

高等学校教材
专科适用

单片机原理及其应用

沈阳电力高等专科学校 高天元 主编



高等学校教材

//////////////////// 专 科 适 用 //////////////////////

单片机原理及其应用

沈阳电力高等专科学校 高天元 主编

中国电力出版社

内 容 简 介

本书为高等学校统编教材(专科适用),共十一章,系统介绍了MCS-51系列单片机的硬件结构、组成原理、指令系统、软件结构程序设计、系统扩展的设计思想和实现方法,以及常用输入和输出设备的接口技术。书中有许多应用实例可供学习,每章后面都附有启迪性的思考题和习题。

本书可作为大专院校工科相关专业的教材,也可供从事微型计算机应用工作的工程技术人员参考。

图书在版编目(CIP)数据

单片机原理及其应用/高天元主编.-北京:中国电力出版社,1998

高等学校教材

ISBN 7-80125-588-7

I. 单… II. 高… III. 单片微型计算机-高等学校-教材 IV. TP368.1

中国版本图书馆CIP数据核字(97)第29613号

中国电力出版社出版

(北京三里河路6号 邮政编码100044)

三河市实验小学印刷厂印刷

新华书店北京发行所发行·各地新华书店经售

*

1998年5月第一版 1998年5月北京第一次印刷

787毫米×1092毫米 16开本 18.25印张 415千字

印数 0001—4070册 定价 18.00元

版 权 专 有 翻 印 必 究

(本书如有印装质量问题,我社发行部负责退换)

前 言

由于单片微型计算机具有体积小、价格便宜、功能多，抗干扰能力强、性能稳定、可靠性高等优点，因而在自动控制、智能化仪器仪表、通信、家用电器等方面得到了广泛的应用，同时也渗透到其它各个科技领域。在改造传统产业与工艺方法上，在产品升级换代上，单片微型计算机的应用引起了各行各业的极大关注。单片微型计算机应用技术作为一项新型的工程应用技术得到了迅速地发展。

本书以 MCS—51 系列单片机为主线，阐述基本概念、系统结构、指令系统、软件程序设计等基本原理和方法，从应用的角度，介绍了应用系统扩展的设计思想和实现方法，以及与常用输入输出设备的接口技术。本书列举的一些实例有启发性。本书可作为大专院校工科相关专业的教材，也可供广大科技人员从事单片机应用工作的参考。本书是一本结构比较完整、系统、实用性强的单片机参考书。

本书由高天元、胡汉才编写，由高天元任主编。

本书共十一章，第四、五、六、七、八、九章由胡汉才执笔，其余各章由高天元执笔。

本书由侯紫达担任主审，他仔细审阅了全部书稿，对本书提出了许多宝贵意见。在此表示衷心地感谢。

由于水平所限，时间仓促，本书内容难免有错误和不妥之处，敬请读者批评指正。

编 者

1995 年 9 月

目 录

前 言

第一章 计算机基础知识	1
第一节 计算机概述	1
第二节 微型计算机的发展概况	1
第三节 微型计算机的基本结构	2
第四节 微型计算机系统	9
第五节 微型计算机主要技术指标	13
第六节 计算机中的数和编码系统	14
复习思考题	21
第二章 单片微型计算机	23
第一节 概述	23
第二节 单片机的特点和发展	25
第三节 MCS—51 系列单片机的硬件结构	26
复习思考题	44
第三章 MCS—51 系列单片机指令系统	45
第一节 概述	45
第二节 寻址方式	47
第三节 MCS—51 系列单片机指令系统	49
第四节 指令练习	64
复习思考题	66
第四章 汇编语言程序设计	68
第一节 简单和分支程序设计	68
第二节 循环和查表程序设计	75
第三节 子程序和运算程序设计	83
复习思考题	92
第五章 半导体存储器	94
第一节 概述	94
第二节 只读存储器 ROM	96
第三节 随机存取存储器 RAM	101
第四节 MCS—51 系列单片机和外部存储器的连接	105
复习思考题	111
第六章 MCS—51 系列单片机中断系统	113
第一节 概述	113
第二节 MCS—51 系列单片机的中断系统	115

第三节 MCS—51 系列单片机外部中断源的扩展	123
复习思考题	132
第七章 并行 I/O 接口	134
第一节 概述	134
第二节 MCS—51 系列单片机内部并行 I/O 接口的使用	139
第三节 并行 I/O 端口的扩展	142
第四节 MCS—51 系列单片机与 LED、键盘的接口	153
第五节 MCS—51 系列单片机内部定时/计数器	164
复习思考题	171
第八章 D/A 和 A/D 转换器及接口电路	173
第一节 D/A 转换器	173
第二节 MCS—51 系列单片机和 D/A 转换器的接口	177
第三节 A/D 转换器	184
第四节 MCS—51 系列单片机和 A/D 转换器的接口	189
复习思考题	194
第九章 总线及其接口	197
第一节 概述	197
第二节 板级总线	199
第三节 通信总线	202
复习思考题	212
第十章 串行接口	213
第一节 数据串行传输的基本概念	213
第二节 串行接口的组成和特性	217
第三节 串行接口的工作方式	218
第四节 波特率	225
第五节 串行口多机通信	228
第六节 串行口的编程和应用	229
第七节 8251A 芯片及其串行通信接口电路	236
复习思考题	243
第十一章 单片机应用系统设计	244
第一节 单片机应用系统的研制过程	244
第二节 智能仪器仪表的设计	248
第三节 数据采集和处理系统的设计基础	257
第四节 单片机巡回检测系统硬件设计	265
第五节 单片机巡回检测系统软件设计	267
第六节 单片机应用系统的抗干扰设计	275
复习思考题	279
附表 A 常用字符与 ASC II 代码对照表	280
附表 B MCS—51 系列单片机指令表	281
参考文献	285

