1.解：

CISC：优点：(1)因为单个指令完成的任务量大，对编译器和程序存储空间要求较低

(2)指令长度灵活，寻址方式灵活

缺点：(1)指令使用率不均衡

(2)硬件设计复杂，测试验证难度较高，不利于采用先进结构改进性能

RISC：优点：(1)硬件设计较为简单，适合利用流水线提升性能

(2)指令精简，使用率均衡

缺点：(1)对编译器设计的要求较高

(2)程序代码密度较低

2.解：

RISC-V中的基础指令集主要是整数指令集（用I表示），根据寄存器位宽的大小分为32I，64I，128I三种整数指令集，而每一种中又有六种基本指令类型，如R-type，I-type，S-type，B-type，U-type，J-type

扩展指令集：M：乘除法，取模求余指令

F：单精度浮点指令

D：双精度浮点指令

A：原子操作指令，例如CAS和LL/SC指令等

C：压缩指令，主要用于改善程序大小

4.解：

(1) RV32I的add和RV64I的addw指令的opcode为0111011 不同

RV32I的add指令和RV64I的add指令的opcode都为0110011，相同

分析：RV32I的add和RV64I的addw实现的功能不同，虽然都是32位数的加法，但addw处理的是imm的低32位，而RV32I的add处理32位imm的全部位数，所以opcode不同，而RV32I和RV64I的add都是处理整个imm进行加法，操作相同，所以opcode相同

(2) 不需要；RV64I中的addw和addiw指令的目标寄存器中存放的32位计算结果已经是加法计算后经过符号扩展的结果，所以在后续计算中，不必再进行第二次的符号扩展

5. 解：

HINT指令又称为提示指令，通常向微架构传达性能提示，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，如X86的NOP指令一样，仅为空操作。

大多RV32I HINTs都被编码为rd=x0的整数计算指令，如ADDI指令，令rd=x0，则为HINT指令，32位减去7位opcode，5位rd=x0和3位funct3，剩下17位，编码位点为2^{17}，加上其余指令HINT的编码位点，就构成了RV32I的HINT空间

6. 解：a2寄存器中值为3，a3寄存器中值为1

div除法指令为有符号数间的除法，在执行操作前，要将rs1，rs2中数进行有符号扩展

rem余数指令为有符号数取余，在执行前，对rs1，rs2进行有符号扩展，一般来说余数为正

11. 解：(1) 偏移量寻址

(2) 寄存器间接寻址

(3) 立即数寻址

(4) 寄存器直接寻址

(5) 偏移量寻址