

IBM-PC/XT

软硬件系统分析与应用

(修订本)

张载鸿 编著

中国科学院
北京科海培训中心

一九九〇年一月

IBM-PC/XT

软硬件系统分析与应用

(修订本)

张载鸿 编著

中 国 科 学 院

北京科海培训中心

一九九〇年一月

再 版 前 言

IBM-PC/XT是美国IBM公司推出的IBM微型计算机序列中的一台扩展型个人计算机。该机以先进的系统结构和丰富的系统软件代表着当代微型计算机发展的新潮流。自八十年代初引进我国后，很快受到计算机行家的重视和各界用户的普遍欢迎。

随着IBM个人计算机及其兼容机在我国的推广普及，也相继出版了不少IBM-PC系列机的有关书籍和资料，但至今大多是机器原理、技术手册和用户指南等一类的内容，缺少对该系统的软件、硬件及应用进行全面、系统的分析，而这正是计算机用户对IBM-PC系列机进一步开发、应用和维护所不可缺少的。

为此，本人在研究、分析了美国IBM公司所提供的有关原版的技术资料的基础上，重点对以下几部分进行了分析：

- DOS的结构和系统功能调用
- BIOS的结构和中断调用
- 机器指令系统及汇编程序设计
- 系统部件的结构及外围接口芯片的应用
- 存储系统的结构及功能
- 彩色/图形显示系统的结构及应用
- 软盘子系统的结构及应用
- 硬盘子系统的结构及应用
- 针式打印机系统的结构及应用
- 异步串行通信系统的结构及应用

本书的编写采用软硬相结合、系统和应用相结合的方式，在对系统各部分结构特点及硬件逻辑进行分析的基础上，说明如何用汇编语言实现程序设计，以达到对系统开发和应用的目的。

本书修订再版，纠正了第一版内的某些错误，重制了全部插图。

本书能顺利再版与读者见面，得到了中国科学院北京科海培训中心主任华根娣同志、编辑夏非彼同志热情的支持和帮助，在此表示衷心的感谢。

由于本人水平所限，书中难免还有错误和不足，谨请读者提出宝贵意见。

作者

于北京计算机学院

目 录

第一章 概述	(1)
1.1 微型计算机的发展.....	(1)
1.2 微型计算机系统综述.....	(2)
1.2.1 总体结构.....	(3)
1.2.2 系统总线.....	(3)
1.2.3 中央处理器CPU	(6)
1.2.4 微处理机时序.....	(11)
1.2.5 半导体存储器.....	(12)
1.2.6 I/O设备与接口	(17)
1.2.7 微型计算机的软件系统.....	(30)
第二章 IBM-PC/XT系统综述.....	(34)
2.1 IBM微型计算机系列.....	(34)
2.2 IBM-PC/XT系统.....	(35)
2.2.1 系统结构.....	(35)
2.2.2 系统部件.....	(36)
2.2.3 扩展部件.....	(44)
2.3 系统软件.....	(55)
2.3.1 PC操作系统	(55)
2.3.2 BIOS系统	(103)
2.3.3 调试程序DEBUG.....	(124)
2.3.4 编辑程序EDLIN	(132)
2.3.5 连接程序LINK	(134)
2.3.6 汇编程序.....	(137)
2.3.7 编译程序.....	(155)
2.3.8 解释程序.....	(163)
2.3.9 诊断程序.....	(187)
第三章 中央处理器指令系统	(202)
3.1 8088结构及操作特点.....	(202)
3.1.1 CPU结构框图	(202)
3.1.2 寄存器结构及操作.....	(206)
3.1.3 存储空间的编址.....	(210)
3.1.4 I/O设备的存取	(212)
3.1.5 CPU工作模式	(215)
3.1.6 总线周期操作.....	(220)

3.1.7 处理机控制特征	(222)
3.1.8 多处理机特征	(229)
3.2 8088寻址方式	(233)
3.2.1 指令格式	(233)
3.2.2 寻址方式	(236)
3.3 8088指令系统	(241)
3.3.1 数据传送指令	(241)
3.3.2 算术运算指令	(245)
3.3.3 逻辑运算指令	(255)
3.3.4 数据串操作指令	(259)
3.3.5 程序转移指令	(262)
3.3.6 处理机控制指令	(268)
第四章 系统部件结构及应用	(270)
4.1 系统板综述	(270)
4.2 系统中央处理机	(272)
4.2.1 系统时钟及等待状态产生器	(272)
4.2.2 处理机与总线	(275)
4.3 8259A中断控制器及应用	(277)
4.4 8253-5定时器/计数器及应用	(285)
4.5 8255A-5可编程外围接口及应用	(298)
4.6 8237A-5DMA控制器及应用	(309)
4.7 系统RAM和扩充RAM	(327)
4.8 系统ROM和汉卡逻辑	(337)
第五章 IBM键盘接口及应用	(342)
5.1 键盘工作原理	(342)
5.2 IBM键盘硬件接口	(343)
5.3 键盘操作的软件编程	(349)
第六章 IBM打印机系统	(374)
6.1 针式打印机结构及原理	(374)
6.2 打印机适配器及应用	(382)
6.3 汉字打印驱动程序	(391)
第七章 彩色/图形显示系统	(398)
7.1 彩色/图形显示原理	(398)
• 彩色CRT结构	(398)
• 光栅扫描	(400)
• 字母或图形的显示	(401)
• 扫描电流与视频信号时序	(402)
7.2 MC6845CRT控制器及应用	(402)
• MC6845芯片接口信号	(403)

