Status update of RISC-V P extension task group

Chuan-Hua Chang
Senior Director
Andes Technology Corp.

2019 RISC-V Workshop Zurich

RISC-V DSP (P) Extension TG

- **P extension task group charter**
  - Define and ratify Packed-SIMD DSP extension instructions operating on XLEN-bit integer registers for embedded RISC-V processors.
  - Define compiler intrinsic functions that can be directly used in high-level programming languages.
- **Chair: Chuan-Hua Chang, Andes Technology**
- **Co-chair: Eric Flamand, Greenwaves Technology**

RISC-V DSP (P) Extension Proposal

- **DSP instruction set proposal based on AndeStar™ V3 DSP ISA.**
  - Use RV32 and RV64 XLEN-bit GPRs.
  - Support saturation and rounding.
  - Support fixed-point and integer data types.
  - **SIMD**-instructions with 8b, 16b, 32b element size.
  - **Partial/Non-SIMD** DSP instructions operating on 8-bit, 16-bit, 32-bit and 64-bit data types.
  - 64-bit signed/unsigned addition & subtraction (RV32)
  - 64-bit addition with 16b/32b multiplications
    - E.g., 64 = 64 + 16x16 + 16x16
    - E.g., 64 = 64 + 32x32

GPR vs Separate Register

- GPR-based SIMD is a more efficient, low power DSP solution for embedded systems running applications in various domains such as audio/speech decoding and processing, IoT sensor data processing, wearable fitness devices, etc.
- It addresses the need for **high performance generic code processing**, as well as **digital signal processing**.

