




Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики



Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

РАЗРАБОТКА БИБЛИОТЕКИ ВЕКТОРНЫХ МАТЕМАТИЧЕСКИХ ФУНКЦИЙ ДЛЯ ПРОЦЕССОРОВ АРХИТЕКТУРЫ RISC-V



VIII митап Альянса RISC-V

Нижний Новгород, 14 ноября 2026

Волокитин В.Д., Мееров И.Б.

Кафедра высокопроизводительных вычислений и системного программирования

Высокопроизводительные вычисления

Численное моделирование, суперкомпьютинг, оптимизация вычислительного ПО под целевые архитектуры, разработка переносимого ПО, переносимость производительности

Системное программирование

Разработка базовых математических библиотек на C и C++.

Компьютерная графика, компьютерное зрение, ИИ

Инженерная графика, научная визуализация, обработка изображений (гиперспектральные изображения, ДЗЗ, приложения в биомедицине и с/х). Геоинформационные системы. Глубокое обучение и ИИ в приложениях. Бенчмаркинг и оптимизация инференса.

Квантовые вычисления

Разработка алгоритмов для квантовых компьютеров.

Параллельные алгоритмы на графах

Разработка параллельных алгоритмов. Приложения.



Эпоха экзафлопсных вычислений: некоторые открытые проблемы

- **Архитектурная проблема “Memory wall”**: разрыв между производительностью вычислений и стоимостью работы с памятью
- **Проблема приложений**: неэффективность использования дорогостоящих ресурсов в задачах с низкой арифметической интенсивностью и нерегулярным доступом к памяти (графы, разреженная линейная алгебра…)
- **Проблема переносимости**: гетерогенность; решается проблема переносимости кода, но не производительности
- **Проблема отказоустойчивости и надежности**
- **Проблема взаимодействия** с новыми архитектурными и программными парадигмами (квантово-классические вычисления, вычисления в смешанной точности, методы ИИ)

Место RISC-V в будущем мире HPC

Открытая архитектура — гибкость и расширяемость:

- RISC-V как основа сопряженной разработки ПО и архитектуры
- Гибкая векторизация RVV
- Разработка новых расширений системы команд
- Возможность создания гетерогенных узлов из устройств с единой системой команд

Текущие проблемы экосистемы:

- Нехватка устройств серверного типа
- Нехватка аппаратных датчиков и соответствующего системного ПО
- Недостаточная зрелость стека программного обеспечения

Чтобы раскрыть возможности устройства, необходимо оптимизированное системное и прикладное ПО!

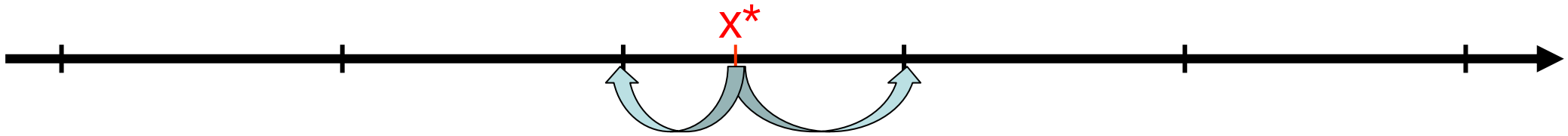
Стандарты математических функций

Стандарт IEEE-754

- Международный стандарт, описывающий формат представления чисел с плавающей запятой
- Используется повсеместно: ПО, компиляторы, аппаратная реализация
- В редакции **IEEE 754-2008** появились рекомендации для математических функций
- **Требование к точности:** ≤ 0.5 **ulp** (корректное округление)

Стандарт ISO/IEC 10967-2

- Международный стандарт, определяющий языконезависимые спецификации для элементарных численных функций
- Направлен на подробное описание поведения элементарных функций
- Разделяет функции на базовые и дополнительные
- **Требования к точности:** базовые до 0.75 **ulp**, дополнительные до 1.5 **ulp**



Обзор библиотек

- **CRLibm** – библиотека, обеспечивающая корректное округление во всех точках функции

Преимущества:

- Математическая гарантия точности (0.5 ulp)
- Поддерживается автоматическая векторизация
- Есть векторная реализация (x86 - SSE)

Недостатки:

- Из-за проверок точности и алгоритмов уточнения работает **в 2-10 раз дольше** высокопроизводительных библиотек
- В исходном виде проект больше не поддерживается, его развитие продолжается в виде множества разрозненных форков/библиотек

Обзор библиотек

- **LibM** – стандартная библиотека, которая поставляется вместе с компилятором

Преимущества:

- Общепринятая и кроссплатформенная
- В компиляторе может быть специализированная векторная версия (для RISC-V пока нет общепринятой версии)
- Известны максимальные ошибки (документация Glibc) и независимые исследования (Gladman B. et al., 2025)

Недостатки:

