

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	★ ★

## 2.2 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

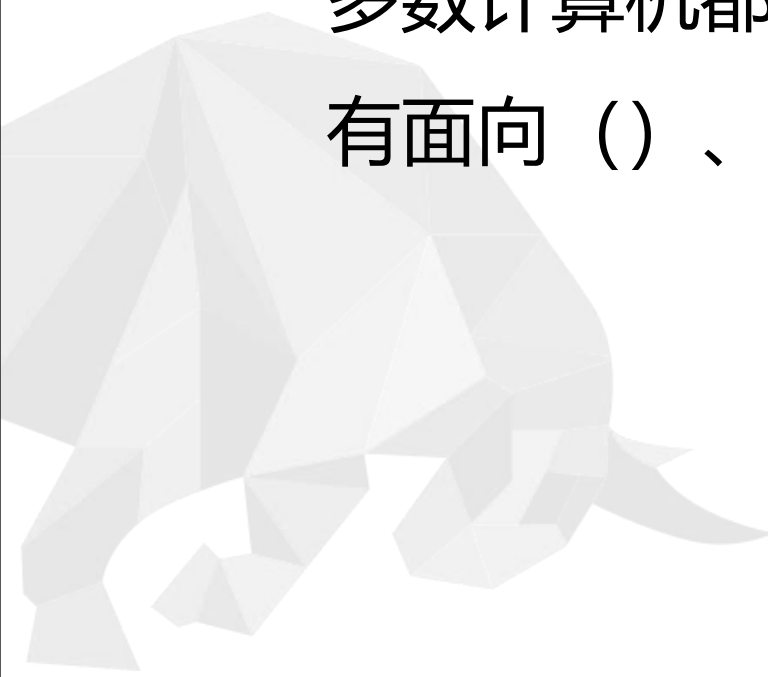
### 什么是寻址方式

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（ ）**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

