RISC-V P Extension Implementation and DSP Application Practice

Dmitry Zakharov
08.2024

CloudBEAR Introduction

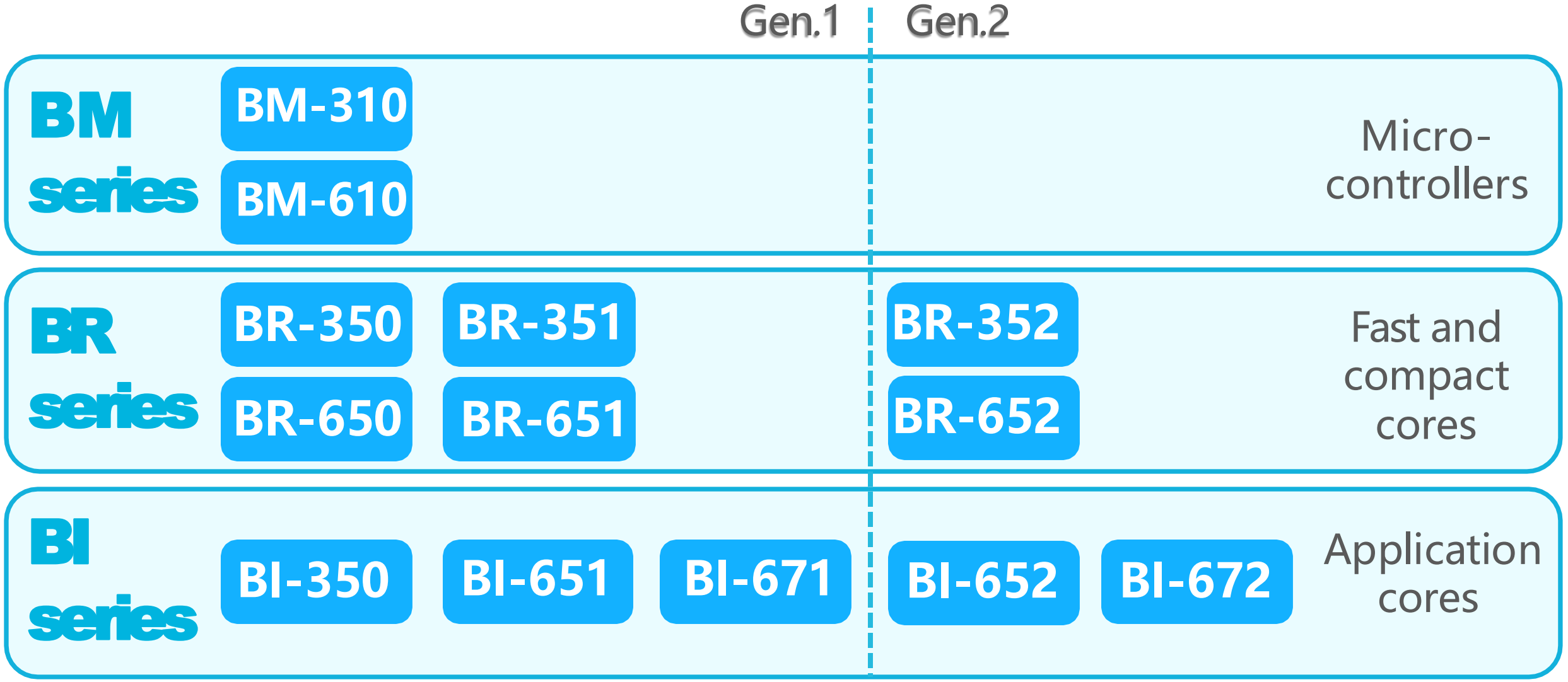
- RISC-V IP company
- Est. 2015

- Gen.1 IP multiple time licensed
- Gen.1 IP based silicon in mass production

- Gen.2 cores are available for licensing

RISC-V Processor IP

CloudBEAR



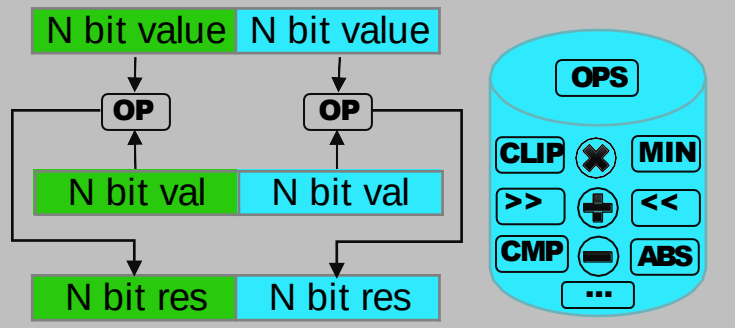
Introduction

What is a P extension?

