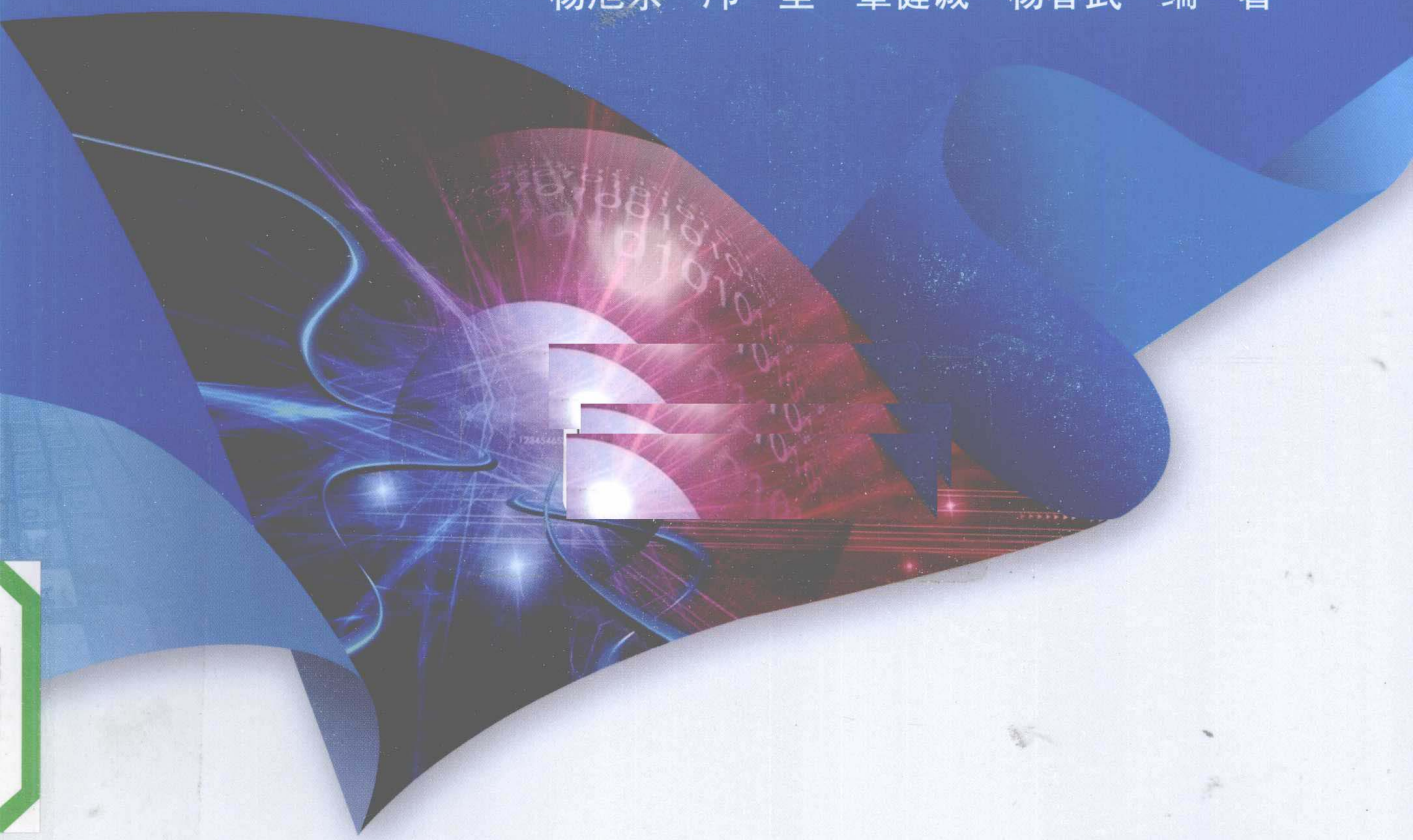
普通高等教育计算机类特色专业系列规划教材

计算机系统结构

（第三版·网络版）

白中英 主编
杨旭东 邝 坚 覃健诚 杨春武 编著



科学出版社
www.sciencep.com

普通高等教育计算机类特色专业系列规划教材

计算机系统结构

（第三版 · 网络版）

白中英　　主编
杨旭东　邝　坚
覃健诚　杨春武　编著

科学出版社
北京

内 容 简 介

本书介绍了计算机系统结构的基本概念、基本原理、基本分析和设计方法。全书共9章，内容包括：计算机系统结构的相关概念、时间并行技术、指令级并行技术、向量处理机、互连网络、阵列处理机、多处理机、机群系统和课程设计实验。

