

微 处 理 机 入 门

[美]查尔斯 杰莫著

上海卢湾区业余大学《微处理机入门》翻译组译

上海科学技术文献出版社

微 处 理 机 入 门

〔美〕查尔斯·杰莫著

上海卢湾区业余大学《微处理机入门》翻译组译

*

上海科学技术文献出版社出版

(上海高安路六弄一号)

新华书店上海发行所发行

上海市印刷十二厂印刷

*

开本: 787×1092 1/32 印张: 6 字数: 139,000

1980年8月第1版 1981年5月第2次印刷

印数: 13,301—18,550

书号: 15192·95 定价: 0.77 元

《科技新书目》167-220

目 录

第 一 章	引言	1
第 二 章	微处理机的术语	11
第 三 章	二进制算术	26
第 四 章	基本逻辑	38
第 五 章	微处理机用作中央处理单元	68
第 六 章	存贮器	90
第 七 章	输入输出	102
第 八 章	输入输出设备	113
第 九 章	微处理机指令系统	126
第 十 章	程序设计概念	142
第十一章	两种微处理机——6800 与 8080	154
第十二章	微处理机的实践	170
附 录 1	2 的幂	176
附 录 2	ASCII 字符表	177

第一章 引 言

“什么是微处理机?”微处理机是做在一块或两块小小的集成电路片上的数字计算机。“为什么半导体工业界要花这么大的力气来做到这一步呢?”“为什么说集成电路数字计算机的出现,促进了电子界的‘革命’呢?”要解答这些问题,我们有必要回顾一下导致微处理机发展的道路。

微处理机的发展

数字计算机技术是五十年代开始发展的。现在我们已学会了如何使数字计算机成为大型复杂的,具有人们以前难以想象的那种庞大的处理数据能力的机器。我们已经学会了如何制造和使用比较小的、价格比较低的计算机,即小型计算机。这类小型计算机现在在许多方面得到了应用。而在不多几年以前,这些应用方面还不能采用数字计算机,因为那时只有大型计算机,而大型计算机价格太贵,负担不起。

读者不禁要问:“数字计算机能干什么呢?”数字计算机能做的事情真是举不胜举。计算机所做的事情是由人们为它编制的程序所决定的。如果我们愿意,我们可以编制程序,使计算机成为一只数字钟、做电视游戏或成为复杂的计算器。事实上,计算机今后还能够做许多我们现在甚至还没有想到的事情。然而,即使使用最便宜的小型计算机来做这些事情,代价也还是太高了。因此,许多场合还不能使用计算机。

五十年代以来，另一门技术也在不断发展着。这就是飞跃发展的半导体技术。五十年代发展了二极管和三极管。到六十年代，集成电路的集成度从一块硅片上只能集成几只晶体管发展到能够集成数以千计的晶体管，硅片的面积没有增加，但却能够完成许许多多极为复杂的任务。

今天我们已能买到各种各样的专用集成电路。许多数字钟表都是由复杂的专用集成电路制成的。计算器工业转眼之间从最简单的四则运算器发展成为非常复杂、可编程序的科研和商业用计算器。每一种新计算器的出现，都是集成电路技术向前发展一步的结果。最新投入电子市场的一种产品是电视游戏。最初出现的是只能做一种游戏的黑白电视，而现在由于集成电路技术的不断发展，已经有了彩色的电视游戏，只要轻轻按一下开关就可在两个、三个甚至更多种游戏中任选一种。

在这种形势下，人们不约而同地出现了这样一种想法：只要我们能制造出一种非常便宜的数字计算机，我们就毋需为不同的用途设计不同的专用集成电路了。我们只要根据新的用途，重新编一下计算机的程序，一项新产品就马上可以出来了。

