

# 计算机系统结构官方笔记

## 一、思维导图



## 二、知识点回顾

### 1、数据表示与数据结构

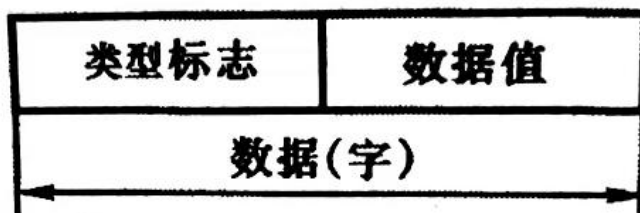
数据表示指的是能由计算机硬件识别和引用的数据类型,表现在它有对这种类型的数据进行操作的指令和运算部件。

早期的计算机只有定点数据表示,要想用浮点数就得用两个定点数来分别表示其阶码和尾数。

### 2、自定义数据表示

自定义 (Self-defining) 数据表示包括标志符数据表示和数据描述符两类。

#### (1) 标志符数据表示



主要优点:

- 1) 简化了指令系统和程序设计
- 2) 简化了编译程序

- 3) 便于实现一致性校验
- 4) 能由硬件自动变换数据类型
- 5) 支持数据库系统的实现与数据类型无关的要求, 使程序不用修改即可处理多种不同类型的数据
- 6) 为软件调试和应用软件开发提供了支持。

## 2) 数据描述符和标志符的差别 (简答)

标志符是和每个数据相连的, 合存在一个存储单元中, 描述单个数据的类型特征; 数据描述符则是与数据分开存放, 用于描述所要访问的数据是整块的还是单个的, 访问该数据块或数据元素所要的地址以及其他信息等。

描述符			
101	各种标志位	长度	地址

数据	
000	数 据

## 3、向量、数组数据表示

为向量、数组数据结构的实现和快速运算提供更好的硬件支持的方法是增设向量、数组数据表示, 组成向量机。

## 4、堆栈数据表示

堆栈计算机表现于 (特点):

- 1) 由高速寄存器组成的硬件堆栈, 并附加控制电路, 让它与主存中的堆栈区在逻辑上构成整体, 使堆栈的访问速度是寄存器的, 容量是主存的。
- 2) 有丰富的堆栈操作指令且功能很强, 可直接对堆栈中的数据进行各种运算和处理。
- 3) 有力地支持了高级语言程序的编译。
- 4) 有力地支持了子程序的嵌套和递归调用。

## 5、确定计算机的数据表示

原则 1 看系统的效率是否有显著提高, 包括实现时间和存储空间是否有显著减少。实现时间是否减少又主要看主存和处理机之间传送的信息量是否减少。

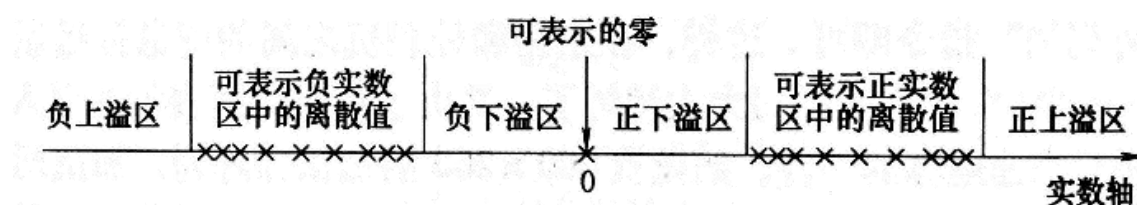
传送的信息量越少，实现时间就越少。

原则 2 看引入这种数据表示后，其通用性和利用率是否提高。如果只对某种数据结构的实现效率很高，而对其他数据结构的实现效率很低，或者引入这种数据表示在应用中很少用到，那么为此所耗费的硬件过多却并未在性能上得到好处，必然导致性能价格比的下降，特别是一些复杂的数据表示。

## 6、浮点数尾数基值的选择

阶码部分包含了阶符和阶值两部分。阶码部分可用原码、补码或增码（也称移码）表示。不管怎么表示， $p + 1$  位阶码部分中影响阶值大小的实际只有  $p$  位。