• MC6845内部寄存器组	(404)
• MC6845芯片功能	(406)
• MC6845芯片应用	(408)
7.3彩色/图形监视器适配器逻辑	(409)
• MC6845初始化及控制	(409)
• 字符显示逻辑电路	(414)
• 图形显示逻辑电路	(420)
7.4 彩色/图形监视器适配器编程应用	(426)
• 字符显示硬件滚动	(426)
• 图形显示画点程序	(430)
第八章 5-1/4吋软盘子系统	(435)
8.1 软盘驱动器原理及结构	(435)
• 磁记录编码方式	(435)
• 软盘片记录格式	(438)
• 软盘驱动器结构	(440)
8.2 软盘驱动器适配器硬件逻辑	(446)
• 数字控制端口	(448)
• 软盘控制器电路	(449)
• 写预补偿电路	(454)
• 数字锁相和分离电路	(455)
• 适配器与驱动器接口	(460)
8.3 软盘驱动器适配器编程应用	(460)
• 软盘控制器命令	(460)
• 软盘控制器命令应用	(472)
第九章 5-1/4吋硬盘子系统	(477)
9.1 硬盘驱动器	(477)
• 温盘机结构	(477)
• 温盘机电路逻辑	(478)
9.2 硬盘驱动器适配器硬件逻辑	(481)
• 处理机接口逻辑	(483)
• 智能主控器逻辑	(486)
• 温盘机接口逻辑	(490)
9.3 硬盘驱动器适配器编程应用	(491)
• 状态检测	(491)
• 设备控制块	(491)
• HDC命令功能	(494)
• HDC命令编程应用	(500)
第十章 异步串行通信系统	(503)
10.1 异步串行通信的协议及操作	(503)

• 异步串行通信协议	(504)
• 通用异步通信接口	(505)
10.2 异步通信适配器硬件逻辑	(512)
• 适配器地址译码电路	(513)
• 处理机接口逻辑	(514)
• 异步通信接口逻辑	(515)
10.3 8250通信编程应用	(516)
• 8250寄存器功能	(516)
• 8250通信编程	(523)
• BIOS通信接口及应用	(525)

第一章 概 述

1.1 微型计算机的发展

自1946年诞生第一台电子数字计算机ENIAC以来,在整整四十年期间,计算机技术得到了迅猛的发展,计算机应用已渗透到科学计算、宇航飞行、地质勘探、气象预报、工业过程自动控制等国家经济建设、国防建设的各个领域;尤其进入八十年代,计算机被广泛普及到商业事务管理、数据采集和处理、情报文献检索、文字翻译、各种辅助设计以及办公室自动化、通信网络等社会生活的许多方面。目前,随着计算机硬件价格的下降及各类软件的开发应用,一个普及和应用计算机特别是微型计算机的浪潮正冲击着各个企事业单位。可以预见,用不了多长时间,计算机将成为人们进行现代化建设中一种不可缺少的工具,它所带来的巨大经济效益和社会效益是无法估量的。

四十年期间,计算机走过了“电子管”、“晶体管”、“小规模集成电路”到“大规模集成电路”的发展道路。目前,在硬件芯片上正向“超大规模集成电路”方向发展,以进一步缩小计算机体积,提高其可靠性;而在系统结构方面,探求崭新的总体结构和机器语言,使新一代计算机具有人脑思维、判断及推理能力,被人们称之为“第五代计算机”将很快会脱颖而出。

微处理机属于第四代计算机产品。它于1971年由美国INTEL公司采用PMOS工艺制造的4位微处理机INTEL4004而问世的。4004微处理机虽然只能完成串行的十进制运算,使用机器语言和简单的汇编,但它的诞生标志着计算机进入一个崭新的发展阶段。

随着半导体工艺技术的发展,INTEL公司在1972年推出了8位CPU的第一批产品——8008。在8008的基础上,出现了有代表性的第二代微处理机:如INTEL公司的8080、Motorola公司的M6800、Zilog公司的Z-80;除上述的CPU芯片外,这三家公司又相继研制了与CPU配套使用的时钟发生器、总线控制器、中断控制器,并行接口芯片、定时器/计数器、DMA控制器、打印机控制器、CRT控制器及软盘控制器等等种类繁多、功能完善的各种外围支持芯片,同时配置了内容丰富的系统软件和应用软件。因此,在七十年代中期,由这样一批8位CPU、外围接口芯片以及软件程序组合成的微型计算机系统以强有力的性能价格比冲击着整个计算机市场。随着微型计算机的大量销售,打开了计算机普及应用的新局面。

