

# Разработка RISC-V

## Оглавление

Разработка RISC-V .....	1
Обзор главы .....	2
Цели обучения .....	2
Руководство по архитектуре набора команд RISC-V .....	3
Определение архитектуры набора инструкций (ISA) .....	3
Чем отличается ISA RISC-V .....	3
Модель совместного развития .....	3
Создание и курирование открытых спецификаций .....	4
Жизненный цикл расширений RISC-V .....	5
Непривилегированная спецификация .....	7
Организация спецификаций .....	7
Внутри непривилегированной спецификации .....	8
Расширения базового ISA .....	8
Расширение стандарта "M" .....	9
Расширение стандарта "F" .....	9
Расширение стандарта "C" .....	9
Привилегированная спецификация .....	11
Обзор .....	11
ISA машинного уровня (M-Mode), версия 1.11 .....	11
Немаскируемые прерывания (NMI) .....	11
Атрибуты физической памяти (PMA) .....	12
Технические характеристики не-ISA .....	13

## **Обзор главы**

Эта глава посвящена техническим деталям архитектуры набора инструкций (ISA) RISC-V и ее отличиям от других ISA. Мы рассмотрим некоторые правила и нормы, регулирующие RISC-V International, а также то, как разрабатываются расширения и стандарты ISA.

## **Цели обучения**

К концу главы вы научитесь:

- Описывать процесс, использованный для разработки RISC-V ISA и расширений.
- Различать базовые ISA, расширения и стандарты RISC-V.
- Понимать основы спецификаций Unprivileged и Privileged.

# **Руководство по архитектуре набора команд RISC-V**

## **Определение архитектуры набора инструкций (ISA)**

Архитектура набора команд (ISA) – это абстрактная модель компьютера. Ее также называют архитектурой или архитектурой компьютера. Реализация ISA, например, центральный процессор (CPU), называется имплементацией. Некоторые ISA, о которых вы могли слышать, включают x86, ARM, MIPS, PowerPC или SPARC. Все эти ISA требуют лицензии для их имплементации. С другой стороны, RISC-V ISA предоставляется под лицензией с открытым исходным кодом, которая не требует платы за использование.

## **Чем отличается ISA RISC-V**

Наиболее заметным отличием RISC-V от других ISA является то, что RISC-V разрабатывается организацией-членом, в которую можно вступить совершенно бесплатно, и лицензирует свои ISA на основе разрешительных лицензий с открытым исходным кодом. Это означает, что любой может внести свой вклад в спецификации, и ни одна компания или группа компаний не может определять направление развития стандартов.

RISC-V International управляется Советом директоров. Совет состоит из членов, избранных для представления всех классов членства, чтобы обеспечить стратегический голос на всех уровнях. Кроме того, Технический руководящий комитет (TSC) обеспечивает руководство нашими техническими инициативами в определении долгосрочной стратегии, формировании тактических комитетов и рабочих групп, а также утверждении технических результатов для ратификации или выпуска.

## **Модель совместного развития**

Спецификация RISC-V начинает свою жизнь как только целевая группа утверждается Техническим руководящим комитетом (TSC). После того, как целевая группа получила утвержденный устав, она начинает публичную работу на GitHub, записывая свои документы в формате AsciiDoc. Эти репозитории на GitHub могут получать запросы на исправление только от членов RISC-V International, однако работа ведется публично и прозрачно. Для групп, решивших вести протоколы, протоколы заседаний целевых групп также публикуются в открытом доступе. Общество может свободно отправлять вопросы в репозиторий GitHub, чтобы дать раннюю

обратную связь по любой спецификации. Спецификации и стандарты, не относящиеся к ISA (например, трассировка процессора, архитектурные тесты, наложение программного обеспечения), разрабатываются аналогичным образом.

