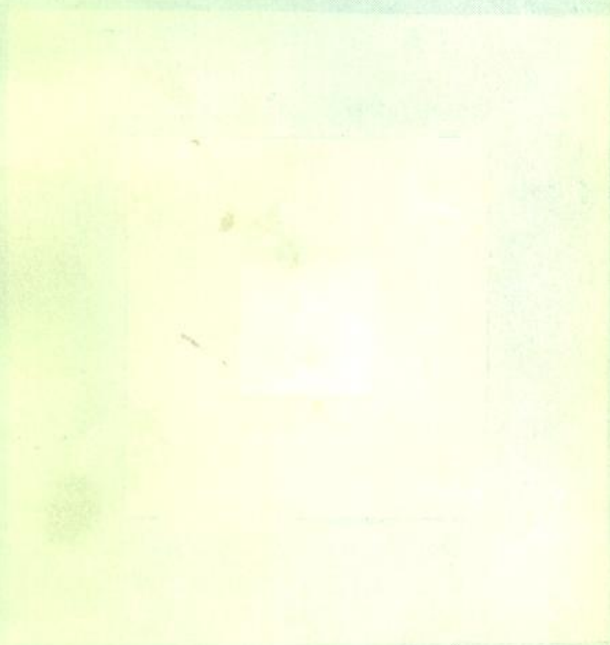


微处理机及 微程序设计手册



〔美〕B.沃德 著

科学出版社

73.876
234
3

微处理机及微程序设计手册

〔美〕B. 沃德 著

欧阳景正 周汉华 吴检保 译

欧阳景正 校



科学出版社

1982

1111333

JS131/27-3
内 容 简 介

这是一本介绍微处理机的科普读物。作者深入浅出地讲述了微处理机的工作原理、功能和用途。书中对目前国际上最为流行的微处理机系列做了具体介绍，告诉读者如何操作和编写微程序并举例说明之。

本书可供从事数据处理，机器控制和仪器仪表制造专业的人员以及广大科技工作者和大专院校的师生阅读和使用。

Brice Ward
MICROPROCESSOR/
MICROPROGRAMING HANDBOOK
Tab Books, 1975

微处理机及微程序设计手册

美 B. 沃德 著

欧阳景正 周汉华 吴检保 译

欧阳景正 校

责任编辑 隋启水

科学出版社出版

北京朝阳门内大街137号

中国科学院开封印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

1982年11月第 一 版 开本：787×1092 1/32

1982年11月第一次印刷 印张：10 1/2

印数：0001—14,500 字数：218,000

统一书号：15031·451

本社书号：2846·15—8

定价：1.30 元

前 言

微处理机的出现在数据处理和机器控制领域内开始了一场小型的革命,但它不是通过添加、而是去掉费用昂贵的一些技术指标来达到的。我们看到,对于大多数计算及控制应用来说,都不需要最现代化的、超高速的、超级价格的电子计算机。这样的一些应用倒是使用小型的、便宜的装置更为有利。这些装置应当适当地满足每种应用的特殊要求。(人们肯定不需要价值百万美元的计算机仅用来计算营业税!)甚至低档的计算机——所谓的小型计算机对于许多可能的应用场合也都显得太贵了。

现代计算机的麻烦在于它是通用型机,是专门设计来处理各种想象得到的计算事务的。为达到这种能力,现代计算机应当是高速的,有大容量的存储器来存储数据和指令以及能够很快地改变程序以解决新问题。然而,目前应用范围日益增加,这并不单单要求速度快、存储容量大或者能够改变程序,实际上大多数应用还是专用的而不是通用的,因此通常只要设计成不变的内部程序就够了。

计算机的所有基本处理部件能够装置在一个或几个很小的集成电路中,构成微处理机。为了进行实际算术和逻辑运算,微处理机包含一个中央处理部件。为使微处理机始终记得它所要完成的功能,将它的指令永远地存放在“只读存储器”中;为了记住不断变化的数据和输入/输出信息,微处理

