

四川省 自动化与仪器仪表学会 资料
科学仪器专业委员会

智能仪器设计

——微处理器在仪器仪表中的应用

上 册

成都科学仪器厂
科学仪器专业委员会
成都电讯工程学院

上册目录

序	1
绪论	3
1. 硬联接逻辑和可编程逻辑电路	3
2. μP 的特点与优点	4
第一章 微处理器系统概观	
1.1 数制和数字代码	8
1.2 微处理器系统的基本组成	11
1.3 MPU的内部组织	13
1.4 外围设备接口和I/O操作	15
1.5 存储器的访问	17
第二章 微处理器的结构和操作	
2.1 典型的微处理器结构	20
2.2 微处理器的基本操作	29
2.3 微处理器的指令系统	31
第三章 8080/8085和Z-80 微处理器	
3.1 概述	43
3.2 微处理器结构	44
3.3 8080微处理器的指令系统	50
3.4 8080中央处理器模块的组成	60
3.5 8080的定时及指令的执行	63
3.6 8085微处理器	72
3.7 Z-80微处理器	81
第四章 MC6800微处理器和MCS6502微处理器	
4.1 概述	97
4.2 MC 6800的基本特性	98
4.3 MC6800MPU的内部结构	98
4.4 MC 6800所使用的寻址方式	100
4.5 指令集	103
4.6 MC6800 MPU的外部信号	114
4.7 MC6800的时钟和指令执行	116
4.8 6800系列的时钟器件	122
4.9 MCS6502 MPU的特性	124
4.10 MCS6500 MPU内部结构	125
4.11 MCS6502使用的寻址方式	128

4.12	MCS6502MPU的指令集	129
4.13	MCS6502 的外部信号	135
4.14	6502 的时钟信号及指令执行	137
第五章 半导体存储器		
	引言	139
5.1	半导体存储器在微处理器系统中的应用	139
5.2	随机存取存储器	139
5.3	只读存储器	140
5.4	微处理器与存储器的连接	141
5.5	地址译码	147
第六章 输入/输出		
6.1	基本问题	153
6.2	输入/输出接口的编址与控制	153
6.3	输入/输出接口与外设之间的挂钩	155
6.4	Intel 8255可编程的外设接口	157
6.5	motorola 6821外设接口适配器	164
6.6	中断	173
6.7	Intel 8080/8085 的中断系统	174
6.8	motorola 6800/6802 的中断系统	178
6.9	点名查询的中断系统	181
6.10	多矢量中断系统	184
第七章 编程入门		
7.1	软件系统	189
7.2	编程要领	189
7.3	数据的传递	190
7.4	并行接口的I/O	193
7.5	串行接口的 I/O	198
7.6	串行 I/O及换码编程举例	201
7.7	并行 I/O编程举例	204
7.8	计算和打印程序举例	207
7.9	延时程序	209
7.10	交通灯自动管理程序	210
附录A	8080的指令系统	213
B ₁	按功能分类的Z-80CPU指令表	217
B ₂	按记忆码分类的Z-80CPU指令表	230
B ₃	按操作码分类的Z-80CPU指令表	237
附录C	MC6800指令表	244
D	6502指令表	247
E	指令对照表	248

序

微处理器 (Microprocessor), 通常简称为 μP , 这是一种用大规模集成电路 (LSI) 实现的可程序逻辑电路。它的突出优点是小巧、灵活、方便、可靠、省电、价廉。自从1971年正式问世以来, 发展非常迅速, 到1978年各种不同的微处理器型号已超过了200种。目前, 微处理器应用, 从先进的科技部门如电子计算机、雷达、通信、测量仪器等, 以至工业过程自动控制、电子医疗护理系统等等, 甚至直到家用设备如汽车、电视机、录音机、缝纫机、洗衣机以至儿童玩具等等, 举凡新一代电子设备, 都几乎全部用上了微处理器。有人把这看成是“第二次工业革命”, 而其影响所及则远远不止限于工业。

迄今国内外已出版了大量有关微处理器的书籍, 国内也举办过不少微处理器的学习班。然而, 一般大都是从如何利用微处理器来组成微型计算机这样一个角度来讲述的。本书则试图从另一个角度来阐述, 即是把微处理器看作为组成电子设备的一种基本电路元件, 如何把微处理器用于电子设备的实际设计中。

本书中的原理阐述, 不限于一种具体的微处理器型号, 意欲使之具有一定的普遍适用性。然而在许多地方, 却又不可能脱离具体的微处理器来说明; 为此我们较为详细地介绍常见的两大类微处理器类型作为举例, 一类是Intel 8080型的微处理器, 稍后问世的8085及Z80等基本上属于这一类型, 在本书中亦有所兼及。另一类是Motorola MC—6800型的微处理器, 后来的MCS—6500也属于这种类型, 本书亦兼及之。

本书中的许多实用举例, 以电子仪器的设计为主, 电子仪器是“麻雀虽小, 肝胆俱全”。其中所述, 对于其他电子设备的设计, 亦同样适用。电子仪器是最早应用微处理器的电子设备, 早在1972年就有布恩通 (Boonton) 公司生产了使用微处理器的76A型自动电容电桥。电子测量仪器中也比较充分利用了微处理器的潜力, 例如建立误差模型由微处理进行误差改正, 从而可以大大降低模拟式电路硬件的要求; 利用适当的算法来使得模拟式电路硬件超过其原有的性能特性; 凡此种种, 在其他电子设备中是不很常见, 却是大可供作借鉴的。

