

1.

(1) CISC是复杂指令计算机，其体系结构使用大量指令，包括复杂指令。其主要优势为：

① 实现相同操作所需指令数少，~~指令类型~~ 单条指令就可以完成复杂的功能

② 指令类型丰富，操作灵活，指令长度灵活

③ 对编译器和程序存储空间的要求较低

④ 寻址方式灵活

其主要缺点为：

① 硬件设计复杂，测试验证难度高，研发成本高

② 指令使用率不均衡

(2) RISC是精简指令集计算机，是一种指令类型较少的计算机。

其主要优势为：

① 指令格式统一、类型简单，~~单指令~~ 指令长度相对固定

② 硬件设计较为简单，适合利用流水线提升性能，硬件开发周期短。

③ 指令精简，使用率均衡，程序执行效率高

其主要缺点为：

① 对编译器要求较高，程序的代码密度较低

② 指令数较少，在指令的灵活性上受到限制

③ 寻址方式不够灵活。

2. RISC-V 中的基本指令集 ~~不是~~ 是整数的基本指令集，有以下三种：  
RV32I (使用 32 位寄存器的基本 32 位整数指令)  
RV32E (只使用 16 位寄存器的基本 32 位指令，适用于低端嵌入式应用)  
RV64I (使用 64 位寄存器的基本 64 位整数指令)

扩展指令集：

M：扩展了整数乘法和除法指令

A：扩展了并发操作中的原子指令

F：扩展了 IEEE 标准单精度浮点数运算指令，增加了 32 个 32 位 浮点寄存器

D：扩展了 IEEE 标准双精度浮点数运算指令，增加了 32 个 64 位 浮点寄存器

Q：扩展了四精度浮点数运算指令

L：扩展了 IEEE 标准的 64 位或 128 位十进制浮点数运算指令

C：定义了部分指令的 16 位版本，用于小内存的嵌入式应用

V：扩展了向量操作指令

B：扩展 J 位操作指令

T：扩展了事务性内存指令

P：扩展了对 SIMD 指令的支持

RV128I：扩展了对 128 位地址空间访问的支持

4. (1) RV32I 中的 add 指令和 RV64I 中的 addw 指令并不具有相同的  
指令操作数。

RV32I 中的 add 指令和 RV64I 中的 addw 指令也不具有相同的  
指令操作数。

这样设计的目的是可以根据操作码的不同来区分执行指令的不同。

(2) 需要进行符号扩展，~~32 位的符号位在 64 位中是正带数字加~~  
~~位置，不扩展符号位会导致错误~~

addw 与 addiw 作用在 32 位数值上，产生有符号的 32 位结果，溢出被忽略，结果要扩展成 64 位再写入目标寄存器，否则会生成错误

5. HINT 指令被设计成未来支持增加微体系结构提示，这些提示可能影响性能，但不能影响体系结构状态。HINT 编码已经被选定，因此简单的实现可以忽略 HINT 编码，并将 HINT 指令作为常规指令执行，不改变体系结构状态。

6.  $a_2$  中的值为 -3

$a_3$  中的值为 1

除法和余数指令，默认有符号-有符号运算，包括  $s$  和  $t$  为有符号-无符号运算，包括  $u$  和  $v$  为无符号-无符号运算，包括  $h$  和  $m$  可实现 16 位 32 位 运算和对输出结果进行高 低位取。

## 11. (1) 偏移量寻址

(2) ~~寄存器直接寻址~~ 寄存器间接寻址

(3) 立即数寻址

(4) 寄存器直接寻址

(5) 偏移量寻址