1、简要分析RISC和CISC架构各自的优势和劣势.

CISC架构单个指令可完成的任务量大且功能复杂，指令长度灵活.

优点：对编译器和程序存储空间的要求较低

缺点：硬件设计复杂，测试验证难度较高

RISC架构单指令完成的任务量少，指令长度相对固定.

优点：硬件设计较为简单，适合利用流水线提升性能.

缺点：对编译器设计要求较高，程序代码密度较低.

2. RISC-V中的基本指令集是什么？列举五个常见的RISC-V标准扩展指令集，并简要说明它们的作用和应用范围。

基本指令集：RV32I、RV32E、RV64I、~~RV64E、RV128I~~

扩展指令集：①M扩展（乘法、除法），用于支持各种数据类型的乘除法运算，可用于计算密集的领域。②F扩展：支持单精度浮点数运算，在处理实数数据的领域实用。③~~双精度~~D扩展：支持双精度浮点数运算，在对精度有高要求的领域实用。④A扩展，提供原子操作指令，可支持多线程和多核系统的并发编程。⑤C扩展，提供16位压缩指令，适用于嵌入式等领域，可减少存储与能耗压力。

4. 阅读RISC-V规范以回答以下问题：

1) RV32I中的add指令与RV64I中的addw指令，均为32位整型加法指令，它们是否具有相同的指令操作数（opcode）？此外，RV32I中的add指令和RV64I中的add指令是否具有相同操作数？为什么采取这样的设计？

2) 在RV64I中，addw和addiw指令的目标寄存器存放的32位计算结果是否需要进行额外的符号扩展才能用于后续的64位计算？请说明理由。

1)、RV32I中的add指令与RV64I中的addw指令的操作数不同。对RV-32I中的add，其操作码可表示为OP-IMM，而addw在RV64中则是OP-IMM-32，是专门用于处理32位整数加法并扩展至64位有符号数的指令。而RV32I与RV64I中的add指令有相同的操作数

设计好处：①相同操作数的add指令可保持指令集的一致性与兼容性，而在RV64中，add指令既可用于32位和64位的加法，而addw则专为32位整数运算设计。此外，add和addw还可以轻松判断结果是否溢出，如果发生溢出，两个指令的结果将会不同，这比RV32I中对溢出的判断更简单。

2) addw和addiw的结果不需要再进行专门的符号扩展，因为指令本身包含了自动符号扩展，只计算两数的低32位，并将结果符号扩展到64位。这么做可以在64位环境下高效进行32位整数加法运算。

5. 什么是RISC-V的I标准指令集中存在的HINT指令空间？它有什么作用？

HINT指令空间是为架构提示指令预留的指令集空间，是一组未分配或~~使用频率低~~的指令编码。HINT指令空间可以用于实现特定架构的优化，在软件层面上，HINT指令没有直接的操作数与作用，但可为处理器提供实现的相关的建议等。

6. 考虑如下指令序列：

```
div  a2, a0, a1
rem  a3, a0, a1
```

假设寄存器a0和a1的初始值分别为16和-5，则上述指令执行完成后a2和a3寄存器中的值分别为多少？简要说明RISC-V的M标准指令集中对除法和余数指令的符号规定。

$a_2 = a_0 / a_1 = 16 / (-5) = -3$; $a_3 = a_0 \% a_1 = 1$

符号规定：除法指令(div)，结果的符号为被除数与除数符号的异或。~~此外，结果不是整数的情况下~~结果向零舍入.

余数指令(rem)：结果的符号同被除数相同，且有余数=被除数-(商×除数)的关系式。

11. 写出以下指令使用的寻址模式.

1. jal ra, 0x88 偏移量寻址
2. jalr x0, ra, 0 ~~偏移量寻址~~ 寄存器间接寻址
3. addi a0, a1, 4 立即数寻址
4. mul a0, a1, a2 寄存器立接寻址
5. ld a4, 16(sp) 偏移量寻址