本书分为上、下二册。上册为《实用微处理器基础》, 这是为未曾学习过微处理器的读者编写的。内容包括:

1. 微处理器系统概观
2. 微处理器的结构和操作
3. Intel 8080/8085/Z80系列微处理器
4. MC—6800/MCS—6500系列微处理器
5. 存储器
6. 输入输出接口

7. 编程入门

下册《智能仪器设计》是为对 μP 已有一定基础的读者写的，在本书上册内容的基础上，下册进一步阐述 μP 在仪器实际应用中的一系列关键问题。目的在于为读者提供解决实际设计问题所必需的知识，使他们领会如何将 μP 用于他们的设计，来代替一大堆逻辑电路；如何利用 μP 来改善模拟式硬件的性能，或降低对硬件的技术要求；如何利用算法来扩展设备的功能；如何实现设备的自动化和智能化。为此，下册包括下列各方面的内容：

1. 设计的基本问题
2. 输入输出技术
3. A/D和D/A变换及数据采集
4. 软硬结合的若干特殊技术
5. 通用接口的设计
6. 自检测和告警
7. 误差模型和误差改正
8. 测量算法和功能扩展
9. 程序编制的若干问题
10. 智能仪器的设计
11. μP 系统的开发、仿真和调试
12. 附录：常用子程序汇编

本书由成都电讯工程学院电子仪器与测量教研室的张世箕、郭成生、杨安禄、王树菁、陈光耦分章执笔。本书的这种写法，可以说是一种新的尝试。由于作者们的水平和经验所限，不当之处尚祈读者惠予指正，于此预致谢意。

本书承四川省自动化与仪器仪表学会科学仪器专业委员会及成都科学仪器厂负责出版事宜，才能促成本书与广大读者见面。学会的宋水庆、孟宪武、欧阳亦金等同志功劳尤多。于此一并致谢。

作者

1982年3月于成都

绪 论

微处理器 (Microprocessor), 常简称为 μP , 是一种用大规模集成 (LSI) 工艺实现的可编程逻辑电路。自从1971年开始问世以来, 发展非常迅速。目前, 几乎任何一种新一代电子设备中, 都可以看到微处理器的应用。在电子仪器中的应用尤为普遍。

不同的人对微处理器有不同的理解。许多人把微处理器系统看作是微型电脑, 或者看成是一种微型计算机。而许多电子设计工程师则把它看成是构成电路或电子设备的一种万能的基本组成元件, 用它来取代许多晶体管、电阻、电容、开关器、等等。

看起来, 微处理器的体系结构 (存贮、计算和控制单元的安排) 似乎同小型计算机没有根本的差异, 而且小型计算机所用的一切编程技术和手段 (如汇编程序、编辑程序乃至高级语言、等等) 亦同样可以被采用于微处理器。然而, 微处理器毕竟不全等于微计算机, 前者只不过是后者中的一个组成元件, 还要配上一系列的随机存取存贮器 (RAM)、只读存贮器 (ROM)、各种接口电路以及适当的外围设备, 组成成为一个系统, 这才成为一部真正的微计算机 (现在也出现了除外围设备外全部容纳在一片甚大规模电路集成块内的微计算机)。此外, 正如许多实际应用所表明的, 微处理器的作用并不仅限于作为一种微计算机。

另一方面, 单纯把微处理器看作为一种大规模集成的电路元件, 而忽视了它在逻辑和计算方面的强大能力, 也同样是失之偏颇, 从而将不能更有效地发挥微处理器作为万能元件的潜在能力。

为此, 要正确而有效地发挥微处理器的作用, 把它用作为电子设备中的主要组成部分, 设计者对于微处理器上述两方面 (作为电路元件和作为计算机) 的特点, 都必需有足够的了解。目前, 大多数有关微处理器的书籍, 大都是从微计算机这一角度来阐述的。对于不熟悉电子计算机科学的电子工程师来说, 这些书籍当然是大有裨益的。然而, 这还不能完全解决他们的问题: 如何实际把微处理器用于他们的设计? 特别是如何用一块微处理器片子 (外加适当的支持片子) 来取代传统的一大堆逻辑电路; 如何利用微处理器系统来提高模拟式硬件的工作性能特性, 或者反过来在获得相同的性能的情况下降低对硬件的技术要求; 如何用微处理器来实现对设备各部分的控制, 从而实现自动化和智能化; 如何使设计出来的机器灵活方便, 达到一机多能。

本书的目的, 就是试图给予读者解决这许多实际设计问题所必需的适当基础。

1. 硬联接逻辑和可编程逻辑电路

在未有微处理器以前, 我们用一些能实现基本逻辑和基本算术运算的电路来构成数字系统, 如“与”门、“或”门、比较器、加法器等等, 再加上一些多路选择器、多路分配器、移位寄存器等。这样的数字系统, 或其局部电路, 一旦设计制造出来之后,

就成为一种有着固定联接而不能随时随意改变的系統。这就是所谓固定联接或“硬”联接(hard-wired)逻辑电路。

随着电路集成化的规模日益增大,电路的体积、重量、耗能以及价格日益下降。因内部联接线缩短,而性能亦相应提高。由于外部联线大减,因而可靠性也就大为改善。另一方面,大规模集成电路中各单元部分之间的相互联接,都封装在集成块之内,这种联接就更“硬”了,更加不可能改动了。集成电路设计成什么样子,就是什么样子;它只能实现一定的预定功能,无法改作别用。如要作他用,就只好另行设计制造一种具有所需其他功能的集成电路片子。随着集成电路集成规模的增大,这种死硬而不灵活的缺点就更加突出。

