

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



# 《计算机系统结构》

课程代码：02325

版本：2012年版

主编：李学干

出版社：机械工业出版社



# 考试题型介绍

100分

单选题：10题 \* 1分=10分

填空题：10题 \* 2分=20分（一题两空）

简答题：6题 40分

综合题：3题 \* 10分=30分

# 学习资料



**真题：**共12套（2013年——2019年）12年以前也有几套

**我的ppt：**对书中各个知识点的详细讲解（标注知识点考察方式）

**总结资料：**考纲全部知识点的考点都包括



# 标注

**识记：** 单选+填空

**领会：** 单选+填空+简答

**应用：** 单选+填空+简答+计算

**简单了解：** 考试不考，为以后铺垫，课下可以深入研究

**没有标记：** 没考过，一般是基础概念，辅助学习的



# 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1章：计算机系统结构概论



# 第1章 计算机系统概论



计算机系统的层次结构



计算机系统结构、计算机组成和计算机实现



计算机系统的软硬件取舍及定量设计原理



软件、应用、器件的发展对系统结构的影响



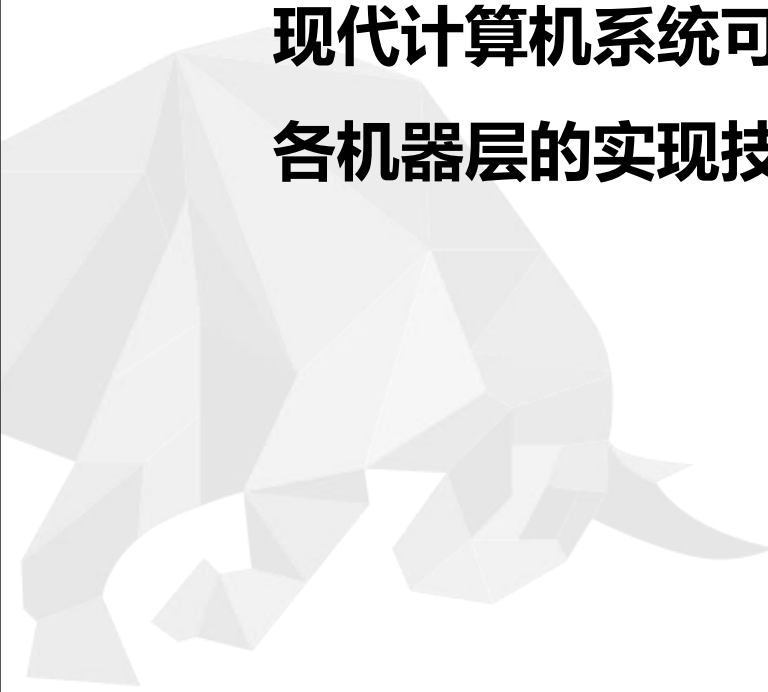
系统结构中的并行性开发

# 1.1 计算机系统的层次结构

**本节主要内容：（识记=选择+填空）**

**现代计算机系统可分为哪几层**

**各机器层的实现技术**



# 计算机系统的组成

**计算机系统=硬件+软件**

**硬件：（5）运算器、控制器、存储器、输入设备、输出设备**

**软件：（2）系统软件、应用软件**

# 1.1 计算机系统的层次结构

## 1、计算机系统的多级层次结构

从使用语言的角度，一台由软、硬件组成的通用计算机系统可以被看成是**按功能划分**的多层机器级组成的层次结构。



# 1.1 计算机系统的层次结构

## 1、计算机系统的多级层次结构

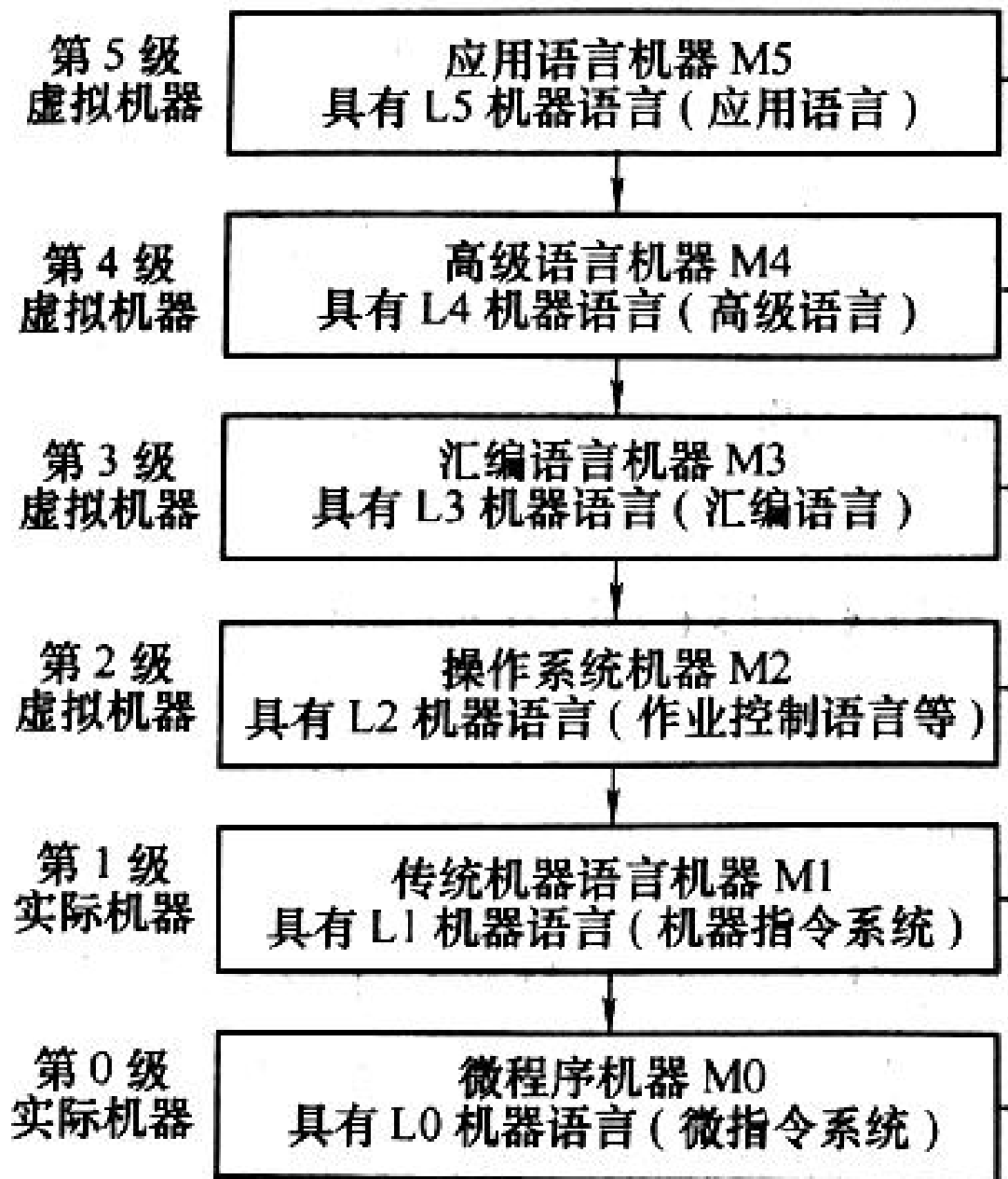
从使用语言的角度，一台由软、硬件组成的通用计算机系统可以被看成是**按功能划分**的多层机器级组成的层次结构。

