

9. 10. 12. 18. 19. 20

9. (1) JAL 指令可跳转到当前 PC 所在的地址空间加 (2^9-1) 或减 2^9 字节 内之任何位置

(2) BNE 指令可以向前跳转当前 PC - 2^9 字节 或向后跳转当前 PC + (2^9-1) 字节之位置

(3) 可以, 将 32 位绝对地址拆成 高 20 位 (high-address), 低 12 位 (low-address)

实现: lui x1, high-address # 将高位地址放入 X1 寄存器

jalr x0, x1, low-address # 将低位地址放入并实现跳转

10. 条件: 1. 指令只涉及到 16 个通用寄存器中的其中一个或两个寄存器

2. 指令只涉及到 12 位立即数

3. 指令没有使用特权操作码或保留操作码

且不是 32 个通用整型寄存器都能用, 大部分可用, 一般可用 x8-x15 和 x18-x25

12. (1) Linux Kernel 机器模式

(2) Boot ROM 机器模式

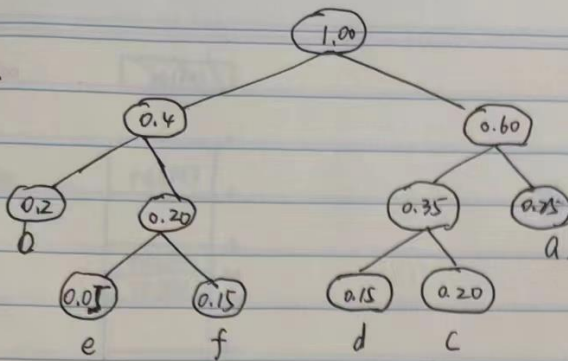
(3) Boot Loader 管理员模式

(4) USB Driver 管理员模式或用户模式

(5) vim 用户模式

18

18.



a_i	$C(a_i)$	l_i	p_i
a	11	2	0.25
b	00	2	0.2
c	101	3	0.2
d	100	3	0.15
e	010	3	0.15
f	011	3	0.15

$$H = -\sum p_i \log_2(p_i) = 2.466$$

$$L_H = \sum p_i l_i = 2 \times 0.25 + 2 \times 0.2 + \dots + 3 \times 0.15 = 2.55,$$

~~冗余度 = 2.55 - 2.466 = 0.084~~

$$R = \frac{L_H - H}{L_H} = 0.031$$

19. 栈溢出指的是在使用栈空间时,将栈指针超过了栈的最大空间限制,导致栈的数据和指针覆盖其他内存空间,从而引发程序错误或崩溃

措施: 1. 尽可能避免函数嵌套调用层数过多,可以通过优化代码结构,减少递归调用

2. 增加栈空间的大小,申请更多的栈空间

3. 使用堆内存代替栈内存

4. 使用尾递归,将递归调用转为循环调用,避免递归层数过多

20.

ra(F1)	0
	8
	16
to(F1)	20
so(F1)	24
	32
ra(F2)	40
to(F2)	44
tl(F2)	48
so(F2)	52
sl(F2)	56
	64
ra(F3)	72