



# 计算机组成原理

理论课：共15次，每周一次

实验课：共6~7次，两周一次

考试：70%期末+30%作业（布置4~5次）

答疑：每周三3,4节（16-306）+双周上机时间  
+课间时间

学习方法：上课认真听讲；细致阅读教材；充分利用答疑时间，解决疑难问题；注意平时作业，多动手，不要抄袭。





# 这门课在计算机科学中的位置？





# 第1章 计算机系统概论

1.1 计算机发展历程

1.2 计算机系统的层次结构✓

1.3 计算机系统的组成✓

1.4 计算机的工作过程✓

1.5 计算机性能✓

1.6 计算机的分类和应用

本章小结

习题1





## 1.1 计算机发展历程 ■

电子数字计算机是一种能够自动、高速、精确地进行信息处理的现代化电子设备。其基本特点是运算速度快、记忆能力强、有逻辑判断能力。电子数字计算机最初只是作为一种计算工具，现已应用于人类生产和生活的各个方面。 ■





## 1.1.1 冯·诺依曼型计算机的特点和功能

■ 1943年开始研制的ENIAC，美国第一台由程序控制的电子数字计算机，新型火炮的弹道进行计算。

■ 1950年第一台存储程序计算机EDVAC(Electronic Discrete Variable Automatic Computer)诞生，由美籍匈牙利数学家冯·诺依曼与莫尔小组合作研制。特点：（1）EDVAC计算机采用了二进制的表示方法，并设置了能对二进制数进行运算的算术逻辑运算单元(ALU)；（2）该计算机采用**存储程序工作方式**，程序存储在主存储器中，可以随时修改。





## 冯·诺依曼计算机

冯·诺依曼在他发表的《关于电子计算装置逻辑结构初探》报告中提出了存储程序的思想。该思想可以概括如下：

计算机要自动完成解题任务，必须将事先设计好的、用以描述计算机解题过程的程序**如同数据一样**采用二进制形式存储在机器中，计算机在工作时自动高速地从机器中逐条取出指令加以执行。 ■



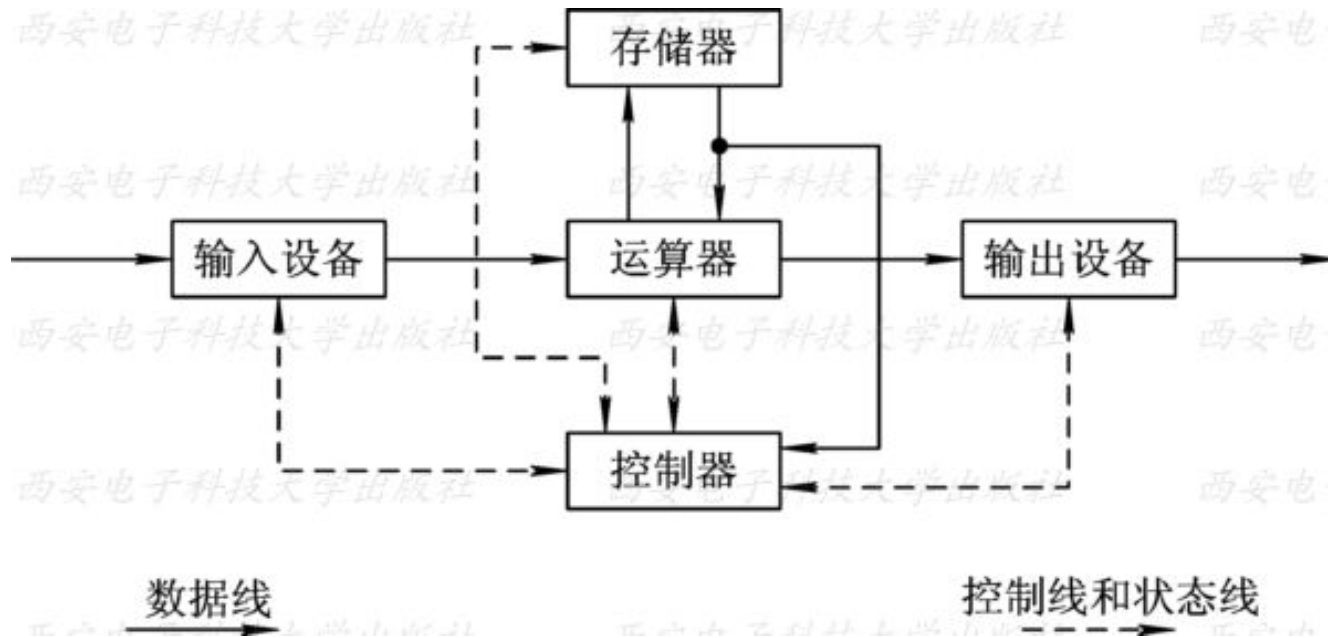


图1.1 冯·诺依曼型计算机硬件组成图





按照存储程序原理，冯·诺依曼型计算机必须具备**五大功能**：

(1) **输入/输出功能**。计算机必须有能把原始数据和解题步骤输入到机器中，同时也可以把计算结果和计算过程中的情况输出给使用者。

(2) **记忆功能**。计算机能够存储(记忆)原始数据和解题步骤，以及解题过程中产生的一些中间结果。

(3) **计算功能**。计算机应能进行一些基本的计算，并能利用这些基本计算组合成使用者所需的一切计算。

(4) **判断功能**。计算机在完成一步操作后，应具备从预先无法确定的几种方案中选择一种操作方案的能力，以保证解题过程的正确性。

(5) **自我控制功能**。计算机应能保证程序执行的正确性和各部件之间的协调性。







## 非冯·诺依曼模型:

神经网络——利用人脑模型的思想作为计算范式

基因算法——利用生物学和DNA演化的思想开发的算法

量子计算机——在量子效应和量子力学基础上开发的进行高速  
数学和逻辑运算、存储及处理量子信息的计算机

光子计算机——利用光子取代电子进行数据运算、传输和存  
储的计算机

纳米计算机——采用纳米技术研发的新型高性能计算机  
大规模并行处理机等





## 纳米

一个纳米等于 $10^{-9}$ 米，大约是氢原子直径的10倍。纳米技术正从微电子机械系统起步，把传感器、电动机和各种处理器都放在一个硅芯片上而构成一个系统。应用纳米技术研制的计算机内存芯片，其体积只有数百个原子大小，相当于人的头发丝直径的千分之一。纳米计算机不仅几乎不需要耗费任何能源，而且其性能要比今天的计算机强大许多倍。