机还需要有一个很小的“随机存取存储器”。这就组成了一个专用的装置，它能做你想要它做的事，这多省事啊！

作为设计者，你能够确定你的特别要求，并可能借助于通用机来发展一种单一的程序——微程序，以适用于你的微处理机系统，使其能完成预想的操作和运算。你的程序可以永久地记录到只读存储器中。为了增加灵活性，也可使用几个只读存储器，只要插入一片新的电路，就可以改变基本设计。可是设计的简化实际上还是由于现代微处理机系统发展引起的。

为了使您方便起见，本书附录中收集了大量的微处理机及部件制造厂地址。作者在此对于三个工厂表示感谢。他们是摩托罗拉公司（MC6800系列），英特尔公司（MCS系列、接口设备及存储器），全国半导体公司（微处理机片）。十分感谢他们为本书提供的大量资料。

B. 沃德

目 录

第一章 微处理机入门	(1)
1-1 是微处理机还是微计算机	(1)
1-2 现代计算机的演变	(3)
1-3 计算器与计算机的比较	(5)
1-4 基本指令	(6)
1-5 计算机的积木式结构	(7)
1-6 跟上微处理机的发展	(9)
第二章 计算机的基本操作	(11)
2-1 数字计算机	(11)
2-2 中央处理机的结构	(12)
2-3 程序指令	(13)
2-4 子程序	(14)
2-5 指令代码	(15)
2-6 运算器	(17)
2-7 标志位	(18)
2-8 控制电路	(18)
2-9 系统中断	(20)
第三章 微处理机的内部	(22)
3-1 存储器存储系统	(22)
3-2 基本机器周期	(25)
3-3 程序计数器	(26)
3-4 存储器地址寄存器	(27)
3-5 程序指令	(28)

3-6	寄存器操作	(30)
3-7	增加指令和存储器容量	(31)
第四章	近代微处理机系统	(32)
4-1	制造方法	(32)
4-2	双极型的例子	(33)
4-3	某些重要的定义	(35)
4-3-1	处理机	(35)
4-3-2	计算机	(36)
4-3-3	程序设计	(36)
4-3-4	芯片系列	(38)
4-4	中央处理机(CPU)	(39)
4-5	只读存储器(ROM)	(41)
4-6	随机存取存储器(RAM)	(45)
第五章	MCS-4 型微处理机系统	(49)
5-1	三态输出	(49)
5-2	MCS-4 CPU	(50)
5-2-1	机器定时	(50)
5-2-2	变址寄存器	(50)
5-2-3	程序计数器及栈寄存器	(52)
5-2-4	子程序嵌套	(54)
5-2-5	程序计数器增量器	(56)
5-2-6	寻址及译码指令	(56)
5-3	MCS-4 ROM	(58)
5-4	MCS-4 RAM	(59)
5-5	MCS-4 系统的定时	(60)
5-5-1	系统的时钟	(60)
5-5-2	定时图表	(61)
5-6	MCS-4 的操作	(63)

5-7 系统的连接	(65)
5-8 MCS-4 指令系统	(66)
5-8-1 无操作(NOP) 指令	(70)
5-8-2 条件转移(JCN)指令	(71)
5-8-3 立即直接取数(FIM)指令	(74)
5-8-4 发送寄存器控制(SRC)指令	(75)
5-8-5 间接取数(FIN)指令	(75)
5-8-6 间接转移(JIN), 无条件转移(JUN)和 转移到子程序(JMS)指令	(76)
5-8-7 增量(INC)和增“1”为零跳(ISZ)指令	(77)
5-8-8 加法(ADD)和减法(SUB)指令	(77)
5-8-9 装入(LD)和交换(XCH) 指令	(78)
5-8-10 转移返回及装入 (BBL) 指令	(78)
5-8-11 直接装入(LDM)指令	(78)
5-8-12 数据传送指令	(79)
5-8-13 数据操纵指令	(80)
第六章 MCS—40微处理机	(81)
6-1 4004中央处理机	(84)
6-1-1 功能方框图	(85)
6-1-2 地址寄存器和地址增量器	(87)
6-1-3 变址寄存器	(87)
6-1-4 4位加法器	(88)
6-1-5 指令寄存器、译码器和控制电路	(89)
6-1-6 外围电路	(89)
6-2 4040中央处理机	(90)
6-2-1 一些新的概念	(91)
6-2-2 基本的4040CPU 定时	(98)
6-2-3 独特的操作特点	(99)

