

1. CISC 优点: 对编译器和程序存储要求低, 代码密度高。

缺点: 硬件设计复杂, 测试验证难度高, 指令利用率低, ^{占用}存储空间大。

RISC: 优点: 硬件设计简单, 适合流水线形式; 指令简短, 利用率高

缺点: 对编译器要求高, 程序代码密度低

2. 基本指令集有 RV32I, RV32E, RV64I. 包含基本指令。

标准扩展指令集: M (整数乘/除) A (并发操作中的原子指令)

F (扩展单精度浮点运算指令, 增加32个32位浮点寄存器)

D (扩展双精度浮点运算指令, 增加32个64位浮点寄存器)

Q (扩展四精度浮点运算指令)

4. (1) 操作数不同, $OP(ADDW)=0111011$, $OP(ADD)=0110011$, RV32I与RV64I的ADD指令操作数相同, 因ADDW的操作并不是让两个操作数直接相加输出结果, 需要进一步将结果高32位截断再扩展符号位, 操作内容不等同于ADD, 需加以区分。

(2) 不需要, ADDW与ADDW指令已经进行了符号位扩展。

5. HINT指令是为了支持将来增加的微架构的提示指令。执行时不会对架构状态产生任何影响。

6. $a_2 = -3$ $a_3 = 1$

DIV. 带符号除法. 对RV32I与RV64I通用, 结果(商)向0舍入

DIVU. 无符号除法. 向0舍入

REM 带符号取余, 结果为被除数 - (除数 \times 带符号商)

REMU 无符号取余, 结果为被除数 - (除数 \times 不带符号商)

DIVW: 仅 RV64I, 取商的低32位, 并符号位扩展为64位

DIVUW: 仅 RV64I, 取商的低32位, 进行无符号扩展为64位

REMW/REMUW: 对余数取低32位, 并进行有/无符号扩展为64位.

ll. jal ra, 0x88 偏移量寻址

jalr x0, ra, 0. 寄存器间接寻址

addi a0, a1, 4. 立即数寻址

mul a0, a1, a2 寄存器寻址

ld a4, 16(sp) 偏移量寻址