1974年INTEL公司推出了第三代微处理机:3000位片式微处理机。此外有代表性的位片如AMD公司的2900位片系列、Motorola公司的M10800位片。这类位片采用积木式结构,用若干片位片外加配套的控制芯片组成位片式的微处理机。它的最大特点是速度快,数据吞吐量大,并便于扩充。

在七十年代后期,INTEL公司研制了第四代微处理机产品8088、8086芯片;接着,Zilog公司的Z-8000、Motorola公司的M68000都是有代表性的16位微处理机。它们都采用H-MOS高密度集成的半导体工艺技术。这三种典型的16位微处理机的运算速度比8位处理机快5倍左右,基本指令的执行时间约300ns。配套微计算机系统的软件更加多样化:不

仅能在多种操作系统环境下进行，并且可使用多种高级语言及数据库管理系统。16位微型机的出现使传统的小型机遇到了前所未有的挑战，两者展开了激烈的竞争。

表1-1列出了微处理机四代产品的主要特性。

表 1-1 微处理机四代产品主要特性

位数	典型产品名称	引脚	工 艺	指 令 (条)	基本指令执行时间 (μ S)
4位	Intel 4004	46	PMOS	46	10.8
	TMS 1000	28	PMOS	43	15
8位	Intel 8080	40	NMOS	78	2
	Z80	40	NMOS	158	1.6
	M6800	40	NMOS	72	2
16位	Intel 8086	40	HMOS	100	0.3
	Z8000	48	NMOS	116	0.4
32位	M68000	64	HMOS	56	0.667

八十年代以来，微处理机走向系列化研制的发展道路。首先由Intel公司推出8位、16位和32位的iAPX系列。继8088、8086之后，出现了80186，80286，80386等寻址范围更大、运算速度更快，性能更完善的16位微处理器。不久，又发表了32位微处理机iAPX 432系列。

随着微处理机的性能价格比的日益提高，由几十个、几百个甚至上千个微处理机互连在一起构成的阵列处理机系统已成为现实。而由多个微型计算机所组成的微机局部网络是当前一个令人瞩目的发展领域。

1.2 微型计算机系统综述

在计算机日益推广普及的今天，人们常常用到“微型机”这三个字。其实，从计算机软件 and 硬件相接合的角度来考察“微型机”，它实质上是对“微计算机系统”的简称。

一个完整的微计算机系统 (Microcomputer system，略写MCS) 其核心部件称“微计算机”，除外应配置外围接口及相应设备、电源装置和必不可少的系统软件。

微计算机的核心器件是微处理机 (Microprocessor，简称 μ P或CPU) 也称为微处理器。它只包含常规计算机中的控制器、运算器及数据通路，能执行机器语言描述的指令系统；但仅完成对数据信息的控制和处理，要对信息进行存贮、输入和输出，应为微处理机配备存贮器、输入/输出接口电路以及信息传输用的系统总线而构成一台微计算机 (Mic

rocomputer, 简称 μ c或MC)。

下面, 分若干方面综述微型计算机系统。

1.2.1 总体结构

电子数字计算机的组成遵循“冯·诺依曼结构”的准则。这种结构具有如下特点:

- (1) 数据信息和控制信息按存储地址存放在存储器中。
- (2) 由一个“指令计数器”控制指令的执行。

为满足上述原则, 计算机的结构组成一般划分为控制器、运算器、存储器、输入设备和输出设备等五大部件。

由于大规模集成电路的出现, 使控制器和运算器组合在一块集成芯片CPU上 (Control Processing Unit), 这样, 一台微型计算机的基本结构如图1-1所示。

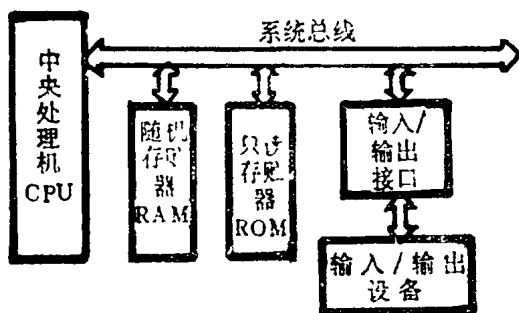


图 1-1 微型计算机基本结构

根据图中的结构, 微型计算机的基本工作原理大致如下

(1) 首先在只读存储器中固化了基本的输入/输出操作系统, 简称BIOS。它是由制造厂家提供的一种系统程序, 其功能包括诸如上电诊断、初始化设备、系统自举以及提供各类I/O设备的基本调用模块; 除外, 有的微型机还固化基本的高级语言如BASIC等。

