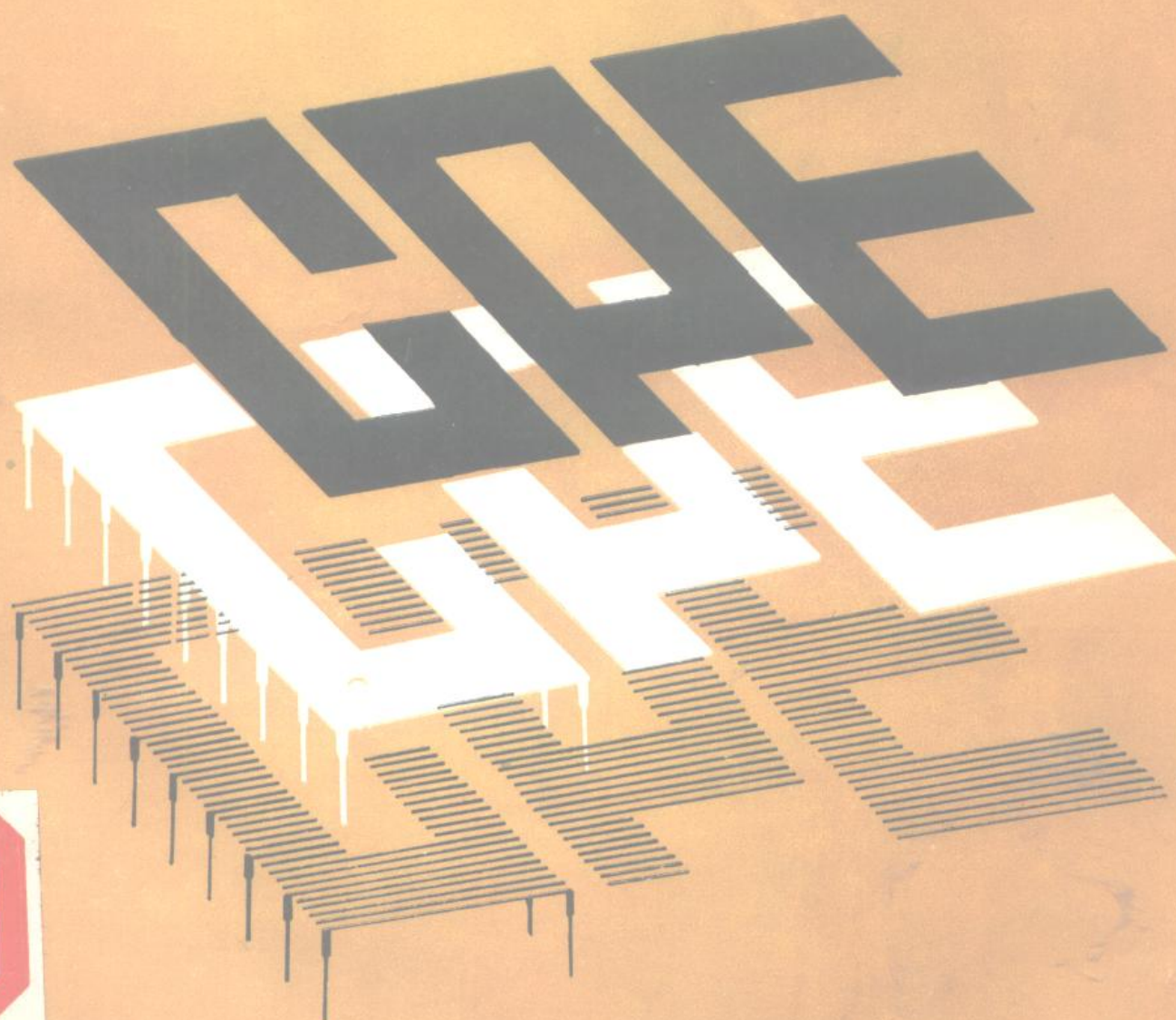


微处理器应用实验

[美] T.W. 戴维斯 著



机械工业出版社

微 处 理 机 应 用 实 验

〔美〕 T.W.戴维斯 著

《微处理器应用实验》翻译组 译



机 械 工 业 出 版 社

本书介绍了40个微处理机应用实验，各实验均可独立进行。每个实验包括预备知识、实验准备、实验内容及思考题四个部分。实验中使用TTL集成电路、D/A和A/D转换器、七段数字显示器以及键盘等常用设备。给出的接口线路与程序框图是通用的，不限定使用的机种。

