No. Date

1、CISC优势：①指令类型丰富、功能强大且复杂，指令长度灵活

②程序编写较容易，对编译器要求低

③新指令/微代码设计时间较短，易于创新

④支持多种寻址方式，对程序存储空间要求较低

CISC劣势：①指令集复杂⇒CPU结构复杂、硬件设计复杂，不利于VLSI实现且测试验证难度较高。

②指令利用率不均衡，执行效率受指令类型影响大。

③不利于采用先进结构(如流水线)提高性能。

RISC优势：①结构简单、易于设计与实现

②单条指令执行速度快，可达主频较高。

③指令精简、易于理解、规范均衡。

RISC劣势：①对编译器设计要求高、程序代码密度较低

②寻址方式不够多灵活

③访存可能较频繁，存在一些影响效率的复合操作。

2、基本指令集 RV32I、RV64I (32位、64位)。

扩展指令集如：

RV64M：用于乘除法计算(包含乘除法指令)。用于整数(有/无符号)乘除法计算。

RV64F：包含单精度浮点指令，作用为实现一系列单精度加减乘除运算，适用于许多需要高效浮点运算的领域

RV64D：包含双精度浮点指令，作用为实现一系列双精度加减乘除等运算，适用于许多需要高精度浮点计算的场景

RV64A：包含原子操作指令，可实现内存原子操作、加载保留/条件存储、用于实现原子的比较交换，适用于多线程、并发

RV64V：包含向量操作指令，作用为实现数据并行计算，可用于神经网络计算等领域

RV64C：将通用指令位数压缩。

No. Date

4、(1) RV32I中add指令与RV64I中addw指令opcode不同，前者0110011后者0111011
但RV32I中add指令与RV64I中add指令opcode相同，均为0110011
这样设计是因为add指令被定为执行X位加法操作，保证opcode相同可保持指令集的一致性，而通过扩展出64位的addw指令来实现对64位数的低32位操作，这需用不同的opcode来区分，这样的设计有助于提高指令集的兼容性和可扩展性，保持一致性，使得编译器和硬件工程师可以在不改变opcode的同时支持对应体系位数的运算，即允许在不同的数据宽度上使用相同opcode，提高了代码的可重用性和~~可扩展性~~可移植性

(2) 不需要，在对运算结果低32位截断后，会~~自动~~对其进行一次符号扩展再存入目的寄存器，可直接用于后续64位计算

5、HINT指令(提示指令)通常用于向微架构传达~~某些~~性能提示，除推动PC及任何可用的性能计数器外，并不改变任何体系结构可见的状态，是一类无实际操作意义的指令，HINT空间指为该类型指令保留的编码空间

6、 a_2寄存器为 -3　　a_3寄存器为 1

对于除法：div指令为有符号除法　divu为无符号除法
对于有~~符~~符号除法div指令，所得商符号由除数和被除数同时决定

对于余数：rem指令为有符号求余　remu为无符号求余
对于有符号求余rem指令，所得余数符号与被除数符号相同

11. ~~均有~~ 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