

9.

- (1) JAL 指令使用 C 类格式，立即数丁 编码一个 2 的倍数的有符号偏移量，偏移量符号扩展后加到 PC 上，跳转的范围达到 $\pm 1MB$
- (2) bne 指令使用 B 类格式，12 位立即数编码了一个以 2 字节倍数的有符号偏移量，并被加到当前 PC 上，条件分支范围是 $\pm 4KB$
- (3) 可以

LUI 指令将目标地址的高 20 位加载到寄存器中，然后 JALR 指令可以加上低 12 位，所以可以跳转到 32 位绝对地址空间的任何地方。

10.

32 位指令被压缩为 16 位 RVC 指令的条件：

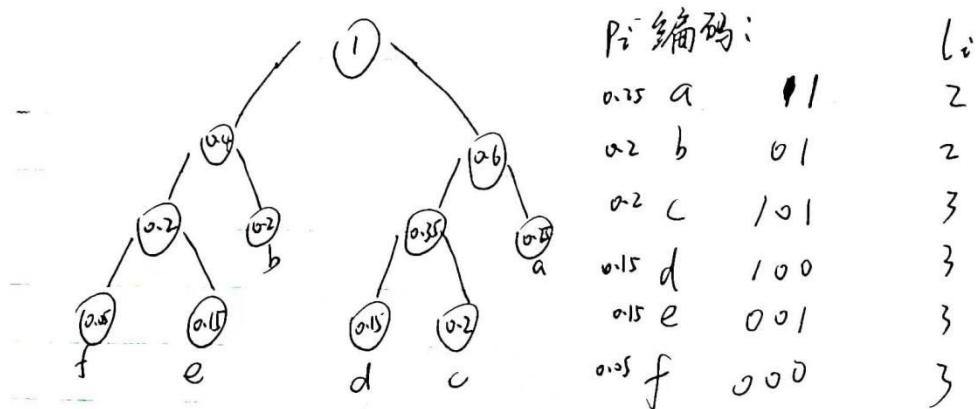
- ① 立即数或地址偏移量较小
- ② 其中一个寄存器是零寄存器 (X_0)、AB1 链接寄存器 (X_1) 或 AB2 栈寄存器 (X_2)
- ③ 目标寄存器和第一个源寄存器相同

并不是所有指令都可以使用所有 32 个 RV1 寄存器

CR、C1、CS 格式的压缩指令可以使用所有 32 个 RV1 寄存器

而 CJW、CL、CS、LB ~~类~~ 格式的指令被限制为只能使用 32 个寄存器中的 8 个寄存器。

18.



$$L^* = \sum_i P_i l_i = 2.55$$

$$H = - \sum_i P_i \log_2 (P_i) = 2.47$$

~~$$R = 1 - \frac{H}{L^*} = 0.03 = 3\%$$~~

操作码平均长度为 2.55
信息冗余度为 3%.

19.

(1) 在计算机中，函数调用是通过栈这种数据结构实现的，每当程序执行进入一个函数调用，栈就会加一层栈帧，每当函数返回，栈就会减一层栈帧。由于栈的大小不是无限的，所以，递归调用的次数过多，就会导致栈溢出。函数参数是通过栈来传递的，在调用中会占用线程的栈资源。递归调用在到达最后的结束点后，函数才能依次退栈清栈，如果递归调用层数过多，就可能导致占用的栈资源超过线程的最大值，从而导致栈溢出，程序异常退出。

(2) 尾递归优化可以避免栈溢出

尾递归优化，无论调用多少次，都只占用一个栈帧，不会出现溢出

20. $\gamma_a(F_1)$

$\gamma_a(F_2)$

$t_0(F_1)$

$s_0(F_2)$

$\gamma_a(F_3)$

$t_0(F_2)$

$t_1(F_2)$

$s_0(F_3)$

$s_1(F_3)$