

Risc V Cheat Sheet

RISC-V Instruction-Set			
Arithmetic Operation			
Mnemonic	Instruction	Type	Description
add	add rd, rs1, rs2	R	$rd \leftarrow rs1 + rs2$
addw	addw rd, rs1, rs2	R	$rd \leftarrow rs1 + rs2$
addi	addi rd, rs1, imm	I	$rd \leftarrow rs1 + imm$
addiw	addiw rd, rs1, imm	I	$rd \leftarrow rs1 + imm$
addi4spn	addi4spn rd, zero, imm	I	$rd \leftarrow 4 \times imm$
addi16spn	addi16spn rd, zero, imm	I	$rd \leftarrow 16 \times imm$
addi32spn	addi32spn rd, zero, imm	I	$rd \leftarrow 32 \times imm$
addi64spn	addi64spn rd, zero, imm	I	$rd \leftarrow 64 \times imm$
addi128spn	addi128spn rd, zero, imm	I	$rd \leftarrow 128 \times imm$
addi256spn	addi256spn rd, zero, imm	I	$rd \leftarrow 256 \times imm$
addi512spn	addi512spn rd, zero, imm	I	$rd \leftarrow 512 \times imm$
addi1024spn	addi1024spn rd, zero, imm	I	$rd \leftarrow 1024 \times imm$
addi2048spn	addi2048spn rd, zero, imm	I	$rd \leftarrow 2048 \times imm$
addi4096spn	addi4096spn rd, zero, imm	I	$rd \leftarrow 4096 \times imm$
addi8192spn	addi8192spn rd, zero, imm	I	$rd \leftarrow 8192 \times imm$
addi16384spn	addi16384spn rd, zero, imm	I	$rd \leftarrow 16384 \times imm$
addi32768spn	addi32768spn rd, zero, imm	I	$rd \leftarrow 32768 \times imm$
addi65536spn	addi65536spn rd, zero, imm	I	$rd \leftarrow 65536 \times imm$
addi131072spn	addi131072spn rd, zero, imm	I	$rd \leftarrow 131072 \times imm$
addi262144spn	addi262144spn rd, zero, imm	I	$rd \leftarrow 262144 \times imm$
addi524288spn	addi524288spn rd, zero, imm	I	$rd \leftarrow 524288 \times imm$
addi1048576spn	addi1048576spn rd, zero, imm	I	$rd \leftarrow 1048576 \times imm$
addi2097152spn	addi2097152spn rd, zero, imm	I	$rd \leftarrow 2097152 \times imm$
addi4194304spn	addi4194304spn rd, zero, imm	I	$rd \leftarrow 4194304 \times imm$
addi8388608spn	addi8388608spn rd, zero, imm	I	$rd \leftarrow 8388608 \times imm$
addi16777216spn	addi16777216spn rd, zero, imm	I	$rd \leftarrow 16777216 \times imm$
addi33554432spn	addi33554432spn rd, zero, imm	I	$rd \leftarrow 33554432 \times imm$
addi67108864spn	addi67108864spn rd, zero, imm	I	$rd \leftarrow 67108864 \times imm$
addi134217728spn	addi134217728spn rd, zero, imm	I	$rd \leftarrow 134217728 \times imm$
addi268435456spn	addi268435456spn rd, zero, imm	I	$rd \leftarrow 268435456 \times imm$
addi536870912spn	addi536870912spn rd, zero, imm	I	$rd \leftarrow 536870912 \times imm$
addi1073741824spn	addi1073741824spn rd, zero, imm	I	$rd \leftarrow 1073741824 \times imm$
addi2147483648spn	addi2147483648spn rd, zero, imm	I	$rd \leftarrow 2147483648 \times imm$
addi4294967296spn	addi4294967296spn rd, zero, imm	I	$rd \leftarrow 4294967296 \times imm$
addi8589934592spn	addi8589934592spn rd, zero, imm	I	$rd \leftarrow 8589934592 \times imm$
addi17179869184spn	addi17179869184spn rd, zero, imm	I	$rd \leftarrow 17179869184 \times imm$
addi34359738368spn	addi34359738368spn rd, zero, imm	I	$rd \leftarrow 34359738368 \times imm$
addi68719476736spn	addi68719476736spn rd, zero, imm	I	$rd \leftarrow 68719476736 \times imm$
addi137438953472spn	addi137438953472spn rd, zero, imm	I	$rd \leftarrow 137438953472 \times imm$
addi274877906944spn	addi274877906944spn rd, zero, imm	I	$rd \leftarrow 274877906944 \times imm$
addi549755813888spn	addi549755813888spn rd, zero, imm	I	$rd \leftarrow 549755813888 \times imm$
addi1099511627776spn	addi1099511627776spn rd, zero, imm	I	$rd \leftarrow 1099511627776 \times imm$