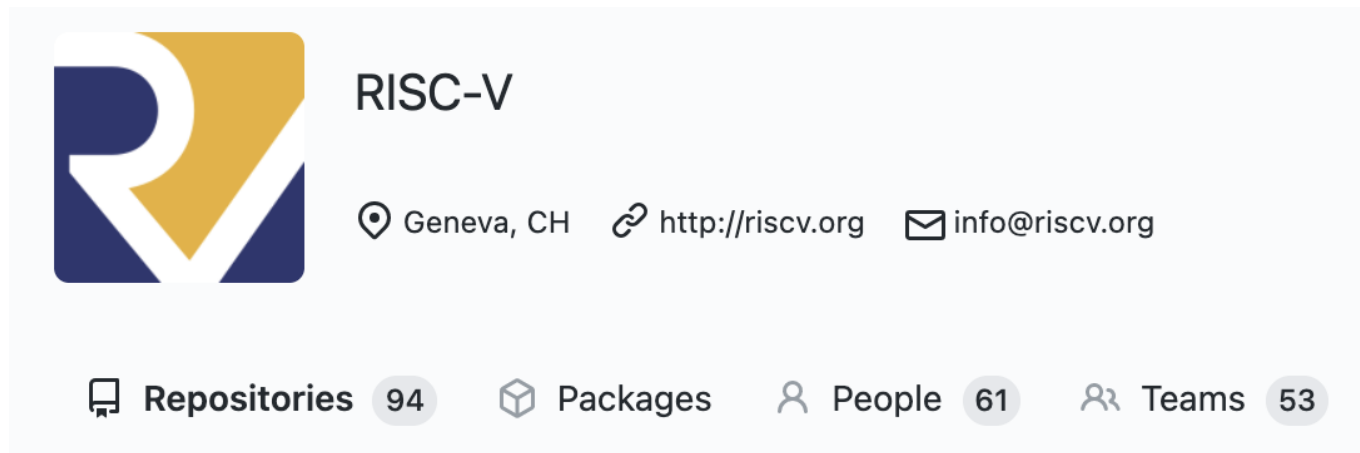


Рисунок 1. RISC-V на GitHub.

Спецификации RISC-V размещены на GitHub и находятся рядом с десятками программных проектов. Смотрите [список ратифицированных спецификаций](#)<sup>1</sup> и ссылки на их репозитории на GitHub.

### Создание и курирование открытых спецификаций

Процесс написания спецификаций обычно возглавляет архитектор аппаратного обеспечения в одной из организаций-членов RISC-V International. Они могут не писать фактический текст, но они выступают в качестве председателя целевой группы, контролирующей разработку спецификации. Для завершения разработки спецификации группе может потребоваться от нескольких месяцев до более года. О жизненном цикле расширения мы поговорим позже в этой главе.

То, что делает этот процесс разработки открытым, зависит от трех ключевых фактов:

1. Список рассылки целевой группы является общедоступным.
2. Документ спецификации находится в открытом доступе, и к нему можно оставлять комментарии.
3. Существует публичный список адресатов (рассылки), в котором любой желающий может отправить электронное письмо. ([isa-dev@groups.riscv.org](mailto:isa-dev@groups.riscv.org))

Используя эту методологию, даже не являющиеся членами организации, могут участвовать в разработке любой спецификации или стандарта, задавая вопросы, внося предложения или просто следя за ходом работы. Более того, в процессе ратификации существует 45-дневное окно, когда вся

---

<sup>1</sup> <https://riscv.org/technical/specifications/>

работа над спецификацией должна быть заморожена, а спецификация опубликована публично для ознакомления. В это время все желающие могут высказать свои замечания, и все вопросы будут решены до голосования за ратификацию.

Хотя стать членом RISC-V International - самый простой способ внести свой вклад в открытые спецификации, это не единственный способ. Любой может внести свой вклад, взаимодействуя с целевыми группами на публичных форумах, таких как список адресатов (рассылки) и GitHub.

### **Жизненный цикл расширений RISC-V**

Каждое расширение RISC-V проходит несколько этапов на пути к ратификации (рис. 2). В этом разделе мы кратко рассмотрим каждый этап, известный как "веха".

#### **1. План**

Целевая группа разрабатывает окончательный устав и устанавливает некоторые временные рамки.

#### **2. Разработка**

Группа выпускает несколько версий, которые считаются нестабильными.

#### **3. Заморозка**

Группа выпускает полный окончательный проект спецификации, в котором нет основных неизвестных и ожидаемых изменений (только исправление проблем, но без новых возможностей).

#### **4. Готовность к утверждению**

Проект спецификации рассылается для публичного рассмотрения, любые комментарии или вопросы общественности рассматриваются, и Технический руководящий комитет ставится в известность о необходимости голосования.

