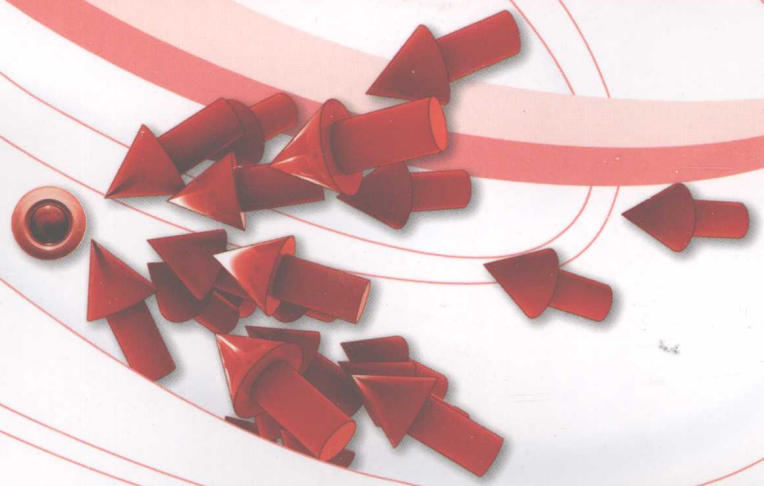


高等学校计算机规划教材

计算机组成原理

任国林　编著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

计算机组成原理

任国林 编 著

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书系统地介绍了冯·诺依曼模型单处理器计算机系统中各部件的组成结构和工作原理。全书共分7章，内容包括：计算机系统概述、数据的表示与运算、存储系统、指令系统、中央处理器、总线及I/O系统，涵盖了全国硕士研究生入学统一考试计算机科学与技术联考大纲的全部内容。本书突出整机概念，注重知识点融合，强调部件工作原理与硬件组织相结合，通过性能分析与优化设计来加深对基本原理的理解与掌握。

本书既适合讲授、又适合自学，可作为高等院校计算机及相关专业学生的教材，或计算机专业研究生入学统考的复习用书，也可作为计算机领域科技人员的参考书。

图书在版编目（CIP）数据

计算机组成原理 / 任国林编著.—北京：电子工业出版社，2010.2
ISBN 978-7-121-10288-2

Ⅰ. 计… Ⅱ.任… Ⅲ.计算机体系结构－研究生－入学考试－自学参考资料 Ⅳ.TP303

中国版本图书馆 CIP 数据核字（2010）第 015363 号

策划编辑：董亚峰
责任编辑：毕军志　　文字编辑：裴　杰
印　　刷：北京市李史山胶印厂
装　　订：
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编　100036
开　　本：787 × 1092　1/16　印张：19　字数：488 千字
印　　次：2010 年 2 月第 1 次印刷
印　　数：4000 册　　定价：28.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。
质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。
服务热线：（010）88258888。

前　言

随着计算机应用的不断普及，各高等学校大多设置了计算机相关专业，而电类专业更是大多开设了计算机硬件课程。“计算机组成原理”是一门很重要的计算机硬件基础课程，对深入了解和掌握计算机系统相关技术至关重要。“计算机组成原理”作为全国硕士研究生入学统一考试计算机科学与技术联考课程，可见其重要性。

目前，国内“计算机组成原理”的教材很多，在使用过程中，普遍感到内容不够新颖，知识点融合不够，讲法五花八门，有些教材甚至连概念都存在问题。为了适应计算机科学与技术专业的教学需要，同时为参加全国硕士研究生计算机专业统考的考生提供一本好的参考教材，笔者根据多年的教学实践，结合全国硕士研究生入学统一考试计算机科学与技术联考大纲的要求，编写了本书。

全书共分 7 章，系统介绍了冯·诺依曼模型单处理器计算机系统的组成结构及工作原理。第 1 章为计算机系统概论，简要介绍了冯·诺依曼模型计算机系统的基本组成及其性能指标。第 2 章为数据的表示与运算，主要分析了各种数据在计算机中的表示方法，及相应运算在计算机中的实现方法。第 3 章为存储系统，分别讨论了层次结构存储系统中主存及 Cache 的组成原理及优化方法，介绍了虚拟存储器的组成工作过程。第 4 章为指令系统，重点介绍了指令功能与格式的约定方式，以及数据存放及寻址方式。第 5 章为中央处理器，在分析 CPU 功能的实现需求基础上，讨论了控制器满足这些实现需求的组织与设计方法，并介绍了流水线的工作原理。第 6 章为总线，重点介绍了总线的传输与控制原理及总线结构。第 7 章为输入/输出系统，在介绍 I/O 设备组成原理及与主机连接方法的基础上，重点讨论了几种传送控制方式的原理，分析了相关软、硬件的组织方法。