- Скалярные алгоритмы **плохо поддаются** автоматической **векторизации**

Таблица ошибок Glibc 2.41 (ULP)

	RISC-V 64-bit
expf	1
exp	1
expl	1
expf128	-
exp10f	-
exp10	2
exp10l	2
exp10f128	-

Обзор библиотек

- **veclibm** — векторная библиотека, написанная специально для расширения RISC-V Vector Extension (RVV)

Преимущества:

- Высокопроизводительная реализация
- Реализована с использованием **RVV Intrinsics**
- Достаточно точная (заявляется $< 1\ ulp$, хотя мы находили функции $> 1\ ulp$)

Недостатки:

- Реализовано **только для double**
- Использует интерфейс для работы с массивами с оптимально подобранным параметром LMUL
- Тестировалась в эмуляторе QEMU, проект давно не обновлялся

Обзор библиотек

- **SLEEF** (SIMD Library for Evaluating Elementary Functions) — кроссплатформенная векторная библиотека математических функций

Преимущества:

- Универсальная кроссплатформенная библиотека
- Реализована на обобщенных псевдоинтринсиках, транслируемых под целевую SIMD-архитектуру

Недостатки:

- Для RVV реализована только при LMUL = 1, 2
- Реальная точность вычислений иногда **не соответствует заявленной**

Библиотека RVVMF

❑ **RVVMF** – векторная библиотека математических функций под RVV

Цель:

- Высокопроизводительная библиотека с использованием RVV
- Точность удовлетворяет требованиям ISO/IEC 10967-2 (**0.501 ulp** статистически)

Тестовая система:

- Проверка точности (FP64, FP32, FP16)
- Проверка флагов FE_EXCEPTION и монотонности функции
- Оценка производительности (задержка и пропускная способность)

Что уже готово – широкий набор функций (FP64, FP32, FP16):

`exp, exp2, expm1, log, log2, log10, log1p, sin, cos, tan, asin, acos, atan, atan2, sinh, cosh, tanh, sqrt, pow, cbrt, hypot, trunc, floor, ceil, round`

Планы на будущее:

- Расширение списка функций и оптимизация
- Поддержка вычислений со сниженной точностью **3.5 ulp**

*Проект библиотеки RVVMF поддержан ООО «КНС ГРУПП»

Основные принципы реализации

Редукция аргумента

- Математическая редукция
- Предвычисленные таблицы (LUT)

- Уменьшение области определения
- Баланс нагрузки на память и АЛУ

Аппроксимация

- Минимаксный полином на малом отрезке

- Минимизация максимальной ошибки

Восстановление ответа

- Точная сборка ответа с компенсацией потери точности

Проблема потери точности вычислений

Представление числа или константы в виде суммы двух чисел соответствующей точности (**double-double арифметика**):

$$c = a * b; \Rightarrow c_h = a * b;$$
$$c_l = fma(a, b, -c_h);$$

Особенности

Корректная работа с флагами исключений FE_EXCEPTION:

- **`FE_INEXACT`**: контроль ослаблен, так как стандарт допускает установку флага при большинстве промежуточных операций
- **`FE_DIVBYZERO`**: возникает только в специфических функциях и особых точках (например, $pow(0.0, -2)$)
- **`FE_INVALID`, `FE_UNDERFLOW` и `FE_OVERFLOW`**: строгий контроль во всех особых точках и граничных областях

Обеспечение высокой производительности:

- Сложности автоматического профилирования кода на целевой архитектуре RISC-V ⇒ **Решено анализом конвейера и ручной оптимизацией инструкций**

Результаты точности. Статистика

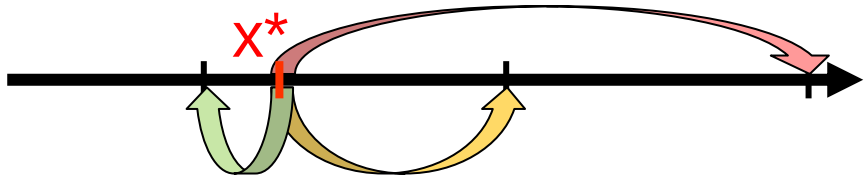
	exp	exp2	expm1	log	log2	log10	log1p	sin	cos	tan	asin	acos	atan
RVVMF FP64	**0.5000**	**0.5000**	**0.5000**	**0.5000**	**0.5000**	**0.5000**	**0.5000**	0.5009	0.5009	**0.5007**	**0.5000**	**0.5000**	**0.5000**
SLEEF FP64	0.5004	0.5005	0.5002	0.5002	0.5001	0.5001	0.5006	0.5087	0.5096	0.5060	0.5005	0.5001	0.5003
veclibm FP64	0.5001	0.5001	2.0	**0.5000**	**0.5000**	**0.5000**	0.5001	0.5091	0.5090	0.5028	0.5004	0.5001	2.0
libm FP64	**0.5000**	**0.5000**	0.5005	**0.5000**	**0.5000**	2.0	0.5006	**0.5007**	**0.5007**	2.0	**0.5000**	**0.5000**	**0.5000**
RVVMF FP32	**0.5000**	**0.5000**	**0.5000**	**0.5000**	**0.5000**	**0.5000**	0.5001	**0.5006**	**0.5006**	**0.5001**	0.5009	**0.5000**	0.5005
SLEEF FP32	0.5040	0.5025	0.5016	0.5004	0.5003	0.5003	2.0	0.5262	0.5301	0.5636	0.5017	0.5003	0.5010
libm FP32	**0.5000**	**0.5000**	**0.5000**	0.5002	0.5002	**0.5000**	**0.5000**	0.5068	0.5066	0.5185	**0.5000**	**0.5000**	**0.5000**
RVVMF FP16	0.5001	0.5001	0.5006	0.5003	0.5001	0.5003	0.5008	0.5008	0.5007	0.5009	0.5007	0.5006	0.5004

