VSDSquadron Mini Internship 2024

by

VLSI System Design

Documented By: Maaz Mahmood
KIET Group of Institutions
Department of Electronics and Communication Engineering

WHAT IS RISC-V?

- RISC-V is an open-source instruction set architecture (ISA) that allows developers to processors for specific applications.
- RISC-V is based on reduced instruction set computer principles and is the fifth generation of processors built on this concept.
- RISC-V can also be understood as an alternative processor technology which is free and open, meaning that it does not require you to purchase the license of RISC-V to use it.

INSTRUCTIONS FORMAT IN RISC-V

The instructions format of a processor is the way in which machine language instructions are structured and organized for a processor to execute. It is made up of series of 0s and 1s, each containing information about the location and operation of data.

There are 6 instruction formats in RISC-V:

1. R-format
2. I-format
3. S-format
4. B-format
5. U-format
6. J-format

Let's discuss each of the instruction formats in detail with examples.

R-type Instruction

In RV32, each instruction is of size 32 bits. In R-type instruction, R stands for register which means that operations are carried on the Registers and not on memory location. This instruction

type is used to execute various arithmetic and logical operations. The entire 32 bits instruction is divided into 6 fields as shown below.

31 25	24 20	19 15	14 12	11 7	6 0
func7	rs2	rs1	func3	rd	opcode

- The first field in the instruction format is known as **opcode,** also referred as operation code. The opcode is of length 7 bits and is used to determine the type of instruction format.
- The next subfield is known as **rd** field which is referred as Destination Register. The rd field is of length 5 bits and is used to store the final result of operation.
- The next subfield is **func3** also referred as function 3. Here the '3' represents the size of this field. This field tells the detail about the operation, i.e., the type of arithmetic and logical that is performed.
- The next two subfields are the source registers, **rs1 and rs2** each of length 5 bits. These are mainly used to store and manipulate the data during the execution of instructions.
- The last subfield is **func7** also referred as function 7. Here '7' represents the size of the field. The function of func7 field is same as that of func3 field.
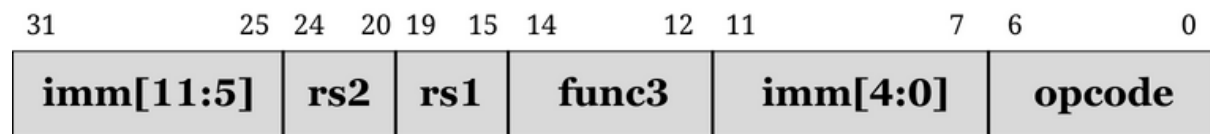
I-type Instruction

In RV32, each instruction is of size 32 bits. In I-type instruction, I stand for immediate which means that operations use Registers and Immediate value for their execution and are not related with memory location. This instruction type is used in immediate and load operations. The entire 32 bits instruction is divided into 5 fields as shown below.
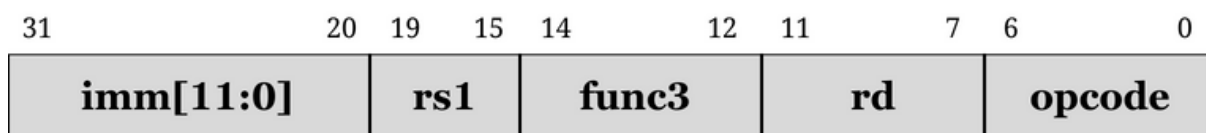
31 20	19 15	14 12	11 7	6 0
imm[11:0]	rs1	func3	rd	opcode

- The first field in the instruction format is known as **opcode,** also referred as operation code. The opcode is of length 7 bits and is used to determine the type of instruction format.
- The next subfield is known as **rd** field which is referred as Destination Register. The rd field is of length 5 bits and is used to store the final result of operation.
- The next subfield is **func3** also referred as function 3. Here the '3' represents the size of this field. This field tells the detail about the operation, i.e., the type of arithmetic and logical that is performed.
- The next subfield is the source registers, **rs1** of length 5 bits. It is mainly used to store and manipulate the data during the execution of instructions.
- The only difference between R-type and I-type is rs2 and func7 field of R-type has been replaced by 12-bits signed immediate, **imm[11:0]**
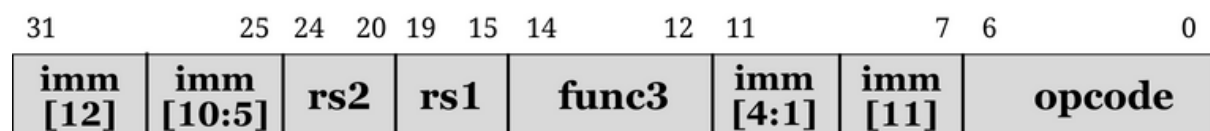
S-type Instruction

In RV32, each instruction is of size 32 bits. In S-type instruction, S stand for store which means it is store type instruction that helps to store the value of register into the memory. Mainly, this instruction type is used for store operations. The entire 32 bits instruction is divided into 6 fields as shown below.