本书具有如下特点：

（1）突出计算机组成的一般原理，采用自顶而下的方法，强调系统性，注意知识点融合，容易形成计算机的整机概念。

（2）强调计算机工作原理与硬件组织方法的相互结合，使抽象原理实物化，有利于对基本原理的理解，有利于知识点融合。

（3）重视部件组成的性能分析及优化，大量的量化分析可加深对基本原理的掌握，典型的优化设计更便于增加对计算机最新技术的了解，进一步增加学习兴趣。

（4）本书内容涵盖了全国硕士研究生入学统一考试计算机科学与技术联考大纲的全部内容，是一本理想的考研参考教材。

本书在编写过程中力求语言通俗易懂，文字简洁明了，便于读者自学。东南大学计算

机科学与工程学院的朱怡健、徐造林、杨全胜、王晓蔚副教授对本书的编写提出了许多宝贵意见，在此表示诚挚的谢意。

作者从事“计算机组成原理”、“微机原理与接口技术”、“计算机系统结构”课程教学工作多年，在本书编写过程中尽可能进行了一些探索，力求从计算机系统的角度，关联本课程与相关课程的相关概念和原理，反映计算机硬件的新技术。由于编写时间紧迫及水平有限，书中难免有疏漏和欠缺之处，敬请广大读者和同行专家批评指正。

作　者

2009 年 11 月

目　录

第1章　计算机系统概论

1.1　计算机系统简介

电子计算机是一种以电子方式计算的机器，有电子模拟计算机和电子数字计算机两种类型。电子模拟计算机采用模拟装置进行计算，数值用连续量表示、计算过程是连续的，如早期的电表等，电子模拟计算机的运算精度及运算能力均有限。

电子数字计算机模仿算盘计算方式，数值用多位数字表示、采用按位及跳动式计算，多位运算可提高运算精度，跳动式运算便于实现逻辑判断功能，进而实现模拟人类“思维过程”功能，故常被称为电脑，并逐渐独占了电子计算机、计算机这两个名称。

计算机的基本功能包括数据处理、数据存储、数据传送、过程控制4个方面。

（1）数据处理：指计算机应具有算术运算、逻辑运算和关系运算3个方面的计算能力，除处理数值数据外，还应能够处理逻辑数据、文字、符号、视频和音频等多种非数值数据。

（2）数据存储：指计算机应具有能够长期保存数据的能力，以便于实现数据的多次应用，如数据检索和更新等，数据通常以文件形式存放在存储器中。

（3）数据传送：指计算机应具有内部部件间数据传递，与外部设备间的输入/输出功能。

（4）过程控制：指计算机应具有根据逻辑判断结果，控制所有功能实现顺序的功能，否则计算机无法按照人类的预期设想来工作。

1.1.1　计算机系统的软、硬件

众所周知，计算机系统由“硬件”和“软件”两大部分组成。

“硬件”是指计算机系统中看得见，摸得着的实体部分，一般有输入、输出、存储、处理等部件或设备，均由电子元器件、光/机/电设备等组成。不同部件可以实现不同的操作或处理功能，但功能是否实现、何时实现取决于给硬件所发出的控制信号是否有效及何时有效。

“软件”是指计算机系统中看不见摸不着的、由人们预先编制的、实现各类特定处理功能的程序，程序由有序的指令串组成，表示了处理步骤及各操作步骤的需求。为便于实现程序的多次使用，通常将程序存放在存储器中，使用时从存储器中取出即可。

现代计算机系统的解题过程通常有编辑、编译、运行3个步骤，如图1.1所示。编辑时形成高级语言程序（称为源程序），编译时形成机器语言程序（称为目标程序），运行时执行机器语言程序可得到具体的解题结果。

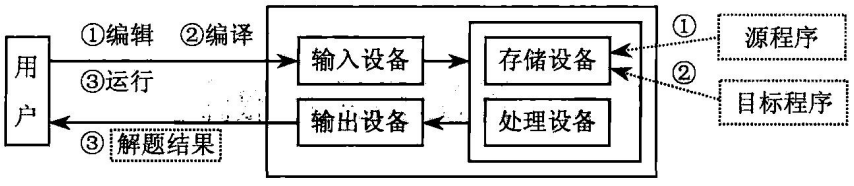


图 1.1 计算机解题过程示意图

从图 1.1 中可以看出，计算机硬件具备数据处理、数据存储、数据传送、过程控制功能，所用器件及其组织方法影响硬件的性能；计算机软件表示具体应用在数据处理、数据存储、数据传送及其过程控制方面的功能需求，所用算法及编程技巧影响软件的性能；只有在硬件上执行相应软件，才可以实现具体应用所需的功能。

