

1. CISC架构的优势: ①指令集比较丰富, 可以完成很多复杂的操作 ②单个指令可以完成这个操作, 提高了计算机效率 ③可以使用更少的指令完成更多任务, 减少存储器开销.

CISC架构的劣势: ①指令复杂, 需要更多芯片面积来实现 ②指令执行时间不一定稳定, 影响处理器性能不易优化 ③CISC处理器的设计和制造成本较高

RISC架构优势: ①指令简单, 执行时间稳定, 方便处理器性能优化 ②指令集精简, 易于设计和制造 ③通常比CISC处理器具有更高时钟频率, 因此有更快的速度.

RISC架构劣势: ①每个操作都要单独指令, 所以需要更多的指令来完成复杂的任务. ②指令集比较简单, 不能完成一些复杂的操作 ③需要更多指令来完成复杂任务, 因此需要更多存储器.

2. RISC-V的基本指令集由RV32I和RV64I组成, 分别适用于32位和64位的RISC-V处理器. 从基本指令集包括一组最小的指令, 例如加载存储指令、算术指令、条件分支指令、无条件跳转指令等.

①RV32M和RV64M: 添加了乘除指令, 提高处理器运算能力. 这些指令在数字信号处理、图像处理、加密算法等领域得到广泛应用.

②RV32F和RV64F: 添加了浮点运算指令, 包括浮点加减、乘除、比较等指令, 可以处理浮点数运算. 这些指令在科学计算、图形处理、数据分析和信号处理等领域得到广泛应用.

③RV32D和RV64D: 添加了双精度浮点运算指令, 可以处理更高精度的浮点数运算. 这些指令在需要高精度计算的领域, 如天文、大分子、工程计算等领域得到广泛应用.

④RV32G和RV64G: 这些扩展指令集将RV32I和RV64I与RV32M、RV64M、RV32F、RV64F以及RV32D、RV64D指令结合起来, 提供了一个完整的指令集, 支持基本的整数和浮点数运算.

⑤RV32A和RV64A: 添加原子操作指令, 可以支持多线程程序的同步操作. 这些指令在并行计算、操作系统、数据库等领域得到广泛应用.

①不同 ②不同

③这样的设计是为了支持RV64I架构中的32位操作. 因为RV64I中的寄存器是64位的, 但是在实际编程中, 经常需要对32位数据进行操作. 采用RV32I中的add指令对32位数据进行操作可能会导致代码变得不兼容, 因为RV64I中的寄存器不支持32位操作数. 因此, RV64I引入了一种新的addw指令, 该指令专门用于对32位数据进行加法操作, 以保持与RV32I中操作数的一致性.

④需要. addw指令将两个常量的32位整数相加, 并将结果截断为32位, 然后将结果符号扩展为64位, 并将结果存入目标寄存器. addw指令, 目标寄存器是64位, 但RV64I中所有寄存器都是64位的, 所以即使目标寄存器是64位的, 也需要符号扩展.

5. RV32I为HINT指令保留了很大一片编码空间。HINT指令空间。HINT指令通常用于向微架构传达性能提示，并且就像NOP一样，除推动PC以及任何可用性能计数器外，并不改变任何结构可见的状态。

11. (1) 偏移寻址

(2) 寄存器直接寻址

(3) 立即数寻址

(4) 寄存器直接寻址

(5) 寄存器间接寻址

6. a_2 寄存器中值为3, a_3 寄存器中值为1

如果是不符号运算，则商和余数都是正数，如果是有符号运算，则商的正负与除数、除数、是否同号相关，而余数与除数同号。