addi2199023255552spn	addi2199023255552spn rd, zero, imm	I	$rd \leftarrow 2199023255552 \times imm$
addi4398046511104spn	addi4398046511104spn rd, zero, imm	I	$rd \leftarrow 4398046511104 \times imm$
addi8796093022208spn	addi8796093022208spn rd, zero, imm	I	$rd \leftarrow 8796093022208 \times imm$
addi17592186044416spn	addi17592186044416spn rd, zero, imm	I	$rd \leftarrow 17592186044416 \times imm$
addi35184372088832spn	addi35184372088832spn rd, zero, imm	I	$rd \leftarrow 35184372088832 \times imm$
addi70368744177664spn	addi70368744177664spn rd, zero, imm	I	$rd \leftarrow 70368744177664 \times imm$
addi140737488355328spn	addi140737488355328spn rd, zero, imm	I	$rd \leftarrow 140737488355328 \times imm$
addi281474976710656spn	addi281474976710656spn rd, zero, imm	I	$rd \leftarrow 281474976710656 \times imm$
addi562949953421312spn	addi562949953421312spn rd, zero, imm	I	$rd \leftarrow 562949953421312 \times imm$
addi1125899906842624spn	addi1125899906842624spn rd, zero, imm	I	$rd \leftarrow 1125899906842624 \times imm$
addi2251799813685248spn	addi2251799813685248spn rd, zero, imm	I	$rd \leftarrow 2251799813685248 \times imm$
addi4503599627370496spn	addi4503599627370496spn rd, zero, imm	I	$rd \leftarrow 4503599627370496 \times imm$
addi9007199254740992spn	addi9007199254740992spn rd, zero, imm	I	$rd \leftarrow 9007199254740992 \times imm$
addi18014398509481984spn	addi18014398509481984spn rd, zero, imm	I	$rd \leftarrow 18014398509481984 \times imm$
addi36028797018963968spn	addi36028797018963968spn rd, zero, imm	I	$rd \leftarrow 36028797018963968 \times imm$
addi72057594037927936spn	addi72057594037927936spn rd, zero, imm	I	$rd \leftarrow 72057594037927936 \times imm$
addi144115188075855872spn	addi144115188075855872spn rd, zero, imm	I	$rd \leftarrow 144115188075855872 \times imm$
addi288230376151711744spn	addi288230376151711744spn rd, zero, imm	I	$rd \leftarrow 288230376151711744 \times imm$
addi576460752303423488spn	addi576460752303423488spn rd, zero, imm	I	$rd \leftarrow 576460752303423488 \times imm$
addi1152921504606846976spn	addi1152921504606846976spn rd, zero, imm	I	$rd \leftarrow 1152921504606846976 \times imm$
addi2305843009213693952spn	addi2305843009213693952spn rd, zero, imm	I	$rd \leftarrow 2305843009213693952 \times imm$
addi4611686018427387904spn	addi4611686018427387904spn rd, zero, imm	I	$rd \leftarrow 4611686018427387904 \times imm$
addi9223372036854775808spn	addi9223372036854775808spn rd, zero, imm	I	$rd \leftarrow 9223372036854775808 \times imm$
addi18446744073709551616spn	addi18446744073709551616spn rd, zero, imm	I	$rd \leftarrow 18446744073709551616 \times imm$
addi36893488147419103232spn	addi36893488147419103232spn rd, zero, imm	I	$rd \leftarrow 36893488147419103232 \times imm$
addi73786976294838206464spn	addi73786976294838206464spn rd, zero, imm	I	$rd \leftarrow 73786976294838206464 \times imm$
addi147573952589676412928spn	addi147573952589676412928spn rd, zero, imm	I	$rd \leftarrow 147573952589676412928 \times imm$
addi295147905179352825856spn	addi295147905179352825856spn rd, zero, imm	I	$rd \leftarrow 295147905179352825856 \times imm$
addi590295810358705651712spn	addi590295810358705651712spn rd, zero, imm	I	$rd \leftarrow 590295810358705651712 \times imm$
addi1180591620717411303424spn	addi1180591620717411303424spn rd, zero, imm	I	$rd \leftarrow 1180591620717411303424 \times imm$