可编程逻辑电路则不然,它正好是为了克服上述的缺点和困难而被创造出来的。微处理器作为一种大规模逻辑电路,它内部的实际电路联接也是固定的。然而,它所实现的功能,却可以很容易地通过其执行程序的编制而使之改变。正如电子计算机那样,用一个加法程序就使它实现加法器的功能;用另一个程序可使它变成一个积分器。微处理器就是这样的一种大规模集成的可编程逻辑电路,它具备了一般硬联接大规模集成电路的一切优点,但却避免了死硬不变的缺点。因而其使用就更加灵活方便而且经济。

通常所说的微处理器片子,有二种含义。狭义的微处理器片子,是指微处理器单元(MPU)它相当于电子计算机内的中央处理器单元(CPU),通常包含有一个算术逻辑单元(ALU)和控制器,以及适当的累加器、寄存器等。广义的微处理器片子,则泛指MPU片子以及与之配套联合工作的一系列大规模集成电路片子,如ROM、RAM、时钟以及其他许多支持电路,如输入输出(I/O)片子及各色各样的接口片子等等。以MPU为中心,用这些广义的微处理器片子为主(间或添上若干中、小规模集成电路)而组成的电路系统,称为微处理器系统。在此基础上设计出来的仪器,称为以微处理器为基础的仪器。这里是指广义的仪器,雷达是一种测量位置和速度的仪器,电视是一种绘图仪器,等等。

现在已出现了把一整个微处理器系统集成在一块甚大规模(VLSI)集成电路中的产品,常称为单片计算机(μC),不过它并不是真正的微计算机,因为它没有包括计算机的外围设备。一个 μC ,是一个 μP 系统;但一个 μP 系统却并不一定是一个 μC 。

μP 或 μP 系统可以成为一个不带外围设备的 μC ,或者可以成为计算机的一个组成部分(例如接口电路、外围控制器等),也可以成为外围设备(例如一个电传打字机或CRT显示器)中的电路的主要构成部分。

2. μP 的特点与优点

μP 的基本特点是它的工作的可编程性。一个MPU在一定的指令作用之下,能完成一定的基本操作,例如,加、减、左移一位、右移一位、“与”、“或”、等等。在一个时刻,微处理器只能执行一条指令,完成一种基本操作。指令与操作是一一对应的,即是对应于一条指令,只有一种唯一的操作。这些基本操作或指令是相当有限的,而且各种不同型号的微处理器也不尽相同。一个微处理器所能执行的指令的总体,称为该微处

理器的指令集。例如Intel 8080型微处理器的指令集包含78条指令；Motorola 6800型微处理器的指令集则含有83条指令。微处理器在工作时，按照一个接着一个的指令依次执行，完成一系列有序操作，来实现某一特定功能。这种有序的指令序列就称为程序。利用不同的程序编排，就可使微处理器实现不同的功能。尽管指令集颇为有限，然而通过不同的程序编排，却可以实现千变万化的功能。例如，一般微处理器中都只有加法操作而无乘法操作，但我们却可以通过对同一数的连续多次相加来实现乘法功能。如欲得到 4×3 ，我们可以令微处理器这样来工作： $4 + 4 + 4$ 。对于78或83条指令，取一切不同的排列，即可分别得到

$$78! \approx 10^{115} \text{ 和 } 83! \approx 4 \times 10^{124}$$

种样式。这里还未计及这样的情况：任一指令在程序中都可以重复多次，而程序的长度也远远不限于七、八十条指令。此外，也未计及同一指令的操作数的千变万化，所谓操作数，就是操作的对象，例如加法操作中的被加数，这是可以容许在一定限度之内（例如8位或16位二进制数）千变万化的。这样，实际上我们有可能编排出无穷多种不同的程序，从而使微处理器有可能实现无穷多种的功能，达到随心所欲的地步。

正由于此，我们就有可能利用为数不多的集成块组成的一个微处理器系统，来代替一大堆硬联接逻辑电路，而且还可以相当随意地改变它所实现的功能，以达到一机多能的目的。

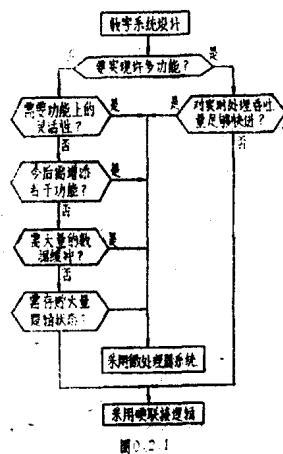
事先编制好一系列的程序存放在微处理器系统内，就可以达到机器的自动化。在程序中充分利用指令集中的逻辑运算和判断能力来编出适当的程序，就能使机器实现智能化。

既然可以只使用为数不多的集成电路片子来构成一个系统，所以整个系统的体积、重量、耗能以及价格也就相当有限了。此外，这些为数不多的集成块彼此之间的外部联接线也就比较有限，从而也就大大提高了系统的可靠性。

由此可见，采用微处理器，优点是很多的。

另一方面，微处理器既然是一种可编程逻辑电路，所以它也就属于序列逻辑的范围。要它完成一指定的功能，就要用一个程序来使它按部就班地依次作出一序列的逻辑操作，这就需要一定的时间。换言之，与为了一定功能目的而专门设计的硬联接随机逻辑电路相比较，微处理器的工作速度就比较慢，甚至可能慢很多。这就是微处理器的主要缺点之所在。然而，对许多实际应用而言，微处理器的工作速度仍然是足够快的。

