



ОБЗОР ВОЗМОЖНОСТЕЙ ОТКРЫТОЙ МИКРОПРОЦЕССОРНОЙ АРХИТЕКТУРЫ RISC-V

В. А. Черняков, Е. А. Суворова

Санкт-Петербургский государственный университет аэрокосмического приборостроения

Наиболее распространенными архитектурами микропроцессоров в современном мире являются x86-x64 и ARM, однако в последнее время набирает популярность новая открытая архитектура RISC-V. В данной статье рассмотрена архитектура RISC-V. А также описаны преимущества данной архитектуры и причины ее популярности.

Ключевые слова: RISC-V, архитектура микропроцессора.

Для цитирования:

Черняков, В. А. Обзор возможностей открытой микропроцессорной архитектуры RISC-V / В. А. Черняков, Е. А. Суворова // Системный анализ и логистика. – 2023. – № 3(37). – с. 15 – 20. DOI: 10.31799/2077-5687-2023-3-15-20.

REVIEW OF FEATURES OF OPEN-SOURCE MICROPROCESSOR ARCHITECTURE RISC-V

V. A. Cherniakov, E. A. Suvorova

St. Petersburg State University of Aerospace Instrumentation

The most common microprocessor architectures nowadays are x86-x64 and ARM, but recently new open-source architecture called RISC-V has been gaining popularity. In this article RISC-V is reviewed. And the advantages of the architecture and reasons of its popularity are described.

Keywords: RISC-V, microprocessor architecture.

For citation:

Cherniakov, V. A. Review of features of open-source microprocessor architecture RISC-V / V. A. Cherniakov, E. A. Suvorova // System analysis and logistics. – 2023. – № 3(37). – p. 15 – 20. DOI: 10.31799/2077-5687-2023-3-15-20.

Введение

RISC-V – это открытый стандарт архитектуры с сокращенным набором инструкций (reduced instruction set computer). Изначально RISC-V проектировался для использования в образовании и исследованиях компьютерных архитектур. Появился данный стандарт в Беркли в 2010-х годах, сегодня же он распространился по всему миру, создаются организации, занимающиеся развитием RISC-V в отдельных странах. Примером тому является организация «Альянс RISC-V» в России. Целью организации является создание открытого сообщества разработчиков программного и аппаратного обеспечения, контролируемого участниками сообщества для дальнейшего развития архитектуры RISC-V в России [1]. В альянс уже вступили такие компании как «Лаборатория Касперского», «Yadro», «Астра», «Байкал электроникс», а также ВУЗы – СПбПУ, СПбГУ, «ЛЭТИ». В России существуют компании разрабатывающие различные вариации ядра на базе RISC-V, например, cloudBear. Все это свидетельствует о высоком интересе российской промышленности и науки к данной архитектуре и стандарту, как самому перспективному и эффективному в микроэлектронике.

Архитектура

Главными целями сообщества, развивающего архитектуру RISC-V, является создание свободной архитектуры, подходящей для реализации «на железе», избегающей чрезмерного усложнения для какого-то микроархитектурного стиля. Базовая архитектура определяется набором команд и регистров для работы с целочисленным типом данных. Данный набор инструкций (именуемый RV32I или RV64I, в зависимости от ширины регистров



проектируемой системы) является минимальным набором, достаточным для построения системы. К базовому набору можно добавлять различные «дополнительные» архитектуры. Таким образом получается следующий список базовых архитектур и дополнений:

Таблица 1 – Архитектуры и расширения RISC-V

Базовая архитектура или дополнение	Описание
RV32I	Базовый 32-битный целочисленный набор инструкций с 32 регистрами.
RV64I	Базовый 64-битный целочисленный набор инструкций.
RV128I	Базовый 128-битный целочисленный набор инструкций.
RV32E	Базовый 32-битный набор инструкций с 16 регистрами для встроенных систем.
M	Расширения с инструкциями умножения и деления.
A	Расширение с атомарными инструкциями.
F	Расширение с инструкциями для чисел с плавающей точкой одинарной точности.
D	Расширение с инструкциями для чисел с плавающей точкой двойной точности.
Q	Расширение с инструкциями для чисел с плавающей точкой четырехкратной точности.
L	Расширение с инструкциями, поддерживающими дробные значения по стандартам IEEE.
C	Расширение с инструкциями для 16-битных чисел и операций над ними.
V	Будущее расширения для векторизации.
B	Будущее расширение с инструкциями для работы с битовыми полями.
T	Будущее расширение с инструкциями для работы с транзакционной памятью.
P	Будущее расширение для packed SIMD
G	Объединение расширений IMAFD, Zicsr, Zifencei. RV32G и RV64G часто называют базовыми архитектурами общего назначения.

