

高等教育质量工程信息技术系列示范教材

计算机 组成原理 解题参考

(第7版)

张基温 编著



清华大学出版社

高等教育质量工程信息技术系列示范教材

计算机组成原理解题参考

(第7版)

张基温 编著

清华大学出版社

北京

内 容 简 介

本书是《计算机组成原理教程(第7版)》(以下简称《教程(第7版)》)的姊妹篇,并按照《教程(第7版)》的章结构组织,每一章的内容都由如下4部分组成:《教程(第7版)》对应章中包含的知识要点、全部习题的解析、自测练习题目和自测练习参考答案。

本书题目丰富,解析详尽,既有知识要点,又有供自测的习题,可以作为计算机组成原理课程的参考教材,也可以供考研者复习参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

计算机组成原理解题参考/张基温编著. —7版. —北京:清华大学出版社,2017
(高等教育质量工程信息技术系列示范教材)
ISBN 978-7-302-47630-6

I. ①计… II. ①张… III. ①计算机组成原理—高等学校—题解 IV. ①TP301-44

中国版本图书馆 CIP 数据核字(2017)第 153992 号

责任编辑:白立军 战晓雷

封面设计:常雪影

责任校对:梁毅

责任印制:刘海龙

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京泽宇印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:10.25 字 数:244千字

版 次:1998年9月第1版 2017年7月第7版 印 次:2017年7月第1次印刷

印 数:1~1500

定 价:29.00元

产品编号:069558-01

前 言

《计算机组成原理教程》(以下简称《教程》)一书在 1998 年第 1 版出版后,曾经应广大读者要求编写了《计算机组成原理教程题解与实验指导》一书,由清华大学出版社于 2001 年 1 月出版。之后《教程》几经改版,由于每次改版时修改的内容都比较多,习题也变化较大,不少读者希望能有一本习题解析。但由于笔者事情繁多,一直没有能满足读者们的要求。

2007 年《教程》第 4 版出版之际,孙仲美高级工程师和李爱军教授愿意承担习题指导的任务,才使我如释重负。在我们三人的合作下,完成了针对《教程》第 4 版的习题指导改写。之后,《教程》又进行了一次改版,但由于笔者忙于他务,没有对习题指导进行修改。《教程》第 5 版出版后,我从多个渠道获知有大量读者希望提供相应的习题指导。现在,《教程》第 7 版的改写即将完成,在改写时对其结构也进行了一些调整,于是腾出时间改写对应的习题指导。考虑自己的学术水平有限,不再想用“指导”二字,采用了“参考”二字。相信有更高明的读者可以做出更合适的答案来。

《教程》从第 6 版起分为 7 章。第 7 章为未来计算机展望,仅作为知识扩展和思维启示,许多方面尚无确定性结论,所以本书没有包括这一章。

考虑到读者的实际需要,本书的每章仍由 4 部分组成:知识要点、习题解析、自测练习和自测练习参考答案。

在本书即将出版之际,谨对孙仲美高级工程师和李爱军教授深表谢意。因为本书中仍然有他们辛劳的痕迹。在本次修订中,赵忠孝、姚威、史林娟、张展为、戴璐、董兆军、张秋菊、陈觉参加了部分写作。

由于本书涉及了许多最新知识,可以参考的书籍较少,所以,尽管本人尽力进行了校订,但不足和错误之处在所难免。广大读者在使用中如果发现问题或者有改进意见,请毫无保留地反馈给我。我在此谨表衷心感谢。

张基温

2017 年 5 月于羊城小海之畔

目 录

第 1 章 计算机系统结构概述	1
1.1 知识要点	1
1.1.1 自动计算机的形成轨迹	1
1.1.2 现代计算机体系结构	1
1.1.3 电子数字计算机硬件工作原理	2
1.1.4 冯·诺依曼计算机体系的改进	8
1.1.5 计算机系统的主要性能指标	11
1.1.6 计算机系统性能测试工具与天梯图	14
1.2 习题解析	15
1.3 自测练习	28
1.3.1 选择题	28
1.3.2 填空题	30
1.3.3 判断题	31
1.3.4 简答题	31
1.4 自测练习参考答案	32
1.4.1 选择题参考答案	32
1.4.2 填空题参考答案	32
1.4.3 判断题参考答案	33
1.4.4 简答题参考答案	33
第 2 章 存储系统	37
2.1 知识要点	37
2.1.1 主存储器概述	37
2.1.2 主存储器组织	40
2.1.3 DRAM 内部操作与性能参数	42
2.1.4 辅助存储器	48
2.1.5 存储体系	50
2.2 习题解析	53
2.3 自测练习	64
2.3.1 选择题	64
2.3.2 填空题	67
2.3.3 判断改错题	68
2.3.4 简答题	68
2.3.5 综合题	70