(2) 机器上电启动进入系统后, 用户程序和数据在CPU的控制下, 通过输入设备和相应的接口, 从系统总线存储在随机存储器RAM里。

(3) 用户程序的首地址存放在CPU中的指令计数器PC里, 由它控制程序的逐条顺序执行。

(4) 每条程序的执行过程都遵循如下过程: 先由PC指针从存储器中取来指令, 经译码分析或从CPU的内部寄存器或从存储器中取来操作数, 按指令操作码的规定, 在CPU中完成相应的操作, 其操作结果通过内部寄存器最终存放在存储器中。

(5) 最后, 由用户程序规定可随时从存储器里取出结果数据, 通过系统总线的传送, 经输出接口输出到相应的输出设备上。

1.2.2 系统总线

由图1-1所知, 微型计算机内各功能芯片的互连是采用总线结构的方式。

就总线分类而言, 一般分成以下三种:

(1) 内部总线

又称片级总线或局部总线。它由CPU芯片及其它外围芯片组成的一个单板微型机或一块CPU系统硬件板使用的板上总线, 用于芯片一级的互连。

(2) 系统总线

又称板级总线或内总线。它是微型计算机系统特有的一种总线。用于系统中各插件板之间的连接, 便于系统的扩充和更新。

(3) 外部总线

又称通信总线或外总线。它用于微型计算机系统间的通信网络或用于微型计算机系统

与电子仪器或其它设备间的互连。

前二种总线的格式标准随微型计算机系统制造厂商而定,但目前较为流行的系统总线,是由美国IEEE作为标准草案推荐的有两种:一个是S—100总线(修订后取名IEEE696),另一个是Intel公司的多总线(Multibus)。第三类总线不是微型计算机系统所特有的,在微型计算机的应用中,较为著名的外部总线有IEEE488总线和EIA-RS232c、EIA-RS 422等。

图1-2表示了三种总线在微型计算机系统的位置及相互关系。

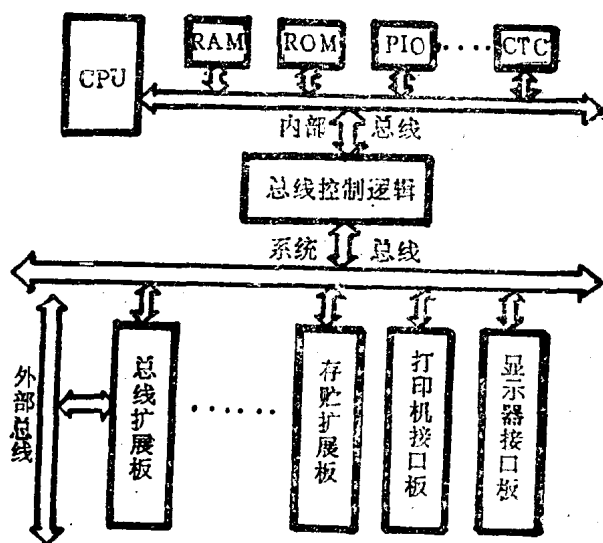


图 1-2 微型计算机系统三级总线

为源端)传送,且为三态控制。

(2) 数据总线

在微计算机系统中,用于传送数据的信号线称之为数据总线。通常,8位微型计算机有8根数据总线,16位微型计算机有16根数据总线。

数据总线均是双向(CPU和外围芯片之间,系统板与接口板之间均可来回传输数据)也采用三态控制。

(3) 控制总线

在微型计算机中CPU对外围芯片和I/O接口的控制以及这些芯片接口对CPU的应答、请求等信号组成的总线称之为控制总线。

在同一台微型机中,控制总线根据使用的环境有单向或双向之分,有三态或双态之分。这是总线中最复杂、最灵活、功能最强的一类总线。其数量随机型的各异而不同。

(4) 电源线和地线

电源的种类视微型计算机的不同特性而变。常用的电源有 $\pm 5\text{Vdc}$, $\pm 12\text{Vdc}$ 等几种。为减少信号噪声干扰和失真,总线中的地线分布和使用是很讲究的。

(5) 备用线

留给制造厂家功能扩充或用户特殊要求的使用。

图1-2中的总线控制逻辑实现总线请求的仲裁。它是统一管理总线的的一个部件。因为

微型计算机系统由于采用了总线结构,因此,芯片间、插件间及系统间信号的传输都由总线提供信息通路。按信息的类型分,总线中的信号大致为以下五种类型:

(1) 地址总线

微型计算机用于传送地址信号的信号线称之为地址总线。CPU芯片能直接寻址的范围即决定了地址线的数目。若8位CPU的地址输出线为16条,则寻址范围是64k字节。16位CPU的寻址范围通常是兆数量级,因此,地址总线的信号线至少需20根。

