

尚德机构

# 计算机系统结构

讲师：孙小涵

学习是一种信仰！ IN LEARNING WE TRUST

SUNLAND





# 讲师介绍

- 主讲老师：孙小涵（尚德机构-小涵老师）
- 主讲课程：计算机类、数学类
- 邮箱：sunxiaohan@sunlands.com



# 课程章节

## 计算机系统结构

第1章 计算机系统结构概论

第2章 数据表示、寻址方式与指令系统

第3章 存储、中断、总线与I/O系统

第4章 存储体系

第5章 标量处理机

第6章 向量处理机

第7章 多处理机

第8章 数据流计算机和归约机

## 第2章 数据表示、寻址方式与指令系统

## 第2章 数据表示、寻址方式与指令系统

●	数据表示	★ ★
●	寻址方式	★
●	指令系统的设计和优化	★
●	指令系统的发展和改进	★ ★

## 2.1 数据表示

**本节主要内容：**

**数据表示的定义**

**数据表示与数据结构的关系**

**引入数据表示的原则**

**标志符数据表示的优点，与数据描述符的差别**

**堆栈数据表示、向量数据表示**

**浮点数尾数基值**

**下溢处理**

## 2.1 数据表示

### 1、数据表示与数据结构（填空）

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。

$$-0.1010011 \times 2^{+3}$$

## 2.1 数据表示

### 1、数据表示与数据结构（填空）

数据表示指的是能由计算机硬件（ ）和（ ）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。

$$-0.1010011 \times 2^{+3}$$



## 2.1 数据表示

### 1、数据表示与数据结构（填空）

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。

$$-0.1010011 \times 2^{+3}$$

## 2.1.2高级数据表示

### 1.自定义数据表示（填空，简答）

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。

## 2.1.2高级数据表示

### 1.自定义数据表示（填空，简答）

自定义（Self-defining）数据表示包括（ ）数据表示和（ ）类。



## 2.1.2高级数据表示

### 1.自定义数据表示（填空，简答）

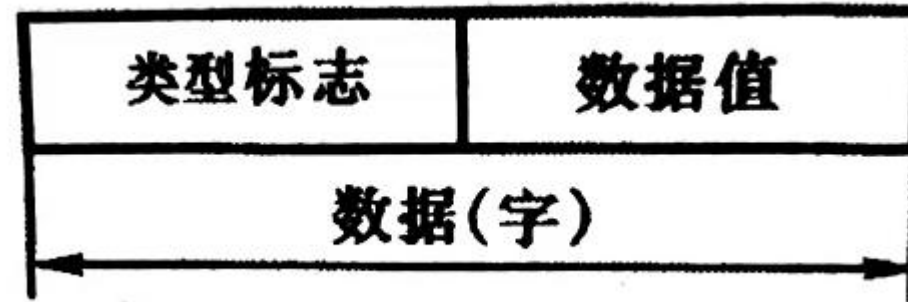
自定义（Self-defining）数据表示包括标志符数据表示和数据描述符两类。



## 2.1.2高级数据表示

### 1.自定义数据表示

#### (1) 标志符数据表示（单选、填空）



## 2.1.2高级数据表示

### 1.自定义数据表示

#### (1) 标志符数据表示（单选、填空）

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。

## 2.1.2高级数据表示

### 1.自定义数据表示

#### (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		

## 2.1.2高级数据表示

### 2.向量、数组数据表示（单选）

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。





## 2.1.2高级数据表示

### 3.堆栈数据表示（单选、简答）

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。

## 2.1.2高级数据表示

### 3.堆栈数据表示（单选、简答）

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（ ）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（ ）。