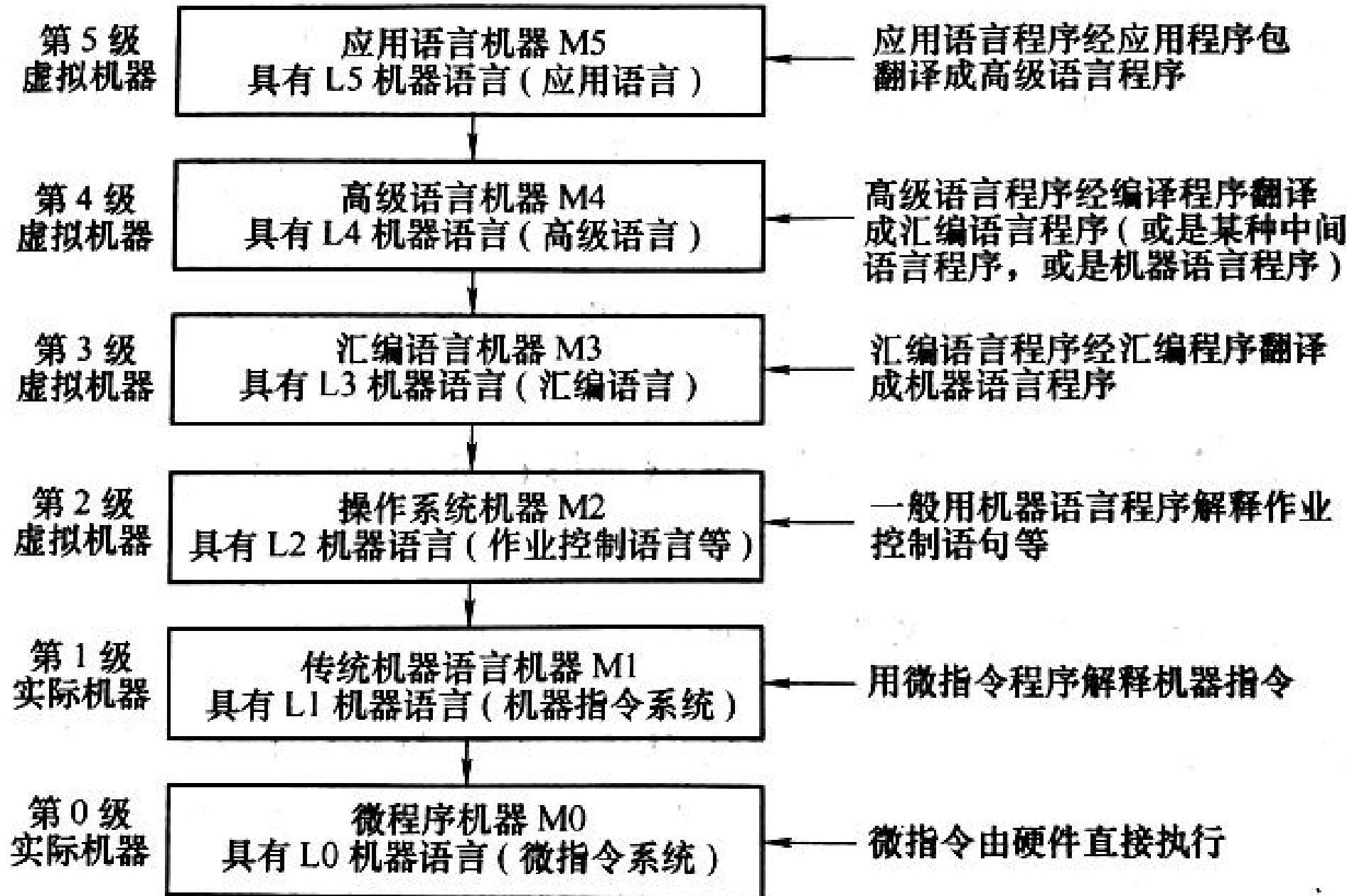
机器语言      汇编语言      高级语言

# 1.1 计算机系统的层次结构

## 1、计算机系统的多级层次结构（选择）

层次结构由高到低





## 1.1 计算机系统的层次结构

### 1、计算机系统的多级层次结构（填空、单选）

“机器”被定义为 能**存储**和**执行**相应语言程序的**算法**和**数据结构**的集合体。

实际上，只有**二进制机器指令**，即传统所讲的机器语言与机器硬件直接对应，方可直接被硬件识别和执行。



## 1.2 计算机系统结构、计算机组成和计算机实现

**本节主要内容：（领会=单选+填空+简答）**

**组成、和实现的定义以及研究方面**

**三者之间的关系和影响**

**透明性概念**

## 1.2 计算机系统结构、计算机组成和计算机实现

### 1.2.1 计算机系统结构的定义和内涵（单选）

从计算机的层次结构角度来看，系统结构（System Architecture）是对计算机系统中各级界面的定义及其上下的（ ）分配。每级都有其自己的系统结构。

## 1.2 计算机系统结构、计算机组成和计算机实现

### 1.2.1 计算机系统结构的定义和内涵（单选）

从计算机的层次结构角度来看，系统结构（System Architecture）是对计算机系统中各级界面的定义及其上下的**功能分配**。每级都有其自己的系统结构。

## 1.2 计算机系统结构、计算机组成和计算机实现

### 1.2.1 计算机系统结构的定义和内涵（单选）

就目前的通用机来说，计算机系统结构的属性包括：

- 1) 硬件能**直接**识别和处理的数据类型及格式等的**数据表示**。
- 2) **最小**可寻址单位、寻址种类、地址计算等的**寻址方式**。
- 3) 通用/专用寄存器的设置、数量、字长、使用约定等的寄存器**组织**。
- 4) 二进制或汇编指令的操作类型、格式、排序方式、控制机构等的**指令系统**。

## 1.2 计算机系统结构、计算机组成和计算机实现

### 1.2.1 计算机系统结构的定义和内涵（单选）

- 5) 主存的**最小**编址单位、编址方式、容量、最大可编址空间等的存储系统组织。
- 6) **中断**的分类与分级、中断处理程序功能及入口地址等的中断机构。
- 7) 系统机器级的**管态**和**用户态**的定义与切换。
- 8) **输入/输出设备**的连接、使用方式、流量、操作结束、出错指示等的机器级I/O结构。
- 9) 系统各部分的**信息保护**方式和保护机构等属性。

## 1.2.2计算机组成与计算机实现的定义和内涵

### 1.计算机组成（填空）

计算机组成（Computer Organization）指的是计算机系统结构的**逻辑实现**，包括机器级内部的**数据流**和**控制流**的组成以及逻辑设计等。

计算机组成着眼于机器级内部各事件的**排序方式**与**控制机构**、各部件的功能及各部件间的联系。

## 1.2.2计算机组成与计算机实现的定义和内涵

### 1.计算机组成（单选、简答）

计算机组成设计要确定的方面一般应包括：

- 1) 数据通路宽度
- 2) 专用部件的设置
- 3) 各种操作对部件的共享程度。
- 4) 功能部件的并行度
- 5) 控制机构的组成方式
- 6) 缓冲和排队技术。
- 7) 预估、预判技术
- 8) 可靠性技术

要求，不要迟到早退，在线听直播2小时+随堂考最后一节课可获得汇总资料

## 1.2.2计算机组成与计算机实现的定义和内涵

### 2.计算机实现

计算机实现（Computer Implementation）指的是计算机组成的**物理实现**，包括处理机、主存等部件的物理结构，器件的集成度和速度，器件、模块、插件、底板的划分与连接，专用器件的设计，微组装技术，信号传输，电源、冷却及整机装配技术等。



## 1.2.2计算机组成与计算机实现的定义和内涵

### 2.计算机实现（单选、填空）

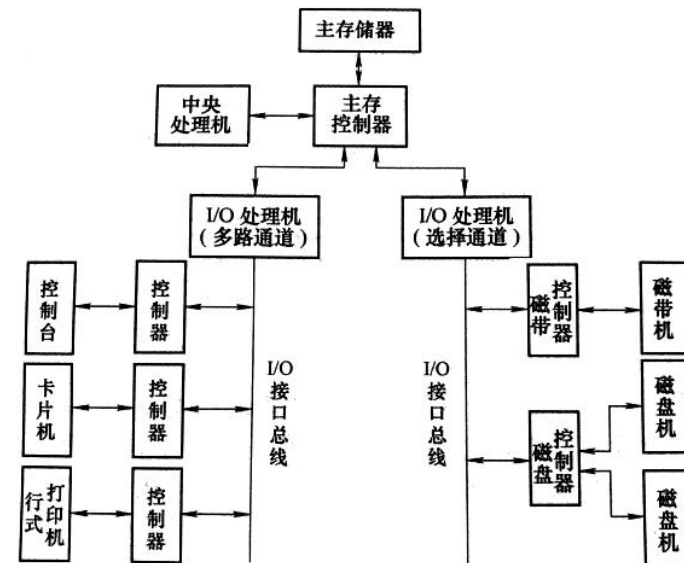
计算机实现的设计着眼于器件技术和微组装技术，  
其中，器件技术起着主导作用。

确定指令系统中是否要设置乘法指令属于计算机系统结构  
主存容量与编址方式（按位、按字节，还是按字访问等）的确定属于  
计算机系统结构。

## 1.2.2计算机组成与计算机实现的定义和内涵

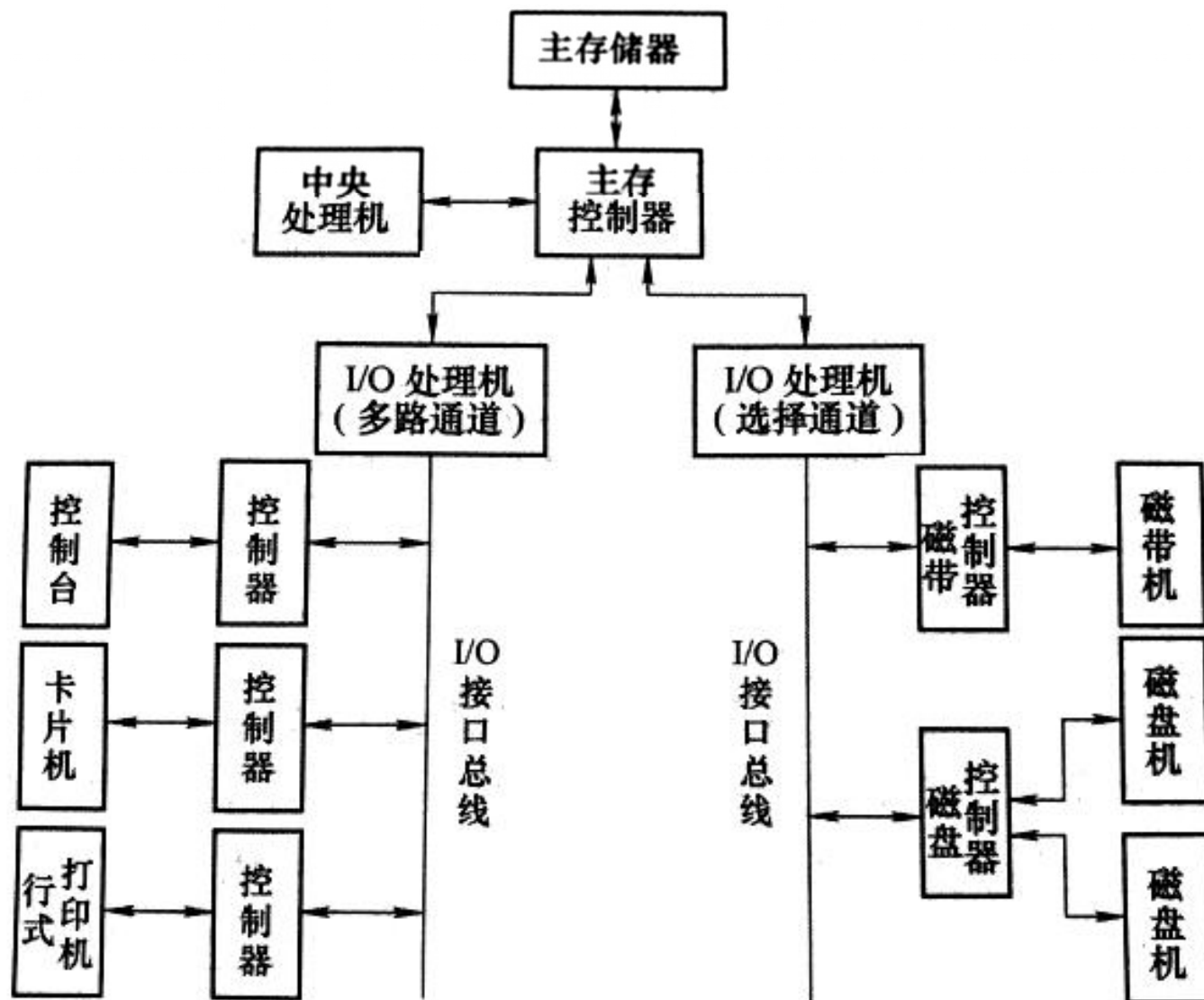
### 2.计算机实现（简单了解）

【例】 IBM370系列有115、125、135、145、158、168等由低档到高档的多种型号机器。从汇编语言、机器语言程序设计者角度看到的概念性结构如图1-2所示。