地址总线均是单向(由CPU作

不论是单处理机微型机系统还是多处理机微型机系统，系统总线只有一条。任何时刻它只能被一个主设备（单机系统）或一个处理机（多机系统）所占用。换言之，该主设备或该处理机取得总线的控制权，因而，其它的设备或处理机就无法占用总线。如果，在某一时刻有几个设备或几个处理机都想获得总线的控制权，那只得依靠总线控制逻辑对提出总线请求的各个设备或各个处理机按优先级的排队次序进行仲裁判优，让同时提出申请的设备或处理机中优先级较高的占用总线；等待优先级高的设备或处理机使用完总线，再收回总线控制权让位给提出申请的那些优先级较低的设备或处理机。

下面，以IEEE696总线裁决逻辑为例，说明总线控制的过程

图1-3展示了一个永久性主设备和16个暂时性主设备之间经过裁决而获得总线控制权的控制逻辑。图中的符号名称说明如下：

HLDA：永久性主设备对获得优先的暂时性主设备总线请求的响应（高有效）。

HOLD：暂时性设备向永久性主设备提出的总线请求（低有效）。

BREQ：暂时性主设备的总线请求。（高有效）。

MINE：暂时性主设备获得总线控制权（高有效）。

ISME：经总线仲裁逻辑，优先级高的暂时性主设备获准总线请求（低有效）。

BR₃~BR₀：总线请求线的优先级值（自1111至0000共十六种六状态）。

总线裁决的过程是这样

的。

(1) 初始状态时HLDA信号为低，表示总线在永久性主设备控制之下。若此时HOLD为高，意味着没有暂时性主设备提出总线请求。

(2) 当某一时刻，有一个或一个以上的暂时性主设备同时提出总线请求时，即BREQ为高。这样，满足三个条件 ($\overline{\text{HLDA}} \cdot \text{HOLD} \cdot \text{BREQ}$) 使与非门1产生有效的低信号。通过D₁的R端建立HOLD信号有效即向永久性主设备提出总线请求；同时，通过D₂的S端置触发器D₂为1，允许提出请求的设备将其优先级送至仲裁总线ID₃~ID₀。（由图中开关设定，开关闭合表示0）。

(3) 优先级裁决电路在图中下方。这个逻辑实质是二进制减法器的预借位电路。ID₃~ID₀的二进制值愈大其优先级愈高，而BR₃~BR₀传送的是具有较高优先级ID的反码。

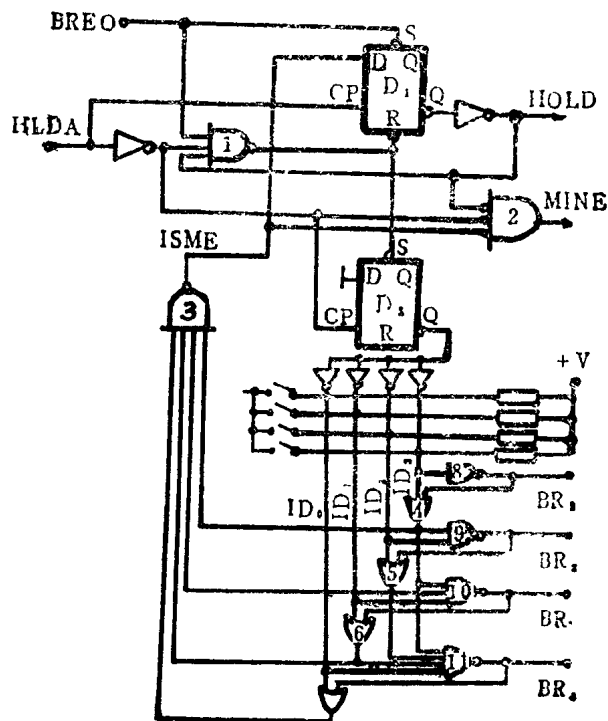


图 1-3 总线控制逻辑

注：8, 9, 10, 11是集电极开路门

从当前ID的值减去BR的值，产生四个低借位有效信号。但只有具备当前最高优先级的ID请求，才能保证四个借位信号无效（高），使与非门3输出有效低信号。而优先级较低的ID值比较之后会产生低电平借位，从而屏蔽借位以下的所有集电极开路门，使与非门3输出为无效高电平。

(4) 一旦某个设备的ISME信号为低，则当永久性主设备时总线请求作出响应时（即HLDA有效为高），利用HLDA的上升边沿，将当前最高优先级设备的D₁触发器继续为0，使HOLD保持有效，而最终建立总线控制权信号MINE有效（高）。

(5) 与此同时，未建立起ISME信号的请求者，则在HLDA信号上升沿作用下，使各自的D₁触发器复位为1，相应的HOLD信号被撤除，等待下一次判优周期再参与竞争总线时建立。