那么意气风发地  
走在成功的道路上

## 真题练练手

1、自定义数据表示包括标志符数据表示和（ ） 1910

A:标题数据表示

B:数据描述符

C:向量数据表示

D:堆栈数据表示



那么意气风发地  
走在成功的道路上

## 真题练练手

1、自定义数据表示包括标志符数据表示和（ ） 1910

A:标题数据表示

B:数据描述符

C:向量数据表示

D:堆栈数据表示

答案： B



那么意气风发地  
走在成功的道路上

## 真题练练手

2、下列选项中属于高级数据表示的是 ( ) 1810 1904

A:向量

B:定点数

C:逻辑数

D:字符串



那么意气风发地  
走在成功的道路上

## 真题练练手

2、下列选项中属于高级数据表示的是 ( ) 1810 1904

A:向量

B:定点数

C:逻辑数

D:字符串

答案：A



那么意气风发地  
走在成功的道路上

## 真题练练手

3、自定义数据表示又分为\_\_\_\_\_表示\_\_\_\_\_和两类。1204 1304 0804



那么意气风发地  
走在成功的道路上

## 真题练练手

3、自定义数据表示又分为\_\_\_\_\_表示\_\_\_\_\_和两类。1204 1304 0804

答案：标志符数据 数据描述符





那么意气风发地  
走在成功的道路上

## 真题练练手

4、堆栈机器(堆栈数据表示)的特点不包括( ) 1204

A:有力地支持子程序的嵌套和递归调用

B:有丰富的堆栈操作指令且功能强大

C:有力地支持高级语言程序的编译

D:访问堆栈的速度低



那么意气风发地  
走在成功的道路上

## 真题练练手

4、堆栈机器(堆栈数据表示)的特点不包括( ) 1204

A:有力地支持子程序的嵌套和递归调用

B:有丰富的堆栈操作指令且功能强大

C:有力地支持高级语言程序的编译

D:访问堆栈的速度低

答案：D

## 2.1.3引入数据表示的原则

### 确定计算机的数据表示（填空、简答）

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在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。

## 2.1.3引入数据表示的原则

### 确定计算机的数据表示（填空、简答）

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。



那么意气风发地  
走在成功的道路上

## 真题练练手

1、高级数据表示的引入的基本原则是:一是看系统的\_\_\_\_\_是否提高,二是看引入数据表示后其通用性和\_\_\_\_\_是否高。1410



那么意气风发地  
走在成功的道路上

## 真题练练手

1、高级数据表示的引入的基本原则是:一是看系统的\_\_\_\_\_是否提高,二是看引入数据表示后其通用性和\_\_\_\_\_是否高。1410

答案: 效率      利用率

## 2.1.4浮点数尾数基值大小和下溢处理方法的选择

### 1.浮点数尾数基值的选择（计算）

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。

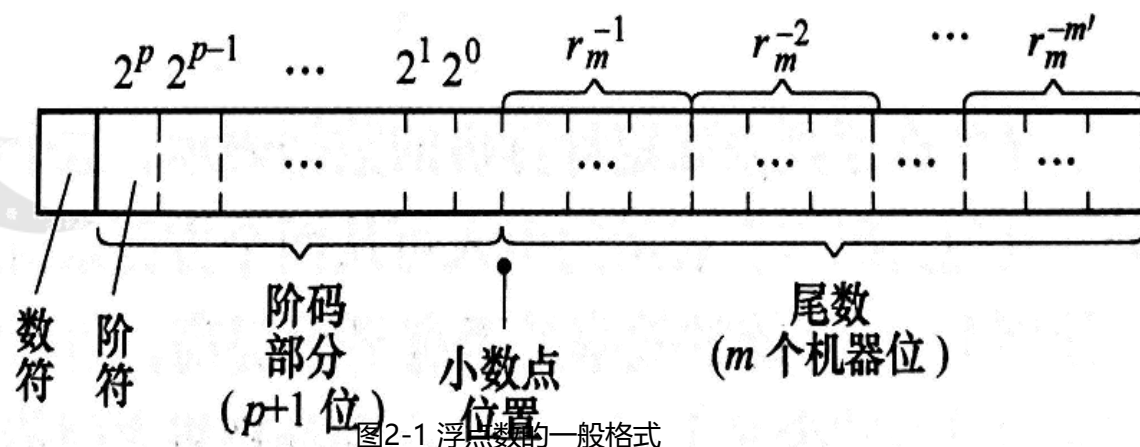
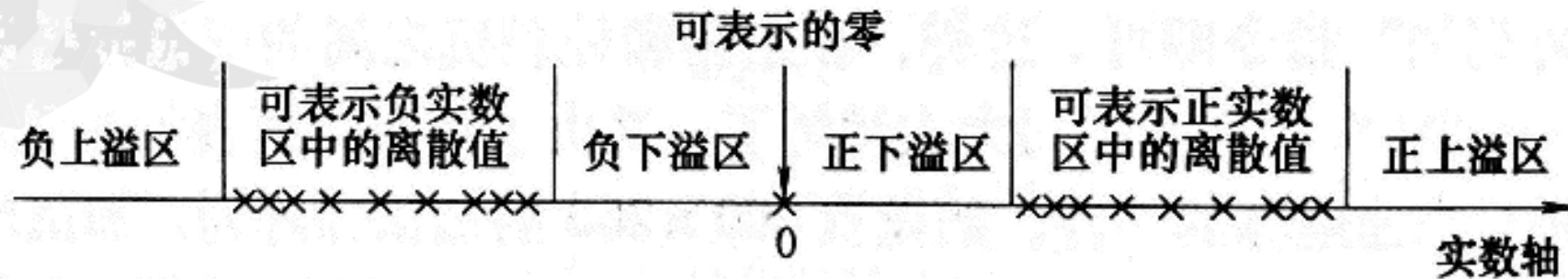


