Assembly Language Lecture Series: RISC-V: An Introduction

Roger Luis Uy
College of Computer Studies
De La Salle University
Manila, Philippines

RISC-V

- RISC-V is an open-source RISC-based Instruction-set Architecture (ISA)
- Designed to support computer architecture research and education
- Also, standard free and open architecture for industry implementation
- An ISA separated into 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

RISC-V

- RISC-V was chosen to represent the fifth major RISC ISA design from UC Berkeley
  - RISC-I (1981)
  - RISC-II (1981)
  - SOAR (RISC-III) (1984)
  - SPUR (RISC-IV) (1988)
  - RISC-V (2011)

RISC-V base

- RISC-V consists of a mandatory base integer ISA and optional extensions
- The base specifies
  - registers (and their sizes)
  - memory and addressing
  - instructions (and their encoding)
  - control flow
  - logic (i.e., integer) manipulation
  - ancillaries
- The base alone can implement a simplified general-purpose computer, with full software support, including a general-purpose compiler.

RISC-V base

Name	Description	Version*	Status
RV32I	Base integer instruction set, 32-bit	2.1	Ratified
RV32E	Base integer instruction set (embedded), 32-bit, 16 registers	1.9	Draft
RV64I	Base integer instruction set, 64-bit	2.1	Ratified
RV128I	Base integer instruction set, 128-bit	1.7	Draft

*as of December 14, 2019

*The ISA modules marked Ratified have been ratified at this time.
*The modules marked Draft are expected to change before ratification.
*The modules marked Frozen are not expected to change significantly before being put up for ratification.

RISC-V extension

Name	Description	Version	Status
M	Standard extension for integer multiplication and division	2.0	Ratified
A	Standard extension for atomic instructions	2.1	Ratified
F	Standard extension for single-precision floating-point	2.2	Ratified
D	Standard extension for double-precision floating-point	2.2	Ratified
Zicsr	Control and Status register (CSR)	2.0	Ratified
Zifencei	Instruction-fetch fence	2.0	Ratified

The base, four extensions, CSR and instruction-fetch fence are collectively called RV32G

RISC-V extension

Name	Description	Version	Status
Q	Standard extension for quad-precision floating-point	2.2	Ratified
C	Standard extension for compressed instructions	2.0	Ratified
Counters		2.0	Draft
L	Standard extension for decimal floating-point	0.0	Draft
B	Standard extension for bit manipulation	0.0	Draft
J	Standard extension for dynamically translated languages	0.0	Draft
T	Standard extension for transactional memory	0.0	Draft
P	Standard extension for packed-SIMD instructions	0.2	Draft
V	Standard extension for vector operations	0.7	Draft
Zam	Standard extension for misaligned atomics	0.1	Draft
Ztso	Standard extension for total store ordering	0.1	Frozen

RISC-V registers

- 32 integer registers (x0 to x31)
- 32 floating point registers (f0 to f31)
- Size of register depends on the base ISA (i.e., 32-bit for RV32I; 64-bit for RV64I; 128-bit for RV128I)
- Floating point register (32-bit for single-precision; 64-bit for double-precision)

RISC-V memory access

- Load – store architecture
- Arithmetic and logic operations are register-register
- Memory is addressed as 8-bit bytes
- Little-endian order
- Words, up to the register size, can be accessed with load and store instructions

RISC-V instruction format

- 4 core instruction formats: (R-type/I-Type/S-type/U-type)
- 2 branch instruction formats: (B-type/J-type)
- Fixed instruction format: 32-bit

Observations:
- opcode: 7 bits
- rs1/rs2/rd: 5 bits
- rs1/rs2/rd are in the same locations
- imm are sign-extended and sign-bit in bit 31
- imm are always on the leftmost bits

	31-25	24-20	19-15	14-12	11-7	6-0
Register-register	funct7	rs2	rs1	funct3	rd	opcode

	31-20	19-15	14-12	11-7	6-0
Immediate	imm[11:0]	rs1	funct3	rd	opcode

	31-25	24-20	19-15	14-12	11-7	6-0
Store	imm[11:5]	rs2	rs1	funct3	imm[4:0]	opcode

	31-12	11-7	6-0
Upper immediate	imm[31:12]	rd	opcode

RISC-V instruction Format

	31	30-25	24-20	19-15	14-12	11-8	7	6-0
Branch	imm[12]	imm[10:5]	rs2	rs1	funct3	imm[4:1]	imm[11]	opcode

	31	30-21	20	19-12	11-7	6-0
jump	imm[20]	imm[10:1]	imm [11]	imm[19:12]	rd	opcode