(6) 当掌握总线控制权的那个暂时性主设备使用完总线时，便释放BREQ总线请求，使HOLD信号无效（变高），并清除总线获准信号MINE，使之复位为0。此时，永久性主设备撤除HLDA信号，收回总线控制权。利用它的下降沿使D₂触发器复位为0，清除仲裁总线ID₃~ID₀上的优先权值，等待下一个仲裁周期的到来。

在实现总线控制权转移的过程中，利用MINE有效信号去控制其它相应的选通信号，使永久性主设备的地址、数据总线的驱动器无效（即处于高阻抗态），而使暂时性主设备的地址、数据总线的驱动器有效。

最后，归纳一下微型计算机采用总线结构的几个优点：

(1) 简化软件、硬件的设计

对硬件设计师来说，只需按总线接口规范设计CPU插件，存储器插件和I/O接口插件。在组装微型机时，将这些插入总线即可工作。由于这些插件具有互换性与通用性，便于厂家大批量的生产和组装调试。

对软件设计师来说，插件式的硬件结构使软件编制采用模块化结构，便于调试和修改，节省软件的开发周期。

(2) 使系统便于功能扩充和更新

在总线上多插几个同类型的插件板，即可实现规模上扩充，而按总线标准设计新的插件，便可在功能上实现扩充。对于系统的更新，只需将所更新的插件板插在原来的位置上。

(3) 适应计算机应用的需要

由于微型计算机应用领域极为广泛，各行各业的用户都会对计算机接口提出各种各样的要求。若有了某种标准的总线接口，那不同的用户可根据自身的需要，购买相应的接口插件，组装成适合应用要求的微型计算机系统。采用总线结构就提供了这样的灵活性。

1.2.3 中央处理器CPU

采用LSI技术将运算器和控制器集成在一块芯片内组成的中央处理器，其典型的基本结构如图1-4所示。

对照图1-4，下面分别说明各部件的功能。

一、指令控制部件

该部件的功能是取指令、分析指令并发出相应的控制信号。它由以下各部分组成。

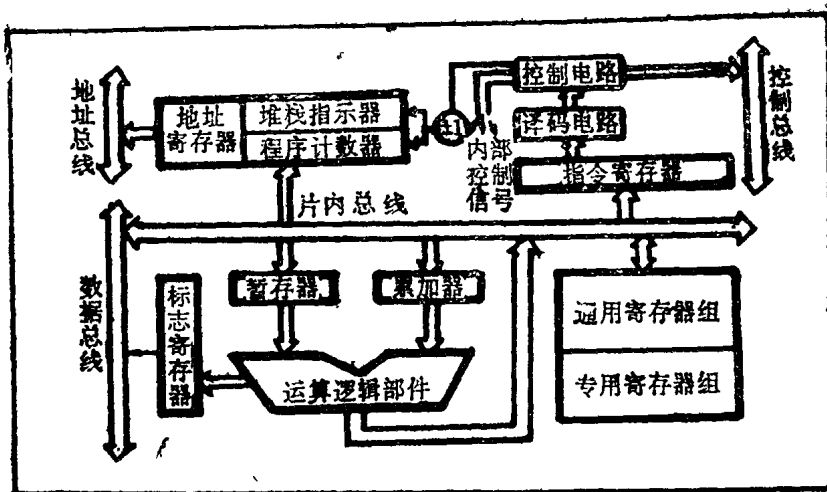


图 1-4 典型的CPU基本结构

(1) 程序计数器

又称指令计数器。它包含下条将要执行的指令地址，且具有计数功能。在顺序执行期间，一条指令取来时该计数器就自动加1或加2。这取决于一条指令的长度占1个字节或2个字节。当指令的长度占几个字节时，则要连续地读取几次。而在转移执行期间，把转移的指令地址直接送入该计数器来改变程序的执行顺序。

一般而言，程序计数器的长度（位数）要比微处理机的字长来得长。因为，前者给出的指令地址直接反映CPU的寻址范围。在8位微处理机中寻址空间达64k字节，则计数器的位数应16位，因此，在改变程序执行顺序时，需两次把8位数据分别置放在计数器的低字节和高字节。在16位微处理机中，寻址的范围往往在几兆甚至十几兆字节，但程序计数器一般也取16位，它所指出的指令地址并非内存的绝对地址，而只是指令代码段内的相对地址。这样，只需设置一个代码段寄存器专门指示代码段的起始地址，将代码段寄存器和程序计数器放在一起连用，便能寻址整个兆字节数量级的空间。

(2) 指令寄存器

它存放当前正在执行的指令。根据程序计数器指出的指令地址，从存储器中取出的指令经数据总线传送给该寄存器。

指令寄存器的长度与指令的长度有关。但在设计中希望指令的长度愈短愈好。对于种类繁多，功能各异的指令其长度采取不同的措施区别对待。因而，一般取指令寄存器长度与微处理机的字长相同，这样，既可减少指令访问存储器的次数，提高指令的执行速度，又可节省存储空间。