相关概念：（单选、填空）

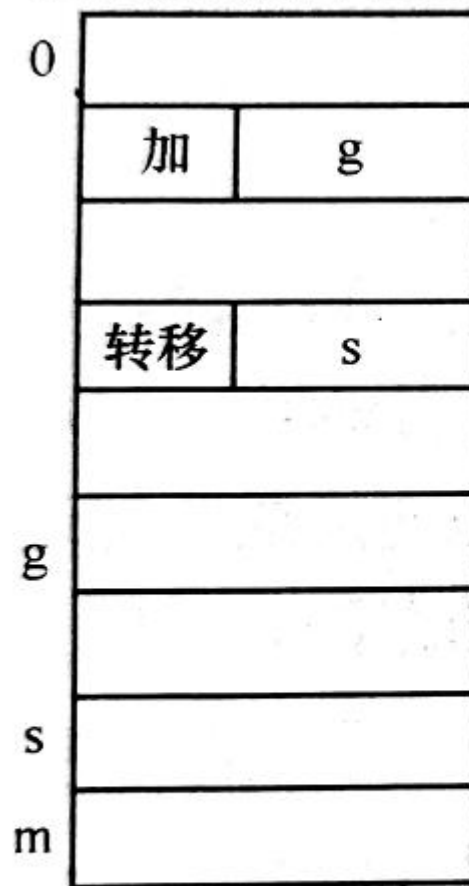
**逻辑地址**是程序员编程用的地址。

主存**物理地址**是程序在主存中的实际地址。

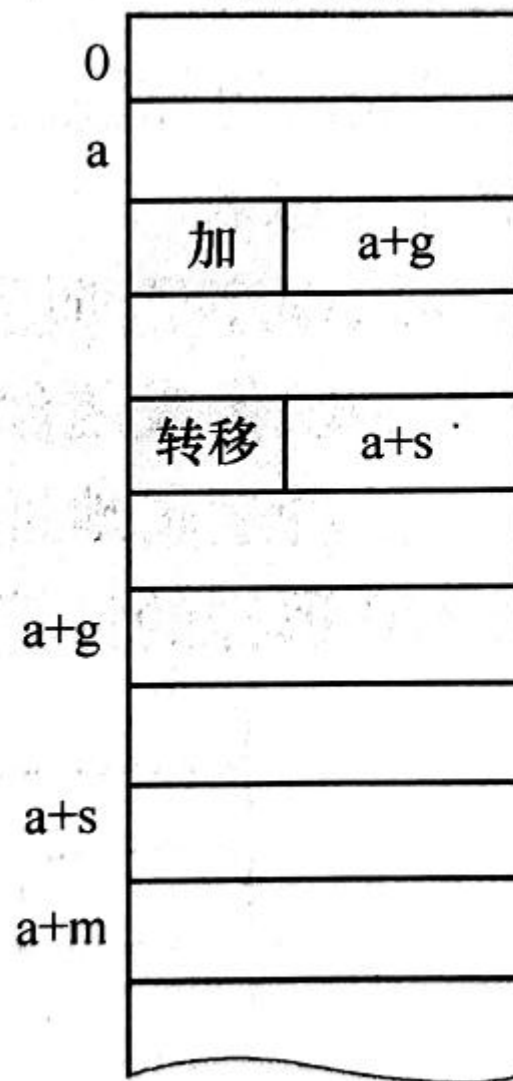
## 2.2.3程序在主存中的定位技术

逻辑地址  
物理地址

A 道程序的逻辑地址空间



主存空间



A 道程序的物理地址空间

## 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。

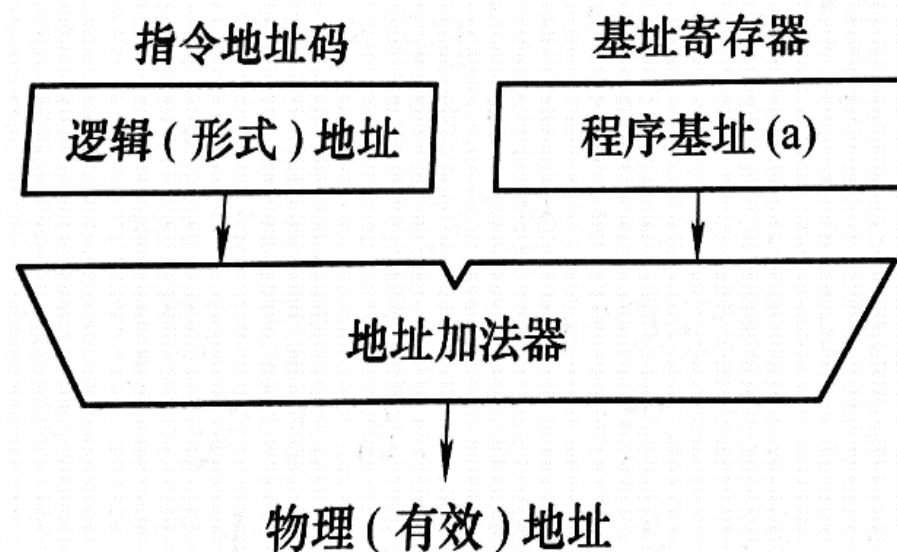


图2-6 基址寻址

## 2.2.3程序在主存中的定位技术

### (2) 动态再定位

基址寻址（逻辑地址空间到物理地址空间变换的支持）

变址寻址（数据块运算的支持）

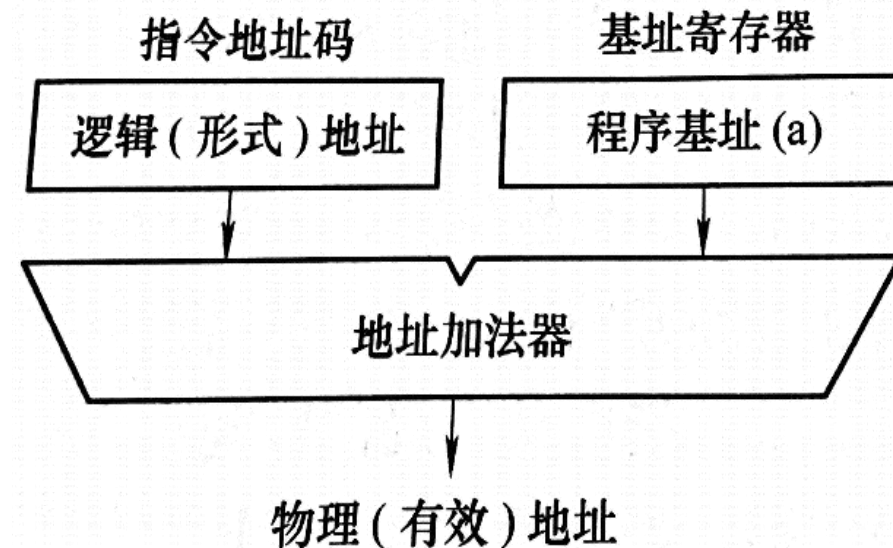


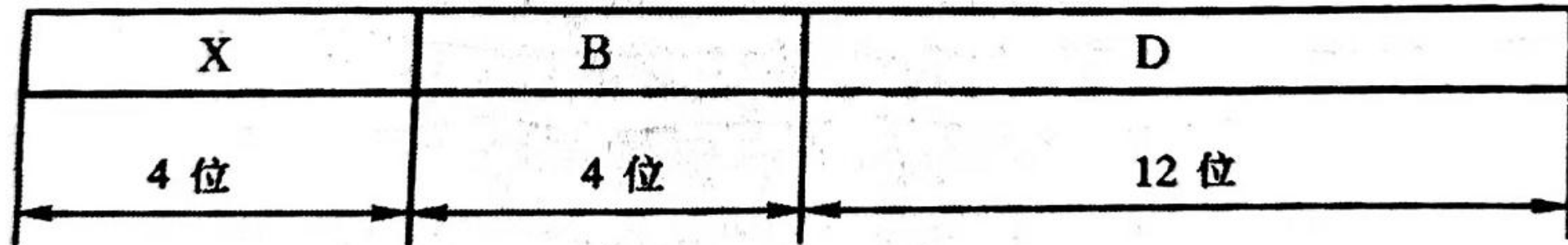
图2-6 基址寻址

## 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。





那么意气风发地  
走在成功的道路上

## 真题练练手

3、主存中的实际地址称为 ( ) 1710

A:物理地址

B:逻辑地址

C:虚拟地址

D:相对地址



那么意气风发地  
走在成功的道路上

## 真题练练手

3、主存中的实际地址称为 ( ) 1710

A:物理地址

B:逻辑地址

C:虚拟地址

D:相对地址

答案：A

## 2.3 指令系统的设计和优化

**本节主要内容：**

**指令格式优化的含义**

**哈夫曼编码、优化的扩展操作码编码，能够求出操作码平均码长**

**综述指令格式优化设计措施**

**优化设计指令格式**

## 2.3 指令系统的设计和优化

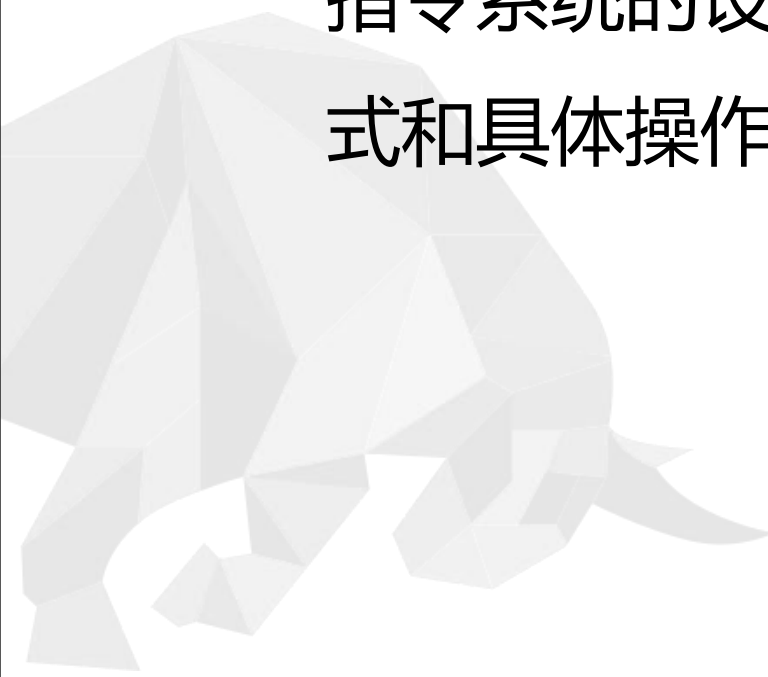
### 2.3.1 指令系统设计的基本原则

指令系统是程序设计者看计算机的主要属性，是软、硬件的主要界面，它在很大程度上决定了计算机具有的基本功能。