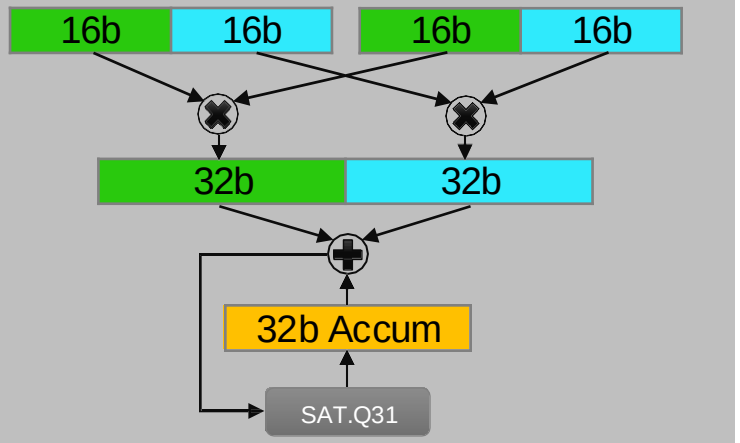
RISC-V [P]acked SIMD Extension

CloudBEAR

- Instruction set to **accelerate integer and fixed point math** using general purpose registers (GPR).
- **~330 instructions** for SIMD, Partial-SIMD, Non-SIMD operations.
- Supports
  - RV32 and RV64 architecture
  - [8, 16, 32, 64] bit depth data
  - **Q notation** data (Q7, Q15, Q31)
  - **Merged math** operations to combine multiply, accumulate, shift, round, saturate.
- P extension is useful to **speed up DSP** algorithms at edge devices with limited resources



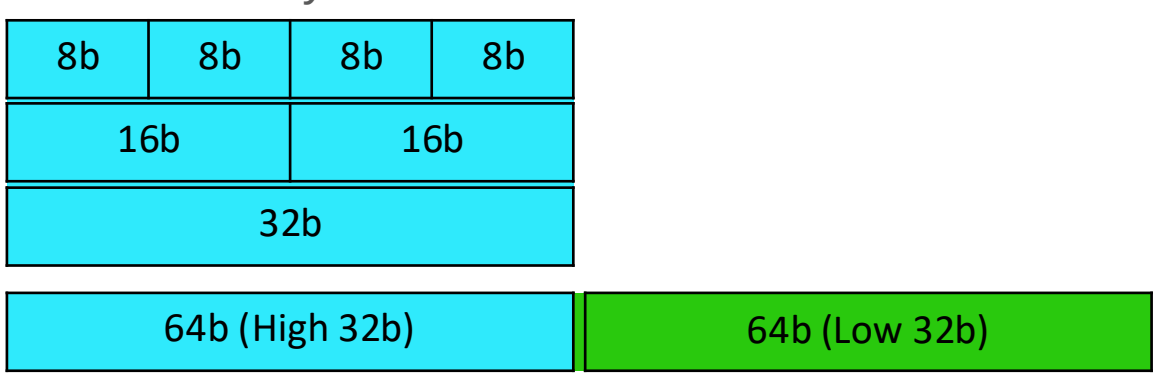
Calculation Up to 4 values in parallel (RV32)



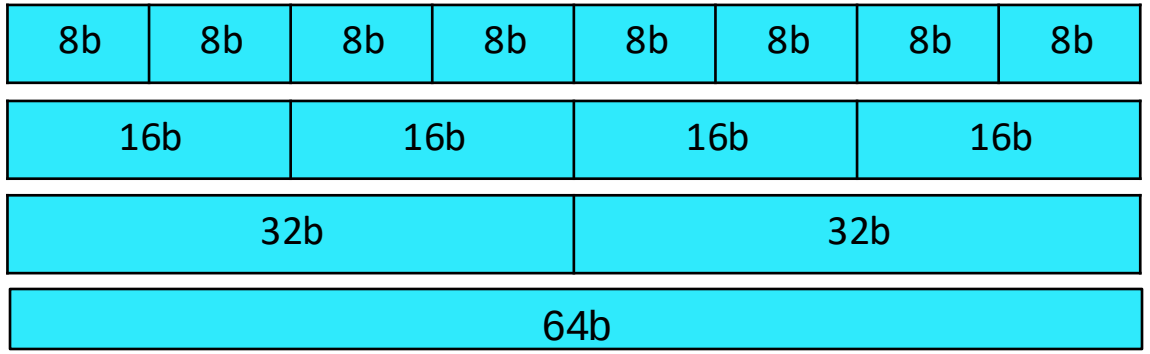
6 Basic operations in 1 instruction (RV32)

Programming Model

- The instructions operate on XLEN general purpose integer registers (GPRs).
- Registers are considered as small vectors divided into N values of a given bit depth. The specific interpretation of input and output registers is determined only by the instruction itself.
- An operation is performed on all values in the vectors. The output vector format can be preserved (N-to-N) or changed (N-to-M or N-to-1).
- RV32 supports 64-bit data. Such operands are composed of an even-odd pair of 32 bit GPRs. Only the even register is used explicitly in the instruction .
- Overflow/saturation is reflected in the VXSAT control-status register (CSR). No other configuration CSRs are added by the extension.



RV32: xN GPR **RV32: x(N+1) GPR**



RV64: xN GPR

Retrospect of Key Events

[2016.12]	A decision on Packed SIMD ISA extension development has been made[1]
[2017]	Andes contributed own Packed SIMD design proposal as a base for RISC-V P
[2018]	RISC-V Packed-SIMD (P) extension Task group officially started
[2021.12]	The specification version 0.9.11[2] has been published. After that, development slowed down
[2022.09]	An active revision of the specification has begun
[2024.05]	New Packed-SIMD Extension Chairs
[2024.08]	Specification is in development

- The specification has increased from ~150 instructions in the initial proposal to approximately 330.
- After a period of stagnation, work on the specification has resumed. John Hauser is now involved[3].
- **CloudBEAR** team has successfully **implemented P v0.9.11 in 2022** and It has already been licensed by our customers.