第一章 计算机基础知识

第一节 计算机概述

电子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装置。根据它所处理的信息是数字量或模拟量,电子计算机又可分为三类:即数字计算机、模拟计算机和混合计算机。由于目前数字计算机的迅速发展和广泛应用,因此,通常讲的计算机是指数字电子计算机。

1946年美国宾夕法尼亚大学 J·W·Mauchley 和 J·P·Eckert 研制成世界上第一台电子数字计算机 ENIAC (Electronic Numerical Integrator And Calculator)。当时它的字长 12 位,使用 18800 只电子管,1500 多个继电器,耗电 150kW,占地面积 150m²,质量达 30t,每秒钟能完成 5000 次加法运算。

ENIAC 开创了人类信息革命的新时代。从此,计算机作为一个崭新的技术科学领域得到了空前的发展。从三十多年发展史来看,可分为四代。第一代(1946~1958)是电子管数字计算机,代表机种是 ENIAC 机;第二代(1958~1964)是晶体管数字计算机,美国研制的第一台晶体管计算机是 TRANSAC S—1000 型机;第三代(1964~1971)是中小规模集成电路数字计算机,代表机型是美国 IBM360 系列机;第四代(1971 年以后)是大规模集成电路数字计算机,代表机型是美国 IBM—370 系列机。目前世界上美国、日本等许多国家正在研制以超大规模集成电路和人工智能为主要特征的第五代计算机。

当前计算机的发展趋势是微型化、巨型化、网络化和智能模拟化。计算机已从科学家使用的时代发展到大众普及,微型计算机已经开始迈入千家万户。计算机已经被广泛应用到工业、农业、交通运输、商业、国防、文教、卫生、通信、企事业管理以及人们日常生活等各个领域。并且日益显示出强大的生命力,发挥着巨大的作用。

展望未来,计算机科学将会有许多新的突破。可以预测未来的计算机将是半导体技术、超导技术、激光技术、电子仿生技术等相结合的产物。新一代的光学计算机,超导计算机和人工智能计算机的诞生将为时不远,它必将把计算机科学的发展推向一个崭新的阶段。

第二节 微型计算机的发展概况

在琳琅满目的计算机大家族中,微型计算机一枝独秀,得到了超常规的发展。随着微型机性能的不断提高,价格逐渐下降,应用日益普及,必将成为人们工作和日常生活中不可缺少的工具。

随着大规模集成电路技术和计算机技术的迅速发展,70 年代初诞生了一代新型电子计算机——微型计算机。它采用大规模集成电路技术把计算机的中央处理单元 CPU (Central

Processing Unit)——计算机的运算器和控制器集成在一块芯片上,或称微处理器 MP (Micro Processor) 上。微处理器配上由大规模集成电路制作的存储器,输入/输出接口电路及系统总线便组成了微型计算机 (Microcomputer)。有的微型计算机把微处理器、存储器和输入/输出接口电路都集成在一块芯片上,被称之为单片微型计算机或简称单片机。

一般所讲的微型计算机,实际上指的是微型计算机系统。它是以微型计算机为中心,并配以相应硬件和软件组合而成。硬件指的是外部设备、电源和辅助电路等,这些均是看得见摸得到的实体。软件是指指挥管理微型计算机工作的各种程序。

微型计算机系统的发展是以微处理器的发展为表征的。微型计算机的换代,通常是按其微处理器的字长、位数和功能来划分的。自 1971 年微型计算机问世以来,发展非常迅速,至今已经历了四代的演变,现已进入第五代。

第一代 (1971~1973 年) 是 4 位、低档的 8 位微处理器。其典型产品为 Intel 4004 和 Intel 8008。第一代产品的集成度约在 2000 晶体管/片左右,时钟频率约为 1MHz,采用 PMOS 工艺。软件采用机器语言及简单的汇编语言。

第二代 (1973~1975 年) 是中档的 8 位微处理器。其典型产品是 Intel 8080 和 M6800 (Motorola 公司生产)。集成度约为 5000 管/片,时钟频率为 2MHz,采用 NMOS 工艺。软件一般采用汇编语言和高级语言。

第三代 (1975~1977 年) 是高档的 8 位微处理器。其典型产品为 Intel 8085, M6809 和 Z80 (Zilog 公司)。集成度约为 10000 管/片左右,时钟频率为 2.5~5MHz。软件配有操作系统。

