Лекция 14

**RISC-V** (читается, как «РИСК-Пять»[[1]](https://ru.wikipedia.org/wiki/RISC-V#cite_note-1)) — расширяемая открытая и свободная [система команд](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%BD%D0%B0%D0%B1%D0%BE%D1%80%D0%B0_%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4) и [процессорная архитектура](https://ru.wikipedia.org/w/index.php?title=%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80%D0%BD%D0%B0%D1%8F_%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0&action=edit&redlink=1) на основе концепции [RISC](https://ru.wikipedia.org/wiki/RISC)[[2]](https://ru.wikipedia.org/wiki/RISC-V#cite_note-faq-2), предназначенная для создания [процессоров](https://ru.wikipedia.org/wiki/%D0%9C%D0%B8%D0%BA%D1%80%D0%BE%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80)/[микроконтроллеров](https://ru.wikipedia.org/wiki/%D0%9C%D0%B8%D0%BA%D1%80%D0%BE%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D0%BB%D0%B5%D1%80) и разработки ПО. Спецификации архитектурных описаний RISC-V свободно доступны и бесплатны для любого использования, включая коммерческие реализации непосредственно в [кремнии](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D1%85%D0%B5%D0%BC%D0%B0) или для конфигурирования [ПЛИС](https://ru.wikipedia.org/wiki/%D0%9F%D0%9B%D0%98%D0%A1). Участие в проектировании и обсуждении спецификаций архитектурных описаний открытое. Система команд имеет зарезервированные в спецификации биты для кодирования расширений без ограничения области применения.