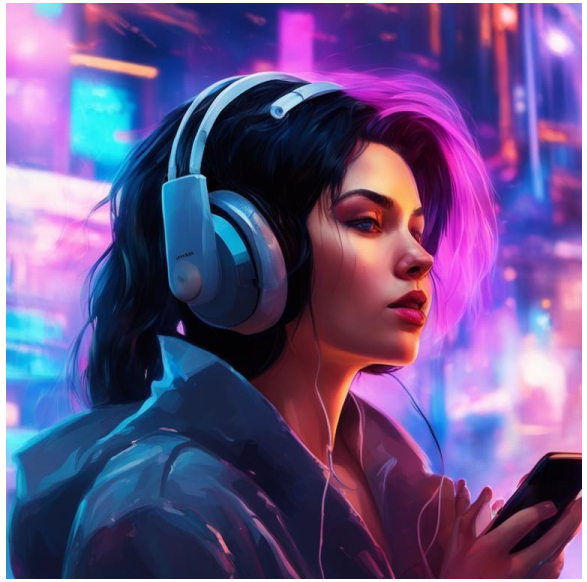
[1] 5th RISC-V Workshop Dec 2016.
[2] https://github.com/riscv/riscv-p-spec
[3] http://www.jhauser.us/RISCV/ext-P/RISCV-20220911-P-extension.pdf

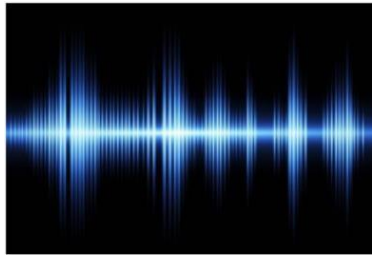
Applications

Where is it used?

Application Areas

- Audio/Speech Processing
- Motor Control
- Tiny ML
- Codecs
- Sensors Signal Processing





Solutions with P Extension

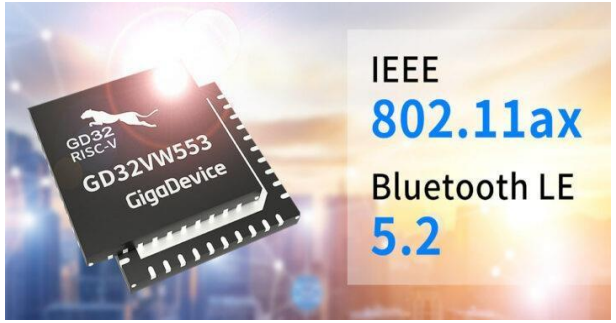
At least **4 vendors** provide P extension in commercial cores.

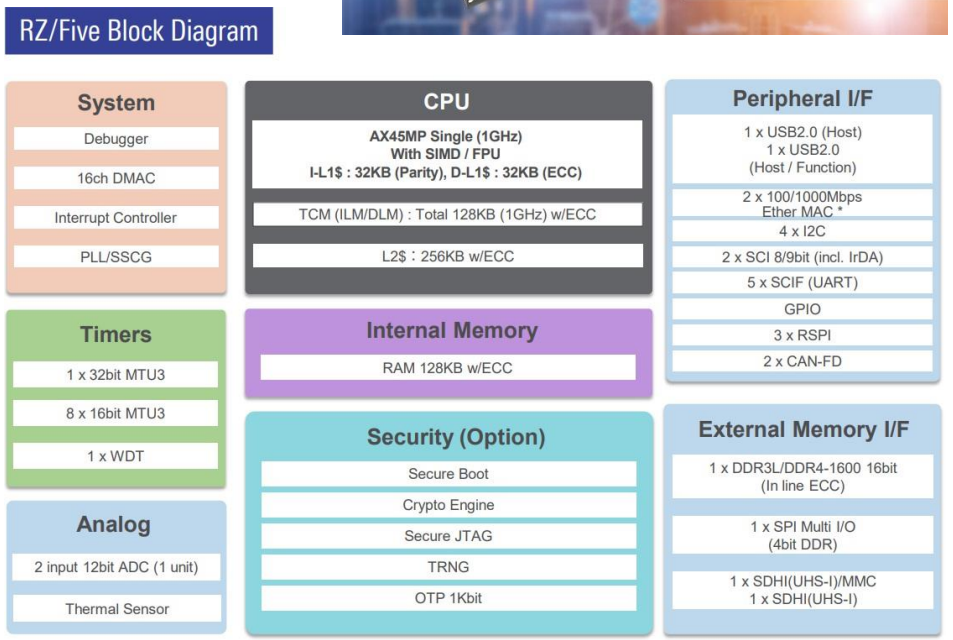
Several released processors with P extension:

- **Telink TLSR9 Series:** RV32IMAFDCPB @ 96 MHz
  Audio & IoT Chip
- **Gigadevice GD32VW553:** RV32IMAFDCPB @ 160 MHz
  IoT chip with wide range of wireless interfaces
- **Renesas RZ/Five:** RV64GCPN @ 1 GHz
  Civil Infrastructure Platform (CIP) Linux core for industrial applications

