

intel单片机应用技术

● 陈建铎 编

● 陕西科学技术出版社

Intel 单片机应用技术

陈建铎 编

陕西科学技术出版社

Intel 单片机应用技术

陈建铎 编

陕西科学技术出版社出版发行
(西安北大街 131 号)

西安电子科技大学印制厂印刷

787×1092 毫米 16 开本 44 印张 106 万字

1991 年 6 月第 1 版 1991 年 6 月第 1 次印刷

印数: 1-5,000

ISBN 7-5369-0854-7 / TP · 33

定价: 19.00 元

前 言

近年来,微型计算机发展很快,特别是 Intel 公司于 1976 年研制成第一批 8 位单片微型计算机之后,单片微型计算机的发展尤其迅速,仅在短短的几年之内,就形成了多种系列产品,其中包括 8 位 MCS-48/51、16 位 MCS-96 和 RUPI-44 等系列。

单片微型计算机简称为单片机,也称为微控制器。它将组成计算机的中央处理器 CPU、存贮器 RAM/ROM、定时器/计数器及各种多功能 I/O 接口电路制作在一块集成电路芯片中,构成一个完整的微型计算机。在有的芯片中,还设置有 A/D 转换器、模拟量输出电路和全双工串行接收/发送器,从而使单片机具有多种专用功能。由于单片机是将组成计算机的各组成部件和 I/O 接口电路制作在一块集成电路芯片中,因而结构紧凑,体积小,功耗低,控制功能强,工作可靠稳定。目前已广泛用于生产过程的自动检测、数据采集与处理、工业实时控制、机器人、通讯、导航、医疗卫生、智能仪器仪表、家用电器以及旧设备改造等方面。由于单片机成本低,价格便宜,可以获得较高的经济效益,很受广大用户的欢迎。

该书是为满足广大科技工作者对单片机进行开发应用编写的。全书共分为九章。第一章:概述,简单介绍了微型计算机的发展过程、基本组成原理和单片机的特点与类型。第二章:MCS-48 单片机的组成原理与程序设计,介绍了 MCS-48 系列单片机的基本组成、指令系统和程序设计方法。第三章:MCS-51 单片机的组成原理与程序设计,介绍了 MCS-51 系列单片机的基本组成、指令系统和程序设计方法。第四章:MCS-96 单片机的组成原理与程序设计,介绍了 MCS-96 系列单片机的基本组成、指令系统和程序设计方法。第五章:常用半导体存贮器,主要介绍了与单片机联接的常用随机存取存贮器 RAM 和可改写的只读存贮器 EPROM。第六章:单片机常用外围接口电路,主要介绍了与单片机联接的常用外围接口电路,其中包括并行 I/O 接口电路、串行 I/O 接口电路、总线驱动器、锁存器以及一些专用接口电路。第七章:单片机的开发使用,主要介绍了单片机外部存贮器的扩展、与键盘、数码显示器、微型打印机以及串行通讯接口的联接。第八章:单片机应用系统的设计,主要介绍了单片机数据采集系统、工业实时控制以及智能仪器仪表的设计思想与方法,同时还介绍了单片机多机系统的设计。第九章:RUPI-44 单片通讯处理机的基本组成与应用,主要介绍了 Intel 8044 通讯处理机的基本组成原理以及在通讯方面的应用。

该书既可作为广大科技工作者学习单片机的自学参考书,又可作为单片机开发使用时的参考手册。

该书编写时得到陕西电子编辑部主编张忠智高级工程师的具体指导和帮助,西安交通大学计算机系胡正家教授对全稿进行了详细审阅,并提出许多宝贵意见。在此表示衷心的感谢。由于作者水平有限,难免存在许多缺点和错误,诚请广大读者批评指正。

编 者

1988 年 8 月

第一章 概 述

第一节 单片微型计算机的产生与发展

一、概述

电子计算机是能高速、自动、精确地进行各种数值计算、数据处理、信息存贮、逻辑分析与判断以及实时过程控制的电子机器。这种机器经过许多科学家数百年的潜心研究,得到了不断的发展和完善。但是,真正可以称得起电子数字计算机的机器诞生的历史仅有40余年。世界上第一台电子数字计算机 ENIAC 诞生于1946年,它是在美国科学家穆奇里和爱克尔特博士的主持下,在宾夕法尼亚大学研制成功的。它的诞生,标志着人类文明进入了一个新的历史阶段。40年来的历史表明,在人类历史上,电子计算机的发明与蒸气机的发明具有同等重要的意义。

电子计算机的产生与发展,使过去许多因为计算复杂而长期被搁置的数学难题得以解决,使许多纷繁杂乱的琐事被料理井然,从而有力地推动着科学技术的发展,推动着工农业生产、交通运输、空间技术以及医疗卫生等事业的发展。特别是近十多年来,微处理器与微型计算机的发展,已深入到科研、国防、工业、农业、交通运输、宇航、医疗卫生、企业管理以及人类社会生活的各个领域,其前景无法估量。总之,可以说电子计算机的产生与发展,有力地推动着人类社会的发展,其功绩无法尽述。

在电子工业和计算机行业中,电子计算机的发展推动了微电子技术和半导体工业的发展,使得半导体集成电路得到迅速的发展,近十多年来,产生了大规模集成电路和超大规模集成电路。随着大规模集成电路和超大规模集成电路的发展,人们自然而然地想到将计算机的中央处理器 CPU 制作在一块大规模集成电路中,这样就产生了微处理器。随着微处理器的不断发展和完善,人们又在其中制作了存贮器、I/O 接口电路以及定时器/计数器等部件,这样就构成了本书所要介绍的单片微型计算机。

单片微型计算机简称为单片机,它是把组成微型计算机的各功能部件:中央处理器 CPU、随机存取存贮器 RAM、只读存贮器 ROM、I/O 接口电路、定时器/计数器以及串行通讯接口等部件制作在一块集成电路芯片中,构成一个完整的微型计算机。这样的集成电路称得起超大规模集成电路。因此可以说,单片微型计算机与大规模集成电路和超大规模集成电路的发展并驾齐驱,同是电子工业中的孪生兄弟。