第四代 (1978~1981 年) 是 16 位微处理器。其典型产品是 Intel 8086, M68000 和 Z8000。集成度约为 30000 管/片,时钟频率可达 5MHz 以上。软件配有较丰富的操作系统。

第五代 (1981~今) 是 32 位微处理器。其典型产品是 Intel 80386、M68020。集成度约为 20 万管/片,时钟频率可达 16MHz 以上。软件配有丰富的操作系统,高级语言的软件可以硬化。现在 Intel 80486、奔腾 80586、M68040 等产品已在应用,集成度已达 100 万管/片,时钟频率可达 25MHz 以上。

第三节 微型计算机的基本结构

微型计算机的基本结构如图 1-1 所示。它是由中央处理单元、存储器、接口电路和外部设备组成。它们之间靠总线来连接。总线可分为地址总线、控制总线、数据总线和输入/输出总线。

为了简化问题,先暂不考虑接口电路和外部设备,而是从一个仅由 CPU 和存储器构成的最简单的模型计算机来分析基本结构和基本工作原理。

模型计算机的基本结构如图 1-2 所示。它是由 CPU、存储器等组成。

一、中央处理单元 CPU

中央处理单元由运算器和控制器组成。

(一) 运算器

运算器是在控制器控制下,对二进制数进行算术和逻辑运算以及一些基本操作的装置。运算器由算术逻辑单元 ALU、累加器 A、通用寄存器 H、标志寄存器 F 以及其它逻辑电路所组成。

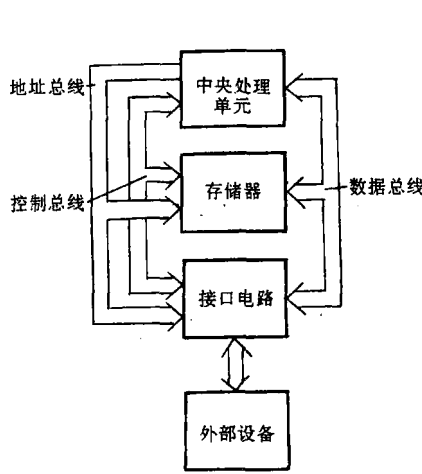


图 1-1 微型计算机基本结构

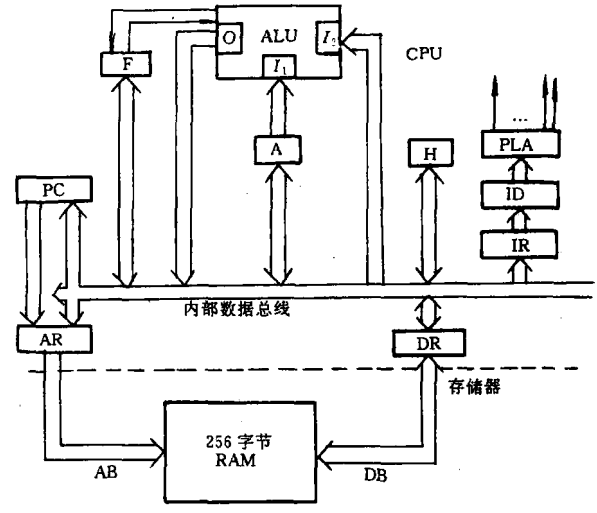


图 1-2 模型计算机基本结构

1. 算术逻辑单元 ALU (Arithmetic Logic Unit)

ALU 是由加法器和其它逻辑电路（如移位器、计数器、控制门）等组成。在指令译码后产生的控制信号的控制下，它能完成各种算术和逻辑运算。它是以累加器 A 的内容作为一个操作数，另一个操作数由内部数据总线供给，可以是通用寄存器 H 的内容，也可以是由数据寄存器供给的由内存读出的内容等，操作结果送累加器 A 或数据总线。同时将操作结果的特征状态送标志寄存器 F 保存。

2. 累加器 A 或 AC (Accumulator)

累加器是运算器的关键部件之一。它一方面是作为 ALU 的一个操作数的输入端，另一方面是存放 ALU 的运算结果。它既是操作数寄存器，又是结果寄存器。从结构上看，累加器实际上是一个有并行输入/输出能力的移位寄存器，其位数等于微型计算机数据字的字长，每一位有一个寄存器。

3. 通用寄存器 H (High-speed buffer register)

通用寄存器用来暂存参加运算的操作数、中间结果或地址。它是为高速处理数据而设置的。其数量根据不同种类的 CPU 有所不同。

4. 标志寄存器 F (Flag Register)

标志寄存器是用来保存 ALU 运算后结果的一些特征状态，这些状态可作为标志，用来判断是否控制程序转移的条件。

(二) 控制器

控制器是计算机的控制指挥中心，犹如人脑的神经中枢一样，控制计算机程序和原始数据的输入，CPU 内部信息的处理，处理结果的输出以及主机与外部设备之间的信息交换

等。计算机是遵循程序存储和程序控制的原理工作的。指令是计算机的执行操作以命令形式写出来的表述。指令可以有許多条，但是通常一条指令对应着一种基本操作。程序是为完成特定任务的一系列指令的集合。程序被预先存放在存储器中。当启动计算机工作时，计算机便进入了自动程序控制状态。控制器的任务是：逐条地取出指令、分析指令、执行指令并为取下一条指令作好准备，直至指令执行完毕。