Данный подход к выстраиванию архитектуры позволяет избежать «нагромождения» инструкций и использовать процессор и ресурсы максимально эффективно и быстро.

Основные составляющие архитектуры RISC-V – это программный счетчик, регистры АЛУ и внешняя память.

Во всех базовых архитектурах, кроме RV32E, 32 регистра (рисунок 1). Ширина регистра вынесена в название базовой архитектуры, возможные варианты: 32 бит, 64 бит, 128 бит. В спецификации ширина регистра обычно обозначается XLEN. Далее в статье будем рассматривать базовую архитектуру RV32I, как основную и минимально достаточную для работы.

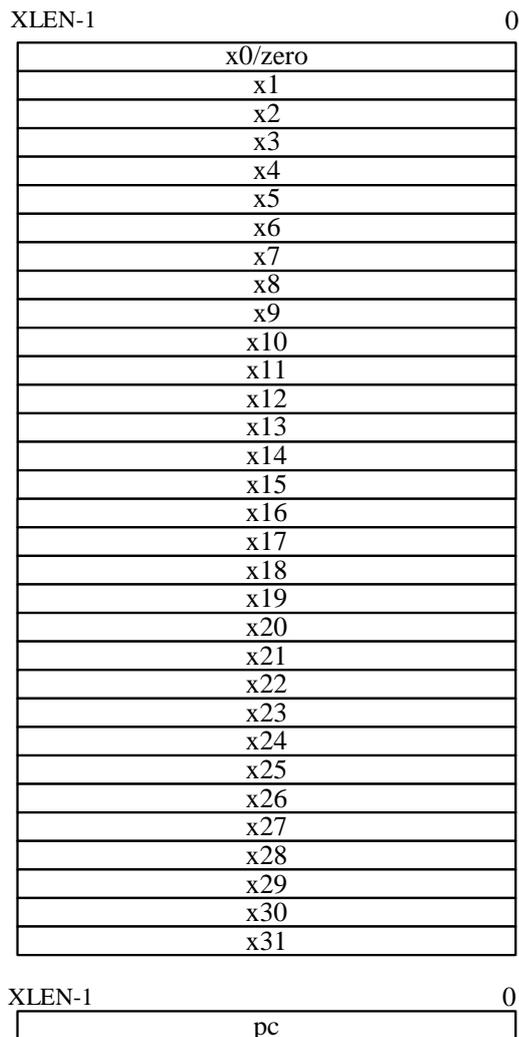


Рис. 1. Регистры в стандартной архитектуре. XLEN - ширина регистра

Формат базовых инструкций

В базовой архитектуре RV32I существует четыре формата инструкций – R, I, S и U. Все они фиксированной длины – 32 бита и должны быть выравнены по 4-байтовой границе в памяти [2]. В противном случае будет вызвано исключение или произойдет безусловный переход. В стандарте закреплено положение регистров-источников и регистров назначения, что заметно на рисунке 2. Исключением являются инструкции с использованием непосредственных значений.

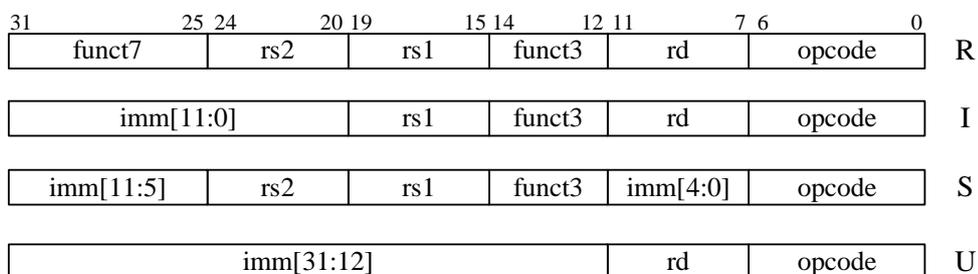


Рис. 2. Структура базовых инструкций. Некоторые значения imm распределены в разные "ячейки" инструкции, поэтому в квадратных скобках указывается порядок бит значения imm



Существует еще два формата инструкций, предназначенных для работы с непосредственными значениями. Это В и J, представленные на рисунке 3.

