1、解：(1)CISC架构

优势：① 指令集丰富，单个指令完成的任务量大且功能复杂，指令长度灵活

② 对编译器和程序存储空间要求较低；

劣势：① 硬件设计复杂；

② 测试验证难度较高。

(2) RISC架构

优势：① 硬件设计较为简单，适合利用流水线提升性能；

② 内存访问次数较少，能更好地利用缓存；

劣势：① 对编译器的要求高，程序的代码密度较低；

② 缺乏某些高级功能，需编写额外代码。

2、解：(1) RISC-V 中的基本指令集是

① 数据传输指令；

② 算术逻辑指令；

③控制转移指令；

④浮点、运算指令。

(2) RISC-V 标准扩展指令集：

① RV32M/64M：乘法扩展指令集，增加了乘法和除法指令，可在图像处理、数字信号处理中提高性能；

② RV32I/64I：基本整数指令集，包含加载、存储、逻辑运算等操作，被用于嵌入式系统、低功耗处理器等领域；

③ RV32A/64A：原子操作扩展指令集，支持多线程共享内存并发访问时对共享变量进行原子性操作，在服务器、路由器等领域中使用；

④ RV32F/64F：单精度浮点数扩展，支持单精度浮点数运算，用于图形渲染、人工智能等领域；

⑤ Zicsr 扩展：定义了一组控制状态寄存器操作的新命名空间，并增加了一组CSR寄存器来管理 cpu 状态信息，可帮助开发者更好地调试程序或监视 CPU 状态。

4. 答：(1)① RV32I 中的 add 指令的操作数为 0110011

RV64I 中的 addw 指令的操作数为 0111011

因此二者的操作数不同

② RV32I 和 RV64I 中的 add 指令操作数均为 0110011

这样设计的目的是提高代码的兼容性和可移植性，便于编译器和汇编程序的处理。同时，RV64I 中扩展的新指令，也增强了指令集的功能与灵活性。

(2) 需要进行额外的符号扩展

理由：在 RV64I 中，多数操作码对64位寄存器进行操作。因此

32位运算结果需扩展为64位用于后续运算

5. 答：HINT指令空间包含一些有特殊目的、不执行任何操作但可以影响处理器行为和性能的指令，用于优化代码性能或进行调试。

包括：① nop：无操作

② ecall/ebreak，用于触发系统调用或软件断点，进入异常处理流程

③ csrrw/csrrs/csrrc：对特定的一些寄存器进行读写；

④ fence：控制内存访问顺序

6. 解：a_2寄存器中的值为 -3，a_3寄存器中的值为 1

对于除法和余数指令的符号规定为：

DIV：有符号整型除法：将rs1中的值除以rs2中的值，结果存在rd中。

DIVU：无符号整型除法：操作与DIVU相同，处理无符号数。

REM：有符号取余：两数相除取余数。余数的符号与被除数相同

REMU：无符号取余：操作与REMU相同，处理无符号数

11. 解：(1) 偏移量寻址

(2) 寄存器间接寻址

(3) 立即数寻址

(4) 寄存器寻址

(5) 直接寻址