## 生物

20世纪80年代以来，生物工程学家对人脑、神经元和感受器的研究倾注了很大精力，以期研制出可以模拟人脑思维、低耗、高效的第六代计算机——生物计算机。

## 神经

它是一种有知识、会学习、能推理的计算机，具有能理解自然语言、声音、文字和图像的能力，并且具有说话的能力，使人机能够用自然语言直接对话，它可以利用已有的和不断学习到的知识，进行思维、联想、推理，并得出结论，能解决复杂问题，具有汇集、记忆、检索有关知识的能力。





## 1.2 计算机系统的层次结构及计算机组成 ■

### 1.2.1 计算机系统的层次结构 ■

现代计算机的解题过程通常是先由用户用高级语言编写程序(称做源程序), 然后将它和数据一起送入计算机内, 再由计算机将其翻译成机器能识别的机器语言程序(称做目标程序), 机器自动运行该机器语言程序, 并将结果输出。计算机的解题过程如图1.2所示。



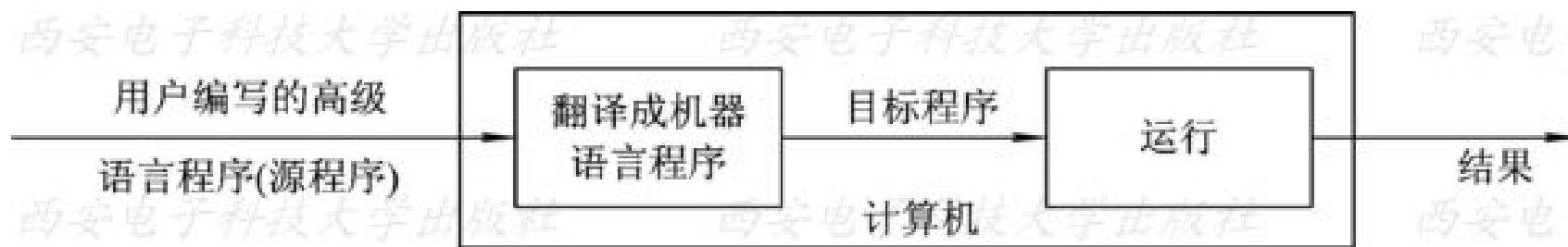


图1.2 计算机的解题过程



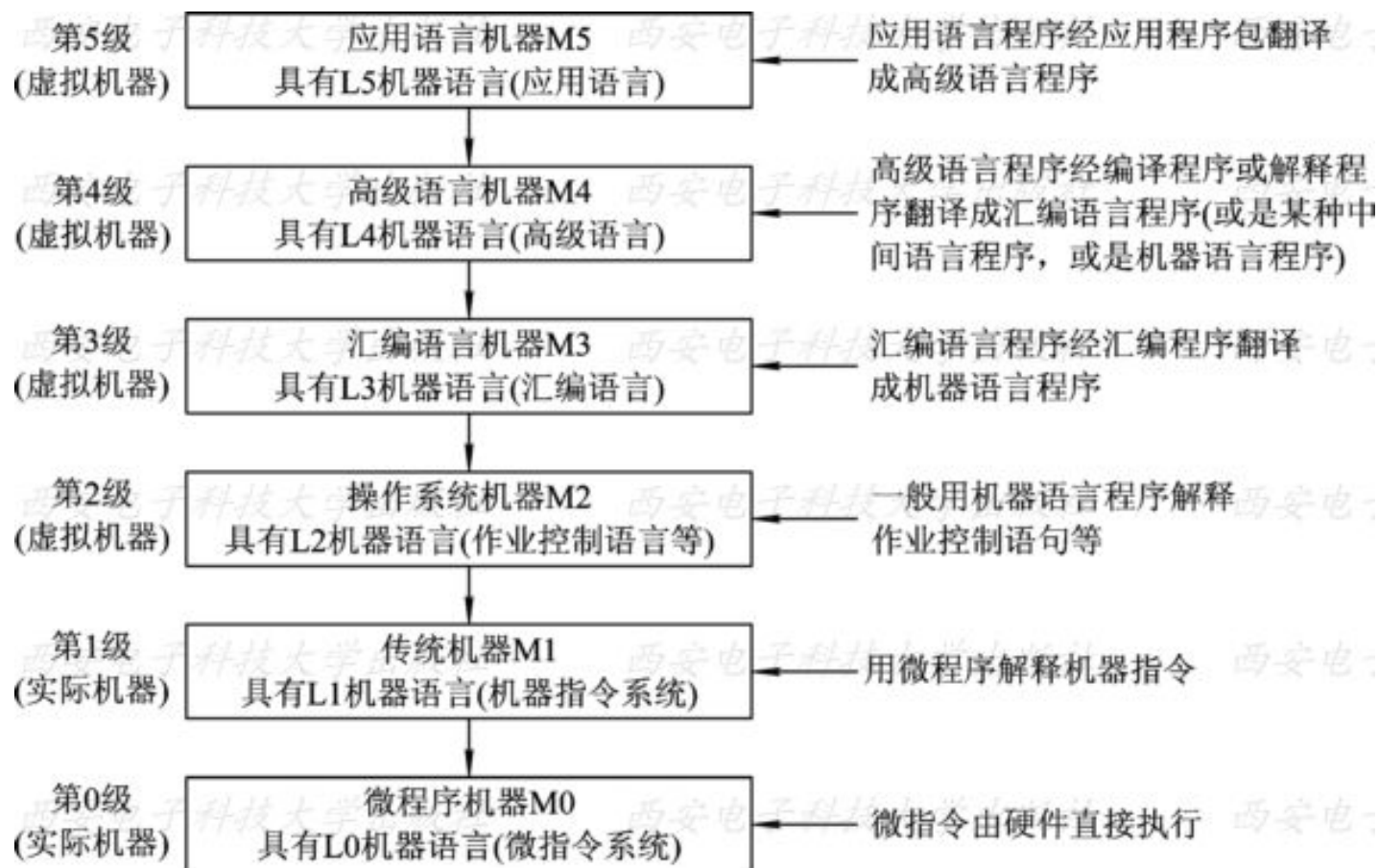


图1.4 计算机系统的多级层次结构





## 1.2.2 计算机系统结构和计算机组成

在学习计算机组成时，应当注意如何区分计算机系统结构与计算机组成这两个基本概念。

■ Amdahl等人把**系统结构**定义为：**由程序设计者所看到的一个计算机系统的属性，即概念性结构和功能特性。**计算机系统结构研究的主要内容是计算机系统的多层次结构中各级之间界面的定义及其上下的功能分配。

■ **计算机组成**又称**计算机设计**，是指**计算机系统结构的逻辑实现。**它研究的内容主要包括**机器内部的数据流和控制流的组成以及逻辑设计等。**本教材主要讨论**传统机器M1和微程序机器M0**的组成原理及设计思想，其他各级虚拟机的内容，将在其他的软件课程中讲





## 1.3 计算机系统的组成 ■

一个完整的计算机系统由硬件和软件两大部分组成。所谓硬件，是指计算机中的电子线路和物理装置。它由看得见摸得着的各种电子元器件、各类光学、电子、机械设备的实物组成，如主机、外设等。所谓软件，按照国际标准化组织(ISO)的定义，是指计算机程序及运用数据处理系统所必需的手续、规则和文件的总称。软件通常存储在介质上，人们可以看到的是存储软件的介质，而软件本身是看不见摸不着的。





