对阶:小阶看齐大阶,阶码小的尾数右移一位,阶加一,直到阶码相等 尾数求和 尾数按照定点数加减规则运算 最高数值位与符号位不同即为规格化形式 规格化 左规:尾数左移1位,和的阶码减1 直到00.1XX或者11.0XXX 右规,尾数求和结果溢出(10.XXX或者01.XXX) 尾数右移一位,和的阶码加1 尾数右移时,移去的最高数值位为0,则舍去 可能会导致溢出,此时需要再一次右规 0舍1入法 尾数左移时,移去的最高数值位为1,则尾数末位加1 舍入 运算步骤 恒置1法 尾数右移,不论最高数值位丢掉的是1还是0,都将尾数末位恒置为1 只有右规后,仍然溢出,此时才是真正溢出 溢出判断 上溢出:进入中断处理 2.3.2 浮点数的加减运算 下溢出:按机器零处理 char-->int 在前面补0 int <---->unsigned 彼此都可能因为溢出丢失数据 强制类型转换 float转换为int可能会出现精度损失和溢出 int<---->float int转换为float 可能会出现数据舍入 现代计算机通常是按字节编址,即每个字节对应1个地址 边界对齐 通常也支持按字、按半字、按字节寻址