第七章 MCS-80 微处理机	(106)
7-1 堆栈结构	(110)
7-2 8080的基本指令周期	(112)
第八章 MC6800微处理机	(118)
8-1 结构	(118)
8-2 MPU操作	(119)
8-2-1 数据和地址操作	(119)
8-2-2 中断和复位操作	(121)
8-2-3 寄存器操作	(123)
8-3 MC6800的指令系统	(125)
8-4 MC6800系列的其它部分	(133)
第九章 存储器系统	(136)
9-1 随机存取存储器(RAM)	(137)
9-1-1 半导体RAM	(138)
9-1-2 静态RAM	(138)
9-1-3 场效应晶体管	(140)
9-1-4 电荷存储器件	(141)
9-1-5 动态RAM	(143)
9-2 只读存储器(ROM)	(146)
9-3 4001掩模可编程序ROM	(147)
9-3-1 操作方式	(149)
9-3-2 ROM方框图的说明	(150)
9-4 4002RAM和输出端口	(151)
9-4-1 RAM 操作	(151)
9-4-2 I/O操作	(155)
9-5 标准的存储器和I/O接口	(157)
9-5-1 电路操作	(157)
9-5-2 WPM指令	(160)

第十章 微程序设计入门	(162)
10-1 软件和硬件	(163)
10-2 8080指令系统	(163)
10-3 一些程序例子	(183)
10-3-1 十进加法程序	(183)
10-3-2 十进减法程序	(186)
10-3-3 二进乘法循环	(188)
10-4 小结	(191)
第十一章 用机器语言的微程序设计	(193)
11-1 机器语言指令	(194)
11-1-1 指令系统的格式	(196)
11-1-2 变址寄存器结构	(199)
11-1-3 堆栈寄存器的结构	(201)
11-1-4 命令总线操作	(202)
11-2 4040指令系统	(204)
11-3 计算机仿真	(223)
11-3-1 用纸仿真	(223)
11-3-2 ROM 仿真器	(224)
11-3-3 RAM 仿真器	(225)
11-3-4 CPU 仿真器	(226)
11-4 程序例子 1	(230)
11-5 程序例子 2	(240)
11-5-1 可能的寻址错误	(241)
11-5-2 程序继续下去	(242)
11-5-3 小结	(254)
第十二章 汇编语言的程序设计	(256)
12-1 汇编程序	(257)
12-1-1 标号字段	(257)

12-1-2	操作码字段	(259)
12-1-3	操作数字段	(259)
12-1-4	注解字段	(259)
12-2	程序例子 3	(260)
12-2-1	把问题反复想一想	(260)
12-2-2	产生时钟脉冲	(261)
12-2-3	循环的构成	(270)
12-2-4	程序的解释	(271)
12-2-5	计算机仿真	(272)
12-3	程序例 4	(310)
12-4	程序例 5	(311)
12-5	程序例 6	(314)
12-6	程序例 7	(315)
12-7	程序例 8	(318)
12-8	程序例 9	(319)
12-8-1	中断操作	(319)
12-8-2	写出一个中断程序	(320)

第一章 微处理机入门

很容易想象得到，微处理机是由少量的集成电路做成的，微处理机实际所指与其说是任何特定的器件，倒不如说是一个概念。微处理机是很小的处理机，它是为了处理信息而专门设计的，也就是为了完成特定的任务而制造的。微处理机蕴藏着巨大的能力，因为它给工、商业的各个领域提供了极其强有力的和廉价的设计手段。