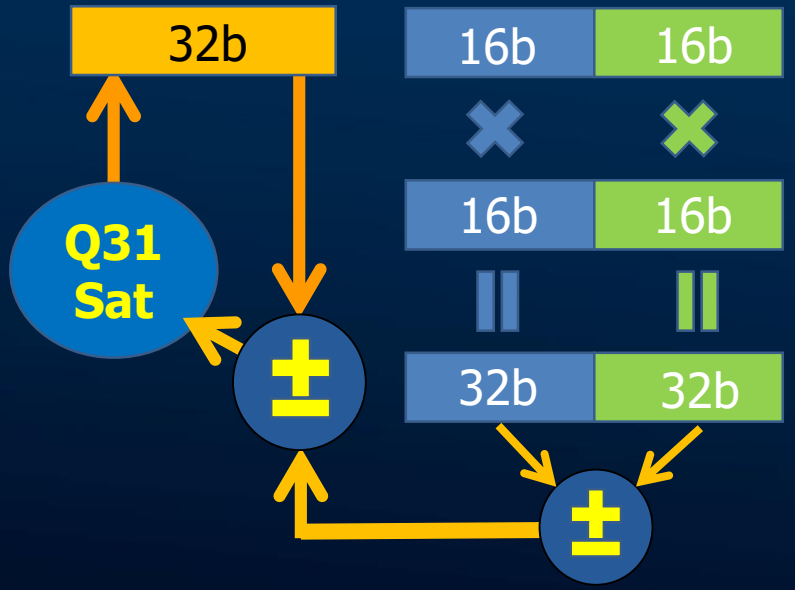
16-Bit SIMD Instructions

- \- + ×
- << Min Max
- >> Clip ABS
- Compare
- Signed
- Unsigned

8-Bit SIMD Instructions

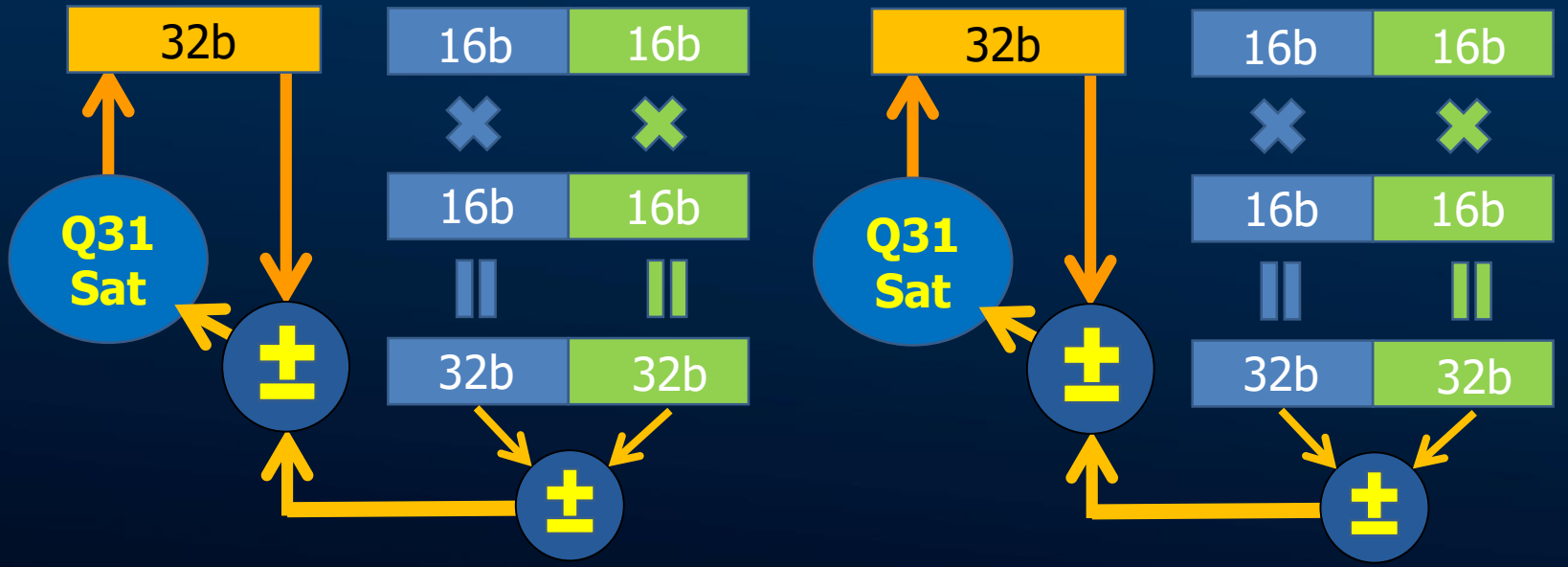
- Min
- Max
- Unpack
- ABS
- Compare
  - Signed
  - Unsigned

Dual 16x16 & 32-Bit Add/Sub (RV32)



