

计算机组成原理II —— 第一次作业

计科一班 王昊恩 3018216021

本次作业源代码已经上传至 [github](#) 。

[github](#) 链接：

1. 本章书后习题 2.63 。

- 将下面的C函数代码补充完整。函数 `srl` 用算术右移（由值 `xsra` 给出）来完成逻辑右移，后面的其他操作不包括右移或者除法。函数 `sra` 用逻辑右移（由值 `xsrl` 给出）来完成算术右移，后面的其他操作不包括右移或者除法。可以通过计算 `8*sizeof(int)` 来确定数据类型 `int` 中的位数 w 。位移量 k 的取值范围为 $0 \sim w-1$ 。

```
1 unsigned srl(unsigned x,int k) {
2     /*Perform shift arithmetically*/
3     unsigned xsra = (int) x >> k;
4
5     int w = 8 * sizeof(int);
6     int mask = (1 << (w - k)) - 1; //前k位为0, 后w-k位为1
7     return xsra & mask; //前k位置为0, 后k位不变
8 }
9
10 int sra(int x, int k) {
11     /*Perform shift logically*/
12     int xsrl = (unsigned) x >> k;
13
14     int w = 8 * sizeof(int);
15     int sign_mask = xsrl & (1 << (w - k - 1)); //判断符号位为1还是0
16     int mask = (-sign_mask) << (w - k);
17     //如果符号位为1, mask为111...1100000...000000
18     //如果符号位为0, mask为000...0000000...000000
19     return xsrl | mask; //前k为改为mask前k位, 实现符号位扩展
20 }
```

2. 假设尾数部分位宽为 n ，阶码位宽为 k 。请给出最小/最大非规格化负数、最小/最大规格化整数、 -1 的数值表达式。

名称	数值表达式
最小非规格化负数	$(-1) \times (1 - 2^{-n}) \times 2^{-2^{k-1}+2}$
最大非规格化负数	$(-1) \times 2^{-n-2^{k-1}+2}$
最小规格化整数	$(-1) \times (1 - 2^{-n-1}) \times 2^{2^{k-1}}$
最大规格化整数	$(1 - 2^{-n-1}) \times 2^{2^{k-1}}$
-1	$(-1) \times (2^0) \times 1$

3. 本章书后习题 2.87 。

- 2008版 IEEE 浮点标准，即 IEEE 754-2008，包含了一种 16 位的“半精度”浮点格式。他最初是由计算机图形公司设计的，其存储的数据所需的动态范围要高于 16 位整数可获得的范围。这种格式具有 1 个符号位、5 个阶码位 ($k=5$) 和 10 个小数位 ($n=10$)。阶码偏置量是 $2^{5-1} - 1 = 15$ 。

对于每个给定的数，填写下表，其中，每一列具有如下指示说明：

Hex: 描述编码形式的 4 个十六进制数字。

M: 尾数的值。这应该是一个形如 x 或 $\frac{x}{y}$ 的数，其中 x 是一个整数，而 y 是 2 的整数幂。例如：0、 $\frac{67}{64}$ 和 $\frac{1}{256}$ 。

E: 阶码的整数值。

V: 所表示的数字值。使用 x 或 $x \times 2^z$ 表示，其中 x 和 z 都是整数。

D: (可能近似的) 数值，用 `printf` 的格式规范 `%f` 打印。

- 举一个例子，为了表示数 $\frac{7}{8}$ ，我们有 $s=0$ ， $M=\frac{7}{4}$ 和 $E=-1$ 。因此这个数的阶码字段为 01110₂ (十进制值 $15-1=14$)，尾数字段为 1100000000₂，得到一个十六进制的表示 3B00。其数值为 0.875。

标记为“—”的条目不用填写。

以下 *D* 输出已经过验证，浮点数默认保留六位小数。