因此，一方面，软件以硬件为平台，即软件的功能依靠硬件实现；另一方面，硬件以软件为窗口，即硬件的性能依靠执行软件反映。所以说，计算机系统的性能为软件性能和硬件性能的总和，或计算机系统性能是计算机软件和计算机硬件相互作用的结果。

1.1.2 计算机系统的层次结构

既然计算机系统是由软件和硬件结合而成的整体，那么如何在计算机上进行编程呢？从程序员的角度看，计算机系统有多个层次结构，在每个层次上都可以进行程序设计。计算机系统层次结构的发展，反映了计算机系统的发展历程。

1. 早期计算机系统的层次结构

早期的计算机中，程序员只可以使用机器语言编写程序，机器语言的指令格式为二进制（0 或 1）代码，每种指令可以实现简单的操作功能，如数据传送、加法运算等。之所以称这种语言为机器语言，是因为指令可直接被硬件所识别，指令功能可直接在机器硬件上实现，即具体指令由硬件解释并直接控制相应数字电路实现其功能。

通常，将用机器语言编写的程序称为机器语言程序，将可使用机器语言编写程序并执行程序的机器称为机器语言级机器。将硬件直接识别和执行程序的机器称为实际机器，因此，机器语言级机器是一种实际机器，这种计算机系统的层次结构如图 1.2 所示。

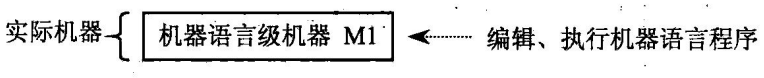


图 1.2 早期的一个层次结构的计算机系统

机器语言级机器编程很不方便，20 世纪 50 年代首次出现了符号式程序设计语言，即汇编语言，它用符号 ADD、SUB 等表示加、减等指令操作，也可用符号表示操作数或指令的地址，程序员可不再使用繁杂且易错的二进制代码编写程序，因此立即得到广泛的应用。

使用汇编语言编写的程序称为汇编语言程序。实际上，这种汇编语言程序不可以被任何硬件识别和直接执行，必须先将汇编语言程序翻译成机器语言程序，然后才能被机器语言级机器所识别和执行。这个翻译过程是由一种称为汇编程序的软件完成的。

通常，将使用汇编语言编程的机器称为汇编语言级机器。汇编语言级机器是一种虚拟

机器。虚拟机器是指那些只可编写程序和翻译程序，但不能直接在硬件上运行程序的机器。因为汇编语言级机器上的汇编语言程序必须通过汇编程序翻译成机器语言程序，方可在机器语言级机器上运行。所以，从层次结构上来看，汇编语言级机器应该在机器语言级机器之上，其层次结构如图 1.3 所示。

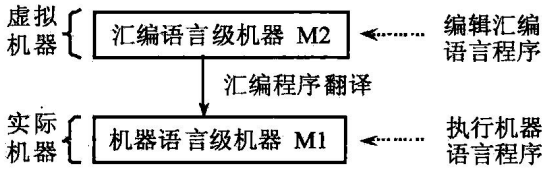


图 1.3　早期的两个层次结构的计算机系统

尽管汇编语言方便了编程，但本质上汇编语言是一种面向实际机器的程序设计语言，它要求程序员对硬件组成及指令系统比较熟悉、且通用性不好（不同机器的机器语言不同），不利于计算机的广泛应用及发展。

2. 现代计算机系统的层次结构

20 世纪 60 年代先后出现了多种面向问题的高级语言，如 BASIC、FORTRAN、PASCAL、C 等。这类语言程序描述问题的效率很高、通用性较好，程序员不需要了解实际机器的硬件组成及指令系统，只要掌握语言本身的语法和语义，就可以直接使用高级语言进行编程，给程序员带来了极大的方便。同样，高级语言程序不能被实际机器所识别和运行，必须翻译成汇编语言程序（或中间语言程序）后、再翻译成机器语言程序，或直接翻译成机器语言程序，然后才可在机器语言级机器上运行。通常，将使用高级语言编程的机器称为高级语言级机器。自然，高级语言级机器也是一种虚拟机器，其层次结构如图 1.4 所示。

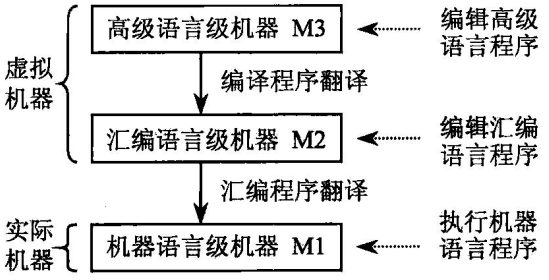


图 1.4　3 个层次结构的计算机系统

实现程序翻译功能的软件称为翻译程序。翻译方法有解释和编译两种类型。

解释是将源程序的一条语句翻译成机器语言的一条或几条指令，并且立即执行这一条指令或逐条执行这几条指令，接着再翻译并执行源程序的下一条语句，直到完成源程序的全部翻译任务，如早期的 BASIC 语言就是用解释方法完成翻译和执行的。