### 1.3.1 计算机的硬件系统

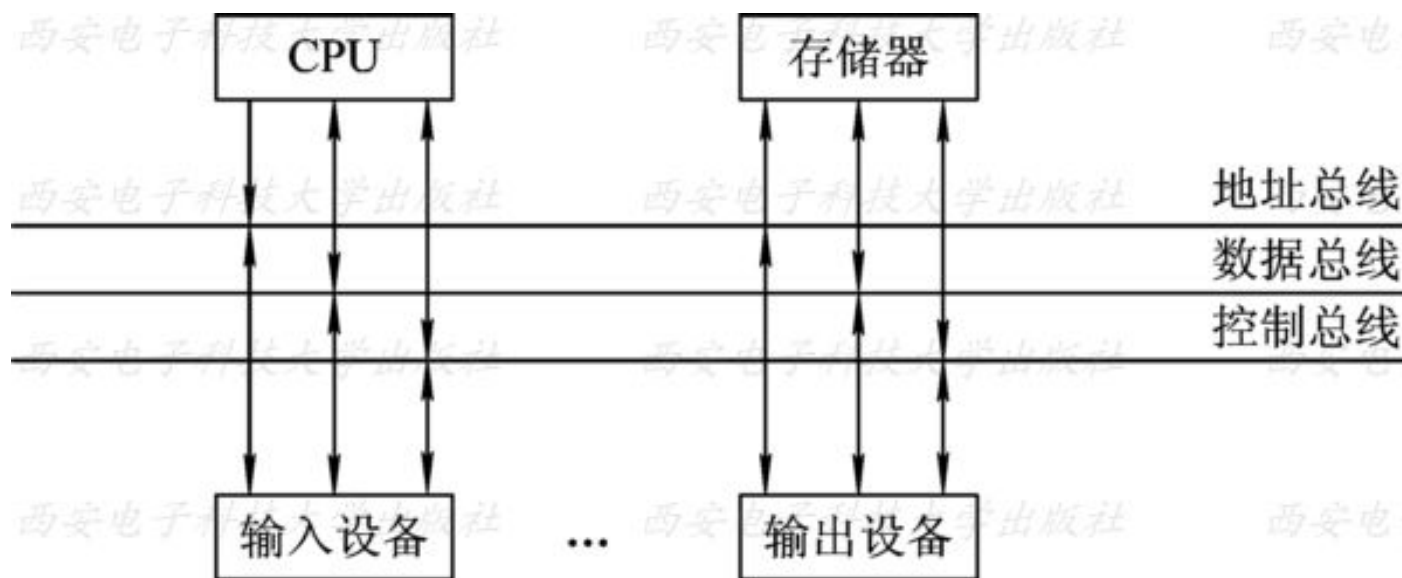


图1.5 以系统总线连接的计算机结构框图







## 1. 运算器

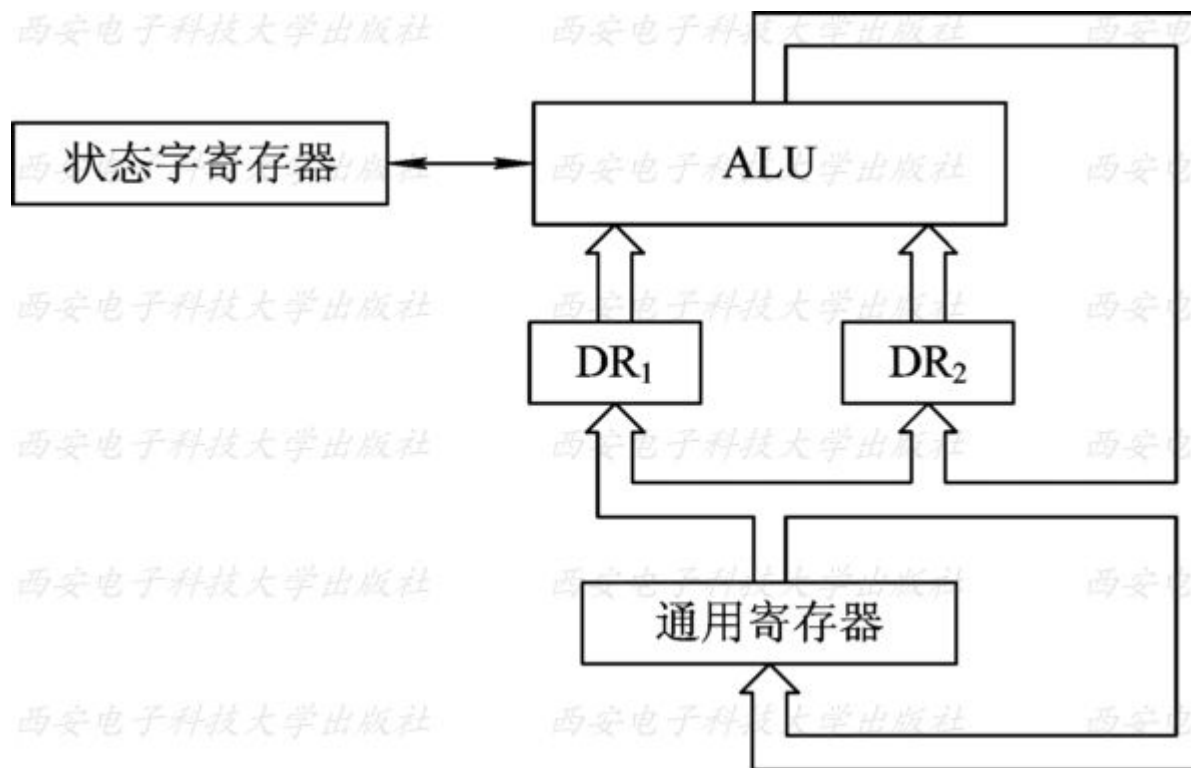


图1.6 运算器组成简图





## 2. 控制器

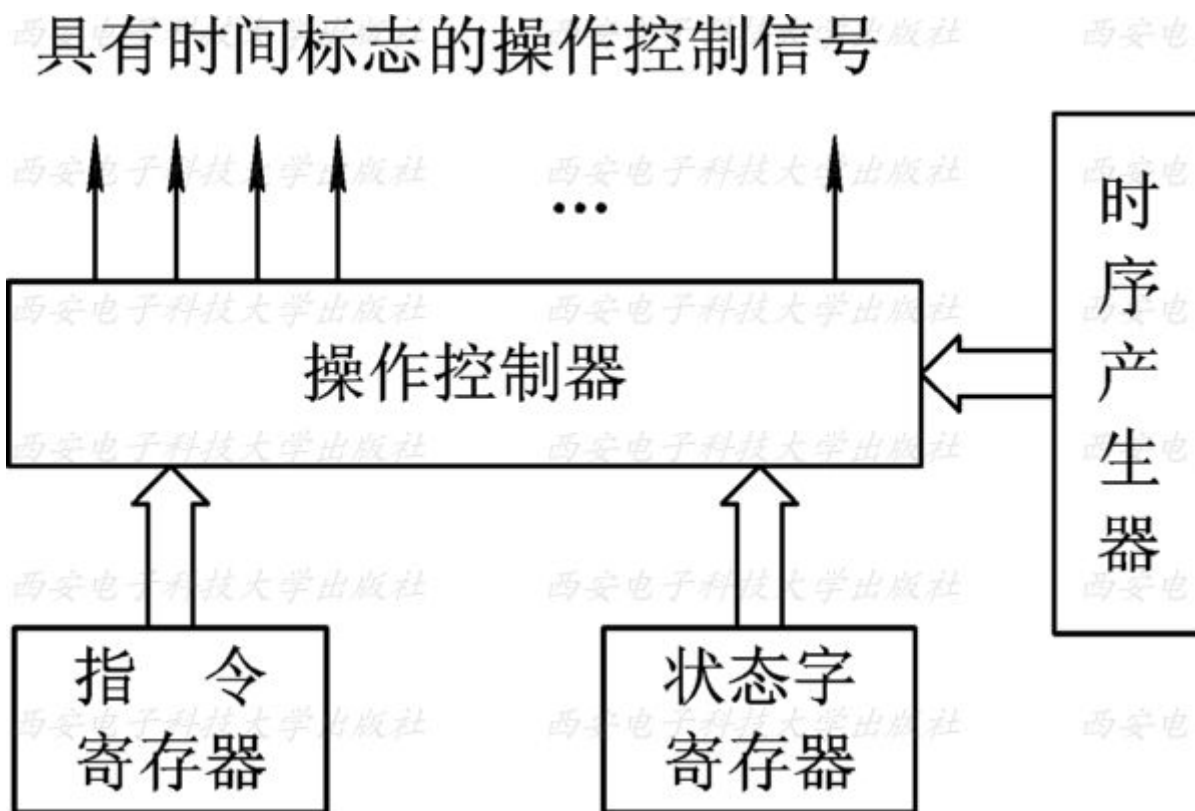


图1.7 控制器的组成简图





### 3、存储器

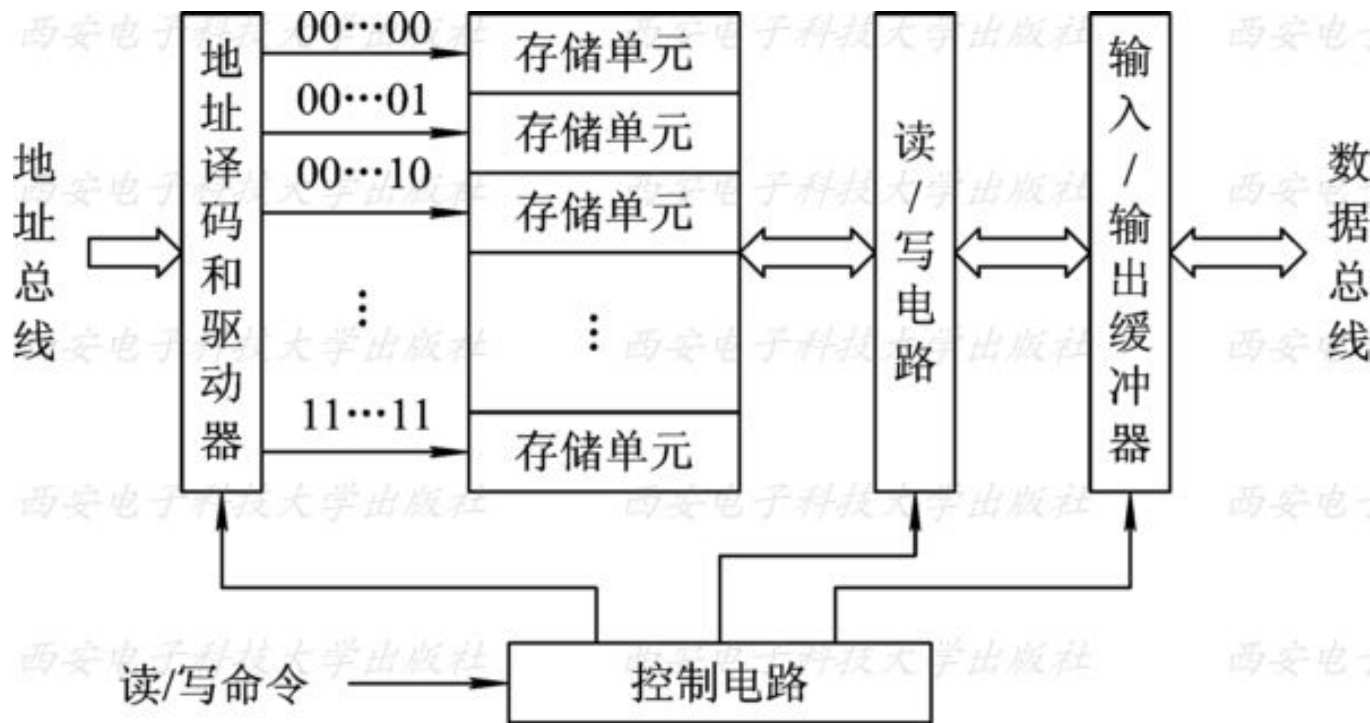


图1.8 主存储器的组成简图





## 1.3.2 计算机的软件系统 ■

按照前面国际标准化组织(ISO)对软件的定义，一般认为软件由程序与文档两部分组成。程序是为了实现一定功能而编写的指令序列，文档则是描述程序操作及使用的有关资料。

程序可由计算机执行，而文档则不能由计算机执行。程序是计算机软件的主体，故一般所说的软件都是指程序。 ■

一台计算机中全部程序的集合，统称为这台计算机的软件系统。计算机软件按功能的不同可分为**系统软件**和**应用软件**两大类。