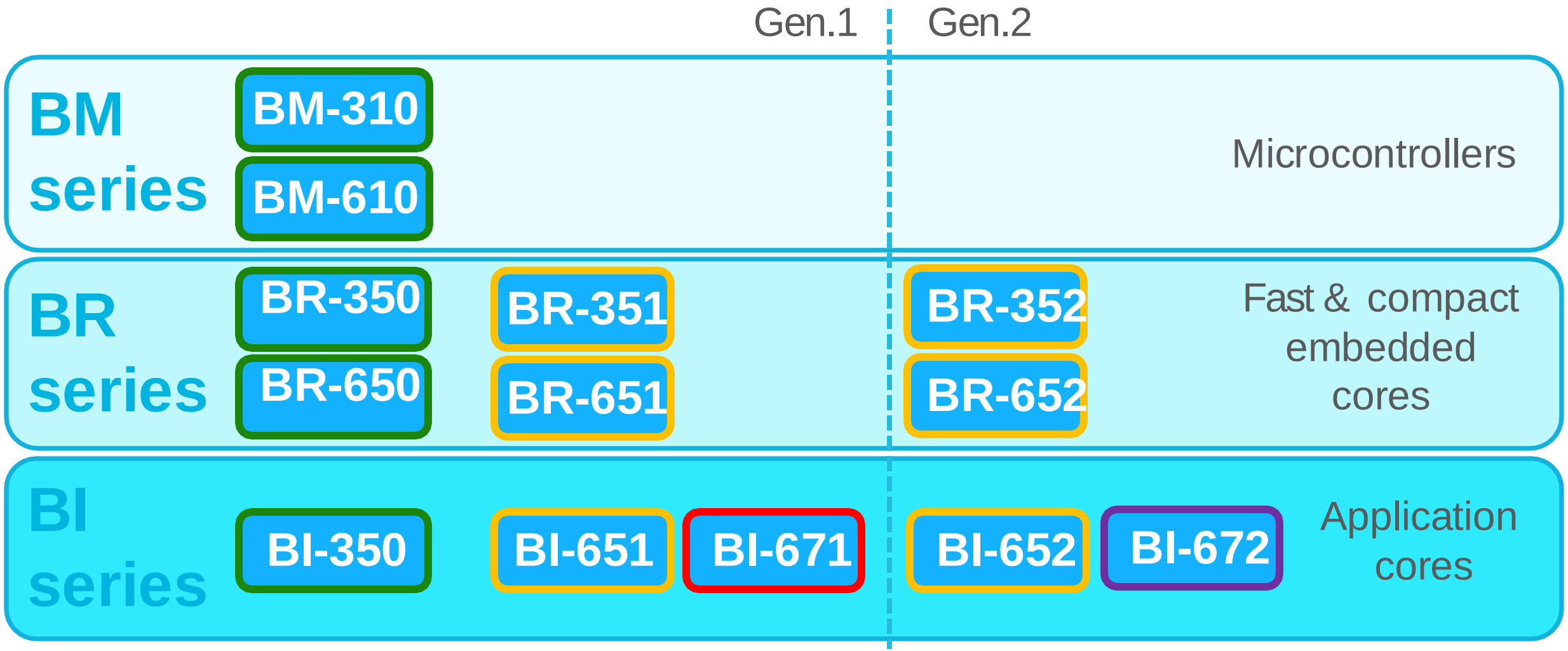
CloudBEAR Customers usecase:

- **SOC A (Dual core BR-350):** RV32IMACBKFPN @ 200 MHz
  General purpose MCU with motor control features
- **SOC B (Dual core BR-351):** RV32IMACBKFDPN @ 200 MHz
  Motor control MCU





RISC-V Cores Product Line



CMSIS-DSP

CloudBEAR

A suite of common signal processing functions for use on Cortex-M and Cortex-A processor based devices.[1]

13 subsets of functions, including:

- Basic Math: Pointwise operations (add, sub, mul, bitwise, etc)
- Transform: FFT, DCT, MFCC
- Filtering: FIR, Biquad IIR, Convolution, Correlation, etc.
- Motor Control: PID, Park and Clark Transforms, Sine Cosine
- Classic ML: SVM, Bayes classifier.
- .. 8 more groups ..

Supported data types:

- Fixed point: [q7, q15, q31]
- Floating point: [f16, f32, f64]

CMSIS-DSP compatible libraries are **provided by many vendors** who offer **embedded cores with DSP** acceleration. This is also true for the RISC-V ecosystem.

[1] https://github.com/ARM-software/CMSIS-DSP

BEAR-DSP

CloudBEAR

BEAR-DSP library is **a collection of optimized DSP primitives** with intensive usage of P extensions and C level API functions.

Library Model

- **A set of C functions** each of which implements one specific pre-optimized DSP primitive.
- **A set of P-extension intrinsics** for use in user specific algorithms.
- Library **sources as a set of examples** for intrinsic usage and optimization approach.

Library Status

Version 1.0 is about to be released. Scope of this version:

- Subset of most requested **CMSIS-DSP** compatible functions.
- Support of fixed point and single precision floating point data **[q7, q15, q31, fp32]**.
- **98 functions** are included, **49 of which have been optimized** with P

BEAR-DSP v1.0.0

P extension intrinsics
Controller Q31
Fast Math Q31
Filtering (IIR Biquad) Q31
Filtering (IIR Biquad) FP32
Controller FP32
FastMath FP32
Basic Math Q31|Q15|Q7
FastMath Q15
Transform (FFT) Q15|Q31
Filtering (FIR) Q15|Q31
Matrix FP32

Performance

How efficient it is?

