

计算机原理与体系结构

李敬章 编著



中山大学出版社

TP303

L260

计算机原理与体系结构

李敬章 编著

中山大学出版社

·广州·

版权所有 翻印必究

图书在版编目 (CIP) 数据

计算机原理与体系结构/李敬章编著. —广州: 中山大学出版社, 1998. 2
ISBN 7-306-01364-5

I. 计… II. 李… III. 计算机系统结构 IV. TP303

中山大学出版社出版发行

(广州市新港西路 135 号)

番禺市市桥印刷厂印刷 广东省新华书店经销

787 毫米×1092 毫米 16 开本 30.25 印张 700 千字

1998 年 2 月第 1 版 1998 年 2 月第 1 次印刷

印数: 1—4000 册 定价: 39.80 元

前 言

计算机体系结构 (Computer Architecture) 是关于计算机构造、作用以及设计的研究, 是国内外权威的计算机教育机构规定的一个主科目, 其教学指导原则和大纲在参考文献 [21] 中作了详细的介绍。由于体系结构 (或称“系统结构”) 的主要课题涉及到对计算机整体属性和作用的本质理解, 因此体系结构课程往往是其他课程 (如“操作系统”、“计算机网络”等) 的基础。关于体系结构方面的知识, 无论对于硬件还是软件的应用、开发和维修, 都具有极为重要的指导意义。

本书是在作者多年来讲授“计算机原理与系统结构”、“数字逻辑”、“微机原理与接口技术”、“计算机网络”等课程教学材料基础上精炼而成, 以计算机认知方法论作指导, 全面而有重点地介绍计算机体系结构的主要课题, 且特别注重可读性、科学性、系统性和实用性。书中备有大量的专门设计的图表以及精选的例子和试题, 以方便广大读者使用。

本书取材新颖, 反映了本课程知识领域的最新发展, 例如: UCS 结构, SPEC 评测, 数据压缩, PCI 局部总线, 新兴存储技术, 多媒体技术, 大规模并行处理 (MPP), 因特网 (Internet), 等等。

本书可作为《计算机原理与系统结构》课程或其他类似名称课程的教材, 也可供高等教育自学考试、计算机软件专业技术资格和水平考试辅导班作为硬件应试辅导教材以及供从事计算机系统的应用、开发和维修的工程技术人员参考。

全书共分九章。第一章计算机系统概论, 介绍计算机的经典结构及演变, 虚拟机概念和多级层次结构, 计算机系统的组成、性能指标、应用领域以及学习计算机的观点和方法。第二章数据表示, 讨论数制及其转换, 定点数、浮点数的机器码表示, 西文字符和汉字的编码, 世界通用多八位编码字符集 (UCS), 奇偶校验码、海明校验码和循环冗余码, 最后讨论数据压缩编码。第三章数字逻辑概要, 紧密结合本课程需要, 介绍逻辑代数理论、逻辑设计方法以及计算机逻辑部件的功能特征和应用。第四章运算方法与运算器, 介绍数字系统的描述方法和设计原则, 讨论各种算法的特点以及根据算法设计运算部件的观点和方法。第五章指令系统, 讨论指令的结构及格式优化, 寻址方式的特点和应用, 指令系统的演变及改革。第六章控制器, 讨论控制器的基本功能及实现途径, 通过实例阐明组合逻辑控制器和微程序控制器的设计原理, 介绍微程序设计的技术和应用。第七章存储系统和存储体系, 介绍主存和辅存的结构、使用及性能指标, 讨论存储体系构成原理, 介绍交叉存储、双口存储、虚拟存储和高速缓存。第八章输入输出系统, 介绍总线技术和接口, 输入输出控制方式。第九章其他体系结构, 包括目前广泛流行的流水线结构, RISC 结构, 并行处理机结构, 计算机网络和多媒体系统, 其中“计算机网络”知识单元以计算机本质属性为出发点对计算机网络的概念、数据通信技术、网络的层次化协议、网间互连等问题作了精练的描述, 对快速、高效了解网络的主要问题有很大参考价值。

本书的出版得到自学成才大学校长黄远葆、《现代计算机》杂志社主编吴相辉以及中山大学出版社和冀建平、杨朝晖、李冀蕾等同志的大力支持和帮助, 在此对他们表示衷心感谢!