## 1.2.2计算机组成与计算机实现的定义和内涵

### 2.计算机实现



## 1.2.2 计算机组成与计算机实现的定义和内涵

### 2. 计算机实现

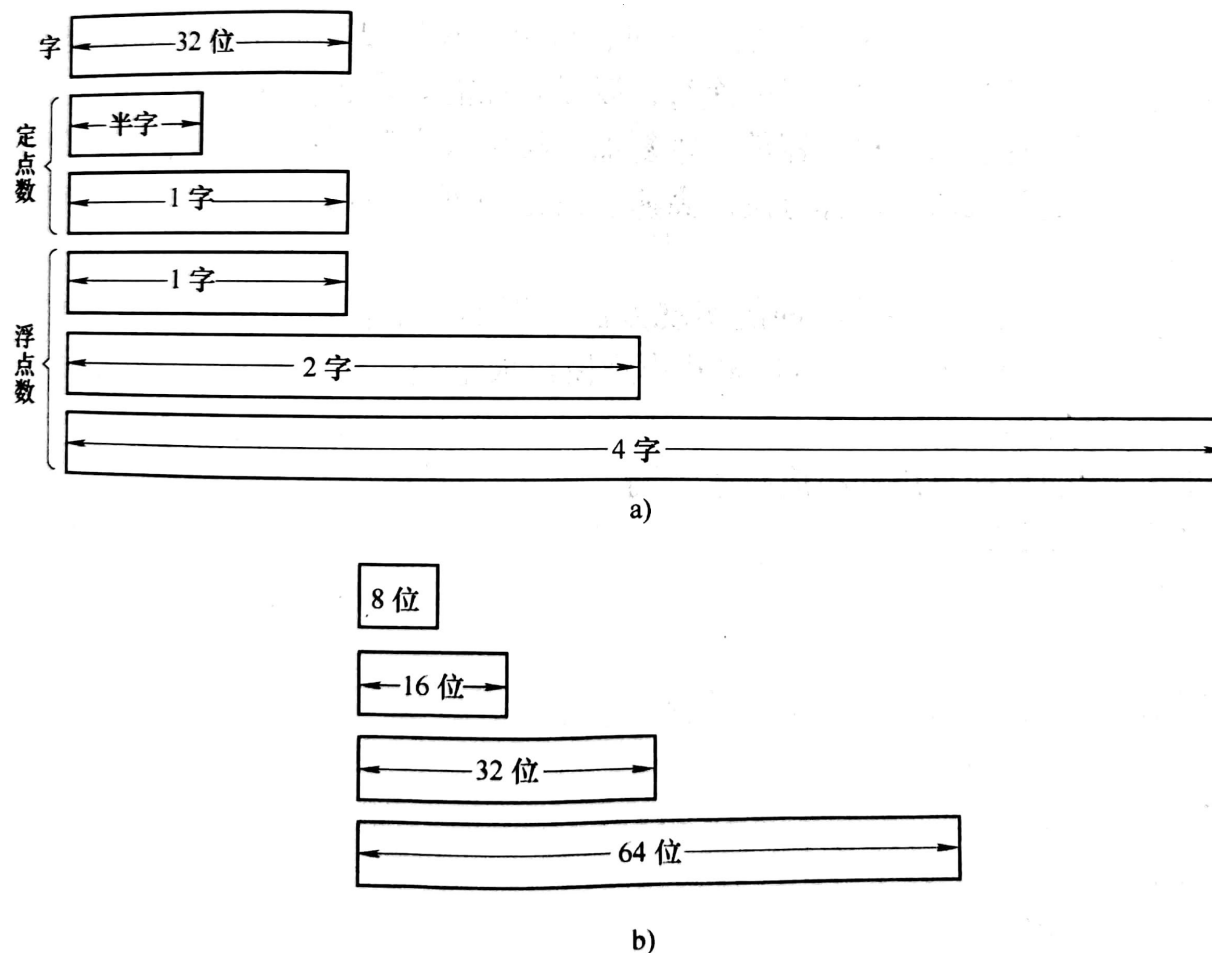


图 1-3 IBM 370 系列字长、数的表示和数据通路宽度

a) 统一的字长与定、浮点数表示 b) 不同的数据通路宽度

## 1.2.3 计算机系统结构、组成和实现的相互关系和影响

相同结构（如指令系统相同）的计算机，可以因速度不同而采用不同的组成。

同样，一种组成可有多种不同的实现方法。

结构不同会使可能采用的组成技术不同。

## 1.3计算机系统的软、硬件取舍及定量设计原理

**本节主要内容：**

**软、硬件实现的优缺点**

**软硬件取舍的基本原则**

**计算机系统的定量设计**

**3种设计思路和存在的问题**

**“由中间开始”的设计方法**

## 1.3.1软、硬件取舍的基本原则

### 软件的功能实现（填空）

软件的功能可以用**硬件**或**固件**完成，硬件的功能也可以用**软件模拟**完成，只是它们在性能、价格、实现的难易程度上是不同的。

## 1.3.1 软、硬件取舍的基本原则

### 取舍的基本原则（简答）

提高硬件功能比例可**提高解题速度**，**减少**程序所需的**存储空间**，**增加硬件成本**，**降低硬件利用率**和计算机系统的**灵活性及适应性**；  
提高软件功能的比例可降低硬件成本，提高系统的灵活性、适应性，但解题速度会下降，软件设计费用和所需的存储器用量增加



## 1.3.1软、硬件取舍的基本原则

原则1 应考虑在现有硬、器件（主要是逻辑器件和存储器件）条件下，系统要有高的性能价格比，主要从实现费用、速度和其他性能要求来综合考虑。

## 1.3.1软、硬件取舍的基本原则

原则2 要考虑到准备采用和可能采用的组成技术，使之尽可能不要过多或不合理地限制各种组成、实现技术的采用。

## 1.3.1 软、硬件取舍的基本原则

原则3 不能仅从“硬”的角度考虑如何便于应用组成技术的成果和便于发挥器件技术的进展，还应从“软”的角度把如何为编译和操作系统的实现以及为高级语言程序的设计提供更多、更好的硬件支持放在首位。

## 1.3.1 软、硬件取舍的基本原则

总结：三个方面（硬件上、技术上、软件上）





那么意气风发地  
走在成功的道路上

## 真题练练手

1、下面说法中不正确的是 ( )

A:软件设计费用比软件重复生产费用高

B:硬件的设计费用比软件的设计费用低

C:硬件的生产费用比软件的生产费用高

D:硬件功能只需实现一次，而软件功能可能要多次重复实现



那么意气风发地  
走在成功的道路上

## 真题练练手

1、下面说法中不正确的是 ( )

A:软件设计费用比软件重复生产费用高

B:硬件的设计费用比软件的设计费用低

C:硬件的生产费用比软件的生产费用高

D:硬件功能只需实现一次，而软件功能可能要多次重复实现

答案：B



那么意气风发地  
走在成功的道路上

## 真题练练手

2、在计算机系统结构设计中，提高软件功能实现的比例可( )

A:提高解题速度

B:减少需要的存储器容量

C:提高系统的灵活性

D:提高系统的性能价格比



那么意气风发地  
走在成功的道路上

## 真题练练手

2、在计算机系统结构设计中，提高软件功能实现的比例可( )

A:提高解题速度

B:减少需要的存储器容量

C:提高系统的灵活性

D:提高系统的性能价格比

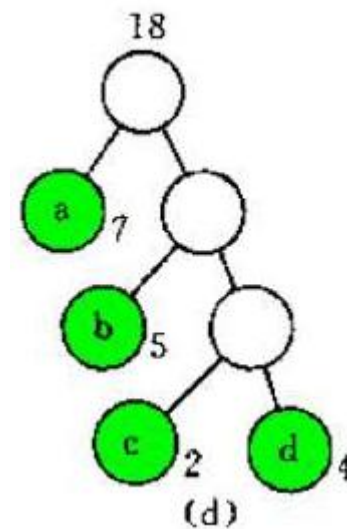
答案：C



## 1.3.2 计算机系统的定量设计原理

### 1. 哈夫曼 (Huffman) 压缩原理 (单选)

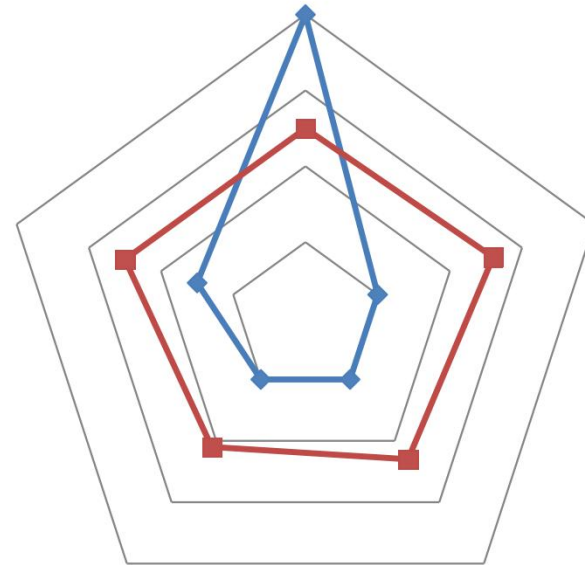
尽可能**加速处理高概率**的事件远比加速处理概率很低的事件对**性能的**  
**提高**要显著。



## 1.3.2 计算机系统的定量设计原理

### 2. Amdahl定律 (单选、填空)

改进效果好的高性能系统应是一个各部分性能均能平衡得到提高的系统，不能只是其中某个功能部件性能的提高。



## 1.3.2 计算机系统的定量设计原理

### 2. Amdahl定律（单选、填空）

改进效果好的高性能系统应是一个各部分性能均能平衡得到提高的系统，不能只是其中某个功能部件性能的提高。

### 3. 程序访问的局部性定律

程序访问的局部性包括了时间上和空间上的两个局部性。

内存

0
1
2
3
4
5
6
7
8
9
...



