习作四

1、CISC：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：优点：硬件设计较简单，适合用流水线提升性能
缺点：对编译器设计要求高，程序代码密度较低

2、RISC-V 基本指令集：
例如：32位整数指令集 RV32I，
~~64位整数指令集 RV32E~~ RV64I，
128位整数指令集 RV128I，
32位嵌入式整指集 RV32E

常见RISC-V标准扩展指令集：
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
Q：四倍浮点指令
A：原子操作指令

4、(1) ~~RISCV~~ RV32I 中 add 与 RV64I 中 addw 具有不同 opcode，
可看是 0110011；后者是 0111011；
RV32I add 指令与 RV64I 中 add 指令具有相同 opcode
好处：add ~~忽略了溢出与而~~ addw ~~考虑了溢出，使得加法正确执行的范围会更大，这两种~~ 操作结果完全不同，得到的结果也不同，故而用 opcode 区分；
而 RV32与RV64 add 指令~~者~~操作一样，故 opcode 一样

(2) 不需要，addw 与 addiw ~~支持有符号的运算~~，若忘 ~~溢出~~ ~~计算过程中产生的溢出~~。会自动将计算结果进行有符号位扩展，故无需再进行一次扩展

5、RISC-V中I标准指令集中的HINT指令，是用于向微架构传达性能提示的提示指令，可以推动pc以及任何可用性能计数器，rd与x0的add指令其实就是一个HINT，但实际上该指令不会造成架构上可见的影响。这些HINT指令可用在关于内存系统时间和空间区域性提示、分支预测提示等。

6、a_2 值为 -3，a_3 的值为 1

M指令中 div，rem 支持有符号运算，div结果直接去掉小数点，rem结果符号与被除数相同；remu，divu 支持无符号运算。

若0为除数，即 $x/0$，则结果为

~~指令~~	div	divu	二进制
结果	-1	$2^{XLEN}-1$	11111……111（32个1）

指令	rem	remu
结果	x	x

若div、rem指令中产生符号溢出，即

$-2^{XLEN-1}/-1$，则结果为：

~~指令~~	div	rem
结果	-2^{XLEN-1}	0

11. 1) jal ra, 0x88.

即 ra = PC+4, PC = PC + imm

偏移量寻址

2) jalr x0, ra, 0.

即 x0 = PC+4, PC = ra + 0.

寄存器直接寻址

3) addi a0, a1, 4

即 a0 = a1 + 4

立即数寻址

4) mul a0, a1, a2

即 a0 = a1 * a2

寄存器直接寻址

5) ld a4, 16(sp)

即 a4 = sp + 16.

偏移量寻址