李敬章 1997 年 7 月

内容简介

本书以计算机认知方法论作指导，全面而有重点地介绍计算机体系结构的主要课题，内容包括概论、数据表示、逻辑代数和逻辑部件、运算方法和运算器、指令系统、控制器、存储系统和存储体系、输入输出系统，以及在基本体系结构基础上发展起来的其他体系结构，包括流水线结构、RISC 结构、并行处理机结构、计算机网络、多媒体系统。

本书取材新颖，反映了本课程知识领域的最新发展，例如：UCS 结构，SPEC 评测，数据压缩，PCI 局部总线，新兴存储技术，多媒体技术，大规模并行处理（MPP），因特网（Internet），等等。

本书可作为《计算机原理与系统结构》课程或其他类似名称课程的教材，也可供高等教育自学考试、计算机软件专业技术资格和水平考试辅导班作为硬件应试辅导教材以及供从事计算机系统的应用、开发和维护维修的工程技术人员参考。

目 录

第一章 计算机系统概论	(1)
1.1 计算机的经典结构及演变	(1)
1.1.1 计算机的抽象模型	(1)
1.1.2 冯·诺依曼型机器结构	(4)
1.1.3 冯·诺依曼型机器结构的演变	(6)
1.2 计算机系统的组成	(7)
1.2.1 硬件组成	(7)
1.2.2 软件组成	(8)
1.3 计算机系统的多级层次结构.....	(11)
1.3.1 虚拟机概念及多级层次结构	(11)
1.3.2 透明及视图	(13)
1.3.3 计算机系统结构的定义	(14)
1.3.4 关于体系结构的几个基本观点	(16)
1.4 计算机系统的主要性能指标.....	(17)
1.5 计算机的发展阶段及应用领域.....	(21)
第二章 数据表示	(23)
2.1 数值数据的机器码表示.....	(23)
2.1.1 数制	(23)
2.1.2 数的定点表示	(28)
2.1.3 数的浮点表示	(36)
2.1.4 十进制数的二进制编码	(41)
2.2 非数值数据的机器码表示.....	(45)
2.2.1 逻辑数据	(45)
2.2.2 七位和八位专用编码字符集	(45)
2.2.3 汉字编码	(47)
2.2.4 通用多八位编码字符集 (UCS)	(50)
2.3 校验码.....	(53)
2.3.1 奇偶校验码	(53)
2.3.2 水平垂直奇偶校验码	(55)
2.3.3 海明校验码	(57)
2.3.4 循环冗余码 (CRC)	(67)
2.4 数据压缩编码.....	(75)
2.4.1 数据压缩的基本概念	(75)

2.4.2 统计编码	(76)
第三章 数字逻辑概要	(80)
3.1 逻辑代数基础	(80)
3.1.1 逻辑命题与逻辑函数	(80)
3.1.2 逻辑代数的定律、定理、规则和常用公式	(83)
3.1.3 逻辑函数的化简	(87)
3.2 逻辑电路	(89)
3.2.1 集成电路的基本知识	(89)
3.2.2 逻辑门和触发器	(94)
3.2.3 逻辑电路的物理特性	(99)
3.2.4 组合逻辑电路	(103)
3.2.5 时序逻辑电路	(106)
3.3 逻辑部件	(107)
3.3.1 寄存器	(107)
3.3.2 计数器	(109)
3.3.3 数据多路选择器	(112)
3.3.4 译码器/数据多路分配器	(114)
3.3.5 编码器	(116)
3.3.6 存储器	(117)
3.3.7 可编程逻辑器件 (PLD)	(124)
第四章 运算方法及运算器	(127)
4.1 数字系统的描述与设计综述	(127)
4.1.1 数字系统的描述方法	(127)
4.1.2 设计分级与分级设计	(130)
4.2 定点加减法运算	(132)
4.2.1 定点加减法公式及溢出的检测	(133)
4.2.2 超前进位加法器的设计	(136)
4.2.3 十进制加法器的设计	(144)
4.3 定点乘法运算	(145)
4.3.1 原码乘法	(147)
4.3.2 补码乘法	(150)
4.3.3 组合阵列乘法器	(155)
4.3.4 保留进位乘法器	(156)
4.3.5 提高乘法运算速度的途径	(158)
4.4 定点除法运算	(159)
4.4.1 原码除法	(160)

