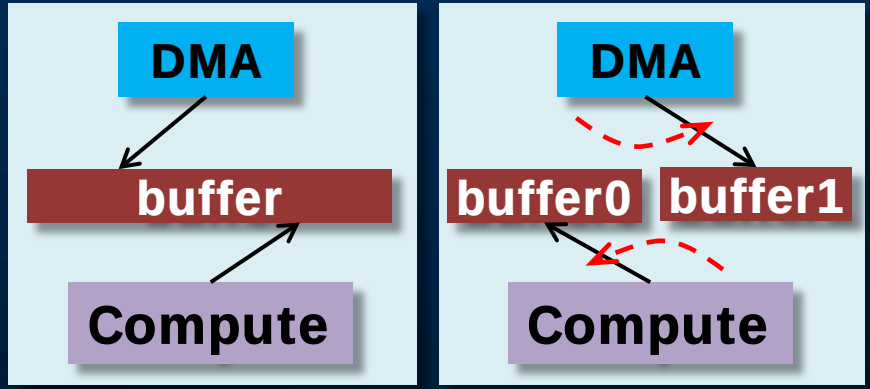


- **1~4 A25/AX25 CPUs:**
  - RV-GCNP + V5 extensions
  - P-extension draft
  - Supporting SMP Linux
- **Cache coherence with duplicated tags**
- **L2 cache controller**
  - 16-way with bank interleaving
  - I/D prefetch
  - Flexible SRAM timing
- **PLIC for interrupt handling**
- **Debug/trace support**

Accelerating Data-Intensive Computation

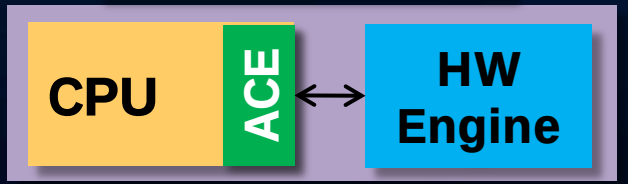
DMA for data transfers:

1. DMA with a single (ring) buffer:
   - with contention
2. DMA with double buffers:
   - Switch buffers on phase changes
   - No contention



ACE focuses on compute acceleration.

1. ACE to perform computation. E.g.
   - Dot products of 2 64x8 vectors, or matrix convolution
2. ACE to control existing HW Engine. E.g.
   - Send a 90-bit (or wider) command in one cycle
     - Can be viewed as an emulation of VLIW



Andes Custom Extension™ For DSA

- C code
- Verilog
- Attributes

- scalar/vector
- direct IO
- wide operands

COPILOT

Custom-OPtimized Instruction develOpment Tools

Automated Env. For Cross Checking

Test Case Generator

Extended ISS

Extended RTL

Extended Tools

Extended ISS

Extended RTL

Compiler
Asm / Disasm
Debugger
IDE

CPU ISS
(near-cycle accurate)
Standalone or
SystemC

CPU RTL

Extensible Baseline Components

A new CPU and its tools

Inner Product of Vectors with 64 8-bit Data

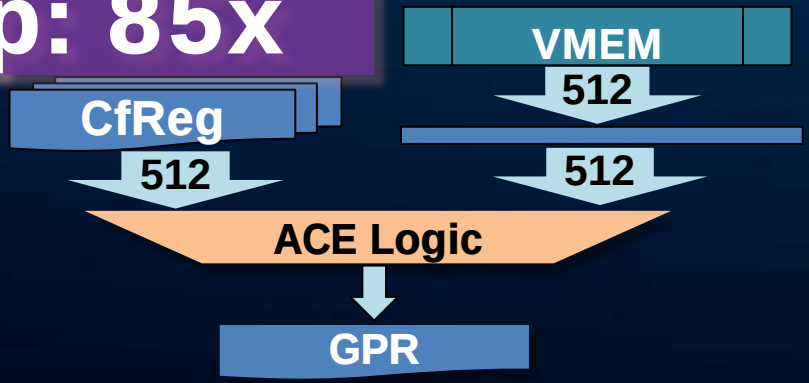
```
reg CfReg {              //coef. Custom Register
    num= 4;
    width= 512;
}
ram  VMEM {              //vector Custom Memory
     interface= sram;
     address_bits= 3;          //8 elements
     width= 512;
}
insn ip64B {
  operand= {out gpr IP,
                in CfReg C, in VMEM V};
  csim= %{               //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
};
```

ip64B.ace

```
//ACE_BEGIN: ip64B
assign IP= C[ 7:0]   * V[ 7:0]
         + C[15:8] * V[15:8]
             . . .
         + C[511:504] * V[511:504];
//ACE_END
```

ip64B.v

Speedup: 85x

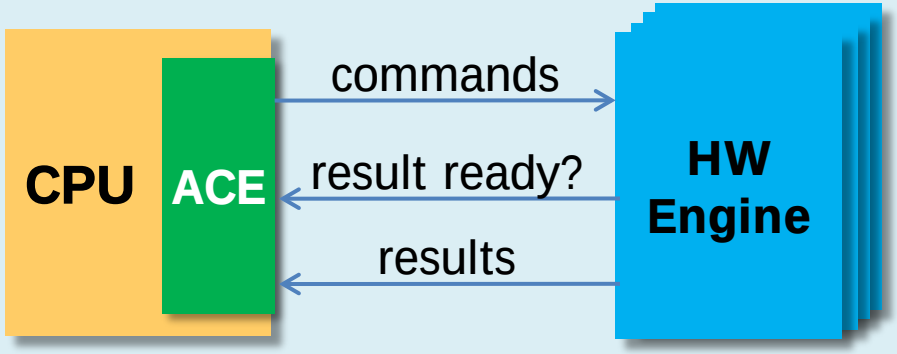


Intrinsic: long **ace_ip64B**(CfReg_t, VMEM_t);

Custom Port (ACP) for Direct HW Engine Control

```
port command {      //a 90-bit output port to
                    // all 4 HW engines
    width= 90;      //including a valid bit and
                    // a HW engine ID
    io_type= out;
}
//4 HW engines
port ready {        //4 ready signals
    num=4;
    io_type= in;
}
port results {      //4 256-bit input ports
    num= 4;
    width= 256;
    io_type= in;
}
```

CPU controls 4 HW engines.



App. code sequence:

prepare command (say, thru ACR);
send command;
do other useful work;
wait for results to be ready;
get results;

Andes RISC-V Serving Emerging SoC from Edge to Cloud

Performance	• Leading PPA and smallest code size • First to have complete SIMD/DSP instructions
Configurability & Extensibility	• Flexible configurations on rich features • Tools to simplify design of custom instructions
Maturity	• Optimizations of compiler and libraries • Comprehensive features of IDE • Product packages and custom engineering

Thank you