addi2361183241434822606848spn	addi2361183241434822606848spn rd, zero, imm	I	$rd \leftarrow 2361183241434822606848 \times imm$
addi4722366482869645213696spn	addi4722366482869645213696spn rd, zero, imm	I	$rd \leftarrow 4722366482869645213696 \times imm$
addi9444732965739290427392spn	addi9444732965739290427392spn rd, zero, imm	I	$rd \leftarrow 9444732965739290427392 \times imm$
addi18889465931478580854784spn	addi18889465931478580854784spn rd, zero, imm	I	$rd \leftarrow 18889465931478580854784 \times imm$
addi37778931862957161709568spn	addi37778931862957161709568spn rd, zero, imm	I	$rd \leftarrow 37778931862957161709568 \times imm$
addi75557863725914323419136spn	addi75557863725914323419136spn rd, zero, imm	I	$rd \leftarrow 75557863725914323419136 \times imm$
addi151115727451828646838272spn	addi151115727451828646838272spn rd, zero, imm	I	$rd \leftarrow 151115727451828646838272 \times imm$
addi302231454903657293676544spn	addi302231454903657293676544spn rd, zero, imm	I	$rd \leftarrow 302231454903657293676544 \times imm$
addi604462909807314587353088spn	addi604462909807314587353088spn rd, zero, imm	I	$rd \leftarrow 604462909807314587353088 \times imm$
addi1208925819614629174706176spn	addi1208925819614629174706176spn rd, zero, imm	I	$rd \leftarrow 1208925819614629174706176 \times imm$
addi2417851639229258349412352spn	addi2417851639229258349412352spn rd, zero, imm	I	$rd \leftarrow 2417851639229258349412352 \times imm$
addi4835703278458516698824704spn	addi4835703278458516698824704spn rd, zero, imm	I	$rd \leftarrow 4835703278458516698824704 \times imm$
addi9671406556917033397649408spn	addi9671406556917033397649408spn rd, zero, imm	I	$rd \leftarrow 9671406556917033397649408 \times imm$
addi19342813113834066795298816spn	addi19342813113834066795298816spn rd, zero, imm	I	$rd \leftarrow 19342813113834066795298816 \times imm$
addi38685626227668133590597632spn	addi38685626227668133590597632spn rd, zero, imm	I	$rd \leftarrow 38685626227668133590597632 \times imm$
addi77371252455336267181195264spn	addi77371252455336267181195264spn rd, zero, imm	I	$rd \leftarrow 77371252455336267181195264 \times imm$
addi154742504910672534362390528spn	addi154742504910672534362390528spn rd, zero, imm	I	$rd \leftarrow 154742504910672534362390528 \times imm$
addi309485009821345068724781056spn	addi309485009821345068724781056spn rd, zero, imm	I	$rd \leftarrow 309485009821345068724781056 \times imm$
addi618970019642690137449562112spn	addi618970019642690137449562112spn rd, zero, imm	I	$rd \leftarrow 618970019642690137449562112 \times imm$
addi1237940039285380274899124224spn	addi1237940039285380274899124224spn rd, zero, imm	I	$rd \leftarrow 1237940039285380274899124224 \times imm$
addi2475880078570760549798248448spn	addi2475880078570760549798248448spn rd, zero, imm	I	$rd \leftarrow 2475880078570760549798248448 \times imm$
addi4951760157141521099596496896spn	addi4951760157141521099596496896spn rd, zero, imm	I	$rd \leftarrow 4951760157141521099596496896 \times imm$
addi9903520314283042199193293792spn	addi9903520314283042199193293792spn rd, zero, imm	I	$rd \leftarrow 9903520314283042199193293792 \times imm$
addi19807040628566084398386587584spn	addi19807040628566084398386587584spn rd, zero, imm	I	$rd \leftarrow 19807040628566084398386587584 \times imm$
addi39614081257132168796773175168spn	addi39614081257132168796773175168spn rd, zero, imm	I	$rd \leftarrow 39614081257132168796773175168 \times imm$
