

1. CISC, 复杂指令集计算机。单个指令完成的任务量大且功能复杂, 指令长度灵活, 对编译器和程序存储空间的要求较低, 缺点是硬件设计复杂, 测试验证难度较高。

RISC, 精简指令集计算机。单个指令完成的任务量少且功能单一, 指令长度相对固定, 如 RISC-V, MIPS 是常见架构。硬件设计较为简单, 适合利用流水线提升性能, 对编译器设计的要求较高, 程序的代码密度较低。

2. RISC-V 的基本指令集包括 RV32I 和 RV64I, 分别用于 32 位和 64 位系统。

1. RV32/64G: 通用指令集扩展, 包括原子操作、浮点数操作和压缩指令等。常用于运行复杂的应用程序, 如机器学习和计算机视觉等。

2. RV32/64M: 乘法和除法指令集扩展, 支持整数乘除和累加运算。可用于数值计算和信号处理等需要大量乘法和除法运算的应用领域。

3. RV32/64A: 原子指令集扩展, 该指令集扩展可用于多线程编程、同步和互斥操作等应用领域。

4. RV32/64C: 压缩指令集扩展。可用于内存受限的嵌入式系统和无线传感器网络等应用领域。

5. RV32/64F 和 RV32/64D: 浮点指令集扩展, 该指令集扩展可用于科学计算、图像处理和信号处理等应用领域。

4. (1) RV32I 中 add 和 RV64I 中 addw 的 opcode 分别为 0110011, 0111011 不同。RV32I 中 add 和 RV64I 中 add 的 opcode 均为 0110011 相同。

原因: add 指令操作数相同; 处理不同的指令集时不需要修改代码, 提高兼容性。  
不同: addw 指令为 32 位整数加法, 而 RV64I 的寄存器有 64 位, 设计特殊的 opcode 使得避免对 64 位整数进行不必要的操作, 提高处理器效率。

(2) 不需要。低 32 位计算结果存储到目标寄存器后, 其高 32 位已清零, 无须扩展。



5. HINT指令空间用于定义一些对处理器性能、功耗和调试等方面产生提示作用的指令。这些指令在处理器的行为上没有实际影响,但可以向处理器发出提示,以便处理器在运行时进行一些优化或调试操作。

6.  $a_2 = -3$      $a_3 = 1$

若被除数和除数符号相同,则商为正数

若符号不同,则商为负数。

余数为正。

11. (1) 偏移量寻址。

(2) 偏移量寻址。

(3) 立即数寻址

(4) 寄存器直接寻址

(5) 偏移量寻址。