### 1.3.3 计算机硬件与软件的逻辑等价性 ■

软件和硬件在逻辑功能上是等效的，同一逻辑功能既可以用软件也可以用硬件或固件实现。从原理上讲，软件实现的功能完全可以用硬件或固件完成，同样，硬件实现的逻辑功能也可以由软件的模拟来完成，只是性能、价格以及实现的难易程度不同而已。例如，在计算机中实现十进制乘法这一功能，既可以用硬件来实现，也可以用软件来完成。用硬件实现，需设计十进制乘法机器指令，用硬件电路来实现该指令，其特点是完成这一功能的速度快，但需要更多的硬件。而用软件来实现这个功能，则要采用加法、移位等指令通过编程来实现，其特点是实现的速度慢，但不需增加硬件。





## 1.4 计算机的工作过程 ■

### 1.4.1 使用计算机求解的一个简单例子 ■

采用现代计算机系统求解问题时，一般遵循如下步骤：首先由用户提出任务并建立数学模型，其次是要确定便于计算机实现的算法，然后是选择合适的语言编写程序，最后是上机调试运行。无论采用何种程序设计语言编写程序，都必须翻译成机器语言程序，即目标程序后才能在计算机上运行。机器语言程序是实现一定功能的机器指令(简称指令)序列。计算机的工作过程，就是执行指令的过程。





## 1. 一个简单例子 ■

[例1.1] 用计算机求解 $z=x+y$ ，其中 $x$ 和 $y$ 为已知数。