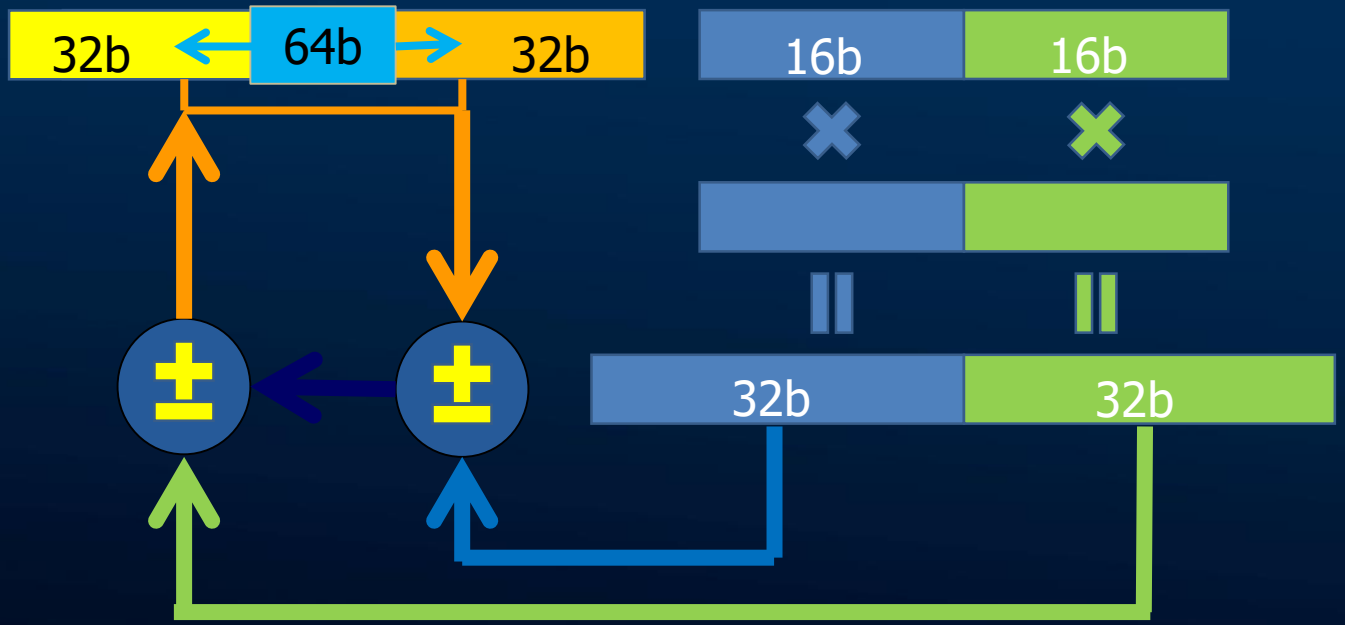
6x Baseline operation

2 Dual 16x16 & 32-Bit Add/Sub (RV64)



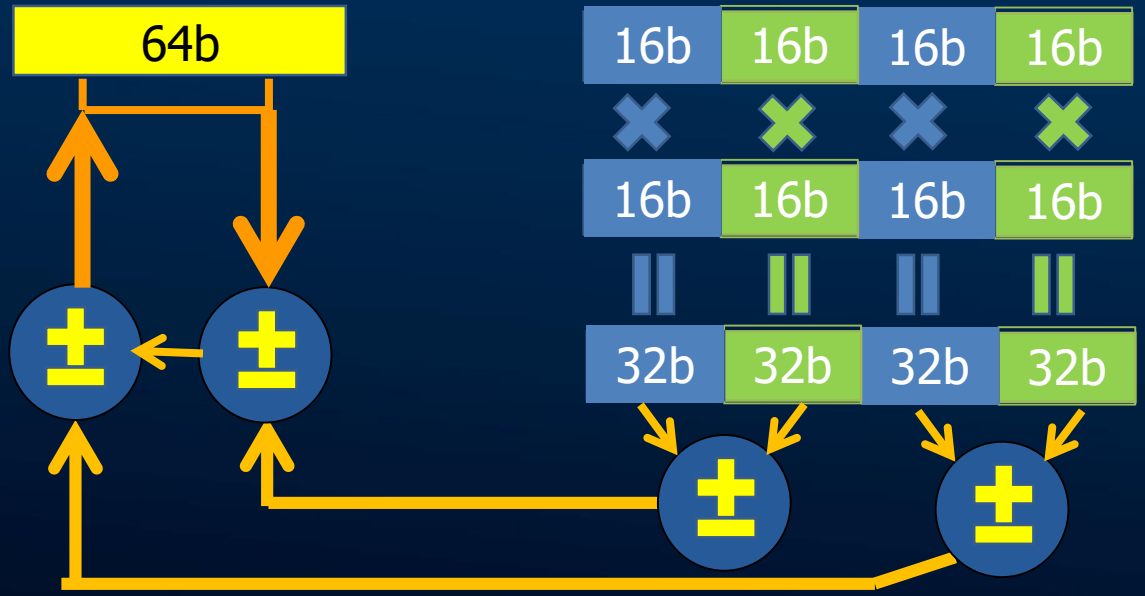
12x Baseline operation

Dual 16x16 & 64-Bit Add/Sub (RV32)