为了完成上述功能，控制器一般具有指令部件、时序部件和微操作控制电路。

1. 指令部件

指令部件主要包括有程序计数器 PC，指令寄存器 IR 和指令译码器 ID。

(1) 程序计数器 PC (Program Counter)。程序是指令的有序集合。计算机工作时，计算机按顺序执行存放在存储器中的程序。首先要由 PC 指出所要执行程序的首地址。当第一条指令取出后，PC 的内容自动加 1，指向按顺序排列的下一条指令的地址。每当取出一条指令后，PC 就加 1。因此，PC 的内容总是要执行的下一条指令的地址。一般情况下，CPU 按顺序逐条地执行指令。只有在执行转移指令或子程序调用及返回指令（或中断）时，把转向的地址赋给 PC。

程序计数器 PC 的位数取决于微处理器所要寻址的存储空间。在列举的模型机中，存储器的存储容量是 256 个字节，因此 PC 只需 8 位。

(2) 指令寄存器 IR (Instruction Register)。IR 是用来存放从内存中读出当前要执行的指令内容。一条指令一般包括有操作码和操作数两部分。操作码表示计算机要执行什么样的操作，操作数表示要参加的操作数的本身或操作数所在的地址。

(3) 指令译码器 ID (Instruction Decode)。ID 是指令的分析部件，它是将 ID 送来的操作码经过译码后产生相应操作的控制电位。例如 8 位的操作码经过指令译码器后可以产生出 256 种操作控制电位，其中每一种控制电位对应一种特定操作。

2. 时序部件

计算机的工作就是在程序的控制下，自动地，逐条地从内存中取出指令、分析指令、执行指令。这一系列操作的顺序都需要有精确的定时安排。时序部件正是满足这一需要，用来产生所需要的定时信号。它由时钟系统（含脉冲源、启停逻辑）、时钟脉冲分配器、微操作控制等部件组成。

(1) 时钟系统。时钟系统一般含脉冲源和启停逻辑两部分。脉冲源产生具有一定频率和一定宽度的脉冲信号，为微处理器的操作提供一个基准，同步计算机内的一切操作。计算机一旦电源接通，脉冲源即以固有的频率连续发出矩形主脉冲。相邻两个脉冲前沿的时间间隔称为一个时钟周期或 T 时态。不同的微型机，由脉冲源产生的主频也不相同。现在高档的微型机的主频在几百兆赫兹以上。一般地讲，主频越高，计算机的工作速度越快。

时钟启停逻辑用来控制主脉冲信号的开关，从而启动或停止计算机的工作。

(2) 时钟脉冲分配器（节拍脉冲分配器）。它用以产生节拍电位或节拍脉冲的定时信号，来控制 and 协调计算机各部分有节奏地动作。每个节拍和脉冲信号指挥机器完成一个微操作。

(3) 微操作控制部件。它是根据指令来产生计算机各部件所需要的控制信号。这些控制信号是由指令译码器的输出电位，节拍脉冲发生器产生的节拍电位、节拍脉冲以及外部

的状态信号等进行组合而产生的。它按一定的时间顺序定时发出一系列微操作控制信号，以完成指令所规定的全部操作。

微操作控制部件一般采用可编程逻辑阵列（PLA）控制方式来设计。即通过程序设计来执行特定逻辑功能的组合逻辑结构。使微操作按一定的顺序执行。

二、存储器

存储器是存放数据和程序的装置。是由按矩阵排列的存储单元所组成的。每个存储单元存储二进制信息的长度（位数），称为字长。通常将 8 位二进制数位称 1 个字节，内存容量可用字节数表示。每个存储单元都有一个确定的地址编码来标识。因此，存储器一个显著的特点是按地址进行存取数据。一般把从存储器取出数据，称为读操作，读出数据（不破坏原存信息）；把存入存储器操作叫写操作，写入数据。

存储器根据工作方式的不同可分为随机存储器 RAM（Random Access Memory）和只读存储器 ROM（Read Only Memory）。RAM 可以随机地存入或取出信息，ROM 对存入的信息只能读出，不能进行写入操作。

一般把设在主机内部的存储器简称内存。通常都采用半导体存储器（第五章将详细介绍）。

存储器的内部结构通常由存储单元阵列、地址寄存器、地址译码器和数据缓冲寄存器等四个部分组成。如图 1-3 所示。

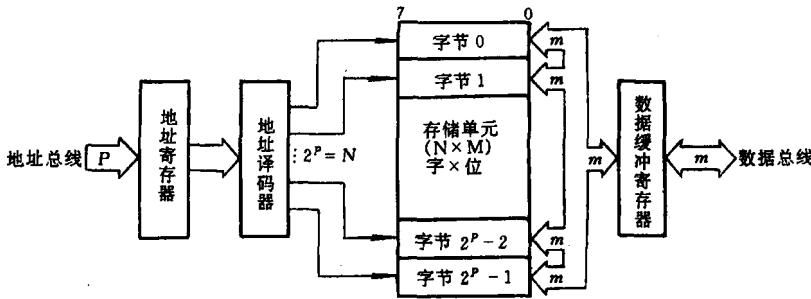


图 1-3 存储器结构

从图 1-3 中可以看出，当访问某一指定的存储单元时，须将该存储单元的地址送入地址寄存器。再经地址译码器译码，从而选中指定的存储单元。图中 P 是指地址总线的位数， N 是经 2^P 地址译码所能寻址的存储器的单元总数， m 是数据总线的位数（等于存储单元的字长）。