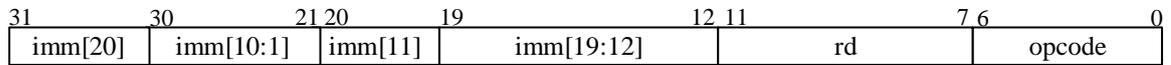
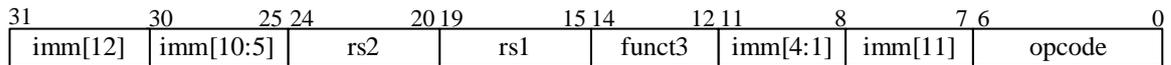


Рис. 3. В и J типы инструкций с непосредственными значениями

В инструкциях выделены 7 бит для opcode (машинной команды). Существует карта машинных команд. Основная карта машинных команд указана в таблице 2.

Таблица 2 – Стандартная карта машинных кодов

inst[4:2]	000	001	010	011	100	101	110	111
inst[6:5]								
00	LOAD	LOAD-FP	custom-0	MISC-MEM	OP-IMM	AUIPC	OP-IMM-32	48b
01	STORE	STORE-FP	custom-1	AMO	OP	LUI	OP-32	64b
10	MADD	MSUB	NMSUB	NMADD	OP-FP	reserved	custom-2/rv-128	48b
11	BRANCH	JALR	reserved	JAL	SYSTEM	reserved	custom-3/rv128	≥80b

Коды в правой колонке зарезервированы для дальнейшего использования с инструкциями длины большей 32 бит. Custom-0 и Custom-1 зарезервированы для дополнительных пользовательских инструкций. Custom-2 и custom-3 зарезервированы для использования в архитектуре RV128, но в RV32G и RV64G используются для пользовательских инструкций.

Ассемблерные команды

Выделим из команд в архитектуре RISC-V 3 блока и рассмотрим их:

- Вычислительные команды;
- Команды загрузки/сохранения;
- Команды передачи управления.

Для всех команд характерно наличие трёх операндов. Это решение позволяет избежать постоянных перемещений значений между регистрами при арифметических действиях, а также позволяет использовать одно и то же значение в одном регистре несколько раз подряд без использования команд загрузки. В ассемблерных командах архитектуры x64 при сложении результат всегда сохраняется в одном из регистров-источников, что заставляет разработчиков и программистов постоянно использовать команду *mov*. Данное решение ускоряет работу кода и упрощает структуру компилятора.

Целочисленные вычислительные команды выглядят следующим образом: *<command> dst, sr1, src2* или *<command> dst, sr1, imm*. В первом варианте за командой идет регистр, в который будет сохранен результат, затем два регистра-источника. Во втором варианте последний операнд – это некоторое допустимое значение регистра, пример: *add x1, x2, 5*. Данные команды представляются в виде R и S инструкций соответственно. Ни одна целочисленная вычислительная инструкция не вызывает арифметических исключений [2].

RV32I – это архитектура, в которой только инструкции загрузки и сохранения имеют доступ к памяти, а арифметические инструкции могут использовать только регистры [2]. Команды загрузки/сохранения имеют формат: *<operation> dst, offset(base)*. Адрес



определяется парой «базовый адрес - сдвиг». Команды загрузки декодируются в инструкции типа I, а инструкции сохранения – в тип S. Существуют следующие варианты команд загрузки: lw, lh, lb, lhu, lbu – загрузка в регистр полного слова (32 бит), половины слова (16 бит с дополнением в соответствии со знаком), байта слова (8 бит, аналогично предыдущему), беззнаковая загрузка половины слова, байта слова соответственно. Для сохранения в память существуют команды: sw, sh, sb – сохранение слова, половины слова, байта слова соответственно. При загрузке половины слова или байта слова знаковый бит заполняет все биты левее загруженной части.