2.4	自测练习参考答案	71
2.4.1	选择题参考答案	71
2.4.2	填空题参考答案	71
2.4.3	判断改错题参考答案	72
2.4.4	简答题参考答案	73
2.4.5	综合题参考答案	76
第3章	总线与主板	80
3.1	知识要点	80
3.1.1	总线的有关概念	80
3.1.2	总线工作原理	81
3.1.3	总线标准	84
3.1.4	主板	84
3.2	习题解析	85
3.3	自测练习	89
3.3.1	选择题	89
3.3.2	填空题	90
3.3.3	简答题	91
3.4	自测练习参考答案	92
3.4.1	选择题参考答案	92
3.4.2	填空题参考答案	92
3.4.3	简答题参考答案	93
第4章	I/O 机制	97
4.1	知识要点	97
4.1.1	外部设备	97
4.1.2	输入输出中的数据传送控制	98
4.1.3	设备接口	99
4.1.4	I/O 设备管理	106
4.2	习题解析	106
4.3	自测练习	118
4.3.1	选择题	118
4.3.2	填空题	120
4.3.3	简答题	122
4.4	自测练习参考答案	122
4.4.1	选择题参考答案	122
4.4.2	填空题参考答案	122
4.4.3	简答题参考答案	123

第 5 章 控制器逻辑	128
5.1 知识要点	128
5.1.1 处理器的外特性——指令系统.....	128
5.1.2 指令的时序.....	129
5.1.3 控制器设计.....	130
5.2 习题解析	130
5.3 自测练习	139
5.3.1 选择题.....	139
5.3.2 填空题.....	140
5.3.3 简答题.....	141
5.4 自测练习参考答案	141
5.4.1 选择题参考答案.....	141
5.4.2 填空题参考答案.....	141
5.4.3 简答题参考答案.....	142
第 6 章 处理器架构	144
6.1 知识要点	144
6.1.1 指令级并行技术.....	144
6.1.2 向量处理机.....	145
6.1.3 线程级并行技术.....	146
6.1.4 超线程技术.....	146
6.1.5 多核处理器.....	146
6.1.6 处理器并行技术小结.....	147
6.2 习题解析	147
6.3 自测练习	151
6.3.1 选择题.....	151
6.3.2 填空题.....	152
6.3.3 简答题.....	152
6.4 自测练习参考答案	153
6.4.1 选择题参考答案.....	153
6.4.2 填空题参考答案.....	153
6.4.3 简答题参考答案.....	154
参考文献	156

第1章 计算机系统结构概述

1.1 知识要点

本章是全书的概述,内容主要包括如下5个方面:

- (1) 自动计算机的形成轨迹。
- (2) 现代计算机的体系结构。
- (3) 现代电子数字计算机硬件的基本工作原理。
- (4) 冯·诺依曼计算机的改进。
- (5) 计算机系统的主要性能指标与测试。

1.1.1 自动计算机的形成轨迹

今天,电子计算机已经是无处不在,无所不能。之所以如此,是因为它是一种可以自动工作的机器。这种自动工作机制的形成是沿着如图 1.1 所示的 4 条轨迹发展而成的:

- (1) 从外程序控制到内程序控制——从算盘口诀到提花机,再到杰卡德织机,再到巴贝奇分析机,最后到冯·诺依曼体系。
- (2) 从外动力到内动力驱动——从手工驱动到电气驱动,再到电子世界。
- (3) 从十进制到二进制表示——布尔代数和原码、补码、反码、移码。
- (4) 从人工管理到程序自动管理——从裸机到操作系统管理。

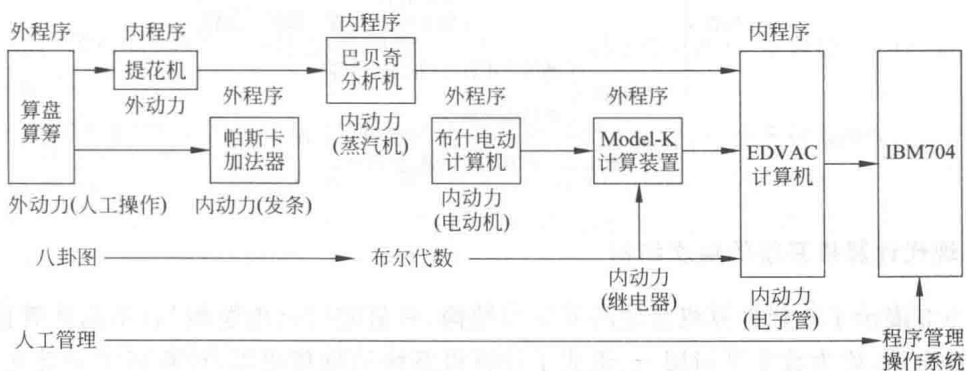


图 1.1 现代计算机形成轨迹

1.1.2 现代计算机体系结构

1. 冯·诺依曼型计算机体系结构

冯·诺依曼结构也称普林斯顿结构,其主要特点如下:

- (1) 使用单一的处理部件来完成计算、存储以及通信的工作。

- (2) 存储单元是定长的线性组织。
- (3) 存储空间的单元是直接寻址的。
- (4) 使用低级机器语言,指令通过操作码来完成简单的操作。
- (5) 对计算进行集中的顺序控制。
- (6) 计算机硬件系统由运算器、存储器、控制器、输入设备、输出设备五大部件组成,并规定了它们的基本功能。
- (7) 采用二进制形式表示数据和指令。
- (8) 在执行程序和处理数据时,必须将程序和数据首先从外存储器装入主存储器中,然后才能使计算机在工作时自动地从存储器中取出指令并加以执行。

2. 计算机系统的模块结构

图 1.2 为组成一个完整的计算机系统的软硬件模块结构。可以看出,计算机系统由硬件和软件两大部分组成。



图 1.2 计算机系统模块组成

3. 现代计算机系统的层次结构

图 1.3 展示了现代计算机系统的 6 层次结构,其最底层是由逻辑门(由晶体管做成)组成的逻辑电路,称为数字逻辑层,它组成了计算机系统的物理机器,计算机的全部功能都建立在此物理机器之上。

1.1.3 电子数字计算机硬件工作原理

电子数字计算机硬件部分的工作原理主要涉及如下 5 个部分:

- (1) 信息的 0、1 码表示方法。
- (2) 运算器工作原理。
- (3) 计算机存储器基本原理。

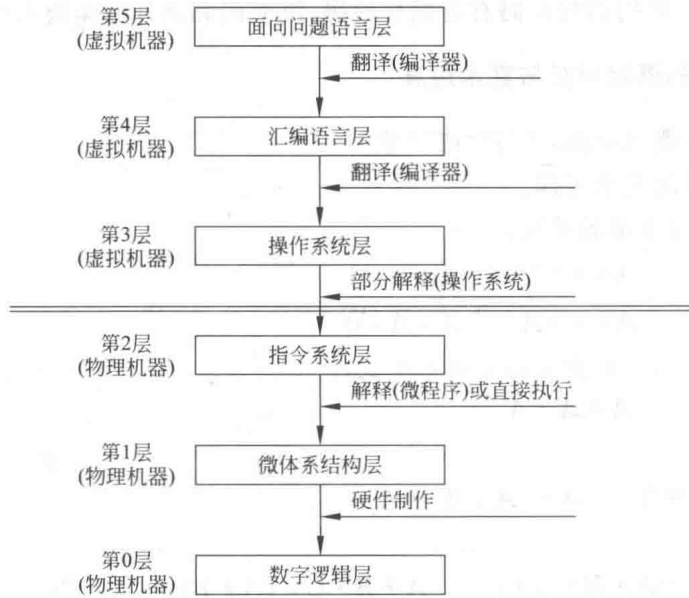


图 1.3 计算机系统的 6 层次结构

- (4) 控制器工作原理。
- (5) 计算机的时序控制。

1. 信息的 0、1 码表示方法

(1) 数值数据的 0、1 编码主要涉及以下 3 个问题：

- 十进制数与二进制数之间的相互转换。
- 二进制数的计算规则。
- 如何解决符号位参加计算的问题——将 0、1 编码的机器码分为原码、反码、补码和移码。

(2) 字符数据的 0、1 编码从字符的输入到存储,再到输出,涉及 3 种编码：

- 外码,主要指汉字的输入编码,如字形码、拼音码、流水码以及组合。
- 内码,如 ASCII 码、Unicode、GB 2312—1980 和 GB 2312—1990 等。
- 字库。

(3) 声音的 0、1 编码主要涉及模拟值的离散化处理——采样和量化,并用两个技术参数——采样频率和量化精度来衡量。

(4) 图像的 0、1 编码主要有两种手段：

- 矢量法。
- 位图法。有两个基本质量参数——分辨率和色彩深度(也称像素深度和位分辨率)。

(5) 指令的 0、1 编码通常将指令分为操作码和地址码两部分。操作码的位数决定了指令的多少,地址码可以分为单地址码、二地址码和三地址码等。每个地址码的位数决定了可寻址空间的大小。