由于在单片机中设置有只读存贮器,可用来存放控制单片机及外围设备进行工作的程序。于是从外部来看单片机成为一个独立的控制装置,可广泛用于实时控制、数据采集与处理等方面,因此单片机又称为微控制器。

二、单片机的产生与发展

单片机的产生与发展和微处理器的产生与发展大体上同步,也经历了四个阶段。

第一阶段：1971~1974年。1971年11月美国 Intel 公司首先设计集成度为 2000 只晶体管/片的 4 位微处理器 Intel 4004，并且配以随机存取存储器 RAM、只读存储器 ROM 和移位寄存器等芯片，构成了第一台 MCS-4 微型计算机。1972 年 4 月 Intel 公司又研制成功能较强的 8 位微处理器 Intel 8008。在此期间，Fairchild 公司也研制成 F8 微处理器。这些微处理器虽说还不是单片微型计算机，但是它拉开了研制微处理器的序幕，也拉开了研制单片机的序幕。因此可以认为，单片机的研制工作也是从那时开始的。

第二阶段：1974~1978 年，中、低档 8 位单片机。1973 年 Intel 公司在研制成第二代微处理器 Intel 8080 之后，致力于高性能微处理器的研制，不断提高大规模集成电路的集成度，提高微处理器的性能，减少构成微型计算机的芯片数。于是，在 1975~1976 年研制成集成度更高、功能更强的微处理器 Intel 8085 和第一批真正的单片微型计算机 MCS-48 系列产品，单片机的典型代表是 Intel 8048。在此之后，General Instrument 公司也推出了自己的单片机系列产品 PIC1650。由于这类产品只增加了并行 I/O 接口、简单的定时器/计数器，内部存储器的容量也比较小，因此称为中、低档单片机，主要用于工业控制、汽车、仪器仪表以及家用电器中。

第三阶段：1978~1983 年，高性能 8 位单片机。Intel 公司研制成第一批 MCS-48 系列单片机，引起各方面人士的极大兴趣，于是许多厂家也都开始着手于单片机的研制。在这一阶段，单片机无论在性能和种类方面都有很大的发展，其内部不但具有多个并行 I/O 接口，而且还配置了串行通讯接口。内部 RAM 和 ROM 的容量也有了很大的提高，有的还增加了 A/D 转换电路，构成符合不同要求的专用单片机，有的还把 BASIC 语言的解释程序固化在片内，使单片机可以直接使用高级语言。例如，Intel 公司研制成高性能的 MCS-51 系列单片机，Zilog 公司研制成 Z8 系列单片机，Motorola 公司研制成 MC6801 系列单片机。在这些单片机中，都增加了串行通讯接口。其中，在 Z8 系列的 Z8671 中固化有 tiny-BASIC 语言的解释程序。此外，Intel 公司还推出了许多专用单片机，例如 Intel P80A49H/P80A39HL、P80A48L 及 P80A48H/P80A35HL 采用低成本的塑料封装，能在 $-40^{\circ}\text{C} \sim +110^{\circ}\text{C}$ 范围内工作，可作为车辆专用单片机。Intel 8022 内部配置有二输入通道的 8 位 A/D 转换器，专门用于数据采集和实时控制。Intel 80C48、80C49 是一种低功耗单片机，工作电压范围宽，可用电池供电，专门用于各种便携式袖珍仪器仪表、远程无线电通讯设备、自动化设备、家用电器以及一些高档日用消费品中。在这一阶段，Rockwell 公司也推出了自己的系列产品 6500，National Semiconductor 公司也推出了自己的系列产品 NSC-800 等。

第四阶段：1983 年至今，高性能 8~16 位单片机。在这一阶段单片机的类型增加到 100 多种，16 位单片机也有许多种类。其中，典型代表是 Intel 公司的 MCS-96 系列单片机。它的集成度为 12 万只晶体管/片，寻址范围为 64K，片内 ROM 为 8K，RAM 为 232 个字节，有 5 个 8 位并行 I/O 接口、一个全双工串行通讯接口、4 个 16 位可编程序定时器/计数器、4~8 输入通道的 10 位 A/D 转换器，采用 8 级中断管理方式，可进行位、字节、字、双倍字操作。在指令系统中设置有 16 位 $\times$ 16 位的乘法指令和 32 位 $\div$ 16 位的除法指令，若采用 12MHz 的时钟，指令平均执行时间为 $1 \sim 2\mu\text{S}$ ，进行一次乘法或除法运算也只需要 $6.5\mu\text{S}$ ，其功能可与一台多片微机系统相媲美。这一阶段，在发展 16 位单片机的同时，8 位单片机仍在不断地改进、完善和发展，其种类愈来愈多，功能愈来愈

愈强。例如，Intel 公司在推出 MCS-96 系列单片机之后，又推出了介于 MCS-51 与 MCS-96 之间的 CHMOS 型 83C252 / 87C252 / 80C252 单片机。这种单片机的指令系统与 MCS-51 系列完全兼容，但又具备 MCS-96 系列的高速输入 / 输出功能，因此其性能得到很大提高。

目前，单片机仍在不断地发展和完善，应用范围仍在不断地扩大和延伸，逐步成为微型计算机应用领域中的重要角色。不难看出，在不久的将来，将会有更多更好，功能更为齐全的单片大量涌现出来，一些采用单片机的仪器仪表、专用设备、家用电器等也会大量涌现出来。

近年来，我国科技工作者在单片机的引进、开发和使用方面做了大量的工作，研制成许多单片机开发系统、单片单板机、单片微型机系统和单片机仿真器。把单片机用于数据采集、巡回检测、过程控制、智能仪器仪表和家用电器等方面，为我国的计算机事业和四个现代化建设做出了积极的贡献。例如，在最近几年，研制成 TP801MCS 单片机开发系统、DSG-51 仿真器、SCE-48 单片机开发装置、DVCC-51 开发型单片单板机、DVCC-96 / 98 单片机开发装置、DSG-96 / 98 仿真器、TP STD 8098 应用开发系统等。在单片机的应用方面，研制成单片机电话调度系统、WDK-1 单片机微电脑控制器、PC8031 工业控制器、RC51 车床控制专用单片机以及 JK 型间歇吸落棉控制仪器等。