本书依据短学时教学要求编写，知识完整、结构合理、重点突出、概念清楚，注重实践环节与能力培养，形成了文字教材、电子教材、试题库、课程设计及教学仪器等综合配套的教学体系。

本书文字流畅，便于自学，有广泛的适应面，可作为高等院校计算机科学与技术专业本科生教材，也可作为成人教育教材和全国计算机等级考试（四级）参考书。

图书在版编目（CIP）数据

计算机系统结构：网络版/白中英主编；杨旭东等编著. —3版. —北京：科学出版社，2010.7

（普通高等教育计算机类特色专业系列规划教材）

ISBN 978-7-03-028126-5

I.①计… II.①白… ②杨… III.①计算机系统结构-高等学校-教材 IV.①TP303

中国版本图书馆CIP数据核字（2010）第119437号

责任编辑：贾瑞娜/责任校对：朱光光

责任印制：张克忠/封面设计：耕者设计工作室

科学出版社 出版

北京东黄城根北街16号

邮政编码：100717

http://www.sciencep.com

铭浩彩色印装有限公司印刷

科学出版社发行 各地新华书店经销

*

2002年7月第 一 版 开本：787×1092 1/16

2006年1月第 二 版 印张：13 4/3 插页：2

2010年7月第 三 版 字数：296 000

2010年7月第4次印刷 印数：15 000—19 000

定价：28.00元（含光盘）

（如有印装质量问题，我社负责调换）

前　言

数字逻辑、计算机组成原理、计算机系统结构是计算机科学与技术专业本科生硬件一条线的必修课程。第一门课是技术基础课，讲授逻辑部件级和数字系统级的分析与设计方法。第二门课是专业基础课，讲授单处理机系统的组成分析和设计方法，偏重于处理机的整机概念。而第三门课属于专业课，着重讲授并行计算机系统的基本概念、结构、分析和设计方法。

本教材的宗旨是：避免与先修课程**计算机组成原理**在内容上重复，突出以时间并行技术和空间并行技术为核心的并行计算机系统结构。

编写本教材的指导思想是：

（1）知识结构完备，基本概念清楚，内容少而精，以满足短学时教学的需要；

（2）力图反映新理论新技术，以适应计算机科学技术发展变化快的需要；

（3）理论教学与实践教学结合，注重学生的能力培养；

（4）文字教材和电子教材结合，采用先进的教学方法和手段，以获得良好的教学效率和教学质量。

根据作者多年从事理论教学和实践教学的经验，从传授知识和培养能力的目标出发，并结合本课程教学的特点、难点和要点，作者学习了国内外优秀教材，进行了课程体系、教学内容、教学方法和教学手段方面的改革，使文字教材、CAI 课件、网络教材、试题库、实验仪器、课程设计综合配套，力求形成“理论、实验、设计”三个过程相统一的教学体系。课内教学计划 40～48 学时，实践教学内容单独安排。

杨春武、祁之力、吴琨、白媛、杨秦、张杰、靳秀国、齐承军、宋丹杰、周柳忠、杨蕾、张春、张蓉蓉、段国乐、盛利、王锋、王军德、冯一兵、金丽霞等参加了 CAI 课件、网络教材、试题库、教学仪器的研制工作，限于篇幅，未能在封面上一一列名。

中国科学院计算技术研究所国家智能计算机研究开发中心陈鸿安研究员审阅了文字教材，在此表示衷心感谢！

作　者

2010 年 5 月 5 日

目　录

第1章 计算机系统结构的相关概念

本章介绍计算机系统的有关术语，计算机系统结构与技术的关系，系统结构的定量分析技术，系统结构的并行性发展。重点是：①计算机系统的层次结构、计算机系统结构、计算机组织、计算机实现三者的定义及其关系；②透明性、局部性原理、MIPS和MFLOPS定义；③Amdahl定律；④CPU性能公式。