■ 解题步骤为： ■

① 将 $x$ 的值从主存单元取出，存入某一个寄存器； ■

② 将 $y$ 的值从主存单元取出，存入另一个寄存器； ■

③ 将 $x$ 和 $y$ 的值相加，运算结果存入寄存器； ■

④ 将结果从寄存器取出，存入主存单元 $z$ 中。 ■

将上述解题步骤按照计算机的指令格式和指令系统编写成对应的机器指令，就完成了程序的编写。





## 2. 指令与指令系统简介 ■

指令是要求计算机执行某种操作的命令，是计算机硬件能够直接识别和执行的二进制机器指令。指令系统是指一台计算机中所有机器指令的集合。一条指令通常由操作码和地址码两个字段组成，操作码指出了指令要执行的操作类型，即给出了指令的功能。地址码指出了与操作数相关的地址或操作数本身，即规定了指令操作的对象。在指令格式中，不同操作类型的指令其地址码的个数可以不相同，可以是零个到多个。







例1.1中计算机的指令格式如图1.9所示，每条指令的长度均为16位，指令的前4位为操作码，后12位为地址码，地址码的个数可以是0个、2个或3个，对应的指令分别称为零地址指令、二地址指令和三地址指令。假设该机指令系统中只包含有4条指令，如表1.1所示。



