第2章习题

9. 回答以下问题

1) jal指令包含20位的有符号立即数偏移量(J-type)，该指令相较当前PC可以跳转的地址空间范围是多少？

Jal指令使用20位有符号立即数，表示这个偏移量会被加在当前PC上进行跳转，范围是±1MiB（$1\text{MiB} = 2^{20}$ bytes）

2) 条件分支指令（如bne）包含12位的有符号立即数偏移量(B-type)，这类指令相较当前指令可以跳转的地址空间范围是多少？

条件分支指令的跳转范围为±4KiB（$1\text{KiB} = 2^{10}$ bytes）

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？

jalr的包含12位立即数偏移量，为I-type格式，而LUI的立即数偏移量为20位的U-type

lui指令可以将高20位指令放在目标寄存器rd中，并将低12位置0，因此可以使用lui和jalr指令的组合完成任意32位绝对地址的跳转。首先将该地址的高20位加载进一个寄存器中，再使用jalr指令跳转到该寄存器与jalr立即数表示的低12位偏移相加后的地址，即可实现任意32位绝对地址的跳转操作。

例如，设绝对地址为addr[31:0]

```
lui t0, addr[31:12]
jalr x0, addr[11:0](t0)
```

可完成操作。

10. 调查RVC压缩指令集的编码，说明一条常用的32位指令能够被压缩到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？

RVC压缩的条件为以下之一：

① 立即数或地址偏移量较小。

② 其中一个寄存器为x0，ABI linker register（链接寄存器）x1或ABI指针寄存器（x2）

③ 目标寄存器与第一个源寄存器相同

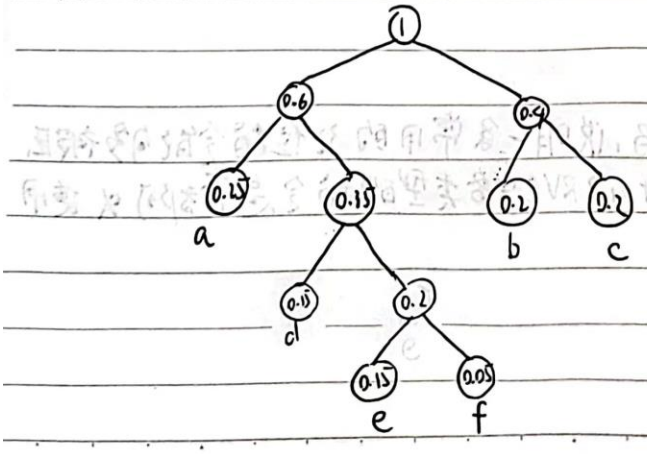
④ 使用8个最常用的寄存器

并非所有类型的RVC指令都可以使用完整的32个通用整型寄存器，例如CR、CI、CSS可使用任意32个RVI寄存器，但CIW、CL、CS、CB则仅可使用其中8个通用整型寄存器（映射到x8-x15）

18. 有一组操作码，它们的出现几率如下表所示：

ai	pi
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数进行编码，计算操作码的平均长度和信息冗余度。



平均长度计算如下：

ai	li	Pi
a	2	0.15
b	2	0.20
c	2	0.40
d	3	0.15
e	3	0.15
f	4	0.05

操作码平均长度 $\sum_{i=1}^{6} Pili = 2.40$

信息冗余量 $R = 1 - \frac{-\sum_{i=1}^{6} Pi\log_2 Pi}{\lceil \log_2 n \rceil} = 1 - \frac{2.466}{\lceil 2.587 \rceil} \approx 18\%$

19. 回答以下问题：

1) 当函数嵌套调用层数过多（如递归陷入死循环）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) 程序使用栈来管理过程所需要的存储空间，在调用函数时，程序会为将要执行的函数的局部变量等调用信息分配新的栈帧，并在调用结束后释放。然而，如果函数的写入数据大小未被良好控制，例如递归死循环使得嵌套过多，就会不断累积栈帧，直到消耗完整个栈空间，这就是栈溢出，它可能造成未定义行为或导致程序崩溃。

2) 避免与缓解方法：

① C语言编程时一些函数无法限制读入字符串的长度，例如gets等导致内存溢出的函数，应使用更安全的函数如fgets等方法规范编码习惯，避免栈溢出。

② 编写递归函数时，可限制递归深度，确保函数可以终止。

③ 现代编译器和操作系统还使用了很多机制来避免栈溢出攻击，例如栈随机化（栈的位置在每次程序运行时都有变化）、栈破坏检测（添加"栈保护者"，检查这个值是否被改变）、限制可执行代码区域等方法。

20. 假设有三个函数:F1、F2和F3,其中F1包含1个输入参数,计算过程中使用寄存器t0和s0;F2包含2个输入参数,计算过程中使用寄存器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F2首次调用F3时栈内保存的可能内容,并在每行括号内标注该值时被哪个函数所保存的,第一行内容已给出。

t0、t1为caller saved,而s0、s1为callee saved

栈结构如下:

ra(F1)

t0(F1)

s0(F1)

ra(F2)

t0(F2)

t1(F2)