编译是将源程序的全部语句一次性翻译成机器语言程序（或中间语言程序），在需要时再执行机器语言程序（或解释执行中间语言程序），如 FORTRAN、C 等语言就是用编译方法实现翻译的。编译方法的优点是：只要源程序不改变，执行时就无须再次翻译。

随着应用的不断普及，用户对系统性能的要求及系统好用性的要求也不断提高。

为解决系统好用性问题，现代计算机中均通过操作系统程序负责所有软件及硬件资源的管理和控制。操作系统程序由机器语言程序和一组广义指令组成，广义指令是由操作系统定义的、供用户或软件使用的、具有特定管理与控制功能的命令，广义指令同样通过解释执行相应机器语言程序来实现其功能。操作系统程序实际上在机器语言级机器和汇编语言级机器间形成了操作系统级机器。同样，操作系统级机器也是一种虚拟机器。

为解决系统性能问题，现代计算机均不断地扩充硬件功能以提高软件性能，使得机器语言级机器的指令系统越来越复杂。为了减少实现复杂性，较好的方法是将一条机器语言指令翻译成一组微指令（即一个微程序）、硬件直接执行微程序中各条微指令以实现机器语言指令功能，由此在机器语言级机器之下产生了一个新的实际机器层次，即微程序级机器。微程序级机器的核心思想是用微程序解释机器语言指令，由于微程序级机器也是实际机器，为区别起见，常将机器语言级机器称为传统级机器。现代计算机系统的层次结构如图 1.5 所示。

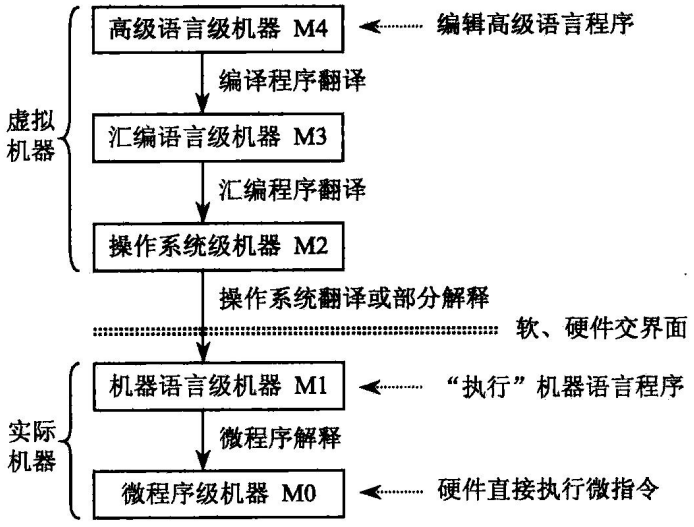


图 1.5　现代计算机系统的层次结构

随着计算机系统应用领域的扩大，虚拟机器还可以进一步向上延伸，形成如应用语言级机器等虚拟机器。

从计算机系统的多层次结构来看，硬件研究的主要对象为传统级机器、微程序级机器，软件研究的对象是操作系统级及以上的各级虚拟机器。而软件与硬件交界面在层次结构中的位置是由计算机系统的设计者所决定的，向上移动可提高性能、但增加了成本，向下移动可减少成本、但降低了性能。随着硬件性价比的提高，目前软、硬件交界面正向软件方向发展。

1.1.3　计算机结构与组成

从如图 1.5 所示的层次结构可以看出，与硬件有关的研究包括软、硬件交界面的确定，传统级机器及微程序级机器的组成两个方面。实际上，硬件的研究还包含机器语言指令或微指令功能的数字电路实现方面。应注意区分相应的计算机系统结构、计算机组成、计算机实现这 3 个基本概念。

计算机系统结构是指程序员所看到的计算机系统的属性，包括概念性结构和功能特性两个方面。精确地讲，计算机系统结构是指机器语言程序员或编译程序编写者所看到的传统级机器的属性，包括指令集、数据表示、寻址方式、存储系统组织、I/O 系统组织等抽象属性或参数，主要研究计算机系统软、硬件交界面的定义及其上下的功能分配。

计算机组成是指计算机硬件设计人员所看到的系统属性，它包含了对许多层次程序员来说是透明的硬件属性，包括专用部件及数据通路宽度设计、功能部件并行度、控制机构组成方式等。主要研究如何合理地逻辑实现计算机系统结构分配给硬件的功能。

