

9.10.18 - 19.20

9.11)

范围为  $[-2^{20}, 2^{20}-2]$

(2)

~~$[-2^{12}, 2^{12}-2]$~~   $[-2^6, 2^6-2]$

1.  $x_5, IMM1C$   $IMM16(20\#)$   
将到

jar  $x_6, IMM2(x_5)$

$IMM2 \neq IMM6(20\#)$

10.

• 这部分数据量相对地还偏小

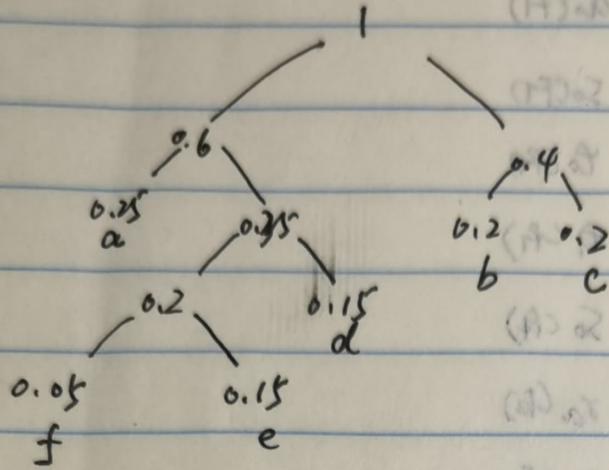
• 因路由器和单一个源寄存器而用

• 寄存器之一是累加器 ( $x_0$ )、AB2转路寄存器 ( $x_1$ )、AB3堆栈指针 ( $x_2$ )

CR, CS 和 CCS 可以用 32 位 RV32I 寄存器中的任何一个

从 CIN, CL, CS, CR 选 8 个

18.



(A) 11

(A) 11

05

(A) 10

(A) 10

05

(A) 12

(A) 12

05

(A) 12

(A) 12

05

(A) 12

(A) 12

05

(A) 11

(A) 11

05

(A) 12

(A) 12

05

(A) 12

(A) 12

05

19 (1) 每当程序执行进入一个函数调用，栈就会加一层栈帧，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，在调用中会占用程序的栈资源，递归调用在最后的语句处，才能依次退栈  
清场，如果递归调用层数过多，就可能导致占用的栈资源超过内存最大值  
从而导致栈溢出。

(2) 尾递归：一个函数中的递归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同

(3) 循环替代递归。

归中不用做任何操作

20

$r_a(F_1)$	$r_{al}(F_1)$
$a_o(F_1)$	$a_o(F_1)$
<del><math>t_o(F_1)</math></del>	$s_o(F_1)$
$s_o(F_1)$	$t_o(F_1)$
$t_o(F_2)$	$t_1(F_1)$
$t_1(F_2)$	$s_o(F_1)$

$r_a(F_2)$

$s_o(F_1)$

$s_1(F_2)$