#### **5. Развитие экосистемы**

Расширение ратифицировано и поддерживается как часть RISC-V ISA. Сообществу еще предстоит решить множество задач. Например, спецификация Vector была ратифицирована, однако добавление автовекторизации в компиляторы является частью списка задач по развитию экосистемы этой спецификации.

# Specification Lifecycle

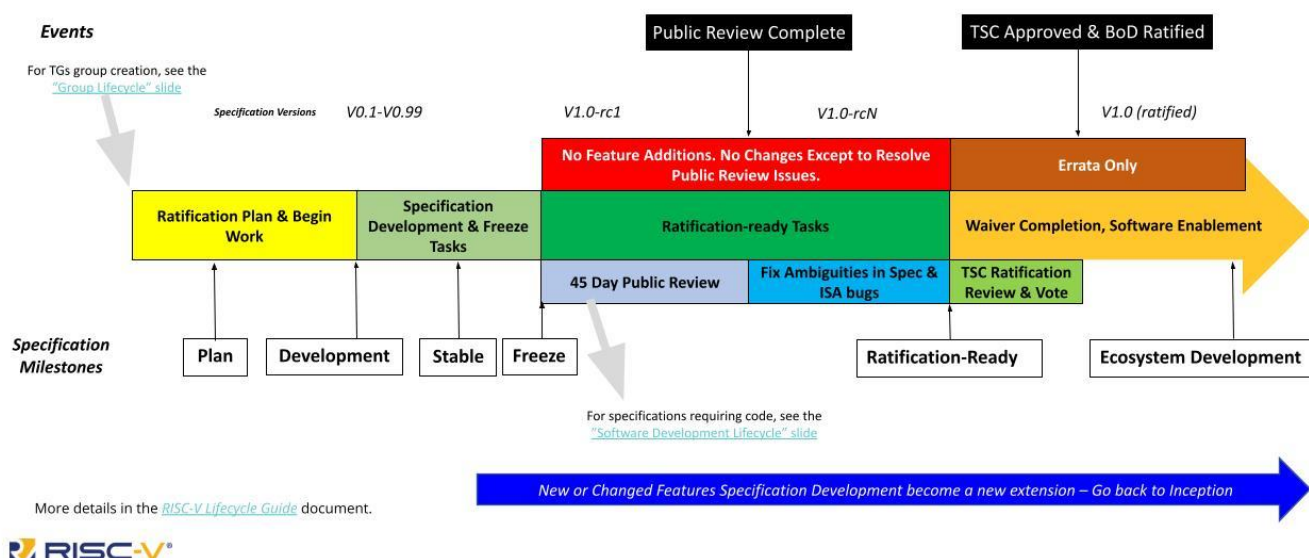


Рисунок 2. Жизненный цикл спецификации.

После ратификации расширения оно добавляется либо в непривилегированную, либо в привилегированную спецификацию. Иногда спецификация создается как часть отдельного документа, наиболее распространенным примером является спецификация отладки. Однако это редкий случай и обычно указывает на то, что расширение не является частью ISA, а скорее "стандартной" или "не-ISA спецификацией". Теперь мы рассмотрим непривилегированную и привилегированную спецификации более подробно.

# Непривилегированная спецификация

## Организация спецификаций

RISC-V ISA состоит из двух частей:

- Том 1, Непривилегированная спецификация
- Том 2, Привилегированная спецификация

Чтобы понять, почему спецификация разбита на две разные части, мы должны сначала немного разобраться в компьютерной архитектуре и безопасности. Исторически процессоры использовали иерархические домены защиты, часто называемые кольцами защиты, для защиты данных и кода от злоумышленников.