所采取的措施通常是采用累加器操作、隐含地址、堆栈指令以及按功能区分指令长度等。具体细节请参见第四章指令系统有关内容。

(3) 指令译码器

译码器的功能是对存放在指令寄存器中的指令按操作码识别指令的操作类型。按地址码分解源操作数地址或目的操作数地址。

(4) 指令控制电路

根据译码的结果，产生执行此条指令所需要全部时序和控制信号。

这些控制信号分内部和外部两类。内部控制信号是操纵CPU内各部件的动作，外部控制信号是实现CPU与外围芯片（包括存储器芯片和各类I/O接口芯片）进行信息的通信。后者组成控制总线。

控制总线的控制信号有如下几类：

- 时钟输入CLK
- 系统复位RESET
- 准备就绪READY、系统等待WAIT
- 总线请求/响应HOLD/HLDA
- 存储器读/写RD/WR
- 存储器与I/O控制IO/M
- 中断请求/应答INT/INTA
- 非屏蔽中断请求NMI
- 地址允许AEN
- 数据允许DEN
- 数据发送/接收DT/ $\overline{R}$
- 系统状态SS

上述列举的只是最基本的控制信号，除外，随微处理器的功能差异其控制信号也略有不同。

二、算术逻辑运算部件

该部件完成算术逻辑运算指令所指定的运算操作，并将操作结果通过内部数据总线传送到指令地址码所规定的目的地址中。它主要由以下各部分组成。

(1) 算逻部件

该部件简称ALU (Arithmetic-Logic Unit)。主要完成运算指令所指定的运算操作。参加运算的两个操作数，一个称源操作数或取自存储器或取自寄存器，存放在暂寄存器中；另一个称目的操作数隐含在累加器中。运算的结果或隐含在累加器中或按目的地址传送。

(2) 累加器

在微处理机中往往使用累加器来缩短指令的长度和减少指令访问存储器的时间。把累加器作为参加运算的一个操作数源地址并且是操作结果的目的地址。这对于求一批数据的累加和是很方便的。然而，在使用累加器指令时，需在事前使用“装入累加器”指令，事后使用“存贮累加器”的指令，以免发生错误。由于累加器指令的方便、快速的优点，在程序编制过程中要经常用到。为避免累加器成为数据通路的“瓶颈”，有的处理机设置多个累加器，有的处理器设置通用寄存器组来代替累加器。

(3) 标志寄存器

这是一个十分重要的寄存器。在所有的微处理机中是不可缺少的。它反映一条指令执行后操作结果的特征和CPU的内部状态。前者称为状态标志，后者称为控制标志。标志位的长度和各位的功能随CPU不同而变化。

下面列举的标志位是最基本的。

状态标志有：

- 进位标志 (C)：操作结果的最高位产生进位则置1，否则置0。
- 零标志 (Z)：操作结果为全零则置1，否则置0。
- 符号标志 (S)：操作结果为负则置1，否则置0。
- 奇偶标志 (P)：操作结果 (8位) “1” 的个数为偶数 (偶校验) 或奇数 (奇校验) 时则置1，否则置0。
- 溢出标志 (O)：操作结果超出表示范围溢出则置1，否则置0。
- 辅助进位标志 (AC)：在二一十进制 (BCD码) 操作中，结果的低位数字产生向前进位则置1，否则置0。

控制标志有：

- 允许中断位 (IE)：允许CPU响应外部可屏蔽的中断请求则置1，否则置0。
- 跟踪方式位 (T)：CPU执行单步跟踪方式则置1，否则置0。
- 管态方式位 (M)：CPU在管理状态下运行则置1，否则置0。

有关标志寄存器的具体细节请参见第三章8088的结构的相关内容。

标志位的用处是多方面的。状态标志反映了上一条指令执行后结果的特征，那紧接着的下一条指令可根据C、Z、S、P、O等标志的不同值对程序的流向进行判断和转移；而控制标志的设置能控制系统在不同的方式下工作。如在程序调试时置单步执行方式，则系统每执行一条指令即产生一次内部单步中断，能把CPU的内部寄存器值全部显示；如在管态方式下，所有的指令都可运行。在开发系统软件时能使用管态指令 (又称特权指令)，而在运行用户程序时，禁止使用这些特权指令以保护系统的安全性。

最后，要注意并非所有指令执行后都对状态标志位产生影响。有的部分地或全部地产生影响，有的如传送指令对标志位根本没有影响。因而，在编制程序时，要了解和熟悉各条指令对标志位的影响程度 (一般在指令系统的说明书中均可查到)，使汇编程序的运行不致发生错误。

