

第六周作业

9. 回答以下问题:

- 1) jal 指令包含 20 位的有符号立即数编码 (J-type), 该指令相较当前 PC 可以跳转的地址空间范围为多少?
- 2) 条件分支指令 (如 bne) 包含 12 位的有符号立即数编码 (B-type), 这类指令相较当前 PC 可以跳转的地址空间范围为多少?
- 3) 是否可以使用一条 lui 指令和一条 jalr 指令的组合完成任意 32 位绝对地址的跳转操作?

(1) jal : $\text{next pc} \leftarrow \text{current pc} + \text{sign_extend}(\text{imm20} \ll 1)$

即加上一个 21 位二进制数能表示的数: $-2^{20} \sim 2^{20}-1$

指令可跳转范围为 $\pm 1\text{MB}$ 的地址空间.

(2) bne : $\text{next pc} \leftarrow \text{current pc} + \text{sign_extend}(\text{imm12} \ll 1)$

即加上一个 13 位二进制数能表示的数: $-2^{12} \sim 2^{12}-1$

指令可跳转范围为 $\pm 4\text{KB}$ 的地址空间.

(3) 可以. 绝对地址: $\text{address}[31:0]$

lui to $\text{address}[31:12]$

将高 20 位加载至 to 中.

jalr ti, to, $\text{address}[11:0]$

跳转至 to + 低 20 位的绝对地址处

10. 调查 RVC 压缩指令集的编码, 说明一条常用的 32 位指令能够被压缩为 16 位 RVC 指令的条件是什么? RVC 中各类型的指令是否都可以使用完整的 32 个通用整型寄存器?

(1) 常用 32 位指令能被压缩条件:

① 常常只访问寄存器堆的某几个.

② 目标寄存器常为源寄存器之一, 或访问的寄存器数量少.

③ 立即数较小

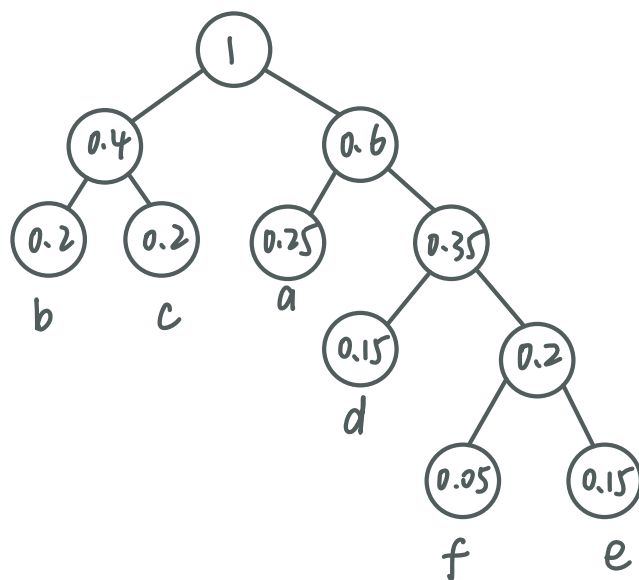
(2) 不能. 由于指令长度减小, 某些 RVC 指令寄存器地址只有 3 位, 即只能访问其中 8 个寄存器.

18. 有一组操作码，它们的出现几率如下表所示。

a_i	p_i
a	0.25
b	0.20
c	0.20
d	0.15
e	0.15
f	0.05

请按照霍夫曼编码对这组操作码进行编码，计算操作码的平均长度和信息冗余度。

a_i	p_i	l_i	$cl(a_i)$
a	0.25	2	10
b	0.2	2	00
c	0.2	2	01
d	0.15	3	110
e	0.15	4	1111
f	0.05	4	1110



平均长度： $\sum_{i=1}^6 p_i l_i = 0.5 + 0.4 + 0.4 + 0.45 + 0.8 = 2.55$

最短平均长度： $H = -\sum_{i=1}^6 p_i \log_2(p_i)$
 $= 2.47$

信息冗余度： $R = 1 - \frac{H}{2.55} = 1 - \frac{2.47}{2.55} = 3.1\%$

19. 回答以下问题：

- 1) 当函数嵌套调用层数过多（例如递归陷入死循环时），可能会造成栈溢出，请简述其原理。
- 2) 有什么办法可以缓解或避免特定情况下的栈溢出问题？

(1) 原理：每调用一个子函数都会开辟一段栈空间，来存放该函数的返回地址及临时变量等数据，当嵌套调用层数过多时，存放数据超出栈总空间容量，则发生上溢。

(2) 避免方法:

- ① 减少每层函数使用的栈空间.
- ② 对占用较大空间的参数应传指针.
- ③ 用循环代替递归函数, 减少函数调用层次.

20. 假设有三个函数: F1、F2 和 F3。其中 F1 包含 1 个输入参数, 计算过程使用寄存器 t0 和 s0; F2 包含 2 个输入参数, 计算过程使用寄存器 t0-t1 及 s0-s1, 返回一个 int 值。F1 执行过程中会调用 F2, F2 执行过程中会调用 F3。下表模拟了 F1 执行过程中栈的内容, 其中第一行为 F1 函数被首次调用时 sp 寄存器指向的位置。请在表中填入当 F2 函数首次调用 F3 前栈内保存的可能内容, 并在每行的括号内标注该值是被哪个函数所保存的。第一行的内容已经给出。(可根据需要增删行数)

ra (F1)
s0 (F1)
a0 (F1)
t0 (F1)
ra (F2)
s0 (F2)
a0 (F2)
a1 (F2)
t0 (F2)
t1 (F2)

F1: ra caller
s0 callee
t0 caller (F2使用t0, 调用前需保存)
a0 caller

F2: ra
s0
a0
a1 > caller (F3可能使用, 需保存)
t0-t1
s1: callee (F1没使用s1, 不需保存)