现在，发展微处理机的道路可清楚了。如果微处理机真是一种集成电路数字计算机，那末，如果有什么新的用途，我们只要为我们的微处理机编个程序就行了。

正是这样的思路引导我们去发展微处理机：第一，我们知道，几乎任何要做的事，都可以编成程序让数字计算机去做。第二，这种数字计算机的造价要和过去我们为了特定的用途而制造的专用集成电路的价格一样低。第三，我们必须用数字集成电路技术来制造这种集成电路数字计算机。

到七十年代初期，我们在数字计算机和集成电路方面的成就已经可以做到这一点。因此，就诞生了第一台微处理机。但

它还不能满足全部要求。在许多要采用微处理机的场合还嫌它过于昂贵。特别是用在消费品方面更是如此。但是，尽管在当时微处理机还不能贮存足够的信息去做小型计算机所能做的许多工作，它毕竟还是出现了。在过去的几年里，微处理机取得了进展。每片微处理机集成电路的价格从几百美元跌到5美元不到。微处理机作为数字计算机，它的功能已达到如此复杂的程度，以致于它和小型计算机几乎没有什么区别。

现在我们已经看到，以微处理机形式出现的数字计算机在许多应用上已取代了专用集成电路。微处理机已被用于计算器，用于电视游戏，用来控制汽车废气的排放，甚至用来管理自动唱机和演奏电子琴。

导致微处理机发展的途径，也是了解微处理机的最好途径。如果我们掌握了数字计算机的基本工作原理，那末理解微处理机的原理也就不成问题了。好在数字计算机的原理是简单的。计算机如果从整机来看，那它是一瞬间能完成千万步操作的、我们无论如何也跟不上的机器。这使我们难以认识它。但是，如果我们选择一个简单的操作例子，一步步的把它放慢下来，我们就可以理解它。说实在的，把一台计算机分解成它的基本部件和基本动作，它确实是非常简单的。

微处理机有多大？

从外形上看，所有微处理机的体积是一模一样的。它们都是制在一块大小约为0.1到0.2英寸见方的硅集成电路片上的。这些集成电路，采用塑料或陶瓷封装。其典型的封装形式如图1-1所示。

衡量微处理机大小的真正的尺度是它能处理的数据字的长度和它的存储器所能贮存的字数。为了便于处理和贮存，同一微处理机内的字长都是一样的。微处理机的字长就是一个字



图 1-1 通常用于微处理机和有关电路的典型双列直插式封装。

(A) 16 插脚式 (B) 24 插脚式 (C) 40 插脚式

的位数。一位就是一个二进制的数。微处理机的字长一般是 4、8、12 或 16 位。因为用得最多的是 8 位字，因此 8 位字有一个专门的名称叫做字节。一个字节就是一个 8 位字。

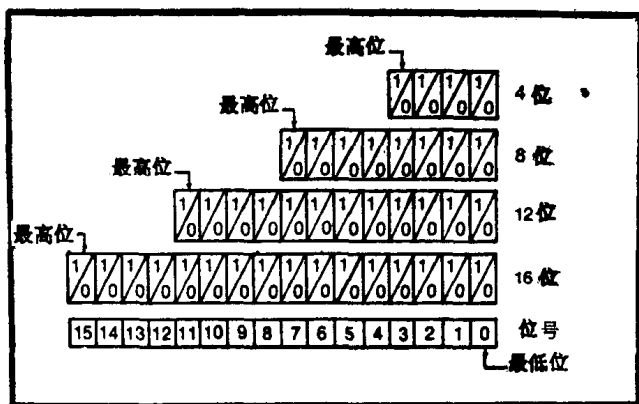


图 1-2 4、8、12 和 16 位字。

注意：字的每一位可以是 0 或 1。字中的位数从被称作 0 位的最低位 (LSB) 数起，数到最高位 (MSB)。最高位可以是第 3 位，第 7 位，第 11 位或第 15 位，视字长而定。

数字计算机是二进制性质的。在二进制数字电路中只存在两种可能状态——导通或截止。因此一个字里的每一位也只可能有两种状态存在——导通或截止。通常把导通看作“1”，把截止看作0。图1-2中是几种位数不同的字。这些字的区别在于它的字长。每一个字都有它的最低位(LSB)和最高位(MSB)。