目前，在我国单片机的发展方兴未艾，它以超小型化、抗干扰能力强、易于开发使用、工作可靠、对环境要求不高、价格低廉等优点，正在逐步占领广大的微机应用领域。可以预计，在不久的将来，各种开发应用系统、自动控制装置、智能仪器仪表以及各种标准化的应用产品在我国也将会大量涌现出来。

第二节 微型计算机的工作原理

一、计算机的基本组成

计算机的基本组成如图 1-1 所示，主要包括运算器、控制器、存储器、输入设备和输出设备等五个部分。这五个组成部分通常称为计算机的硬件设备，是构成计算机的物理实体。在计算机中各硬件设备之间通过专门的线路来联接，进行信息

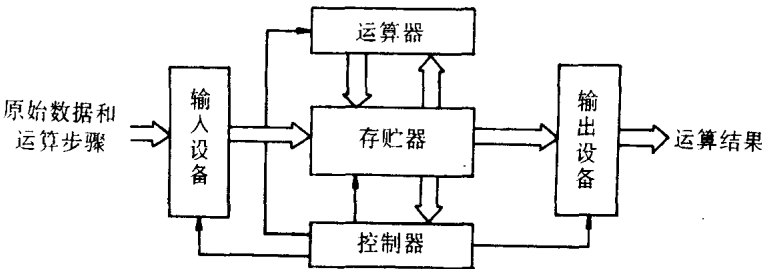


图 1-1 计算机的基本组成

传送，这些专门的线路称为总线。总线一般可分为三类，一类用来传送数据，称为数据总线，用 DB 表示；另一类用来传送地址，称为地址总线，用 AB 表示；第三类用来传送控制信号，称为控制总线，用 CB 表示。下面针对五个组成部分和总线结构作以简单介绍。

1. 存储器

存储器的基本组成如图 1-2 所示, 由存储体、地址寄存器、地址译码器、三态输入输出缓冲器和读/写控制逻辑电路组成, 用来存放参加运算的操作数和指示计算机进行某一运算的程序。计算机之所以能够高速、自动地进行各种复杂的运算, 重要原因就是它能把解题所需要的各种数据和指示计算机如何进行运算的程序事先存入存储器。解题时再从存储器中将程序和数据快速取出, 送控制器和运算器进行加工, 这就是所谓的“程序存储”。由此可见, 存储器是计算机的重要组成部分。

在存储器中, 各种信息都是存放在存储体中。存储体由许多存储单元组成, 如图 1-2 (b) 所示。这些存储单元依照阵列方式进行排列, 因此存储体也称为存储矩阵。在存储矩阵中, 每一个单元各有自己的编号, 这一编号通常称为地址。各种信息在存储器中存放时, 均按地址进行, 因此有条不紊, 不会发生差错。在微型计算机中, 一个地址单元一般只能存放 8 位二进制数, 这 8 位二进制数称为一个字节。在图 1-2 (a) 中, 存储器地址由 8 位二进制数表示, 经译码后可产生 $2^8 = 256$ 个地址编号, 因此图 1-2 (a) 所示存储器最多能够存放 256 个 8 位二进制数, 这个“256”称为存储器的容量, 也称为 8 位地址的最大寻址范围。对于 10 位地址, 其最大寻址范围为 $2^{10} = 1024$, 通常称为 1K。对于 20 位地址, 其最大寻址范围为 1024K, 通常称为 1M (兆)。对于 n 位地址, 其最大寻址范围 2^n 。

在图 1-2 (a) 中, 地址寄存器用来接收由地址总线 $A_7 \sim A_0$ 送来的地址, 该地址由译码器进行译码, 以选择存储器中的某一单元。译码器选用多—1 译码, 两个多—1 译码

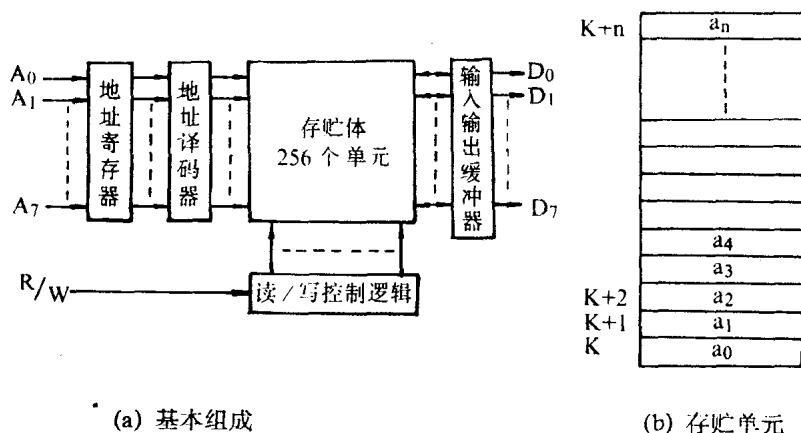


图 1-2 存储器的基本组成

器交叉使用则构成复合译码方式, 若由单一的多—1 译码器进行译码, 则构成字译码方式, 也称为线选译码方式或线选法。

三态输入输出缓冲器用来暂存由存储器中取出的数据, 或要存入存储器的数据。通常, 向存储器中存入数据称为写入, 由存储器中取出数据称为读出。数据的写入或者读出由读/写控制逻辑电路来控制, 每向存储器写入一个数据或者读出一个数据, 统称为访问了一次存储器, 即存储器访问。

在向存储器写入数据时，首先要将存储器单元地址送地址寄存器，要写入的数据送输入输出缓冲器，然后由控制器通过读/写控制逻辑电路发写入控制信号，即写命令。这样，三态输入输出缓冲器中的数据就写入到地址寄存器中的地址所指示的单元中去。从存储器中读出数据时，同样要先将存储器单元地址送地址寄存器，然后由控制器通过读/写控制逻辑电路发读出控制信号，即读命令。这样，地址寄存器中的地址所指示单元中的数据就被读出到三态输入输出缓冲器中。存储器一般多采用非破坏性读出的器件组成，其中的内容可多次读出，而不受破坏，这样的读出称为非破坏性读出。在写入时，原来的内容被新写入的数据所取代。