**Содержание**

* [1История](https://ru.wikipedia.org/wiki/RISC-V#%D0%98%D1%81%D1%82%D0%BE%D1%80%D0%B8%D1%8F)
* [2Базовая спецификация «RV32I»](https://ru.wikipedia.org/wiki/RISC-V#%D0%91%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D1%8F_%D1%81%D0%BF%D0%B5%D1%86%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F_%C2%ABRV32I%C2%BB)
* [3Система команд](https://ru.wikipedia.org/wiki/RISC-V#%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4)
	+ [3.1Список наборов команд](https://ru.wikipedia.org/wiki/RISC-V#%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%BD%D0%B0%D0%B1%D0%BE%D1%80%D0%BE%D0%B2_%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4)
	+ [3.2Форматы машинных команд](https://ru.wikipedia.org/wiki/RISC-V#%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D1%8B_%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D1%8B%D1%85_%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4)
	+ [3.3Регистры](https://ru.wikipedia.org/wiki/RISC-V#%D0%A0%D0%B5%D0%B3%D0%B8%D1%81%D1%82%D1%80%D1%8B)
	+ [3.4Вызовы подпрограмм, переходы и ветвления](https://ru.wikipedia.org/wiki/RISC-V#%D0%92%D1%8B%D0%B7%D0%BE%D0%B2%D1%8B_%D0%BF%D0%BE%D0%B4%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC,_%D0%BF%D0%B5%D1%80%D0%B5%D1%85%D0%BE%D0%B4%D1%8B_%D0%B8_%D0%B2%D0%B5%D1%82%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F)
	+ [3.5Арифметические и логические наборы команд](https://ru.wikipedia.org/wiki/RISC-V#%D0%90%D1%80%D0%B8%D1%84%D0%BC%D0%B5%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5_%D0%B8_%D0%BB%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5_%D0%BD%D0%B0%D0%B1%D0%BE%D1%80%D1%8B_%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4)
	+ [3.6Атомарные операции с памятью](https://ru.wikipedia.org/wiki/RISC-V#%D0%90%D1%82%D0%BE%D0%BC%D0%B0%D1%80%D0%BD%D1%8B%D0%B5_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%B8_%D1%81_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%8C%D1%8E)
	+ [3.7Сокращённые команды](https://ru.wikipedia.org/wiki/RISC-V#%D0%A1%D0%BE%D0%BA%D1%80%D0%B0%D1%89%D1%91%D0%BD%D0%BD%D1%8B%D0%B5_%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D1%8B)
	+ [3.8Команды для встраиваемых применений](https://ru.wikipedia.org/wiki/RISC-V#%D0%9A%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D1%8B_%D0%B4%D0%BB%D1%8F_%D0%B2%D1%81%D1%82%D1%80%D0%B0%D0%B8%D0%B2%D0%B0%D0%B5%D0%BC%D1%8B%D1%85_%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D0%BD%D0%B5%D0%BD%D0%B8%D0%B9)
	+ [3.9Привилегированные наборы команд](https://ru.wikipedia.org/wiki/RISC-V#%D0%9F%D1%80%D0%B8%D0%B2%D0%B8%D0%BB%D0%B5%D0%B3%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5_%D0%BD%D0%B0%D0%B1%D0%BE%D1%80%D1%8B_%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4)
	+ [3.10Битовые операции](https://ru.wikipedia.org/wiki/RISC-V#%D0%91%D0%B8%D1%82%D0%BE%D0%B2%D1%8B%D0%B5_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%B8)
	+ [3.11Компактный набор команд для SIMD](https://ru.wikipedia.org/wiki/RISC-V#%D0%9A%D0%BE%D0%BC%D0%BF%D0%B0%D0%BA%D1%82%D0%BD%D1%8B%D0%B9_%D0%BD%D0%B0%D0%B1%D0%BE%D1%80_%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4_%D0%B4%D0%BB%D1%8F_SIMD)
	+ [3.12Операции с векторами](https://ru.wikipedia.org/wiki/RISC-V#%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%B8_%D1%81_%D0%B2%D0%B5%D0%BA%D1%82%D0%BE%D1%80%D0%B0%D0%BC%D0%B8)
	+ [3.13Команды для отладки](https://ru.wikipedia.org/wiki/RISC-V#%D0%9A%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D1%8B_%D0%B4%D0%BB%D1%8F_%D0%BE%D1%82%D0%BB%D0%B0%D0%B4%D0%BA%D0%B8)
* [4Реализации](https://ru.wikipedia.org/wiki/RISC-V#%D0%A0%D0%B5%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%B8)
	+ [4.1IP-ядра](https://ru.wikipedia.org/wiki/RISC-V#IP-%D1%8F%D0%B4%D1%80%D0%B0)
	+ [4.2Процессоры и микроконтроллеры](https://ru.wikipedia.org/wiki/RISC-V#%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80%D1%8B_%D0%B8_%D0%BC%D0%B8%D0%BA%D1%80%D0%BE%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D0%BB%D0%B5%D1%80%D1%8B)
		- [4.2.1Микропроцессоры](https://ru.wikipedia.org/wiki/RISC-V#%D0%9C%D0%B8%D0%BA%D1%80%D0%BE%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80%D1%8B)
		- [4.2.2Микроконтроллеры](https://ru.wikipedia.org/wiki/RISC-V#%D0%9C%D0%B8%D0%BA%D1%80%D0%BE%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D0%BB%D0%B5%D1%80%D1%8B)
* [5См. также](https://ru.wikipedia.org/wiki/RISC-V#%D0%A1%D0%BC._%D1%82%D0%B0%D0%BA%D0%B6%D0%B5)
* [6Примечания](https://ru.wikipedia.org/wiki/RISC-V#%D0%9F%D1%80%D0%B8%D0%BC%D0%B5%D1%87%D0%B0%D0%BD%D0%B8%D1%8F)
* [7Литература](https://ru.wikipedia.org/wiki/RISC-V#%D0%9B%D0%B8%D1%82%D0%B5%D1%80%D0%B0%D1%82%D1%83%D1%80%D0%B0)
* [8Ссылки](https://ru.wikipedia.org/wiki/RISC-V#%D0%A1%D1%81%D1%8B%D0%BB%D0%BA%D0%B8)

**История**

[[править](https://ru.wikipedia.org/w/index.php?title=RISC-V&veaction=edit&section=1) | [править код](https://ru.wikipedia.org/w/index.php?title=RISC-V&action=edit&section=1)]

Идея RISC-V появилась в 2010 году как продолжение исследований по проектированию вычислительных систем в [Калифорнийском университете Беркли](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D0%BB%D0%B8%D1%84%D0%BE%D1%80%D0%BD%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%83%D0%BD%D0%B8%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%82%D0%B5%D1%82_%D0%B2_%D0%91%D0%B5%D1%80%D0%BA%D0%BB%D0%B8) в США, при непосредственном участии [Дэвида Паттерсона](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%82%D1%82%D0%B5%D1%80%D1%81%D0%BE%D0%BD%2C_%D0%94%D1%8D%D0%B2%D0%B8%D0%B4)[[3]](https://ru.wikipedia.org/wiki/RISC-V#cite_note-xakep-riscv-3)[[4]](https://ru.wikipedia.org/wiki/RISC-V#cite_note-contributors-4) — одного из авторов и сторонников применения концепции RISC. Нынешние участники процесса развития RISC-V являются добровольцами из многих научных организаций, университетов и компаний разных стран. В отличие от других академических проектов, сосредоточенных на образовательных целях, RISC-V изначально проектируется для широкого круга компьютерных применений.