图2-1 浮点数的一般格式

## 2.1.4浮点数尾数基值大小和下溢处理方法的选择

### 1.浮点数尾数基值的选择

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。





## 2.1.4浮点数尾数基值大小和下溢处理方法的选择

### 1.浮点数尾数基值的选择

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。

## 2.1.4浮点数尾数基值大小和下溢处理方法的选择

### 1.浮点数尾数基值的选择

例： $2^3$      $10^3$      $8^3$

基数：2、10、8

阶值：3

$r_m$ 浮点数尾数的基数

这个基数要占多少位： $\log_2 r_m$

$m$ 尾数的计算机位数

$m'$ ： $r_m$ 进制的尾数的数位

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

## 2.1.4浮点数尾数基值大小和下溢处理方法的选择

### 1.浮点数尾数基值的选择

例:  $rm=2$   $m'=m$

$rm=8$   $m'=m/3$

$rm=16$   $m'=m/4$

$rm=10$   $m'=m/4$

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。



那么意气风发地  
走在成功的道路上

## 真题练练手

1、浮点数系统使用的阶基 $r_p=2$ ,阶值位数 $p=2$ ,尾数位数 $m=4$ ,尾数基值 $r_n=2$ 时,在非负阶正尾数、规格化的情况下可表示的最小尾数是 ( ) 1904

A:1/2

B:1/4

C:1/8

D:1/16



那么意气风发地  
走在成功的道路上

## 真题练练手

1、浮点数系统使用的阶基 $r_p=2$ ,阶值位数 $p=2$ ,尾数位数 $m=4$ ,尾数基值 $r_n=2$ 时,在非负阶正尾数、规格化的情况下可表示的最小尾数是 ( ) 1904

A:1/2

B:1/4

C:1/8

D:1/16

答案: A



那么意气风发地  
走在成功的道路上

## 真题练练手

2、浮点数表示中,当阶值位数一定时,不会受到尾数进制影响的是 ( )