图1.9 三类指令的指令格式





表 1.1 例 1.1 中的指令系统及功能描述

操作码 (十六进制)	地址码 (十六进制)	功能描述
1	RXY	$(XY) \rightarrow R$ , 将主存地址 XY 单元中的数据取出, 存入寄存器 R 中
2	RXY	$(R) \rightarrow XY$ , 将寄存器 R 中的数据存入主存地址 XY 的单元中
3	RST	$(S) + (T) \rightarrow R$ , 将寄存器 S 与 T 中的数据相加, 结果存入寄存器 R 中
4	000	停机, 指令代码为 4000





### 3. 编写机器语言程序 ■

按照图1.9中的指令格式和表1.1中的指令系统，结合解题步骤，编写出一个计算 $z=x+y$ 的机器语言程序，如表1.2所示。设程序在主存空间的起始地址为0， $x$ 和 $y$ 的值已事先存入到地址为10、11(十六进制)的主存单元，运算结果 $z$ 要求存入到地址为12(十六进制)的主存单元。 ■

从表1.2可以看出，指令和数据都是以代码的形式存储于同一存储器的，计算机在执行时才加以区分，这是冯·诺依曼型计算机的基本原理。



# 第 1 章 计算机系统概论

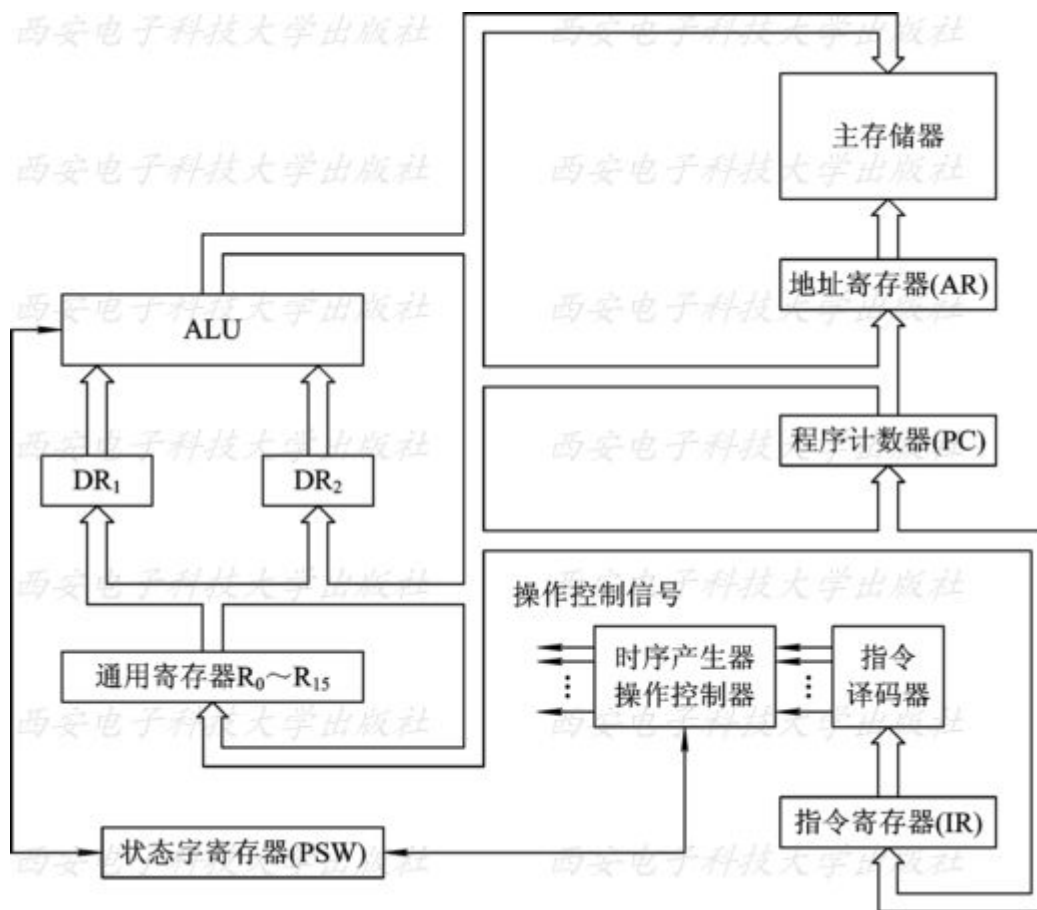


图1.10 一个简单计算机的组成框



表 1.2 计算  $z=x+y$  的机器语言程序

主存地址 (十六进制)	指令或数据 (十六进制)	注 释
0	1110	将数 $x$ 从主存单元取出, 存入寄存器 $R_1$
1	1211	将数 $y$ 从主存单元取出, 存入寄存器 $R_2$
2	3012	将寄存器 $R_1$ 与 $R_2$ 中的数据相加, 结果存入寄存器 $R_0$
3	2012	将寄存器 $R_0$ 中的数据存入主存单元 $z$ 中
4	4000	停机
⋮	⋮	⋮
10	0002	原始数据 $x$
11	0006	原始数据 $y$
12		存放结果 $z$
⋮	⋮	⋮





① 启动机器后，操作控制器发出控制信号，将程序计数器（PC）的内容(这里设置的初值为0)送至地址寄存器（AR），然后启动对主存的读操作，从主存的0地址中读出一条指令，并将指令代码“1110”(十六进制)送入指令寄存器（IR），同时PC的内容加1，形成下一条指令在主存中的地址。从而完成了第一条指令的取指操作过程。





② 通过指令译码器对指令寄存器(IR)中的操作码“1”(十六进制)进行分析, 识别出该指令的功能为取数指令, 于是操作控制器发出操作控制信号, 将指令寄存器(IR)中的地址码部分“10”(十六进制)送入地址寄存器(AR), 然后启动对主存的读操作, 从主存的10(十六进制)地址中读出x的内容0002送入通用寄存器R1。从而完成了第一条指令的分析过程和执行过程。





③ 重复①、②的操作过程，由于PC的值在每次取指操作完成时都进行了加1操作，因此计算机会自动地、顺序地逐条取出指令、分析指令和执行指令，直到执行完停机指令后，机器便自动停机。 ■