对存储器进行读操作时，例如将地址为 88H 的存储单元的内容读入 CPU 中。首先，由 CPU 的程序计数器 PC 将其所指定单元的地址 88H，经地址总线送到内存地址寄存器，接着由地址译码器译码，找到 88H 单元，然后 CPU 发出读命令，于是 88H 单元的内容被读出，经数据总线送到数据缓冲寄存器中。

当对存储器进行写操作时，例如将数据缓冲寄存器的内容写入指定的地址为 56H 的存储单元。首先，由 CPU 的程序计数器 PC 将其所指定单元的地址 56H 经地址总线送到内存地址寄存器，接着由地址译码器译码，找到 56H 单元，然后 CPU 发出写命令，于是数据缓

冲寄存器的内容，经数据总线写入到 56H 单元。

三、简单程序的执行过程

下面通过一个简单的加法例子来说明程序执行的操作过程。

求 8 与 18 的和。计算机要执行这样的操作就要为它编排程序。按照单片机的指令系统，可用下面二条指令完成。

```
MOV  A, #08H           ; 机器码是 74H, 08H
ADD  A, #18H           ; 机器码是 24H, 18H
```

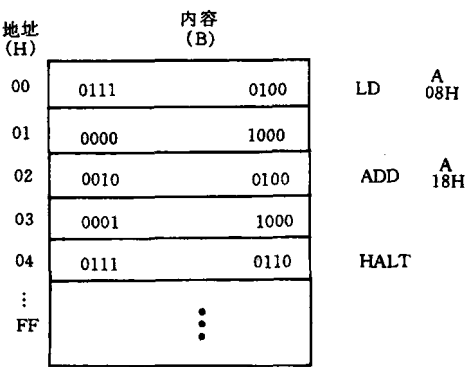
第一条指令为 01110100；操作码（取数操作）

00001000；操作数（08H）

这是一条二字节指令，把指令第二字节的操作数 08H 送累加器 A。

第二条指令 00100100；操作码（加法操作）

00011000；操作数（18H）



这也是一条二字节指令，是将累加器 A 中的内容与指令中第二字节的操作数 18H 相加，其结果放在累加器 A 中。

完成这次加法运算需二条指令，占 4 个字节。程序应放在存储器中，若程序存放在以 00H 开始的存储单元中，加上停机指令，共占 5 个存储单元，如图 1-4 所示。

(一)第一条指令的取指令过程和执行过程

1. 取第一条指令的操作过程

在执行程序时，要先给程序计数器 PC 赋以第一条指令所在的地址，然后启动程序就进入第一条指令的取指令阶段，如图 1-5 所示，操作过程如下：

- (1) PC 的内容 00H 送至地址寄存器 AR。
- (2) 当 PC 的内容送入 AR 之后，PC 的内容自动加 1 变为 01H。
- (3) AR 将地址信号 00H 通过地址总线 AB 送至存储器，经地址译码器译码，选中 00H 号单元。
- (4) CPU 发出读命令。
- (5) 所选中的 00H 号单元的内容 74H 送至数据总线 DB 上。
- (6) 经数据总线将所读出的内容送至数据寄存器 DR 中。
- (7) 因是在取指阶段，取出的必为指令，所以 DR 将其送至指令寄存器 IR 中，然后经过指令译码器译码后，发出执行这条指令所需要的各种控制命令。

2. 执行第一条指令的操作过程

取完第一条指令后，转入执行第一条指令的阶段。CPU 经过对操作码的译码后，判定这是一条将操作数送至累加器 A 的指令，操作数放在指令的第二字节。因此，执行第一条指令，必须取出指令第二字节中的操作数。如图 1-6 所示，操作过程如下：