4.4.2 补码除法	(163)
4.5 逻辑运算及 ALU 设计	(165)
4.5.1 逻辑运算	(165)
4.5.2 基于函数发生器的 ALU 设计原理及应用	(168)
4.6 定点运算器的组成与结构	(174)
4.6.1 定点运算器的基本组成	(174)
4.6.2 位片及专用 ALU 的结构	(176)
4.6.3 数据通路及其控制	(178)
4.7 浮点运算器的组成与结构	(182)
4.7.1 浮点数的基本运算规则及实现方法	(182)
4.7.2 浮点算术部件的基本结构	(187)
4.7.3 浮点运算器举例：80387 协处理器	(187)
第五章 指令系统	(194)
5.1 CPU 的功能及基本结构	(194)
5.1.1 CPU 的基本功能	(194)
5.1.2 CPU 的基本结构	(194)
5.1.3 指令执行过程	(198)
5.2 指令的结构和格式优化	(200)
5.2.1 指令的结构格式	(200)
5.2.2 操作码的优化表示	(201)
5.2.3 指令格式的优化表示	(204)
5.3 寻址方式	(204)
5.3.1 直接寻址方式	(205)
5.3.2 间接寻址方式	(208)
5.3.3 带位移量的寻址方式	(210)
5.3.4 零地址寻址方式	(213)
5.3.5 位、页面、块寻址方式	(214)
5.4 指令的功能类型及指令系统的变革	(215)
5.4.1 指令系统的完备性、有效性和兼容性	(215)
5.4.2 指令的功能类型	(216)
5.4.3 指令的冗余性	(217)
5.4.4 CISC 与 RISC	(219)
第六章 控制器	(222)
6.1 控制器的功能及实现综述	(222)
6.2 指令时序及控制方式	(227)
6.2.1 时钟周期、机器周期和指令周期	(227)
6.2.2 时序信号系统	(228)

6.2.3 控制方式	(231)
6.3 组合逻辑控制器	(232)
6.3.1 硬布线控制逻辑设计综述	(232)
6.3.2 组合逻辑控制器的设计原理	(233)
6.3.3 组合逻辑控制器设计举例	(235)
6.4 PLD 控制器	(240)
6.5 微程序控制原理	(242)
6.5.1 微程序控制的基本概念	(242)
6.5.2 微程序控制器设计举例	(243)
6.5.3 机器指令连续执行与微程序执行的交叉过程	(250)
6.6 微程序设计技术及应用	(251)
6.6.1 微程序设计技术	(251)
6.6.2 微程序设计的应用	(262)
第七章 存储系统及存储体系	(265)
7.1 存储器的分类及基本特征	(265)
7.1.1 存储器技术多样化及其原因	(265)
7.1.2 存储器的分类	(265)
7.1.3 存储器件的特征	(267)
7.2 主存储器	(269)
7.2.1 主存的基本结构及读/写操作	(269)
7.2.2 主存的阵列组织方式	(271)
7.2.3 半导体存储器的使用及存储容量的扩展	(274)
7.2.4 多体交叉存储器	(279)
7.2.5 双口存储器	(282)
7.2.6 主存校验	(284)
7.3 大容量辅助存储器	(286)
7.3.1 磁记录原理与磁记录方式	(286)
7.3.2 磁盘存储器	(289)
7.3.3 磁带存储器	(294)
7.4 存储体系原理	(296)
7.4.1 存储体系的形成和发展	(296)
7.4.2 存储体系的性能参数	(301)
7.4.3 程序的局部性	(303)
7.4.4 地址映像和变换	(304)
7.4.5 替换策略	(310)
7.5 虚拟存储器	(311)
7.5.1 虚拟存储器概念	(311)
7.5.2 页式虚拟存储器	(312)

