(一)数制与编码

- # (一)数制与编码
  - ## 1.进位计数制及其相互转换
    - ### 1）概念
      - ①**基数**：每个数位所用到的不同数码的个数。
      - ②**位权**：每个数码所表示的数值等于该数码本身乘以一个与它所在数位有关的常数。
      - ③一个进位数的数值大小就是它各位数码按权相加。
      - ④小数是离散的，并不是每个十进制小数都可以准确地用二进制表示；任意一个二进制小数都可以用十进制小数表示。
    - ### 2）使用二进制原因
      - ①只有两种状态，两个稳定状态的物理器件就可以表示二进制数的每一位，制造成本比较低。
      - ②1和0正好与逻辑值“真”和“假”对应，实现逻辑运算和程序中的逻辑判断提供了便利条件。
      - ③编码和运算规则都很简单，通过逻辑门电路能方便地实现算术运算。
    - ### 3）十进制转换
      - ①**除基取余法**(整数部分)：整数部分除基取余，最先取得的余数为数的最低位，最后取得的余数为数的最高位(即除基取余，先余为低，后余为高)，商为0时结束。
      - ②**乘基取整法**(小数部分)：小数部分乘基取整，最先取得的整数为数的最高位，最后取得的整数为数的最低位(即乘基取整，先整为高，后整为低)，乘积为1.0(或满足精度要求)时结束。
  - ## 2.定点数的表示

原码、反码、补码、移码，各表示范围（字长为n+1）			
原码（反码）	纯小数：$-(1-2^{-n})\sim 1-2^{-n}$	纯整数：$-(2^n-1)\sim 2^n-1$	
补码	纯小数：$-1\sim 1-2^{-n}$	纯整数：$-2^n\sim 2^n-1$	为了消除减法
移码	移码全0，真值最小值-2^n	移码全1，对应最大值2^n-1	可以直接反应真值大小

    - ### 1）定点数的编码表示
      - ***通常用定点补码整数表示整数，用定点原码小数表示浮点数的尾数部分，用移码表示浮点数的阶码部分。***
      - ①**原码**：真值零的原码表示有正零和负零两种形式。
      - ②**补码**：零的补码是唯一的，***小数补码比原码多表示一个“-1”***，整数补码比原码多表示一个“-(2^n)”。从补码到真值和从真值到补码都是按位取反再加一。
      - ③**移码**：形式上等于将补码的符号位取反，数字上移码=真值+偏移值，其中偏移值=2^(n-1)，如8位移码的偏移值为128。

- 2）移码特点
  - ① 常用来表示浮点数的阶码(整数)，只能表示整数。
  - ② 移码保持了数据原有的大小顺序，移码大真值就大。
  - ③ **定义：**在真值x上加上一个常数(偏置值)，取2^n。[x]移=2^n+x，其中机器字长n+1，相当于向右移位2^n。
- 3）关于0的表示
  - (1) 表示唯一
    - ① **补码：**[+0]补=[-0]补=0.0000。
    - ② **移码：**[+0]移=2^n+0=[-0]移=2^n-0=100…0（n个“0”）；移码中“1”表示正，“0”表示负。
  - (2) 表示不唯一
    - ① **原码：**[+0]原=00000；[-0]原=10000
    - ② **反码：**[+0]反=0,0000；[-0]反=1,1111

3.定点数的运算

- 1）移位运算
  - (1) 原则
    - 当某个十进制数相对于小数点做n位左移，相当于该数乘以10；
    - 右移，相当于该数除以10。
  - (2) 分类
    - ① **算术移位：**针对有符号数，移位过程中，符号位保持不变。
      - ●对于正数，原码=反码=补码，不论左移还是右移都是补0。
      - ●对于负数，原码左移和右移补0，反码左移和右移补1，补码由于左边和反码相同，右边和原码相同，所以补码左移补0，右移补1。
      - ●由于位数有限，所以有时候算数移位并不能完全等效于乘除运算。
    - ② **逻辑移位：**针对无符号数。符号位参与，左移右移都添0。
      - ●逻辑左移时，高位移丢，低位补0。
      - ●逻辑右移时，低位移丢，高位补0。
    - ③ **循环移位：**特别适合将数据的低字节数据和高字节数据对换。
      - **分类：**不带进位标志位CF的循环移位；带进位标志位CF的循环移位。
  - (3) 总结

	码制	添加代码	
正数	原、补、反码	0	
负数	原码	0	使符号位不变，其空位均添加0
	补码	左移添0(同原码)	左移，空位出现在低位，补0
		右移添1(同补码)	右移，空位出现在高位，补1
	反码	1	移位后添加与原码相反，全添加1

- ① **对于正数：**
  - ●左移时最高位丢1，结果出错；右移时最低数位丢1，影响精度。
- ② **对于负数：**
  - ●原码，真值与原码一样，左移丢1，出错；右移最低数位丢1，影响精度。
  - ●反码，左移丢0，结果出错；右移丢0，影响精度。
  - ●补码，右移同原码，丢1，影响精度；左移同反码，丢0，结果出错。

- **2）溢出概念和判别方法**
  - **(1) 定义**
    - **上溢(正溢出)**：大于机器所能表示最大正数。
    - **下溢(负溢出)**：小于机器所能表示最小负数。
    - **定点小数**：①两个定点小数相加大于/等于1→上溢；②两个定点小数相加小于-1→下溢
  - **(2) 特点**
    - 仅当两个符号相同的数相加/两个符号相异数相减，才可能产生溢出，数值位“跑”到符号位中，改变了符号性质
    - ① 对于加法{正+正，负+负}；对于减法{正-负，负-正}
      - 不论加减，两个操作数符号相同，结果与原操作数符号不同→溢出
    - ② 数值部分最高位进位和符号位产生进位，进行异或：为1，溢出；为0，无溢出
  - **(3) 双符号位**（模4补码，最高位为真正符号）
    - **a)特点：**
      - ① 不但可以检测是否溢出，可以检测正溢出/负溢出；
      - ② 两位符号位连数值部分一起参加运算；
      - ③ 高位符号位产生进位直接丢弃
    - **b)原则**
      - ① 00——结果为正→未溢出；11——结果为负→未溢出；10——结果应为负→负溢出；01——结果应为正→正溢出。
      - ② 两位异或：为1→溢出；为0→无溢出。
      - ③ 存储器/寄存器中只需保存一位，相加时，两位同时送到加法器输入端。
  - **(4) 乘法溢出**
    - **a) 2n位保存结果**
      - nbit 乘 nbit 结果一定能用 2n bit表示，不会发生溢出。
    - **b) 只保留后n位**
      - ①**无符号数**：前 nbit 全为0时，舍弃前 nbit 不影响结果的真值，故不发生溢出；前nbit只要有出现1时，舍弃前nbit就会影响结果的真值，故发生溢出；当发生溢出时，PSW的OF同样会被置为1。

- ②**有符号数**：前n+1bit全为0/1，舍弃前nbit不影响结果的真值，故不发生溢出；前nbit不是全为0/1，舍弃前nbit就会影响结果的真值，故发生溢出。
- 原因是补码的符号扩展：当n位补码最高位为0时，扩展为2nbit，则扩展的nbit为全0；当n位bit最高位为1时，扩展为2nbit，则扩展的nbit全为1；故前n+1bit全0/全1时，舍弃前nbit不影响真值。