1-1 是微处理机还是微计算机

微处理机往往被认为是微计算机，因为它包括通常计算机所具有的全部算术和逻辑功能。但是，微处理机本身并不就是计算机，尽管它也能用来组成小型计算机，或在大型计算机以及它们的外围设备中应用。仅仅由于微处理机具有各式各样的用途这一事实，可以充分说明微处理机的能力足以解决许多原先要由硬件和分立逻辑电路承担的任务。

的确，微处理机可能完成你想要它做的任何事情。要使这成为可能，微处理机元件制造厂设计了许多微处理机的电路和外部设备，并且都是十分便宜和非常灵活的。这就使得微处理机系统具有特殊的本领来处理各种事务和进行各种控制操作，而不需要很大的代价。

近代的大型计算机当然也能对付同样多的处理功能。然

而，近代计算机体积很大价格很高。由于近代大型计算机专门设计来解决任何可能想象得到的处理事务，它要有灵活性，要具备既迅速而又容易的方法，来改变控制操作的程序。这样，近代计算机多是通用型的计算机，它所付出的主要代价是为了追求通用性。

实际上大量的应用都不是通用性的，而是颇为单一性的。例如，在近代现金收入记录机终端里的微处理机只是用来完成现金收入记录机的一般需要完成的功能，如：计算销售总量，计算营业税，计算零钱以及还可能进行货存盘点等等。这里就不需要它提供象通用计算机那么昂贵的功能。事实上，在这种专一用途中所必须考虑的问题是如何以最小的代价去完成全部必需的功能。在这些类型的应用里，微处理机能做得很出色。

微处理机与大型计算机的区别主要在于制造微处理机时就把控制它运行的程序“冻结”下来，它不象可编程序计算机中的程序那样可以变动。然而，微处理机也不象分立元件的设计那样可以利用硬线连接编制程序，那种方法不仅制造成本很高，而且几乎不能改变。与上面那些方法都不一样，微处理机的程序通常是永久性地存储在只读存储器(R O M)中的，这是一种既简单又便宜的存储程序的方法。改变程序的办法也很简便，只要插进一片不同的只读存储器就行了。如此一来，在设计阶段，微处理机就保持了计算机式的灵活性和编制程序的简便性，这就使设计师有可能化较少功夫，获得复杂的处理功能。并且在制造阶段，微处理机也保留着低成本和操作的熟练性，这就使它能够满足日益增长的用途的需要。由于微处理机最能满足多方面的用途，因而它被誉

为当代的一个最富有革命性的概念是一点也不奇怪的。

现代的微处理机是两种工艺发展的产物，即大规模集成电路和低成本半导体存储器。大规模集成电路(LSI)使其能够用来制造复杂的微处理机，这些微处理机可以用程序的方法解决复杂的处理问题。另一方面，由只读存储器(ROM)和随机存取存储器(RAM)组成的半导体存储器可以给微处理机的程序指令和数据提供一种廉价的和紧凑的存储手段。其结果是现在已经可能利用少量的集成电路设计出完整的微处理机系统。但是，正确的微处理机功能依旧需要由设计工程师或技师来做。他们应当知道微处理机的内部操作和程序编制技术，以便实现所需的功能。本书的目的就是提供足够的材料，使读者了解微处理机的操作并能给它编程序。

1-2 现代计算机的演变

这里不想尝试追述和提出各种计算机，而只想指出这些机器在近三十年至四十年发展概况。它们是从百贝芝时代开始发展起来的，在当时，需要用到快速算术计算的机器，特别是在天文和其它科学领域中为了达到期望的结果，需要冗长的计算时间。

在第二次世界大战期间，用了机械凸轮、齿轮和差分器等发展起来的计算机以计算炮火控制问题和处理一定类型的导航系统。这些装置当时并不是世界上最精密的装置，但是，它们能节省时间。

二次大战以后，计算机在别的领域中的需要也增长了。国际商业机器公司(IBM)在商业界中起了带头作用。要求

商业计算机能分类、按字母排列和保存地址名单以及诸如此类的有关商业界的信息。数学计算是需要的，但这仅是整个用途中的一小部分。

在某一时期，计算机曾粗略地被分成“科研用”和“商业用”类型。早期的IBM设备采用IBM型何勒内斯码穿孔卡。这种卡片在IBM公司发展的分类和整理机中用起来十分满意。它们最终成为一种主要系统，用于存放各类原始数据和程序。