31 25	24 20	19 15	14 12	11 7	6 0
imm[11:5]	rs2	rs1	func3	imm[4:0]	opcode

- The first field in the instruction format is known as **opcode,** also referred as operation code. The opcode is of length 7 bits and is used to determine the type of instruction format.
- S-type instructions encode a 12-bit signed immediate, with the top seven bits imm[11:5] in bits [31:25] of the instruction and the lower five bits imm[4:0] in bits [11:7] of the instruction.
- S-type instruction doesn't have rd fields which states that these instructions are not used to write value to a register, but to write/store a value to a memory.
- The value to be stored is defined in **rs1** field and address to which we have to store this value is calculated using **rs1 and immediate** field. The width of the operation and types of instruction is defined by **func3,** it can be a word, half-word or byte.

B-type Instruction

In RV32, each instruction is of size 32 bits. In B-type instruction, B stand for branching which means it is mainly used for branching based on certain conditions. The entire 32 bits instruction is divided into 8 fields as shown below.

31	25	24 20	19 15	14 12	11	7	6 0
imm [12]	imm [10:5]	rs2	rs1	func3	imm [4:1]	imm [11]	opcode

- The first field in the instruction format is known as **opcode,** also referred as operation code. The opcode is of length 7 bits and is used to determine the type of instruction format.
- B-type instructions encode a 12-bit signed immediate, with the most significant bit imm[12] in bit [31] of the instruction, six bits imm[10:5] in bits [25:30] of the instruction, four bits imm[4:1] in bits [11:8] and one bit imm[11] on bit[7].
- There are two source registers **rs1 and rs2** on which various operations are performed based on certain conditions, and those conditions are defined by **func3** field.
- After performing operations on the source register based on the conditions, it is evaluated that if the condition is true, Program Counter value gets updated by PC = Present PC Value + Immediate Value, and if the condition is false then PC will be given as PC = Present PC value + 4 bytes, which states that PC will move to next instruction set.
- RV32 instructions are word-aligned, which means that address is always defined in the multiple of 4 bytes.
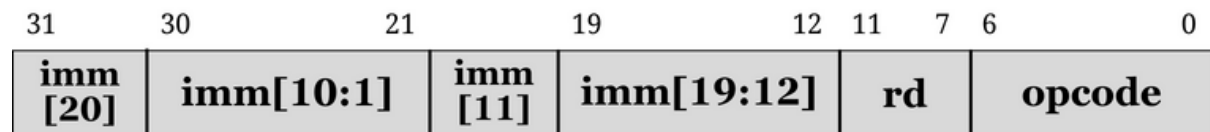
U-type Instruction

In RV32, each instruction is of size 32 bits. In U-type instruction, U stand for Upper Immediate instructions which means it is simply used to transfer the immediate data into the destination register. The entire 32 bits instruction is divided into 3 fields as shown below.

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

- The first field in the instruction format is known as **opcode,** also referred as operation code. The opcode is of length 7 bits and is used to determine the type of instruction format.
- The U-type instruction only consists of two instructions, i.e., LUI and AUIPC.
- For Example, lets take the instruction **lui rd, imm** and understand this instruction.
  lui x15, 0x13579: This instruction will be executed and the immediate value 0x13579 will be written in the MSB of the rd x15, and it will look like x15 = 0x13579000

J-type Instruction

In RV32, each instruction is of size 32 bits. In U-type instruction, J stand for jump, which means that this instruction format is used to implement jump type instruction. The entire 32 bits instruction is divided into 6 fields as shown below.

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

- The first field in the instruction format is known as **opcode,** also referred as operation code. The opcode is of length 7 bits and is used to determine the type of instruction format.
- The J-type instruction only consists of single instruction, JAL.
- J-type instruction encode 20 bits signed immediate which is divided into four fields.
- The J-type instructions are often used to perform jump to the desired memory location. The address of the desired memory location is defined in the instruction. These instructions are also used to implement loops.