Benchmark Environment

CloudBEAR

Evaluation cores:

BM-310: Compact single issue core with 3-stage pipeline
BR-351: Fast 32-bit Gen1 dual Issue core with deeper pipeline
BR-651: RV64 version of BR-351

CMSIS-DSP Test Framework[1]

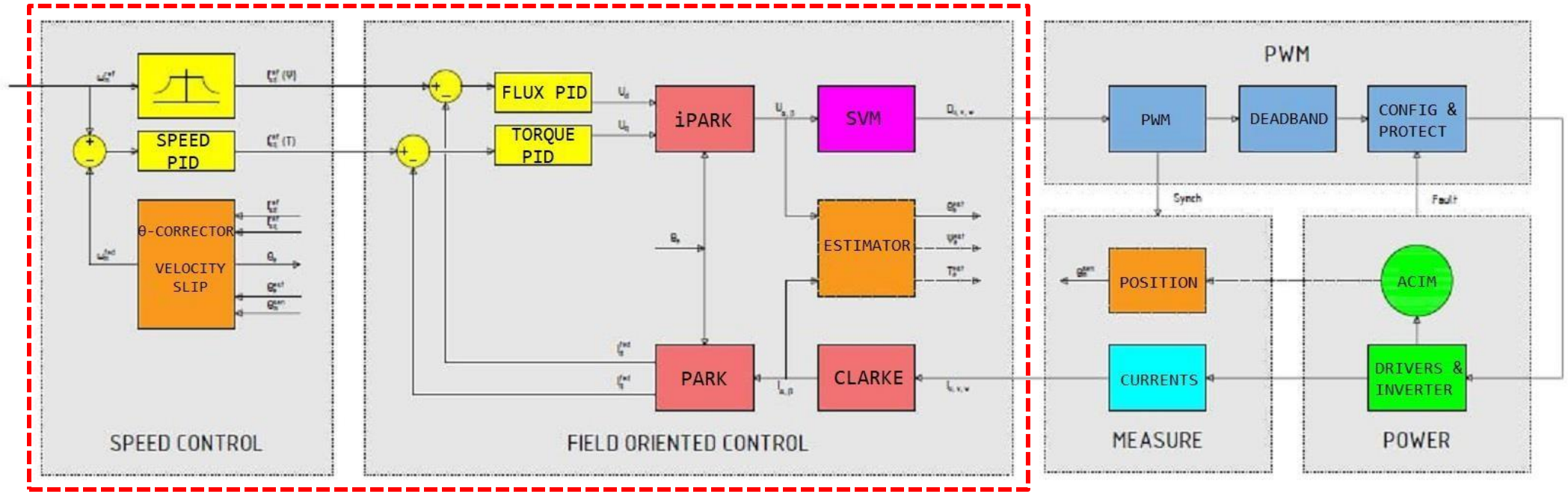
The test framework within the CMSIS-DSP repository.

```
Group : Root (1)
  Group : DSP Benchmarks (1)
    Group : Spectral Transformations (9)
      Suite : Mel Frequency Cepstral Coefficients Q15 (1)
        Profile MFCC (test_mfcc_q15 - 1) : PASSED (cycles = 49051) 256,20,13
                        <- 256 input points; 20 fbank; 13 output features Profile MFCC
        (test_mfcc_q15 - 1) : PASSED (cycles = 50347) 256,20,20
        ...
```

[1] https://github.com/ARM-software/CMSIS-DSP/blob/main/Testing/README.md

Customer Use Case: Motor Control

The evaluation prototype of the motor control system: Speed Control and Field Oriented Control submodules.



Target Algorithms:

- Park and Clark Transforms
- PID Controller
- Space Vector Modulation (SVM)
- CORDIC algorithm for nonlinear functions
- Sine/Cosine

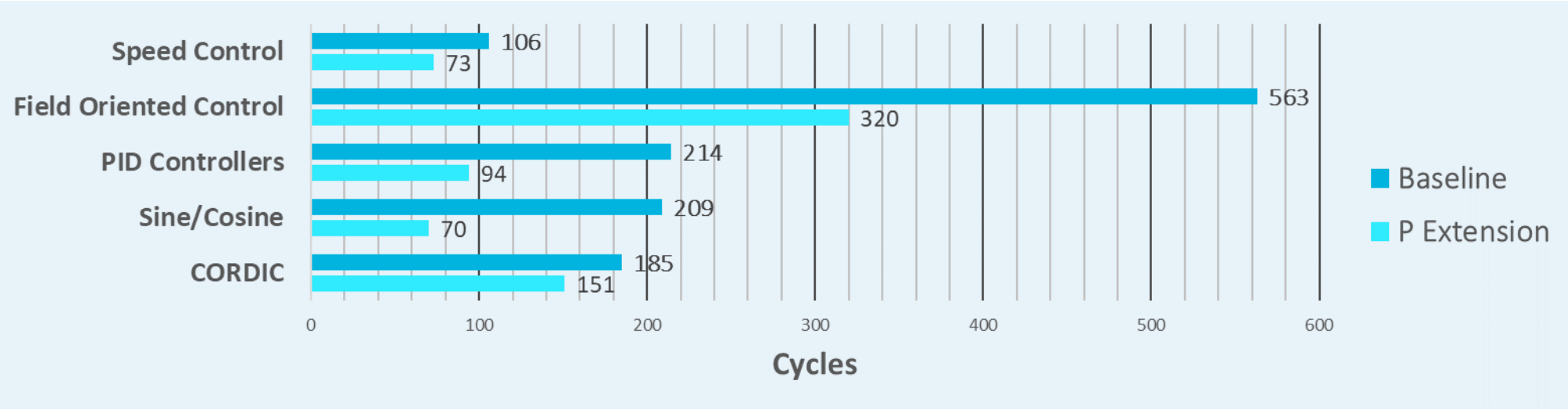
Motor Control Performance

CloudBEAR

Motor Control Prototype Performance

	Cycles per Iteration	Speed Up to Baseline	Cycles count reduction[%]
BM-310	554	x1.55	35%
BR-351	393	x1.70	41%

BR-351 Performance Details



BEAR-DSP Speedup

CloudBEAR

Speedup - The ratio of optimized code execution cycles to reference code execution cycles.

