

嵌入式第六周

9. 由 jal 指令包含 20 位的有符号立即数编码 (I-type), 该指令相较于当前 PC 可以跳转的地址空间范围是多少?

由 jal 指令提供的 20 位的偏移量, 若被符号扩展为 32 位, 则相较于当前 PC 可以跳转地址空间范围是 -2^{20} 到 $2^{20}-1$.

由条件分支指令 (如 bne) 包含 12 位的有符号立即数编码 (B-type), 这类指令可以相较于当前 PC 跳转的地址空间范围是多少?

由偏移量被符号扩展为 13 位, 则跳转范围是 -2^{12} 到 $2^{12}-1$.

是否可以由一条 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.

解: 可以. lui 指令可以将一个 16 位的立即数左移 16 位, 然后存储在目标寄存器的高 16 位, 低 16 位填充为 0. 因此 lui 指令可以用来加载一个 32 位绝对地址的高 16 位.

jalr 用于将程序计数器 PC 设置为目标地址, 并将返回值存储在寄存器中, 通常用于实现过程调用和返回操作. 可以用目标寄存器中值和另一个寄存器中值计算出跳转地址.

如果将目标寄存器设置为 lui 指令的目标寄存器, 将另一个包含低 16 位地址偏移量的寄存器, 那么就计算出 32 位绝对跳转地址.

10. 一条常用的 32 位指令能够被压缩为 16 位 RVC 指令的条件? RVC 指令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 指令编码规则是:

每个 RVC 指令都包含一个 16 位的压缩码和一个可选的 16 位的扩展码. 用于存储操作数. 每个 RVC 指令都对应一个完整的 32 位指令.

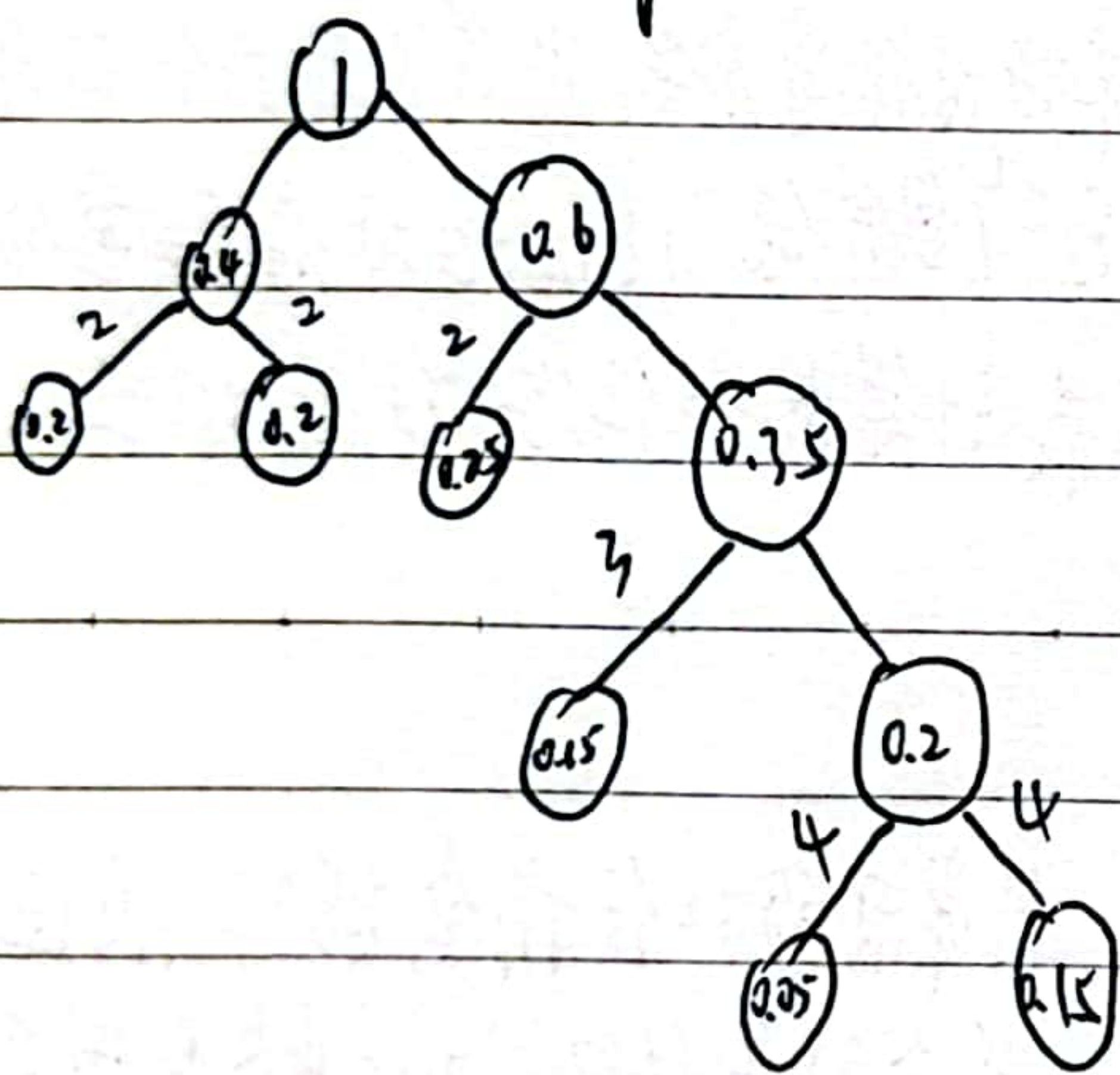
由一条常用的 32 指令. 条件是: 该指令仅使用 3 个通用整型寄存器中的任意一个, 且 没有使用立即数或跳转指令 (?).

1. 条件补充: ① 指令的目标寄存器或源寄存器的编号必须在 0~7 范围内。

② 操作数、偏移量等必须在一定取值范围内, 比如立即数只能为 0 位, 偏移量只能为 8 位, 等。

③ 并非所有类型指令都可以用 RVC 指令集实现压缩, 但在 RVC 中, 各类型的指令都可以使用完整的 32 个通用寄存器 (整型)。

18. 解: 构造 Huffman 树如下:



19. ① 栈溢出是由于函数调用过程中, 栈空间不足以存储所有的函数调用栈帧 (除

函数的参数、局部变量、返回地址等信息,

即写入了栈空间以外的内存区域, 递归陷入死循环时, 函数会不断调用自身, 不释放

栈上的内存, 导致无法继续调用函数。

② ① 尽可能减少递归调用次数或层数。

② 针对递归算法, 可以使用循环或迭代方法来避免栈溢出。

③ 增加栈空间大小; 将局部变量改为全局变量或静态变量。

④ 改为非递归算法。

⑤ 使用动态内存分配, 使用堆内存代替栈内存, 但要手动释放。

20. $sp \rightarrow ra(F_1)$

$t_0(F_1)$

$s_0(F_1)$

$arg(F_1)$

$ra(F_2)$

$t_0(F_2)$

$s_0(F_2)$

$t_1(F_2)$

$s_1(F_2)$

$arg_0(F_2)$

$arg_1(F_2)$

$$\bar{L} = \sum_{i=1}^6 p_i l_i = 4 \times 0.05 + 4 \times 0.15 +$$

$$3 \times 0.15 + 2 \times (0.2 + 0.2 + 0.25) = 2.55$$

$$R = 1 - \frac{-\sum_{i=1}^n p_i \log_2 p_i}{\bar{L}} = 0.033$$