从以上所述，可以看出，微处理器具有较多的突出的优点，对于一般电路设计而言，采用微处理器是一种较好的选择，然而也不一定是唯一可行的选择。至于它的缺点，在许多情况下完全可以置之不顾，然而在某些要求快速实时处理的情况下，速度慢的缺点则可能是致命的。对于一个特定电路或设备的设计，到底采用或不采用微处理器，将取决于设计所要求的多方面特性。大体上，不妨按照图0.2.1所示进行选择。



3. 以UP为基础的电子设备的优点

除了上述从电路结构特性方面所看到的微处理器的一些优点之外,从具体应用的角度来看,采用微处理器系统的电子设备还可能具有如下的一些优点。

(1) 使用简便

利用微处理器系统可以通过预先编好的程序实现自动控制和智能化的能力,可以大大减少设备所需的人工控制,设备面板上的按钮数目可以大为减少,使得设备的操纵非常简单。这就使得有可能由非熟练人员、甚至非技术人员来使用相当复杂的电子设备,遇到操作员的操作有误时,机器还可以提出警告,如亮灯、鸣笛、显示文字等。必要时,电子设备还可以用适当方式告诉使用者应如何操作,例如显示出文字说明、绘出图形等。在操作出错时,可发出出错消息,指出发生了什么错误,乃至应如何改正错误等信息。此外,在机器工作的过程中,也有可能利用简单的按键随时更改机器的复杂工作。

(2) 输入输出方便而多样

在使用微处理器系统的情况下,有可能针对不同的具体情况采用适当的方便方式来输入控制指令或数据。例如,使用拨轮、专用按键(特定功能按键)、通用按键等。例如一组0~9的十个按键,可用以输入任何位数的任何十进值;又如用一个电传打字机式键盘可以输入任何文字和数字信息。还可以在CRT屏幕上开列出可供任意点用的项目清单,用光笔指向某一项目或用手指点在该名目上,机器即可遵命执行。较长的程序或数据可以事先录在穿孔卡片、穿孔纸带、磁带或软磁盘上,需用时即可由机器自动输入,或者随意或自动寻找其中某段程序来执行,或寻找某一组数据以供机器内部使用或输出。即使图形也可以有方便的手段输入。例如用一个数字化器(digitizer),把光标移到图形的某一点上,一按按钮即自动把图形该点的座标自动编码后输入到机器中。新型的绘图仪中不少也兼有作为数字化器使用的能力,而这种能力也是利用微处理器系统来实现的。

至于输出,也同样可有多种多样的方式。除简单的指示灯外,有用七段数码管显示数字的,有用点阵字符显示器显出字母、数字和符号的,有在CRT屏幕上显示文字和图形的,有利用绘图仪来描图并写字的,有用打印机打出文字、数表和曲线的,等等。输出信息也可以穿在卡片或纸带上,录于磁带或磁盘中。还可能产生声音输出,甚至还有能听懂人语命令并能讲话的。也可以利用光笔指着CRT显示出来的曲线上某一点,而使机器用文字显出该点的具体数据达到六位或十二位有效十进数。诸如此类,不胜枚举。

(3) 一机多能

如前所述,充分利用可编程逻辑的基本特点,不难实现一机多能。特别是在测量仪器中,可以籍助于间接测量而获得很多测试功能。仪器可以设计成只测量为数不多的几个基本参量的较为简单的仪器,而充分利用微处理器系统的运算能力,从少数基本参数

换算出许许多多各种各样的导出参数来。

(4) 降低硬件要求, 提高性能特性

利用可编程逻辑不仅可代替一大堆硬联接逻辑电路, 而且还可以代替若干模拟式硬件, 并可能获得更好的性能。例如, 以数字滤波过程代替模拟式滤波器就是富有成效的。除此以外微处理器系统与适当的模拟式硬件配合工作, 还可以大大提高模拟式硬件的性能特性, 或使之远远超出它本来使用的极限。在电子仪器中, 这通常通过建立误差模型来施行误差修正的方法来达到这一点。例如, 各种传感器的动态范围与其输入输出变换的直线性之间常有矛盾。利用微处理器系统进行误差修正, 可以容许传感器工作远远超过其正常线性动态范围。尽管这样做可能导致很大的非线性而致的误差, 不过可以事先测定其偏离线性的误差数据, 存贮起来, 而在测量时逐点对测量结果进行修正。这些工作都可以由微处理器系统自动完成。这样一来, 甚至可能获得比普通单纯的传感器在更窄动态范围内的精确度更高的精确度。也可以使用动态范围较窄而且直线性较差的传感器, 而仍可以获得相当于动态范围很宽而且直线性很好的结果, 甚至还超过良好的传感器所能得的结果。又例如, 元、器件的漂移或噪声等对测量结果的影响, 可以通过大量反复测量(由于仪器的自动化, 不难在很短时间内作完), 然后由微处理器系统对这些存贮下来的大量数据作统计处理。这样即可消除漂移或噪声的影响, 而获得很高的测量精确度。在此情况下, 使用质量较差的元器件也就无妨了。总之, 微处理器系统在降低对硬件的要求以及提高设备的性能特性方面, 是大有可为的。

