

3月28日

9. (1) 可跳转范围  $[PC - 2^{19}, PC + 2^{19} - 1]$

(2) 可跳转范围  $[PC - 2^{11}, PC + 2^{11} - 1]$

(3) 可以, 可用 `lui` 指令将要跳转的地址的高20位存到PC寄存器, 然后再用 `jalc` 指令加上要跳转地址的低12位, 实现32位绝对地址的跳转。

10. 条件需要满足下述条件之一:

① 立即数或地址偏移量很小

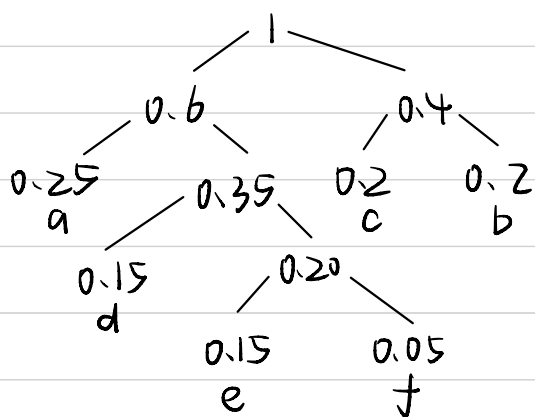
② 其中一个寄存器是 `X0/X1/X2`

③ 目标寄存器和第一操作数寄存器是相同的

④ 所使用的寄存器是最流行的8个

不是, `CR`, `CL`, `CS` 可以使用32个寄存器中任意一个, 但 `CW`, `CL`, `CS`, `CB` 仅被限制在其中8个以内。

18.



at

a

b

c

d

e

f

pt

0.25

0.20

0.20

0.15

0.15

0.05

et

2

2

2

3

4

4

clail

00

01

11

101

1001

1000

$$\text{平均长度} = \sum_{i=1}^7 p_i l_i = 2.55$$

$$\text{信息冗余度 } R = 1 - \frac{-\sum_{i=1}^n p_i \log_2 p_i}{2.55} = 3.2958\%$$

19. 解 (1) 每一次函数调用时都会将当前函数的信息压入栈中, 当调用函数次数过多时, 栈中信息会不断增加, 超出栈的容量导致栈溢出。

(2) ① 减少栈空间需求, 避免定义使用内存较多的变量

② 函数参数中不要传递大型结构、联合对象, 应该使用引用或指针作为函数参数

③ 减少函数调用层次, 慎用递归函数, 例如  $A \rightarrow B \rightarrow C \rightarrow A$  环式调用。

20.	$r_0(F_1)$	
	$t_0(F_1)$	
	$s_0(F_1)$	
	$r_0(F_2)$	
	$t_0(F_2)$	
	$t_1(F_2)$	
	$s_0(F_2)$	
	$s_1(F_2)$	
	$r_0(F_3)$	