RISC-V INSTRUCTION TYPE

INTRODUCTION:

RISC-V is a new instruction set architecture (ISA) designed to support computer architecture research and education. Our goals in defining RISC-V include: Provide a realistic but open ISA that captures important details of commercial general purpose ISA designs and that is suitable for direct hardware implementation. Provide a small but complete base ISA that avoids over-architecting for a particular microarchitecture style (e.g., micro coded, in-order, decoupled, out-of-order) or implementation technology (e.g., full-custom, ASIC, FPGA), but which allows efficient implementation in any of these. Support both 32-bit and 64-bit address space variants for applications, operating system kernels, and hardware implementations. Support highly-parallel multicore or manycore implementations, including heterogeneous multiprocessors. Support an efficient dense instruction encoding with variable-length instructions, improving performance and reducing energy and code size. Support the revised 2008 IEEE 754 floating-point standard. Be fully virtualizable. Be simple to subset for educational purposes and to reduce complexity of bringing up new implementations. Support experimentation with user-level ISA extensions and specialized variants. Support independent experimentation with new supervisor-level ISA designs.

INSTRUCTION TYPES:

1.Arithmetic and Logic Instructions
2.Control Flow Instructions
3.Memory Instructions

R-Type Instructions:

R-type instructions in RISC-V are used for register-to-register operations, such as arithmetic and logic operations .Uses only registers as sources and destinations. R-type instructions have two source registers, rs1 and rs2 and a destination register ,rd

R-TYPE:

R-type instructions specify two source registers (rs1 and rs2) and a destination register (rd). The funct10 field is an additional opcode field

R4-TYPE:

R-type instructions specify two source registers (rs1 and rs2) and a destination register (rd). The funct10 field is an additional opcode field

I-Type Instructions:

Uses a register and an immediate value. The immediate value that's interpreted as either two's complement or an unsigned integer depending on the opcode.

Rs1 is the source register.

funct3 is the additional opcode field.

rd is the destination register.

opcode is the opcode.

I-type instructions specify one source register (rs1) and a destination register (rd). The second source operand is a sign-extended 12-bit immediate, encoded contiguously in bits 2110. The funct3 field is a second opcode field.

S-Type Instructions:

The characteristic of S-type instruction is that there is no rd register. In this type of instruction, the immediate is divided into two parts, the first part is in bit11-5, and the second part is in bit4-0. Instructions that store a value from a register into memory are implemented using this s-type format.

B-Type Instructions:

The branch instructions that conditionally transfer control to a newly specified instruction address are the type of instructions that use the b-type format.

B-type instructions specify two source registers (rs1 and rs2) and a third source operand encoded as a sign-extended 12-bit immediate. The immediate is encoded as the concatenation of the upper 5 bits in bits 3127, and a lower 7 bits in bits 1610. The funct3 eld is a second opcode field.

U-Type Instructions:

For performing instructions that load 20-Bit immediate value into a register, use the u-type format.

J-Type Instructions:

The j-type format is used to implement jump type instructions that unconditionally transfer control to a new instruction address. J-type instructions encode a 25-bit jump target address as a PC-relative o set. The 25-bit immediate value is shifted left one bit and added to the current PC to form the target address.