(6) 校验码与纠错码：校验码仅能发现数据传输有无出错;纠错码不仅可以发现错误,

还可以纠正错误。常用的校验码有奇偶校验码、汉明码和循环冗余检验码(CRC)。

2. 开关电路的逻辑运算与算术运算

(1) 三种基本逻辑电路：“与”“或”“非”。

(2) 逻辑代数的基本定律。

① 关于变量与常量的关系：

$$A+0=A \quad A+1=1 \quad \bar{A}+A=1$$

$$A \cdot 0=0 \quad A \cdot 1=A \quad A \cdot \bar{A}=0$$

② 重复律：

$$A \cdot A=A \quad A+A=A$$

③ 吸收律：

$$A+A \cdot B=A \quad A \cdot (A+B)=A$$

④ 分配律：

$$A(B+C) = A \cdot B + A \cdot C \quad A+B \cdot C = (A+B) \cdot (A+C)$$

⑤ 交换律：

$$A+B=B+A \quad A \cdot B=B \cdot A$$

⑥ 结合律：

$$(A+B)+C=A+(B+C) \quad (A \cdot B) \cdot C=A \cdot (B \cdot C)$$

⑦ 反演律：

$$\overline{A \cdot B \cdot C \cdot \dots} = \bar{A} + \bar{B} + \bar{C} + \dots \quad \overline{A+B+C+\dots} = \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \dots$$

(3) 一位加法电路——全加器。

如表 1.1 所示,进行加法运算时某一位相加需要有下列 5 个变量：

输入：被加数 X_i 、加数 Y_i 、低位进位 C_{i-1} 。

输出：本位进位 C_i 、本位全和 S_i 。

表 1.1 全加器的真值表

X_i	Y_i	C_{i-1}	C_i	S_i	X_i	Y_i	C_{i-1}	C_i	S_i
0	0	0	0	0	1	0	0	0	1
0	0	1	0	1	1	0	1	1	0
0	1	0	0	1	1	1	0	1	0
0	1	1	1	0	1	1	1	1	1

在真值表中,将函数值(C_i 或 S_i)为 1 的各参数(X_i, Y_i, C_{i-1})的“与”项相“或”,就组成了与该函数的逻辑表达式。如全加器的本位和有 4 项,全加器的本位进位也有 4 项,即有

$$\begin{aligned} \bar{S}_i &= \bar{X}_i \cdot Y_i \cdot \bar{C}_{i-1} + X_i \cdot \bar{Y}_i \cdot C_{i-1} + \bar{X}_i \cdot Y_i \cdot C_{i-1} + X_i \cdot Y_i \cdot \bar{C}_{i-1} \\ &= X_i \oplus Y_i \oplus C_{i-1} \end{aligned}$$

$$\begin{aligned} \bar{C}_i &= X_i \cdot Y_i \cdot C_{i-1} + \bar{X}_i \cdot Y_i \cdot C_{i-1} + X_i \cdot Y_i \cdot \bar{C}_{i-1} + X_i \cdot Y_i \cdot C_{i-1} \\ &= X_i \cdot Y_i + (X_i \oplus Y_i) \cdot C_{i-1} \end{aligned}$$