当我们提到数字计算机的“字”时，不妨举简单的计算器为例。一个简单的6位计算器，总是做6位数的加、减、乘、除。虽然前置零是无效的，但是，6位计算器仍按6位数运算。我们不能从显示器上消除或增加位数。微处理机也一样。如果它的字是由8位组成的，它就必须每次处理8位，不多也不少。数字计算机的8位同计算器的区别仅在于所用的数不同。计算机中的每一位数不是0就是1，而计算器中每一位的数则可以是从0

可能组合数	第3位	第2位	第1位	第0位
1	0	0	0	0
2	0	0	0	1
3	0	0	1	0
4	0	0	1	1
5	0	1	0	0
6	0	1	0	1
7	0	1	1	0
8	0	1	1	1
9	1	0	0	0
10	1	0	0	1
11	1	0	1	0
12	1	0	1	1
13	1	1	0	0
14	1	1	0	1
15	1	1	1	0
16	1	1	1	1

图1-3 1个4位字的16种组合形式。注意：各种组合都是彼此不同的。
组合总数= $2^4=16$ 。8位字有256种组合，16位字有4,096种组合。

到 9 的任何数。

如果每位不是 1 就是 0, 那末, 一个一位数就只可能存在两种可能的组合——1 或 0。如果有 2 位数, 那就可能存在四种组合——00, 10, 01 和 11。对一给定字长的二进制字, 它可能出现的组合总数等于 2 的 N 次方。其中 N 是字的位数。图 1-3 是一个 4 位字的 16 种可能的组合。这是根据公式 $2^4=16$ 得出的, 即 $2 \times 2 \times 2 \times 2 = 16$ 。2 的低次方的值, 容易记住, 要记住高次方就较为困难了。为方便起见, 附录表 A 给出了在微处理机运算中经常使用的 2 的 N 次方的值。

衡量微处理机大小的一个指标是它的字长, 字长也必然会反映到微处理机存贮器的大小上。存贮器必须能存贮一定数量这种字长的字。例如, 一个 8 位的微处理机, 如果它的存贮器可贮存 10 个字, 那它就具有 80 位的存贮量。通常我们并不关心存贮器所能存贮的位数, 而是关心它能存贮多少字。我们规定一个 4 位微处理机就设有存贮 4 位字的存贮器, 一个 16 位微处理机就设有存贮 16 位字的存贮器。

所以, 存贮器的大小就用它可以存贮多少个字来衡量。而被我们称为微处理机的集成电路片中常常没有或只有很小的存贮器, 这些微处理机的存贮器是制在另外的集成电路片上的。

由于一块集成电路片上可用来连接外电路的接线脚数是有限的, 因此必须将微处理机集成电路同存贮器集成电路间的连线限制到最小的数目。为了做到这一点, 微处理机对各个存贮单元的选址采用二进制字的形式。因为采用了二进制字, 接线脚的利用率就高得多。例如, 如果存贮器只存贮两个字, 那末两只插脚就可对所需的存贮字进行选址; 如果要取用字 A, 就使用对应于字 A 的插脚。如果要取用字 B, 就使用对应于字 B 的插脚。然而, 如果在 A 和 B 两个插脚上采用二进位制的四种组合

形式即 00, 10, 01 和 11, 那末, 两个插脚就可用来选择 4 个存贮字。从上例可看出, 如果使用 4 个插脚, 则可选取的字就多达 16 个。

因为在选址中采用了二进制字, 所以微处理机的存贮字数往往等于 2 的乘方。例如 1,024 字 (2^{10})。由于 1000 是最接近 1,024 的数, 我们简称 1K (K 代表 1000)。因此, 1K 存贮器的存贮量实际上为 1,024 字。对于非常小的微处理机系统来说, (一个微处理机系统包括一个微处理机, 相应的存贮器、集成电路及其它所需的线路) 也许只采用 128, 256, 或 512 字的存贮器。大一些的微处理机系统采用 1K, 2K, 4K, 8K, 16K, 32K 或 65K 字的存贮器。为了方便起见, 把实际字数中不到 1000 字的另数去掉, 简化成最接近的千字数。