Function	Speedup[1]		
	BM-310	BR-351	BR-651
cfft_q15	3.2	2.8	6.1
cfft_q31	1.5	1.3	2.8
biquad_cascade_q15	4.2	2.7	1.5
biquad_cascade_q31	3.8	3.2	1.7
dot_prod_q7	5.1	5.0	8.1
pid_q31	1.8	1.6	1.5
sincos_q31	3.9	4.2	1.4

Color Coding:

- **Less than the first quartile (Q1 = 1.6)**
- **Bigger than the third quartile (Q3 = 4.2)**

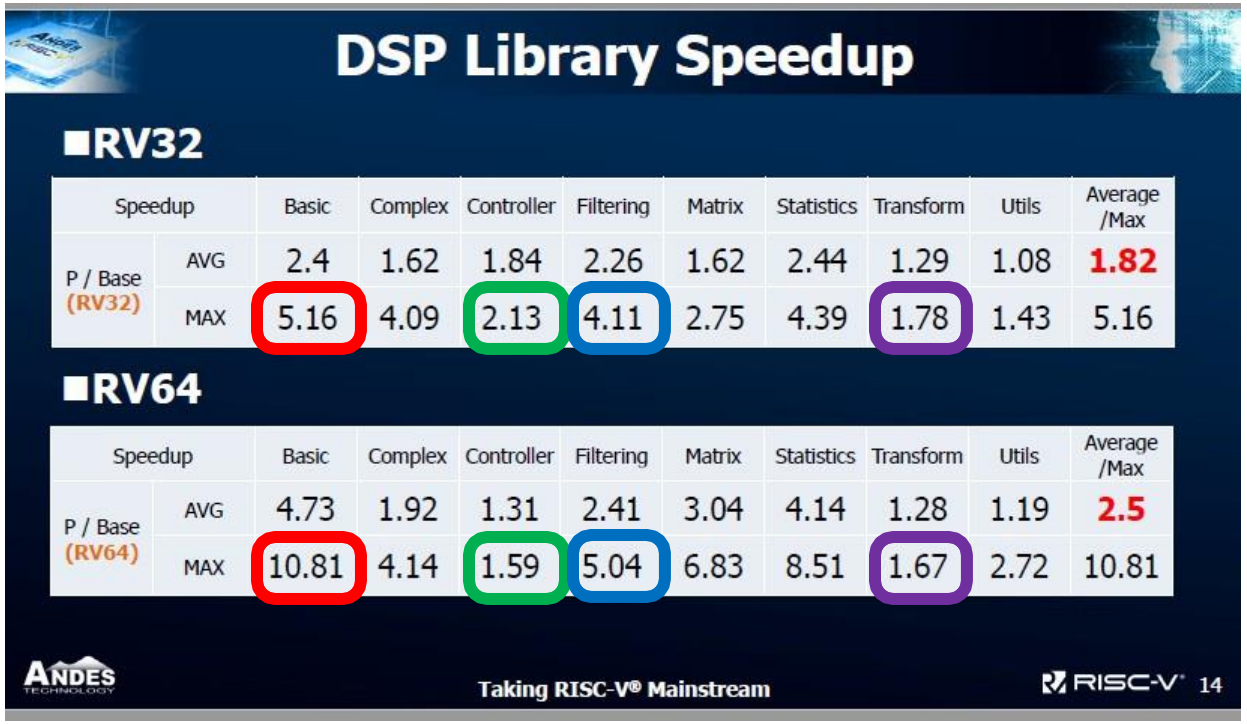
[1] A value X>1.0 means that reference code spends X times more core cycles than P Extension accelerated code.

Public Performance Data: Andes

The results were published[1] in 2019 for the AndeStar V3 cores during the P extension specification design process.

MAX Speedup

Group	BM-310	BR-351	BR-651
	RV32	RV32	RV64
BasicMath	6.33	6.67	8.88
Controller	6,24	4,72	3,46
Filtering	5,05	4,54	1,93
Transform	3,24	2,79	6,18