A:精度

B:数符

C:范围

D:离散程度



那么意气风发地  
走在成功的道路上

## 真题练练手

2、浮点数表示中,当阶值位数一定时,不会受到尾数进制影响的是 ( ) 1904

A:精度

B:数符

C:范围

D:离散程度

答案: B

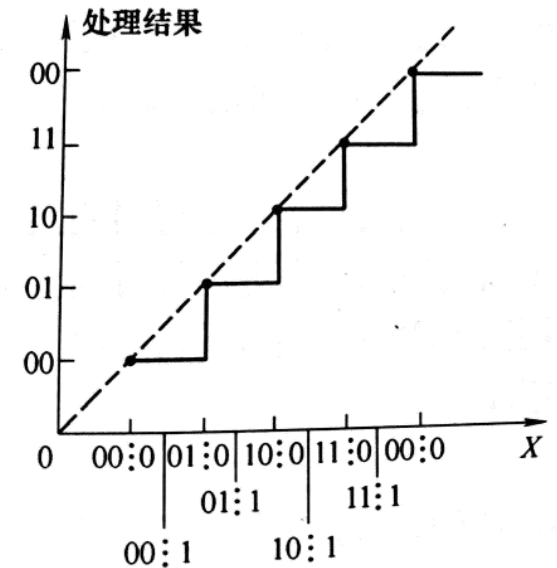


## 2.1.4浮点数尾数基值大小和下溢处理方法的选择

### 2.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



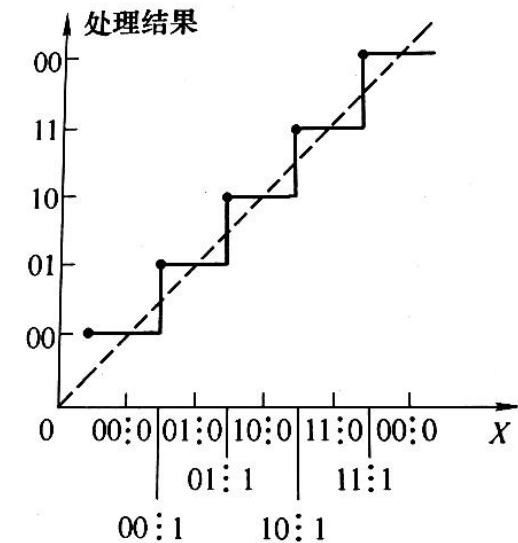
## 2.1.4浮点数尾数基值大小和下溢处理方法的选择

### 2.浮点数尾数的下溢处理方法

#### (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



## 2.1.4浮点数尾数基值大小和下溢处理方法的选择

### 2.浮点数尾数的下溢处理方法

#### (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。

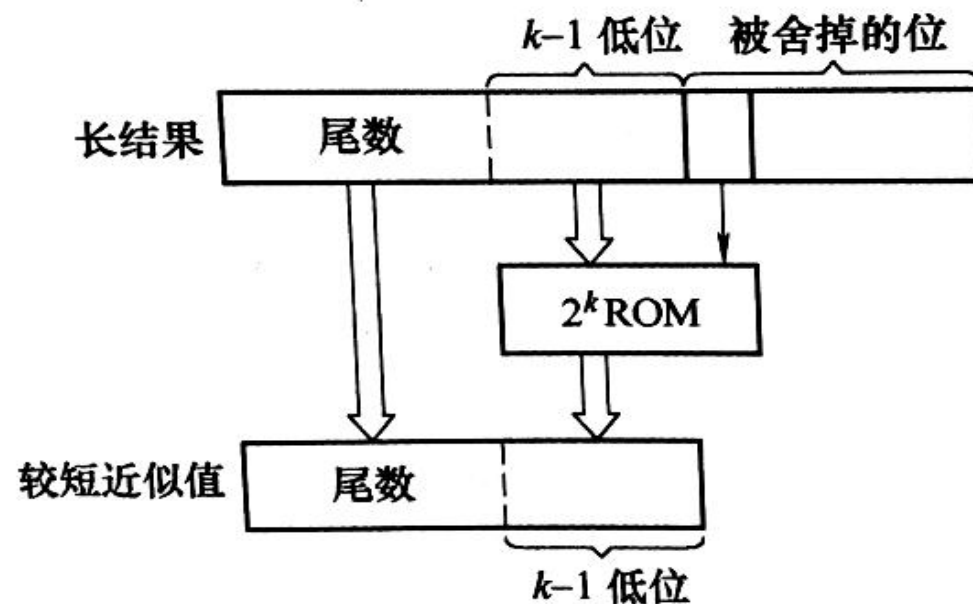
主要缺点是最大误差最大，比截断法的还要大。

## 2.1.4浮点数尾数基值大小和下溢处理方法的选择

### 2.浮点数尾数的下溢处理方法

#### (4) 查表舍入法

查表舍入法是用ROM或PLA存放下溢处理表



## 2.1.4浮点数尾数基值大小和下溢处理方法的选择

### 2.浮点数尾数的下溢处理方法



背

#### (4) 查表舍入法 (文字题)

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。

## 2.1.4浮点数尾数基值大小和下溢处理方法的选择

### 2.浮点数尾数的下溢处理方法

#### (4) 查表舍入法

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。



那么意气风发地  
走在成功的道路上

## 真题练练手

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



那么意气风发地  
走在成功的道路上

## 真题练练手

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