几乎所有的微处理机集成电路都具有很大的选址能力。然而, 正如上面提到的, 小型微处理系统并不用足它的这种能力, 而大型系统则可能用其一半或全部能力。

计算机的基本运算

可以用一个简单的、叫做“威格莱大叔”的儿童游戏来帮助理解数字计算机的工作过程。在这游戏中, 参加游戏的人先抽一张上面写有走子说明的白牌, 当然要有人记住游戏该轮到谁和下一次抽什么牌。阅读走子说明并根据说明走子。走子以后, 依次从牌中抽取下一张牌。游戏当然不会这样简单。其中某些牌会改变正常的抽牌顺序。例如, 一条要求从红牌中抽牌的说明就把通常的从白牌中顺序抽牌改变为暂时的从红牌中顺序抽牌。在走了一定步数以后, 还可能会出现其它一些复杂情况。例如, 参加游戏的某一方可能要轮空一次等。

现在让我们再重复一次该游戏的过程。但是这次请把上面写着走子说明的牌看作是存贮在计算机存贮器内的程序。把游

戏者看作翻译走子说明的称为中央处理单元 (CPU) 的数字电路。把“子”看作待处理和进出存贮器的数据。在正常的游戏规则后面, 我们加一个括号, 括号里给出同这些步骤等效的计算机术语。

在这游戏中, 参加游戏的人 (CPU) 先抽 (取出) 一张上面写有走子说明的白牌 (从存贮器中取出的字), 当然要有人记住游戏该轮到谁和下一次抽什么牌 (程序计数器计数)。阅读走子说明 (由 CPU 译出), 并根据说明 (处理) 走子 (数据)。走子以后 (指令执行后), 依次从牌 (存贮器) 中抽取 (取出) 下一张牌 (字)。游戏当然不会这样简单。其中某些牌 (指令字) 改变正常的抽牌顺序。例如, 一条要求从红牌中抽出一张牌的说明 (转子程序) 就把通常的从白牌中顺序抽牌改变为暂时的从红牌中顺序抽牌。(子程序完成后才能返回)。在走了一定步数以后, (当数据处理到某一预定点时) 可能会出现其它一些复杂情况。例如, 参加游戏的某一方可能要轮空一次等 (跳步)。

有了上面的简单概念, 我们就可更详细地考察微处理机的

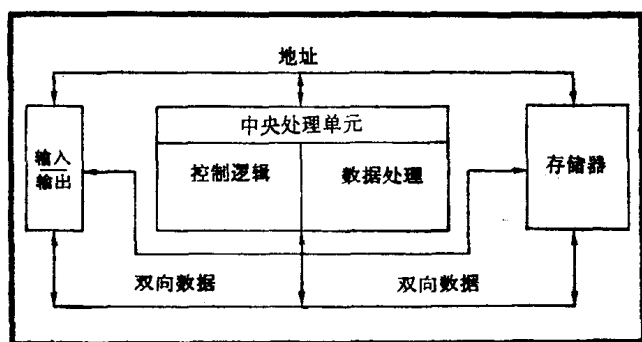


图 1-4 微处理机或微型计算机的简明方框图。有的微处理机既有中央处理单元也有存贮器, 有的只含有中央处理单元。少数微处理机可能还含有 I/O 驱动器。

内部结构。

简单的微处理机系统由三部分组成：中央处理单元(CPU)、存贮器和输入/输出装置(I/O)。微处理机中一定有中央处理单元。有些微处理机除了中央处理单元外还有存贮器，少数微处理机还有某种形式的输入输出能力，如图 1-4 微处理机的简明方框图所示。注意图中的数据(代表数的字)可以进出中央处理单元、存贮器或 I/O。中央处理单元具有把地址信息送入存贮器或 I/O 的能力。这种地址告诉存贮器哪个字要被取出作为指令，或者数据将存贮在哪一个位置或从哪一个位置取出。

