

1. 分析 RISC 和 CISC 架构优、劣势

CISC: 优点: 对编译器和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. RISC-V 基本指令集有

RV32I 32位整数指令集, 使用32位寄存器的32位整数指令.

RV32E RV32I的子集, 用于小型嵌入式场景, 使用16个寄存器的基本32位指令.

RV64I 使用64位寄存器的基本64位整数指令.

扩展指令集:

M 扩展了整数乘法和除法指令.

A 扩展了并发操作中的原子指令.

F 扩展了IEEE标准单精度浮点数运算指令, 增加了32个32位浮点寄存器.

D 扩展了IEEE标准双精度浮点数运算指令, 增加了32个64位浮点寄存器.

B 扩展了四精度浮点数运算指令.

4. (1) RV32I中 add 与 RV64I中 addw 是否有相同指令操作数? RV32I中 add 与 RV64I中 add 是否有相同 opcode? 分析为什么采取这种设计.

RV32I中 add 与 RV64I中 addw 具有相同操作数, 而 RV32I中 add 与 RV64I中 add 不具有相同 opcode.

使用RV64I相对RV32I的一个优点是寻址大于4GB的地址空间, 有更大的地址空间和能够在64位寄存器中进行64位计算.

(2) RV64I中, addw 和 addiw 指令目标寄存器中存放的32位计算结果, 不需要进行额外扩展进行后续64位计算, addw 和 addiw 只进行低32位计算, 高32位不变, 不需要额外扩展.

5. HINT 指令空间表示该操作码为未来的标准微体系结构保留. HINT 指令设计的目的支持将来添加可能影响性能但不会影响体系结构状态的微体系结构.

6. a_2 为 -3, a_3 为 1.

除法符号由除数、被除数共同决定, 商的符号由被除数决定.

11. 1) jal ra, 0x88 偏移量寻址

2) jalr x0, ra, 0 寄存器直接寻址

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.