还可以列举出其他一些优点。然而, 仅此数端, 已足见微处理器系统的应用所带来的无可比拟的优越性。

第一章 微处理器系统概观

1.1 数制和数字代码

微处理器系统是一种数字电路系统，它所使用的信息是以二进制数字式电信号的形式出现的。二进制信号只有二种最基本的信息：“真”和“假”。用电路的语言来说，也可以说是“接通”和“开断”。用电压状态来表示，也可以表达为“高电平”和“低电平”二种状态。用数字来表示，就是“1”和“0”。下面给出各种表示的等效对照：

逻辑真	通	高	1
逻辑假	断	低	0

当然也不妨反过来，以低电平代表“真”或“1”，而以高电平代表“假”或“0”；在此情况下，我们就称为采取“负真逻辑”表示，以别于上面表格所示的“正真逻辑”表示。

这样的一个小信息单位（一个“1”或一个“0”），称为一个“比特”（bit），或称为二进制的一位，也简称为“位”。若干个比特组合起来，可以表达复杂的信息。例如，若把有序排列的若干个1或0的组合看成是二进制（binary）来看待，则可以表达任何的数。在二进制中是“逢二进一”，于是我们就可以得到下列等效：

	二进制	十进制
一位数	0	0
	1	1
二位数	1 0	2
	1 1	3
三位数	1 0 0	4
	1 0 1	5
	⋮	⋮

于是，用若干个比特就可以表示任何一个数，为了不致混淆起见，可以在某个数加上一个下标来表明它所属的数制。例如

$$101_2 = 5_{10}$$

二进制中只有二个基本数，不是1就是0，别无其它。若把它看成是通或断，高或低电平，那么这种数对于机器或电路来说，是很容易辨认或识别的。

二进制变换为十进等效，方法很简单。我们知道，十进制是“逢十进一”，例如

$$1982_{10} = 1 \times 10^3 + 9 \times 10^2 + 8 \times 10^1 + 2 \times 10^0$$

同理，我们有

$$10101_2 = (1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0)_2$$

$$= (16 + 0 + 4 + 0 + 1)_{10} = 21_{10}$$

同理，一组有序比特也可以等效为其他数制的数，例如“逢八进一”的八进制(octave)，或“逢十六进一”的十六进制(Hexadecimal)。在十六进制数中，除了0~3外，用A、B、C、D、E、F分别表示 $10_{10} \sim 15_{10}$ 。我们看到：

$$1111_2 = 7_{10} = 7\text{Oct} \quad 1111_2 = 15_{10} = \text{FHex}$$

$$1000_2 = 8_{10} = 10\text{Oct} \quad 10000_2 = 16_{10} = 10\text{Hex}$$

如此类推，就不难看出，把二进制数中自右至左每三个比特集合起来，分别写出其十进等效数，即得到八进制数的表示。同样，把二进制数自右至左每四个比特集合起来，分别写出其十进等效数，并以A~F代替 $10 \sim 15$ ，就得到十六进制的表示。显然，一个数写成八进或十六进的形式，比其二进制表示短得多。

再如，若要表示图1.1.1a中所示那样的一个“工”字，我们可以用0代表图1.1.1a中的白格，用1表示黑格，于是就得到如图1.1.1b的结果。若把这些0和1的有序排列看作为二进制数，那么也可以得到其他数制的等效表示，如图1.1.1b的右边所示。这样一来，

我们就能用数来代表一个文字或者一个图形。或者说，对任何一个文字或图形进行数字编码，就象汉文电报代码那样。按照我们这里的特殊编码方法，图1.1.1a的图形，或汉文的“工”字，它的编码就是01110, 00100, 11111或16, 08, 37, 或OE, 08, 1E, 或15, 8, 31。

依此类推，任何信息都可以进行某种特定形式的数字编码，用数字代码来表示原来的信息。因此，任何信息也都可以看成是“数据”。把它作为数来看待，也就可以对它进行算术运算：加、减、乘、除等等。作为二进制数，则还可以逐位进行逻辑运算，运算规则如下：

$$\begin{array}{lll} 0 \wedge 1 = 0 & 0 \vee 1 = 1 & 0 \nabla 1 = 1 \\ 1 \wedge 0 = 0 & 1 \vee 0 = 1 & 1 \nabla 0 = 1 \\ 0 \wedge 0 = 0 & 0 \vee 0 = 0 & 0 \nabla 0 = 0 \\ 1 \wedge 1 = 1 & 1 \vee 1 = 1 & 1 \nabla 1 = 0 \end{array}$$

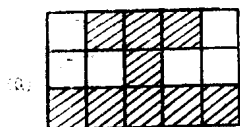
$$\overline{0} = 1 \quad \overline{1} = 0$$

其中 $\wedge$ 表示“与”(AND)，或“布尔积”，有时也写作为 $\cdot$ ，

$\vee$ 表示“或”(OR)，或“布尔和”，有时也写作为 $+$ ，

∇ 表示“异”(Exclusive OR)或“布尔差分”，有时也写作为 $\oplus$ ，

$\overline{()}$ 表示“非”(NOT)，或“布尔补”，有时也写作为 $()'$



	Oct	Hex	Dec
0 1 1 1 0	16	0E	14
0 0 1 0 0	08	08	8
1 1 1 1 1	37	1F	31

图1.1.1

“异”也称为“异或”。

利用逻辑和算术运算，就可能对任何信息进行复杂的、巧妙的处理。

有一种常用的国际通用的信息交换用的七比特码，称为ISO-646（这是国际标准化组织的一份标准文件的编号），我国称为SJ-939（中华人民共和国第四机械工业部标准），在美国称为ASCII（American Standard Code for Information Interchange），具体的编码见附录。新型的七比特或八比特电传打字机（TTY = Tele-Type）就采用这种编码，计算机用的键盘-显示器也用这种编码，不少新型电子仪器也用这种编码作输入输出。