计算机实现则主要研究器件和微组装等技术，即研究如何高效地物理实现计算机组成所要求的功能。

例如，一台传统级机器是否具有乘法指令是计算机系统结构研究的内容，乘法功能是用乘法器实现还是用“加法器＋移位器＋循环”实现是计算机组成研究的内容，具体的器件设计则是计算机实现研究的内容。

这 3 个概念反映硬件方面 3 个层次的内容，任何时候都要注意区分，因为同一种系统结构可以有多种不同的组成方式，同一种组成方式可以有多种不同的实现方式，而不同的组成或实现方式意味着机器具有不同的性能或性价比。

计算机实现的内容在“数字电路”课程中已经学习过，本课程不再赘述。本课程主要研究计算机组成，即传统级机器和微程序级机器的组成原理和设计思想。需要注意的是，计算机系统结构负责确定软、硬件交界面结构特性及参数，计算机组成的研究是在计算机系统结构的基础上进行的，即计算机组成是计算机系统结构的逻辑实现。有关计算机系统结构的内容将在“计算机系统结构”课程中讲述。

1.2　计算机系统基本组成

1.2.1　冯·诺依曼模型计算机

早期的计算机存储容量很小、程序及数据的输入/输出很不方便。1946 年，数学家冯·诺依曼提出了以存储程序原理为核心的计算机模型，该计算机模型一直沿用至今。通常称该计算机模型为冯·诺依曼模型，采用该思想设计的计算机为冯·诺依曼模型计算机。

计算机模型主要包括结构模型和工作模型两个方面。

在结构模型方面，冯·诺依曼模型计算机由运算器、控制器、存储器、输入设备和输出设备组成。其中输入/输出设备负责指令和数据的输入/输出；存储器负责存储指令和数据；运算器负责处理数据；控制器负责指挥和控制各部件协调工作，以实现程序期望的功能，其结构如图 1.6 所示。

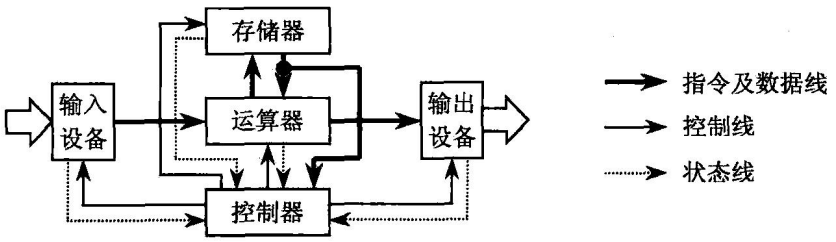


图 1.6　冯·诺依曼模型计算机结构框图

在工作模型方面，冯·诺依曼模型计算机采用的是被称为“存储程序”的工作原理。存储程序原理是指程序和数据预先存放在存储器中，机器工作时自动按程序的逻辑顺序从存储器中逐条取出指令并执行。

如何理解存储程序原理呢？

首先，我们来看程序存储方式。由于存储器是可以连续被访问的（只要连续发出命令即可），因此程序和数据存放在存储器中为计算机连续、自动工作奠定了基础。

由于程序由有序的指令组成，这就要求存储器为按地址访问的存储空间；为了提高存储器的使用效率，指令和数据应可以混合存放在同一存储器中，因此存储器地址为一维地址即可，并且存储单元长度应该是指令和数据长度的最小者的长度。由此可以看出，存储器结构应是由定长单元组成的、按地址访问的、一维线性空间结构。

通常，我们将程序在存储器中的指令存放顺序称为程序的存储顺序，而将程序希望根据逻辑判断结果实现的指令执行顺序称为程序的逻辑顺序。由于存储器是一维线性空间，因此指令类型有顺序型和转移型两种，转移型指令用来实现改变程序逻辑顺序的功能。

其次，我们来看程序控制机制。程序控制的根本目标是按程序的逻辑顺序、自动执行所有指令。

从自动执行所有指令角度看，程序执行过程是循环的指令执行过程；从按程序的逻辑顺序执行指令角度看，由于程序的存储顺序与逻辑顺序可能不同，因此程序中指令只能逐条执行，并且程序执行过程的下条指令地址由当前指令的结果产生。

表 1.1 的 C 语言程序可以帮助我们理解冯·诺依曼模型的程序控制机制，程序中指令按顺序存放，指令按程序逻辑顺序（非存储顺序）逐条执行。

表 1.1　$S=1+2+\cdots+n$ 的 C 语言程序的逻辑顺序

程序的存储顺序		程序的逻辑顺序	
指令地址	指令内容	程序逻辑顺序	逻辑顺序中下条指令地址的形成方法
A+0	int nCount=0;	(1)	=当前指令地址+1
A+1	int nSum=0;	(2)	=当前指令地址+1
A+2	LP: nSum+=nCount;	(3)(6)(9)(12)	=当前指令地址+1
A+3	nCount++;	(4)(7)(10)(13)	=当前指令地址+1
A+4	if (nCount<5) goto LP;	(5)(8)(11)(14)	条件为真时，=A+2（当前指令指定） 条件为假时，=当前指令地址+1
A+5	Printf (“%d”, nSum);	(15)	=当前指令地址+1

