2.(1,2,4,5,6,11)

1. CISC和RISC之间区别在于指令的复杂性和数量。CISC的指令集比RISC更复杂，每条指令可以执行更多操作，而RISC的每条指令只执行最基本的操作。

CISC架构优点：

(1) 指令集丰富，每条指令可执行多个操作，可减少编程工作量

(2) 可以利用微码和专用硬件实现多个操作，加快执行速度

(3) 数据通路复杂，支持多种数据类型和数据结构

CISC缺点：

(1) 指令集复杂导致芯片设计和制造成本高，也使编译器的设计和代码优化困难。

(2) 指令集繁杂，不利于指令流水线进行，降低了指令执行速度

(3) 一些指令很少用，浪费了芯片上的空间

RISC优点：

(1) 指令集较简单，每条指令执行操作较少，使指令执行更快更good

(2) 芯片设计和制造成本低

(3) 编译器设计和优化代码优化更容易，提高了程序执行效率

RISC缺点：

(1) 编程工作量大

(2) 只支持基本的数据类型和数据结构

2. RISC-V 的基本指令集包括 RV32I，RV64I 和 RV128I

五个常见的 RISC-V 标准扩展指令集：

(1) RV32M 和 RV64M：这些指令扩展添加了整数乘法和整数除法指令，使得 RISC-V 更高效地执行整数乘除操作，广泛用于数字信号处理、嵌入式系统等领域

(2) RV32F 和 RV64F：添加了单精度浮点数的支持，广泛用于科学计算、图形处理等领域

(3) RV32D 和 RV64D：添加了双精度浮点数的支持，广泛用于科学计算、图形处理等领域

(4) RV32A 和 RV64A：添加了原子操作指令，支持在多线程环境下进行原子性操作，使 RISC-V 可以更高效地执行同步和互斥操作，广泛用于并行计算、分布式系统等领域

(5) RV32C 和 RV64C：提供了压缩指令集，可以减小代码大小，提高代码密度，广泛用于嵌入式系统、移动设备等资源受限的环境。

4. (1) RV32I 中的 add 指令和 RV64I 中的 addw 指令具不同指令操作数。RV32I 中的 add 指令的 opcode 为 0110011，而 RV64I 中的 addw 指令的 opcode 为 0111011。

RV32I 中的 add 指令和 RV64I 中的 add 指令是相同的 opcode，即 0110011。这种设计是为了在 RISC-V 中提供灵活的指令集。RISC-V 允许使用不同的指令长度，并具有不同的指令操作数，使指令集更灵活和可扩展，同时也使得硬件的实现更加简单和高效

(2) 在 RV64I 中 addw 和 addiw 指令的目标寄存器中存放的 32 位计算结果需要进行符号扩展才能用于后续 64 位计算。这是因为在 RISC-V 中，所有的运算数

都是按照64位的方式进行的，即使操作数是32位的。因此，当使用addw和addi指令时，它们会将32位的结果符号扩展为64位，然后存在目标寄存器中。这样可以确保所有整数运算都是按照64位进行的，从而提高指令集的一致性和可移植性。

5. HINT指令空间是一个指令集合，这个集合中的指令并不会直接改变程序的控制流或者对数据进行操作，而是向处理器发出一些提示信息，帮助处理器更好地进行指令的调度和执行，提高处理器效率。

6. (1) 寄存器a0和a1的初值分别为16和-5，执行div a2, a0, a1后，a2的值为-3（16÷(-5)的结果为-3余1），rem a3, a0, a1后a3的值为1（余数）

(2) RISC-V的M标准指令集中规定，对于除法和余数指令，其符号的选择由指令的第3个操作数决定，如果该操作数的最高位为1，则执行带符号操作，否则执行无符号操作。

11. (1) 直接寻址模式

(2) 基于寄存器的偏移寻址模式

(3) 立即数寻址模式

(4) 寄址器寻址模式

(5) 基于栈指针的偏移寻址模式