此外，在电子仪器中还有一种常见的二进与十进混合的编码，称为BCD码（Binary Coded Decimae），即二进编码的十进数，也称为二-十进码。因为人类自古以来就用双手十指计数，一直习惯于十进数。但数字系统则易于使用二进数而不易使用十进数。为了两便起见，就创造了这种混合的BCD编码，它在二进与十进之间的变换过程中，过程较简。把十进数的每一位数都用其等效的4比特二进数表示，然后仍按十进数的数位排列，就得到BCD码。例如

十进数	1	9	8	2
BCD码	0001	1001	1000	0011

也就是说， $0001100110000011_2 = 4096 + 2048 + 256 + 128 + 2 + 1 = 6531$ ；但 $0001100110000011_{BCD} = 1 \times 10^3 + 9 \times 10^2 + 8 \times 10^1 + 3 = 1983$ 。

在数据的传输过程中，可能有二种方式，即比特串行或比特并行。按照一个数据中各个比特排列的先后顺序，依次先后改变一条信号传输线（当然还应该有一条接地回线）上的电平高低以表示每个比特的值，这就是比特串行传递，即是一个比特跟随另一个比特之后传出或传入。这种传递方法事实上是把各比特的空间序列变换为时间序列来发送，而接收时则反之。另一种传递方式是用N条信号线来传递N个比特，每一条线以其电平的高低分别表示每个比特的值，各个比特同时传递。显然并行比特传送比串行快速得多，但其代价则是信号线的数目增多。

如果信息包含成千上万比特，我们事实上不可能用成千上万条线来作并行传递，而机器也不可能同时对这么多比特同时进行处理。为此，有必要把信息分成含比特数不多的若干部分，例如每部分含16比特，然后再一部分一部分地传递和处理。这好比把一篇大块文章分开一个字一个字地写或读。因此这16比特就叫做一个“字”（word），一个字为16比特，这是小型计算机的习惯。一个“字”又可再划分为若干部分，每部分含更少的比特，以便更容易处理。“字”的这样的一个部分称为“字节”或“拜特”（byte），小型计算机习惯把一个16比特的字分为二个8比特的字节。于是在传送过程中，大多数采用比特并行、拜特串行的方式，即是用8条信号线，同时传送8个比特组成一个拜特，一个拜特跟随另一个拜特之后鱼贯而行。一个拜特中所含的比特数也常称为数据“宽度”。目前多数微处理器的数据宽度为8比特，或8位。

一个字或字节的二进数写出来时，位于最右的一个比特，其值对整个数值的影响最小，故称为LSB（Least Significant Bit），即最低位比特。位于最左的一个比特，其值对整个数的意义最大，故称为MSB（Most Significant Bit），即最高位比特。

1.2 微处理器系统的基本组成

一个基本的微处理器系统包括六个基本组成部分，如图1.2.1所示。

(1) 时钟CLK (Clock)

这通常是一个石英晶体振荡器，频率多在兆赫级，经过适当分频以后，形成矩形脉冲列，称为时钟脉冲，时钟脉冲用来控制微处理器系统内各部分工作的节拍，规定各部分动作的时间关系，即所谓“定时”(timing)，从而使各部分的工作同步，达到有节奏的合拍的协同动作，而不致错乱。

(2) 只读存储器ROM (Read Only Memory)

这是用来存储程序(指令)的。程序事先以某种特殊方式写入ROM内，微处理器系统在工作时，只从ROM内逐条读出指令，然后执行它；但微处理器系统却不能把任何信息写入ROM内，也就是说，ROM的存储内容在微处理器系统的工作过程中是不能改变的。这也就是“只读”这个名称的来由。当电源关断以后，ROM的存储内容依然存在于ROM中，不会消失。

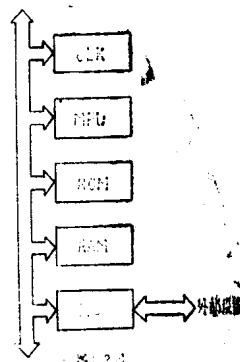
ROM内所存信息是在这种大规模集成块的制造过程中利用光刻掩模来写入的。一旦写入之后，就不能再修改或重新写入了。从向半导体工厂提供程序一直到生产出ROM来，一般需3~6个星期。这类ROM在大批量生产(例如1000块以上)时，成本较低，故适用于成熟的大批量生产。

另外还有一种PROM，即可编程的(Programmable)ROM，是供用户自己来写入程序的。PROM内原本每一个存储单元的内容都是0。用户根据自己的需要编制程序。利用适当的装置把所需要的1信息写入适当的存储单元中。一般是通入适当的脉冲电源，把指定的存储单元中的一条熔丝烧断，即能使原存内容由0变为1。因此，平常也说“烧”一个ROM。显然，这种PROM在写入(烧好)了之后，其存储内容也是不好更改的。你可以把原有的0改为1，但却无法把1改为0。这类PROM对于实验性的系统，或生产数量甚少的系统是适宜的。