由于指令是存放在存储器中的，因此，指令执行过程有取指令、执行指令两个阶段。取指令是指用当前指令地址从存储器中读出一条指令，并分析所取指令包含的信息；执行指令指根据所取指令包含的信息执行相应的操作功能。

由于程序执行过程是循环的指令执行过程，因此，循环变量为指令地址，理论上的每轮循环有取指令、执行指令、改变指令地址 3 个阶段的动作，如图 1.7 所示。对顺序型指令而言，（下条）指令地址=（当前）指令地址+1，对转移型指令而言，（下条）指令地址=（当前）指令中信息的操作结果。

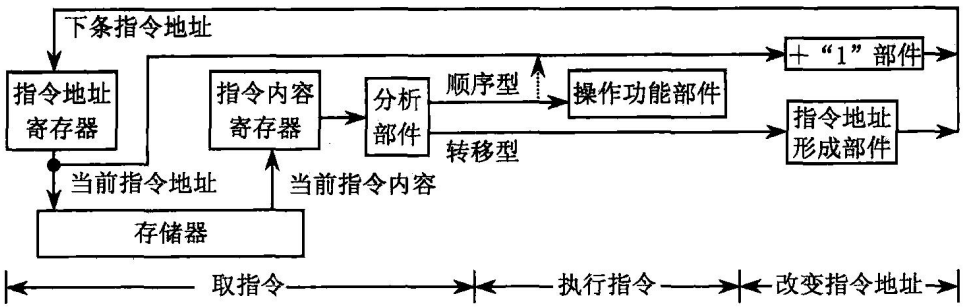


图 1.7　冯・诺依曼模型计算机的程序执行过程示意图

综上所述，冯・诺依曼模型计算机的特点可归纳为如下几点：

（1）计算机由运算器、控制器、存储器、输入设备和输出设备组成；

（2）存储器是由定长单元组成的、按地址访问的、一维线性空间结构；

（3）程序由指令组成，指令和数据以等同地位存放在存储器中；

（4）机器工作时自动按程序的逻辑顺序从存储器中逐条取出指令并执行；

（5）指令由操作码和地址码组成，操作码用于表示操作的性质，地址码用于表示操作数在存储器中的地址；

（6）指令和数据均采用二进制方式表示，运算也采用二进制方式；

（7）机器以运算器为中心，输入/输出设备与存储器间的数据传送都经过运算器。

冯・诺依曼模型一直沿用至今，因此，深入理解冯・诺依曼模型思想是基础中的基础，对计算机组成原理的学习至关重要。

1.2.2　计算机硬件的基本组成

1. 现代计算机硬件结构

冯・诺依曼模型的 5 大部件中，运算器主要完成算术和逻辑运算，其核心部件为算术逻辑单元（Arithmetic Logic Unit，ALU）；控制器主要控制各部件工作以完成程序指令过程，其核心部件为控制单元（Control Unit，CU）；运算器和控制器在逻辑关系和电路结构上联系均十分密切，常集成在同一芯片上以提高性能。因此，通常将它们合称为中央处理器（Central Processing Unit，CPU），又称 CPU 由 ALU 及 CU 组成。而输入设备与输出设备功能相近，通常合称为 I/O 设备。

现代计算机结构大多采用冯・诺依曼模型思想，但逐步在其基础上进行改进，以提高系统的性能。改进主要包括以存储器为中心、多种存储器共存、采用总线互连 3 个方面。

（1）以存储器为中心的结构

冯・诺依曼模型计算机以运算器为中心，即输入及输出的所有数据均通过运算器中转。由于指令和数据均存放在存储器中，为了实现数据传送与数据处理可以并行执行，现代计算机全部采用以存储器为中心的硬件结构来提高性能，如图 1.8 所示。