在计算机中，存储器按其器件的性能可分为两类。一类是随机存取存储器，简称为 RAM。这类存储器在计算机工作时，其中的内容可以随机地读出或者写入。另一类是只读存储器，简称为 ROM。这类存储器中的内容在计算机工作之前已经写入，计算机工作时，可以随机读出，而不能写入。只读存储器又可分为三种，一种是固定只读存储器，用 ROM 表示，其中的内容在工厂生产时已经写入。用户买回后，其内容只能读出使用，而不能进行改写。另一种是可写入只读存储器或称为可编程只读存储器，用 PROM 表示。这种存储器，用户买回时，其内容为全“1”或者全“0”，用户可根据自己的需要，进行一次写入。一旦写入，则不能再进行任何改写。第三种是可改写只读存储器，或称为可改写可编程只读存储器，用 EPROM 表示。这种存储器其内容可通过一定的方式进行清洗，清洗后可重新进行写入。

上述存储器是指设置在计算机内部的存储器，这种存储器通常称为内存存储器，简称为内存。在微型计算机中，内存存储器的容量一般较小，往往不能满足用户的需要。为了解决这一矛盾，计算机通常还配备有另一种存储器。这种存储器一般容量很大，但是设置在计算机的外部，因此称为外存储器或者辅助存储器，简称为外存或者辅存。外存储器种类较多，目前常用的有磁盘、磁带、穿孔纸带等。微型计算机工作时，可将大量的程序和数据存放在外存储器中，当某一程序段需要执行，或某些数据需要处理时，调入内存执行或处理，执行或处理完后，再将结果送回外存储器存放。

另外，在内存存储器中还设置有一个特殊的工作区域。在这个区域中，数据的存取遵循“后进先出（也称为先进后出）”的原则，其示意如图 1-3 所示。数据在存放时，先存入的数据在最底下，后存入的数据在最上面，数据取出时，上面（后存入）的数据先取出，下面（先存入）的数据后取出，这一特殊的工作区域称为堆栈。在向堆栈中存入数据时，称为数据进栈或压栈，在从堆栈中取出数据时，称为数据出栈或弹栈。在堆栈中，最上面的数据单元称为栈顶，最下面的数

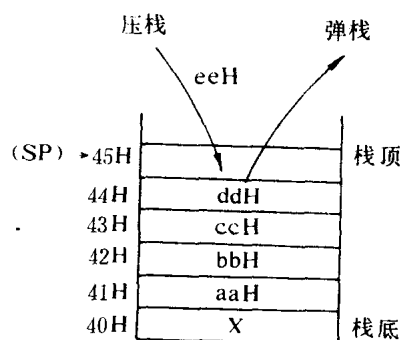


图 1-3 堆栈示意图

据单元称为栈底。为了控制堆栈可靠工作，使数据在弹栈或者压栈时有条不紊，围绕堆栈设置有一个特殊的地址寄存器，用来存放栈顶地址。这个寄存器通常称为堆栈指针或堆栈指示器，用 SP 表示。堆栈指示器具有自动加 1 和自动减 1 功能。当数据压栈时，堆栈指

示器自动加 1，数据压入。当数据弹栈时，数据弹出，堆栈指示器自动减 1，这样可使堆栈指示器始终保持指示栈顶地址。在图 1-3 中，高地址在堆栈的上面，低地址在堆栈的下面，这样设置堆栈的方式称为向上生成。如果相反，则称为向下生成。在微型计算机中，堆栈通常用来存放返回地址、现场数据和一些临时需要保护的信息。

2. 运算器

运算器也称为算术逻辑部件，其内部组成如图 1-4 所示，主要包括加法器 ALU、累加器 A、暂存寄存器和状态标志寄存器。

加法器主要用来进行算术逻辑运算，它由以下几个操作部件组成：

(1) 具有算术、逻辑运算功能的全加器；

(2) 加 1、减 1 电路；

(3) BCD 码修正电路等。

它所具备的功能有以下几点：

(1) 算术运算：加、减、加 1、减 1、(乘、除)；

(2) 逻辑运算：“与”、“或”、“异或”；

(3) 比较；

(4) 移位：算术移位、逻辑移位、循环移位；

(5) 位操作：位测试、位置“1”、位置“0”；

(6) BCD 码修正。

加法器 ALU 是运算器的主要组成部分，它有两组输入端，分别与两个暂存寄存器联接，其中一个通过数据总线接收存储器或其它寄存器送来的数据，另一个接收累加器送来的数据。累加器 A 是运算器的重要组成部分，用来暂存参加运算的数据、中间结果或者结果，它与加法器配合完成运算器的所有功能。在一般微型计算机中，数据总线的条数与累加器、加法器的位数相同。计算机工作时，在控制器发出一系列控制信号的控制下，暂存寄存器中的数据送入加法器进行算术/逻辑运算。运算结果由输出端输出，再经内部数据线送入累加器或者存储器，或者直接输出打印/显示。在有的计算机中，其中的一个暂存寄存器与累加器合并使用。在运算器中还设置有十进制调整电路，十进制调整电路也称为 BCD 码修正电路，在进行 BCD 码运算时，其结果由十进制调整电路进行修正。

在运算过程中，运算结果可能为零、可能为负数，或者最高位产生进位，或者发生溢出等。为了便于控制器进行检查和处理，在运算器中还设置有一个状态标志寄存器，专门用来存放运算结果的特征状态。状态标志寄存器也称为程序状态寄存器或状态特征寄存器或状态控制寄存器。在不同的书中，有不同的名称，但其作用是一样的。它的位数由各种机器特定，其中每一位