实际上，计算机所做的就是完成游戏中那种简单的“取指”/“执行”循环而已。

正如存贮器必须有了地址之后，才能使数据输入或输出一样，对于 I/O 设备也必须要有地址。系统中通常具有一个以上的 I/O 设备。因此，中央处理单元必须决定把数据传到那一个 I/O 设备中去。中央处理单元也是用选址方式通知它所联系的那个 I/O 设备的。

至此，我们已讨论了从存贮器取出指令字和数据进出存贮器的问题。上面的讨论意味着在数据字和指令字之间是没有什么差别的，两者均可存贮在存贮器中的任何地方。事实上这就再一次证实了数字计算机的通用性。为了执行某一程序，程序员可以留出一块内存区域来存贮程序，而留出另一块区域用作数据存贮。如果程序员愿意，这种安排在执行另一程序时可以改变。唯一的要求是构成计算机程序的指令字必须按它被使用时的次序存放。

现在让我们更详细地看一下构成微处理机系统的各个主要部分。让我们先看一下中央处理单元，因为它是计算机的心脏部分。后面几章将讨论存贮器和 I/O 设备，掌握系统中这几个

部分对于了解微处理机是至关重要的。

正如前面提到的,所有微处理机并不都是一样的。多种微处理机都有一个中央处理单元。许多微处理机还有存储器,而某些微处理机还可能有 I/O 接口。显而易见,掌握中央处理单元的构成及其工作原理,是掌握微处理机的关键所在。中央处理单元和微处理机这两个术语有时候是同义词。当然,这样做似乎随便了些,但也并不是完全不确切的。因为所有的微处理机确实都有一个中央处理单元,而且在许多场合,微处理机就是微处理机系统或者微型计算机的中央处理单元。用另一种讲法来表达,那就是一个微处理机可以用作叫做微型计算机的中央处理单元。微型计算机同小型计算机通常是以它的中央处理单元是采用微处理机还是采用许多数字集成电路组成来区分的。

本书的使用

读者如何使用这本书取决于读者对数字电路和微处理机的了解程度。有一点是肯定的,即使你不去学电子学或电学,从这本书里也可学到有关许多微处理机的知识。当然如果你打算比较深入的学习或做实验,你就必须学些电子学的基本知识。然而和高传真设备或业余无线电收发报相比,这里所需要学的东西就少得多了。

下一章是术语汇编。再后面几章将分别详述微处理机的各个组成部分。如果你想读得快一些,可以选择阅读你最感兴趣的章节。遇到生疏的术语,可查阅术语汇编,如果嫌术语的说明不够详尽,可根据每一定义的参考书目,查阅有关章节中的详细内容。

如果读者从未读过数字电子学方面的书,那末可先从第三章数字逻辑读起。

第二章 微处理机的术语

本章对书中所用到的有关微处理机的术语给出简短的定义。它们的详细内容将在有关章节中介绍。

绝对地址(absolute address)

访问存贮器指令中存贮单元的实际二进制码地址。

存取时间(access time)

自调入数据至取出数据所需的时间。或自送入数据到完成存贮所需的时间。

累加器(accumulator)

用来存放算术和逻辑运算结果的中央处理单元(CPU)中的主要寄存器。有些微处理机具有两个累加器或把多个寄存器用作累加器。

加法器(adder)

由门电路组成的、用来完成两个二进制输入信号相加的逻辑电路。一个全加器通常包括进位输入、总和及进位输出。

地址(address)

标识存贮单元或输入/输出口的二进制数字。通常地址是一个12位或16位二进制数。此数也可以转换成8进制或16进制数。

地址译码器(address decoder)

