

#### 四周

1.

CISC

RISC

优

指令集成高, 程序员工作量低

指令条数少

可直接访问内存

指令集简单易实现, 速度快

冗余低, 节约空间

控制简化

劣

硬件开销大

指令冗余, 浪费存储

功耗大

指令条数多

不可直接访问内存

编程复杂

基本: { 整数寄存器, 寄存器操作: ADD, SUB, AND, OR, XOR, SLL, SRL, SRA  
加载/存储: LOAD, STORE  
分支: BRANCH, JAL, JALR  
特权: MRET, ECALL, EBREAK

扩展: M: 整数乘法

A: 原子指令

C: 压缩指令

FD: 浮点操作

V: 向量操作

可提高由32位系统扩展至64位时的兼容性

不需要, 在 `addw` 与 `addiw` 中已经进行符号扩展

⑤. Hint: 用于给处理器提供提示信息的指令, 信息包含是否可以重排指令

是否在等待时可以进行其它任务, 是否休眠等

作用: 提高资源利用率, 提高性能与能效

$$6. a_2 = a_0 / a_1 = -3$$

$$a_3 = a_0 / a_1 = 1$$

对于 `div` 与 `rem`, 均视操作数为有符号数

若需视操作数为无符号数, 可用 `divu` 与 `remu`

11. ① 偏移量寻址. 寻址为:  $pc + 0x88$

② 基址+偏移量寻址.  $ra + 0$

③ 立即数寻址.  $a_1$  与指令中的4相加

④ 寄存器直接寻址.  $a_1$  与  $a_2$

⑤ 基址+偏移量寻址.  $sp + 16$

① 不相同

② 相同

`add`: 均为对应位数的加法, 无视溢出

`addw`: RV64I中, 对32位加法的溢出截断后, 有符号扩展为64位