1.1 系统结构的有关术语

1.1.1 计算机系统的层次结构

现代计算机系统是硬件、固件和软件组成的十分复杂的系统。为了对这个系统进行描述、分析、设计和使用，人们从不同的角度提出了观察计算机的观点和方法。其中常用的一种方法，就是从机器语言的角度出发，把计算机系统按功能划分成多级层次结构，如图1.1所示。

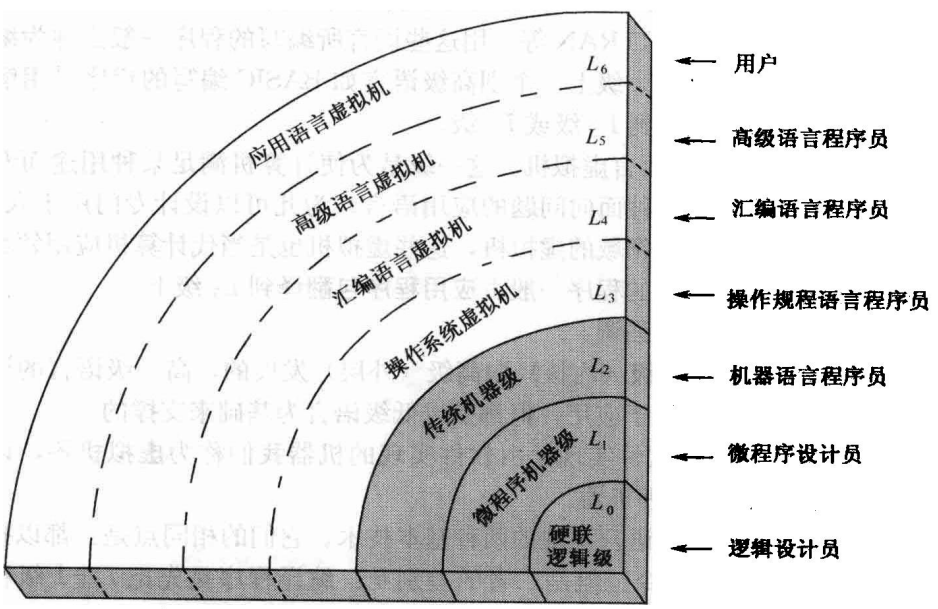


图1.1 计算机系统的多级层次结构

计算机的语言可分成一系列的层次级，最内层级语言的功能最简单，最外层级语言的功能最强。对于用某一层级语言编写程序的程序员来说，他一般不管其程序在机器中

是如何执行的，只要程序正确，他就能得到预期的结果。这样，对这层语言的程序员来说，他似乎有了一种新的机器，这层语言就是这种机器的**机器语言**，该机器能执行用该层语言编写的全部程序。因此，计算机系统就可以按语言的功能划分成多层次结构，每一层以一种不同的语言为特征。

最内层的 L_0 级是硬联逻辑级。这一级由门、触发器等逻辑电路组成，它是由逻辑设计员采用布尔语言设计的硬件内核。

L_1 级是微程序机器级。这一级的机器语言是二进制编码的**微指令集**。程序员用微指令编写的微程序由固件/硬件来解释实现。L_1 级的机器语言是二进制语言。

L_2 级是传统机器级。这一级的机器语言是**机器指令集**。程序员用机器指令集编写的程序可以由 L_1 级微程序进行解释，也可直接采用 L_0 级硬联逻辑进行解释。这个解释程序运行在 L_1 级上。

L_3 级是操作系统虚拟机。这一级机器语言中的多数指令是传统机器级指令。此外还提供操作系统级指令，如打开文件、读/写文件、关闭文件等指令。用这一级语言编写的程序，若与 L_2 级指令相同则由微程序解释，而操作系统级指令则由操作系统进行解释。操作系统是运行在 L_2 级上的解释程序。