数学中实数在数轴上是连续分布的。但由于计算机字长有限，浮点数只能表示出数轴上分散于正、负两个区间上的部分离散值，如图 2-2 所示。



浮点数阶值的位数  $p$  主要影响两个可表示区的大小，即可表示数的范围大小，而尾数的位数  $m$  主要影响在可表示区中能表示值的精度。由于计算机中尾数位数限制，实数难以精确表示，因此，不得不用较接近的可表示数来近似表示，产生的误差大小就是数的表示精度。

【重点】 $m' = m / \log_2 r_m$  ( $r_m$ 、 $m$  都是已知的)

表 2-1 采用尾基为  $r_m$  的浮点数表示的特性及其举例

条件: 非负阶、正尾数、规格化	阶值: 二进制 $p$ 位 尾数: $r_m$ 进制 $m'$ 位	若 $p=2, m=4$	
		当 $r_m=2$ (即 $m'=4$ ) 时	当 $r_m=16$ (即 $m'=1$ ) 时
可表示最小尾数值	$1 \times r_m^{-1}$ (即 $r_m^{-1}$ )	1/2	1/16
可表示最大尾数值	$1 - 1 \times r_m^{-m'}$ (即 $1 - r_m^{-m'}$ ) 特例: $1 - 2^{-m}$	15/16	15/16
最大阶值	$2^p - 1$	3	3
可表示最小值	$r_m^0 \times r_m^{-1}$ (即 $r_m^{-1}$ )	1/2	1/16
可表示最大值	$r_m^{(2^p-1)} \times (1 - r_m^{-m'})$ 特例: $r_m^{(2^p-1)} \times (1 - 2^{-m})$	7.5	3840
可表示的尾数个数	$r_m^{m'} \times (r_m - 1) / r_m$ 特例: $2^m \times (r_m - 1) / r_m$	8	15
可表示阶的个数	$2^p$	4	4
可表示数的个数	$2^p \times r_m^{m'} \times (r_m - 1) / r_m$ 特例: $2^p \times 2^m \times (r_m - 1) / r_m$	32	60

注: 表中特例是指  $r_m$  为 2 的整数次幂时, 用  $r_m^{m'} = 2^m$  代入。

## 7、浮点数尾数的下溢处理方法

### (1) 截断法

截断法是将尾数超出计算机字长的部分截去,

这种方法的好处是实现最简单

### (2) 舍入法

舍入法是在计算机运算的规定字长之外增设一位附加位, 存放溢出部分的最高位, 每当进行尾数下溢处理时, 将附加位加 1 (二进制整数相当于加 0.5, 二进制小数相当于加)。

这种方法的好处是实现简单

### (3) 恒置“1”法

恒置“1”法是将计算机运算的规定字长的最低位恒置为“1”。这种方法的好处是实现最简单, 不需要增加硬件和处理时间, 平均误差趋于 0。

主要缺点是最大误差最大, 比截断法的还要大。

### (4) 查表舍入法

查表舍入法是用 ROM 或 PLA 存放下溢处理表

ROM 表共需  $2^k$  个单元, 地址用  $k$  位二进制码表示, 每个存储单元字长  $k-1$  位。

当存储器  $k$  位地址码之高  $k-1$  位为全“1”时, 对应单元内容填  $k-1$  位全“1” ;  
其余情况按  $k$  位二进制地址码最低位为“0”舍弃, 为“1”进 1 来填  $k-1$  位内容。  
ROM 查表舍入法速度较快, 平均误差可调节到 0, 是较好的方法。  
缺点是硬件量大, 不过随着器件价格的下降和集成度的改进, 使用将会增多。

### 三、练习题

1、四种浮点数尾数下溢处理方法中, 实现最简单的方法是( )1204

- A: 截断法
- B: 舍入法
- C: 恒置“1”法
- D: 查表舍入法

答案: A

2、下列四种浮点数尾数下溢处理方法中, 平均误差可以调节的是( )1404

- A: 截断法
- B: 舍入法
- C: 恒置“1”法
- D: 查表舍入法

答案: D