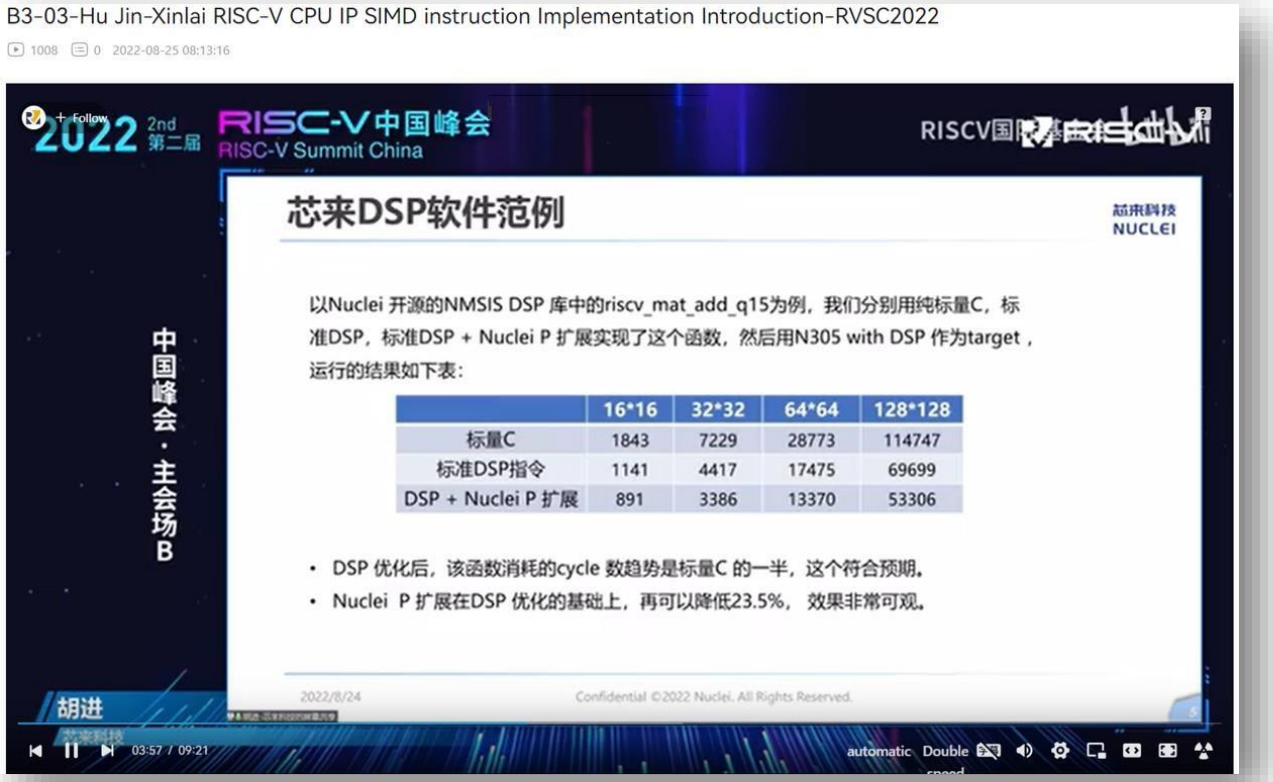


[1] *Status update of RISC-V P extension task group (Chuan-Hua Chang @ RISC-V Workshop Zurich 2019)*

Public Performance Data: Nuclei

- Comparable results were declared at RISC-V Summit China 2022 by Nuclei.
- The *riscv_mat_add_q15* function is almost identical to *riscv_add_q15*. For the latter, there is a P-extension accelerated version in BEAR-DSP.

Matrix Size	Basic Ops	Nuclei N305[1] [cycles]	BM-310 [cycles]	BR-351 [cycles]
16x16	256	891	791 (-11%)	702 (-21%)
32x32	1024	3386	2792 (-18%)	2209 (-34%)
64x64	4096	13370	10842 (-19%)	8286 (-38%)
128x128	16384	53306	43096 (-19%)	32839 (-38%)



[1] *Nuclei N305: 32-bit single issue core with 3-stage pipeline. It is correct to compare it with the BM-310.*

Public Performance Data: ARM

CloudBEAR

The performance of elementwise operations compared to Arm Cycle Models[1]

Eltwise Case [Type: size]	Cortex -M7 [cycles]	BR-351 [cycles (to M7)]		Cortex M55 [cycles]	BR-651 [cycles (to M55)]
Add Q31: 16	266	223 (-16%)		168	187 (+11%)
Add Q31: 256	3557	1156 (-68%)		1848	647 (-65%)
Mult Q31: 16	364	256 (-30%)		172	207 (+20%)
Mult Q31: 256	5052	1490 (-71%)		1912	760 (-60%)

The performance of a Complex FFT compared to STM32[2] MCUs

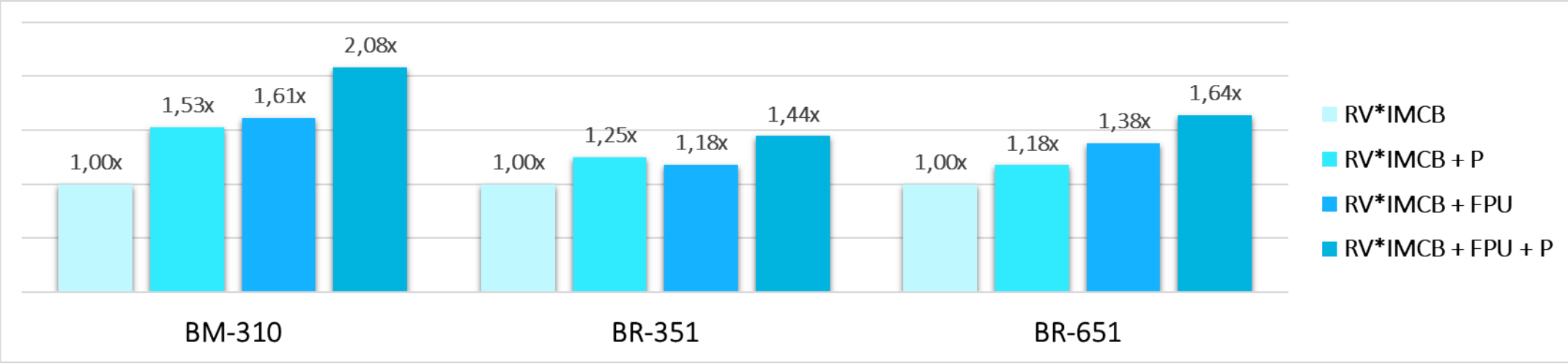
CFFT [Type: Size]	Cortex-M4 [cycles]	BM-310 [cycles (to M4)]		Cortex-M7 [cycles]	BR-351 [cycles (to M7)]	BR-651 [cycles (to M7)]
Q15: 64	3509	2119 (-40%)		2994	1694(-43%)	1134 (-62%)
Q15: 1024	77371	48117 (-38%)		56898	37318(-34%)	22022 (-61%)
Q31: 64	6007	3384 (-44%)		4537	2804 (-38%)	1694 (-63%)
Q31: 1024	144214	80230 (-44%)		93725	65836 (-30%)	37286 (-60%)