那么意气风发地  
走在成功的道路上

## 模拟练一练

1、尽可能加速处理高概率的事件远比加速处理概率很低的事件对性能的提高要显著，这是（ ）的内容。

A:Amdahl定律

B:哈夫曼压缩原理

C:局部性定律

D:计算机系统设计



那么意气风发地  
走在成功的道路上

## 模拟练一练

1、尽可能加速处理高概率的事件远比加速处理概率很低的事件对性能的提高要显著，这是（ ）的内容。

A:Amdahl定律

**B:哈夫曼压缩原理**

C:局部性定律

D:计算机系统设计

答案： B



那么意气风发地  
走在成功的道路上

## 模拟练一练

2、程序访问的局部性包括了\_\_\_\_\_上和\_\_\_\_\_上两个局部性。



那么意气风发地  
走在成功的道路上

## 模拟练一练

2、程序访问的局部性包括了\_\_\_\_\_上和\_\_\_\_\_上两个局部性。

答案：时间、空间

## 1.3.3 计算机系统设计的主要任务和方法

### 1. 计算机系统设计的主要任务（填空）

计算机系统设计的主要任务包括系统结构、组成和实现的设计。





## 1.3.3 计算机系统设计的主要任务和方法

### 1. 计算机系统设计的主要任务（填空、选择）

计算机系统设计的主要任务包括系统结构、组成和实现的设计。

它涉及软硬件功能分配、计算机指令系统设计、功能组织、逻辑设计、集成电路设计、封装、电源、冷却等许多方面。

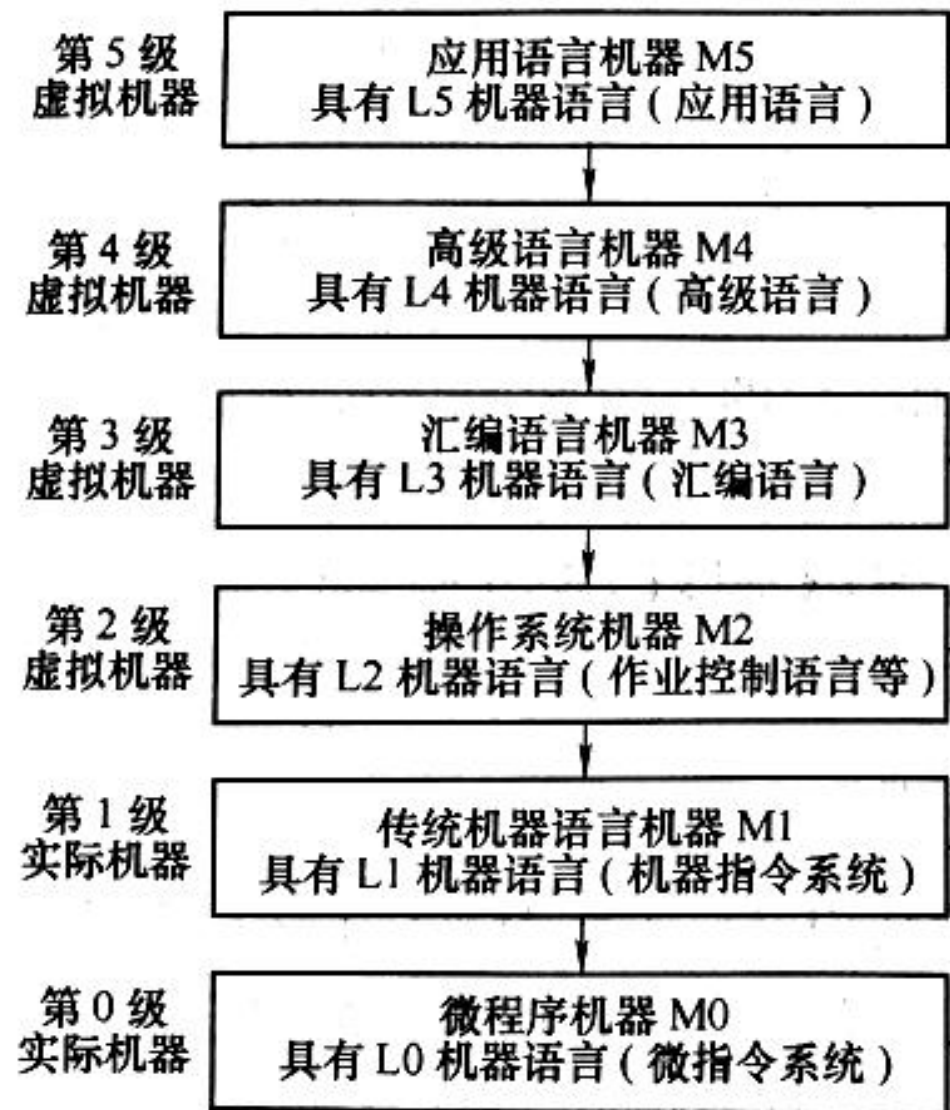
计算机系统设计首先要根据市场和应用情况，确定用户对计算机系统的功能、性能和价格的要求。

其中，应用软件对功能的确定起主要作用。（需求决定一切）

## 1.3.3 计算机系统设计的主要任务和方法

### 1. 计算机系统设计的主要任务（单选）

- 1) 要弄清其应用领域是专用的还是通用的。
- 2) 要弄清软件兼容是放在哪级层次。
- 3) 要弄清对操作系统有何种要求。
- 4) 要如何保证有高的标准化程度。



### 1.3.3 计算机系统设计的主要任务和方法

## 2. 计算机系统的设计方法（单选）

- 1) “由上往下”设计，也称“由顶向底”设计。
- 2) “由下往上”设计，也称“由底向顶”设计。
- 3) “从中间开始”向两边设计。这是通用机一般采用的方法。



那么意气风发地  
走在成功的道路上

## 真题练练手

1、在计算机系统设计中,对功能确定起主要作用的是 ( ) 1810

A:应用软件

B:系统软件

C:机器指令

D:高级语言



那么意气风发地  
走在成功的道路上

## 真题练练手

1、在计算机系统设计中,对功能确定起主要作用的是 ( ) 1810

A:应用软件

B:系统软件

C:机器指令

D:高级语言

答案：A



那么意气风发地  
走在成功的道路上

## 真题练练手

2、“从中间开始”设计的“中间”目前多数是在( ) 0904

A:微程序机器级与汇编语言机器级之间

B:操作系统机器级与汇编语言机器级之间

C:传统机器语言机器级与微程序机器级之间

D:传统机器语言机器级与操作系统机器级之间



那么意气风发地  
走在成功的道路上

## 真题练练手

2、“从中间开始”设计的“中间”目前多数是在( ) 0904

A:微程序机器级与汇编语言机器级之间

B:操作系统机器级与汇编语言机器级之间

C:传统机器语言机器级与微程序机器级之间

D:传统机器语言机器级与操作系统机器级之间



那么意气风发地  
走在成功的道路上

## 真题练练手

3、计算机系统的设计思路中，较好的是( ) 1204

A:由上往下

B:由下往上

C:由中间开始

D:由两边向中间





那么意气风发地  
走在成功的道路上

## 真题练练手

3、计算机系统的设计思路中，较好的是( ) 1204

A:由上往下

B:由下往上

C:由中间开始

D:由两边向中间

## 1.4软件、应用、器件的发展对系统结构的影响

### 1.4.1软件发展对系统结构的影响

1.统一高级语言

2.采用系列机

例如, IBM370,115.....