## 2.3 指令系统的设计和优化

### 2.3.1 指令系统设计的基本原则（填空）

指令系统的设计包括指令的**功能**（操作类型、寻址方式和具体操作内容）和指令**格式**的设计。



## 2.3 指令系统的设计和优化

### 2.3.1 指令系统设计的基本原则（填空、简单了解）

指令类型一般分**非特权型**和**特权型**两类。

非特权型指令主要供应用程序员使用，也可供系统程序员使用。

非特权型指令包括算术逻辑运算、数据传送、浮点运算、字符串、十进制运算、控制转移及系统控制等子类。

## 2.3 指令系统的设计和优化

### 2.3.1 指令系统设计的基本原则（填空、简单了解）

指令类型一般分**非特权型**和**特权型**两类。

特权型指令只供系统程序员使用，用户无权使用。用户只有先经访管指令（非特权型）调用操作系统，再由操作系统来使用这些特权指令。

其中，有“启动I/O”（多用户环境下）、停机等待、存储管理保护、控制系统状态、诊断等子类。

## 2.3 指令系统的设计和优化

### 2.3.1 指令系统设计的基本原则（简答题）

编译程序设计者要求指令系统应设计具有：

- 1) 规整性。
- 2) 对称性。
- 3) 独立性和全能性。
- 4) 正交性。
- 5) 可组合性。
- 6) 可扩充性。



## 2.3 指令系统的设计和优化

### 2.3.1 指令系统设计的基本原则（简单了解）

系统结构设计者则还希望：

- 1) 指令码密度适中。
- 2) 兼容性
- 3) 适应性。



那么意气风发地  
走在成功的道路上

## 真题练练手

1、指令系统的设计包括指令（ ）和指令（ ）的设计。1904



那么意气风发地  
走在成功的道路上

## 真题练练手

1、指令系统的设计包括指令（ ）和指令（ ）的设计。1904

答案：功能 格式



那么意气风发地  
走在成功的道路上

## 真题练练手

2、非特权型指令主要供（ ）程序员使用,特权型指令主要供（ ）程序员使用。1810



那么意气风发地  
走在成功的道路上