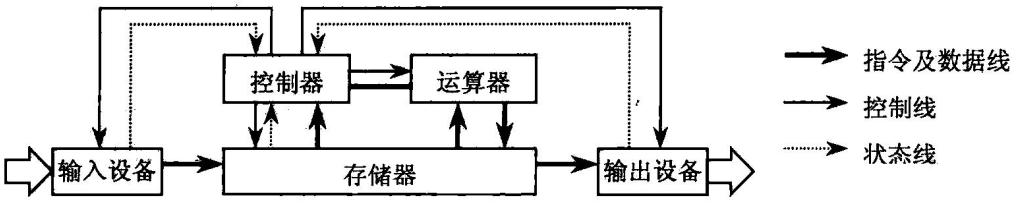


图 1.8　现代计算机以存储器为中心的结构

如何才能够使数据传送与数据处理并行呢？缓冲技术和 DMA 技术是较好的解决方案。

缓冲技术的思想是：CPU 先从存储器中取一部分指令和数据到特定的缓冲区中，CPU 在处理缓冲区中数据的同时，I/O 设备与存储器间可以进行数据传送，只有当 CPU 所需数据不在缓冲区中时，数据处理与数据传送才不能并行执行。缓冲技术与下面的“多种存储器共存”方法异曲同工，为数据处理与数据传送并行执行提供了可行性。

DMA（Direct Memory Access）技术的思想是：在 CPU 进行数据处理的同时，由专用部件控制 I/O 设备与存储器间的数据传送、而不需要 CPU 干预，技术细节在第 7 章中进行讨论。

（2）多种存储器共存的结构

由于计算机软件对存储器容量的需求很大，用户又不希望计算机成本过高，所以，实际的存储器速度相对 CPU 而言很慢。

为了解决存储器速度与容量的矛盾，现代计算机一般用主存储器及辅助存储器组成存储系统，来代替冯·诺依曼模型中的存储器。辅助存储器又称外存或辅存，如硬盘、磁带等，其速度慢、成本低，存放所有的程序和数据；主存储器又称内存，其速度快、成本高，存放部分程序和数据。CPU 直接访问主存可以提高访存速度，主存容量较小时成本增加有限。现代计算机中，CPU 和主存储器合起来常称为主机，而将输入/输出设备及辅存统称为 I/O 设备，其结构如图 1.9 所示。

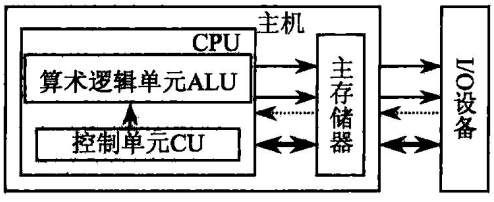


图 1.9　现代计算机多种存储器共存的结构

（3）采用总线互连的结构

由于计算机的应用越来越广泛，不同应用所需 I/O 设备的种类和数量不同，为减少硬件实现的复杂性、提高系统的可扩展性，通常采用总线方式实现各部件的互连，总线上各部件采用统一的协议实现交互。

由于各种 I/O 设备的速度、对外接口均有所不同，为实现总线操作的标准化，则需要在 I/O 设备与总线间增加被称为 I/O 接口或适配器或控制器的电路。

采用总线互连的计算机硬件结构如图 1.10 所示。为了解决速度差异很大的 I/O 设备的共存问题，现代计算机常采用多总线结构，总线间通过称为“桥”的部件进行互连。

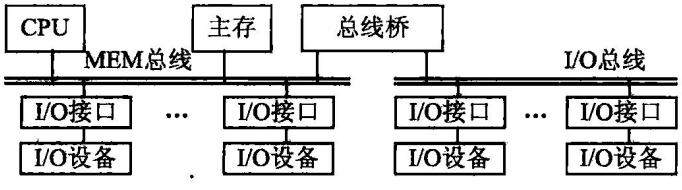


图 1.10　现代计算机采用总线互连的结构

3 种改进方法的详细技术细节在后续的章节中进行讨论。

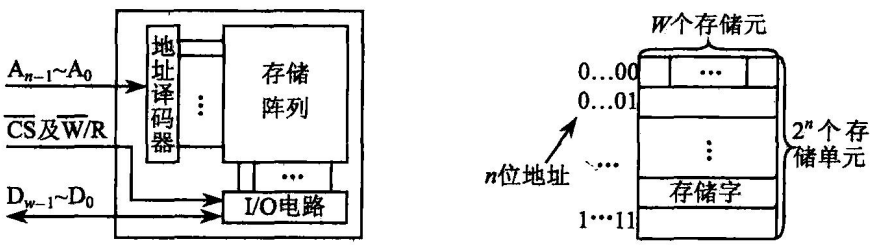
2. 计算机部件基本组成

下面将简要介绍 5 大部件的组成、工作方式，注意理解相关术语及基本概念。

（1）存储器

存储器的主要功能是实现信息存储，可通过操作的形式实现信息的写入和读出。

根据冯·诺依曼模型计算机的要求，存储器是由定长存储单元组成的、按地址访问的、一维线性空间结构，从“数字电路”课程所学的知识可知，存储器主要由存储阵列、地址译码器、I/O 电路等组成，其基本结构如图 1.11（a）所示。其中能够存放一个二进制信息（0 或 1）的元件称为存储元；能够同时存放一次存储器操作所有信息的元件称为存储单元，存储单元由多个存储元（假设为 w 个）组成；存储阵列由多个存储单元（假设为 2^n 个）组成。根据前述假设，则存储器地址引脚数量为 n 个，即地址信号可对应 2^n 个存储单元中任一存储单元；数据引脚为 w 个，即访问任一存储单元时，均同时写入或读出 w 位二进制数据。



