

3.14

1. 答: CISC 优势: 对编译器和程序存储空间要求低, 劣势是硬件设计复杂、测试验证难度较高。~~代码编译难度大~~

RISC 优势: 硬件设计较为简单, 适合利用流水线提升性能; 劣势: 对编译器设计的要求较高, 代码密度低。

2. RISC-V 中的基本指令集为 RV32I、RV32E 和 RV64I, 即使用 32 位寄存器的基本 32 位整数指令、只使用 16 个寄存器的基本 32 位指令和只使用 64 位寄存器的基本 64 位整数指令。

扩展指令集: ① M: 扩展整数^{乘法和}和除法指令。

② A: 扩展并发操作中的原子指令

③ 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

④ 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

⑤ Q: 扩展了四精度浮点数运算指令。

4. (1) RV32I中 add 操作码与RV64I中 add 操作码均为 0110011, 而 RV64I中 addw 操作码为 0111011, 因为其比 add 指令增加了对符号扩展的操作。这样设计更方便有符号数运算。

(2) 不需要, 因为 addw 和 addiw 指令在运算完成后已经扩展为 64 位, 无需后续额外扩展。

5. HINT 指令空间是支持未来增加微指令结构提示, 用于长度超过 16 位的指令的标记, 将该操作保留给未来微指令结构提示。这样可能影响性能, 但不影响体系结构状态, 保留原有^{操作码}指令, 并交由未来微指令结构指令执行。

6. a_2 寄存器为 a_3 ; a_3 寄存器为 1

div 指令将 $x[rs1]$ 除以 $x[rs2]$ 值, 向 0 舍入, 将其视为二进制补码, 将商写入 $x[rd]$ 即 a_2 , 其 div 为有符号除法, divu 无符号;

rem 指令取 $x[rs1]$ 除 $x[rs2]$ 值, 向 0 舍入后, 都视为 2 的补码, 余数写入 $x[rd]$, rem 为有符号取余, remu 为无符号取余。

11. (1) jal, ra, 0x88 直接寻址
(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 寄存器寻址
(5) ld a4, 16(sp) 偏移量寻址