



T9 1) [current pc - 2<sup>19</sup>, current pc + 2<sup>19</sup> - 1]

2) [current pc - 2<sup>11</sup>, current pc + 2<sup>11</sup> - 1]

3) lui 指令将绝对地址高20位读取, 存入rs1寄存器

jalr 指令自12位立即数取绝对地址的12位

jalr 跳转时  $pc = rs1 + imm$ , 从而实现任意32位地址跳转

T10 ① 立即数或地址偏移量较小

② 指令使用的某个寄存器是  $x_0, x_1, x_2$

③ 目标寄存器与某个源寄存器相同

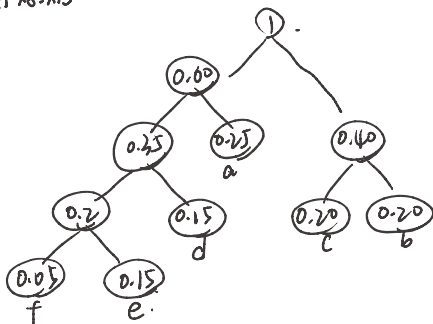
④ 使用的寄存器是8个最常用的  $x_8: x_{15}$

不是所有的 RVC 指令都可以使用32个完整的通用寄存器, CR, CI, CSS 可以

CIW, CL, CS, CA, CB 只占用  $x_8: x_{15}$

T18

$q_i$	$p_i$
a	0.25
b	0.20
c	0.20
d	0.15
e	0.15
f	0.05



$$L = \sum p_i l_i = (0.25 + 0.2 + 0.2) \times 2 + 0.15 \times 3 + (0.15 + 0.05) \times 4 = 2.55$$

$$R = 1 - \frac{-\sum p_i \log_2 p_i}{2.55} = 0.033$$

T19. 1) 当函数调用层数过多时, 栈空间中的内存会被不断地分配. 直到栈空间被耗尽. 此时再次调用函数时就会发生栈溢出. 即指向向栈中写入已超出限定长度的数据. 溢出的数据会覆盖栈中的其他数据, 从而影响程序运行

2) ① 减少函数的嵌套调用层数, 优化代码逻辑.

② 尽量避免使用递归函数.

③ 使用指针作为结构参数.

④ 多线程程序可增加线程栈大小.

T20.

20. 假设有三个函数: F1、F2 和 F3。其中 F1 包含 1 个输入参数, 计算过程使用寄存器 t0 和 s0; F2 包含 2 个输入参数, 计算过程使用寄存器 t0-t1 及 s0-s1, 返回一个 int 值。F1 执行过程中会调用 F2, F2 执行过程中会调用 F3。下表模拟了 F1 执行过程中栈的内容, 其中第一行为 F1 函数被首次调用时 sp 寄存器指向的位置。请在表中填入当 F2 函数首次调用 F3 前栈内保存的可能内容, 并在每行的括号内标注该值是被哪个函数所保存的。第一行的内容已经给出。(可根据需要增删行数)

ra (F1)
g0 (F1)
a0 (F1)
t0 (F1)
ra (F2)
s0 (F2)
a0 (F2)
a1 (F2)