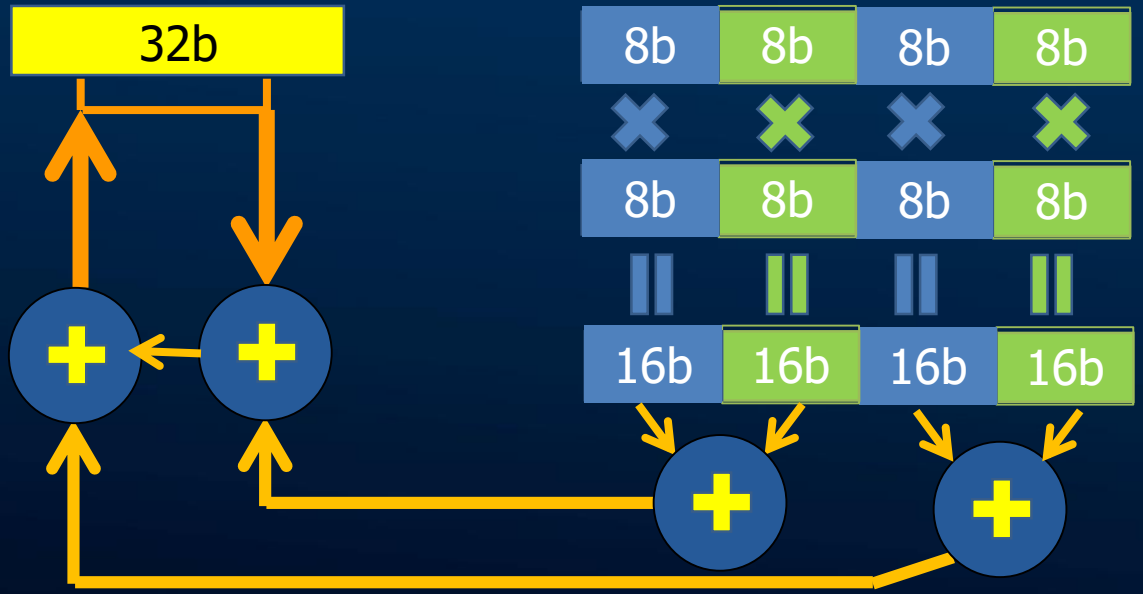
2.6x Q15 dot prod

Quad 16x16 & 64-Bit Add/Sub (RV64)