还有一种可以把已写入的内容抹去而重新把新内容写入的PROM，称为EPROM，E即可抹(Erasable)之意。有时也称为RPROM，意即可重新编程(Re-Programmable)的ROM。它是由写入电脉冲在存储单元中感生蓄积电荷而形成1信息的。当以强烈紫外线射入EPROM的透明窗口时，照射5~10分钟就能使蓄积电荷中和掉而回复到全部0态，一般大约可反复抹写几十或几百次。EPROM价格较昂，不过它非常便于设计的试验阶段使用。

(3) 随机存取存储器RAM (Random Access Memory)

这类存储器也称为读/写存储器，因为在微处理器系统的工作过程中，可以随时把信息写入RAM内存贮起来，也可以从RAM中读出已存的信息。RAM中所存内容，可



以抹去或重写。把一个新的比特（不论为0或为1）写入RAM的某个存贮单元时，它就代替了原来所存的旧有比特。即是说，写入新内容时即自然把原有的旧内容“冲”掉。因此，重写时，不必先行抹去旧的内存再写。若要抹掉已存的一部分内容，则在相应的存贮单元内写入0即可。

当电源开断时，RAM的存贮内容即消失。因此不能用RAM来存贮永久性的程序，而只能用来存贮一些临时性的程序，或者存贮临时性的数据或中间计算结果等。需较长期保存的程序或数据，在编制好后，可先存入RAM中，以便试验，遇有问题便于修改；试验证明无误后，再由RAM读出而转录到磁带或磁盘上，这就形成了所谓“文件”（File）或“档案”。以后要用时，又可由磁带或磁盘再把文件装入（写入）RAM中，以供使用。

RAM有二种类型。一种称为静态（static）RAM，它是以触发器电路（双稳态电路）的形式来存贮信息的。因此，只要电源不断，只要不抹掉它（全部写入0），它所存的内容是不会消失的。另一种称为动态（dynamic）RAM，它是以电荷蓄存形式来存贮信息的。由于漏电作用，所存内容在几毫秒之内就会消失。所以必需每隔1~2毫秒就得刷新（refresh）一次。也就是把行将消失的存贮内容读出来，再重新原样写回去，不断地周而复始。这也就是“动态”名称的由来。为此，动态RAM必需外加为完成刷新操作的支持电路。刷新过程需由时钟保持严格的同步，定时要足够准确，否则就会扰乱系统的工作。由于刷新需要化费一定时间，这也会使系统的工作速度相应降低1~5%。然而，动态RAM有二大优点，其一是存贮密度比静态RAM大得多，也就是存贮容量比静态RAM大得多；其二是价格比静态RAM低廉。这二个优点合起来，就使其单位存贮容量的费用低得多。尽管要用支持电路，但对于需用存贮容量很大的场合中，总的来说仍比使用静态RAM更经济，而且也省电。

有时即使在断电的情况下也需要保存RAM中所存的内容。例如，当供电电源的保证成问题，可能会发生临时停电的情况。又如，在长期稳定度测量过程中，可能每次测量时间（采样时间）很短，而每二次测量（采样）之间的时间则很长（例如24小时），在此情况下，让测量仪器一直通电未必合理；然而每次采样数据必需保留，以供最后作统计分析处理。又如，把仪器周期性检定（例如每隔半年）所得数据存入，以供测量时作误差修正之用。在诸如此类的情况中，可以用干电池或蓄电池对RAM供电。并可用自动开关在交流断电时自动切换电源，不过应注意防止在切换过程中可能出现对RAM供电的瞬间掉电。

(4) 输入输出I/O (Input/Output Ports)

I/O口是微处理器系统同外界（外围设备）交换信息的通道。微处理器系统，视具体需要而定，可能需要有多个I/O口。在某一时间应从哪一个I/O口输入输出，或者容许或禁止哪一个或几个I/O打通，等等，都要有适当控制。

此外，I/O口事实上是微处理器系统与它的外围设备之间的通用“接口”（Interface），负责协调微处理器系统同外围设备在传输方面的差异，而使二者能互相相容。例如，并行或串行传输，传输速率，同步或异步传输，码的格式，等等许多方面，如双方不一致，传输就不能完成。为了使双方相容，接口就需进行一些控制或变换。通用I/O

接口一般分为并行接口和串行接口两类,为了尽可能普遍适用,这些通用I/O接口也都是可编程大规模集成电路。

当通用I/O口的接口能力仍不能适应某些特殊外围设备时,就还要再添配上专门的接口片子或电路。

(5) 母线 (Bus)

母线亦称为总线,这是微处理器系统内各组成部分之间的联线的总称。所谓母线结构,就是各组成部分的输出输入线全部都并联在一起,以便任一组成部分的信息都能传到另一部分去。微处理器系统内的母线又分为三组。一组称为数据母线 (Data Bus),它是数据 (广义的数据) 传递的通路。数据母线所包含的信号线数目,与系统的字节长度 (数据宽度) 相适应。多数微处理器系统采用8比特宽度,即是用8条数据线并行传递8比特的字节。数据母线是双向的,即八条数据线既可供输入之用,也可供输出之用,数据在线上可以向这一个方向传输,也可以向另一个方向传输。

另外一组称为地址母线 (Address Bus),是为寻址而设的。微处理器系统内各部分、存储器的每一个存储单元都编上一个号码 (地址),就好比是地址或电话号码。按照具体的地址就可以寻访一个特定的单元;或者,象打电话一样,拨出适当的号码,就能同某人通话。地址用并行比特作二进制编码。因此,地址线的多寡就决定了地址容量,即决定了可以编排的或可以访问的地址的总数。一般地址母线为十六条,地址容量为 $2^{16} = 65536$ 个,号称64K,实际上是从 $0000_{16} = 0_{10}$ 至 $FFFF_{16} = 65535_{10}$ 。地址母线是单向的,它由MPU输出到除时钟以外的各部分。

