

14: 第二章习题:

1. CISC(复杂指令集计算机): **优势** ①由于指令复杂, 故可以完成复杂的任务, 进行复杂数据处理(图形、视频处理等)。②使编译器在编写程式时更易进行代码优化, 提高代码执行效率。③寻址方式较为灵活。④以微程序控制器为核心, 指令和数据存储器共享同一个物理存储空间, 性能强大。

劣势: ①指令集过于复杂, 导致芯片设计的复杂性, 不利于LSI(超大规模集成电路)的实现, 制造成本较高。②指令集复杂性让处理器执行指令时能耗增加, 导致能耗降低, 且不利于使用先进的架构提升性能。③容易产生错误和漏洞, 使系统的稳定性和安全性降低。

2. RISC(精简指令集计算机): **优势:** ①指令集结构简单, 使处理器设计简便且易于实现。②执行指令时间相对较低, 性能提高, 执行效率高。③更加稳定且安全, 降低出错可能。

劣势: ①指令较少, 无法处理一些如浮点数计算的复杂数据操作与任务。②寻址方式不够灵活, 编译器优化效果相对CISC较低。(对编译器要求高)

2. RISC-V基本指令集: RV32I / RV64I / RV128I: 分别为32位/64位/128位整数指令集。
该指令集中包括: 加载、存储、算术、逻辑、分支、访存和跳转等基本指令。(RV32E: 32位嵌入式指令集)

RISC-V标准扩展指令集: ① RV32M/RV64M: 包含乘除法和取模未除指令, 实现整数乘除法运算。
② RV32F/RV64F...: 单精度浮点指令, 实现浮点运算。③ RV32D/RV64D: 双精度浮点指令, 用以实现双精度运算。
④ RV32A/RV64A...: 原子指令扩展, 包含原子加、修改和写指令, 可实现各种同步锁操作。
⑤ RV32C/RV64C...: 压缩指令, 其单指令长度为16位, 主要用于改善程序的大小。

4. (1) RV32I中的add指令操作码为0110011, RV64I中的addw指令操作码为0111011, 指令操作码不同。
RV32I中的add指令操作码为0110011, RV64I中的add指令操作码为0111011, 指令操作数相同。

分析了: ① RV32I中的add指令和RV64I指令虽然均为32位整型加法指令, 但在RV64I中实则是将两个寄存器操作数相加后结果截断为32位, 并写入存储器, 忽略算术溢出的结果, 故在加法后需多进行一步截断操作, 故操作码不同。
② RV32I和RV64I中的add指令是具有继承关系的, 其执行的操作均为操作数的相加, 操作相同, 故操作码相同。
(2) addw和addiw指令且最高位中有放在32位计算结果需要进行额外的符号扩展才能用于后续64位的计算, 因为虽然结果中有符号扩展, 但64位中在运算时同样需要保持符号不变, 故需将结果符号扩展至64位, 进行后续计算。

5. RISC-V的标准指令集中有HINT指令空间，HINT指令空间可被用以定义一些轻量级的指令，这些指令主要用向微架构传递一些处理器性能的提示信息。且如同NOP指令，该类指令除推动PC及任何可用性能外，并不会改变体系结构中可见的状态，在执行时可被忽略。HINT指令空间故主要用于指令流水线的流程控制和性能优化，改善流水线性能，优化程序执行顺序，提高程序的吞吐量。

6. div a₃, a₀, a₁ 是进行 a₀ / a₁ 的除法运算，并将商存入 a₃ 寄存器中，故 a₃ 寄存器中值为 3。

rem a₃, a₀, a₁ 是进行 a₀ 除以 a₁ 后取余操作，并将余数存入 a₃ 寄存器中，故 a₃ 寄存器中值为 1。

RISC-V标准指令集中① div 指令进行有符号整数除法运算，divu 进行无符号整数的除法运算

若被除数与除数符号相同，结果为正；若被除数与除数符号相反，结果为负。（当出现大零时，DIVU 结果为 $2^{XLEN}-1$ ，DIV 结果为 -1）

② rem 指令进行有符号整数除法后取余运算，remu 进行无符号整数的除法后取余运算

rem 指令运算后的结果符号与被除数的符号相同（当出现大零时，REM 和 REMU 的结果均为 x）。

11. (1) jal ra, 0x88 : 使用偏移量寻址：将下一条指令的地址 (PC+4) 存入 rd 寄存器并往 PC 地址加上 0x88 进行跳转

(2) jalr x0, ra, 0 : 使用立即数寻址：在 x0 中存入现 PC+4 的值，并将目标地址送给寄存器 ra 加上立即数 0，进行跳转

(3) addi a₀, a₁, 4 : 使用立即数寻址：将 a₁ 寄存器与立即数 4 相加，并将结果存储在寄存器 a₀ 中

(4) mul a₀, a₁, a₂ : 使用寄存器直接寻址：将 a₁ 寄存器和 a₂ 寄存器中原操作数相乘的结果存储在寄存器 a₀ 中。

(5) ld a₄, 1b(sp) 使用偏移量寻址：将栈指针 sp 指向的地址加上立即数 1b 后地址处的数据读取并存储在寄存器 a₄ 中