L_4 级是汇编语言虚拟机。这一级的机器语言是**汇编语言**。用汇编语言编写的程序，首先翻译成 L_3 级和 L_2 级语言，然后再由相应的机器执行。完成汇编语言翻译的程序称作**汇编程序**。

L_5 级是高级语言虚拟机。这一级的机器语言是各种高级语言。目前高级语言已有上百种，如 C、C ++、FORTRAN 等。用这些语言所编写的程序一般由称为**编译程序**的翻译程序翻译到 L_4 级或 L_3 级上。个别高级语言如 BASIC 编写的程序采用解释的方法实现，即用**解释程序**翻译到 L_4 级或 L_3 级。

最外层的 L_6 级是应用语言虚拟机。这一级是为使计算机满足某种用途而专门设计的，因此这一级语言就是各种面向问题的应用语言。为此可以设计专门用于人工智能、教育、管理、计算机设计等领域的虚拟机，这些虚拟机也是当代计算机应用领域的重要研究课题。用应用语言编写的程序一般由**应用程序包**翻译到 L_5 级上。

由上面的叙述我们可以强调：

(1) 计算机语言是由低级（内核）向高级（外层）发展的，高一级语言的语句相对于低级语言功能更强，更便于应用，但都是以低级语言为基础来支撑的。

(2) L_3 级以上完全由软件实现。由软件实现的机器我们称为**虚拟机器**，以区别于由硬件/固件实现的实际物理机器。

(3) 编译和解释是机器语言实现的两种基本技术。它们的相同点是：都以执行一串 L 级指令来实现 $L+1$ 级指令。但是二者的差别是：编译程序是先把 $L+1$ 级程序全部变换成 L 级程序后，再去执行新产生的 L 级程序，在执行过程中 $L+1$ 级程序不再被访问。而解释程序是每当一条 $L+1$ 级指令被译码后，就直接去执行一串等效的 L 级指令，然后再去取下一条 $L+1$ 级的指令，依次重复进行。因此解释过程是边变换边执行的过程。在实现新的虚拟机器时，这两种技术都被广泛使用。一般来说，解释执行比编译执行花的时间多，但占用存储空间较小。

1.1.2　计算机系统结构

计算机系统结构（computer architecture）一词也译成**计算机系统结构**，目前作为专用术语被广泛使用。其经典定义是 1964 年 Amdahl 在介绍 IBM 360 系统时提出的：计算机系统结构是程序员所看到的计算机属性，即概念性结构与功能属性。

按照计算机系统的多级层次结构，不同级程序员所看到的计算机具有不同的属性。例如，传统机器级程序员所看到的计算机主要属性是该机指令集的功能特性，而高级语言虚拟机程序员所看到的计算机主要属性是该机所配置的高级语言所具有的功能特性。显然，不同的计算机系统，从传统机器级或汇编语言程序员的角度来看，具有不同的属性。但是从高级语言（如 C 语言）程序员看，它们就几乎没有什么差别，具有相同的属性。换句话说，这些传统机器级所存在的差别对高级语言程序员来说是“看不见”的，也是他们不需要知道的。在计算机技术中，对这种本来存在的事物或属性，但从某种角度看又好像不存在的概念称为**透明性**。通常，在一个计算机系统中，低层机器的属性对高层机器的程序员往往是透明的，如传统机器级的概念性结构和功能特性，对高级语言程序员来说是透明的。由此看出，在层次结构的各个级上都有它的系统结构。

为了不使概念具有多义性，**计算机系统结构**通常定义为：**机器语言程序员所看到的传统机器级所具有的属性，它包含概念性结构和功能特性两个方面**。这些属性是机器语言程序设计者（或者编译程序生成系统）为使其所设计（或生成）的程序能在机器上正确运行，所需遵循的计算机属性。对通用寄存器型机器来说，这些属性主要是指：