本书可供从事微机应用工作的人员参考，也可供微机实验教学人员使用。

JS238/05

EXPERIMENTATION WITH
MICROPROCESSOR APPLICATIONS

THOMAS W. DAVIS

Reston Publishing Co., Inc. 1981

微处理机应用实验

〔美〕T.W.戴维斯 著

《微处理机应用实验》翻译组 译

* 45

机械工业出版社出版(北京阜成门外百万庄南街一号)

(北京市书刊出版业营业许可证出字第117号)

中国农业机械出版社印刷厂印刷

新华书店北京发行所发行·新华书店经售

*

开本 787×1092 1/16·印张 9 1/2·字数 228 千字

1985年9月北京第一版·1985年9月北京第一次印刷

印数 00,001—11,640·定价2.35元

*

统一书号: 15033 5903

9810102

译 者 的 话

近年来，微处理机及其应用的发展十分迅速。微机由于其廉价、轻小、灵活和可靠等特点，现已深入到工业、科技以至日常生活的各个领域，对企业的技术改造和产品的更新换代起着日趋重要的作用。学习和应用微机的人越来越多。

在微机应用技术的教学、培训或自学过程中，除了系统地学习有关教材，消化技术资料之外，上机实验是个不可缺少的环节。这样才能对学过的知识加以巩固，融汇贯通，为微机的实际应用工作积累经验与技巧。为了对学习和从事微机应用的人员有所帮助，我们翻译了该书。

书中的 40 个实验均可独立进行。针对不同应用对象，在指令与中断系统、接口线路、算法与程序结构几个方面各有所侧重。每个实验都有一预备知识部分，对应用对象、总体结构方式、软硬件权衡等问题进行了较详细的介绍与讨论。各个实验除需用微机本身之外，只使用 TTL 组件、A/D 和 D/A 转换器、七段显示器及键盘等常见设备。接口线路及程序框图都是通用的，对使用的机种没有限定。原作者建议：只使用一种微机进行实验，以避免多次修改硬、软件的麻烦。确实，掌握了设计方法和使用技巧后，改换使用的机种是件方便的事。

参加本书翻译工作的有赵经政、蒋冰、邱永甫、韩景文四人，由赵经政审校。殷切希望读者对本书的翻译提出批评和指正。

目 录

实验 1	计算机和微处理机的发展历史	1
实验 2	熟悉指令	3
实验 3	指令定时	6
实验 4	二进制加法和减法	11
实验 5	输入/输出和逻辑 仿 真	17
实验 6	二—十进制的运算及变换	24
实验 7	按钮开关接口和二—十进制计数	29
实验 8	逻辑设计	33
实验 9	将串 行 的 ASCII 码 变 换 为 BCD 码	37
实验10	键盘输入	40
实验11	十六进制的加法/减法 程 序	45
实验12	微处理机软件	48
实验13	二进制乘法	51
实验14	算法的设计——求平方根	54
实验15	数组和表格	57
实验16	决策过程	62
实验17	栈操作	66
实验18	十进制到二进制及二进制到十进制的转换	69
实验19	七段显示器和多路转换技术	73
实验20	硬件诊断技术	76
实验21	7490 十 进 制 计 数 器 测 试 装 置	79
实验22	中断特性	83
实验23	矢量中断和优先级中断	86
实验24	一个以中断方式工作的实时钟	89
实验25	设计过程	92
实验26	数/模 (D/A) 转换 原 理	95
实验27	模拟量至数字量转换技术	101
实验28	自动选量程数字电压表	106
实验29	数字电容表	109
实验30	数据终端接口	111
实验31	随机数的产生	118
实验32	警戒系统	121
实验33	数字式转速计在汽车中的应用	123
实验34	交通控制	126

实验35	实时数据采集	130
实验36	步进马达控制	133
实验37	计算机之间的通讯	135
实验38	二—十进制码与摩尔斯码之间的转换	139
实验39	定时与控制——电梯仿真	142
实验40	日历程序	144