7.5.3 段式虚拟存储器	(316)
7.5.4 段页式虚拟存储器	(318)
7.6 高速缓冲存储器 (CaCHE)	(319)
7.7 主存储器的管理与保护	(323)
第八章 输入输出系统	(326)
8.1 输入输出系统概述	(326)
8.1.1 外部设备的分类和特征	(326)
8.1.2 I/O 设备的编址方式	(327)
8.1.3 I/O 接口	(329)
8.1.4 总线技术	(331)
8.2 I/O 程序控制方式	(341)
8.2.1 无条件传送方式	(341)
8.2.2 状态驱动或程序查询方式	(342)
8.2.3 I/O 程序控制方式的评价	(344)
8.3 I/O 中断控制方式	(345)
8.3.1 中断的定义、目的及分类	(345)
8.3.2 中断响应条件和中断优先级	(346)
8.3.3 中断过程	(349)
8.3.4 中断判优及入口地址生成	(351)
8.3.5 中断控制器	(358)
8.3.6. I/O 中断控制方式的评价	(360)
8.4 直接存储器存取 (DMA) 控制方式	(361)
8.4.1 DMA 的基本概念及传送过程	(361)
8.4.2 DMA 的工作方式	(363)
8.4.3 DMA 控制器	(365)
8.4.4 DMA 控制方式的评价	(368)
8.5 通道和 I/O 处理机控制方式	(369)
8.5.1 通道处理机方式	(369)
8.5.2 输入输出处理机 (IOP) 和外围处理机 (PPU)	(373)
8.6 常用的输入输出设备	(375)
第九章 其他体系结构	(382)
9.1 流水线结构	(382)
9.1.1 重叠控制与流水控制方式	(382)
9.1.2 流水方式的性能参数	(385)
9.1.3 流水结构的分类	(387)
9.1.4 相关处理	(389)
9.2 RISC 结构	(392)

9.2.1 RISC 的特点	(392)
9.2.2 RISC 的技术类型	(394)
9.2.3 RISC 性能评测	(398)
9.3 并行处理机结构	(400)
9.3.1 并行性概念及分类	(400)
9.3.2 阵列处理机	(402)
9.3.3 向量处理机	(404)
9.3.4 大规模并行处理机 (MPP)	(408)
9.4 计算机网络	(411)
9.4.1 计算机网络综述	(411)
9.4.2 数据通信技术	(418)
9.4.3 计算机网络的体系结构和网间互连	(444)
9.5 多媒体系统	(462)
9.5.1 多媒体技术概述	(462)
9.5.2 多媒体系统的组成与结构	(468)
9.5.3 多媒体个人计算机 (MPC)	(471)
主要参考文献	(473)

第一章 计算机系统概论

本章的学习目的在于从本质上、总体上理解计算机。

内容包括：计算机的由来，“计算”的形式定义，计算机的抽象模型，计算机的经典结构与现代结构，计算机系统的组成，计算机系统的多级层次结构，计算机性能评价的主要指标，计算机的发展阶段及应用领域等。

关于这些问题的本质描述，包含着研究计算机结构的许多重要的观点，并且在以后各章节中反复使用到。

1.1 计算机的经典结构及演变

1.1.1 计算机的抽象模型

1. 计算机的简单定义

计算机的出现是基于社会的需要，即人们对计算的需要。人工计算需要人的大脑作控制，并使用纸、笔和计算工具（如算盘、计算器等）作辅助。人工计算的最大缺点是速度慢和容易出错。为解决这个问题，必须采用计算机来代替人。