## 真题练练手

2、非特权型指令主要供（ ）程序员使用,特权型指令主要供（ ）程序员使用。1810

答案：应用 系统



那么意气风发地  
走在成功的道路上

## 真题练练手

3、简述编译程序设计者要求指令系统应具有的特性。1810



那么意气风发地  
走在成功的道路上

## 真题练练手

3、简述编译程序设计者要求指令系统应具有的特性。1810

答案：


编译程序设计者要求指令系统应设计具有：

- 1)规整性。    2)对称性。    3)独立性和全能性。
- 4)正交性。    5)可组合性。    6)可扩充性。

## 2.3 指令系统的设计和优化

### 2.3.2 指令操作码的优化（填空）

指令是由（ ）和（ ）两部分组成的。




操作码	地址码
-----	-----



## 2.3 指令系统的设计和优化

### 2.3.2 指令操作码的优化（填空）

指令是由操作码和地址码两部分组成的。



操作码	地址码
-----	-----

# 2.3 指令系统的设计和优化

## 2.3.2 指令操作码的优化

利用哈夫曼算法构造哈夫曼树。(综合)

表 2-4 某模型机指令使用频度举

指 令	使用频度( $p_i$ )	指 令
$I_1$	0.40	$I_5$
$I_2$	0.30	$I_6$
$I_3$	0.15	$I_7$
$I_4$	0.05	—

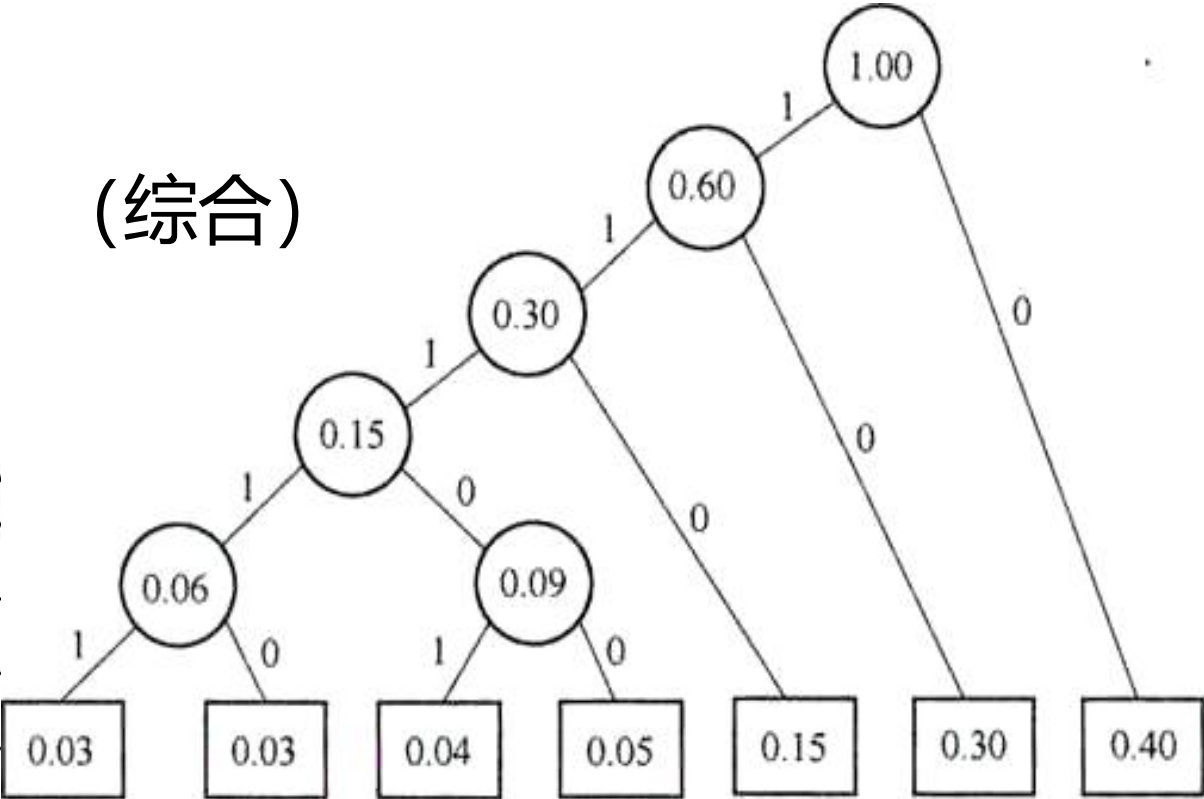


图 2-9 哈夫曼树举例

## 2.3 指令系统的设计和优化

### 2.3.2 指令操作码的优化

【例】现假设某模型机共有 $n$  ( $n=7$ ) 条指令，使用频度见表2-8所示。若操作码用定长码表示需要 $\lceil \log_2 n \rceil$  (即3) 位。而按信息论观点，当各种指令的出现是相互独立时 (实际并不都是如此)，操作码的信息源熵 (信息源所含的平均信息量)  $H = -\sum_{i=1}^n p_i \log_2 p_i$  由表2-8的数据可得)  $H=2.17$ ，说明表示这7种指令的操作码平均只需2.17位即可。现在用3位定长码表示，信息冗余为