由这两个表达式很容易得到相应的组合逻辑电路,如图 1.4(a)所示,并且可以用图 1.4(b)所示的逻辑符号表示。

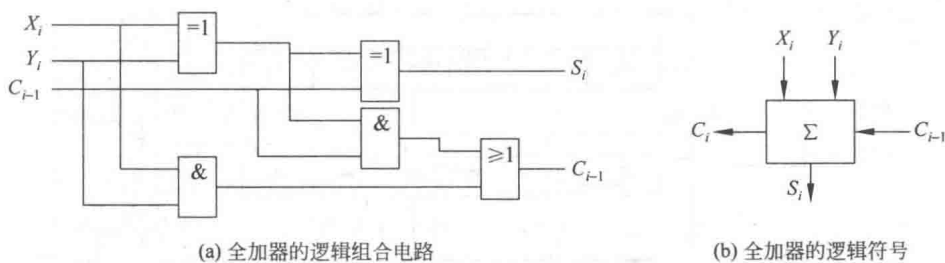


图 1.4 全加器的逻辑组合电路及其符号

实质上,全加器是完成 3 个 1 位数相加,具有两个输出端的逻辑电路。对应于输入端的不同值,将在两个输出端上输出相应的值。

(4) 串行加法电路。

串行运算加法器如图 1.5 所示。

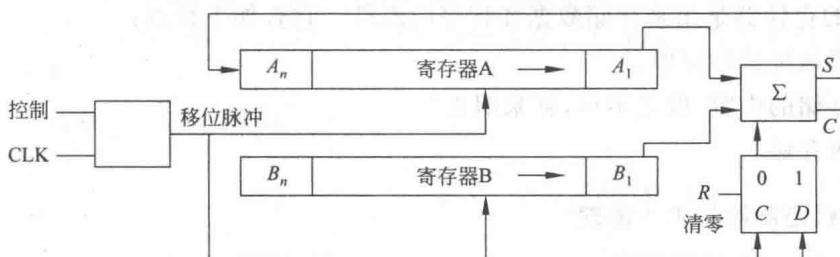


图 1.5 串行运算加法器

(5) 并行加法电路。

两个 n 位二进制数各位同时相加称为并行加法。图 1.6 为 n 位并行加法电路。它由 n 个全加器所组成。运算时由两个寄存器送来的 n 位数据,分别在 n 个全加器中按位对应相加;每个全加器得出的进位依次向高一位传送,从而得出每位的全加和。最后 1 个进位 C_n 为计算机工作进行判断提供了一个测试标志,在某些情况下(如多字节运算)还可以作为运算的一个数据。

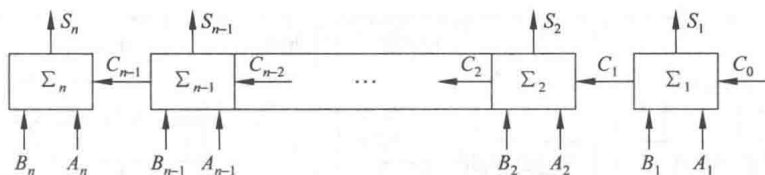


图 1.6 n 位并行加法器

(6) 加/减法电路。

在并行加法器前加一级异或门就可以组成加/减法运算器,如图 1.7 所示。

这样,当 $SUB=0$ 时,有

$$B'_i = \bar{B}_i \cdot SUB + B_i \cdot \overline{SUB} = \bar{B}_i \cdot 0 + B_i \cdot 1 = B_i$$

执行的是 $A+B$ 。

当 $SUB=1$ 时,有

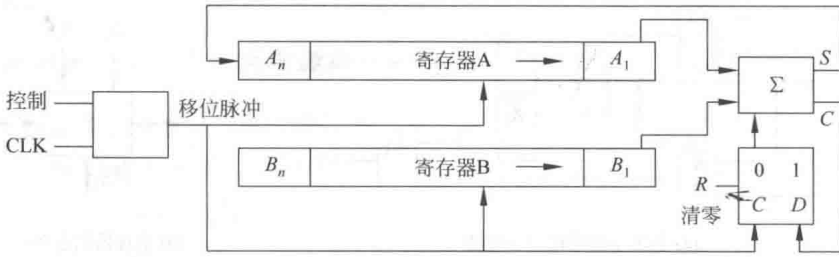


图 1.7 加/减法运算器

$$B'_i = B_i \cdot \text{SUB} + B_i \cdot \text{SUB}' = B_i \cdot 1 + B_i \cdot 0 = B_i$$

执行的是 $A - B$ 。

3. 计算机存储器的特点

计算机的存储器是用来存储数据和程序的部件。它有如下特点：

- (1) 按照地址进行存取。
- (2) 所存储的内容“取之不尽，新来旧去”。
- (3) 分级存储。

4. 计算机控制器的工作原理

1) 控制器的功能

控制器的功能如下：

- 定序。
- 定时。
- 操作控制。

2) 控制器的组成及工作过程

控制器执行一条指令的过程是“取指令—分析指令—执行指令”。如图 1.8 所示，这个过程的基本条件是先将程序(指令码和数据)存储到(内)存储器中。控制器工作时，用程序