（1）数据表示（硬件能直接识别和处理的数据类型）；

（2）寻址规则（最小寻址单元、寻址方式及其表示）；

（3）寄存器定义（各种寄存器的定义、数据及使用方式）；

（4）指令集（机器指令的操作类型和格式、指令间的排序和控制机构）；

（5）中断系统（中断类型、中断响应硬件的功能等）；

（6）机器工作状态的定义和切换（如管态和目态等）；

（7）存储系统（主存容量、程序员可用的最大存储容量等）；

（8）信息保护（信息保护方式、硬件对信息保护的支持）；

（9）I/O 结构（I/O 连接方式、处理机/存储器与 I/O 设备间数据传送的方式和格式、I/O 操作的状态等）。

上述属性是计算机系统中由硬件或固件完成的功能，程序员在了解这些属性后才能编出在传统机器级上正确运行的程序。因此，计算机系统结构概念的实质是确定计算机系统中软硬件的界面，界面之上是软件的功能，界面之下是硬件和固件的功能。

1.1.3　计算机组织和计算机实现

计算机系统结构、计算机组织（computer organization）、计算机实现（computer implementation）是三个不同的概念。它们各自包含不同的内容，但又有紧密的关系。

计算机系统结构是指计算机系统的软、硬件的界面，即机器语言程序员所看到的传统机器级所具有的属性。

计算机组织是指计算机系统结构的逻辑实现，包括物理机器级内的数据流和控制流的组成以及逻辑设计等。它着眼于物理机器级内各事件的排序方式与控制方式，各部件的功能以及各部件的联系。计算机组织也被译成**计算机组成**。

计算机实现是指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，模块、插件、底板的划分与连接，信号传输，电源、冷却及整机装配技术等。它着眼于器件技术和微组装技术，其中器件技术在实现技术中占主导作用。

下面举例说明计算机系统结构、计算机组织、计算机实现三者之间的区别。

例 1.1 （1）机器指令集的确定属于计算机系统结构。

（2）指令的实现，如取指令、取操作数、运算、送结果等具体操作及其排序方式属于计算机组织。

（3）实现指令集中所有指令功能的具体电路、器件的设计、装配技术等属于计算机实现。

例 1.2 （1）确定是否有乘法指令属于计算机系统结构。

（2）乘法指令是用专门的乘法器实现，还是经加法器用重复的相加和右移操作来实现，属于计算机组织。

（3）乘法器、加法器的物理实现，如器件的选定（器件集成度、类型、数量、价格）及所用微组装技术等，属于计算机实现。

例 1.3 （1）主存容量与编址方式（按位、按字节、按字访问等）的确定属于计算机系统结构。

（2）为达到所定性能价格比，主存速度应多快，在逻辑结构上需采用什么措施（如多体交叉存储等）属于计算机组织。

（3）主存系统的物理实现，如存储器器件的选定、逻辑电路的设计、微组装技术的选定属于计算机实现。

可以看出，具有相同计算机系统结构（如指令系统相同）的计算机因为速度要求不同等因素可以采用不同的计算机组织。例如，取指令、译码、取操作数、运算、存结果可以在时间上按顺序方式进行，也可以让它们在时间上按重叠方式进行以提高执行速度。

同样，一种计算机组织可以采用多种不同的计算机实现。例如，主存器件可以采用SRAM 芯片，也可以采用 DRAM 芯片。可以采用大规模集成电路单个芯片，也可以采用中小规模集成电路进行构建。显然，这取决于性能价格比的要求与器件技术的现状。

1.1.4 计算机系统结构的分类

研究计算机系统分类方法有助于人们认识计算机的系统结构和组织的特点，理解系统的工作原理和性能。

通常把计算机系统按其性能与价格的综合指标分为巨型、大型、中型、小型、微型、单片机等。但是随着科学技术的进步，各类计算机的性能指标都在不断进步，以至于过去的一台大型机的性能还比不上今天的一台微型机；而用过去一台大型机的价钱，今天却能买一台性能指标高许多倍的新式大型机。可见，按巨、大、中、小、微、单来

划分的绝对性能标准是随时间而变化的。

按用途分类，计算机系统可分为科学计算、事务处理、实时控制、家用等。一般说来，计算机都是作为通用系统进行设计的，但是在用户编写程序时，却都带有专用性质。为了解决这个矛盾，采取的办法有：灵活地改变系统配置；适应特殊环境要求采取不同的物理安装；增加处理不同数据结构的能力；提供多种语言和操作系统以适应不同的需要。