那么意气风发地  
走在成功的道路上

## 真题练练手

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



那么意气风发地  
走在成功的道路上

## 真题练练手

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

## 2.2 寻址方式

**本节主要内容：**

**寻址方式的3种面向，逻辑地址和物理地址的定义**

**寻址方式在指令中的两种指明方式及其优缺点**

**程序的静态再定位和动态再定位**

**信息在内存中存储**

## 2.2寻址方式

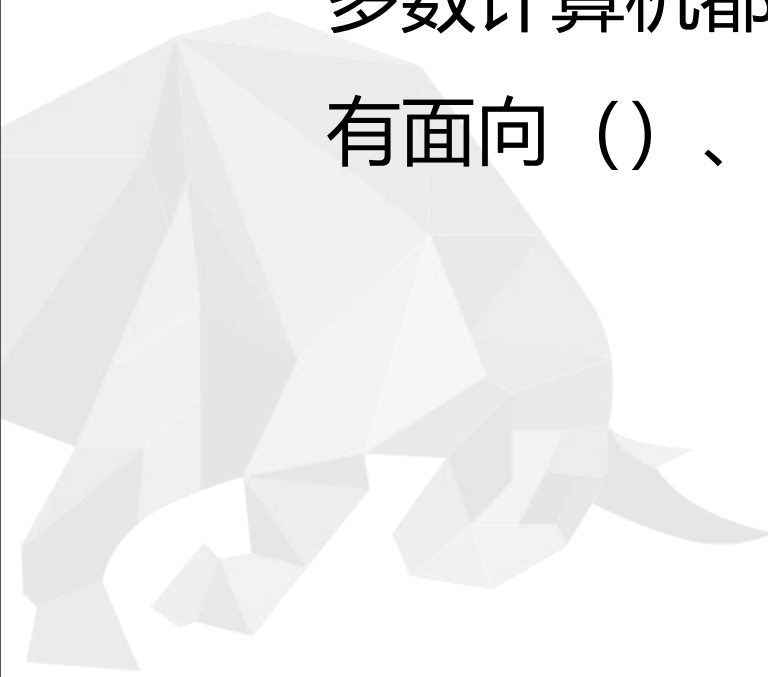
### 什么是寻址方式

寻址方式指的是指令按什么方式寻找（或访问）到所需的操作数或信息的。

寻址方式在多样性、灵活性、寻址范围、地址映像算法和地址变换速度等方面都有了很大的进展。

## 2.2.1 寻址方式的三种面向（单选、填空）

多数计算机都将主存、寄存器、堆栈分类编址，分别有面向（）、面向（）和面向（）的寻址方式。



## 2.2.1 寻址方式的三种面向（单选、填空）

多数计算机都将主存、寄存器、堆栈分类编址，分别有面向主存、面向寄存器和面向堆栈的寻址方式。

面向主存的寻址主要访问主存，少量访问寄存器。

面向寄存器的寻址主要访问寄存器，少量访问主存和堆栈。

面向堆栈的寻址主要访问堆栈，少量访问主存或寄存器。



那么意气风发地  
走在成功的道路上

## 真题练练手

1、下列不属于寻址方式的三种面向的是 ( ) 1804

A:面向主存

B:面向辅存

C:面向寄存器

D:面向堆栈



那么意气风发地  
走在成功的道路上

## 真题练练手

1、下列不属于寻址方式的三种面向的是 ( ) 1804

A:面向主存

B:面向辅存

C:面向寄存器

D:面向堆栈

答案： B



## 2.2.2寻址方式在指令中的指明

寻址方式在指令中一般有两种不同的指明方式，（填空）

一种方式是占用**操作码**中的某些位来指明。

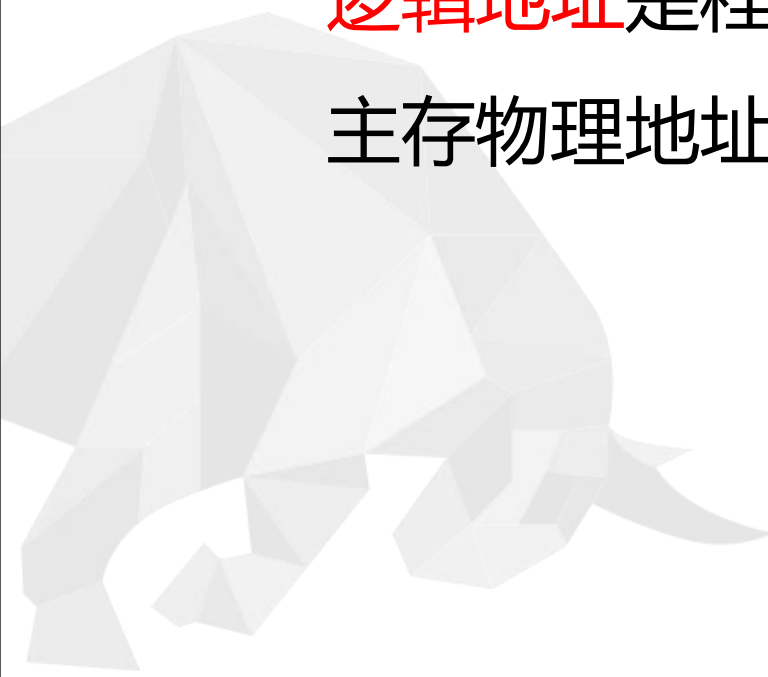
另一种方式是不占用操作码，而是在**地址码**部分专门设置寻址方式位字段指明。

## 2.2.3程序在主存中的定位技术

相关概念：（单选、填空）