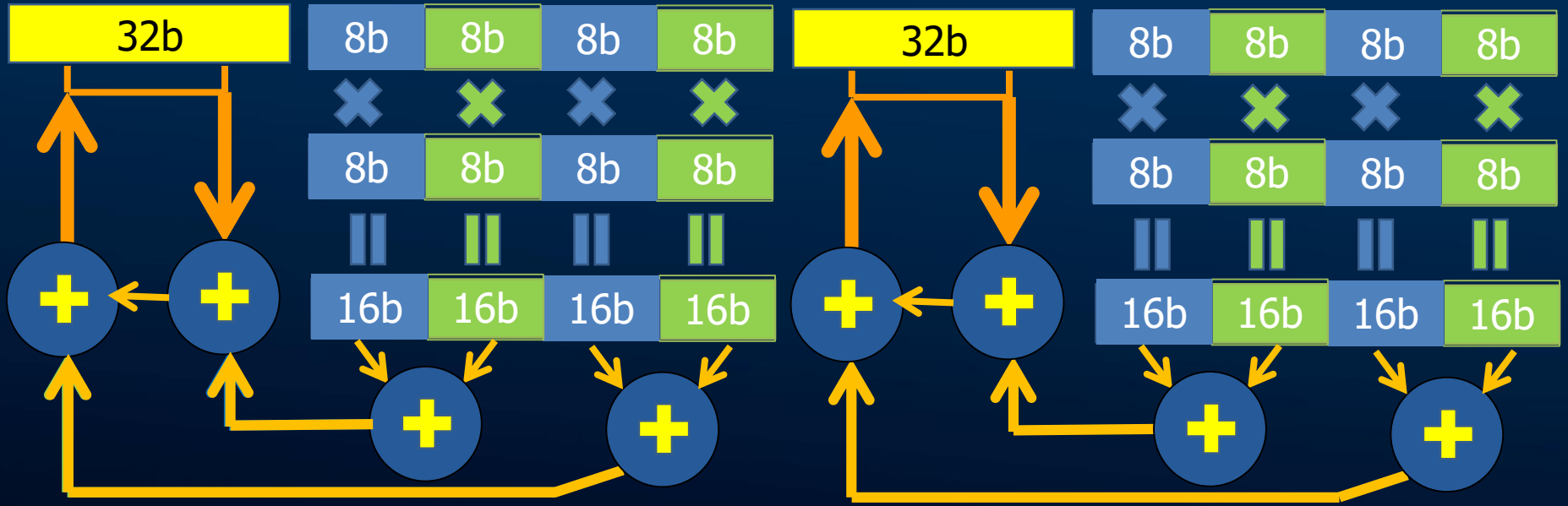
3.6x Q15 dot prod

Quad 8x8 & 32-Bit Add (RV32)



3.4x Q7 dot prod

Dual Quad 8x8 & 32-Bit Add (RV64)



7.7x Q7 dot prod

64-bit Data Type

- Use pairs of GPRs on RV32.
- Use a GPR on RV64.
- Needed for compiler to generate DSP instructions automatically.
- The 64-bit operand type is an interface specification. An implementation can still implement 2R1W register file with multi-cycle reads/writes to support the 64-bit type on RV32.

DSP Library Speedup

RV32

Speedup		Basic	Complex	Controller	Filtering	Matrix	Statistics	Transform	Utils	Average /Max
P / Base (RV32)	AVG	2.4	1.62	1.84	2.26	1.62	2.44	1.29	1.08	**1.82**
	MAX	5.16	4.09	2.13	4.11	2.75	4.39	1.78	1.43	5.16

RV64

Speedup		Basic	Complex	Controller	Filtering	Matrix	Statistics	Transform	Utils	Average /Max
P / Base (RV64)	AVG	4.73	1.92	1.31	2.41	3.04	4.14	1.28	1.19	**2.5**
	MAX	10.81	4.14	1.59	5.04	6.83	8.51	1.67	2.72	10.81

Helix MP3 Decoder

- Compiler only, no hand optimization

GCC Compiler	Decode (MCPS)
Compile with RV32IMC ISA	20.78
Compile with RV32IMC + P ISA	10.65
Speedup	1.95

GCC Compiler	Decode (MCPS)
Compile with RV64IMC ISA	14.97
Compile with RV64IMC + P ISA	11.61
Speedup	1.29

* MCPS: Millions of Cycles Per Second

Voice Codec (P vs Baseline)

	RV32	RV32+P	Speedup	RV64	RV64+P	Speedup
AMRWB En	244.65	66.45	3.68	249.72	70.23	3.56
AMRWB De	82.92	22.24	3.73	83.87	23.52	3.57
G729 En	56.39	15.75	3.58	59.42	19.25	3.09
G729 De	15.43	5.17	2.98	16.46	6.66	2.47
AMRNB En	25.96	22.72	1.14	26.47	22.72	1.17
AMRNB De	4.94	4.41	1.12	5.14	4.41	1.17

* MCPS: Millions of Cycles Per Second

Neural Network (P vs Baseline)

	RV32	RV32+P	Speedup	RV64	RV64+P	Speedup
CIFAR10	54,002,847	9,765,828	5.53	66,726,795	6,077,522	10.98
GRU	31,545	16,981	1.86	32,892	11,144	2.95
PNET	165,274,904	33,550,898	4.93	176,241,323	23,285,489	7.57

* Cycles of Execution

P Task Group Progress

- **Created P extension instruction proposal spreadsheet for TG members to review.**
- **Benchmarking on DSP library functions for the usefulness of these instructions.**
  - About ~100 instructions are used in DSP library and audio/speech codec optimizations.
- **Released detailed instruction operation specification.**
- **Released toolchain and simulator binaries for TG members to evaluate the use of these instructions.**

Thank you