按处理机个数分，计算机系统可分为单处理机、多处理机；按种类分有标量处理机、超标量处理机、超流水处理机、向量处理机、阵列处理机、对称多处理机、大规模并行处理机、机群系统等。

下面从计算机系统结构的并行性能出发，介绍两种常用的分类方法。

1. Flynn 分类法

1966 年 M. J. Flynn 按照指令流（instruction stream）和数据流（data stream）的不同组织方式，把计算机系统的结构分为以下 4 类：

（1）单指令流单数据流 SISD（single instruction stream single datastream）；

（2）单指令流多数据流 SIMD（single instruction stream multiple datastream）；

（3）多指令流单数据流 MISD（multiple instruction stream single datastream）；

（4）多指令流多数据流 MIMD（multiple instruction stream multiple datastream）。

图 1.2 所表示的是这 4 类计算机的基本结构框图。SISD 是传统的顺序处理计算机。

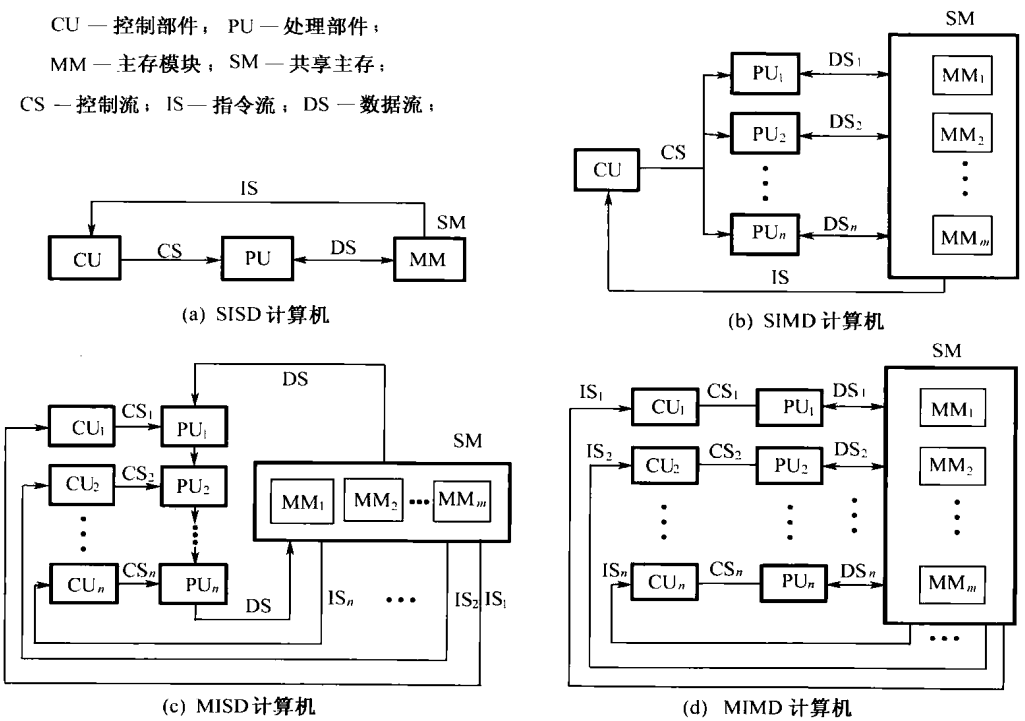


图 1.2　Flynn 分类法的 4 类机器结构

SIMD 以阵列处理机为代表。MISD 在实际中代表何种计算机，存在着不同的看法，有的文献把流水线结构的机器看成是 MISD 结构。MIMD 的代表是多处理机。

2. Händler 分类法

1977 年，Händler 根据并行度和流水线提出了另一种分类法。这种分类方法把计算机的硬件结构分成 3 个层次，并分别考虑它们的可并行-流水处理程度。这 3 个层次是：处理控制器 PCU、算术逻辑部件 ALU（或运算部件 PE）和位级电路 BLC。其中位级电路相当于在 ALU 中要进行 1 位运算时所需的基本逻辑电路。

