



1. 简要分析 CISC 和 RISC 架构各自的优势和劣势。

CISC 优点: 具有复杂的功能和结构的指令，增强了硬件的计算能力
降低了编译的难度和目标代码总量

劣势: 随着指令的增长，支持新指令并兼容旧指令使硬件结构复杂化
设计与调试难度攀升

RISC 优点: 硬件设计较为简单，适合利用流水线提升性能
缺点: 对编译器要求较高，程序代码密度较低

2. RISC-V 中的基本指令集是什么？列举五个常见的 RISC-V 标准扩展指令并简要说明它们的作用和应用范围。

基本指令集: RV32I

扩展指令集: RV32M 包含乘法除法取模、求余指令
RV32F 单精度浮点指令
RV32D 双精度浮点指令
RV32C 压缩指令，用于改善程序大小
RV32A 原子操作指令，如比较并交换、读写.

4. 阅读 RISC-V 规范以回答以下问题：

- 1) RV32I 中的 add 指令和 RV64I 中的 addw 指令均为 32 位整型加法指令，它们是否具有相同的指令操作数 (opcode)？此外，RV32I 中的 add 指令和 RV64I 中的 add 指令是否具有相同的指令操作数 (opcode)？试分析为什么采取这样的设计。
- 2) 在 RV64I 中，addw 和 addiw 指令的目标寄存器中存放的 32 位计算结果是否需要进行额外的符号扩展才能用于后续 64 位计算？请说明理由。

1) RV32I add 0110011 [不同]] 相同
RV64I addw 0111011 [相同]
RV64I add 0110011]

原因: RV32I 是 RV64I 的子集，RV64I 是 RV32I 的超集，
故这样设计易于扩展指令集。

2) 不用进行额外的符号扩展
addw 和 addiw 指令结果裁断为 32 位的有符号数。
但 32 位已经进行过符号扩展到 64 位，并写入了目标寄存器

5. 什么是 RISC-V 的 I 标准指令集中存在的 HINT 指令空间？它有什么作用？

概念：一组用于调试和性能分析的指令的集合。

、这些指令不进行任何操作，而是将一些信息

传递反端给处理器或调试器，或触发一些硬件行为。

作用 使程序员和调试器更好地分析和调试程序，提高性能，在处理器层面上进行更好地协同处理。

6. 考虑如下指令序列：

div a2,a0,a1

rem a3,a0,a1

假设寄存器 a0 和 a1 的初始值分别为 16 和 -5，则上述指令序列执行完成后 a2 和 a3 寄存器中的值分别是多少？简要说明 RISC-V 的 M 标准指令集中对除法和余数指令的符号规定。

$$\begin{array}{ll} \text{div } a_2, a_0, a_1 & \text{Reg}[a_2] = \text{Reg}[a_0] \div \text{Reg}[a_1] = 16 \div (-5) = -3 \\ \text{rem } a_3, a_0, a_1 & \text{Reg}[a_3] = 16 \% (-5) = 1 \end{array}$$

补充：RISC-V M 指令集除法和取余分为有符号与无符号两部分

无符号指令为 divu 和 remu

有符号除法中，结果遵循负负得正

无符号除法，结果与被除数符号相同

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 立即数寻址

4) mul a0,a1,a2 寄存器寻址

5) ld a4,16(sp) 偏移量寻址