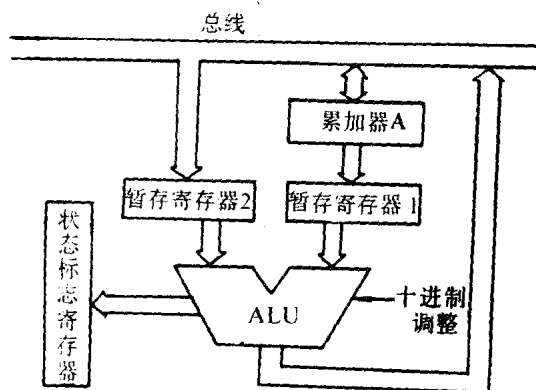


图 1-4 运算器逻辑图

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
C	AC				OV		P

图 1-5 程序状态标志寄存器

表示一种特征状态，如图 1-5 所示。在图 1-5 中，状态标志寄存器共有 8 位，但这里仅使用了其中的 4 位，各位的作用如下：

D₇：进位标志，用 C 表示。在运算过程中，如果结果最高位产生进位，C 置“1”（表示最高位产生了进位），否则 C 置“0”，表示最高位没有进位。

D₆：辅助进位标志，用 AC 表示。在进行 BCD 码运算时，由低位向高位产生的进位称为辅助进位。当低位向高位产生进位时，AC 置“1”（表示运算过程中有辅助进位），否则 AC 置“0”，表示没有辅助进位。

D₂：溢出标志，用 OV 表示。在进行带符号数运算时，若运算结果最高位发生溢出，则 OV 置“1”（表示运算结果发生了溢出），否则 OV 置“0”，表示运算结果没有发生溢出。

D₀：奇/偶标志，用 P 表示。在运算过程中，若累加器奇偶校验和为奇数时，P 置“1”，作为奇标志，否则 P 置“0”，表示奇偶校验和为偶数。

在有的计算机中，状态标志可能设有 6 位或者 8 位。具体位数是在硬件设计时，根据需要而定。如果上述 4 位已经够用，则也可以使用 4 位寄存器。

3. 控制器

控制器是计算机自动工作的指挥中枢，其结构如图 1-6 所示，由程序计数器、指令寄存器、指令译码器、微操作控制电路和时序电路组成。

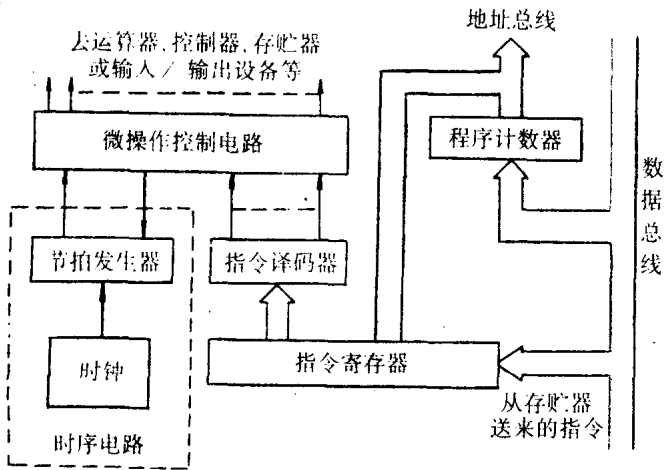


图 1-6 控制器逻辑图

程序计数器也称为指令计数器，用 PC 表示，主要用来指示指令在存储器中存放的地址。程序是为解决某一特定问题所使用的指令的有序集合，它在存储器中顺序存放，执行时按其地址顺序取出。为了不发生混乱，在控制器中设置了程序计数器。程序计数器既可接收数据总线送来的地址，又可通过地址总线将地址送往存储器的地址寄存器。程序计数器具有自动加 1 功能，在程序

执行时，每取一条指令（字节），程序计数器自动加 1，然后指示下一条指令（字节）的地址。

指令寄存器用来存放从存储器中取出的指令。指令从存储器中取出后送指令寄存器暂存，然后在执行时，其操作码送指令译码器进行译码，地址码送地址总线，以确定操作数的地址。

指令译码器主要用来对指令寄存器中的操作码进行译码，以确定指令的性质。译码结果送微操作控制电路，然后由微操作控制电路产生控制计算机各部件进行微小动作的

令。例如，打开某一门电路，接收或发送数据等。这一微小动作称为微操作，这一命令称为微命令。一条指令的执行过程可以分解为若干个微操作，这些微操作由若干条微命令进行控制。因此，一条指令的执行过程实质上就是这若干条微命令的执行过程，微操作控制电路就是根据指令的性质发出一系列微命令的部件。

微操作控制电路可以采用硬件逻辑电路，也可以采用微程序控制方式。在结构比较简单的计算机中，多采用硬件逻辑电路，在结构比较复杂的大、中型计算机中多采用微程序控制方式。

时序电路包括时钟和节拍发生器，主要用来产生计算机工作时所需要的时钟脉冲和节拍电位。

4. 输入设备与输出设备

计算机之所以能够自动地进行工作，主要原因是将解题程序与原始数据事先存入内存贮器。而解题程序与原始数据的存入是由输入设备进行的。在计算机中，任何程序与数据都是由二进制代码表示的，而人们通常喜欢使用的则是字母、文字和十进制数。为了便于计算机执行，这些字母、文字和十进制数必须转换成二进制代码，而输入设备正是用来将参加运算的数据、程序或人们的命令转换成二进制代码输入给计算机的设备。因此，与运算器、控制器和存贮器一样，输入设备也是计算机的主要组成部件。在目前，常用的输入设备有键盘、纸带输入机、磁带机以及数传机等。

输出设备是用来将运算结果或中间数据转换成人们容易识别的字母、文字、十进制数或图表，输出给人们的设备。在目前，常用的输出设备有打印机、显示器、绘图仪、纸带穿孔机以及磁带机等。

