

1. 2. 4. 5. 6. 11

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: 硬件设计较为简单, 适合利用流水线提升性能

缺点: 对编译器设计的要求较高, 程序的代码密度较低

2.

RISC-V 是寄存器-寄存器型指令集

基本指令集

RV32I 使用32位寄存器的基本32位整数指令

RV32E 只使用16个寄存器的基本32位指令, 适用于低端的嵌入式设备

RV64I 使用64位寄存器的基本64位整数指令

扩展指令集

M 扩展整数乘法和除法指令

A 扩展了并发操作中的原子指令

F 扩展了IEEE标准单精度浮点数运算指令, 增加了32个32位浮点寄存器

D 扩展了IEEE标准双精度浮点数运算指令, 增加了32个64位浮点寄存器

Q 扩展了IEEE标准精度浮点数运算指令

4. 1) `add rd, rs1, rs2` 有符号加法指令

操作 $rd \leftarrow rs1 + rs2$ opcode 0110011

`addw rd, rs1, rs2` [低32位有符号加法指令]

$tmp[31:0] \leftarrow rs1[31:0] + rs2[31:0]$

$rd \leftarrow sign_extend(tmp[31:0])$ opcode 0111011

故指令操作数不同

每一条指令都有一个操作码，它表示指令应进行什么性质的操作，不同的操作码可以确保代码的这组操作的一致性，

(2) 需求

RISC-V中的整数寄存器大小为64位，而`addw`和`addiw`指令只能对32位计算有符号整数进行计算，因此它们的计算结果只有32位。如果直接将32位结果存放到寄存器中，那么高32位的值将丢失，可以认为是任意值。

5. 在RISC-V的I标准指令集中，存在一个HINT指令空间。这个指令空间包含了一些用于提高CPU性能和降低功耗的指令，称为HINT指令。这些指令通常不会改变程序的语义，而是向CPU提供了一些有用的提示，帮助CPU更好地执行指令序列，HINT指令作用主要有以下几个方面

1. 提高CPU性能：HINT指令可以向CPU提供了一些有用的提示，例如表明指令序列的分支情况、预测跳转目标等。这些提示可以帮助CPU更好地执行指令序列，从而提高CPU性能。

2. 降低功耗：HINT指令还可以向CPU提供了一些提示，例如表明指令序列的空闲时间，

等待时间，帮助cpu在空间时间内降低功耗

提高可移植性，

6. `div a2, a0, a1`

`rem a3, a0, a1`

$a_2 = -3$ $a_3 = 1$

`div`: 按照二进制补码除法运算，并将结果存储在目标寄存器中

如果余数为0，将商和余数设为0

`rem`指令 同上，将余数存储。

11 `jal ra, 0x88`

立即数寻址 偏移量寻址

`jalr x0, ra, 0`

寄存器间接寻址，通过寄存器 `ra` 中的地址访问
目标地址

`addi` 立即数

`mul` 寄存器寻址

`ld aq, 16(sp)`

相对寻址模式，访问地址偏移量的偏移量为16的地址

↓ 将该地址加到寄存器 `aq`
偏移量