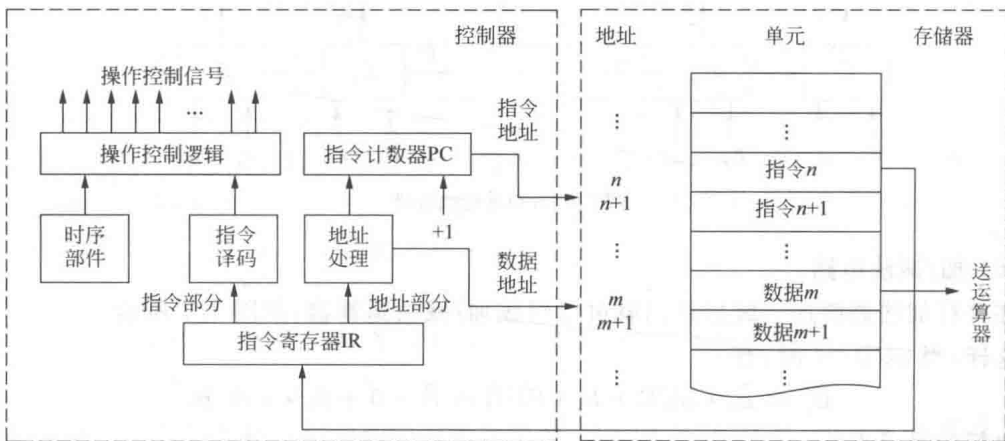


图 1.8 控制器工作原理图

计数器(也称为指令计数器)控制取指令的过程,取出的指令送入指令寄存器,然后送指令部件分析,产生控制信号,控制有关部件产生相应的动作。

5. 计算机工作的时序控制

1) 指令周期

指令周期也称取出-执行周期(fetch-and-execute cycle),指 CPU 从主存中读取一条指令到指令执行结束的时间,或者说,指令周期是并且可以细化为由“送指令地址—指令计数器(PC)加 1—指令译码—取操作数—执行操作”等微操作组成的比较详细过程。由于每种指令的复杂程度不同,其包含的微操作内容不同,所需的指令周期的长短也不相同。

2) CPU 周期

一条指令所包含的微操作之间具有顺序依赖关系。为了正确地执行指令,还需要将指令周期进一步划分为一些子周期——CPU 工作周期(也称工作周期、CPU 周期、机器周期),把一条指令包含的微操作分配在不同的 CPU 周期中。

图 1.9 描述了一个普通指令的 CPU 周期划分情况。它包含了 3 个 CPU 周期。

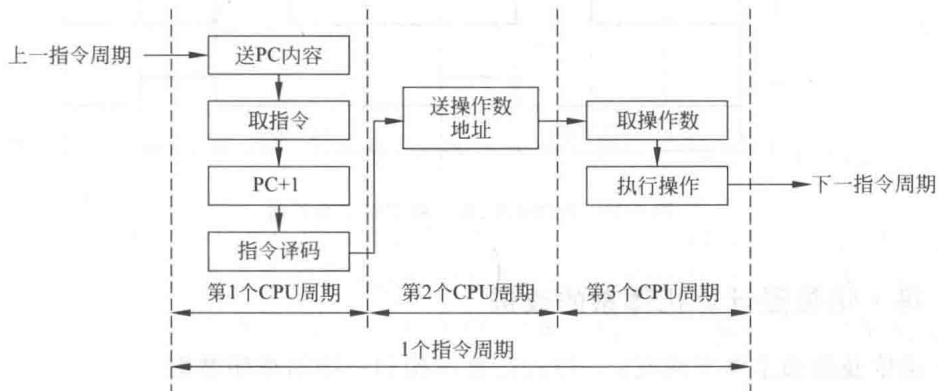


图 1.9 指令周期的 CPU 周期划分

图 1.10 描述了一个转移指令的 CPU 周期划分情况,它只包含了两个 CPU 周期:第 1 个 CPU 周期为取指周期;第 2 个 CPU 周期则是向 PC 中送一个目标地址,指出将要执行的指令的地址,使下一条要执行的指令不再是本指令的下一条指令。

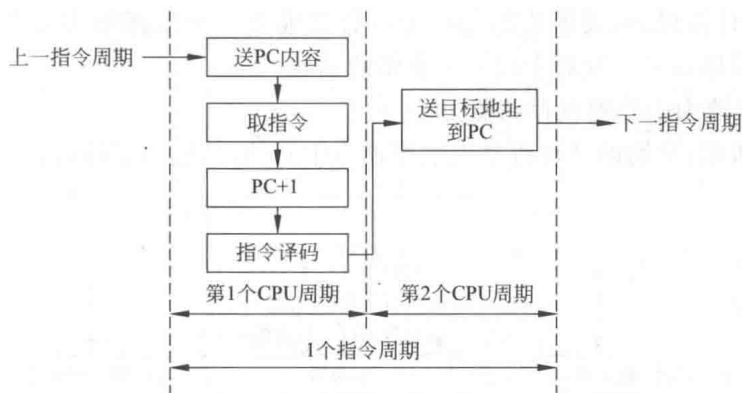


图 1.10 转移指令的 CPU 周期划分

从这两个例子可以看出,不同的指令所包含的 CPU 周期是不同的。

3) CPU 的时序信号体系

计算机是一个高速的复杂系统,为了能让各部件有条不紊地协调工作,需要为指令所包含的微操作在准确的时刻开始操作,并在这些操作信号稳定后才可以发出后续操作信号。为此,系统需要提供一套时序信号进行微操作时序的控制。这套时序信号一般由图 1.11 所示的时钟脉冲、时钟节拍信号组成。每个时钟周期形成一个节拍,一个 CPU 周期包含了多个节拍。每个微操作在规定的节拍中完成,就可以保证整个系统协调工作。

