

## 第四周

1. Q: 比较 RISC 和 CISC 的优劣

A: CISC 优势: 指令灵活, 完成同样操作所需的指令数少, 代码密度高;

对编译器和存储设备的要求较低;

劣势: 指令复杂, 硬件架构设计难度大;

RISC 优势: 指令简单, 方便硬件设计和优化;

劣势: 代码密度低。

2. RISC-V 的基本指令集: RV32I, RV32E, RV64I, RV128I

扩展指令集: M 整数乘法与除法指令

F 单精度浮点数运算

D 双精度浮点数指令

A 原子操作指令

C 压缩指令

4. 1) RV32I 的 add 指令与 RV64I 的 addw 具有不同的操作数,

但前者与 RV64I 的 add 具有相同的 Opcode.

采取这样的设计, 是因为两者的 add 均为整个字长数据的加法, 同样的 Opcode 可以方便硬件设计.

2) 不需要. 32 位的计算结果会在高位补符号位, 可作为 64 位数参与运算.

5. I 标准指令集中的 HINT 指令空间: RISC-V 预留的编码空间, 用于向微架构传达性能提示; 除了推动 PC 递增外, 并不改变体系结构其余部分的状态 (类似 Nop).

6.  $a_0, a_1$  初值为 16, -5.

`div a2, a0, a1`

`rem a3, a0, a1`

此后  $a_2$  值为 -3,  $a_3$  的值为 1.

规定: 整数除法的结果为“近 0 取整”

整数取模的结果, 符号由被除数的符号决定,

$$\text{即: } x \% y = x - (x / y) * y$$

11. Q: 寻址模式?

A: (1) `jal ra, 0x88` 直接寻址

(2) `jalr x0, ra, 0` 寄存器间接寻址

(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