**逻辑地址**是程序员编程用的地址。

主存物理地址是程序在主存中的**实际地址**。



## 2.2.3程序在主存中的定位技术

### (1) 静态再定位

在目的程序装入主存时，由装入程序用软件方法把目的程序的逻辑地址变换成物理地址，  
程序执行时，物理地址不再改变，称这种定位技术为静态再定位。

## 2.2.3程序在主存中的定位技术

### (2) 动态再定位

在程序不做变换直接装入主存的同时，将装入主存的起始地址a存入对应该道程序使用的基址寄存器中。程序执行时，只要通过地址加法器将逻辑地址加上基址寄存器的程序基址形成物理（有效）地址后去访存即可。

## 2.2.3程序在主存中的定位技术

### (2) 动态再定位

人们把在执行每条指令时才形成访存物理地址的方法称为动态再定位。

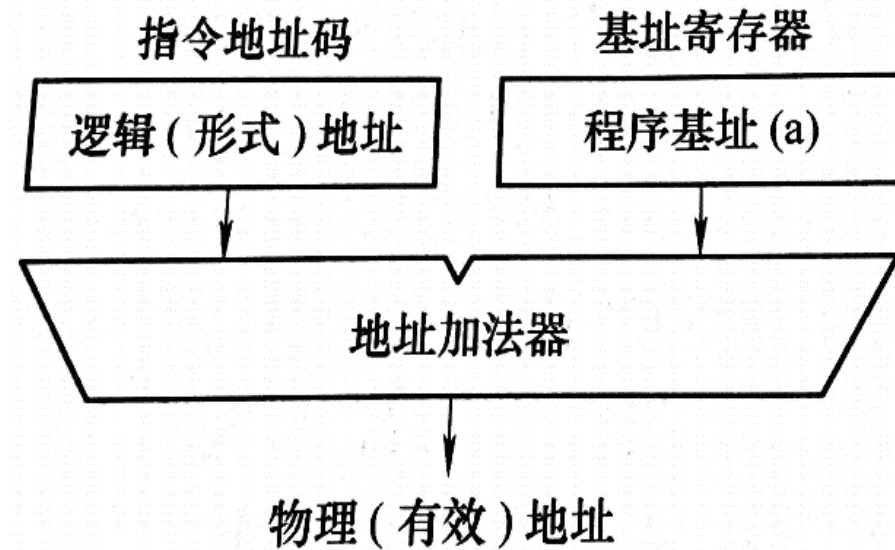


图2-6 基址寻址

## 2.2.3程序在主存中的定位技术

### (3) 虚实地址映像表

地址加界法要求程序员所用编址空间不能超出实际主存的容量。20世纪70年代，采用虚拟存储器增加了映像表硬件后，使程序空间可以超过实际主存空间。

(详情第四章)



那么意气风发地  
走在成功的道路上

## 真题练练手

1、变址寻址的主要作用是（ ） 0807

A:支持程序的动态再定位

B:支持访存地址的越界检查

C:支持向量、数组的运算寻址

D:支持操作系统中的进程调度



那么意气风发地  
走在成功的道路上

## 真题练练手

1、变址寻址的主要作用是（ ） 0807

A:支持程序的动态再定位

B:支持访存地址的越界检查

C:支持向量、数组的运算寻址

D:支持操作系统中的进程调度

答案：C

解析：变址寻址是对诸如向量、数组等数据块运算的支持，以便于实现程序的循环。





那么意气风发地  
走在成功的道路上

## 真题练练手

2、支持动态地址再定位的寻址方式是 ( ) 0607

A:基址寻址

B:间接寻址

C:变址寻址

D:直接寻址



那么意气风发地  
走在成功的道路上

## 真题练练手

2、支持动态地址再定位的寻址方式是 ( ) 0607

A:基址寻址

B:间接寻址

C:变址寻址

D:直接寻址

答案：A

解析：基址寻址是对逻辑地址空间到物理地址空间变换的支持，以利于实现程序的动态再定位。



尚德机构

# ▶ 答疑时间 ◀





尚德机构

▶ THANK YOU ◀