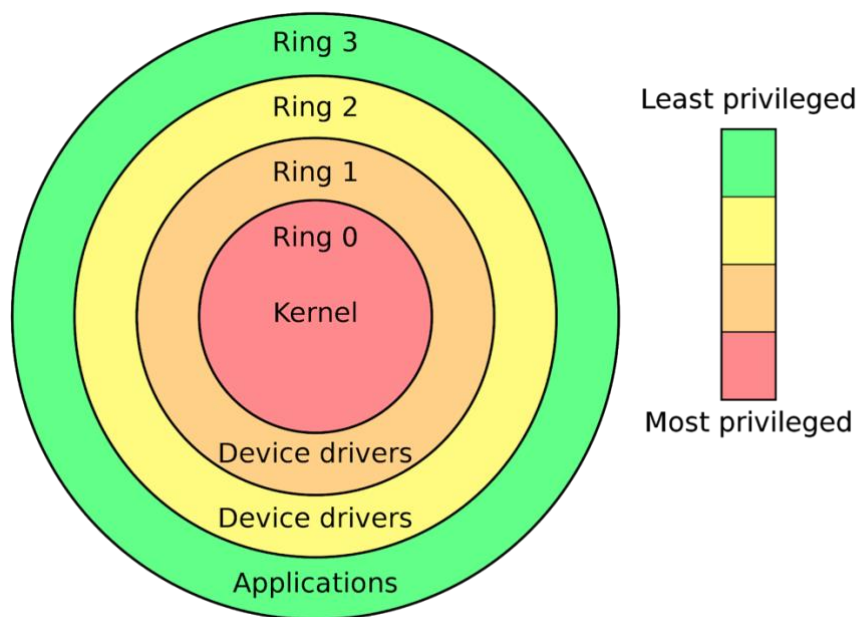


Рисунок 3. Герцшпрунг в английской Википедии, CC BY-SA 3.0.

Наиболее привилегированный код работает в "кольце 0" и имеет доступ ко всей системе. Процессор решает, какие привилегии предоставить исполняемому коду, основываясь на уровне привилегий. Например, доступ к памяти по физическому адресу может быть ограничен "кольцом 0", так что другие кольца должны ссылаться на виртуальное адресное пространство. Обычно процессор может одновременно работать только в одном из режимов привилегий, а для перехода между режимами существуют специальные инструкции. Все эти детали могут меняться от системы к системе, однако они должны соответствовать правилам, установленным в документах спецификации данной архитектуры.

В настоящее время RISC-V имеет три уровня привилегий: Режим пользователя (U-mode), режим супервизора (S-mode) и машинный режим (M-mode). Их можно представить как "кольцо 2",

"кольцо 1" и "кольцо 0" соответственно. Другие режимы, такие как режим гипервизора (H-mode), вероятно, будут добавлены в ближайшем будущем. Как и на рисунке выше, режим U предназначен для пользовательских процессов, режим S - для ядра и/или драйверов устройств, а режим M - для загрузчика и/или микропрограммы. Каждый уровень привилегий имеет доступ к определенным регистрам управления и состояния (CSR), и более высокие уровни привилегий могут получить доступ к CSR менее привилегированных уровней.

### **Внутри непривилегированной спецификации**

Проще говоря, непривилегированная спецификация описывает элементы, не относящиеся к машинному режиму (M-mode) или режиму супервизора (S-mode). Непривилегированная спецификация включает базовую ISA, а также расширения этой базы, такие как целочисленные (I), плавающие (F), двойные (D), сжатые инструкции (C) и многие другие.

Базовые наборы инструкций описывают формат инструкций, основные целочисленные инструкции, инструкции загрузки и хранения, а также другие фундаментальные детали ISA. Мы разбиваем их на несколько баз:

- RV32I - Целое число 32 бита
- RV32E - уменьшенный RV32I для встраиваемых систем
- RV64I - Целое число 64 бита
- RV128I - Целое число 128 бит

Все эти "базовые ISA" либо уменьшают, либо расширяют базовый набор инструкций RV32I. Например, RV64I расширяет целочисленные регистры и поддерживаемое адресное пространство пользователя до 64 бит. Это означает, что инструкции LOAD и STORE работают немного иначе, чем в RV32I, и непривилегированная спецификация содержит главу, объясняющую эти различия.

### **Расширения базового ISA**

Непривилегированная спецификация также содержит описания расширений этих базовых ISA. Опять же, любое расширение, которое не требует M-режима для работы, может быть описано в непривилегированной спецификации.

Каждое расширение базового ISA разрабатывается и поддерживается целевой группой:

- Crypto Task Group работает над криптографическими расширениями, которые могут перенести многие сложные криптографические алгоритмы в аппаратное обеспечение, повышая надежность и скорость.