在指令执行过程中也有可能遇到改变程序计数器PC值的转移类指令，指令执行的具体过程与计算机的硬件结构有着密切的关系，但基本工作过程是相似的。在第5章我们还将结合各类指令、各种寻址方式和时序信号，对计算机的工作过程进行更深入的讨论。







练习:

- 1、计算机系统的五个组成部分是: \_\_\_\_\_
- 2、CPU包括哪些部分: \_\_\_\_\_
- 3、运算器包括: \_\_\_\_\_
- 4、控制器包括: \_\_\_\_\_
- 5、系统总线包括哪三类: \_\_\_\_\_
- 6、冯诺伊曼计算机的特点: \_\_\_\_\_





## 1.5 计算机性能

### 1.5.1 计算机性能的衡量尺度

#### 1. 响应时间

响应时间是指从用户向计算机系统发送一个请求后，到系统对该请求作出响应并获得它所需要的结果所花的等待时间。其中包括访问磁盘和访问主存储器的时间、CPU运算时间、输入/输出动作时间以及操作系统工作的时间开销等。





## 2. 吞吐率

吞吐率是指系统响应用户请求的速率。对于CPU而言，吞吐率可表示为**每秒钟执行的指令数**，或每秒钟执行的浮点操作次数。对于事务处理而言，吞吐率是指**单位时间内能处理的事务数**，对应的单位是TPS(Transactions Per Second)。在工作负载较低时，系统的吞吐率随工作负载的增加而线性增加。当工作负载增加到一定程度时，系统的吞吐率随工作负载的增加而达到一个峰值，这是因为系统的容量接近于饱和。吞吐率越高，计算机系统的处理能力就越强。





### 3. 可扩展性

如果一个计算机系统**能加以扩展**以满足不断增长的对性能和功能的要求，或是能够缩减资源以降低成本，则称此计算机系统具有可扩展性。可扩展性意味着一个扩展后的计算机系统能够提供更多的功能和更好的性能，**并且为扩展所花的代价必须合理，同时具有硬件和软件兼容性。**





#### 4. 可编程性、可靠性和可用性 ■

可编程性主要是指程序设计的方便性。可靠性是指一个计算机系统能**无故障运行的可靠程度**(指无故障运行的工作时间)。可用性是指一个计算机系统**可正常使用的时间占总时间的百分比**。通过大量冗余的处理器、存储器、磁盘、I/O设备、网络及操作系统映像等，可提高计算机系统的可用性。





## 5. 兼容性 ■

兼容性是指一个系统可以通过使用下一代的部件，如一个更快的处理器、一个更快的存储器、一个新版本的操作系统、一个更强功能的编译器等，进行扩展，以使计算机系统具有更强的功能和更好的性能。概括为一句话就是，一个系统应具有**向后兼容性**。通常所说的软件兼容性是指在一台机器上用其指令系统编写的软件可直接在其他机器上运行。为了保护用户在软件开发与投资上的利益，一个系统除了具有向后兼容性外，还应具有向上兼容性。在系列机中，应用软件、操作系统和编译器是兼容的，区别在于运行所需的时间不同。但兼容并不是绝对的、无条件的。





## 1.5.2 反映计算机性能的参数 ■

### 1. 主频 ■

通常所说的主频指的是**CPU的时钟频率**，单位通常为MHz( $1\text{ M}=10^6$ )、GHz( $1\text{ G}=10^9$ )。CPU时钟频率的倒数，称为**CPU时钟周期**，单位通常使用纳秒(ns)。CPU的时钟周期是计算机内部操作的基本时间单位，即计算机内部每一基本功能操作在一个或多个时钟周期的时间内完成。**主频是衡量一台计算机速度的重要参数**，主频越高，计算机的运行速度就越快。





## 2. 机器字长 ■

机器字长是指**运算器一次能运算的二进制数的最多位数**，它与**CPU内通用寄存器的位数**、**CPU内部数据总线的宽度有关**。机器字长越长，数的表示范围也越大，精度也越高。机器字长越长，表示计算能力越强。倘若CPU字长较短，又要运算位数较多的数据，那么需要经过两次或多次的运算才能完成，这样势必会影响整机的运行速度。通常所说的32位机、64位机等指的就是机器字长，它是计算机的重要性能指标。常见的机器字长有8位、16位、32位和64位。







### 3. 数据通路宽度 ■

数据通路宽度是指**数据总线**一次所能并行传送的数据的位数。它关系到信息的传送能力，从而影响计算机的有效处理速度。CPU**内部**的数据通路宽度一般等于机器字长，**而外部数据通路宽度**则取决于系统总线。





## 4. 运算速度 ■

计算机的运算速度与许多因素有关，如机器的主频、所执行的操作、主存本身的速度等。通常采用单位时间内执行指令的平均条数来衡量，单位为MIPS，即每秒执行百万条指令。也可用平均一条指令执行所花的时钟周期数来衡量，单位为CPI。还可用单位时间内执行浮点操作的平均数来衡量，单位为MFLOPS，即每秒执行百万次浮点操作。





## 5. 存储容量 ■

存储容量是指一个存储器中可以容纳的存储单元总数。

存储容量越大，所能存储的信息就越多。存储容量的单位有 B(字节)、KB( $1\text{ K}=2^{10}$ )、MB( $1\text{ M}=2^{20}$ )、GB( $1\text{ G}=2^{30}$ )、TB( $1\text{ T}=2^{40}$ )、PB( $1\text{ P}=2^{50}$ )、EB( $1\text{ E}=2^{60}$ )等。





### 1.5.3 性能因子CPI

设CPU的时钟周期为 $T_C$ ，CPU的时钟周期的倒数 $f_c$ 则是CPU的时钟频率。一个程序在CPU上运行所需的时间 $T_{CPU}$ 可以用下述公式表示：

$$T_{CPU} = I_N \times CPI \times T_C = \frac{I_N \times CPI}{f_c} \quad (1.1)$$

式中， $I_N$ 表示要执行程序中的指令总数(这里指动态执行指令数)，CPI(clock Cycles Per Instruction)表示执行每条指令所需的平均时钟周期数。