addi79228162514264337593546350336spn	addi79228162514264337593546350336spn rd, zero, imm	I	$rd \leftarrow 79228162514264337593546350336 \times imm$
addi158456325028528675187092700672spn	addi158456325028528675187092700672spn rd, zero, imm	I	$rd \leftarrow 158456325028528675187092700672 \times imm$
addi316912650057057350374185401344spn	addi316912650057057350374185401344spn rd, zero, imm	I	$rd \leftarrow 316912650057057350374185401344 \times imm$
addi633825300114114700748370802688spn	addi633825300114114700748370802688spn rd, zero, imm	I	$rd \leftarrow 633825300114114700748370802688 \times imm$
addi1267650600228229401496741605376spn	addi1267650600228229401496741605376spn rd, zero, imm	I	$rd \leftarrow 1267650600228229401496741605376 \times imm$
addi2535301200456458802993483210752spn	addi2535301200456458802993483210752spn rd, zero, imm	I	$rd \leftarrow 2535301200456458802993483210752 \times imm$
addi5070602400912917605986966421504spn	addi5070602400912917605986966421504spn rd, zero, imm	I	$rd \leftarrow 5070602400912917605986966421504 \times imm$
addi10141204801825835211973932843008spn	addi10141204801825835211973932843008spn rd, zero, imm	I	$rd \leftarrow 10141204801825835211973932843008 \times imm$
addi20282409603651670423947865686016spn	addi20282409603651670423947865686016spn rd, zero, imm	I	$rd \leftarrow 20282409603651670423947865686016 \times imm$
addi40564819207303340847895731372032spn	addi40564819207303340847895731372032spn rd, zero, imm	I	$rd \leftarrow 40564819207303340847895731372032 \times imm$
addi81129638414606681695791462744064spn	addi81129638414606681695791462744064spn rd, zero, imm	I	$rd \leftarrow 81129638414606681695791462744064 \times imm$
addi162259276829213363391782925488128spn	addi162259276829213363391782925488128spn rd, zero, imm	I	$rd \leftarrow 162259276829213363391782925488128 \times imm$
addi324518553658426726783565850976256spn	addi324518553658426726783565850976256spn rd, zero, imm	I	$rd \leftarrow 324518553658426726783565850976256 \times imm$
addi649037107316853453567131701952512spn	addi649037107316853453567131701952512spn rd, zero, imm	I	$rd \leftarrow 649037107316853453567131701952512 \times imm$
addi1298074214633706907134263403905024spn	addi1298074214633706907134263403905024spn rd, zero, imm	I	$rd \leftarrow 1298074214633706907134263403905024 \times imm$
addi2596148429267413814268526807810048spn	addi2596148429267413814268526807810048spn rd, zero, imm	I	$rd \leftarrow 2596148429267413814268526807810048 \times imm$
addi5192296858534827628537053615620096spn	addi5192296858534827628537053615620096spn rd, zero, imm	I	$rd \leftarrow 5192296858534827628537053615620096 \times imm$
addi10384593717069655257074107231240192spn	addi10384593717069655257074107231240192spn rd, zero, imm	I	$rd \leftarrow 10384593717069655257074107231240192 \times imm$
addi20769187434139310514148214462480384spn	addi20769187434139310514148214462480384spn rd, zero, imm	I	$rd \leftarrow 20769187434139310514148214462480384 \times imm$
addi41538374868278621028296428924960768spn	addi41538374868278621028296428924960768spn rd, zero, imm	I	$rd \leftarrow 41538374868278621028296428924960768 \times imm$
addi83076749736557242056592857849921536spn	addi83076749736557242056592857849921536spn rd, zero, imm	I	$rd \leftarrow 83076749736557242056592857849921536 \times imm$
addi16615349947311448411318571569963072spn	addi16615349947311448411318571569963072spn rd, zero, imm	I	$rd \leftarrow 1661534994731144841$

