

1. CISC: 优: 对编译器和程序存储空间的要求较低  
劣: 硬件设计复杂, 测试验证难度较高  
RISC: 优: 硬件设计较为简单, 适合利用流水线提升性能  
劣: 对编译器设计的要求较高, 程序的代码密度较低

2. RISC-V中的基本指令集是 RV32I

常见的 RISC-V 标准扩展指令集:

~~RV64I~~ RV64I: 是 RV32I 的扩展, 提供了更大的寄存器和内存地址空间

RV32M: 32位乘法和除法指令集, 提供了整数乘法和除法操作, 以及相关的移位和控制操作

RV64M: 64位乘除法指令集, 是 RV32M 的扩展, 提供了更大的寄存器和内存地址空间

RV32F: 32位浮点指令集, 提供了单精度浮点运算和相关的控制操作

RV32D: 32位双精度浮点指令集, 提供了双精度浮点运算和相关的控制操作

4. (1) RV32I 中的 add 和 RV64I 中的 addw 具有相同的指令操作数

RV32I 中的 add 和 RV64I 中的 add 不具有相同的指令操作数

这样设计, ~~让~~ 让相同功能的指令由同一个指令操作数表示, 更加准确

(2) 不需要, addw 和 addiw 的结果是带符号的 32 位数, 直接扩展到 64 即可。

5. HINT 指令空间是一组用于提示处理器执行优化的指令, <sup>位</sup>

这些指令可以用于在不改变语义的情况下, 向处理器提供关于代码执行的信息, 以便处理器可以根据这些信息来优化代码执行的效率。

如: HINT\_NOP, HINT\_BRANCH, HINT\_RETURN, HINT\_CALL

6.  $a_2$  的值是 -3,  $a_3$  的值是 1

符号规定: 对于  $\text{div}/\text{divu}$ , 商的符号就是正常的计算得到,

如:  $-16/5 = -3$ ,  $-16/5 = -3$ ,  $16/5 = 3$

对于  $\text{rem}$ , 余数的符号和被除数保持一致

11. (1) 偏移量寻址

(2) 寄存器间接寻址

(3) 立即数寻址

(4) 寄存器直接寻址

(5) 内存直接寻址