（a）存储器结构与组成　　（b）存储器的存储空间

图 1.11　存储器组成及存储空间示意图

图 1.11（a）中存储器的存储空间可表示成图 1.11（b）的形式。其中：

存储字指存储单元所存放的信息，存储字长为存储字所含二进制信息的位数，即存储单元所含存储元的数量 w；

存储单元个数＝存储单元地址的个数＝2^n，存储器地址范围则为 n 位地址信号表示的存储单元地址的集合 0～2^n-1；

存储器容量＝存储单元个数×存储字长＝$2^n \times w$ 位

CPU 对存储器的操作通常只有读、写两种。由于存储器是按地址访问的，因此对存储器的读、写操作过程大体上均可分为两个步骤。

对于读操作，第①步是 CPU 向存储器发出欲访问的存储单元地址及读命令，存储器收到命令后，进行地址译码、选择存储单元等动作；第②步是 CPU 等到存储器将对应的存储字送到数据引脚后读取数据，并撤销所发信号，如图 1.12（a）所示。

对于写操作，第①步是 CPU 向存储器发出欲访问的存储单元地址及写命令，存储器收到命令后，进行地址译码、选择存储单元等动作；第②步是 CPU 向存储器发出所写数据，等到存储器将数据写入所选存储单元后，撤销所发信号，如图 1.12（b）所示。

（a）对存储器的读操作步骤　　（b）对存储器的写操作步骤

图 1.12　对存储器的读/写操作步骤

（2）运算器

运算器的主要功能是实现数据加工，即实现算术逻辑运算及暂存运算结果。

算术逻辑运算可以通过称为算术逻辑单元 ALU 的部件实现，通常能够实现四则算术运算和简单的逻辑运算。因此，ALU 一般有两个输入端（源操作数）、一个输出端（目标操作数），ALU 的组成原理将在第 2 章中详细讨论。

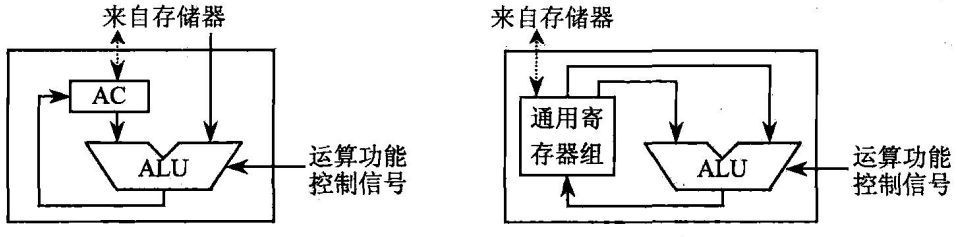
我们先看一个 C 语言程序段：

```
int nSum=0;
for (int i=0; i<100; i++)nSum += i;
```

从这个程序段可以看出，前一条指令的目标操作数经常用作后一条指令的源操作数，而指令和数据是存放在存储器中的。因此，为提高运算操作的性能，可将指令的目标操作数临时存放在运算器中，而不必每次写到存储器中。由“数字电路”课程所学的知识可知，用寄存器暂存数据是一个很好的方法。

由此可见，运算器通常由 ALU、寄存器及相关电路组成。我们将既用作 ALU 的源操作数、又用作目标操作数的寄存器称为累加寄存器（或累加器，记为 AC）。当运算器中只有一个累加器时，常称这种运算器为累加器型运算器，对应的 CPU 称为基于累加器的 CPU，如图 1.13（a）所示。

为了减少程序中指令读/写存储器的次数，现代 CPU 中通常都设置多个既可作源操作数、又可作目标操作数的寄存器，这些寄存器不再称为累加器，常合称为寄存器组（或寄存器堆），常称这种运算器为寄存器型运算器，对应的 CPU 称为基于寄存器的 CPU，如图 1.13（b）所示。



（a）累加器型运算器　　（b）寄存器型运算器

图 1.13　运算器结构图

对运算 $E \leftarrow (A+B) \times (C+D)$ 而言，若 $A \sim E$ 均在存储器中，则寄存器型运算器实现时，比累加器型运算器实现时减少了中间结果操作的 2 次存储器的读操作和写操作。

特别需要注意的是，由于 ALU 由硬件实现，ALU 入端和出端的数据位数是固定的。也就是说，数据在计算机中是按定长的方法存储和运算的，位数不够时在前面或后面补零。