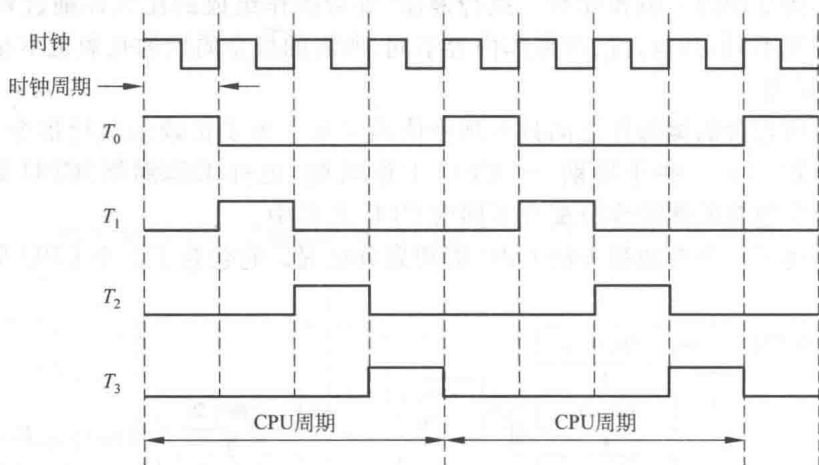


图 1.11 时钟脉冲、节拍和 CPU 周期控制

1.1.4 冯·诺依曼计算机体系的改进

冯·诺依曼瓶颈主要有两点:一维的计算结构和一维的存储结构。

几十年来,人们努力谋求突破传统诺依曼体系的局限,这些努力主要表现在两个方面:

- (1) 对冯·诺依曼计算机的改进。下面先粗略地进行一些介绍,详见第 6 章。
- (2) 跳出冯·诺依曼体系,另辟蹊径。这一部分在主教材第 7 章介绍。

1. 并行与共享

人类制造了计算机,就要使它的每一个部件都能充分地发挥潜力。于是“并行”与“共享”就成了计算机体系结构发展中的一个永恒的话题。

1) 从以运算器为中心到以存储器为中心

如图 1.12 所示,早期的计算机是以运算器为中心的,其有如下特点:

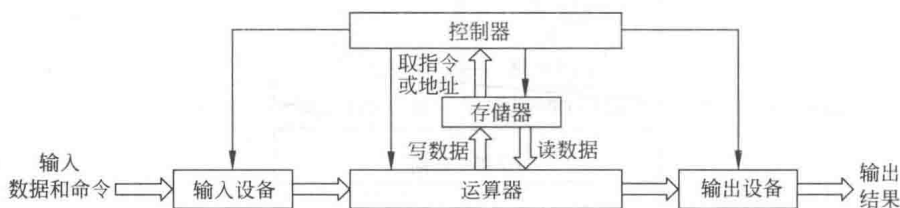


图 1.12 以运算器为中心的计算机结构

- 输入的数据要经过运算器送到存储器。
- 在程序执行过程中,运算器要不断与存储器交换数据。
- 出现中间结果和得到最终结果,要由运算器将它们送到输出设备。

所以运算器是最忙碌的部件,而其他部件都可以轮番处于空闲状态。由于不管高速部件还是低速部件都要直接与运算器一起工作,使得运算器(也就是 CPU)这样的宝贵资源无法把好钢用在刀刃上,而其他部件也不能得到充分利用。随着计算机应用的深入和外部设备的发展,内存与外存等外部设备之间的信息交换日益频繁,使得矛盾日益突出。为了改变这种状况,现在的计算机都采用了以存储器为中心的结构。如图 1.13 所示,在以存储器为中心的结构中,CPU 与输入输出设备并行工作,并共享存储器,运算器可以“集中精力”进行运算,使其效率大大提高。

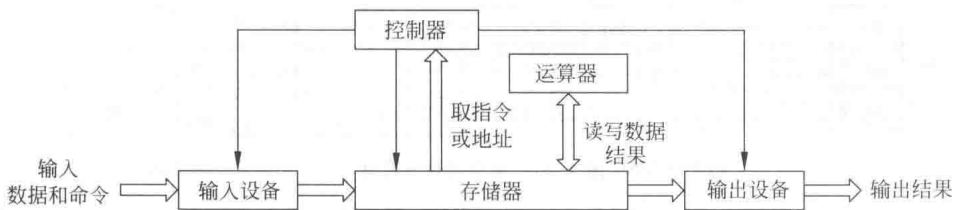


图 1.13 以存储器为中心的计算机结构

在实现以存储器为中心的结构的过程中,形成分时操作系统、中断控制技术、DAM 控制技术和各种总线技术等。

运算器和存储器都是计算机的高速设备,不管是以哪个为中心,都是高速部件为低速部件所共享。但是,从以运算器为中心到以存储器为中心,实现的是以忙设备为中心到以较闲的设备为中心,从总体上均衡了不同部件的负荷,有利于进一步挖掘高速设备的利用率。