	atan2	hypot	sinh	cosh	tanh	sqrt	pow	cbrt
RVVMF FP64	**0.5000**	0.5034	0.5002	**0.5000**	**0.5002**	**0.5000**	**0.5000**	**0.5000**
SLEEF FP64	0.5003	**0.5000**	**0.5001***	**0.5000***	2.0	**0.5000**	0.5001	0.5003
veclibm FP64	2.0	-	**0.5001**	0.5001	**0.5002**	-	0.5001	0.5026
libm FP64	**0.5000**	0.5034	2.0	0.5006	2.0	**0.5000**	**0.5000**	2.0
RVVMF FP32	**0.5000**	0.5129	0.5010	0.5002	0.5003	**0.5000**	0.5002	**0.5000**
SLEEF FP32	0.5013	0.5001	2.0*	0.5001*	2.0	**0.5000**	0.5007	**0.5000**
libm FP32	**0.5000**	**0.5000**	**0.5000**	**0.5000**	**0.5000**	**0.5000**	**0.5000**	**0.5000**
RVVMF FP16	0.5006	0.5470	0.5001	0.5000	0.5007	0.5000	0.5004	0.5000

соответствует **0.501 ulp**

не соответствует **0.501 ulp**

ошибка >**1.0 ulp**



Результаты точности. Максимальная ошибка

	exp	Exp2	expm1	log	log2	log10	log1p	sin	cos	tan	asin	acos	atan
RVVMF FP64	**0.5000**	**0.5010**	**0.5010**	**0.5000**	**0.5000**	**0.5000**	**0.5001**	0.5972	0.6029	**0.5241**	**0.5004**	**0.5000**	**0.5026**
SLEEF FP64	0.7897	0.7823	0.7499	0.6974	0.7079	0.7234	0.9999	0.7599	0.7518	0.9999	0.7178	0.6663	0.7989
veclibm FP64	0.5370	0.5400	1.0570	0.5098	0.5313	0.5057	0.6205	0.7569	0.7426	0.6274	0.6346	0.6300	4+
libm FP64	0.5022	0.5031	0.7564	0.5090	0.5017	1.5509	0.7478	**0.5153**	**0.5145**	0.5559	0.5052	0.5034	0.5118
RVVMF FP32	**0.5001**	**0.5003**	0.5071	**0.5168**	**0.5016**	0.5185	0.5278	**0.5165**	**0.5164**	**0.5016**	0.8289	**0.5000**	0.5992
SLEEF FP32	0.9488	0.8301	0.7497	0.5840	0.5778	0.6241	1.0001	0.8060	0.8425	0.9250	0.6807	0.6117	0.8488
libm FP32	0.5016	0.5016	**0.5000**	0.8035	0.7092	**0.5000**	**0.5000**	0.5604	0.5605	1.2698	**0.5000**	**0.5000**	**0.5000**
RVVMF FP16	0.5002	0.5011	0.5144	0.5083	0.5005	0.5322	0.5086	0.5028	0.6487	0.6412	0.5049	0.5037	0.5024

	atan2	hypot	sinh	cosh	tanh	sqrt	pow	cbrt
RVVMF FP64	**0.5000**	0.8297	0.9999	0.5042	**0.5609**	**0.5000**	0.5052	**0.5000**
SLEEF FP64	0.9149	**0.5052**	0.9999*	**0.5005***	1.0001	**0.5000**	0.5997	0.5414
veclibm FP64	4+	-	**0.6191**	0.5617	0.7373	-	0.7151	0.5111
libm FP64	0.5094	0.8441	1.5613	0.9836	2.0454	**0.5000**	**0.5018**	3.3612
RVVMF FP32	0.5089	0.8849	0.9999	0.5316	0.5262	**0.5000**	0.7099	**0.5000**
SLEEF FP32	0.9935	0.9420	1.0001*	0.5024*	1.0001	**0.5000**	0.7396	**0.5000**
libm FP32	**0.5000**	**0.5000**	**0.5000**	**0.5000**	**0.5000**	**0.5000**	**0.5015**	**0.5000**
RVVMF FP16	0.7499	0.9350	0.5012	0.5000	0.5155	0.4999	0.7504	0.5000