实 验 1

计算机和微处理机的发展历史

实 验 目 的

本实验的目的是深入了解和评述微处理机及其演变过程。此外，要查阅相应的文献资料。

预 备 知 识

计算机能达到目前的技术水平，其发展的全过程可追溯至公元前100年至公元1300年期间出现并广泛使用的算盘。虽然算盘的出现比原始的石子计数是一巨大进步，但到它广泛使用，却经历了好多个世纪。在这之后，各种发明就一个接一个地出现了。

一般公认，“计算机”是由 Charles Babbage 在 1833 年提出的。他所设想的计算机由蒸气驱动，并由许多齿轮和机械零件组成，有 50000 个数的存储器并可执行可编程序的指令。他这一套设想当时被许多人认为是荒谬的，因此没有实施。

第一个真正的数据处理计算机是由 Herman Hollerith 制造的，并将其用于 1870 年人口调查的制表工作，使本来要 20 年才能完成的工作只用了 3 年就完成了。

1937 年 George Stibitz 在他厨房的桌子上用真空管、普通电池、继电器和开关制造了一台二进制加法器。这是第一次将二进制运算及电子逻辑用于数学计算。George Stibitz 后来还研制了其它的硬件。

经过了 7 年的努力，第一台大型自动数字计算机在 1944 年制成了。该机的自动程序控制计算器是由 Howard Aiken 研制的，可用 0.3 秒做一次加法或减法运算，6 秒做一次乘法运算，16 秒做一次除法运算。

第一台电子数字计算机 ENIAC 是在 1946 年制成的，该机占地 1500 平方英尺^①，重 30 吨。使用了 18000 个电子管，500000 个开关。耗电为 130 千瓦，每秒能进行 5000 次加法运算或 300 次乘法运算。

1950 年，只有 60 台计算机在运行，其中有一台是国家标准局制造的。同时期的 SSEC 计算机用了 150 小时计算出一个原子能方面的问题，这个问题若由一个数学家用手算求解，将要用 1500 年。

1954 年约有 50 家公司用上了计算机，UNIVAC I 和 IBM 701 就是在那时出现的。

NCR 计算机公司推出了第一台全晶体管化的计算机，它的 CPU 用了 8000 个二极管和 4000 个三极管，共有 4800 个字的存储器。国际商用机器公司 (IBM) 于 1964 年用 360 系列机形成其计算机“族”。由于电子领域的小型化变革，这些第三代的计算机与以前相比，

^① 1 平方英尺 = 0.093 平方米——译者

体积小而速度快。

小型计算机工业可追溯至 60 年代中期, 我们如今看到的这类小型计算机, 通常认为是由通用设备公司 (DEC) 最先制造的。到 1971 年, 世界上已有 100000 台计算机在使用。

1971 年英特尔 (Intel) 公司制成了第一台微处理机。该器件相当于中央处理单元 (CPU), 全部做在约 100 密耳见方[⊖]的单片集成电路上。耗电为 450 毫瓦, 可以执行 46 种以上的各种指令, 这些指令大多是微秒级的。

对微处理机发展这一课题的深入探讨已超出本文的范围。这些器件的发展历史中, 有许多有趣的事实, 可以使我们对这一重要课题有更深入的理解。

实 验 准 备

对如下每一问题, 作出有关微处理机/微计算机演变的最适当的答案。

1. 第一台微处理机是谁研制的? 在哪一年?
2. 被公认的第一台微处理机是哪一台?
3. 第一台 8 位微处理机是由_____于_____年研制的。
4. 首次推销 16 位微处理机的是哪一家公司?
5. 用图表来描述组成计算机系统的主要部件 (如继电器、电子管等) 的变化。
6. 哪家厂商, 特别是用什么装置首次推销符合军事要求的微处理机?
7. 单片集成电路中密度最高的半导体存储器是哪一种?
8. 最早带微处理机的电子游戏机是哪一台?
9. 用曲线示出 8 位微处理机从出现到目前在价格方面的下降趋势。
10. 目前制造多少种 8 位微处理机? 都是由谁制造的? 16 位机的情况呢?
11. 家用计算机由谁首先推出? 它的基本价格是多少?
12. 在研制微处理机方面哪一家或哪些家的声誉最好?
13. 小型计算机与微型计算机之间的主要差别是什么?
14. 哪些因素导致了微处理机被广泛接受?
15. 微处理机与微计算机之间仍有差别吗? 这差别会一直保持吗?
16. 对未来的微处理机/微计算机领域能作何设想?