这样，一个计算机系统可用 3 对整数来表示：

$$T(\text{系统型号}) = \langle k \times k', d \times d', w \times w' \rangle$$

式中：k 为处理控制器 PCU 的数目；k'为可组成流水线的 PCU 数目；d 为每个 PCU 所控制的 ALU（或 PE）数目；d'为可组成流水线的 ALU 数目；w 为 ALU 或 PE 的字长；w'为在所有 ALU 或一个 PE 中的流水段数目。

如果任意 1 对参数的第二个元素值为 1，则可将其省略不写。

例 1.4 CDC 6600 计算机系统有 1 个 CPU，它的 ALU 有 10 个功能部件，所有的功能部件可连成一条流水线，字长 60 位，则 CDC 6600 系统可描述为

$$T(\text{CDC 6600}) = T\langle 1, 1 \times 10, 60 \rangle$$

例 1.5 CRAY-1 计算机有 1 个 CPU，12 个相当于 ALU 或 PE 的处理部件，最多可以实现 8 级流水线。字长为 64 位，可以实现 1～14 位流水线处理。所以 CRAY-1 的系统结构可表示为

$$T(\text{CRAY-1}) = \langle 1, 12 \times 8, 64 \times (1 \sim 14) \rangle$$

1.2 系统结构发展的因素

1.2.1 存储程序计算机系统结构及其发展

冯·诺依曼等于 1946 年提出了现代计算机的雏形，它由运算器、控制器、存储器、输入设备和输出设备组成，如图 1.3 所示。

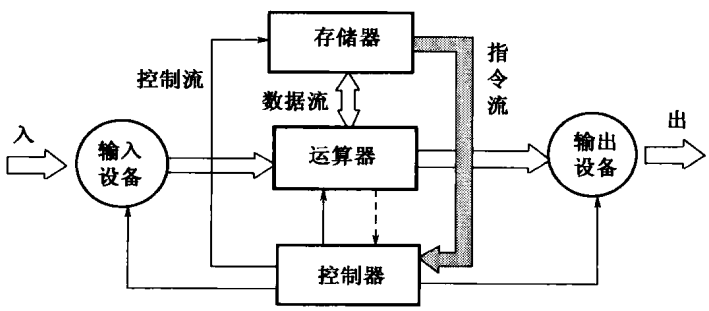


图 1.3 冯·诺依曼型计算机组成框图

冯·诺依曼型计算机以存储程序原理为基础。存储程序原理的基本点是指令驱动，即程序由指令组成，并和数据一起存放在计算机存储器中。机器一旦启动，就能按照程序指定的逻辑顺序逐条把指令从存储器中读出来并加以执行，自动地完成由程序所描述的处理工作。冯·诺依曼计算机的特征可概括如下：