соответствует **ISO/IEC 10967-2**

не соответствует **ISO/IEC 10967-2**

* проблема с областью определения функции

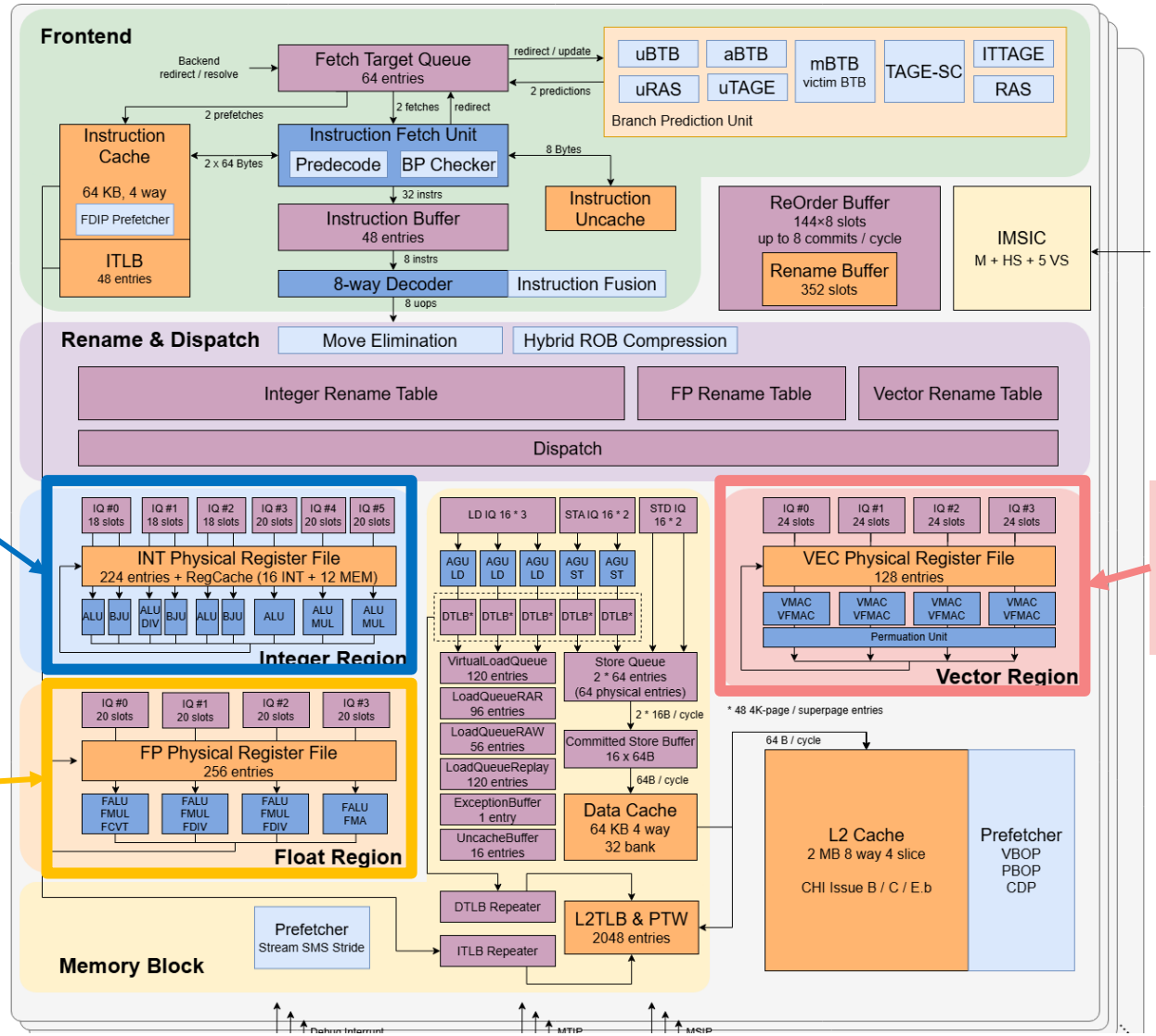
XiangShan

- Масштабный открытый (open-source) проект высокопроизводительного процессора с архитектурой RISC-V, разрабатываемый с использованием языка описания аппаратуры **Chisel**

Архитектура kunminghu-v3 – 3-я итерация 3-го поколения:

