

9.

- (1) 该指令相较当前 PC 可以跳转的地址空间范围为 $\pm 2^{19}-1$
- (2) 该指令相较当前 PC 可以跳转的地址空间范围为 $\pm 2^{11}-1$
- (3) 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，首先，lui 指令用于将高 20 位的绝对地址加载到一个寄存器中。然后，jalr 指令用于将低 12 位的偏移量加载到同一寄存器中，并跳转到该地址。可以执行以下指令：

lui x5, %hi(address)

jalr x0, x5, %lo(address)

10.

一条常用的 32 位指令能够被压缩为 16 位 RVC 指令的条件是：

1. 指令小于等于 32 位，指令中不含 R 型指令、J 型指令、Fence 指令和访存指令等特殊指令。
2. 操作数编码有效。
3. 指令位模式特定。

在 RVC 中，所有指令都可以使用完整的 32 个通用寄存器，无论是 16 位指令还是 32 位指令。

18.

第六周作业

18.

解：霍夫曼编码，考虑

$l_a = l_b = 2, l_c = l_d = l_e = l_f = 3$

操作码平均长度： $L = \frac{(1 \times 0.2 + 2 \times 0.2 + 3 \times 0.15 + 3 \times 0.15 + 3 \times 0.15 + 3 \times 0.2)}{6} = 2.55$

信息冗余度： $1 - \frac{\sum p_i \log_2 p_i}{\log_2 7} = 1 - \frac{-2.3219 \times 0.2 + 0.401 + 2.4 \times 0.15 + 2.3219 \times 0.15 + 2.3219 \times 0.15 + 2.3219 \times 0.2}{2.55} \approx 0.046$

19.

- (1) 造成栈溢出的原因是因为每一个函数在调用时都会在栈中开辟一段空间来存储函数的

局部变量、参数、返回地址等信息。但是栈有大小限制，当函数嵌套调用层数太多时，栈空间可能不足以存储所有函数的信息，从而导致栈溢出。

- (2) 1. 增加栈的大小：可以通过修改系统的配置文件或命令行参数，来增加栈的大小。
2. 使用动态内存分配：通过使用动态内存分配函数，将函数中的一些局部变量转换为动态内存分配。这样可以将这些变量存储在堆中，避免占用栈空间。
3. 使用循环代替递归：递归函数可能会导致栈帧的不断增加，随着嵌套层数增加，可能造成栈溢出，因此可以使用循环来替代递归，减少函数嵌套调用层数。
4. 避免过度的参数传递：函数的参数也会占用栈空间，随着参数数量的增加，占用的栈空间也会增加。因此，在编写函数时，需要避免过度的参数传递。

20.