计算机简单地可以定义为：代替人完成计算任务的机器。但这个定义并未回答以下三个本质的问题：

第一，计算机如何代替人？代替到什么程度？

第二，计算机中的“计算”与人的计算有何不同？

第三，机器与人有何差别？如何缩小这些差别？

这三个问题涉及的领域很广，且自第一台电子数字计算机诞生以来就一直存在和发展着，并继续存在于计算机今后的永恒发展之中。

第一个问题是目前人工智能研究的一个重要课题。

第二个问题的实质是各种各样的程序设计。

第三个问题是信息形式的表示和转换问题，依赖于各种输入输出设备和多媒体技术的发展。

由此可见，为解决这三个问题，计算机需要不断发展，才能满足人类不断增长的需要，发展是永无止境的。

2. 计算的形式定义

计算机中的“计算”，可以理解为求某个函数 $f(x)$ 的值：

$$Z = f(x)$$

其中， x ——原始给定的输入数据的集合，可能包括数字、字符、词句、文件、声音、图形、图像等。

Z ——输出数据的集合，即计算得到的结果，同样可能包括数字、字符、词句、文

件、声音、图形、图像等。

f ——“计算”要求的操作，包括数值计算以及非数值计算（如文件编辑、定理证明、声音合成、图形图像显示等）。

为了用一部特定的计算机求得 $f(x)$ ，最重要的事情是要把 f 表示成一串函数： $f_1, f_2, \dots, f_n$ ，并按顺序完成下列的基本操作：

$$\begin{aligned} y_1 &= f_1(x) \\ y_2 &= f_2(y_1) \\ &\dots \\ y_{n-1} &= f_{n-1}(y_{n-2}) \\ Z &= f_n(y_{n-1}) \end{aligned}$$

这就是计算的形式定义。集合 $P = \{f_1, f_2, \dots, f_n\}$ 通常就是一份程序。

集合 P 归根到底是依据机器的指令系统确定的。指令系统是计算机能够执行的基本功能（操作）的集合。计算机通过执行人们在 P 中给定的指令序列，完成人赋予它的计算使命。人们为了编制程序的方便，又常常通过别的形式，使用其他软件工具诸如高级语言、数据库语言、应用语言等来确定计算所需的集合 P' ，然后经过编译软件自动地将 P' 转换为机器能够懂得的集合 P ，并命令机器代替人完成计算任务。

3. 计算机的抽象模型——Turing 机

在世界公认的第一台电子数字计算机 ENIAC 诞生之前 10 年，即 1936 年，英国数学家图灵（A.M.Turing, 1912 - 1954）提出了一个计算机抽象模型，现在就称之为 Turing 机。

Turing 机有两个基本单元：存储器 M 、处理器 P 。其构成如图 1.1 所示。

Turing 机的存储器是一条无限长的带 M ，带上划分成一个个小方格。每个小方格可以是空白，亦可以记录有限个符号中的任一个。

Turing 机的处理器 P 是一部具有有限个状态的数字机器，它具有一个读写头，能够读出或改变带上任一方格的内容。为了将读写头移动到任一方格上面，

处理器可按照指令要求将读写头从现行位置开始向左或向右移动一个方格。

Turing 机具有很小的指令系统，只定义了四种操作 θ_j ：

- (1) $\theta_j = t_j$ ，其意义是将符号 t_j 写入到带上。 t_j 可以是“1”、 b （空格）以及分隔符等。
- (2) $\theta_j = R$ ，其意义是将读写头从当前方格向右移动一个方格。
- (3) $\theta_j = L$ ，其意义是将读写头从当前方格向左移动一个方格。
- (4) $\theta_j = H$ ，停止计算。

指令的格式如下：

$$S_h \quad t_j \quad \theta_j \quad S_k$$

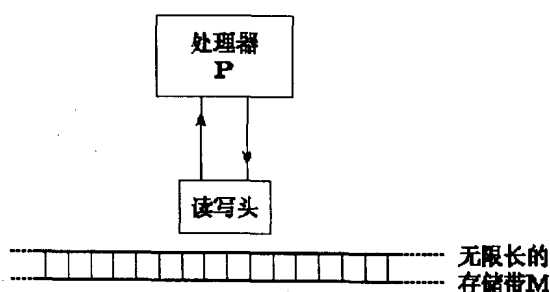


图 1.1 Turing 机的组成

其含义是：当处理器 P 处于 S_h 状态且符号 t_j 当时处于读写头的下面，那么执行操作 θ_j 且将 P 的状态改变为 S_k 。

例 1 若写出以下指令：

$$S_0 \quad 1 \quad R \quad S_1$$

则处理器 P 控制原来处于符号“1”上面的读写头右移一个方格，并从状态 S_0 变到 S_1 。

例 2 若写出以下指令：

$$S_1 \quad b \quad 1 \quad S_2$$

则处理器 P 对读写头下面的空格写入“1”，并从状态 S_1 变到 S_2 。

例 3 求两个自然数 n_1 与 n_2 之和。要求解这个问题，可遵循以下步骤：

(1) 将已知数据 n_1 和 n_2 以适当的编码形式写入到空白带上，用 n_1 个“1”代表数 n_1 ，用 n_2 个“1”代表数 n_2 ，数与数之间用 b（空格）分隔开如下：