- Внеочередное исполнение инструкций (**Out-of-Order**)
- Заявлен набор инструкций RV64GCBSUHV, соответствие профилю **RVA23**
- Оценочная производительность: **~16 баллов/ГГц** на SPEC CPU 2006
- Проект активно развивается (в работе использовалась ревизия от начала 2026 года)

XiangShan



Целочисленный блок
6 очередей выдачи команд

Блок с плавающей запятой
4 исполнительных блока FALU/FMUL/FMA

Векторный блок RVV
4 исполнительных блока VMAC / VFMAC

XiangShan. Счетчики

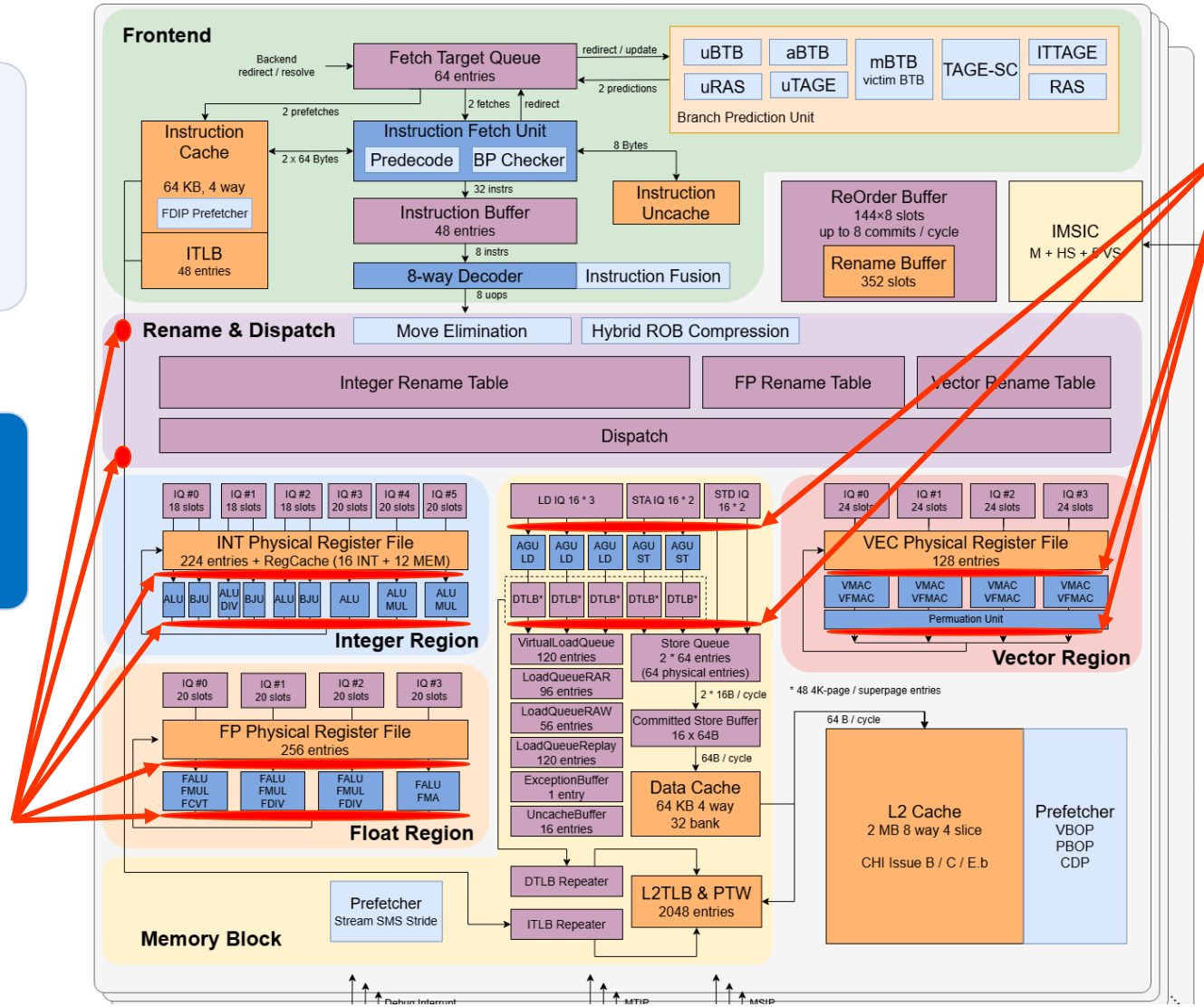
Есть открытые исходники RTL

Доступ к внутренним сигналам ядра на языке Chisel

Спрофилируем конвейер!