苹果手机11系列

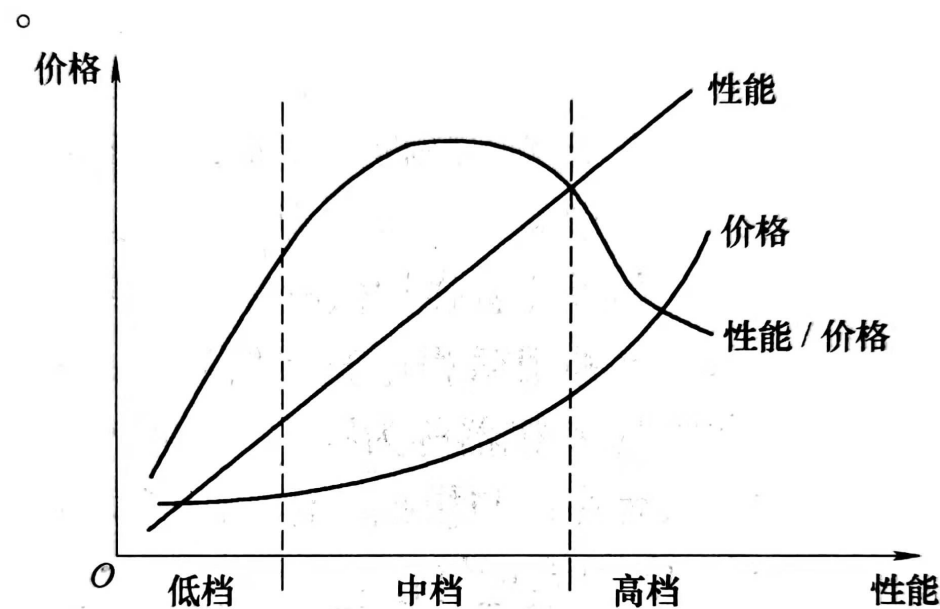


图 1-5 系列机中各档机器的性能价格比状况

## 1.4软件、应用、器件的发展对系统结构的影响

### 1.4.1软件发展对系统结构的影响

- 1.统一高级语言
- 2.采用系列机
- 3.模拟和仿真（单选）
  - (1) 模拟

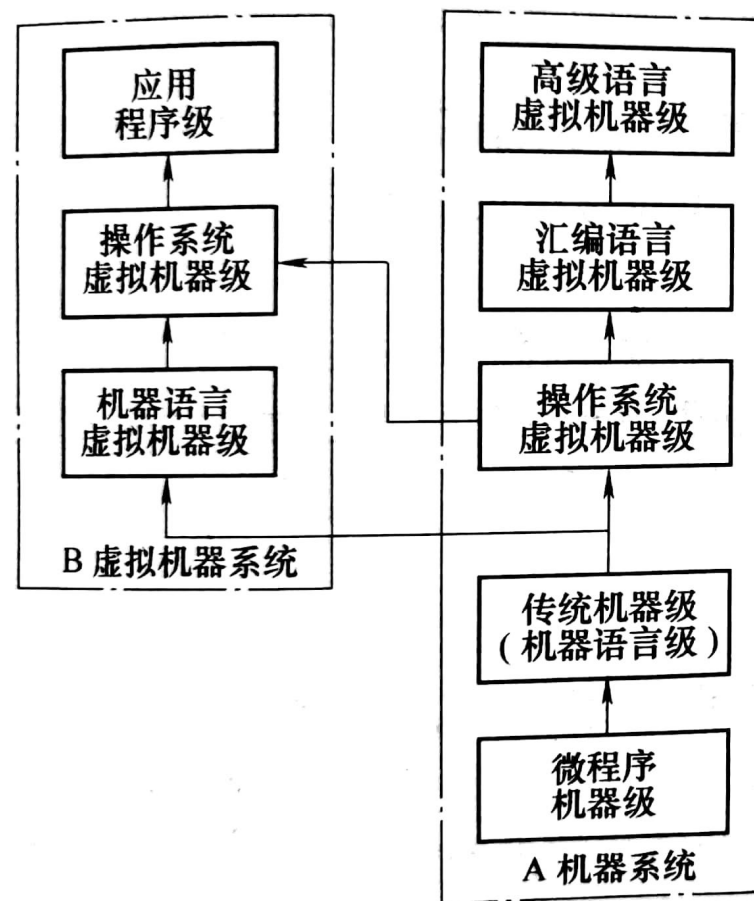


图 1-6 用模拟方法实现应用软件的移植

## 1.4软件、应用、器件的发展对系统结构的影响

### 1.4.1软件发展对系统结构的影响

- 1.统一高级语言
- 2.采用系列机
- 3.模拟和仿真（单选）
  - (1) 模拟
  - (2) 仿真

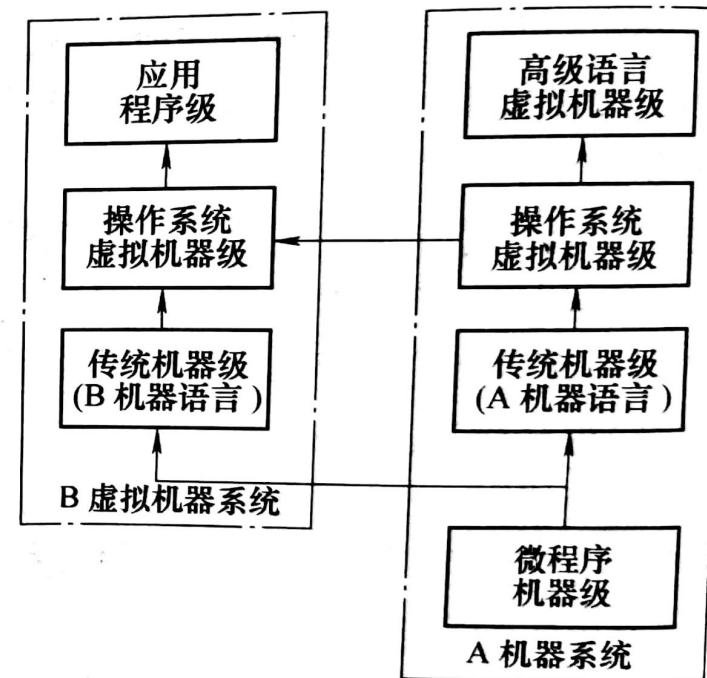


图 1-7 用仿真方法实现应用程序的移植

## 1.4软件、应用、器件的发展对系统结构的影响

### 1.4.1软件发展对系统结构的影响

1.统一高级语言

2.采用系列机

3.模拟和仿真（单选）

（1）模拟      （2）仿真      （3）模拟和仿真的选择

仿真和模拟的主要区别：

模拟是用**机器语言程序**解释，其解释程序存储于**主存**中；

仿真是用**微程序**解释，其解释程序存储与**控制存储器**中。

## 1.4软件、应用、器件的发展对系统结构的影响

### 1.4.2应用的发展对系统结构的影响(填空)

计算机应用可归纳为向上升级的4类，它们是**数据处理**（Data Processing）、**信息处理**（Information Processing）、**知识处理**（Knowledge Processing）和**智能处理**（Intelligence Processing）。



那么意气风发地  
走在成功的道路上

## 真题练练手

1、计算机应用可归纳为向上升级的4类：数据处理、\_\_\_\_\_、\_\_\_\_\_和智能处理。



那么意气风发地  
走在成功的道路上

## 真题练练手

1、计算机应用可归纳为向上升级的4类：数据处理、\_\_\_\_\_、\_\_\_\_\_和智能处理。

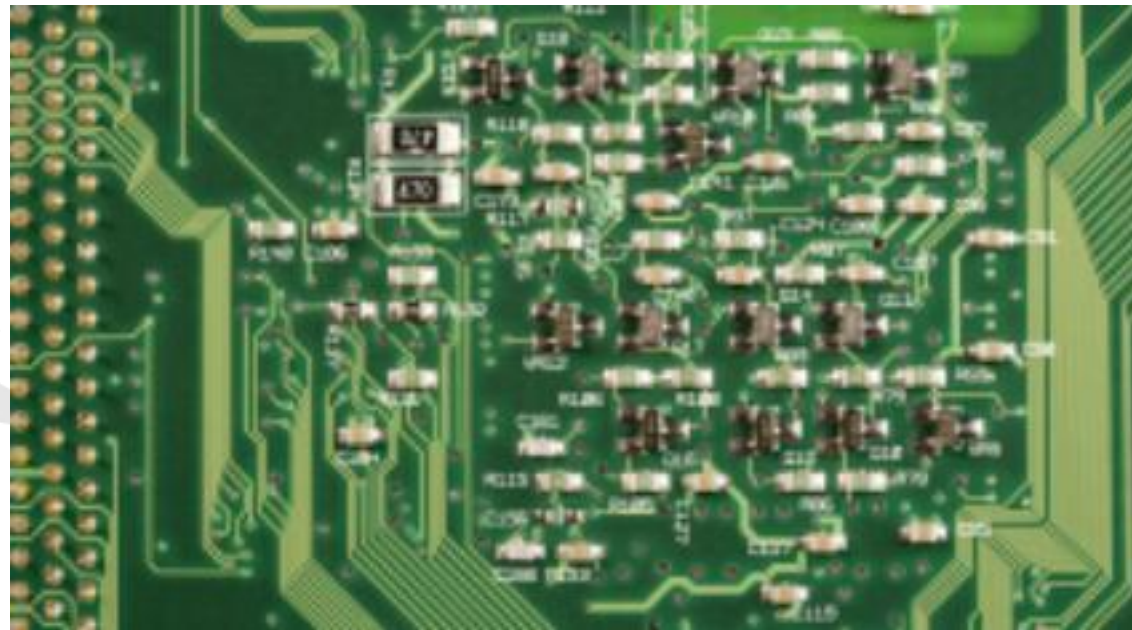
答案：信息处理      知识处理



## 1.4软件、应用、器件的发展对系统结构的影响

### 1.4.3器件的发展对系统结构的影响（简答）

(1) 器件**集成度的提高**，使器件的速度迅速提高，机器主频和速度也有数量级的提高；



## 1.4软件、应用、器件的发展对系统结构的影响

### 1.4.3器件的发展对系统结构的影响（简答）

- (1) 器件**集成度的提高**，使器件的速度迅速提高，机器主频和速度也有数量级的提高；
- (2) 器件**可靠性**有数量级的提高，保证流水技术的实现；
- (3) **高速、廉价的半导体存储器**的出现，使解题速度得以迅速提高的高速缓冲存储器和虚拟存储器的概念真正实现；（价格降低）
- (4) 现场型**PROM器件**，使微程序技术得以实现；（新材料）
- (5) **高速相联存储器**的实现，促进相联处理机这种结构的发展，推动向量机、数组机、数据库机的发展。（新技术）

**二功二料一技术**

要求，不要迟到早退，在线听直播2小时+随堂考最后一节课可获得汇总资料

## 1.4软件、应用、器件的发展对系统结构的影响

### 1.4.3器件的发展对系统结构的影响（简单了解）

结论：软件、应用、器件对计算机系统结构的发展有着很大的影响，反过来，计算机系统结构的发展又会对软件、应用、器件提出新的发展要求。

结构设计不仅要了解结构、组成、实现，还要充分了解软件、应用、器件的发展，这样才能对计算机系统结构进行有效的设计、研究和探索。

# 1.5系统结构中的并行性开发及计算机系统的分类

## 1.5.1并行性的概念与开发

### 1.并行性的含义与级别（单选、填空）

并行性包含**同时性**和**并发性**二重含义。

1) 从计算机系统**执行程序**的角度来看，并行性等级由低到高可分为四级

①指令内部——一条指令内部各个微操作之间的并行执行。

②指令之间——多条指令的并行执行。

③任务或进程之间——多个任务或程序段的并行执行。

④作业或程序之间——多个作业或多道程序的并行执行。

# 1.5 系统结构中的并行性开发及计算机系统的分类

## 1.5.1 并行性的概念与开发

2) 从计算机系统中**处理数据**的角度来看，并行性等级从低到高可以分为四级。

- ① 位串字串
- ② 位并字串
- ③ 位片串字并
- ④ 全并行

# 1.5系统结构中的并行性开发及计算机系统的分类

## 1.5.1 并行性的概念与开发

2) 从计算机系统中**处理数据**的角度来看，并行性等级从低到高可以分为四级。

①位串字串——同时只对一个字的一位进行处理，这通常是指传统的串行单处理机，没有并行性。(00**1**10100, 11010100)

②位并字串——同时对一个字的全部位进行处理，这通常是指传统的并行单处理机，开始出现并行性。(00**110100**, 11010100)

③位片串字并——同时对许多字的同一位（称位片）进行处理，开始进入并行处理领域。(00**1**10100, 11**0**10100)

④全并行——同时对许多字的全部或部分位组进行处理 (00**110100**, **11010100**)