$$\frac{\text{实际平均码长} - H}{\text{实际平均码长}} = \frac{3 - 2.17}{3} \approx 28\%$$

相当大。为减少信息冗余，改用哈夫曼编码。

## 2.3 指令系统的设计和优化

### 2.3.3 指令字格式的优化（单选）

指令字格式优化的措施概括起来包括以下几点：

- 1) 采用**扩展操作码**，并根据指令的频度凡的分布状况选择合适的编码方式，以缩短操作码的平均码长。
- 2) 采用诸如基址、变址、相对、寄存器、寄存器间接、段式存放、隐式指明等多种寻址方式，以缩短地址码的长度，并在有限的地址长度内提供更多的地址信息。
- 3) 采用0、1、2、3等多种地址制，以增强指令的功能，这样从宏观上就越能缩短程序的长度，并加快程序的执行速度。
- 4) 在同种地址制内再采用多种地址形式，如寄存器-寄存器、寄存器-主存、主存-主存等，让每种地址字段可以有多种长度，且让长操作码与短地址码进行组配。
- 5) 在维持指令字在存储器中按整数边界存储的前提下，使用多种不同的指令字长度。



那么意气风发地  
走在成功的道路上

## 真题练练手

1、指令的操作码优化编码方法有（ ）和（ ）。1004



那么意气风发地  
走在成功的道路上

## 真题练练手

1、指令的操作码优化编码方法有（ ）和（ ）。1004

答案：哈夫曼编码      扩展操作码编码



那么意气风发地  
走在成功的道路上

## 真题练练手

2、指令是由（ ）和（ ）两部分组成。1510



那么意气风发地  
走在成功的道路上

## 真题练练手

2、指令是由（ ）和（ ）两部分组成。1510

答案：操作码 地址码





那么意气风发地  
走在成功的道路上

## 真题练练手

3、计算机中优化使用的操作码编码方法是（        ） 0804

A:哈夫曼编码

B:ASCII码

C:BCD码

D:扩展操作码



那么意气风发地  
走在成功的道路上

## 真题练练手

3、计算机中优化使用的操作码编码方法是（        ） 0804

A:哈夫曼编码

B:ASCII码

C:BCD码

D:扩展操作码

答案：D

## 2.4 指令系统的发展和改进

**本节主要内容：**

**按CISC方向发展、改进指令系统**

**按RISC方向发展改进指令系统**

**CISC的问题和RISC的优点**

## 2.4指令系统的发展和改进

### 2.4.1 两种途径和方向（CISC和RISC）

为使计算机系统有**更强的功能**、**更高的性能**和**更好的性能价格比**，满足应用的需要，在机器指令系统的设计、发展和改进上有两种不同的途径和方向。

## 2.4指令系统的发展和改进

复杂指令系统计算机 (Complex Instruction Set Computer, CISC)

一种是如何进一步增强原有指令的功能以及设置更为复杂的新指令以取代原先由软件子程序完成的功能，实现软件功能的硬化。按此方向发展，机器指令系统日益庞大和复杂。

## 2.4指令系统的发展和改进

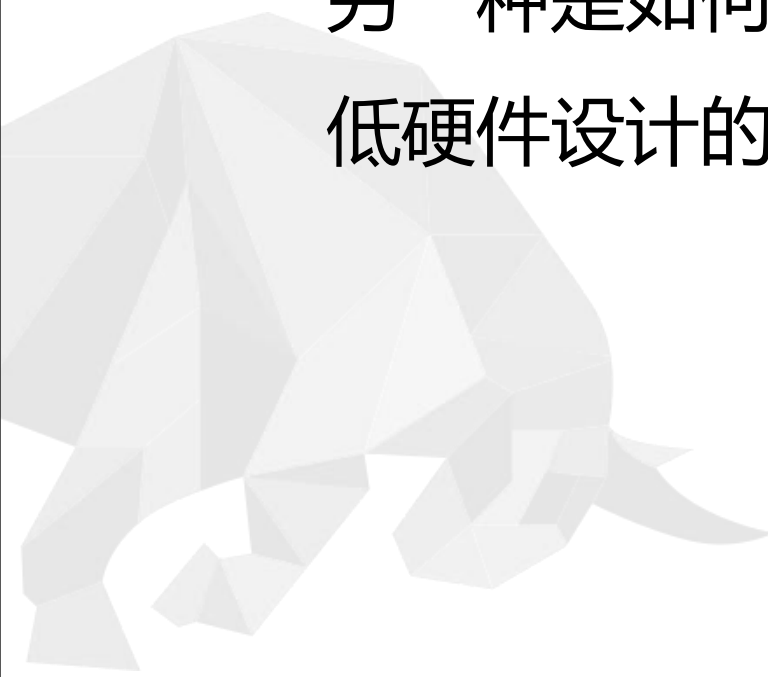
复杂指令系统计算机 (Complex Instruction Set Computer, CISC)

这可从面向**目标程序**、面向**高级语言**、面向**操作系统**  
三个方面的优化实现来考虑 (单选、填空)

## 2.4指令系统的发展和改进

精简指令系统计算机 (Reduced Instruction Set Computer, RISC)