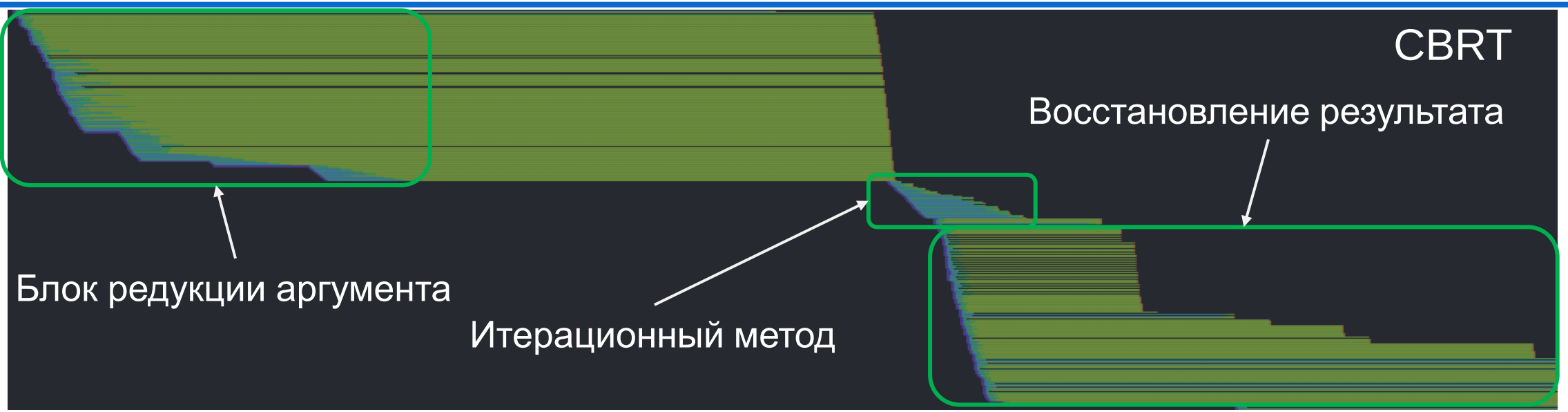
Потактовая визуализация прохождения инструкций

Счетчики ChiselDB



Счетчики ChiselDB

XiangShan. Пример профилирования



Видно полное исполнение функции на конвейере

Позволяет локализовать задержки и сократить критический путь

Примеры адаптаций:

- vloxei → vmerge (при малом размере таблиц)
- альтернативные последовательности инструкций (полиномы)

Выявленная особенность XiangShan:
В блоке RVV инструкции, работающие с одним и тем же регистром, блокируются даже при зависимости чтения после чтения (RAR).

Аппаратно-программное обеспечение

Аппаратное обеспечение – Banana Pi BPI-SM10:

Процессор SpacemiT K3:

- Кластер **8x ядер X100** (Out-of-Order RV64GCV, RVA23, RVV 1.0, **VLEN = 256 бит**)
- Кластер **8x ядер A100** (In-Order RV64GCV, RVA22, RVV 1.0, **VLEN = 1024 бит**)

