

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，适用于低编译带入式应用

$RV64I$ 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 位浮点数

D 扩展了 IEEE 标准双精度浮点数运算指令，增加 32 位双精度浮点数

G 扩展了向量精度浮点数运算指令，增加 32 位向量寄存器

4. (1) add r0, rs1, rs2 有符号加法指令

操作 $r0 \leftarrow rs1 + rs2$ opcode 0110011

addlw r0, rs1, rs2 无符号加法指令

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

$r0 \leftarrow sign_extend(tmp[31:0])$ 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~~ 这些操作的唯一性，

(2). 零宽

RV64I 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 位结果存储到寄存器中，那么 32 位值将是未定义的，可以是任意值。

5. 在 RISC-V 的标准指令集中，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 提供一些信息，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

a2 = -3 a3 = 1

div：按照补码带符号除法运算，并将结果存储至目标寄存器中

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 编移量寻址

jalr r0, 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

(d a4, 16(sp))

相对寻址模式，访问栈指针sp的偏移量为16的地址

将该处地址加载到寄存器a4中

偏移量