另一种是如何通过减少指令种数和简化指令功能来降低硬件设计的复杂度，提高指令的执行速度。



## 2.4指令系统的发展和改进

### 2.4.2按CISC方向发展和改进指令系统

#### 1.面向目标程序的优化实现改进

途径1 通过对大量已有机器的机器语言程序及其执行情况，统计各种指令和指令串 的使用频度来加以分析和改进。



## 2.4指令系统的发展和改进

### 2.4.2按CISC方向发展和改进指令系统

#### 1.面向目标程序的优化实现改进

途径2 增设强功能复合指令来取代原先由常用宏指令或子程序（如双倍长运算、三角函数、开方、指数、二-十进制数转换、编辑、翻译等子程序）实现的功能，由微程序解释实现，不仅大大提高了运算速度，减少了程序调用的额外开销，也减少了子程序所占的主存空间。

## 2.4指令系统的发展和改进

### 2.4.2按CISC方向发展和改进指令系统

#### 1.目标程序（2个）

途径1 通过对大量已有机器的**机器语言程序**及其执行情况，**统计**各种指令和指令串的使用**频度**来加以分析和改进。

途径2 增设强功能**复合指令**来取代原先由常用**宏指令或子程序**实现的功能，由**微程序**解释实现，不仅大大提高了运算速度，减少了程序调用的额外开销，也减少了子程序所占的主存空间。

## 2.4指令系统的发展和改进

### 2.4.2按CISC方向发展和改进指令系统

#### 2.面向高级语言的优化实现改进

途径1 通过对源程序中各种高级语言语句的使用频度  
进行统计来分析改进。

## 2.4指令系统的发展和改进

### 2.4.2按CISC方向发展和改进指令系统

#### 2.面向高级语言的优化实现改进

途径2 应当增强系统结构的规整性，尽量减少例外或特殊的情况和用法，让所有运算都对称、均匀地在存储（寄存器）单元间进行。对所有存储（寄存器）单元同等对待，无论是操作数或运算结果都可无约束地存放在任意单元中。这样，为优化管理通用寄存器的使用可以大大减少很多的辅助开销。

## 2.4指令系统的发展和改进

### 2.4.2按CISC方向发展和改进指令系统

#### 2.面向高级语言的优化实现改进

途径3 改进指令系统，使它与各种语言间的语义差距都有同等的缩小。

## 2.4指令系统的发展和改进

### 2.4.2按CISC方向发展和改进指令系统

#### 2.面向高级语言的优化实现改进

##### 途径4

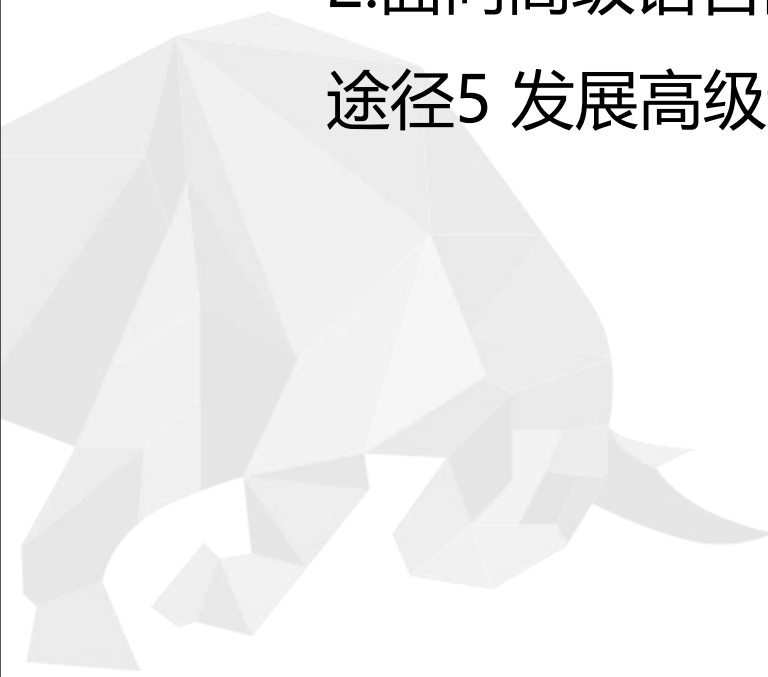
由“以指令系统为主，高级语言为从”方式演变成“以高级语言为主，指令系统为从”的方式。