2) 指令执行的并行与共享

如图 1.14 所示,早期的计算机在同一时间段内处理器只能进行一个指令的作业,一条指令的作业完成后,才能开始另外一条指令的作业,即指令只能一条一条地串行执行。



图 1.14 指令的串行作业

串行作业方式的优点是控制简单,由于下一条指令的地址在前一条指令解释过程的末尾形成,因此不论是由指令指针加 1 方式,还是由转移指令把地址送到指令指针形成下一条指令地址,由当前指令转入下一条指令的时序关系都是相同的。顺序作业方式的缺点是速度慢,因为当前操作完成前,下一步操作不能开始。另外机器各部件的利用率也不高,如取指周期内运算器和指令执行部件空闲。

若把 CPU 分成两个相对独立的部件:指令部件(IU)和执行部件(EU)分别负责指令的解释和执行,则如图 1.15 所示,在一条指令的执行过程同时,指令部件可以取下一条指令并进行解释,这样两个部件就可以同时操作,指令之间呈现重叠执行形式。对于计算机来说,平均执行一条指令的时间缩短。

若把 CPU 分成更多个相对独立的部件,一条指令就被解释为多个子过程,不同的部件

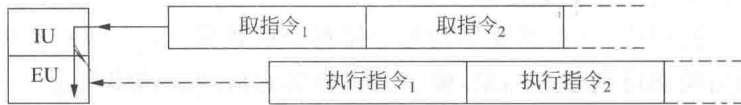


图 1.15 指令的一次重叠

将分别对微指令流中不同的子过程进行操作,于是就形成流水作业方式。流水线是 CPU 实现高速作业的关键性技术。它如同将一条生产流水线分成多个工序,各工序可以同时工作,但加工的是不同的零件。显然,工序分得越多,同时加工的零件就越多。图 1.16 为将 CPU 分为 4 个独立部分——AU、EU、IU、BU 时的指令流水作业情况。

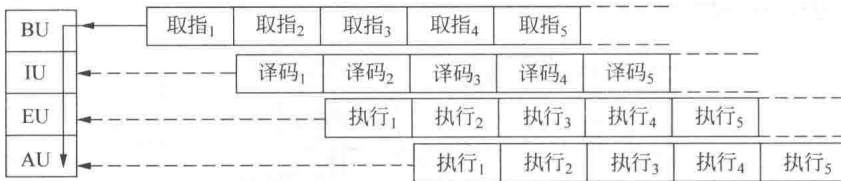


图 1.16 指令的流水作业

采用指令流水线,能使各操作部件同时对不同的指令进行加工,提高了机器的工作效率。从另一方面讲,当处理器可以分解为 m 个部件时,便可以每隔 $1/m$ 个指令周期解释一条指令,加快了程序的执行速度。注意,这里说的是“加快了程序的执行速度”,而不是“加快了指令的解释速度”,因为就一条指令而言,其解释速度并没有加快。

3) 处理器并行与共享

处理器级的并行性开发是指在一台机器中使用多个 CPU 并行地进行计算。处理器的并行性和指令级的并行性开发的主要目的是在硬件条件(集成度、速度等)的限制下,从结构上加以改进,提高系统的运行速度。不过,指令级的并行性开发是从处理器内部的结构入手,它的速度提升效果一般为 5~10 倍。要想成十倍、成百倍地提高处理速度,就要使用处理器级的并行性技术,即建造有多台处理器或多台计算机组成的计算机系统。

(1) SMP(Symmetric Multi-Processing, 对称多处理结构)是指在一个计算机上汇集了一组处理器(多 CPU)。如图 1.17 所示,这种计算机的各 CPU 之间共享内存子系统以及总线结构。在这种技术的支持下,一个服务器系统可以同时运行多个处理器,并共享内存和其他的主机资源。

(2) MPP (Massively Parallel Processing, 大规模并行处理系统)由许多松耦合处理单元(PU)组成,每个 PU 都有自己私有的资源,如总线、内存、硬盘等。在每个单元内都有操作系统和管理数据库的实例副本。这种结构最大的特点在于不共享资源。

(3) CMP(Chip MultiProcessors, 单芯片多处理器,简称多核 CPU),它可以将大规模并行处理器中的 SMP(对称多处理器)集成到同一芯片内,各个处理器并行执行不同的进程。

2. 哈佛结构

哈佛结构是针对冯·诺依曼结构提出的一种计算机体系结构,其基本特点是将程序指令和数据分开存储,即程序存储器和数据存储器是两个独立的存储器,每个存储器独立编