

2-1. 简要分析 CISC 和 RISC 各自的优劣。

答: CISC: 优: 单个指令完成的任务量大且功能复杂, 指令长度灵活, 从而对编译器要求低, 代码密度高, 对程序存储空间要求较低。

缺点: 硬件设计复杂, 测试验证难度较高。

RISC: 优: 指令长度固定, 执行阶段周期固定, 硬件设计较为简单且适合利用流水线提高性能。

缺点: 对编译器设计的要求较高, 且程序的代码密度较低, 对程序存储空间有要求。

2-2. RISC-V 中基本指令集是什么? 列举 5 个常见的 RISC-V 标准扩展指令集并简要说明它们的作用和应用范围。

答: 基本指令集是 RV32I

常见标准扩展: 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, 单指令长度 16 位, 主要用于改善程序大小

RV32A: 原子操作指令, 如比较并交换, 读改写等指令。

2-4. (1) RV32I 中 add 指令和 RV64I 中的 addw 指令均为 32 位整型加法指令, 它们是否具有相同指令操作数? (2) RV32I 中 add 和 RV64I 中 add 指令是否具有相同指令操作数? (3) 为什么采取这样的设计?

答: ① 不同, RV32I: add: 0110011

RV64I: addw: 0111011

② 相同, 都是 0110011

③ 为了兼容 RV32I 中的 add 指令 (RV32I 是 RV64I 子集), 同时区分 32 位加法与 64 位加法。

(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

No.

Date

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。

答：需要，因为进行符号扩展后才能保证数据在后续64位计算中能正确表示其原本的正负性和十进制大小。

2-5. 什么是RISC-V标准指令集中的HINT空间？它有什么作用？

答：RISC-V的I标准指令集中的HINT指令空间是一组用于调试和性能分析的指令。这些指令不执行任何操作，而是将一些信息反馈给处理器或调试器，或触发一些硬件行为。

HINT指令能够允许程序员和调试器更好地分析和调试程序，同时能提高性能，在处理器层面上进行更好的协同处理。

2-6. 写出下列指令序列。

div a2, a0, a1

rem a3, a0, a1

设 a0、a1 初始值分别为 16、-5，则上述指令完成后 a2、a3 的值为？

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。

答：div a2, a0, a1: $a2 = a0 \div a1 = -3$

rem a3, a0, a1: $a3 = a0 \% a1 = 1$

RISC-V 的标准指令集中，除法和取余数指令都分为有符号和无符号。

无符号版为 divu、remu，将数都当作无符号数运算。

而有符号的除法中，结果遵循负负得正的法则，

有符号的取余中，结果与被除数同号。

2-11 写出以下指令使用的寻址模式。

(1) jal ra, 0x88 立即数寻址

(2) jalr x0, ra, 0 立即数寻址

(3) addi a0, a1, 4 立即数寻址

(4) mul a0, a1, a2 寄存器直接寻址

(5) ld a4, 16(sp) 偏移量寻址。