- В Extension Task Group работает над расширениями для работы с битами, которые могут ускорить выполнение многих распространенных математических задач.
- Vector (V) Extension Task Group работает над векторными инструкциями, которые лежат в основе многих вычислений при обработке графики.

После ратификации эти расширения добавляются в непривилегированную спецификацию. Ниже перечислены некоторые из ратифицированных расширений, которые вы можете увидеть в процессоре RISC-V.

### **Расширение стандарта "M"**

Глава 7 непривилегированной спецификации<sup>2</sup> описывает, как должно выполняться умножение и деление целых чисел. В ней описывается поведение каждой из инструкций умножения (MUL, MULH, MULHU, MULHSU, MULW), какие регистры используются для множителя и множимого, и где хранится результат. То же самое делается и для деления, поскольку функционально можно рассматривать деление как просто обратное умножению. Вам может показаться странным, что это расширение не требуется. Однако для многих встроенных процессоров умножение можно выполнять программно, если оно требуется нечасто или вообще не требуется. Удаление этой логики из процессора позволит сэкономить деньги на разработке и снизить стоимость конечного продукта.

### **Расширение стандарта "F"**

В главе 11 описано, как мы добавляем вычислительные инструкции с плавающей точкой одинарной точности, соответствующие стандарту арифметики IEEE 754-2008. Существует множество ресурсов, описывающих детали арифметики с плавающей точкой в вычислениях. Достаточно понять, что эта глава описывает, как этот процесс реализован в RISC-V, и дополняется главой 12 (расширение D), которая описывает вычислительные инструкции двойной точности с плавающей точкой. Наконец, в главе 13 рассматривается расширение стандарта Q для 128-битных двоичных инструкций с плавающей точкой четверной точности. Все эти три расширения соответствуют стандартам IEEE. Опять же, многие встроенные приложения не требуют логики с плавающей точкой, и поэтому это расширение не является частью базовых ISA.

### **Расширение стандарта "C"**

В главе 16 описано расширение сжатого набора инструкций, которое уменьшает статический и динамический размер кода путем добавления коротких 16-битных кодировок инструкций для общих операций. Как правило, 50%-60% инструкций RISC-V в программе могут быть заменены инструкциями RVC, что приводит к уменьшению размера кода на 25%-30%. Расширение C

---

<sup>2</sup> <https://riscv.org/technical/specifications/>

совместимо со всеми другими стандартными расширениями инструкций. Расширение C позволяет свободно смешивать 16-битные инструкции с 32-битными, причем последние могут начинаться с любой 16-битной границы. Таким образом, при добавлении расширения C в любую систему ни одна инструкция не может вызвать исключения, связанные с выравниванием адресов инструкций.

Это охватывает большинство ратифицированных в настоящее время расширений в непривileгированной спецификации. Однако важно отметить, что многие расширения включены в спецификацию на стадии "черновика" или "заморозки". Как мы уже говорили в разделе "Жизненный цикл расширений RISC-V", эти спецификации еще не ратифицированы, и любая реализация должна избегать их использования в производстве.

# Привилегированная спецификация

## Обзор

Как следует из названия, привилегированная спецификация содержит описания RISC-V ISA, которые работают в машинном режиме (M-mode) или режиме супервизора (S-mode). Эти режимы имеют повышенные привилегии и поэтому описаны в совершенно отдельном документе от базового ISA и стандартных расширений. Эта спецификация также содержит дополнительные функциональные возможности, необходимые для запуска богатых операционных систем, таких как Linux.

В первой части каждой главы спецификации привилегий подробно описаны регистры управления и состояния (CSR), доступ к которым возможен только из M- и S-режима. Мы не будем рассматривать эти детали здесь, а сосредоточимся на других деталях, специфичных для этих двух режимов.

## ISA машинного уровня (M-Mode), версия 1.11

В этой главе описаны функции машинного уровня, доступные в машинном режиме (M-режиме). Режим M используется для низкоуровневого доступа к аппаратной платформе и является первым режимом, который включается при сбросе, когда процессор завершает инициализацию и готов к выполнению кода. M-режим также может использоваться для реализации функций, которые слишком сложно или дорого реализовать в аппаратном обеспечении напрямую. Хорошим примером может служить сторожевой таймер, реализованный в низкоуровневом программном обеспечении (микропрограмме), который помогает системе восстанавливаться после сбоев. Мы рассмотрим три важные особенности режима M, описанные в спецификации: немаскируемые прерывания, атрибуты физической памяти и защита физической памяти.