В 2015 году для развития, стандартизации и продвижения RISC-V создан международный фонд RISC-V[[5]](https://ru.wikipedia.org/wiki/RISC-V#cite_note-5) и ассоциация со штаб-квартирой в Цюрихе[[6]](https://ru.wikipedia.org/wiki/RISC-V#cite_note-6) в Швейцарии. С 2018 года фонд RISC-V работает в партнёрстве с [The Linux Foundation](https://ru.wikipedia.org/wiki/The_Linux_Foundation)[[7]](https://ru.wikipedia.org/wiki/RISC-V#cite_note-7). В руководство и технические комитеты входят компании из разных стран, в том числе два российских разработчика процессорных ядер — [Syntacore](https://ru.wikipedia.org/wiki/Syntacore)[[8]](https://ru.wikipedia.org/wiki/RISC-V#cite_note-8) и CloudBEAR, а также два разработчика системного программного обеспечения — [Альт Линукс](https://ru.wikipedia.org/wiki/ALT_Linux_%28%D0%BA%D0%BE%D0%BC%D0%BF%D0%B0%D0%BD%D0%B8%D1%8F%29) и [Астра Линукс](https://ru.wikipedia.org/wiki/Astra_Linux)[[9]](https://ru.wikipedia.org/wiki/RISC-V#cite_note-9).

В феврале 2022 года компания [Intel](https://ru.wikipedia.org/wiki/Intel) объявила[[10]](https://ru.wikipedia.org/wiki/RISC-V#cite_note-10) об инвестировании в развитие RISC-V одного миллиарда долларов и вошла в состав руководства RISC-V. В сентябре 2022 года в России образован Альянс RISC-V[[11]](https://ru.wikipedia.org/wiki/RISC-V#cite_note-11)[[12]](https://ru.wikipedia.org/wiki/RISC-V#cite_note-12). По состоянию на декабрь 2022 года 13 из 25 мест в совете директоров RISC-V занимают китайские компании и организации, ведущую роль из которых занимает [Китайская академия наук](https://ru.wikipedia.org/wiki/%D0%9A%D0%B8%D1%82%D0%B0%D0%B9%D1%81%D0%BA%D0%B0%D1%8F_%D0%B0%D0%BA%D0%B0%D0%B4%D0%B5%D0%BC%D0%B8%D1%8F_%D0%BD%D0%B0%D1%83%D0%BA).

**Базовая спецификация «RV32I»**

[[править](https://ru.wikipedia.org/w/index.php?title=RISC-V&veaction=edit&section=2) | [править код](https://ru.wikipedia.org/w/index.php?title=RISC-V&action=edit&section=2)]

Базовая спецификация RV32I *(RV — RISC-V, 32-разрядная, I означает Integer —*[*целочисленную арифметику*](https://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BB%D0%BE%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%BD%D0%B0%D1%8F_%D0%B0%D1%80%D0%B8%D1%84%D0%BC%D0%B5%D1%82%D0%B8%D0%BA%D0%B0)*), содержит набор из 32 регистров и* включает 39 инструкций. Используется 6 типов кодирования инструкций (форматов).

Базовые расширения:

* M — целочисленное умножение/деление
* A — атомарные операции с памятью
* F и D — вычисления с плавающей точкой с дополнительным набором регистров (одинарной Float и двойной Double точности)
* C — сжатый формат команд (подмножество RV32I), для удвоения плотности упаковки в машинном слове наиболее востребованных стандартных инструкций

Базовый набор RV32Е для встраиваемых систем совпадает по кодированию и набору инструкции с RV32I, но содержит только 16 регистров. Применяется, например, в недорогих микроконтроллерах.

**Система команд**

[[править](https://ru.wikipedia.org/w/index.php?title=RISC-V&veaction=edit&section=3) | [править код](https://ru.wikipedia.org/w/index.php?title=RISC-V&action=edit&section=3)]

В архитектуре RISC-V имеется обязательное для реализации небольшое подмножество команд (набор инструкций I — Integer) и несколько стандартных опциональных расширений.

В базовый набор входят инструкции условной и безусловной передачи управления/ветвления, минимальный набор арифметических/битовых операций на регистрах, операций с памятью (load/store), а также небольшое число служебных инструкций.

Операции ветвления не используют каких-либо общих флагов как результатов ранее выполненных операций сравнения, а непосредственно сравнивают свои регистровые операнды. Базис операций сравнения минимален, а для поддержки комплементарных операций операнды просто меняются местами.

