

9. 10. 18. 19. 20

9. (1) ~~2²⁰~~ 范围为 $[-2^{20}, 2^{20}-2]$

(2) ~~$[2^{12}, 2^{12}-2]$~~ $[-2^{12}, 2^{12}-2]$

(3) $lwi\ X5, IMM1(\text{得到 } IMM16 \text{ 而 } 20/3)$

$jalr\ X0, IMM2(X5)$

$IMM2 \neq IMM16 \text{ 依 } 12/3$

10.

• 立即数或偏移地址偏移值小

• 寄存器之一 是寄存器 $(X0)$ 。ABI 链寄存器 $(X1)$ 或 ABI 堆栈指针 $(X2)$

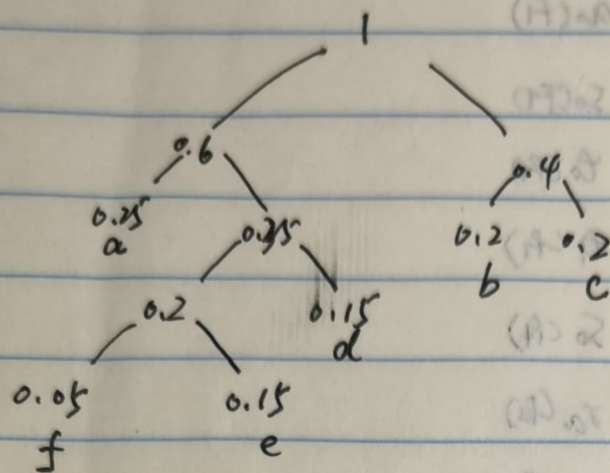
• 目的寄存器和第一个源寄存器相同

CR. C2 和 C5 可使用 32 个 RVI 寄存器中的任何一个

$\angle \text{CCM, CL, CS}$. CB 仅取 8 个

~

18.



19 (1) 每当程序执行进入一个函数调用, 栈就会减一层栈帧, 每当子返回栈就会减一层栈帧, 由于栈大小不是无限的, 而函数的参数是通过 stack 来传递的, 在调用中会占用进程的栈资源, 递归调用在最后的结束点, 才能依次逐层清栈, 如果递归调用层数过多, 就可能导致占用的栈资源超过进程最大值, 从而导致栈溢出。

(2) 尾递归: 一个函数中的递归调用的调用都出现在函数的末尾。

当递归函数调用是整个函数体中最后执行的语句且它的返回值不属于表达式的一部分时, 这个递归就是尾递归。尾递归函数的特点是在递归中不用做任何操作。

(3) 循环替代递归。

20

$r_a(F_1)$

$r_a(F_1)$

$a_0(F_1)$

$a_0(F_1)$

~~$t_0(F_1)$~~

$s_0(F_1)$

$s_0(F_1)$

$t_0(F_1)$

$t_0(F_2)$

$t_1(F_1)$

$t_1(F_2)$

$s_0(F_1)$

$r_a(F_2)$

$s_0(F_1)$

$s_1(F_2)$