

9. 小 \checkmark jal 指令

(1) 跳转范围为 $\pm 1M$; B

(2) 条件分支指令跳转范围是 $\pm 4KiB$ 作为 offset[31:12]

(3) 可以; 先用 lui 将 imm 的 12 位存入 jalr 指令将要调用的 rsi 寄存器内, jalr 指令中的立即数则为 offset[11:0], 最后 jalr 指令使 $PC = rsi + imm$ 即可实现 32 位地址跳转

10. 解: 压缩部件:

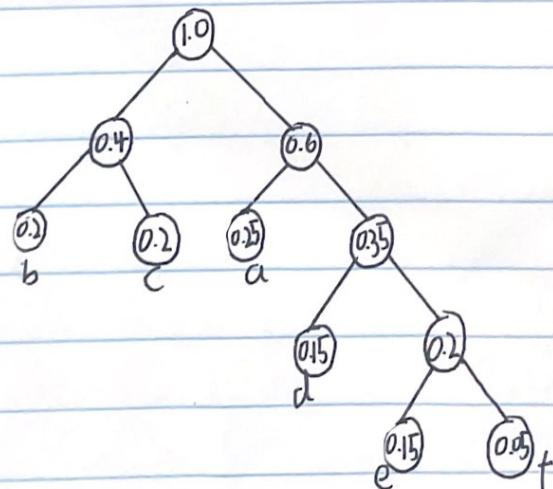
(1) 立即数或者地址偏移量较小时

(2) 其中一个寄存器是 X0, X1 或 X2

(3) 目标寄存器和第一个源寄存器相同

(4) 最常见情况下使用了 8 个寄存器

18. 解:



$$\sum_{i=1}^6 P_i = 2.55$$

$$H = -\sum_{i=1}^6 P_i \log_2 P_i = 2.466$$

$$R = 1 - \frac{H}{\sum_{i=1}^6 P_i}$$

$$= 0.033$$

19.

解：① 每次递归就相当于调用一个函数，^{每次}函数调用时都会将局部数据（在函数内部定义的变量、参数等，如 sp, ra, so 等）放入栈中。这些数据占用的内存直到整个递归结束才释放，若递归次数过多，并且局部数据也多，那么会使用大量栈内存，导致栈溢出。

- ② ① 减少栈空间需求，不要定义占用内存较多的变量，如很大的数组
② 函数参数中不要传递大型结构、联合对象，尽量传递指针等
③ 减少函数调用层次，优化慎用递归函数

20.

ra(F1)

so(F1)

(t0) a0(F1)

ra(F2)

so(F2)

s1(F2)

(t0) a0(F2)

(t1) a1(F2)