[1]Jason Andrews. How to use the Arm Cortex-M55 Processor with the open-source CMSIS library. ARM Community Blogs (https://community.arm.com) March 11, 2020

[2]AN4841: Digital signal processing for STM32 microcontrollers using CMSIS . Application note (https://www.st.com) Feb 23, 2018

Hardware Cost

The impact of P and F(D) extensions on the core area[1]



P and V extensions external comparison[2]

- Only the 128-bit vector register file costs at least 1.6 times more gates than the entire P extension. Usually, this ratio is closer to 3.2x
- *"...When you add all the computation/control circuits for vectors, you can easily find that **RVV** is usually **at least 10x bigger than P-extension**."*

[1] *FPU denotes a single-precision floating point unit (F Extension) for BM-310 and BR-351, and a double-precision floating point unit (FD Extension) for BR-651*

[2] *Dr. Kevin Chen in the "Ensuring the value of P" topic (https://lists.riscv.org/g/tech-p-ext/message/408) May 1, 2024*

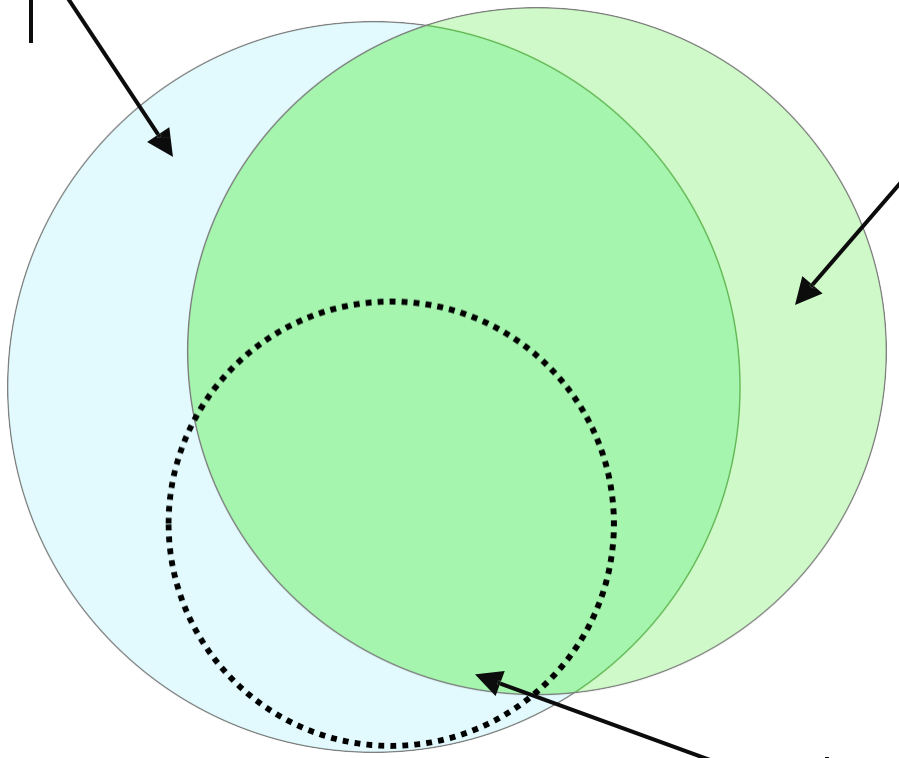
Prospect

What is in the future?

P Extension Instructions Overlap

CloudBEAR

~330 instructions are listed in **P v0.9.11**



~400 instructions for RV32 are mentioned in the latest proposal[1] on **Base P**.

~260 instructions for RV64 (opcodes overlap with RV32)

108 instructions are used in our use cases.

[1]*John Hauser. Proposed Instructions for the RISC-V Base P Extension-009. P Extension Task Group (http://www.jhauser.us/RISCV/ext-P/RVP-baseInstrs-009.pdf) August 5, 2024*

Upcoming Base P Extension

The main direction of the revision is to isolate a comprehensive "base P" subset and ratify it. If necessary, everything else can be added with follow-on Zp* extensions.

- Notable new feature: More instructions for operating with 64-bit data on the RV32 architecture
- Notable removed feature: Instructions for saturating MAC operations

Follow-on Zp* extensions under consideration

- Register-pair double word load/store for RV32
- Double-wide packed-SIMD multiply instructions
- Packed-SIMD extension for audio codecs (includes old saturating operations)

Our strategyfor upcoming release

- Keep P extension v0.9.11 in a product line and continue to support it also after release of Base P. Probably, as a custom extension.
- Analyze the final specification and community activity in parallel to make a decision on implementation update.

Conclusion

What is the key outcome?

Key Takeaways

- A compact ISA extension for **integer and fixed point math using general purpose registers** and (almost) without CSR
- Acceleration due to **fused operations applied to multiple data** by a single instruction.
- Performance **speedup** of the DSP library using P extension ranging **from +30% to +900%**.
- The **hardware cost** of the P extension's is comparable to FPU, and **up to 10 times lower than Vector** extension in terms of gates count.
- **Processors with P are already available on the market**, although the specification design is still ongoing.
- The performance of **CloudBEAR cores** with P is **better** than that of comparable solutions, according to open data.


CloudBEAR