Команды контроля потока исполнения делятся на безусловные переходы и условные ветки исполнения. Формат условного ветвления – *<condition> src1, src2, label*, то есть условие, регистры-источники и метка, на которую должен быть совершен переход при удовлетворенном условии. Для безусловных переходов существует два формата – *jal dst, label* и *jalr dst, base(offset)*. Первая форма совершает переход к метке и сохраняет в *dst* значение PC+4 (значение программного счетчика, увеличенного на 4). Второй формат осуществляет переход к *rf[base] + offset*, сохраняя в *dst* значение PC+4. Команда *jalr* нужна для реализации вызова функций, так как команда *jal* из-за особенностей инструкции имеет ограничение на максимальную «дальность» цели. Это ограничение – 1МБ.

Заключение

В статье рассмотрена базовая архитектура RISC-V. Архитектура является модульной и открытой, что является большим преимуществом, так как позволяет проектировать процессорные ядра индивидуально, под конкретную задачу, не нагружая реализацию лишними инструкциями. Данный фактор в сочетании с базовой архитектурой RV32E, делают RISC-V лучшим выбором для создания микропроцессоров для встроенных систем и аппаратуры.

Структура инструкций и, следовательно, ассемблерных команд упрощает устройство компилятора, так как не требуется создавать дополнительные команды для обращения к памяти. Уменьшение операций обращения к памяти положительно сказывается на быстродействии программного кода.

Активное появление и развитие сообществ профессионалов, развивающих архитектуру, так же положительно сказывается на работоспособности и широте возможностей применения RISC-V, так как ведется работа по созданию расширений, происходит конкурентный отбор реализаций этих расширений.

Всё вышеизложенное делает RISC-V самой перспективной и потенциально самой привлекательной архитектурой для промышленности и исследовательского сообщества.

СПИСОК ЛИТЕРАТУРЫ

1. RISC-V: About RISC-V [Электронный ресурс]. – URL: <https://riscv.org/about/> (дата обращения 12.01.2023).
2. АЛЪЯНС RISC-V: Об альянсе [Электронный ресурс]. – URL: <https://riscv-alliance.ru/> (дата обращения: 12.01.2023).
3. Фролов В. А. Исследование технологии RISC-V / В. А. Фролов, В. А. Галактионов, В. В. Санжаров // Труды ИСП РАН – 2020 г. – том 32, вып. 2. – стр. 81-98. doi: 10.15514/ISPRAS-2020-32(2)-7
4. The RISC-V Instruction Set Manual, Volume I: User-Level ISA, Document Version 20191213 // Editors Andrew Waterman and Krste Asanovic' – RISC-V Foundation, December 2019. – 238 p.



ИНФОРМАЦИЯ ОБ АВТОРАХ

Черняков Владислав Александрович –

студент кафедры аэрокосмических компьютерных программных систем
Санкт-Петербургский государственный университет аэрокосмического приборостроения
190000, Санкт-Петербург, ул. Большая Морская, д. 67, лит. А
E-mail: chernyakov_v@vk.com

Суворова Елена Александровна –

Кандидат технических наук, доцент
Санкт-Петербургский государственный университет аэрокосмического приборостроения
190000, Санкт-Петербург, ул. Большая Морская, д. 67, лит. А
E-mail: wildcat15@yandex.ru

INFORMATION ABOUT THE AUTHORS

Cherniakov Vladislav Aleksandrovich –

student of the Departments of Aerospace Computer Software Systems
Saint-Petersburg State University of Aerospace Instrumentation
67, Bolshaya Morskaja str., Saint-Petersburg, 190000, Russia
E-mail: chernyakov_v@vk.com

Suvorova Elena Alexandrovna –

PhD. tech. Sciences, associate Professor
Saint-Petersburg State University of Aerospace Instrumentation
67, Bolshaya Morskaja str., Saint-Petersburg, 190000, Russia
E-mail: wildcat15@yandex.ru