## 2.4指令系统的发展和改进

### 2.4.2按CISC方向发展和改进指令系统

#### 2.面向高级语言的优化实现改进

途径5 发展高级语言计算机（或称高级 语言机器）。



## 2.4指令系统的发展和改进

### 2.4.2按CISC方向发展和改进指令系统

#### 2.高级语言（5个）

途径1 通过对源程序中各种**高级语言**语句的使用**频度**进行统计来分析改进。

途径2 应当增强系统结构的**规整性**，尽量减少例外或特殊的情况和用法。

途径3 改进**指令系统**，使它与各种语言间的语义差距都有同等的缩小。

途径4 由“以指令系统为主，高级语言为从”方式演变成“**以高级语言为主**，指令系统为从”的方式。

途径5 发展**高级语言计算机**（或称高级语言机器）。



## 2.4指令系统的发展和改进

### 2.4.2按CISC方向发展和改进指令系统

#### 3.面向操作系统的优化实现改进

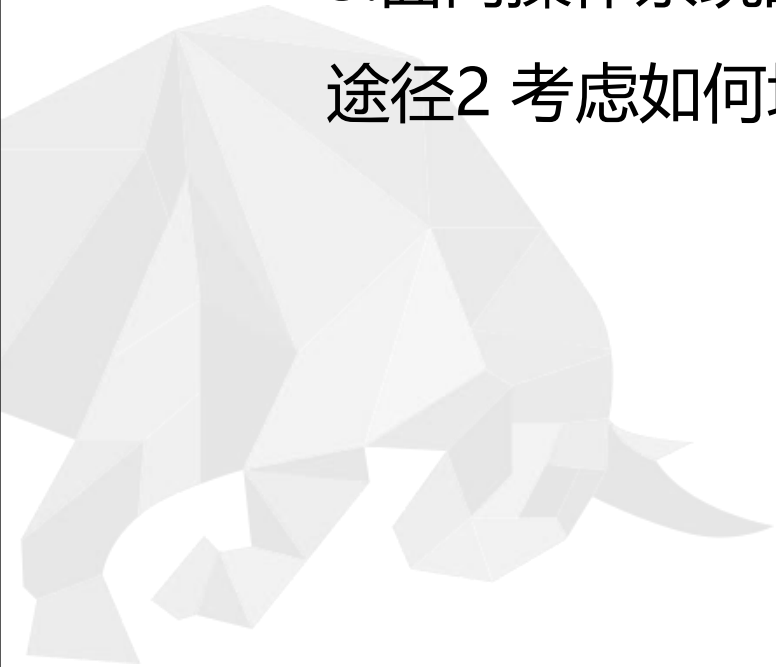
途径1 通过对操作系统中常用指令和指令串的使用频度进行统计分析来改进。但这种改进的效果很有限。

## 2.4指令系统的发展和改进

### 2.4.2按CISC方向发展和改进指令系统

#### 3.面向操作系统的优化实现改进

途径2 考虑如何增设专用于操作系统的新指令。



## 2.4指令系统的发展和改进

### 2.4.2按CISC方向发展和改进指令系统

#### 3.面向操作系统的优化实现改进

途径3 把操作系统中频繁使用的，对速度影响大的机构型软件子程序硬化或固化，改为直接用硬件或微程序解释实现。

## 2.4指令系统的发展和改进

### 2.4.2按CISC方向发展和改进指令系统

#### 3.面向操作系统的优化实现改进

途径4 发展让操作系统由专门的处理机来执行的功能分布处理系统结构。

## 2.4指令系统的发展和改进

### 2.4.2按CISC方向发展和改进指令系统（简答）

#### 3.操作系统

途径1通过对**操作系统**中常用指令和指令串的使用**频度**进行统计分析来改进。

途径2考虑如何**增设专用于操作系统的新指令**。

途径3把操作系统中频繁使用的，对速度影响大的机构型软件子程序硬化或固化，改为**直接用硬件或微程序**解释实现。

途径4 发展让操作系统由专门的处理机来执行的功能**分布处理**系统结构。

## 2.4指令系统的发展和改进

### 2.4.2按CISC方向发展和改进指令系统

#### 1.目标程序（2个）

途径1 通过对大量已有机器的**机器语言程序**及其执行情况，**统计**各种指令和指令串的使用（ ）来加以分析和改进。

途径2 增设强功能**复合指令**来取代原先由常用**宏指令或子程序**实现的功能，由（ ）解释实现，不仅大大提高了运算速度，减少了程序调用的额外开销，也减少了子程序所占的主存空间。

## 2.4指令系统的发展和改进

### 2.4.2按CISC方向发展和改进指令系统

#### 1.目标程序（2个）

途径1 通过对大量已有机器的**机器语言程序**及其执行情况，**统计**各种指令和指令串的使用**频度**来加以分析和改进。

途径2 增设强功能**复合指令**来取代原先由常用**宏指令或子程序**实现的功能，由**微程序**解释实现，不仅大大提高了运算速度，减少了程序调用的额外开销，也减少了子程序所占的主存空间。