以上五个组成部分是计算机的主要组成部件，五者缺一不可。通常人们把运算器、控制器合称为中央处理器，用 CPU 表示，中央处理器与存贮器合称为主机。

5. 总线结构

使用总线进行信息传送，是目前微型计算机中普遍采用的信息传送方式。所谓总线，

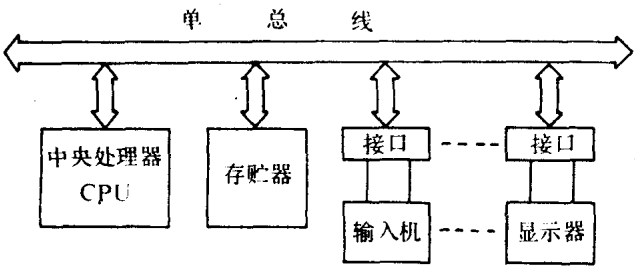


图 1-7 单总线结构

就是信息传送的公共通道。在微型计算机中，各组成部件：CPU、存贮器、输入设备、输出设备等，均作为一个独立的个体挂在总线上，相互之间都通过公共总线进行信息传送，结构如图 1-7 所示。在图 1-7 中，CPU 存贮器、输入输出设备平等地挂在同一总线上，相互之间再没

没有其它信息通路，因此这样的结构称为单总线结构。

总线由三部分组成，一部分专门用来传送数据，称为数据总线 (DATA BUS)，简称为 DB。另一部分专门用来传送地址，称为地址总线 (ADDRESS BUS)，简称为 AB。第三部分专门用来传送控制信号，称为控制总线 (CONTROL BUS) 简称为 CB。各设

备与总线联接时，均通过由三态门构成的总线驱动器（也称为总线缓冲器）联接。三态门如图 1-8(a)所示，由控制信号控制其通导。当控制信号有效时，输出端与输入端接通。控制信号无效时，输出端与输入端不通。由三态门构成的总线驱动器如图 1-8 (b、c) 所示，当控制信号有效时输出端与输入端通导，就使某设备与总线接通，否则输出端与输入端不通，设备不能与总线接通。如果使信息只能沿着一个方向传送，那么这种驱动器就称为单向总线驱动器，如图 (b) 所示。如果使信息可沿两个方向传送，那么这种驱动器就称为双向总线驱动器，如图 (c) 所示。采用单向驱动器与设备联接的总线称为单向总线，采用双向驱动器与设备联接的总线称为双向总线。一般情况下，数据传送使用双向总线，地址和控制信号的传送使用单向总线。

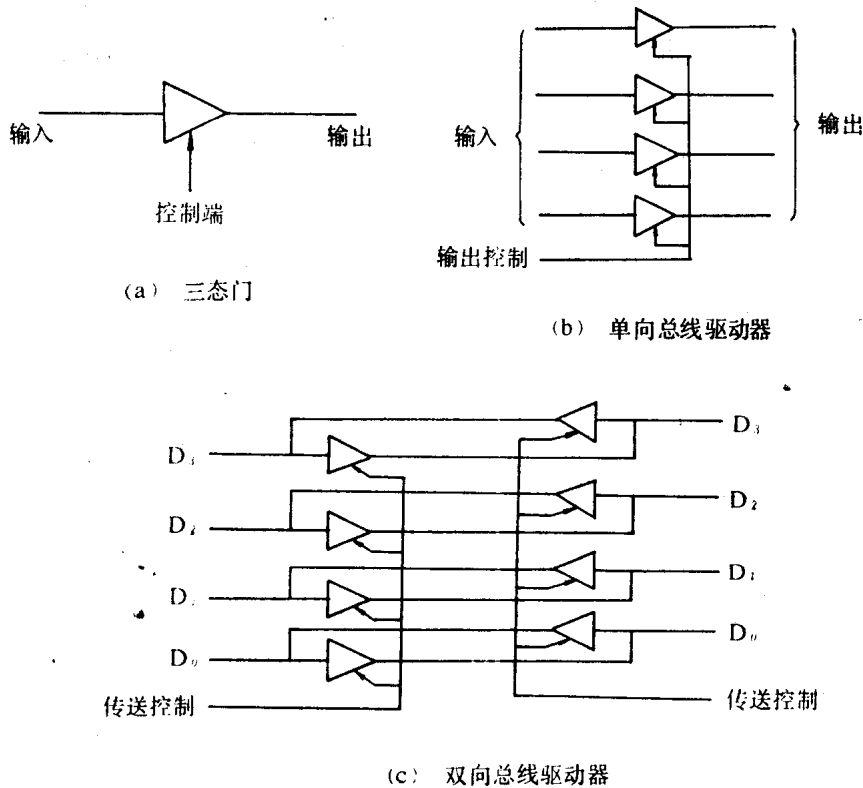


图 1-8 三态门与总线驱动器

以上介绍的仅是计算机的硬件组成原理，如果只有硬件设备，计算机是无法工作的。为了使计算机能够按照人们的意图自动地工作，必须配之以必要软件，构成一个完整的系统。软件愈丰富，计算机的功能愈强，其效益愈高。

二、指令、程序与程序的执行过程

1. 指令与程序

为了能对指令与程序有一个比较直观的了解，下面先看一个简单算式的求解过程。

例如求解 $y = (a+b) \times c$

人们知道，在计算上式时，首先要把原始数据 a 、 b 、 c 记录下来，再按照一定的步骤一步一步地计算，即先求 $a+b$ ，记录中间结果，再求 $(a+b) \times c$ 。然后将计算结果赋值给变量 y ，最后记录结果。

在计算机求解时，首先通过输入设备把原始数据和解题步骤输入给计算机（存放在存储器中），然后启动计算机进行计算。这样，计算机就按照人们规定的解题步骤，一步一步地进行计算。计算完后，将结果通过输出设备显示或者打印出来，整个解题过程可用图 1-9 表示。在图 1-9 中，解题步骤分为 6 步。第一步输入原始数据和解题步骤，第 2 步取 a ，第 3 步取 b ，并求 $a+b$ ，第 4 步取 c ，并求 $(a+b) \times c$ ，第 5 步将结果赋值给 y ，第 6 步显示或打印 y 值，最后结束。

不难看出，图 1-9 比较清楚地描述了计算机的解题步骤和执行过程。通常，把这种用来表示解题步骤和计算机工作过程的框图称为流程图。但是，流程图只能表示大概的过程，而不能实际指示计算机去工作。为了完成全部求解工作，必须根据流程图所指出的步骤向计算机发一道一道的命令，这些命令称为“指令”，这些指令的有序集合，汇集