一种逻辑电路，其输出能选择出一个被二进制数字寻址的

器件。这个二进制数即为译码器的输入。

算法(algorithm)

解决某一问题的一套规则(通常指数学规则)。

算术及逻辑运算部件(ALU)

能够对数据字完成算术和逻辑运算的逻辑电路。ALU 为微处理机提供了加、减、移位、加 1、减 1、“与”、“或”及“异或”的功能。

体系结构(architecture)

系指微处理机的内部结构。它表示计算机中寄存器、ALU、控制逻辑等的各种内部连接方式。不同形式的体系结构构成不同的微处理机。

美国信息交换标准码(ASCII)

ASCII 是用于字母和数字数据串行传输的 8 单位码。头 7 位表示 128 个标准 ASCII 的字符(见附录 B)。第 8 个数字位是用来检错的奇偶位。在 8 个数据位前有一个起始位，数据位后面有一到两个终止位。

汇编程序(assembler)

一种符号翻译程序，它把助记符(符号码或缩写的机器操作码)转换成机器码。汇编程序允许程序员采用名称(标号)而不用数值去编址。汇编程序把标号转换成绝对地址。

异步(asynchronous)

在串行通讯中，发送字和接收字之间的时间间隔没有规律性。而每一个字的起始位与各个字的读出同步。

BASIC 语言

是一种编制程序的人机对话语言，最初由达特默思学院提出。这种语言允许使用简单的英文字和普通的数学符号来完成解题所需的算术和逻辑运算。

波特速率(baud rate)

表示数据传输速率。对异步的字来说，这是一个字内每位数据的传输速率。

双向传输线(bidirectional line)

用于传送和接收数据的数据线。在双向数据总线上，信号可在两个方向流通。

二进制数字系统(binary number system)

用符号 1 和 0 表示量值的数字系统。数字逻辑就使用这种二进制信息。

双极型(bipolar)

采用正掺杂和负掺杂两种半导体材料所组成的普通晶体管结构(PNP 和 NPN 晶体管都是双极型晶体管)。

比特(bit)

二进制数中的位。是数字系统或微处理机中最小的数据单位。

总线(bus)

微处理机和其它有关装置间进行数字通讯的通道。总线一般传输双向数据或单向地址信息。总线通常具有 4、8、12 或 16 条并行线，每条线传输 4、8、12 和 16 位字中的一位。

字节(byte)

由 8 位二进制码组成的数字。

进位(carry)

进行加法运算时产生的溢出。在微处理机中，大于所允许的字长时，就产生进位。

芯片(chip)

上面制有集成电路的小块方形硅片。芯片这个词通常可以和集成电路一词互换。

时钟(clock)

为微处理机、存贮器和 I/O(输入、输出)设备提供公用时钟信号的脉冲发生器。时钟通常分为单相或双向(1 ϕ 或 2 ϕ)。

代码(code)

在微处理机中用来表示数据和编写程序的一系列字母和符号。

组合逻辑(combinational logic)

由门构成的数字电路,用来执行所需要的功能。组合逻辑电路没有存贮能力。

编译程序(compiler)

把用英语语句和普通代数式构成的对某个问题的逻辑解,编译成某一微处理机的一连串机器码指令的程序。

反码(complement)

对逻辑信号求反。就是说,使逻辑 1 转换为逻辑 0 或者把逻辑 0 转换为逻辑 1。

互补型金属氧化物半导体(CMOS)

用来制造某些微处理机和存贮器的一种功耗极小的 MOS 工艺。

控制台(console)

微处理机系统的操作面板和终端设备。

中央处理单元(central processing unit)

指包括寄存器、算术及逻辑运算部件、计时和控制电路在内的微处理机所有逻辑电路。

失控(crash)

微处理机完全丧失控制。当指令顺序发生混乱时,微处理机通常失控。

控制用只读存储器(Control Read Only Memory)

微处理机中用来完成控制顺序的内部存贮器。