# 1.5系统结构中的并行性开发及计算机系统的分类

## 1.5.1并行性的概念与开发

3) 并行性是贯穿于计算机**信息加工**的各个步骤和阶段的, 从这个角度来看, 并行性等级又分为:

- ①存储器操作并行
- ②处理器操作步骤并行
- ③处理器操作并行
- ④指令、任务、作业并行

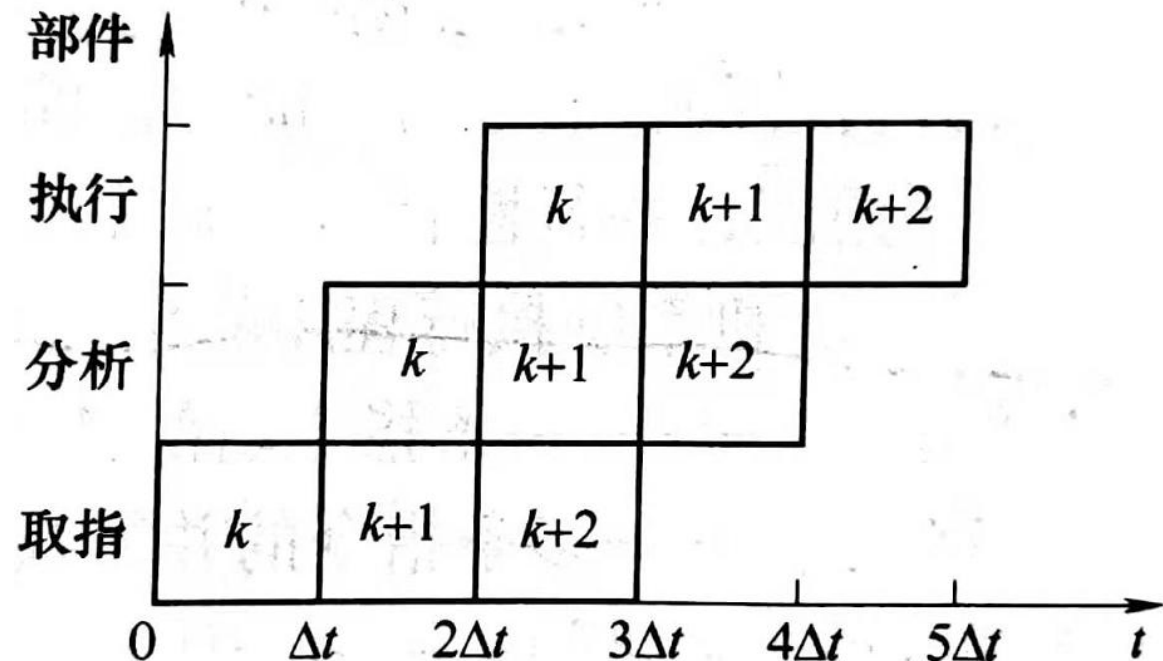


# 1.5系统结构中的并行性开发及计算机系统的分类

## 1.5.1并行性的概念与开发

### 2.并行性开发的途径（单选、填空）

- (1) 时间重叠
- (2) 资源重复
- (3) 资源共享





## 1.5 系统结构中的并行性开发及计算机系统的分类

### 1.5.1 并行性的概念与开发

2. 并行性开发的途径（单选、填空）

- (1) 时间重叠
- (2) 资源重复
- (3) 资源共享

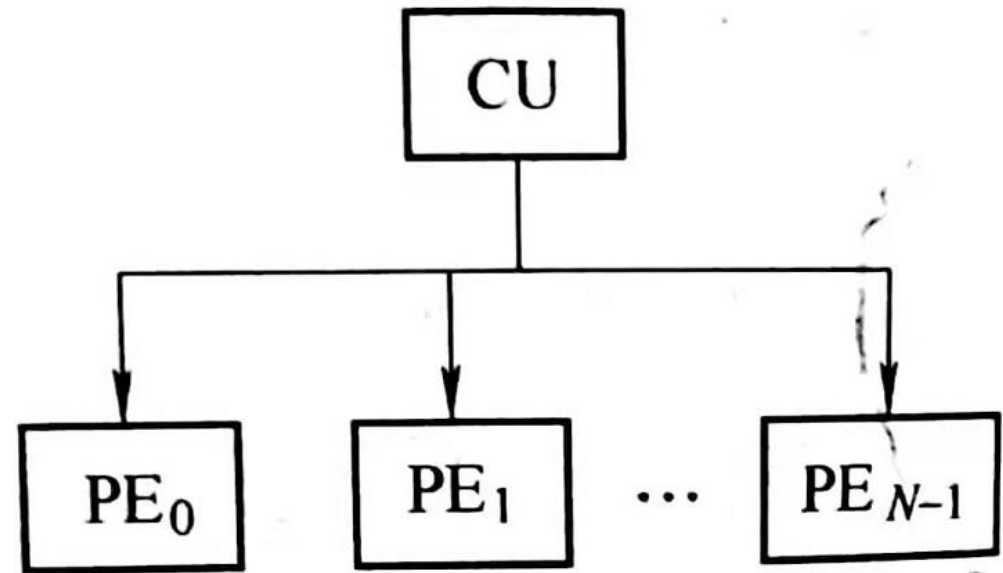


图 1-10 资源重复

# 1.5系统结构中的并行性开发及计算机系统的分类

## 1.5.2计算机系统的分类（单选）

1966年，弗林（Michael J·Flynn）按指令流和数据流的多倍性把计算机系统分成

- (1) 单指令流单数据流（Single Instruction Stream Single Data Stream, **SISD**）、
- (2) 单指令流多数据流(Single Instruction Stream Multiple Data Stream, **SIMD**)、
- (3) 多指令流单数据流（**MISD**）
- (4) 多指令流多数据流（**MIMD**）

要求，不要迟到早退，在线听直播2小时+随堂考最后一节课可获得汇总资料

## 1.5 系统结构中的并行性开发及计算机系统的分类

### 1.5.2 计算机系统的分类（单选）

1972年，美籍华人冯泽云（Tse-yun Feng）提出了用**数据处理的并行度**来定量地描述 各种计算机系统特性的冯氏分类法。他把计算机系统分成四类：

- 1) 字串位串（WSBS）——称位串处理方式。
- 2) 字串位并（WSBP）——称字（字片）处理方式。
- 3) 字并位串（WPBS）——称位（位片）处理方式。
- 4) 字并位并（WPBP）——称全并行处理方式。

## 1.5系统结构中的并行性开发及计算机系统的分类

### 1.5.2计算机系统的分类（单选）

1978年，美国的库克（David J.Kuck）提出用指令流和执行流（Execution Stream）及其多倍性来描述计算机系统总控制器的结构特点。

- 1) 单指令流单执行流（SISE）——典型的单处理机系统。
- 2) 单指令流多执行流（SIME）——带多操作部件的处理机。
- 3) 多指令流单执行流（MISE）——带指令级多道程序的单处理机。
- 4) 多指令流多执行流（MIME）——典型的多处理机系统。



那么意气风发地  
走在成功的道路上

## 真题练练手

- 1、（ ）指的是计算机系统结构的逻辑实现，包括及其级内部的数据流和控制流的组成以及逻辑设计等
- A、计算机组成
  - B、系统结构
  - C、计算机实现
  - D、计算机系统设计



那么意气风发地  
走在成功的道路上

## 真题练练手

1、（ ）指的是计算机系统结构的逻辑实现，包括及其级内部的数据流和控制流的组成以及逻辑设计等

A、计算机组成

B、系统结构

C、计算机实现

D、计算机系统设计

答案：A

解析：计算机组成指的是计算机系统结构的逻辑实现，包括及其级内部的数据流和控制流的组成以及逻辑设计等。



那么意气风发地  
走在成功的道路上

## 真题练练手

- 2、（ ）是在并行概念中引入空间因素，通过重复设置硬件资源来提高可靠性或性能
- A、资源重复
  - B、时间重叠
  - C、资源共享
  - D、时间重复



