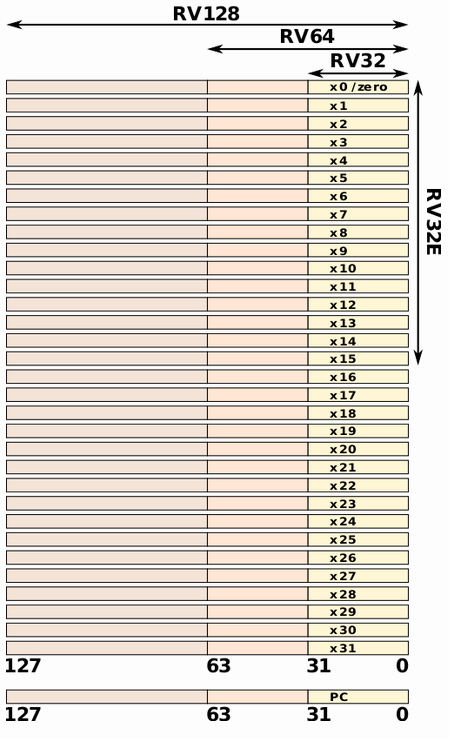
# RISC-V

RISC-V — это открытая спецификация архитектуры набора команд (ISA, Instruction Set Architecture), основанного на принципах сокращенного набора инструкций (RISC, Reduced Instruction Set Computer). Другими словами, RISC-V представляет собой описание ассемблерных инструкций, способа их кодирования, и семантику работы. Придерживание принципов RISC означает, что RISC-V не вводит лишних инструкций без сильной надобности. Например, в нем нет специальных «строковых» операций, таких, как инструкции movs, stos и cmps в ассемблере x86/x64. За счет этого существенно упрощается архитектура процессора.

## Архитектуры

|  |  |
| --- | --- |
| Сокращение | Наименование |
| **RV32I** | Базовый набор с целочисленными операциями, 32-битный |
| **RV32E** | Базовый набор с целочисленными операциями для встраиваемых систем, 32-битный, 16 регистров |
| **RV64I** | Базовый набор с целочисленными операциями, 64-битный |
| **RV128I** | Базовый набор с целочисленными операциями, 128-битный |
|  | **Расширения** |
| **M** | Целочисленное умножение и деление (Integer Multiplication and Division) |
| **A** | Атомарные операции (Atomic Instructions) |
| **F** | Арифметические операции с плавающей запятой над числами одинарной точности (Single-Precision Floating-Point) |
| **D** | Арифметические операции с плавающей запятой над числами двойной точности (Double-Precision Floating-Point) |
| **G** | Сокращеное обозначение для комплекта из базового и стандартного наборов команд (IMAFD) |
| **Q** | Арифметические операции с плавающей запятой над числами четвертной точности |
| **L** | Арифметические операции над числами с фиксированной запятой (Decimal Floating-Point) |
| **C** | Сокращённые имена для команд (Compressed Instructions) |
| **B** | Битовые операции (Bit Manipulation) |
| **J** | Двоичная трансляция и поддержка динамической компиляции (Dynamically Translated Languages) |
| **T** | Транзакционная память (Transactional Memory) |
| **P** | Короткие SIMD-операции (Packed-SIMD Instructions) |
| **V** | Векторные расширения (Vector Operations) |
| **N** | Инструкции прерывания (User-Level Interrupts) |

# Регистры

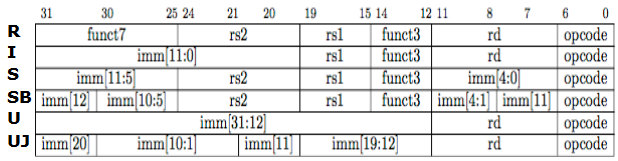
Архитектура набора команд RISC-V имеет 32 регистра стандартного назначения **x1-x31**, которые содержат целочисленные значения. Регистр **x0** всегда равен нулю (hardwired). Существует так же регистр **pc** который содержит адрес текущей инструкции.

Длины этих регистров для 32-х, 64-х и 128-и битных архитектур равны 32, 64 и 128 бит соответственно.

Архитектура RV32E имеет 16 регистров x0-x16 длиной в 32 бита.

Для операций над числами в бинарных форматах плавающей запятой используется набор дополнительных 32 регистров FPU (Floating Point Unit), которые совместно используются расширениями базового набора инструкций для трёх вариантов точности: одинарной — 32 бита (F extension), двойной — 64 бита (D — Double precision extension), а также четверной — 128 бит (Q — Quadruple precision extension).