2. **Simplicity:** The architecture is designed to be simple and clean, enabling easier implementation and understanding.
3. **Modularity:** RISC-V allows for extensions and custom instructions, which means developers can add functionality without changing the core architecture.
4. **Support for Multiple Data Types:** RISC-V supports integer and floating-point operations, making it suitable for a wide range of applications.
5. **Rich Ecosystem:** A growing ecosystem of tools, compilers, and development environments is available, facilitating easier adoption and integration.

RISC-V Instruction Set Overview

The RISC-V instruction set is divided into several categories:

- **Base Integer Instructions (RV32I/RV64I):** The foundation of the RISC-V ISA, providing basic integer operations.
- **Standard Extensions:** Optional features that enhance the instruction set, including:
 - **M:** Integer Multiplication and Division
 - **A:** Atomic Operations
 - **F:** Single-Precision Floating-Point
 - **D:** Double-Precision Floating-Point
 - **C:** Compressed Instructions
- **Custom Extensions:** Users can define their own instructions tailored to specific applications.

Basic RISC-V Instruction Types

RISC-V instructions can be categorized into several formats:

1. **R-Type (Register):** These instructions perform operations on register values. Examples include:
 - **ADD:** Addition
 - **SUB:** Subtraction
 - **AND:** Bitwise AND
2. **I-Type (Immediate):** These instructions use an immediate value and a register. Examples include:
 - **ADDI:** Add Immediate
 - **ANDI:** AND Immediate
 - **LW:** Load Word

3. S-Type (Store): Used for storing data from registers to memory. Example:
 - SW: Store Word
4. B-Type (Branch): Used for branching in code. Examples include:
 - BEQ: Branch if Equal
 - BNE: Branch if Not Equal
5. U-Type (Upper Immediate): Used for loading upper immediate values. Examples include:
 - LUI: Load Upper Immediate
 - AUIPC: Add Upper Immediate to PC
6. J-Type (Jump): Used for jumping to a different instruction address. Example:
 - JAL: Jump and Link

Essential RISC-V Instructions

Understanding some key instructions is crucial for anyone starting with RISC-V. Below is a list of essential RISC-V instructions along with their functions:

1. **ADD**: Adds two registers and stores the result in a destination register.
2. **SUB**: Subtracts one register from another and stores the result.
3. **AND**: Performs a bitwise AND operation on two registers.
4. **OR**: Performs a bitwise OR operation on two registers.
5. **SLT**: Sets the destination register to 1 if the first source register is less than the second.
6. **LW**: Loads a word from memory into a register.
7. **SW**: Stores a word from a register into memory.
8. **ADDI**: Adds an immediate value to a register.
9. **JAL**: Jumps to a specified address and stores the return address.
10. **BEQ**: Branches to a specified address if two registers are equal.

RISC-V Assembly Language

The assembly language for RISC-V uses a simple syntax that allows programmers to write low-level code that directly corresponds to machine instructions. Here's an example of a simple RISC-V program that adds two numbers:

```
```assembly
.data
num1: .word 5
num2: .word 10
result: .word 0

.text
main:
lw t0, num1 Load num1 into register t0
lw t1, num2 Load num2 into register t1
add t2, t0, t1 Add t0 and t1, store result in t2
sw t2, result Store the result in memory
li a0, 10 Exit code
ecall Make system call
```
```

This program demonstrates basic operations like loading values from memory, performing addition, and storing the result back.

Development Tools for RISC-V

To work effectively with RISC-V, several development tools can enhance the programming experience:

- GNU Compiler Collection (GCC): The RISC-V version of GCC supports compiling C and C++ for RISC-V architecture.
- RISC-V Emulator (Spike): An emulator that can run RISC-V binaries and is useful for debugging.
- RISC-V Simulator (RISC-V ISS): Software that simulates RISC-V hardware and allows for testing and performance analysis.
- Integrated Development Environments (IDEs): Tools like VS Code or Eclipse can be configured with RISC-V plugins to facilitate development.

Community and Resources

The RISC-V community is vibrant and growing, with numerous resources

available for learning and collaboration. Here are some key resources:

- RISC-V Foundation: The official organization supporting RISC-V development, providing updates and resources.
- Online Courses: Platforms like Coursera and edX offer courses on RISC-V architecture and programming.
- GitHub Repositories: Many open-source projects and tools related to RISC-V can be found on GitHub.
- Forums and Discussion Groups: Engaging in community forums can provide insights and support from fellow developers.