早期的分类机采用接触电刷来“读出”穿孔卡上的孔，它们利用“插孔”板来编制程序，插在孔中的跨接线形成分类程序。许多早期计算机采用类似的插孔板在卡片上编制程序和存储数据。这种用于程序和数据分开存储的技术被称为哈佛结构。

所有这些机器的发展都是为了做一件事——处理数据。这种处理可能是加法、减法、乘法或除法，但是它仍旧属于数据处理范畴。数据处理仍旧是对任何数据处理系统的主要要求。

这些机器越来越精细。它们从机械系统过渡到继电器，从继电器过渡到电子管，从电子管过渡到晶体管，又从晶体管发展到集成电路。然而其基本要求仍旧不变，即处理数据。

出现了新的情况。早期的机器试图保持十进制运算，但是人们很快就认识到，由于价格的原因，全部运算以采用二进制算法为好。用电的开/关状态来表示二进制，比十进制要容易得多。因此，计算机以及使用它的工程师和技术员，趋向于使用二进制或自然二进制代码，例如八进制或十六进制

等。

人们也开始认识到，为了得到适当的速度，就必须在计算机内部建立解决特定问题的程序。不仅加法，而且决定做加法的顺序和方式的程序也用电或电子的方法来实现，以便最大限度地利用设备所固有的速度。

1-3 计算器与计算机的比较

假设需要把 $341 + 24 + 419$ 加起来。在一个计算器上首先要输入第一项数字。输入动作使数字存入累加寄存器中并用读出装置显示出来。加法按键 $[+]$ 按下后表示要进行加法运算。然后输入第二项数。再次按下 $[+]$ 键，这就使得前面的加法指令执行完毕。相加的结果又存入累加寄存器中并立即显示出来。最后一项数字输入之后，接着按一下 $[=]$ 键，它完成最后的一次相加。其结果存入累加器中并显示出来。整个操作可能需要25到30秒钟。这对于那些具有计算机速度的设备来说，实在是太浪费时间了。

加法时间即使对于一个慢速计算机来说，也只要50微秒左右就够了。这就是说，对于上述的加法运算大约需要150微秒。经过简单的算术计算可以看出，这种慢速计算机在与计算器花费相同的时间内可以进行200,000次加法运算。

然而，为了在这么高的速度下工作，需要处理的数据和控制处理的程序均需存在存储系统中，以使用电子的速度把它们取出来。因此，此系统应当为数据传送设置一种特别的方案。数据应当由某个存储单元转移到累加器中，然后又从累加器转移到同一或另一个新的存储单元中。

1-4 基本指令

我们曾制定数据操作指令和数据传送指令。前者包括简单指令，诸如加法、减法、清除、增量等等。后者包括取出、存储、输入、输出以及类似的指令。

另外还有一些非常重要的指令组，它们可以大致地定义为程序控制指令。其中第一条，而且最富有逻辑性的一条是停机指令，假如计算机执行完一个程序，则操作员应能把计算机停下来，但是，如果编制程序的人给计算机增加一条“停止”指令，那就会容易得多。而大多数计算机都有这样一条指令的。

按规矩这里还应有一个无条件转移指令和一个条件转移指令。如果没有停止或转移指令出现的话，一个给定的程序总是在程序计数器（PC）的控制下按数字序列来执行的。无条件转移就是把新的数字“塞进”程序计数器去，使得下一条指令从转移指令所指出的单元中取出。条件转移指令包括检验，诸如：如果累加器为零时就转移；如果累加器是正值时就转移；如果累加器为负值时就转移或其它类似的条件。如果被检验的条件得到满足，则程序计数器就改变或修改；如果不满足，则程序计数器继续按顺序计数下去。

如前所述，这些指令全部能够粗略地分组并冠以程序控制指令的标题。这些指令就是能存储程序的通用数字计算机与简单计算器之间的真正差别。