## Немаскируемые прерывания (NMI)

Немаскируемые прерывания (NMI) используются только в условиях аппаратных ошибок. При срабатывании они вызывают немедленный переход к обработчику NMI, работающему в M-режиме, независимо от того, как установлен бит разрешения прерывания для этого аппаратного потока. Другими словами, это прерывание будет обслуживаться без возможности заблокировать его в конфигурации. Каждый NMI будет иметь связанный с ним регистр "mcause". Это позволяет реализаторам решать, как они хотят обрабатывать эти прерывания, и позволяет им определять

множество возможных причин. NMI не сбрасывают состояние процессора, что позволяет диагностировать, сообщать и, возможно, локализовать аппаратную ошибку.

### **Атрибуты физической памяти (PMA)**

Физическая карта памяти системы включает в себя такие диапазоны адресов, как: области памяти, управляющие регистры с привязкой к памяти и пустые дыры в адресном пространстве. Некоторые области памяти могут не поддерживать чтение, запись или выполнение; некоторые могут не поддерживать доступ к подслову или подблоку; некоторые могут не поддерживать атомарные операции; а некоторые могут не поддерживать когерентность кэша или иметь различные модели памяти. В системах RISC-V эти свойства и возможности каждого региона физического адресного пространства машины называются атрибутами физической памяти (PMA).

PMA некоторых областей памяти фиксируются во время проектирования микросхемы - например, для ПЗУ на кристалле. Другие фиксируются во время проектирования платы, в зависимости, например, от того, какие другие микросхемы подключены к внешним шинам. Некоторые устройства могут быть конфигурируемыми во время выполнения для поддержки различных применений, которые подразумевают различные PMA - например, оперативная память на кристалле может быть частным кэшем одного ядра в одном конечном приложении, или доступ к ней как к общей некешируемой памяти в другом конечном приложении. В большинстве систем потребуется, чтобы по крайней мере некоторые PMA динамически проверялись аппаратно на более поздних этапах конвейера выполнения после того, как будет известен физический адрес, поскольку некоторые операции не будут поддерживаться по всем физическим адресам памяти, а некоторые операции требуют знания текущей настройки конфигурируемого атрибута PMA.

Для RISC-V мы выделяем спецификацию и проверку PMA в отдельную аппаратную структуру, "PMA checker". Во многих случаях атрибуты известны на этапе проектирования системы для каждой области физического адреса и могут быть жестко подключены к устройству проверки PMA. Если атрибуты настраиваются во время выполнения, могут быть предусмотрены специфические для платформы регистры управления с привязкой к памяти для задания этих атрибутов с гранулярностью, соответствующей каждой области на платформе (например, для статической памяти с произвольным доступом (SRAM) на кристалле, которая может быть гибко разделена между кешируемым и некешируемым использованием).

Детали PMA легко могли бы занять целую главу этого курса. Мы не будем рассматривать PMA с упорядочиванием памяти, PMA с идемпотентностью, PMA с когерентностью или PMA с

возможностью кэширования. Детали РМА подробно описаны в разделе 3.5 [Привилегированной спецификации](#)<sup>3</sup>. Опытные пользователи могут захотеть ознакомиться с этим разделом.

### **Технические характеристики не-ISA**

Целевые группы также могут работать над программным обеспечением или стандартами, которые не являются частью ISA. Например, следующие группы работают над проектами, которые не приводят к написанию спецификаций, а скорее стандартов, которые поощряют сообщества разрабатывать свои продукты на общей основе:

- Debug Task Group, работающая над поддержкой внешней отладки и стандартами.
- Целевая группа по соответствию работает над тестами и рамками соответствия RISC-V ISA.
- Целевая группа по структуре конфигурации работает над тем, как представить структуру конфигурации данной аппаратной реализации как в человекочитаемом формате, так и в двоичном формате.

---

<sup>3</sup> <https://riscv.org/technical/specifications/>