

T₁

CISC主要优点:

- ①指令丰富, 功能强大
- ②寻址方式灵活
- ③以微程序控制器为核心, 指令存储器与数据存储器共享同一个物理存储空间, 性能强大.

CISC主要缺点:

- ①指令使用率不均衡.
- ②不利于采用先进结构提高性能
- ③结构复杂不利于VLSI实现

RISC结构主要优点:

- ①结构简单, 易于设计
- ②指令精简, 使用率均衡.
- ③程序执行效率高

RISC结构主要缺点:

- ①指令数较少, 功能不及CISC强大
- ②寻址方式不够灵活

T₂

基本指令集: 算术和逻辑指令、数据转移指令、控制流指令、系统指令、浮点运算指令、十进制指令、字符串指令、图形指令

RV32I/RV64I: 包括基本的算术、逻辑和分支跳转指令等, 支持32位和64位的处理器架构, 适用于广泛的通用计算领域。

RV32M/RV64M: RV32M/RV64M是RISC-V的乘法扩展指令集, 包括了基本的算术、逻辑和乘法指令等, 适用于数字信号处理和图像处理等需要大量乘法运算的领域。

RV32A/RV64A: 是RISC-V的原子操作扩展指令集, 包括了原子操作指令, 适用于多线程同步和互斥操作, 广泛应用于并行计算和多处理系统等领域。

RV32F/RV64F/RV32D/RV64D: 浮点扩展指令集, 包括了单精度和双精度浮点运算指令, 适用于科学计算、信号处理、图形等需要高精度运算的领域。

RV32C/RV64C: RV32C/RV64C是RISC-V的压缩指令扩展指令集,通过将常用的指令压缩为16位指令来节省存储空间,适用于嵌入式系统和移动设备等对存储空间有限的场合。

T4.

- 1) opcode相同, RV32I中^{和RV64I} 'addw'的opcode是0110011, RV32I和RV64I中"add"具有相同的opcode "0110011"。

这种设计的目的是为了提高指令集的灵活性和可扩展性,降低指令集的复杂度,简化CPU与编译器的实现,方便代码的移植。

- 2) 需要。addw和addiw的加法操作都的结果都是存储在目标寄存器的低32位中。而RV64I中所有寄存器都是64位宽度的,因此目标寄存器的高32位将被清零,因此需进行符号扩展才能用于后续的64位计算。

T5.

HINT指令并不执行任何实际的操作,而是用于向处理器发出一些提示和建议,以帮助更好地执行程序。同时HINT指令空间~~为~~^在为程序员和编译器提供一些额外的控制和优化手段,帮助程序员和编译器更好地控制程序的执行顺序和内存访问。

T6.

a_2 是-3, a_3 是1。除法结果的符号由被除数和除数共同确定,余数的符号与被除数相同。

T11.

1) 相对寻址、2) 立即数寻址、3) 立即数寻址 4) 寄存器寻址 5) 相对寻址