$$\underbrace{b111\cdots 111}_{n_1} b \underbrace{111\cdots 111}_{n_2} b b$$

- (2) 置读写头的初始位置，设为左边的空格 b；
- (3) 将读写头右移 $n_1 + 1$ 次，到达中间空格 b 的上面；
- (4) 对中间的空格写入“1”；
- (5) 将读写头左移 n_1 次，到达最左边的“1”；
- (6) 清除最左边的“1”，写入空格 b；
- (7) 停止计算。

于是，计算结果就在带上：

$$\underbrace{bb111\cdots 1111111\cdots 111b}_{n_1 + n_2}$$

按以上步骤就可以编制出计算两个自然数之和的 Turing 机程序，见图 1.2。

Turing 机有何意义？

图灵本人从提出 Turing 机模型开始，与当时其他数学家一道在 30 年代后半期发表了许多论文，论证了一个问题：凡是可用算法处理的问题，使用自动机来解算的可能性。结论是：如果任何复杂的计算问题都可以分解为执行一系列基本操作来求解，那么 Turing 机可以实现所有可能的计算。

Turing 机的理论为后世所公认。在数学界，所谓“可计算函数”就被定义为：使用 Turing 机在有限步内可以完成计算的函数。

注意到以下的事实是十分重要的：Turing 机中存储带的状态数目是无限的，而实际的计算机存储器的容量都是有限的；Turing 机的解题时间也没有限制，只要是有限步就

指	令				
S_h	t_j	θ_j	S_k	说	明
S_0	b	R	S_0	置初态，并右移一格	
S_0	1	R	S_1	将读写头右移 n_1 格	
S_1	1	R	S_1		
S_1	b	1	S_2	在 n_1 和 n_2 之间的空格处写入 1	
S_2	1	L	S_3	将读写头左移 $n_1 + 1$ 格	
S_3	1	L	S_3		
S_3	b	R	S_4	返回到左边第一个 1	
S_4	1	b	S_5	清除最左边的 1	
S_5	1	H	S_0	停止计算， $n_1 + n_2$ 已在带上	

图 1.2 计算两个自然数之和的 Turing 机程序

可以。而由实际计算问题的求解算法的时间复杂性与/或空间复杂性所决定，其所需存储空间之大及计算时间之长，以致可能没有一台实际的计算机能够求解它们。这类问题称为“难处理的问题”，亦即理论上可求解但实际上不能解的问题。例如， n 个变量的逻辑函数共有 2^q 个，其中 $q=2^n$ 。当 n 比较大时， 2^q 是个很大的数；若 $n=10$ ，则 $2^q \approx 10^{308}$ 。故当使用计算机程序算法来化简逻辑函数时，由于算法的复杂性随输入规模（ n 的大小）而作指数式增加，以致在实际计算机上求解时就不得不对 n 的大小加以限制。 n 过大就变成难处理的问题，因为没有足够的存储空间来存储数据，也不能在有限的、可接受的时间内完成计算。由此看来，计算机的局限性表现在两个方面：时间上的局限性和空间上的局限性。

总之，在电子数字计算机诞生之前，Turing 机理论揭示了研制自动机求解计算问题的前景。后来的计算机无论是硬件设计还是软件编程，无不遵循了 Turing 机理论中将复杂问题分解为基本操作的思路。然而实际的计算机受到计算速度和存储器容量的限制，其计算能力不如 Turing 机。这意味着计算机的发展必须不断地克服其时空局限性，几十年的发展业已证明了这一点。

1.1.2 冯·诺依曼型机器结构

世界公认的第一台电子数字计算机是 1945 年底制成、1946 年予以公布的，名字是 ENIAC，即“电子数字积分机与计算机”的缩写。ENIAC 由宾夕法尼亚大学穆奇里（John Mauchly）和埃克特（J. Prespeer Eckert）两人主持设计，参加人数约 200 人，耗资 40 万美元。