(1) 存储器是字长固定、顺序线性编址的一维结构；

(2) 存储器提供可按地址访问的一级地址空间，每个地址是唯一定义的；

(3) 由指令形式的低级机器语言驱动；

(4) 指令的执行是顺序的，即按指令在存储器中存放的顺序执行，程序分支由转移指令实现；

(5) 机器以运算器为中心，输入/输出设备与存储器之间的数据传送都经过运算器；

(6) 运算器、存储器、输入/输出设备的操作以及它们之间的联系都由控制器集中控制。

虽然至今绝大多数计算机仍基于存储程序原理，但是经过数十年的发展，现代计算机的系统结构有了重大变化和改进。主要有以下几个方面：

(1) 高级语言与机器语言的语义距离缩小，从而出现了面向高级语言机器和直接执行高级语言机器；

(2) 硬件子系统与操作系统和数据库管理系统软件相适应，从而出现了面向操作系统机器和数据库计算机等；

(3) 出现了与 LSI、VLSI 器件相适应的计算机系统结构；

(4) 计算机系统功能分散化、专业化，从而出现了各种功能分布计算机，如外围处理机、通信处理机等；

(5) 为了获得高可靠性，从而研制了容错计算机；

(6) 为了适应特定应用环境，从而出现了各种专用计算机，如快速傅里叶变换机器、过程控制计算机等；

(7) 计算机系统结构从串行算法向并行算法转变，从而出现了向量计算机、阵列计算机、多处理机等；

(8) 计算机系统结构突破了指令驱动型的传统模式，出现了数据驱动型和需求驱动型，从而出现了数据流机和归约机；

(9) 为了处理非数值化信息，出现了人工智能计算机，主要的处理方法已不是依靠精确的算法进行数值计算，而是依靠有关的知识进行逻辑推理。

1.2.2　软件对系统结构的影响

软件是促使计算机系统结构发展的最重要因素。

随着计算机系统的发展和应用范围的扩大，已积累了大量成熟的系统软件和应用软件。因此，用户希望原先已开发的软件仍能在升级换代的新型号机器上使用，这就要求软件具有可兼容性，即可移植性。那么如何解决软件的可移植性呢？根据不同的要求，可采用以下 3 种方法。

1. 系列机方法

所谓**系列机**，是指在一个厂家内生产的具有相同的系统结构，但具有不同组成和实现的一系列不同型号的机器。如 IBM 370 系列机有 115、125、135、145、158、168 等一系列从低速到高速的各种型号。它们有相同的指令系统，但在低档机上指令的分析和执行是按顺序方式进行的，而在高档机上采用流水或其他并行处理方式。从程序设计者来看，各档机器具有相同的 32 位字长。

从系列机的软件兼容理论上讲，有向上兼容、向下兼容、向前兼容和向后兼容 4 种。向上（下）兼容是指按某档机器编写的程序，不加修改就能运行于比它高（低）档的机器。向前（后）兼容是按某个时期投入市场的某种型号机器编写的程序，不加修改就能运行于在它之前（后）投入市场的机器。实际上对向下兼容和向前兼容可以不做要求，向上兼容在某种情况下可能做不到，但是**向后兼容**是肯定要做到的。

不同厂家生产的具有相同系统结构的计算机称为**兼容机**。它的设计思想与系列机的设计思想是一致的。例如，Amdahl 公司照搬 IBM 370 的系统结构，以充分利用 IBM 370 的软件，但又采用了新的组成、实现和器件工艺，研制出性能价格比优于 IBM 370 的 Amdahl 470、480 等机器。兼容机还可以对原有的系统结构进行某种扩充，例如长城 0520 与 IBM PC 兼容，但有很强的汉字处理功能，因而具有较强的市场竞争能力。

2. 模拟和仿真方法

系列机方法只能用在具有相同系统结构的各种机器之间实现软件移植。为了在不同系统结构的机器之间实现软件移植，一般采用模拟或仿真方法。

所谓**模拟**，是指用软件方法在一台现有的计算机上实现另一台计算机的指令系统。若在 A 计算机上实现 B 计算机的指令系统，通常采用解释方法来完成。即 B 机的每一条指令用一段 A 机的指令进行解释执行，如同 A 机上也有 B 机的指令系统一样。此时 A 机称为**宿主机**，被模拟的 B 机称为**虚拟机**。为了使虚拟机的应用软件能在宿主机上运行，除了模拟指令系统外还需模拟其存储系统、I/O 系统、控制台的操作以及操作系统等。对应用软件的模拟也可采用类似方法。由于模拟是采用纯软件解释执行的方法，模拟程序放在主存中，因此运行速度较慢。

如果宿主机本身是采用微程序控制，则对 B 机指令系统每条指令的解释执行可直接由 A 机的一段微程序解释执行。这种用微程序直接解释另一种机器指令系统的方法称为**仿真**。此时 A 机仍称为**宿主机**，B 机称为**目标机**。为仿真所编写的解释微程序称为**仿真微程序**。除了仿真目标机的指令系统外，还需要仿真其存储系统、I/O 系统、控制台的操作等。由于仿真微程序存放在控制存储器中，因此仿真的运行速度比模拟的方法快。但微程序机器更依赖于计算机的系统结构，因此对系统结构差别较大的机器难于完全用仿真方法来实现软件移植，所以通常将模拟和仿真这两种方法混合使用。对于使用频率较高的指令，尽可能用仿真方法以提高运算速度，而对使用频率低且难于用仿真