

1.

CISC: 优点: 对编译器和程序存储空间的要求较低

缺点: 硬件设计复杂, 测试验证难度较高

RISC: 优点: 硬件设计较为简单, 适合利用流水线提升性能,

缺点: 对编译器设计的要求较高, 程序的代码密度较低

2.

1) RISC-V 中基本指令集只包括 4 多条, 包括整数指令, 用于同步、系统调用和什数器操作的其它指令, 其中大多指令是用于实现基本的控制流、算术和逻辑运算等功能。

以及求余等

2) RV32/64M: 包括乘除法指令, 可以加快计算速度, 也可以用于图形处理等场景

RV32/64F: 包括单精度浮点运算指令, 如加减乘除开方等, 常用于图像处理, 信号处理等场景

RV32/64D: 包括双精度浮点运算指令, 如加减乘除开方, 相比于 F 扩展提供了更高的精度

RV32/64C: 可以提供压缩指令, 将指令压缩为 16 位, 减少存储和带宽开销, 解压后运行, 并用于嵌入式系统、移动设备等领域

RV64J: 包括硬件支持的跳转指令, 用于实现更高效的跳转和异常处理

4.

1) RV32I 中的 `add` 与 RV64I 中的 `addw` 有不同的操作数, RV32I 中的 `add` 也与 RV64I 中的 `add` 具有不同的操作数, RV64I 中的 `add` 完成 64 位整数加法, 功能与另两个不同; 而 RV32I 中的 `add` 和 RV64I 中的 `addw` 虽然都是实现 32 位加法, 但由于 RV64I 需要对 RV32I 向后兼容, 为避免冲突和混淆, 设计了新的操作码

2) 仍需要进行额外的符号扩展。因为即使 `addw` 和 `addiw` 最后会执行符号扩展的操作, 但计算结果是 32 位的, 所以需要让 32 位的数据参与 64 位计算, 依然



要进一步符号扩展, 将32位数据扩展为64位

5.

RISC-V 标头指令集中的 HINT 指令空间包含了一组可以提示处理器关于指令流、数据流等信息的指令。

HINT 指令可以通过分支预测、缓存预取、数据依赖等方式, 提供相应的信息来帮助处理器优化指令执行流程, 从而优化程序性能并降低功耗。

6.

a2 的值为3

a3 的值为1

RISC-V M 指令集中, `div`、`divw`、`rem`、`remw` 是对有符号数运算;

`divu`、`divuw`、`remu`、`remuw` 是对无符号数运算

11.

写出以下指令使用的寻址模式

1) `jal ra, 0x88` 偏移量寻址

2) `jalr x0, ra, 0` 内存直接寻址

3) `addi a0, a1, 4` 立即数

4) `mul a0, a1, a2` 寄存器直接寻址

5) `lwl a4, 16(sp)` 偏移量寻址