ENIAC 采用 18000 多个电子管，体积为 $10 \times 100 \times 4$ 英尺，功率要求 50 千瓦，重量 30 吨。用今天的眼光看，如此笨重的机器其性能水平并不高，它只有 20 个累加寄存器作为存储器使用，也只有为数不多的专用寄存器用来存放常数；字长为 10 位十进制数，加法运算速度是 5000 次/秒，乘法运算速度是 360 次/秒。总的来说，ENIAC 的性能水平还不如 70 年代后期的一个小型计算器。然而，ENIAC 的重大意义在于开创了电子计算的先河，为其后的电子数字计算机的研制提供了宝贵经验。

ENIAC 有 750 条指令，没有程序存储器，靠人工设置开关和插入与拔出导线插头的方式来编制程序。程序的输入时间很长，调试修改极为困难。作为 ENIAC 的设计顾问，出生于匈牙利的数学家冯·诺依曼（Jhon von neumann, 1903–1957）对此深有体会，他和同事们吸取了 ENIAC 的研制经验，在其后设计的一台新的计算机 EDVAC（“电子离散变量自动计算机”的缩写）中提出了一种“存储程序”的思想，又在研制新的存储程序计算机 IAS 时提出了一系列新的建议，发表了一系列报告，从而奠定了计算机系统结构设计的基础，后人就将其当时提出的机器结构称之为“冯·诺依曼型机器结构”。

冯·诺依曼型机器结构的出现，对计算机的发展产生了深远的影响。迄今为止，绝大多数计算机的基本结构都基于冯·诺依曼型机器结构。

1. 存储程序思想

存储程序思想指的是，将程序（指令）和数据存储在同一个存储部件中。显然，ENIAC 是不符合存储程序思想的，其编程方法已如前所述。EDVAC 和 IAS 是依据存储程序思想的，尽管 EDVAC 的主存是只有 1K 字（每字 44 位）的超声波汞延迟线存储器，而 IAS

的主存是只有 4K 字（每字 40 位）的阴极射极管存储器。由此可见，ENIAC 设计之所以没有采用“存储程序”的部件，归根到底是受到客观物质条件的限制，当时并无大容量的快速存储器可供利用来存储程序。

存储程序思想的本质是什么？注意到这一点是特别重要的：计算机本身对存储器中存放的程序和数据二者是不加区分的，亦即指令和数据均以二进制代码形式存储，计算机只认识代码，而不管它是指令还是数据。至于存储器中哪些是指令，哪些是数据，是由程序设计人员约定或指定的。这样一来，通过执行程序来修改存储单元的内容，就能修改构成程序本身的指令。简言之，程序能够修改它的自身，或者说允许程序员编制自我修改的程序。存储程序思想的最大好处是使得程序便于编制和修改；但弄得不好，却又成了现代计算机病毒的技术根源。

2. 冯·诺依曼型机器结构的基本特点

① 指令和数据均以二进制代码的形式表示，电子线路采用二进制。这与 ENIAC 根本不同，ENIAC 直接采用十进制。

② 存储器中的指令和数据的形式一致，机器对它们同等对待，不加区分。

③ 指令在存储器中是按其执行顺序存储，并使用一个指令计数器（程序计数器）来控制程序的执行方向，实现顺序执行或转移。解题的算法，只能是顺序型的。

④ 存储器的结构是按地址访问的顺序线性编址的一维结构。

⑤ 计算机由五大部分组成，包括运算器、控制器、存储器、输入和输出设备。方框图如图 1.3 所示，整个结构以运算器为中心，数据流动必须经过运算器，并由控制器集中控制。