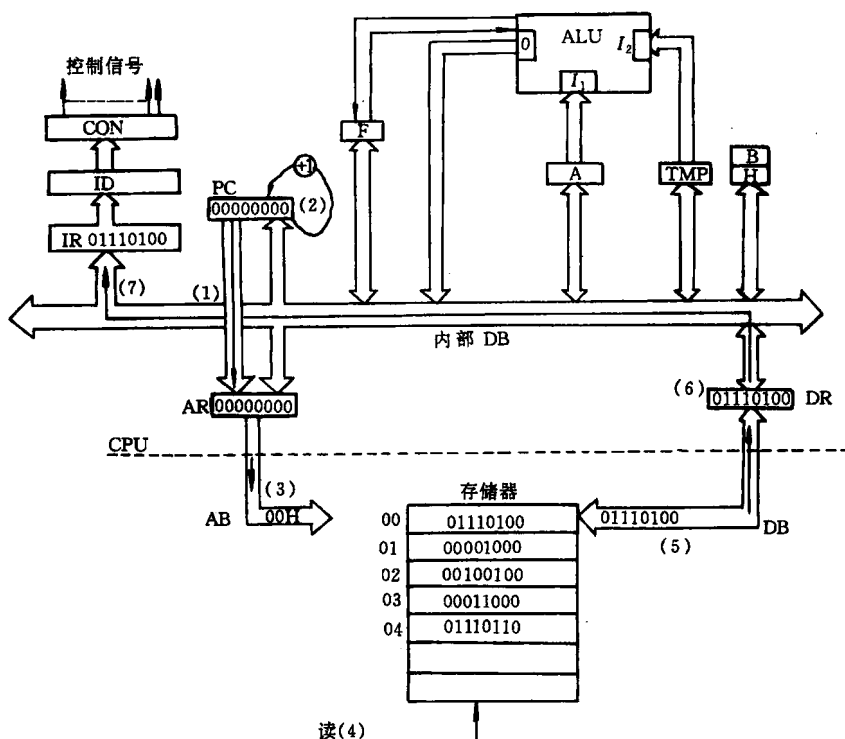


图 1-5 取第一条指令的操作示意图

- (1) 把 PC 的内容 01H 送至地址寄存器 AR。
- (2) 当 PC 的内容送入 AR 后, PC 的内容自动加 1, 变为 02H。
- (3) AR 把地址 01H 通过地址总线 AB 送至存储器, 经过地址译码器译码选中 01H 单元。
- (4) CPU 发出读命令。
- (5) 把所选中的 01H 单元的内容 08H 读到数据总线 DB 上。
- (6) 经数据总线 DB, 把读出的内容送至数据寄存器 DR 中。
- (7) 因已知读出的是操作数, 并且指令要求把它送到累加器 A 中, 故由数据寄存器 DR 通过内部数据总线将操作数送至累加器 A。

至此, 第一条指令执行完毕, 进入第二条指令的取指阶段。

(二) 第二条指令的取指令过程和执行过程

1. 取第二条指令的操作过程

- (1) 将 PC 的内容 02H 送至地址寄存器 AR。
- (2) 当 PC 的内容送到 AR 后, PC 内容自动加 1, 变为 03H。
- (3) 地址寄存器 AR 通过地址总线把地址信号 02H 送至存储器, 经地址译码后, 选中 02H 单元。
- (4) CPU 发出读命令。
- (5) 所选中的 02H 单元的内容 24H, 读至数据总线 DB 上。

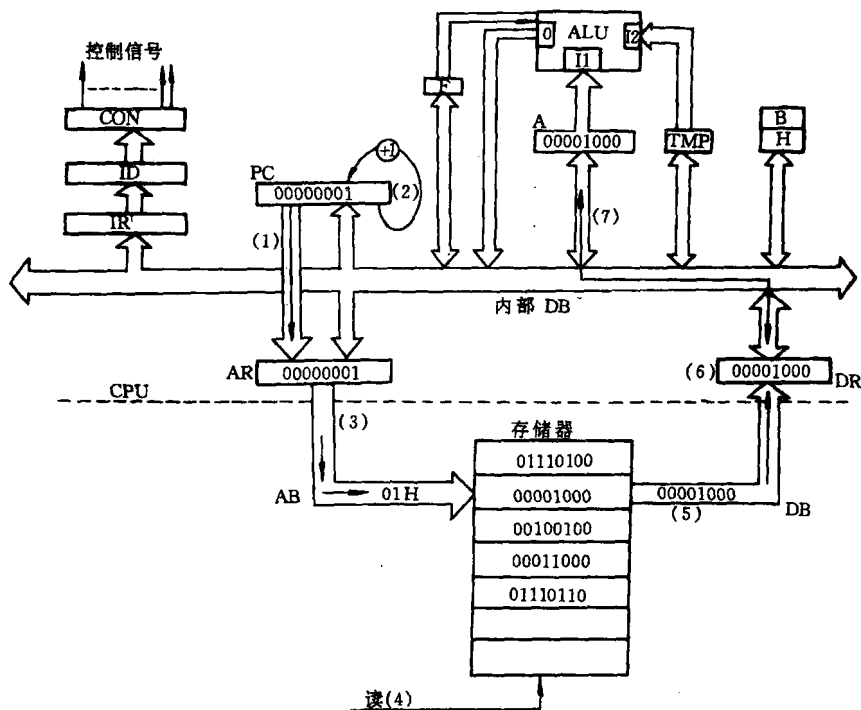


图 1-6 取立即数的操作示意图

(6) 经数据总线 DB 将读出的内容送至数据寄存器 DR 中。

(7) 因是在取指阶段，所读出的是指令，DR 将其内容送至指令寄存器 IR 中，再经过指令译码器 ID 译码，发出执行该条指令的各种控制命令。

2. 执行第二条指令的操作过程

CPU 经过对指令的译码后，判定这是一条加法指令，以 A 中的内容为一操作数，另一个操作数在指令的第二字节中。因此，执行第二条指令，必须取出指令的第二字节的操作数。