Базовое подмножество команд использует следующий набор регистров: специальный регистр x0 (zero), 31 целочисленный регистр общего назначения (x1 — x31), регистр счётчика команд (PC, используется только косвенно), а также множество CSR (Control and Status Registers, может быть адресовано до 4096 CSR).

Для встраиваемых применений может использоваться вариант архитектуры RV32E (Embedded) с сокращённым набором регистров общего назначения (первые 16). Уменьшение количества регистров позволяет не только экономить аппаратные ресурсы, но и сократить затраты памяти и времени на сохранение/восстановление регистров при переключениях контекста.

При одинаковой кодировке инструкций в RISC-V предусмотрены реализации архитектур с 32-, 64- и 128-битными регистрами общего назначения и операциями (RV32I, RV64I и RV128I, соответственно).

Разрядность регистровых операций всегда соответствует размеру регистра, а одни и те же значения в регистрах могут трактоваться как целые числа как со знаком, так и без знака.

Нет операций над частями регистров, нет каких-либо выделенных «регистровых пар».

Операции не сохраняют где-либо биты переноса или переполнения, что приближено к модели операций в языке программирования Си. Также аппаратно не генерируются исключения по переполнению и даже по делению на 0. Все необходимые проверки операндов и результатов операций должны производиться программно.

Целочисленная арифметика расширенной точности (большей, чем разрядность регистра) должна явно использовать операции вычисления старших битов результата. Например, для получения старших битов произведения регистра на регистр имеются специальные инструкции.

Размер операнда может отличаться от размера регистра только в операциях с памятью. Транзакции к памяти осуществляются блоками, размер в байтах которых должен быть целой неотрицательной степенью 2, от одного байта до размера регистра включительно. Операнд в памяти должен иметь «естественное выравнивание» (адрес кратен размеру операнда).

Архитектура использует только модель [little-endian](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%80%D1%8F%D0%B4%D0%BE%D0%BA_%D0%B1%D0%B0%D0%B9%D1%82%D0%BE%D0%B2#%D0%9F%D0%BE%D1%80%D1%8F%D0%B4%D0%BE%D0%BA_%D0%BE%D1%82_%D0%BC%D0%BB%D0%B0%D0%B4%D1%88%D0%B5%D0%B3%D0%BE_%D0%BA_%D1%81%D1%82%D0%B0%D1%80%D1%88%D0%B5%D0%BC%D1%83) — первый байт операнда в памяти соответствует младшим битам значений регистрового операнда.

Для пары инструкций сохранения/загрузки регистра операнд в памяти определяется размером регистра выбранной архитектуры, а не кодировкой инструкции (код инструкции один и тот же для RV32I, RV64I и RV128I, но размер операндов 4, 8 и 16 байт соответственно), что соответствует размеру указателя, типам языка программирования C size\_t или разности указателей.

Для всех допустимых размеров операндов в памяти, меньших, чем размер регистра, имеются отдельные инструкции загрузки/сохранения младших битов регистра, в том числе для загрузки из памяти в регистр есть парные варианты инструкций, которые позволяют трактовать загружаемое значение как со знаком (старшим знаковым битом значения из памяти заполняются старшие биты регистра) или без знака (старшие биты регистра устанавливаются в 0).

Инструкции базового набора имеют длину 32 бита с выравниванием на границу 32-битного слова, но в общем формате предусмотрены инструкции различной длины (стандартно — от 16 до 192 бит с шагом в 16 бит) с выравниванием на границу 16-битного слова. Полная длина инструкции декодируется унифицированным способом из её первого 16-битного слова.

Для наиболее часто используемых инструкций стандартизовано применение их аналогов в более компактной 16-битной кодировке (C — Compressed extension).

Операции умножения, деления и вычисления остатка не входят в минимальный набор инструкций, а выделены в отдельное расширение (M — Multiply extension). Имеется ряд доводов в пользу разделения и данного набора на два отдельных (умножение и деление).

Стандартизован отдельный набор атомарных операций (A — Atomic extension).

Поскольку кодировка базового набора инструкций не зависит от разрядности архитектуры, то один и тот же код потенциально может запускаться на различных RISC-V архитектурах, определять разрядность и другие параметры текущей архитектуры, наличие расширений системы инструкций, а потом автоконфигурироваться для целевой среды выполнения.

Спецификацией RISC-V предусмотрено несколько областей в пространстве кодировок инструкций для пользовательских «X-расширений» архитектуры, которые поддерживаются на уровне ассемблера, как группы инструкций custom0 и custom1.