

9. (1) 指令跳转范围为 $\pm 1MB$ 的地址范围.

(2) $\pm 4KB$

(3) 可以 ($ui\ rs1, imm32$)

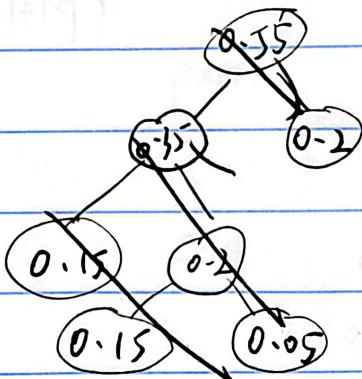
$jalr\ rd, rs1, imm32$

10. 条件: 仅适用于整数指令集, 仅适用于常见指令,
操作数必须在特定范围内, 需要对齐.

在 RVC 指令集中, 除了少部分例外情况, 各类型指令有如下:

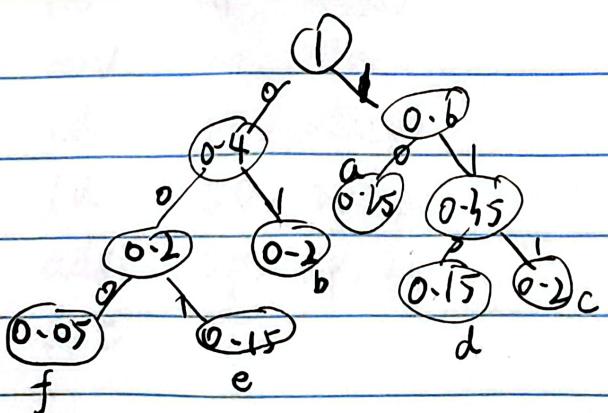
使用完整的 32 个通用寄存器. 例如: $Move.W$ 只能用
 $X8, X9, X10, X11$, 要求 4 字节对齐的指令必须是偶数寄存器.

a_i	p_i
a	0.25
b	0.2
c	0.2
d	0.15
e	0.15
f	0.05



平均长度:

$$\begin{aligned} a: & \cancel{0.10} \\ b: & 0.1 \\ c: & 1.1 \\ d: & 1.0 \\ e: & 0.01 \\ f: & 0.00 \end{aligned}$$
$$\bar{L} = \sum_{i=1}^6 p_i l_i$$
$$= 0.5 + 0.4$$
$$+ 0.6 + 0.45$$
$$+ 0.45 + 0.15$$
$$= 0.9 + 0.9 + 0.6 + 0.15$$
$$= 1.8 + 0.6 + 0.15$$
$$= 2.4 + 0.15 = 2.55$$



$$R = -\frac{\sum_{i=1}^6 p_i \log_2 p_i}{2.55} = 0.03296$$

19.

① 在函数不断地被一层层调用中，返回地址和局部变量等数据会被压入栈中，越来越多，直到超过了栈的容量，导致栈溢出，栈溢出可能会导致程序崩溃，因为栈中数据覆盖了其他重要的数据。

② 1. 减少函数嵌套的层数 2. 增大栈的容量 3. 使用动态内存分配 4. 使用栈溢出保护机制 5. 优化代码。

20.

ra(F_1)

so(F_1)

Paral(F_1)

~~Temp~~ Parato(F_1)

~~Temp~~ so(F_1)

ra(F_2)

so(F_2)

Paral(F_2)

Paral(F_2)

to(F_2)

t1(F_2)

si(F_2)