那么意气风发地  
走在成功的道路上

## 真题练练手

2、（ ）是在并行概念中引入空间因素，通过重复设置硬件资源来提高可靠性或性能

A、资源重复

B、时间重叠

C、资源共享

D、时间重复

答案：A

解析：资源重复是在并行概念中引入空间因素，通过重复设置硬件资源来提高可靠性或性能。





那么意气风发地  
走在成功的道路上

## 真题练练手

3、（ ）是用软件方法，让多个用户按一定时间顺序轮流使用同一套资源来提高资源利用率，相应地也就提高了系统的性能

- A、资源共享
- B、时间重叠
- C、资源重复
- D、时间重复



那么意气风发地  
走在成功的道路上

## 真题练练手

3、（ ）是用软件方法，让多个用户按一定时间顺序轮流使用同一套资源来提高资源利用率，相应地也就提高了系统的性能

A、资源共享

B、时间重叠

C、资源重复

D、时间重复

答案：A

解析：资源共享是用软件方法，让多个用户按一定时间顺序轮流使用同一套资源来提高资源利用率，相应地也就提高了系统的性能。



那么意气风发地  
走在成功的道路上

## 真题练练手

- 4、（ ）称串处理方式，每次只处理一个字中的一位，如早期的位串行机
- A、字串位并
  - B、字串位串
  - C、字并位并
  - D、字并位串



那么意气风发地  
走在成功的道路上

## 真题练练手

4、（ ）称串处理方式，每次只处理一个字中的一位，如早期的位串行机

A、字串位并

**B、字串位串**

C、字并位并

D、字并位串

答案：B

解析：字串位串（WSBS）——称串处理方式，每次只处理一个字中的一位，如早期的位串行机

## 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。

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



## 2.1.2高级数据表示

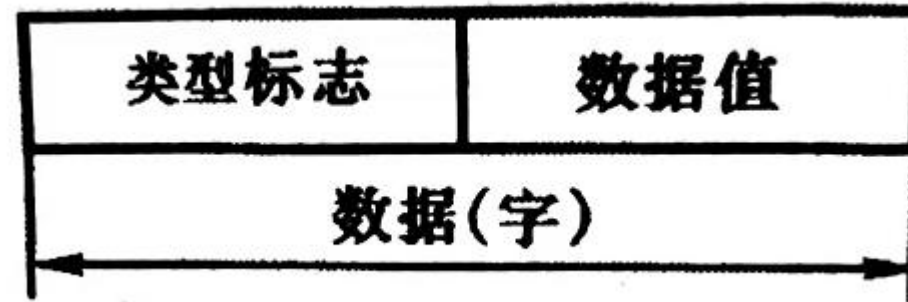
### 1.自定义数据表示（填空，简答）

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。



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

## 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。

## 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。

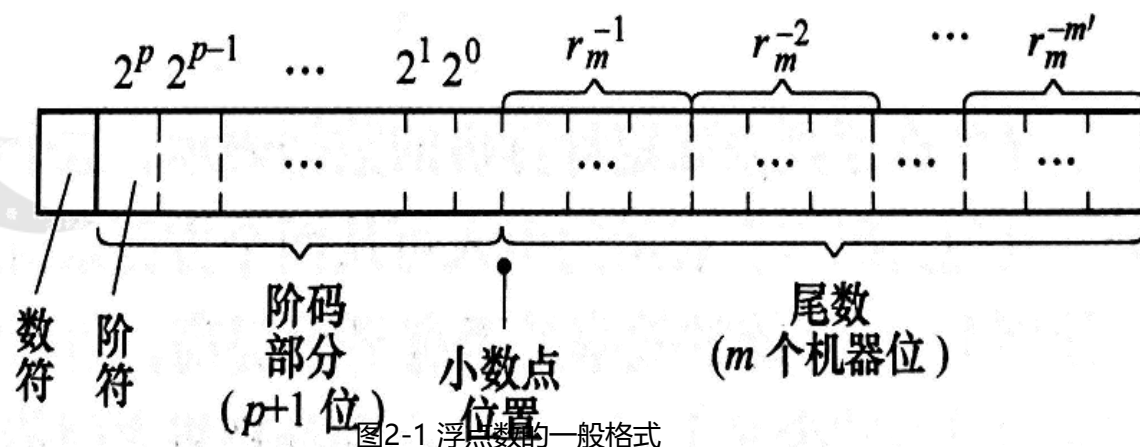
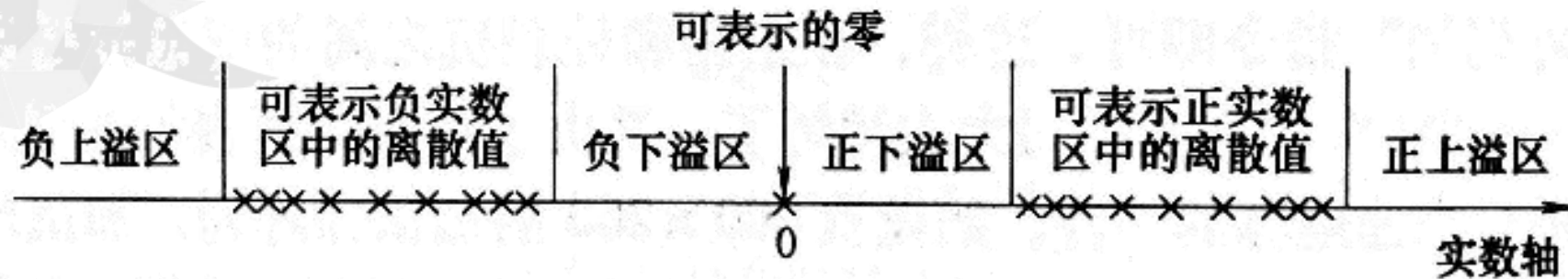


图2-1 浮点数的一般格式

## 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

rm浮点数尾数的基数

这个基数要占多少位： $\log_2 rm$

m尾数的计算机位数

$m'$ ：rm进制的尾数的数位

【重点】 $m' = m / \log_2 rm$  (rm、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