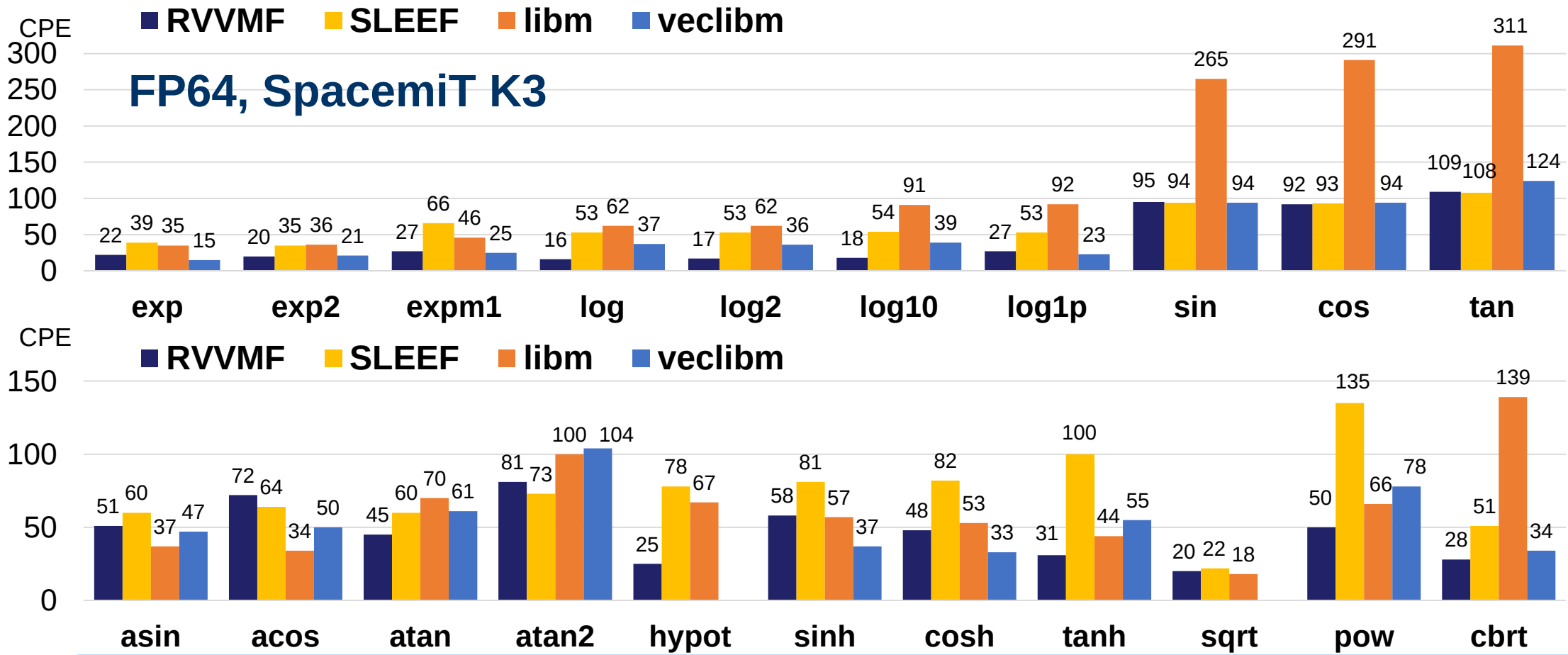
Оперативная память: 8 ГБ LPDDR5



Программное обеспечение:

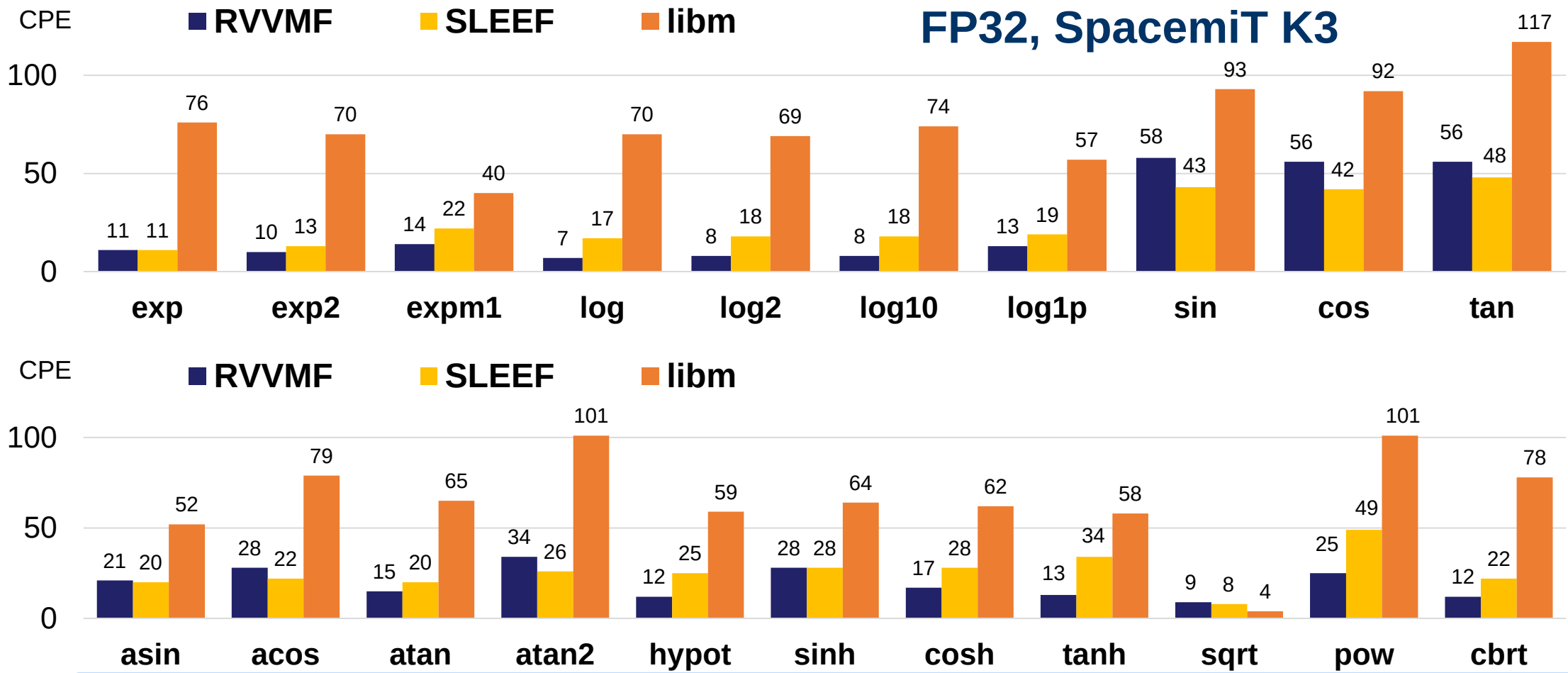
Компилятор: Clang 21 (кросс-компиляция)
Операционная система: Bianbu 4.0.1