## Формат инструкций



## Инструкции

#### RV32I Базовые целочисленные инструкции

|  |  |  |  |
| --- | --- | --- | --- |
| **Инструкция** | **Имя** | **Псевдокод** | **Формат** |
| LUI rd,imm | Load Upper Immediate | rd ← imm | U |
| AUIPC rd,offset | Add Upper Immediate to PC | rd ← pc + offset | U |
| JAL rd,offset | Jump and Link | rd ← pc + length(inst) pc ← pc + offset | UJ |
| JALR rd,rs1,offset | Jump and Link Register | rd ← pc + length(inst) pc ← (rs1 + offset) ∧ -2 | UJ |
| BEQ rs1,rs2,offset | Branch Equal | if rs1 = rs2 then pc ← pc + offset | SB |
| BNE rs1,rs2,offset | Branch Not Equal | if rs1 ≠ rs2 then pc ← pc + offset | SB |
| BLT rs1,rs2,offset | Branch Less Than | if rs1 < rs2 then pc ← pc + offset | SB |
| BGE rs1,rs2,offset | Branch Greater than Equal | if rs1 ≥ rs2 then pc ← pc + offset | SB |
| BLTU rs1,rs2,offset | Branch Less Than Unsigned | if rs1 < rs2 then pc ← pc + offset | SB |
| BGEU rs1,rs2,offset | Branch Greater than Equal Unsigned | if rs1 ≥ rs2 then pc ← pc + offset | SB |
| LB rd,offset(rs1) | Load Byte | rd ← s8[rs1 + offset] | I |
| LH rd,offset(rs1) | Load Half | rd ← s16[rs1 + offset] | I |
| LW rd,offset(rs1) | Load Word | rd ← s32[rs1 + offset] | I |
| LBU rd,offset(rs1) | Load Byte Unsigned | rd ← u8[rs1 + offset] | I |
| LHU rd,offset(rs1) | Load Half Unsigned | rd ← u16[rs1 + offset] | I |
| SB rs2,offset(rs1) | Store Byte | u8[rs1 + offset] ← rs2 | S |
| SH rs2,offset(rs1) | Store Half | u16[rs1 + offset] ← rs2 | S |
| SW rs2,offset(rs1) | Store Word | u32[rs1 + offset] ← rs2 | S |
| ADDI rd,rs1,imm | Add Immediate | rd ← rs1 + sx(imm) | R |
| SLTI rd,rs1,imm | Set Less Than Immediate | rd ← sx(rs1) < sx(imm) | I |
| SLTIU rd,rs1,imm | Set Less Than Immediate Unsigned | rd ← ux(rs1) < ux(imm) | I |
| XORI rd,rs1,imm | Xor Immediate | rd ← ux(rs1) ⊕ ux(imm) | I |
| ORI rd,rs1,imm | Or Immediate | rd ← ux(rs1) ∨ ux(imm) | I |
| ANDI rd,rs1,imm | And Immediate | rd ← ux(rs1) ∧ ux(imm) | I |
| SLLI rd,rs1,imm | Shift Left Logical Immediate | rd ← ux(rs1) « ux(imm) | I |
| SRLI rd,rs1,imm | Shift Right Logical Immediate | rd ← ux(rs1) » ux(imm) | I |
| SRAI rd,rs1,imm | Shift Right Arithmetic Immediate | rd ← sx(rs1) » ux(imm) | I |
| ADD rd,rs1,rs2 | Add | rd ← sx(rs1) + sx(rs2) | R |
| SUB rd,rs1,rs2 | Subtract | rd ← sx(rs1) - sx(rs2) | R |
| SLL rd,rs1,rs2 | Shift Left Logical | rd ← ux(rs1) « rs2 | R |
| SLT rd,rs1,rs2 | Set Less Than | rd ← sx(rs1) < sx(rs2) | R |
| SLTU rd,rs1,rs2 | Set Less Than Unsigned | rd ← ux(rs1) < ux(rs2) | R |
| XOR rd,rs1,rs2 | Xor | rd ← ux(rs1) ⊕ ux(rs2) | R |
| SRL rd,rs1,rs2 | Shift Right Logical | rd ← ux(rs1) » rs2 | R |
| SRA rd,rs1,rs2 | Shift Right Arithmetic | rd ← sx(rs1) » rs2 | R |
| OR rd,rs1,rs2 | Or | rd ← ux(rs1) ∨ ux(rs2) | R |
| AND rd,rs1,rs2 | And | rd ← ux(rs1) ∧ ux(rs2) | R |
| FENCE pred,succ | Fence |  | I |
| FENCE.I | Fence Instruction |  | I |

## Источники

* <https://content.riscv.org/wp-content/uploads/2017/05/riscv-spec-v2.2.pdf>
* <https://ru.wikipedia.org/wiki/RISC-V> архитектуры
* <https://rv8.io/isa.html> инструкции
* <https://www.cl.cam.ac.uk/teaching/1617/ECAD+Arch/files/docs/RISCVGreenCardv8-20151013.pdf> формат инструкций
* <https://github.com/riscv/riscv-opcodes/blob/master/opcodes> опкоды