

2023. 3. 14

1. CISC 架构单个指令完成的任务量大且功能复杂、指令长度灵活  
其优点为对编译器和程序存储空间的要求较低

缺点是硬件设计复杂，测试验证难度较高

RISC 架构单个指令完成的任务量少且功能单一，指令长度相对固定

优点是硬件设计较为简单，适合利用流水线提升性能

缺点是对编译器设计的要求较高，程序代码密度较低

2. RISC-V 基本指令集：RV32I、RV32E、RV64I、RV64E 基础指令集

① M 杜绝扩展指令集 包含对两整数做乘除法的指令 从而简化低端硬件实现

② A 标准 ... 包含对寄存器执行原子性读写以支持不同线程之间的同步

③ F 增加 32 个 32 位的浮点寄存器和一个浮点控制状态寄存器 csr

④ D 增加双精度浮点计算的指令

⑤ Q 增加四精度浮点计算的指令

4. 1) RV32I add 指令操作数是 0b10011

RV64I addw 指令操作数是 0b11011 与 RV32I add 不一样

RV64I addl 指令操作数是 0110011 与 RV32I add - 样

因为 RV64I 是 RV32I 的超集，且 RV64I 中寄存器扩展到 64 bits

RV64I 包括 RV32I 的所有指令 addw 是在 RV64I 中新增的指令。

而为了让 add 指令能普适于 RV32I、RV64I 指令集且与 addw 兼容

采用 add 指令操作数一致而与 addw 不一致

2) 不需要。RV64I 中 addw 和 addl 指令最终会把结果截断为

32 位并把符号位扩展的结果写入 x[rd]，所以不需要额外的符号扩展。

5 HINT 指令是提示指令，通常用于向微架构传达性能提示，且其除推动 PC 以及任何可用性能计数器外，并不改变任何体系结构可见的状态。具体实现可选择忽略这些提示编码。

6 div a2, a0, a1  $\Rightarrow \text{reg}[a_3] = -3$

rem a3, a0, a1  $\Rightarrow \text{reg}[a_3] = 1$

DIV 除法：div rd, rs<sub>1</sub>, rs<sub>2</sub> 即用  $x[rs_1]$  的值除以  $x[rs_2]$  的值向零舍入（正数向下舍入，负数向上舍入）  
将这些数视为二进制补码输入  $x[rd]$ 。

REM 取余 rem rd, rs<sub>1</sub>, rs<sub>2</sub>  $x[rs_1]$  除以  $x[rs_2]$  向零舍入，余数写入  $x[rd]$

1) jal ra, 0x88 : 偏移量寻址

2) jalr x<sub>0</sub>, ra, 0 : 寄存器间接寻址

3) addi a0, a1, 4 : 立即数寻址

4) mtl a0, a1, a2 : 寄存器直接寻址

5) ld a4, 1b(sp) : 偏移量寻址