# Intro to RISC-V

Sparsh Mittal

# Why RISC-V?

Home › Future Tech › AI

## Meta's Bold Move: Embracing RISC-V for AI Acceleration Over GPUs and CPUs

*The RISC-V Revolution Gains Momentum*

BY **ANOCHIE ESTHER** — December 15, 2023 in **AI, Business, Manufacturing, News, Tech** Reading Time: 2 mins read 0

https://techstory.in/risc-v-chosen-over-gpus-and-cpus-by-meta/

MIT Technology Review

Featured Topics Newslett


https://www.technologyreview.com/2023/01/09/1064876/riscv-computer-chips-10-breakthough-technologies-2023/

Qualcomm Products Support Company

Press Note

## Qualcomm to Bring RISC-V Based Wearable Platform to Wear OS by Google

Important first milestone to bring RISC-V compatible CPUs to the Ecosystem

OCT 17, 2023 | SAN DIEGO | Qualcomm products mentioned within this press release are offered by Qualcomm Technologies, Inc. and/or its subsidiaries.

https://www.qualcomm.com/news/releases/2023/10/qualcomm-to-bring-risc-v-based-wearable-platform-to--wear-os-by-

https://riscv.org/news/in-the-news/

# Number of pages and words of ISA manuals

| ISA | Pages | Words | Hours to read | Weeks to read |
|---|---|---|---|---|
| RISC-V | 236 | 76,702 | 6 | 0.2 |
| ARM-32 | 2736 | 895,032 | 79 | 1.9 |
| x86-32 | 2198 | 2,186,259 | 182 | 4.5 |

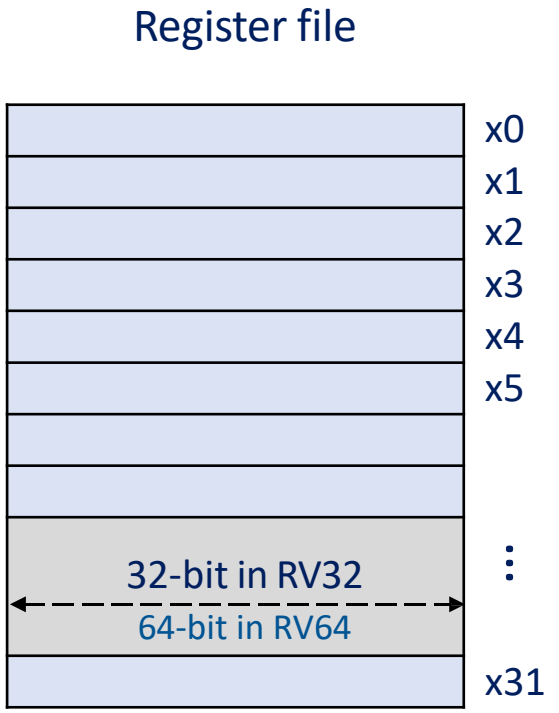
# 2 Versions of RISC-V (based on maximum width of registers supported)

1. RISC-V 32 (RV32): max register width (XLEN) is 32 bits
2. RISC-V 64 (RV64): max register width (XLEN) is 64 bits

RV64 supports RV32 also.

Both versions have 32 registers.

In both of them, each instruction is encoded into 32 bits.



RISC-V ISA includes

– **A small base integer ISA**, usable by itself as a base for customized accelerators or for educational purposes, and

– **Optional standard extensions**, to support general-purpose software development

– **Optional customer extensions**

Mandatory Base integer ISA

**I: Integer instructions:**

– **ALU**

– **Branches/jumps**

– **Loads/stores**

# Standard extensions

- Standard RISC encoding in a fixed 32-bit instruction format
- The "C" extension (compressed extension) offers shorter 16-bit versions of common 32-bit RISC-V instructions (can be intermixed with 32-bit instructions)

| Name | Extension |
|---|---|
| M | Integer Multiply/Divide |
| A | Atomic Instructions |
| F | Single-precision FP |
| D | Double-precision FP |
| G | General-purpose (= IMAFD) |
| Q | Quad-precision FP |
| C | Compressed Instructions |

Do not confuse C extension with C programming language!

# Registers

All the registers can be used as general purpose registers.

Some of them are usually used for specific purposes

| Register | ABI Name | Description | Saver |
|---|---|---|---|
| x0 | zero | Hard-wired zero | — |
| x1 | ra | Return address | Caller |
| x2 | sp | Stack pointer | Callee |
| x3 | gp | Global pointer | — |
| x4 | tp | Thread pointer | — |
| x5 | t0 | Temporary/alternate link register | Caller |
| x6–7 | t1–2 | Temporaries | Caller |
| x8 | s0/fp | Saved register/frame pointer | Callee |
| x9 | s1 | Saved register | Callee |
| x10–11 | a0–1 | Function arguments/return values | Caller |
| x12–17 | a2–7 | Function arguments | Caller |
| x18–27 | s2–11 | Saved registers | Callee |
| x28–31 | t3–6 | Temporaries | Caller |

Dedicating a register to zero is a surprisingly large factor in simplifying the RISC-V ISA

RV32I has 31 registers plus x0. ARM-32 has merely 16 registers while x86-32 has only 8.

# Argument and temporary register

**Argument registers:** x10 to x17 are used to pass arguments to a function. Before calling a function, arguments are copied to these registers.

If more than 8 arguments need to be passed, we use the stack.

**Temporary registers** (t0 to t6): used to hold intermediate values during instruction or function execution.