由此公式可见，程序运行的时间取决于三个特征：CPU的时钟周期、每条指令所需的时钟周期数以及程序中总的指令数。由CPI的含义可得到如下表达式：

$$\text{CPI} = \frac{\text{执行整个程序所需 CPU 时钟周期数}}{\text{程序中指令总数 } I_N} \quad (1.2)$$

在程序执行过程中，要用到不同类型的指令，令 $I_i$ 表示第 $i$ 类指令在程序中的执行次数， $\text{CPI}_i$ 表示执行一条第 $i$ 类指令所需的时钟周期数， $n$ 为程序中所有的指令种类数。则式(1.2)可以改写为





$$\text{CPI} = \frac{\sum_{i=1}^n \text{CPI}_i \times I_i}{I_N} \quad (1.3)$$

因为 $I_N$ 是一个常数，所以式(1.3)可以改写为

$$\text{CPI} = \sum_{i=1}^n \left( \text{CPI}_i \times \frac{I_i}{I_N} \right) \quad (1.4)$$

其中， $I_i/I_N$ 表示第*i*类指令在程序中所占的比例。式(1.4)说明平均CPI或称有效CPI等于每类指令的CPI和该类指令在整个程序中出现的百分比的乘积之和。





## 1.5.4 计算机性能常用指标 ■

1. MIPS(Million Instructions Per Second, 每秒百万条指令)  
这是一个用来描述计算机性能的尺度。对于一个给定的程序, MIPS定义为

$$\text{MIPS} = \frac{I_N}{T_E \times 10^6} = \frac{I_N}{I_N \times \text{CPI} \times T_C \times 10^6} = \frac{f_C}{\text{CPI} \times 10^6} \quad (1.5)$$

在式(1.5)中假定 $T_E = T_{\text{CPU}}$ , 即计算机的指令执行时间就是CPU的执行时间;  $f_C$ 表示CPU的时钟频率, 它是CPU时钟周期 $T_C$ 的倒数。





由式(1.5)可得程序的执行时间 $T_E$ 的表达式为

$$T_E = \frac{I_N}{\text{MIPS} \times 10^6} \quad (1.6)$$

[例1.2] 已知Pentium II 450处理机在运行某一测试程序时所获得的性能为0.5 CPI，试计算Pentium II 450处理机在运行该程序时所获得的MIPS速率。 ■

解： 由于Pentium II 450处理机的 $f_C=450$  MHz， 因此，  
由式(1.5)可求出：







$$\text{MIPS}_{\text{PentiumII450}} = \frac{f_C}{\text{CPI} \times 10^6} = \frac{450 \times 10^6}{0.5 \times 10^6} = 900 \text{MIPS}$$

即Pentium II 450处理机在运行该程序时所获得的MIPS速率为900 MIPS。 ■

在使用MIPS作为性能指标时，存在以下不足之处：① MIPS依赖于指令系统，因此用MIPS来比较指令系统不同的机器的性能是不准确的；② 在同一台机器上，MIPS因程序的不同而变化，有时会相差很大；③ 它只适宜于评估标量机，因为在标量机中执行一条指令，一般可得到一个运算结果，而在向量机中，执行一条向量指令通常可得到多个运算结果。因此，用MIPS来衡量向量机是不合适的。





## 2. MFLOPS(Million Floating point Operations Per Second, 每秒百万次浮点运算) ■

MFLOPS可用如下公式表示:

$$\text{MFLOPS} = \frac{I_{\text{FN}}}{T_{\text{E}} \times 10^6} \quad (1.7)$$

其中,  $I_{\text{FN}}$ 表示程序中的浮点运算次数。 ■

由于MFLOPS取决于机器和程序两个方面, 因此MFLOPS只能用来衡量机器浮点操作的性能, 而不能体现机器的整体性能。例如对于编译程序, 不管机器的性能有多好, 它的MFLOPS都不会太高。





由于MFLOPS是基于操作而非指令的，因此它可以用来比较两种不同的机器。这是因为同一程序在不同的机器上执行的指令可能不同，但是执行的浮点运算却是完全相同的。采用MFLOPS作为衡量单位时，应注意它的值不但会随整数、浮点数操作混合比例的不同发生变化，而且也会随快速和慢速浮点操作混合比例的变化而变化。例如，运行由100%浮点加组成的程序所得到的MFLOPS值将比由100%浮点除法组成的程序要高。





[例1.3] 用一台40 MHz处理机执行标准测试程序，程序所含的混合指令数和每类指令的CPI如表1.3所示，求有效CPI、MIPS速率和程序的执行时间。

表 1.3 标准测试程序的混合指令数和每类指令的 CPI

指令类型	整数运算	数据传送	浮点操作	控制传送
指令数	45 000	32 000	15 000	8000
CPI	1	2	2	2

解：总的指令数为 ■

$$45\ 000 + 32\ 000 + 15\ 000 + 8000 = 100\ 000\text{条}$$





因此各类指令所占的比例分别是：整数运算为45%，数据传送为32%，浮点操作为15%，控制传送为8%。

有效CPI、MIPS速率和程序的执行时间分别计算如下：

■ (1) 有效CPI为 ■

$$1 \times 0.45 + 2 \times 0.32 + 2 \times 0.15 + 2 \times 0.08 = 1.55\text{CPI} \quad \blacksquare$$

(2) MIPS速率为 ■

$$40 \times 10^6 / (1.55 \times 10^6) \approx 25.8 \text{ MIPS} \quad \blacksquare$$

(3) 程序的执行时间为 ■

$$100\,000 \times 1.55 / (40 \times 10^6) = 0.003\,875 \text{ s}$$





# 习 题 1 ■

1、6、14