成求解这一题目的“程序”。

在计算机中，各种部件所做的每一件事情统称为操作，参加运算的数称为操作数，用二进制表示，其位数称为字长。为了便于存取管理，常把 8 位二进制数称为一个字节。当字长为 8 位时，字长与字节位数相等，当字长为 16 位时，一个字长可分为 2 个字节，当字长为 32 位时，可分为 4 个字节。由于计算机只能识别二进制数，因此在计算机中指令也由二进制数来表示，这个二进制数称为指令码或指令字。当指令字长为 8 位时（即一个字节），称为单字节指令，当指令字长为 16 位、24 位或 32 位时，则称为双字节、三字节或四字节指令。指令具有一定的格式，这在不同的机器中有不同的规定，就一般而言，如图

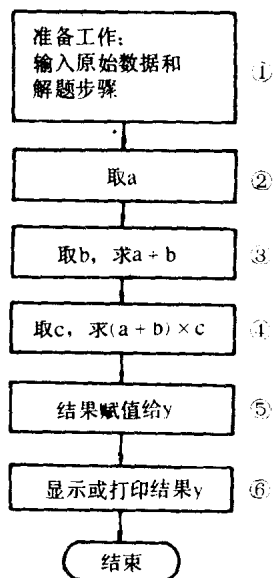


图 1-9 求解 $Y = (a+b) \times c$ 流程图

1-10 所示，包括两个部分。一部分用来表示指令的性质，即指示机器做什么事情，这一部分称为操作码。另一部分用来表示参加运算的数在计算机中存放的地址，这一部分称为地址码。由于在图 1-10 所指示的格式中，只有一个地址码，这种指令称为单地址指令，若有两个地址码，则称为双地址指令，此外还有三地址指令和多地址指令。

在微型计算机中，多采用单地址指令和双地址指令。

在上述求解过程中，可用 6 条指令来完成，这 6 条指令按其先后顺序排列起来，就构成求解上题的程序。

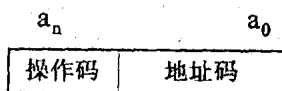


图 1-10

2. 程序的执行过程

计算机工作时，首先要用输入设备把程序和数据通过输入接口电路和数据总线送入内

存贮器，然后逐条取出执行。程序由一系列指令组成，而指令又可分为操作码与地址码。这样，对于一般指令的执行可分为两个阶段。第一阶段取指令、译码、分析指令的性质和地址码的类型。第二阶段取操作数，并按照指令操作码的性质对操作数进行操作，即执行指令。

为了便于读者了解指令的执行过程，结合图 1-11 予以说明。

(1) 首先将程序第一条指令的地址（即首地址）送程序计数器 PC，然后启动计算机运行。在计算机运行后，PC 中的地址通过地址总线送入存贮器的地址寄存器，经译码找到指定地址单元，并进行存贮器读操作。读出的指令通过数据总线送入指令寄存器，这一过程称为取指令。

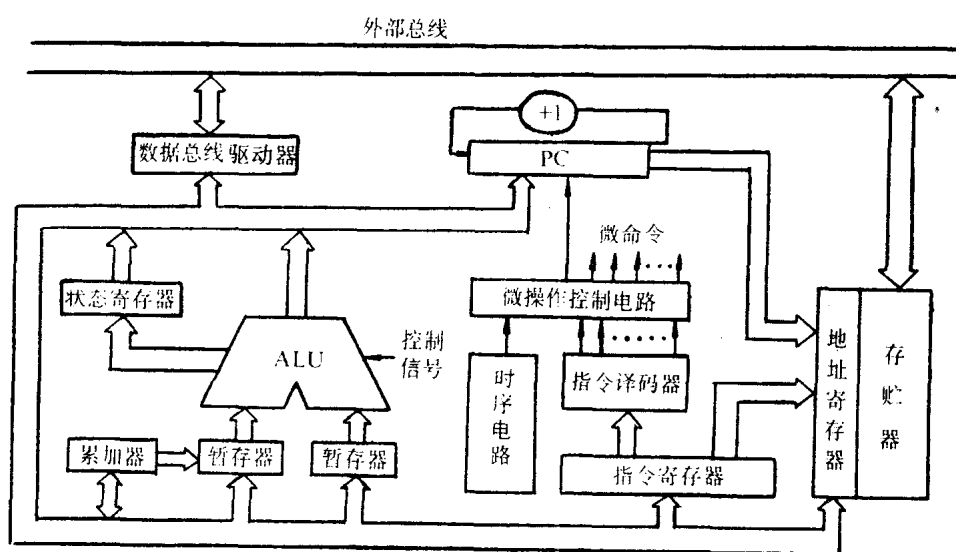


图 1-11 程序执行过程硬件示意图

(2) 指令取出后，程序计数器 PC 自动加 1，指示下一条指令的地址，对于多字节指令则指示下一字节单元。

(3) 将指令寄存器中存放的指令操作码送指令译码器进行译码，以确定指令的性质（即进行何种操作）。地址码通过地址总线送存贮器的地址寄存器，以指示存贮单元地址。

(4) 根据指令操作码的译码结果，在时序电路的配合下，由微操作控制电路发相应的控制信号，即微命令，控制有关部件进行相应的工作，执行命令。

(5) 对于访问存贮器指令，则启动存贮器进行读/写操作。其中，存贮器读指令，将地址码所指示单元中的数读出，送累加器或指定寄存器。存贮器写指令，则将累加器或指定寄存器中的数写入到地址码所指示的单元中去。

(6) 对于算术逻辑运算指令，则将读出的数送入暂存寄存器，然后与累加器中的另一操作数在微命令的控制下，按照指令的性质进行运算，运算结果再经数据总线送回到累加器中，运算结果的状态特征送入状态标志寄存器，以供 CPU 检查。