Conclusion

The **RISC-V cheat sheet** serves as an invaluable tool for understanding and working with this innovative instruction set architecture. With its open standard, modular approach, and rich ecosystem, RISC-V is positioned to play a significant role in the future of computing. Whether you're an academic researcher, a hardware designer, or a software developer, familiarity with RISC-V will open up numerous opportunities in the rapidly evolving landscape of technology.

Frequently Asked Questions

What is RISC-V and why is it important?

RISC-V is an open standard instruction set architecture (ISA) that enables anyone to design and manufacture RISC-V processors without licensing fees. Its importance lies in its flexibility, scalability, and the growing community support, making it a popular choice for academic research and industry applications.

What are the key features of a RISC-V cheat sheet?

A RISC-V cheat sheet typically includes essential information such as the instruction set, register usage, encoding formats, common assembly language commands, and system call conventions, providing a quick reference for developers and engineers.

Where can I find a comprehensive RISC-V cheat sheet?

Comprehensive RISC-V cheat sheets can be found on websites like GitHub, academic resources, or dedicated RISC-V community pages, where contributors share their compilations of important information and resources for developers.

How can a RISC-V cheat sheet assist in programming?

A RISC-V cheat sheet assists programmers by providing quick access to syntax, instructions, and examples, allowing them to write, debug, and optimize code more efficiently without needing to memorize every detail of the RISC-V architecture.

What types of projects can benefit from using RISC-V?

Projects that involve embedded systems, IoT devices, high-performance computing, and educational tools can benefit from using RISC-V due to its customizable nature and open-source ecosystem, allowing developers to tailor solutions to specific needs.

Are there any online tools or simulators that utilize RISC-V cheat sheets?

Yes, there are several online tools and simulators, such as RISC-V simulators like Spike and RISC-V tools from the RISC-V Foundation, that incorporate cheat sheets to help users understand the architecture and test their code in a simulated environment.

Find other PDF article:

<https://soc.up.edu.ph/17-scan/Book?trackid=hbK94-4436&title=digestive-system-webquest-answer-key.pdf>

[Risc V Cheat Sheet](#)

[US] Test your smarts [01-07-22] : r/MicrosoftRewards - Reddit

Jan 7, 2022 · AmySueF [US] Test your smarts [01-07-22] Quiz and Answers News this week quiz answers Pittsburgh 119 Little Caesars ...

BingHomepageQuiz - Reddit

Microsoft Bing Homepage daily quiz questions and their answers

[US] 30 Point Quiz Replaced With 10 Point Single Click - Reddit

Logged on to do my dailies only to find the normal 30 point quiz has been replaced with a 10 point single click option. Checked the ...

[US] Microsoft Rewards Bing - Supersonic Quiz - Reddit

Mar 21, 2023 · [US] Microsoft Rewards Bing - Supersonic Quiz - Aviation? (03/21/2022)

[New Year new you - Monthly punch card & Quiz for January 2022 ... - R...](#)

New Year new you - Monthly punch card & Quiz for January 2022 +150 MR points Punch Card

Reward: 50 MR points for completing ...

Video Embarrassing Boy Busts On Himself When Getting Twerk...

Video Embarrassing Boy Busts On Himself When Getting Twerked On Too Hard At A Party! No description has been added ...

Twerk live - смотреть онлайн все 104 видео на RUTUBE. (322...

Смотрите видео канала Twerk live на RUTUBE. Здесь вы можете посмотреть онлайн все 104 видео в хорошем ...

Sister Twerks on Brother: Funny Family Moment Caught on Cam...

Nov 29, 2020 · TikTok video from LorenSharice (@lorensharice): "Watch as this sister attempts to twerk on her ...

young teen twerks — Yandex: found 28 thousand results

Смотрите также видео: Bandit 7teen - Twerk (Official Visualizer), TWERK 2018 BY ROCIO RAMIREZ / VIVÍ NUESTRO DIA A ...

Twerking on the guy - YouTube

Twerking In A Tree House Night Club On A Saturday IMKOMOS • 621K views • 9 years ago

"Unlock the power of RISC-V with our comprehensive cheat sheet! Discover key concepts

[Back to Home](#)