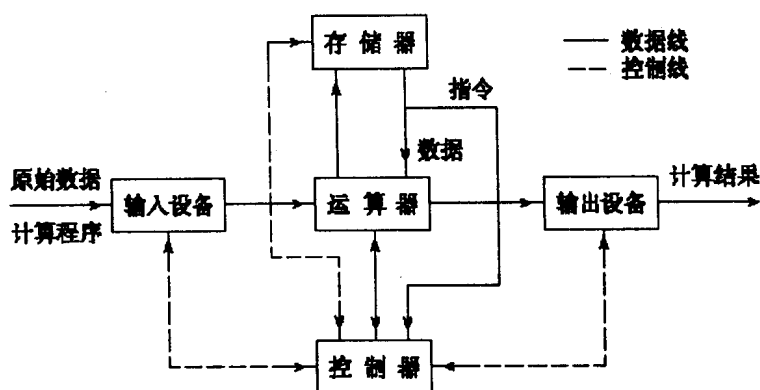


图 1.3 冯·诺依曼型机器结构

⑥ 指令由操作码和地址码两大部分组成。操作码确定操作的类型，地址码指明操作数的地址。

⑦ 一个字的各位同时进行处理，即是并行的字处理。

⑧ 运算器是在加法器的基础上制成的。

基本的五大部件相互关系及工作过程如下：

① 通过输入设备输入原始数据、计算程序以及给控制器的控制命令等，由控制器控

制，将有关部分存入存储器（存储程序）。

② 在控制器控制下从存储器读出一条指令到控制器，经译码分析发出全机的控制信号。需要的话，还要从存储器读出数据到运算器。

③ 运算器在控制器控制下完成规定操作，并将结果送回存储器或送到输出设备输出。

④ 从存储器取出下一条指令，重复②，直至程序中包含暂停指令（程序结束）为止。

1.1.3 冯·诺依曼型机器结构的演变

以上所述的冯·诺依曼型机器结构，是一种宏观的结构，说现代计算机的结构仍然属于冯·诺依曼型机器结构的范畴，也正是从宏观上说的。但是，实际上几十年来计算机的系统结构也是有一定发展的，特别是在微观结构方面，这是基于 LSI/VLSI 工艺的发展而产生的。然而无论如何，冯·诺依曼当年提出的关于计算机系统结构的几个基本点（①～④）至今并无改变。

现代计算机的结构与冯·诺依曼型机器结构比较，主要的演变有以下几方面：

① 现代大多数机器将运算器和控制器集成在一块芯片上，称为微处理器（MPU），并常用它作为中央处理部件（CPU）。甚至将部分存储器和输入输出接口与运算器、控制器集成在一个芯片上，称为单片微处理器。

② 现代机器以 CPU 为中心或以内存为中心，而不再是以运算器为中心。

③ 现代机器采用总线结构。总线的作用是将各个部件互连起来，并实现各部件间正确地传输数据。总线有许多种类，如单重总线、双重总线、多级或多层总线等。单重总线又称单总线，见图 1.4。

单总线包括数据总线、地址总线和控制总线，每一总线又包含若干根传输线。单总线以 CPU 为中心，即由 CPU 来控制总线，如指定哪个部件可以占用总线，何时占用总线，占用时间多长，何时交还总线等，一句话，控制各部件争用总线的问题。

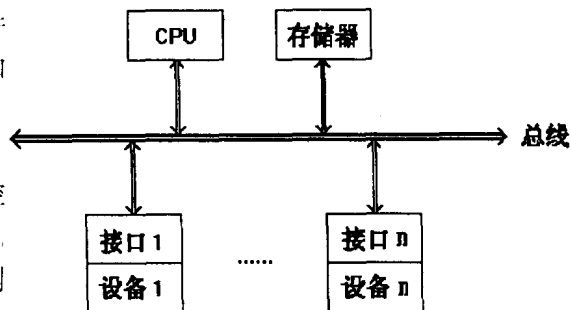


图 1.4 单总线结构

在 CPU 与各部件之间，还通过总线进行状态信息和命令信息的交换。为实现总线结构的连接，各部件的输出部分通常需要采用三态门或 OC 门。

双重总线结构将总线分为两个部分：存储总线和输入输出（I/O）总线。双重总线以 CPU 或存储器为中心。采用双重总线可使存取数据和输入输出数据分别在不同的路径上传输，以加快信息的传送速度。

多级或多层次总线是传统总线技术的最新发展，总线不再是简单的传输线问题，而是复杂的硬件（VLSI 芯片）和软件的结合。有关总线的知识将在本书的各有关部分作出详细介绍。

④ 在微观结构上，亦有很大的发展，例如：

先行控制——实现指令重叠执行；

流水线——提高程序执行的速度；