(7) 一条指令执行完后，再将程序计数器 PC 中的内容送入地址寄存器，以取下一条指令，直到程序完毕为止。

三、CPU 定时时序

计算机是由许多部件联接起来的一个组合体，为了使各组成部件协调一致，有条不紊地工作，在控制器中设置有时序电路。计算机的全部工作都是在时序电路的作用下，一拍一拍地进行。为了便于管理，常把指令的执行过程分解为若干个时间段，即周期，在每一周期中计算机进行特定的工作。在微型计算机中，周期可分为三类，即时钟周期、机器周期和指令周期。控制器严格按照这些周期去控制指令的执行，下面就对三类周期作以必要的说明。

1. 时钟周期

时钟周期是计算机中最基本的定时单位，用 S 表示，于是也称为 S 周期或 S 状态。在一个时钟周期内，计算机仅完成一个最基本的动作，例如将某一门电路打开或者关闭等。对于一台单片微型计算机，若采用 6MHz 的时钟频率，则一个时钟周期为 166.7ns，若采用 4MHz 的时钟频率，则时钟周期为 250ns。

2. 机器周期

机器周期是指计算机完成某一基本工作所需要的时间。为了便于管理，常把一条指令的执行过程分为若干个阶段。每一阶段，计算机进行某一项工作，例如取指令、存储器读、存储器写等。这一项工作称为一个基本操作。完成一项基本操作所需要的时间称为机器周期，一个机器周期通常由若干个时钟周期组成。在 MCS-48 单片微型计算机中，一个机器周期包括 5 个时钟周期 S。

3. 指令周期

指令周期是指计算机执行完一条指令所需要的时间，它由若干个机器周期组成。指令不同，所需要的机器周期数也就不同。对于一些简单的单字节指令，在取指令周期中，指令取出到指令寄存器后，立即译码执行，这样就不再需要其它机器周期。对于一些比较复杂的指令，例如加法指令，则有一个取指令和取操作数的过程。因此，加法指令的执行就需要两个或者两个以上的机器周期。例如在 MCS-51 单片微型计算机中，一条加法指令

的执行需要 1~2 个机器周期，一条乘法或除法指令的执行需要 4 个机器周期。

时钟周期、机器周期与指令周期之间的关系如图 1-12 所示，一个指令周期包括若干个机器周期，一个机器周期包括若

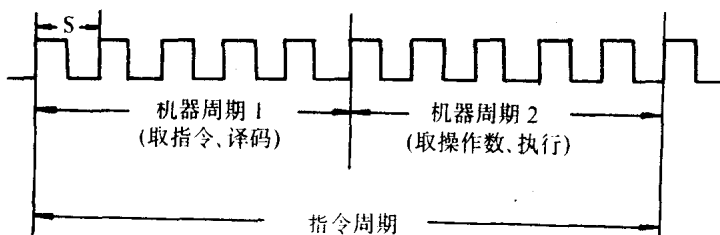


图 1-12 周期状态关系

千个时钟周期。控制器严格按照三类周期的定时关系控制指令的执行。

四、中断技术

1. 中断概念

在一般情况下，计算机总是不断地执行程序，完成人们赋予的工作。但是，在程序运行过程中，可能有意外事件发生，例如机器出现故障或者外围设备请求等。这就要求计算机停止当前程序的运行，去处理机器故障，或为外围设备服务。通常，让计算机停止当前程序的运行，去为意外事件服务的过程称为中断，为意外事件的服务称为中断服务或中断处理，要求服务的意外事件称为中断源，中断源要求为其服务的请求称为中断请求。意外事件的处理（或者为其服务）也是通过程序来实现，这样的程序称为中断处理或中断服务程序。原来运行的程序称为主程序，主程序被停止（断开）的位置（地址）称为断点。在意外事件处理完后，计算机返回到原来被中断的位置，继续运行原来的程序，这样的过程称为中断返回，如图 1-13 所示。在中断服务过程中，可能会有新的意外事件发生，这样计算机就要停止当前的中断服务程序，去为新的意外事件服务。服务完后，再返回到原来的中断服务程序，继续为原来的中断源服务。这一过程也可以从图 1-13 看出，称为中断嵌套。

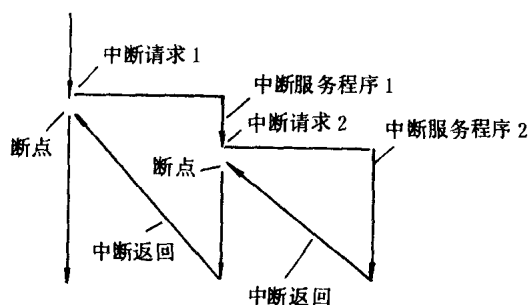


图 1-13 中断与中断嵌套示意图

2. 中断优先权与中断判优

在实际的计算机应用中，中断源往往不是一个，而是几个，甚至几十个。这样，就有可能两个或者两个以上的中断源同时提出中断请求，即出现了竞争现象。这就要求计算机既能区分不同中断源的请求，又能确定首先为哪个中断源服务。为了解决两个或两个以上中断源的竞争现象，通常为各中断源规定了优先级别。当两个或者两个以上的中断源同时提出中断请求时，计算机首先判断哪个中断源的优先级别最高，就先为那一个中断源服务。服务结束后，再去为次高级中断源服务。计算机的这一过程通常称为判优。判优可由专门的优先级排队电路来实现，也可由软件来实现。软件实现方法是通过查询程序对各中断源进行查询，查询时按其优先级别由高到低进行，当优先级别高的中断源有中断请求时，则优先予以响应，并转中断服务程序为其服务。由于在查询时，按其优先级别由高到低进行，这样就保证在有多个中断源同时提出中断请求时，首先为其级别最高者服务，软件查询流程图如图 1-14 所示。优先级排队电路可以自行设计，也可选用现成的集成电路。例如 74LS148 就是一种 8 输入的优先级编码器，若将 74LS148 与 3-8 译码器结合起来，就可构成优先级排队判优电路。

在一般情况下，通常把机器故障设定为最高优先级，其余外围设备根据运行速度进行排队，速度快的设备级别高，速度慢的设备级别低。在中断嵌套时，优先级别高的中断源可以中断优先级别低的中断服务程序，而优先级别低的中断源不能中断优先级别高的或者