

第四周作业

1. CISC
优：对编译器和程序存储空间的要求较低。

劣：硬件设计复杂，测试验证难度较高。

RISC
优：硬件设计较为简单，适合利用流水线提升性能。

劣：对编译器设计的要求较高，程序的代码密度较低。

2. RISC-V属于寄存器-寄存器型，具有典型的Load/Store架构，包括RV32I、

RV64I和RV128I，分别对应32位、64位和128位的RISC-V处理器。

常见的RISC-V标准扩展指令集有：

(1) RV32M：乘法扩展指令集，提供了MUL和DIV等乘法和除法指令，可以加速性能计算、数字信号处理等需要进行乘除运算的应用。

(2) RV32F：单精度浮点扩展指令集，提供了浮点运算指令，适用于科学计算、图形运算等应用。

(3) RV32D：双精度浮点扩展指令集，适用于高性能计算、天文学、物理学等需要双精度运算的应用，提供了双精度浮点运算指令。

(4) RV32A：原子扩展指令集，提供了原子操作指令，可以实现原子性操作，保证多线程编程的正确性。

(5) RV32C：压缩扩展指令集，通过指令压缩技术，将常用指令压缩成16位，减少程序的大小，增高存储器带宽。

4.11 不同。RV32I中add指令的opcode是0110011，而在RV64I中addw指令的opcode是0111011。

而RV64I中add的opcode为0110011，与RV64I中的相同。这样的设计是为了支持不同位宽的数据操作。RV32I中的add指令可以在32位整数操作数上执行，而RV64I中的add指令可以在64位整数操作数上执行。然而，在RV64I中，由于操作数位宽更宽，因此需要一个不同的指令来执行在32位操作数上进行加法，即addw指令。

(2) RV64I中，addw指令的目标寄存器中存放的是32位有符号整数的计算结果，而“addiw”指令的目标寄存器存放的是立即数和32位有符号整数的计算结果。在这两种情况下，如需将结果用于后续的64位计算，那么需要进行符号扩展。

因为在进行64位计算时，高32位必须是符号扩展后的结果，否则可能会出现不正确的结果。

5. HINT是一组用于处理器性能优化的特殊指令。它们被设计为不产生任何操作结果，而是用于向处理器发出指令，以便其在指令流中优化性能。

这些HINT指令的作用取决于具体的实现和处理器架构，但它们通常用于以下几种情况：

1. 分支预测提示：处理器可以使用HINT指令来提示分支预测器，以便更准确地预测分支指令的方向。

2. 数据预取提示：处理器可以使用HINT指令来提示内存预取器，以便提前从主存中取得需要的数据。

3. 处理器状态提示：用HINT指令来提示操作系统或虚拟机的监控程序，以便它们可以更好地管理处理器状态并进行性能优化。

总之，HINT指令空间为处理器提供了一个用于调整和优化其内部工作的工具箱，提高处理器性能并提高系统吞吐量。

6. div指令：将第一个操作数除以第二个操作数，将商存储到第三个操作数中。如果两数的符号不同，则商的符号为负，否则为正。

故 a_2 的值应为 -3

rem指令：若两数符号不同，则余数的符号与除数的符号相同。
故 a_3 的值为 1.

11. (1) 偏移量寻址

(2) 内存直接寻址

(3) 立即数寻址

(4) 寄存器直接寻址

(5) 寄存器间接寻址