## 2.4指令系统的发展和改进

### 2.4.2按CISC方向发展和改进指令系统

#### 2.高级语言（5个）

途径1 通过对源程序中各种**高级语言**语句的使用**频度**进行统计来分析改进。

途径2 应当增强系统结构的（ ），尽量减少例外或特殊的情况和用法。

途径3 改进（ ），使它与各种语言间的语义差距都有同等的缩小。

途径4 由“以指令系统为主，高级语言为从”方式演变成“**以高级语言为主**，指令系统为从”的方式。

途径5 发展（ ）（或称高级语言机器）。



## 2.4指令系统的发展和改进

### 2.4.2按CISC方向发展和改进指令系统

#### 2.高级语言（5个）

途径1 通过对源程序中各种**高级语言**语句的使用**频度**进行统计来分析改进。

途径2 应当增强系统结构的**规整性**，尽量减少例外或特殊的情况和用法。

途径3 改进**指令系统**，使它与各种语言间的语义差距都有同等的缩小。

途径4 由“以指令系统为主，高级语言为从”方式演变成“**以高级语言为主**，指令系统为从”的方式。

途径5 发展**高级语言计算机**（或称高级语言机器）。

## 2.4指令系统的发展和改进

### 2.4.2按CISC方向发展和改进指令系统

#### 3.操作系统

途径1通过对（ ）中常用指令和指令串的使用**频度**进行统计分析来改进。

途径2考虑如何**增设专用于操作系统的新指令**。

途径3把操作系统中频繁使用的，对速度影响大的机构型软件子程序硬化或固化，改为**直接用（ ）或微程序**解释实现。

途径4 发展让操作系统由专门的处理机来执行的功能（ ）系统结构。

## 2.4指令系统的发展和改进

### 2.4.2按CISC方向发展和改进指令系统

#### 3.操作系统

途径1通过对**操作系统**中常用指令和指令串的使用**频度**进行统计分析来改进。

途径2考虑如何**增设专用于操作系统的新指令**。

途径3把操作系统中频繁使用的，对速度影响大的机构型软件子程序硬化或固化，改为**直接用硬件或微程序**解释实现。

途径4 发展让操作系统由专门的处理机来执行的功能**分布处理**系统结构。

## 2.4指令系统的发展和改进

### 2.4.3按RISC方向发展和改进指令系统

#### 1.CISC的问题

- 1) 指令系统**庞大**，一般指令在200条以上。
- 2) 许多指令的操作**繁杂**，执行速度很低，甚至不如用几条简单、基本的指令组合实现。
- 3) 由于指令系统庞大，使高级语言编译程序选择**目标指令的范围太大**，因此，难以优化生成高效机器语言程序，编译程序也**太长、太复杂**。
- 4) 由于指令系统庞大，各种指令的使用频度都不会太高，且**差别很大**，其中相当一部分指令的**利用率很低**。

## 2.4指令系统的发展和改进

### 2.4.3按RISC方向发展和改进指令系统

#### 2.设计RISC的基本原则

- 1) 确定指令系统时，只选择使用频度很高的那些指令，再增加少量能有效支持操作系统、高级语言实现及其他功能的指令，大大减少指令条数，一般使之不超过100条。
- 2) 减少指令系统所用寻址方式种类，一般不超过两种。简化指令的格式限制在两种之内，并让全部指令都是相同长度。
- 3) 让所有指令都在一个机器周期内完成。
- 4) 扩大通用寄存器数，一般不少于32个，尽量减少访存，所有指令只有存（STORE）、取（LOAD）指令访存，其他指令一律只对寄存器操作。
- 5) 为提高指令执行速度，大多数指令都用硬联控制实现，少数指令才用微程序实现。
- 6) 通过精简指令和优化设计编译程序，简单、有效地支持高级语言的实现。

## 2.4指令系统的发展和改进

### 2.4.3按RISC方向发展和改进指令系统

#### 3.设计RISC结构采用的基本技术

- 1) 按设计RISC的一般原则来设计。
- 2) 逻辑实现采用硬联和微程序相结合。
- 3) 在CPU中设置大量工作寄存器并采用重叠寄存器窗口。
- 4) 指令用流水和延迟转移。
- 5) 采用高速缓冲存储器Cache,设置指令Cache和数据Cache分别存放指令和数据。
- 6) 优化设计编译系统。

## 2.4指令系统的发展和改进

### 2.4.3按RISC方向发展和改进指令系统

【例】设A、A + 1、B和B + 1为主存单元，程序

取 A, Ra; (A) → Ra

存 Ra, B; Ra → (B)

取 A + 1, Ra; (A + 1) → Ra

存 Ra, B + 1; Ra → (B + 1)

## 2.4指令系统的发展和改进

### 2.4.3按RISC方向发展和改进指令系统

#### 4. RISC技术的发展

好处：

- 1) 简化指令系统设计，适合VLSI实现。
- 2) 提高计算机的执行速度和效率。
- 3) 降低设计成本，提高系统的可靠性。
- 4) 可直接支持高级语言的实现，简化编译程序的设计。



## 2.4指令系统的发展和改进

### 2.4.3按RISC方向发展和改进指令系统

#### 4. RISC技术的发展

问题和不足，主要有：

- 1) 由于指令少，使原来在CISC上由单一指令完成的某些复杂功能现在要用多条RISC指令才能完成，加重了汇编语言程序设计的负担，增加了机器语言程序的长度，占用存储空间多，加大了指令的信息流量。
- 2) 对浮点运算的执行和虚拟存储器的支持虽有很大加强，但仍显得不足。
- 3) RISC计算机的编译程序比CISC的难写。



那么意气风发地  
走在成功的道路上

## 真题练练手

1、（ ）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

- A、数据表示
- B、数字表示
- C、字符表示
- D、编程表示

答案：A

讲解：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



尚德机构

# ▶ 答疑时间 ◀





尚德机构

▶ THANK YOU ◀