Производительность K3 (X100). FP64



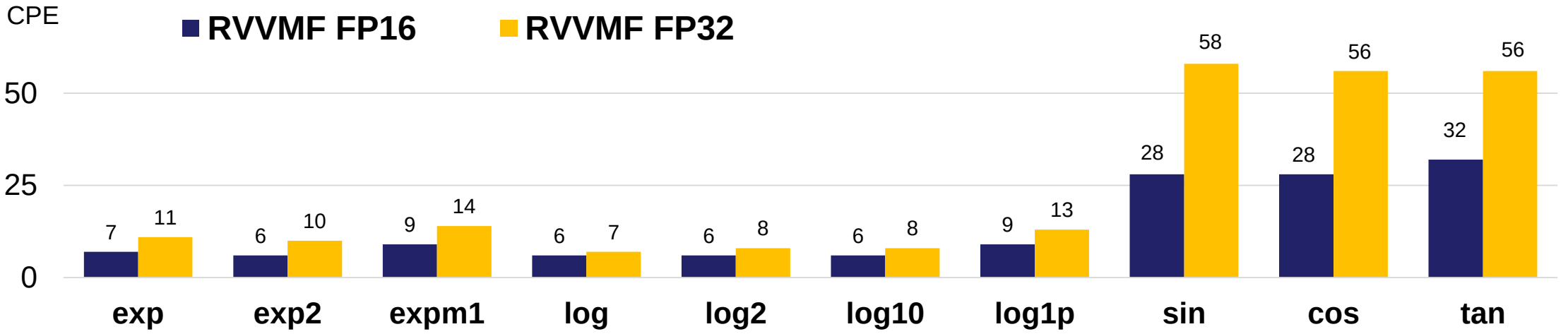
RVVMF в двойной точности по производительности превосходит или выступает на уровне аналогичных библиотек

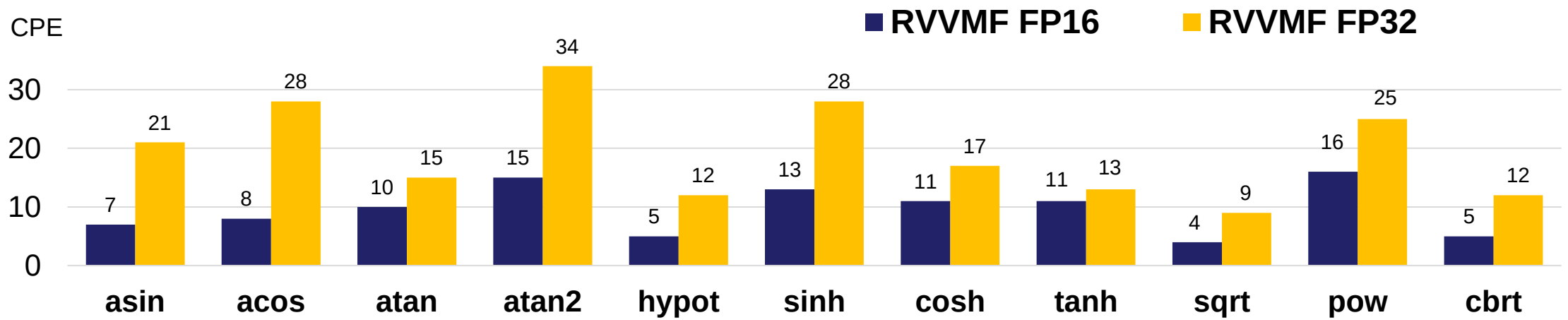
Производительность K3 (X100). FP32



RVVMF в одинарной точности во многих случаях значительно превосходит аналогичные библиотеки по производительности

Производительность K3 (X100). FP16





RVVMF в половинной точности демонстрирует значительно лучшую производительность, чем в одинарной точности (ожидаемый эффект)

Заключение

- Библиотека RVVMF – проект в открытых кодах, доступна по ссылке:

  https://github.com/rvvpl/rvvmf

- Разработана библиотека векторных математических функций с собственной системой тестирования точности и производительности
- Производительность на доступных устройствах на уровне лучших аналогов или опережает их **до 2х раз**
- Сделан шаг в сторону развития стека ПО для будущих приложений на RISC-V



Контакты

Нижегородский государственный университет

http://www.unn.ru

Институт информационных технологий, математики и механики

http://www.itmm.unn.ru

Волокитин Валентин Дмитриевич

volokitin@itmm.unn.ru

Мееров Иосиф Борисович

meerov@vmk.unn.ru