还有一组称为控制母线 (Control Bus),为了实现数据在各部分之间的正确传递,还必须对各部分的工作进行适当的控制。例如,MPU对RAM寻址时,还必须通知它是要读出 (数据从RAM的某一存储单元传递到MPU) 或写入 (数据从MPU传递到RAM的某一存储单元) 于是,就需用一条或二条控制线 (读/写线) 来规定是读出或写入。又如,当I/O口有数据急待输入时,也得设法通知MPU,请它暂时中断手头的工作,以便迅速从I/O口输入数据。这就需用一条“中断请求”IRQ (Interrupt Request) 控制线由I/O口通到MPU去。MPU收到中断请求时,是否就能立即中断手头的工作,也应回答I/O一个信息。这就又需用一条“中断认可”INTACK (Interrupt Acknowledge) 控制线由MPU通向I/O口。后者在收到中断认可信号后,才将数据传入MPU。诸如此类。这些作控制用的信号线的总体,就称为控制母线。时钟脉冲由时钟送到各部分去作定时,这也属于一种控制,所以传送时钟脉冲的线,也属于控制母线。到底总共要设立多少条控制线,各传递什么信息,各型微处理器系统不尽相同。各控制线上的信号传递方向也并不统一,这从上述例子可以看出。

1.3 MPU的内部组织

各种不同型号的MPU,其内部组织不尽相同。若求同存异,取其典型,则大体上可以一般化地表示如图1.3.1所示。

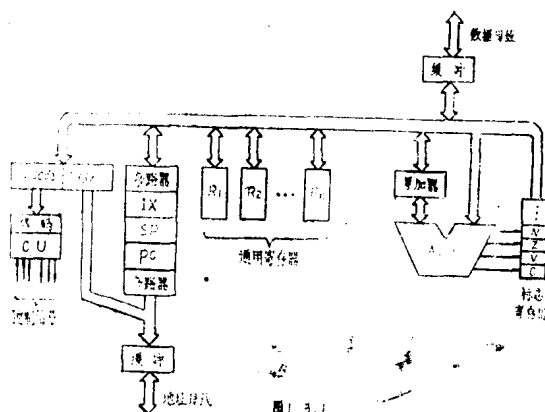


图1.3.1

数据由数据母线传出或传入MPU，首先经过一个缓冲器。缓冲器可以使数据母线同MPU联通，或者使母线同MPU断开或互相隔离（参见下面1.5）。

在一定指令作用之下，数据可以由数据母线经过缓冲器进入一个累加器ACC（Accumulator），然后再由累加器进入算术逻辑单元ALU（Arithmetic and Logic Unit），ALU也称为运算器。在适当的指令作用下，另一数据又可由数据母线经过缓冲器

进入ALU的另一输入端（图1.3.1上ALU的右上方），然后二个数在ALU内进行指定的运算。运算结果一般又由ALU输出到累加器去取代累加器中原来输入的数据，然后这个结果可以输出到MPU内部的数据母线上。视具体的指令而定，运算结果进一步经由缓冲器输出到系统的数据母线，或者输送到指定的MPU内部的寄存器中寄存起来。

与ALU配合工作的还有一个标志（Flags）寄存器，在里面寄存运算结果的某些特殊状况。例如，是否要进位可用一个比特C（Carry）的置1或清0来标志；是否得到负数可用另一个比特N（Negative）作标志；是否得到零值又用另一个比特Z（Zero）作标志。此外，还有数据溢出V（或O，即Overflow），中断I（Interrupt），半进位H（Half Carry）等标志比特。这些不同的标志寄存起来，可供今后的进一步处理作参考之用。半进位也称为辅助进位AC（Auxilliar Carry），即4个比特数（半字节）的进位，在处理BCD运算时作为标志。这些标志，在程序中需作非常灵活的转移（Branch）或跳跃（Jump）时，也是有用的。

MPU内部还有若干寄存器（Register），数目多少视MPU具体型号而异。这些寄存器分为二类。一类是通用寄存器或辅助寄存器，有单个使用的，也有成双使用的。这些寄存器主要用来暂时寄存运算过程中的一些中间数据，以便再取出供随后的运算步骤使用。这些寄存器与MPU内部数据母线作双向传递。

另一类内部寄存器是专用的寄存器，各型MPU内设置的一些专用寄存器基本上相同。其中一个是指令寄存器IR（Instruction Register），用来寄存通过数据母线从ROM或RAM中取得的指令。指令包括操作码OpCode（Operation Code）即指令代码也可能还有地址Addr（Address）。寄存的操作码经译码后进入控制单元CU（Control Unit），去产生和处理各种控制信号。地址则经译码后通过缓冲器输出到地址母线。

另外一个指针（Pointers）寄存器也是一种专用寄存器，它实际上由三个16比特寄存器组成，由多路选择器（开关器）分别可与MPU内部数据母线接通，或与地址母线接通。这三个指针寄存器中，一个称为程序计数器PC（Program Counter）。微处理器系统在执行程序时，一般是逐条指令依次执行的。由PC来计数，记住已执行到第几条指令，或者更准确地说，是记住从某一个存储器地址中取出了一条指令存入指令寄存