取第二字节的操作数与执行加法的操作过程

(1) 将 PC 的内容 03H 送至地址寄存器 AR。

(2) 当 PC 的内容送至 AR 后，PC 内容自动加 1，变为 04H。

(3) AR 通过地址总线把地址号 03H 送至存储器，经过地址译码，选中 03H 单元。

(4) CPU 发出读命令。

(5) 所选中 03H 单元的内容 18H 读至数据总线 DB 上。

(6) 经数据总线 DB 将 18H 送至数据寄存器 DR 中。

(7) 因由指令译码已知读出的是操作数，并且要与累加器 A 中的内容相加，所以数据由 DR 经 DB 送至暂存寄存器 TMP 中。

(8) 累加器 A 中的内容进入算术逻辑单元 ALU 的 I1 端，TMP 中的内容进入 ALU 的 I2 端，进行加法操作。

(9) 将相加的结果，由 ALU 输出至累加器中。

至此完成了两个数进行加法的运算。

通过以上一个简化的模型计算机执行程序的例子，可以看出：计算机执行程序的过程，是周而复始地取指令、分析指令和执行指令的过程，直至完成程序的全部指令，实现预计要求的任务为止。

通过上例的分析，可以把计算机执行程序的过程归纳如图 1-7 所示。

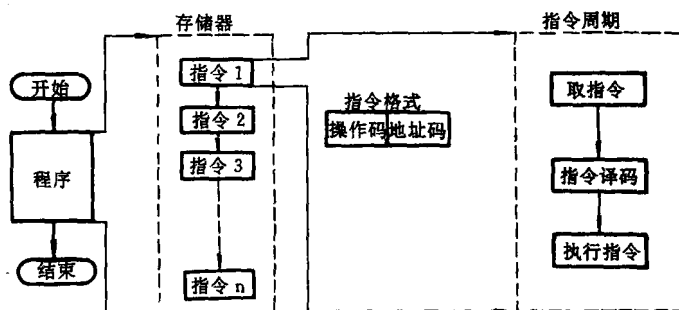


图 1-7 计算机执行程序示意图

四、计算机的解题过程

(1) 编制解题程序，分配存储单元。根据解题需要先绘制程序框图（解题流程图），因上例十分简单，故省略这一步。接着，要分配存储单元。通常，程序是顺序执行的，程序中的指令也按顺序存放在某段连续的存储单元（即程序存放区）中。而操作的数据则存放在另一段存储单元（即数据存储区）中。然后编制程序。

(2) 程序存储。计算机之所以能自动工作，就是因为“程序存储”的特点。把编制好的解题程序和数据预先通过外部设备，如键盘输入到计算机的内存储器中。

(3) 执行程序。程序输入到存储器后，只要给出程序所在的起始地址，即给程序计数器赋初值，然后发一个启动命令，CPU 就能按顺序逐条地从内存储器中取出指令加以执行。由于程序计数器 PC 有自动加 1 功能，这就保证了指令的顺序执行。只有当程序中遇到转移指令、调用子程序或中断时，PC 才能转到相应的程序地址执行。

计算机在程序控制下，自动完成计算，送出结果。这就是迄今为止，计算机共同遵循的程序存储和程序控制原理，也称冯·诺依曼型机（John Von Neumann）。

第四节 微型计算机系统

微型计算机系统同大中小型计算机系统一样，也是由硬件和软件两大部分组成。但它的硬件和软件更是密不可分的。目前把软件功能固化于硬件中（固件），软件硬化的微型计算机系统已成为一个重要的发展方向。

一、硬件

计算机具有许多相似的人脑的工作特点，俗称“电脑”。为了说明计算机的硬件组成和工作原理，先看一下人用算盘、纸、笔来计算 $(1946+93\times 101)$ 的过程。

1. 人运算的大致步骤

(1) 根据给定题目, 确定计算步骤和方法, 并把公式、解题步骤和原始数据写在纸上。

(2) 按先乘后加的原则, 在算盘上进行计算中间结果 $93 \times 101 = 9331$ 的值, 然后在算盘上做加法, 计算最后结果 $1946 + 9331 = 11277$ 。

(3) 把最终的结果 11277 写在纸上, 计算完毕。

2. 用电子计算机计算的必备条件

若用电子计算机模拟上述解题过程, 计算机必须具备下列条件:

(1) 为了进行各种数据计算, 计算机内必须有一个相当于算盘的运算器。

(2) 为了保存和记录原始数据、解题程序和运算的中间结果, 计算机内必须有一个容量足够大的存储器, 这相当于手工计算时用的纸。

(3) 必须有一个按照解题程序指挥和控制各部件协同工作的控制器, 这相当于手工计算时人的大脑。

(4) 必须具备将原始数据和程序送入机器内部的输入设备和给出计算结果的输出设备。这相当于手工计算时的眼、手、笔。

如前所述、硬件包括运算器、控制器、存储器和输入输出设备, 它是计算机赖以工作的物质基础。其中把运算器、控制器和存储器合称为主机; 运算器和控制器是计算机硬件的核心, 合称为中央处理器 CPU; 除主机以外的各种输入输出设备统称为外部设备。