三、寄存器组

除了上面提到的程序计数器、指令寄存器、累加器、标志寄存器和以后提及的堆栈指示器外，在CPU内部设置了一个寄存器组，专门用于存放操作数据和地址。类似于存储器中的一批存贮单元，也用相应的寄存器地址来识别每个寄存器。

显然，在内部设置寄存器组可减少系统访问存储器的次数和减轻单总线的使用压力，从而提高了指令执行的速度。

寄存器组内的个数随CPU性能的提高而增加，常见的在8个至16个之间。寄存器的个数是受制于指令长度的约束。因为，设置寄存器的目的是希望指令快速执行，势必要求指令的长度愈短愈好。而一条指令既包括操作码又要指出源寄存器和目的寄存器的地址，因此在8位微型机中，若各用三位指示源/目的寄存器地址；在16位或32位微型机中，若各用四位指示源/目的寄存器的地址，那在8位和16位的指令长度内已经是相当可观了。

寄存器组的用途不外乎有以下几种：

- 数据寄存器：存放操作数或操作结果。

- **地址寄存器：**存放操作数的地址。
- **变址寄存器：**专门用于变址寻址操作。
- **基址寄存器：**适用数据块操作的基址。
- **累加器用途：**隐含操作数或操作结果。

寄存器的长度取决于它的用途。若作为数据存取，寄存器或以字节为单位或以双字节为单位，有的甚至组合成32位字长的寄存器。如8088/8086微处理机，四个16位字长的寄存器AX、BX、CX、DX。若取高字节操作，则命名为AH、BH、CH、DH；而对低字节寄存器，取名是AL、BL、CL、DL等、以适应不同环境下运算。但对于存放操作数地址，变址、基址等用途时，则一概以16位长度计，而不能使用8位。

四、堆栈和堆栈指针

在存储器中专门开辟一个数据存贮区（其位置和长度均用软件设定），以先进后出的原则有序地存取一批数据，这个数据有序集合体称之为堆栈。

数据存入堆栈称为压入（PUSH），数据取出堆栈称为弹出（POP）。为保证数据的存取不改变堆栈中原来数据的位置，用一指针动态地指示栈顶的位置，使压入或弹出操作均在由该指针指向的栈顶中进行。这个指针称为堆栈指针，用一专门寄存器SP命名。

堆栈地址的增长常见的是向下生长，即栈底位于高地址区，栈顶位置随着压入和弹出动作而变化。图1-5展示了栈顶的变化。（图中的地址值是任选的）

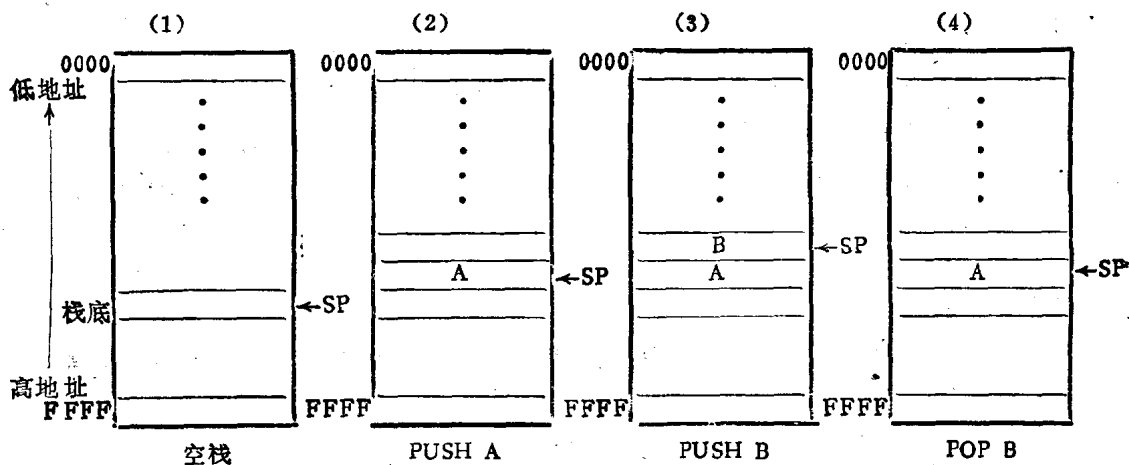


图 1-5 栈顶在压入/弹出时的变化

从图中可知，SP始终指向栈顶位置。当栈空时，栈顶位于栈的底部。当压栈时（图中（2）、（3）），栈顶SP先减后压入数据；当出栈时（图中（4）），先弹出数据栈顶SP后加。

堆栈主要用于中断和子程序服务。当发生中断或调用子程序时系统自动利用堆栈，将断点地址和其它需保护的现场（各类寄存器值）有序地进入堆栈；而在中断返回或子程序返回时，系统又自动地按进栈的反顺序一一地弹出数据，既恢复了被中断的现场，又可继续执行被中断的原程序。图1-6反映了在调用子程序服务过程中的堆栈变化。