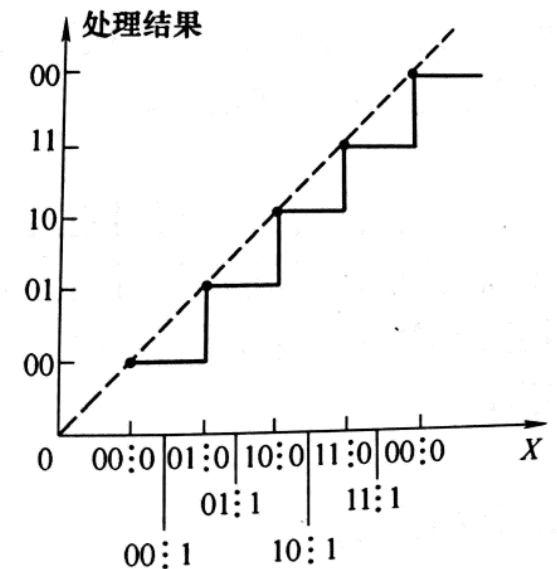
答案: B

## 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



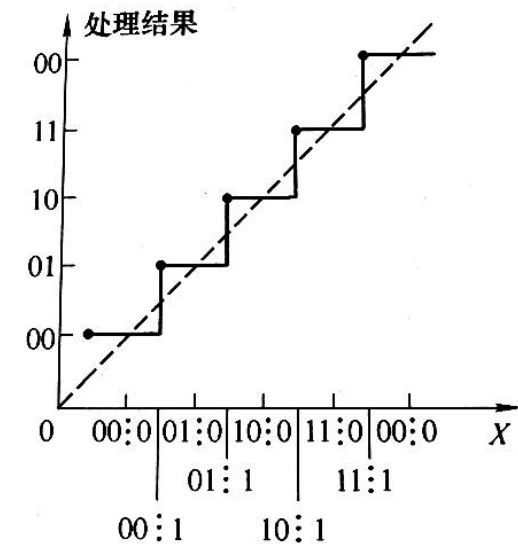
## 2.1.4浮点数尾数基值大小和下溢处理方法的选择

### 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。

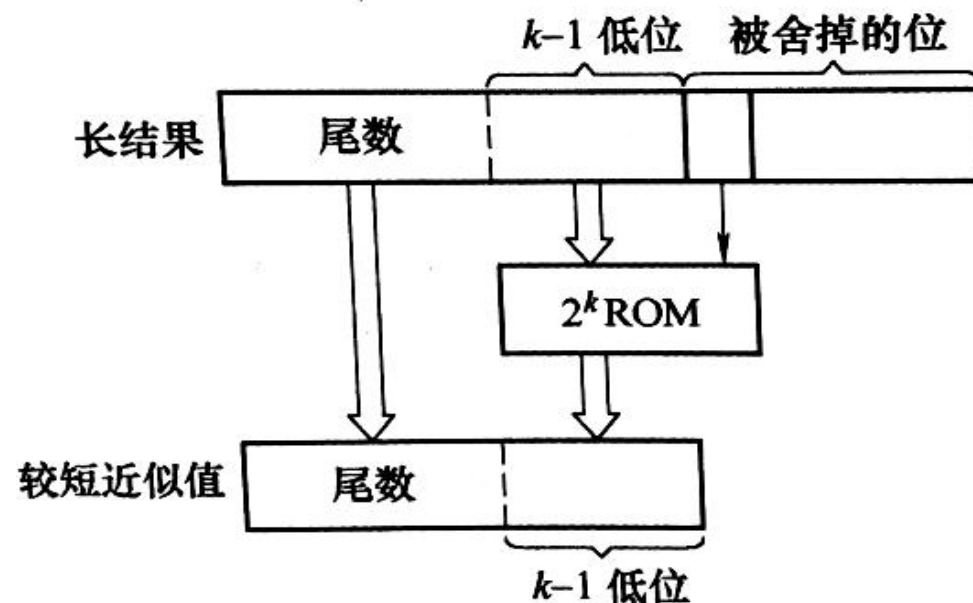
主要缺点是最大误差最大，比截断法的还要大。

## 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。

要求, 不要迟到早退, 在线听直播2小时+随堂考最后一节课可获得汇总资料

## 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。

## 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。

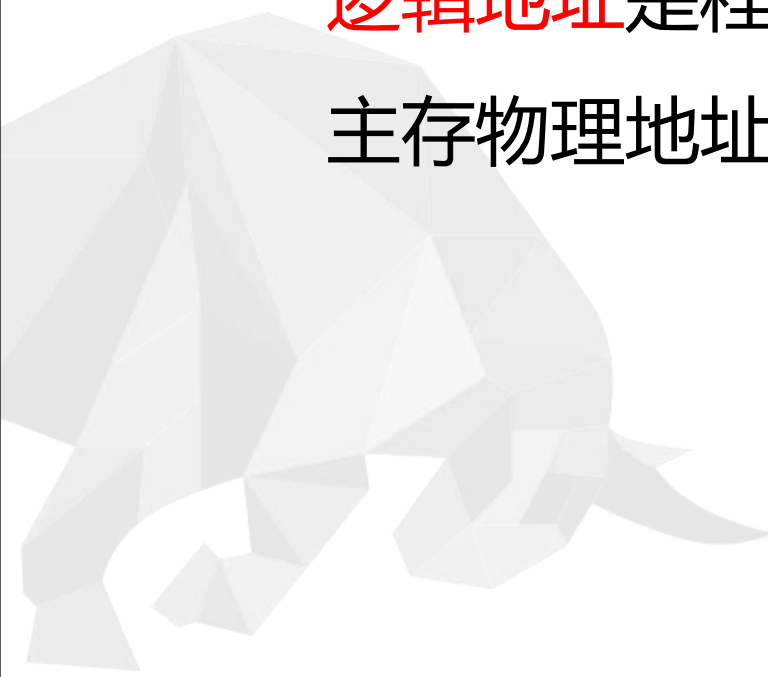
另一种方式是不占用操作码，而是在**地址码**部分专门设置寻址方式位字段指明。

## 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。

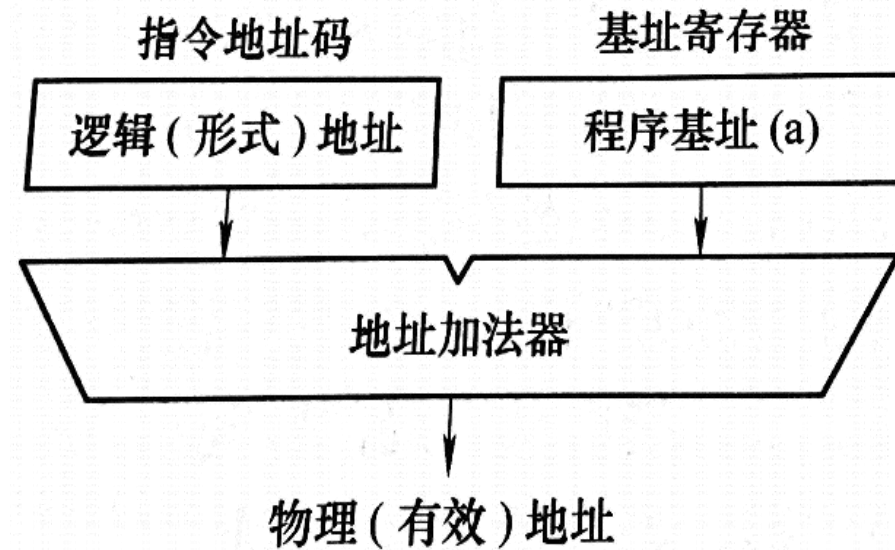


图2-6 基址寻址



## 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。

## 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

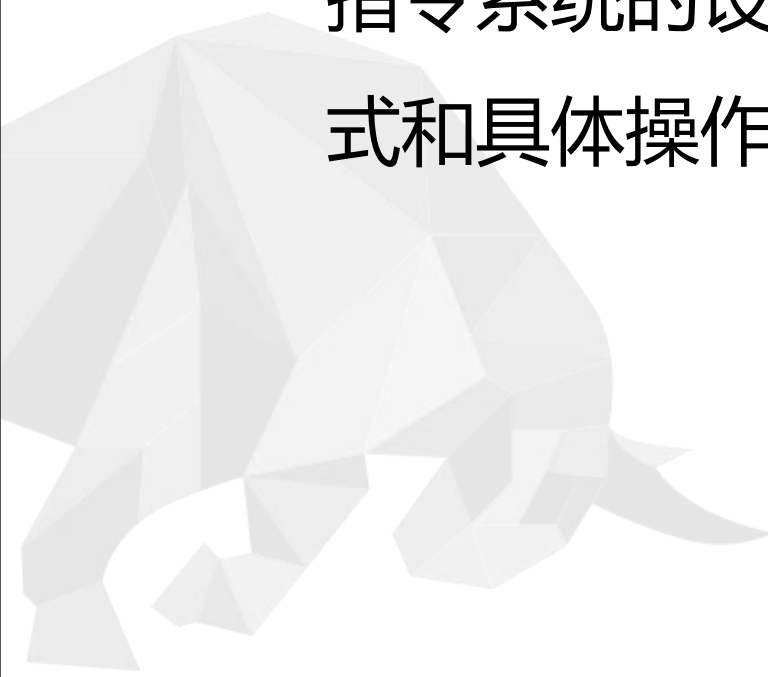
### 2.3.1 指令系统设计的基本原则

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。

## 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	—

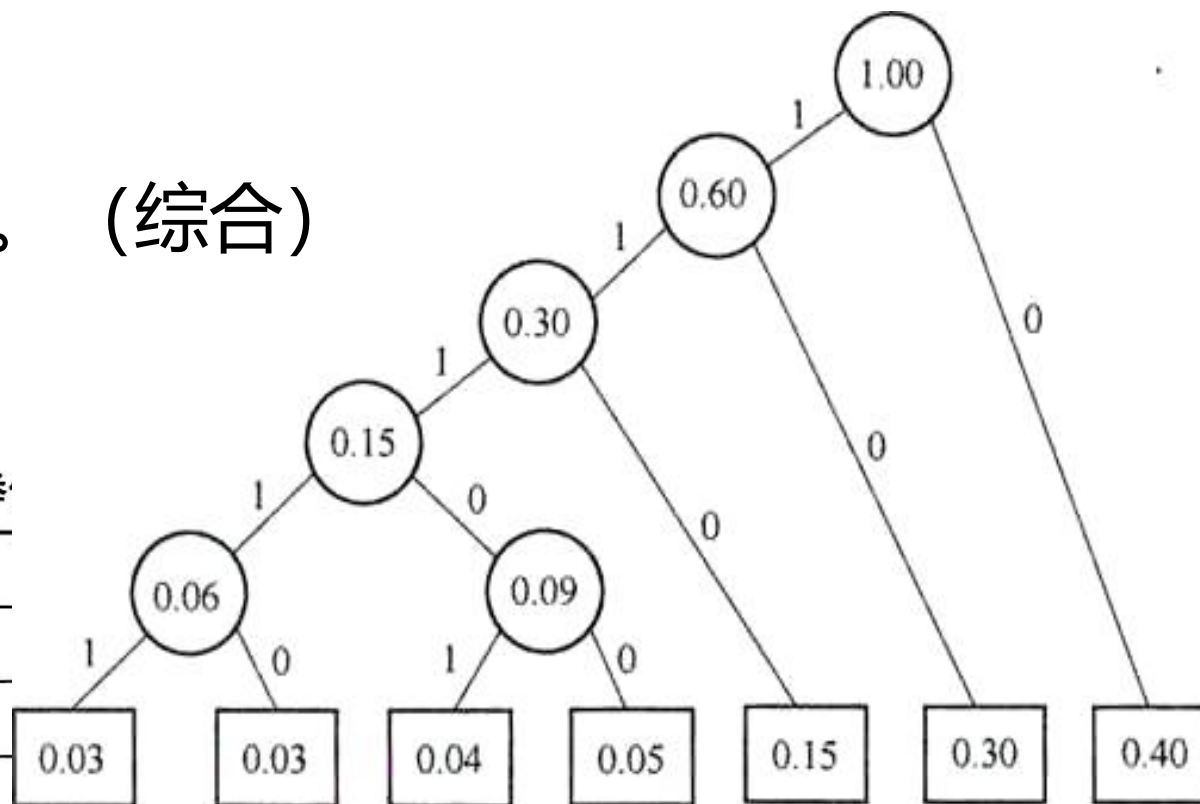


图 2-9 哈夫曼树举例

## 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。

要求，不要迟到早退，在线听直播2小时+随堂考最后一节课可获得汇总资料





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方向发张、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。

要求，不要迟到早退，在线听直播2小时+随堂考最后一节课可获得汇总资料

## 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

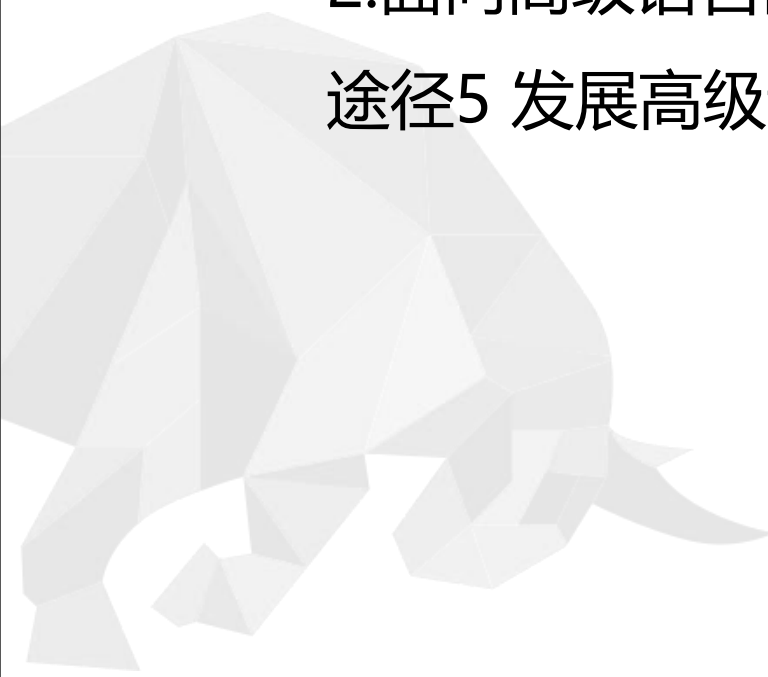
由“以指令系统为主，高级语言为从”方式演变成“以高级语言为主，指令系统为从”的方式。

## 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

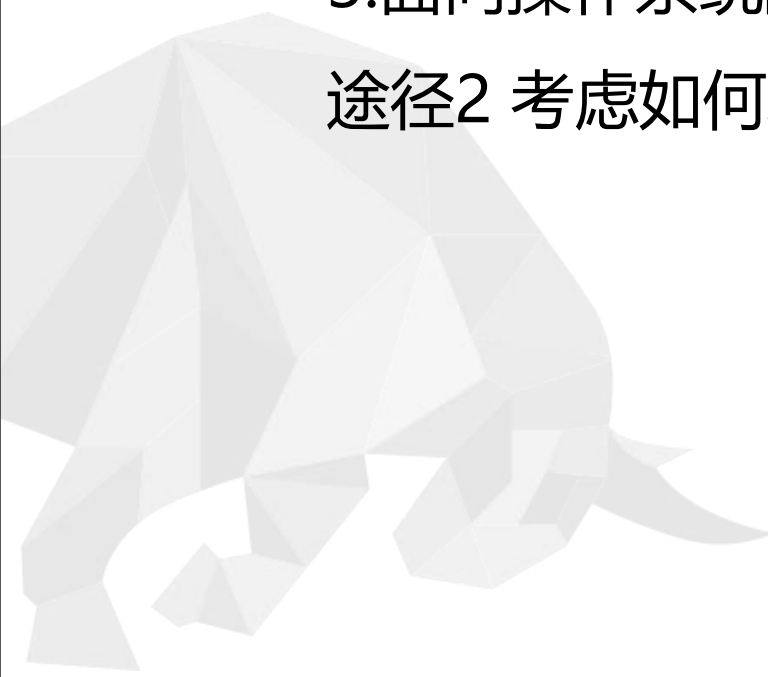
途径1 通过对操作系统中常用指令和指令串的使用频度进行统计分析来改进。但这种改进的效果很有限。

## 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**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**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**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;  $(A) \rightarrow Ra$

存 Ra, B;  $Ra \rightarrow (B)$

取  $A + 1$ , Ra;  $(A+1) \rightarrow Ra$

存 Ra,  $B + 1$ ;  $Ra \rightarrow (B + 1)$



## 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 ◀