图 1-8 给出了计算机硬件组成的原理框图。计算机硬件系统除给出的五个基本组成部件外, 还必须有电源装置、控制台或控制面板等。

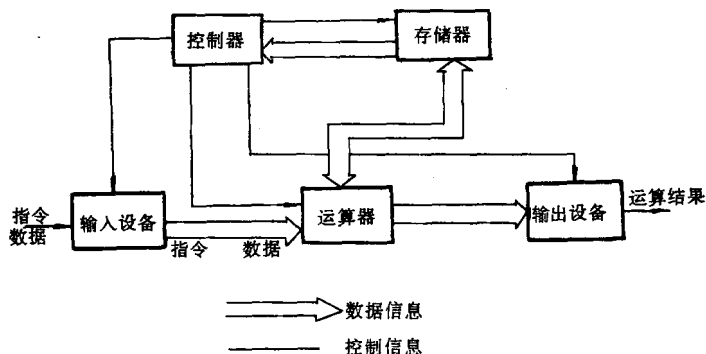


图 1-8 计算机硬件组成原理框图

微型计算机系统硬件组成框图, 如图 1-9 所示。它是由微处理器 (CPU), 存储器和输入/输出接口电路 (I/O 接口电路) 组成, 通过 I/O 接口电路再与外部设备 (I/O 设备) 相连接。它们相互之间通过三组总线——地址总线, 数据总线和控制总线来连接。

存储器有只读存储器 ROM (Read Only Memory) 和随机存储器 RAM (Random Access Memory) 两大类。只读存储器是只能读出数据而不能写入数据, 存储的数据具有不可更改性。随机存储器则可以随机进行读和写。

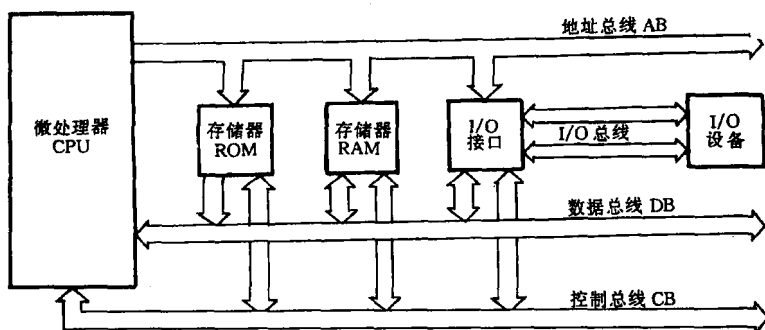


图 1-9 微型计算机系统硬件组成框图

二、软件

软件是构成微型计算机系统必不可缺的部分。要计算机正确的运行以解决各种问题，必须给它编制各种程序。为了运行、管理和维护计算机所编制的各种程序被称为软件。

一般地讲软件包括各种程序设计语言、系统软件、支撑软件和应用软件等。

1. 程序设计语言

程序设计语言是指用来编写程序所用的语言。人与人之间的交流靠自然语言（汉语、英语等），人同计算机之间的信息交流同样靠“语言”。程序设计语言通常可分为机器语言、汇编语言和高级语言三类。

(1) 机器语言。机器语言是一种用二进制代码“0”或“1”的形式来表达的，是能够被计算机识别并直接执行的语言。机器语言对于不同类型的计算机来讲，一般是不同的。这是由计算机的指令系统决定的。用这种二进制代码形式表示的机器语言阅读性差、难记忆、难查错，编写程序直观性差，易出错，而且烦琐费时。

(2) 汇编语言（初级语言）。汇编语言是一种用助记符来表示的面向机器的程序设计语言。这种语言比较直观，而且容易记忆和检查。用这种语言编写出的程序叫汇编语言源程序。源程序不能为计算机直接执行，要经过汇编程序的加工和翻译，变成用机器语言表示的目标程序才能执行。

由于汇编语言的语句与机器指令是一一对应的，对于不同的计算机也是不同的。因此，通用性差，编写仍比较烦琐费时。但用汇编语言编写的源程序，可以直接操作到 CPU 内部和 RAM 单元，同时在程序执行时间和空间利用上效率是比较高的。

(3) 高级语言。高级语言是面向过程的能独立于机器的通用程序设计语言。它不依赖具体计算机的结构和指令系统，是一种接近人的自然语言和常用数学表达的语言。

用高级语言编写的程序必须通过编译程序（或解释程序）“翻译”成用机器指令表示的目标程序才能执行。这样就需要有各种编译程序，例如对 FORTRAN, PASCAL, C 语言等有不同的编译程序。

高级语言有上百种，国际上比较流行的有 BASIC, FORTRAN, COBOL, PASCAL, PL/1, LISP, FORTH 以及 C 语言等。

2. 系统软件