实 验 内 容

在实验课上多花些时间仔细讨论实验准备中出现的问题。对研究中出现的有关情况做一般性的讨论即可。

思 考 题

1. 写一个简短的摘要, 略述有关微处理机的主要发现。
2. 画一个编年图表, 略述微处理机/微计算机技术方面的重大突破。

⊖ 1 平方密耳 = 0.00064516 平方毫米——译注

实 验 2

熟悉指令

实验目的

本实验的目的是熟悉基本指令系统、寻址方式及条件码标志。

预备知识

微处理机各种指令的实际数目可为 40 条到 250 条以上。要使用这样庞大的指令系统，初看起来，象是难以着手的。但是，将这一指令系统分为若干个基本组时，就会发现它不仅易于使用，而且还是易于记忆的，许多指令只不过是其基本指令的几种变种。

一条指令的第一部分指定将要实现的操作方式，称其为操作码或 OP 码。在标准的 8 位处理机中，整个字都作为操作码用。这样就可以产生 256 种不同的组合。但是，其中有些组合是不用的。字长较长的微处理机，如 16 位机，可以利用字长的一部分来指定寻址方式、变址或特殊操作。在 8 位机的指令中，要用别的字或字节来指定操作数、地址，或是在指令执行时用来获得地址的某一信息。

计算机只能执行以二进制形式表达的指令。这种用二进制表示的形式称为机器码。以字母代表机器码一般为一至五个字符，常用这种记忆符来表示一条指令，这样可使程序易读。能把这种记忆符转换为二进制机器码的程序称为汇编程序，这在后面的实验中将要讲到。实验中大部分指令的说明是用记忆符表达法。把记忆符转换成二进制表达式是一个简单的过程，而二进制表达式又常用八进制或十六进制符号来表示。用一张表来完成这种转换是很方便的，在没有汇编程序可借助的情况下，可用此表进行汇编。

各种指令基本上可分为以下几大类：

数据传送类：在存储器中取出或存入数据，在处理机内部传送指令，诸如从累加器到累加器、从累加器到暂存寄存器或输入/输出口。

算术类：加、减、增一、减一、带进位加、求补、比较，有些处理机中还有乘和除指令。

逻辑类：与、或、变反、循环移位、异或等。

分支类：条件和无条件跳转或分支指令、子程序调用、返回及中断指令。

栈操作类：压入、弹出及设置栈指针。

处理机控制类：包括停机、等待及等待中断指令。

微处理机内存储器的结构是每个字有独立的地址。数据和指令的地址使用相同的格式。因处理机具有多种寻址方式，对某一指定地址访问时可有多种过程。寻址方式的分类列举如下。虽然一台处理机不会具有下列全部寻址方式，但所有的处理机都是具有多种寻址方式

的。

直接寻址——在这种情况下，指令的第二字或第二与第三字包含了被操作数据或单元的准确地址。

立即寻址——指令的第二字或第二与第三字即包含着将被操作的数据本身。即指令不仅指定了操作方式，还包含着将被操作的数据（操作数）。

间接寻址——A 寄存器或指令可包含一指针地址。该地址所指向的单元中储存着准确的地址或称绝对地址。

隐含寻址——或称寄存器寻址，指令的操作码即指定了数据的地址。这种形式的指令通常为单字长指令。

变址寻址——地址的设定与一特定的变址寄存器有关，该寄存器包含了地址中的一个变化量。

相对寻址——地址的设定是相对于程序执行的现行地址（即程序计数器的内容）而言的。特殊的规定可限定从指令计数器当前值至被寻地址的距离，这一距离是双向的。

完成同一操作的存储器访问指令可采用上列的一种或多种方式。这是使指令数目庞大的原因之一。

在研究指令操作时，条件码（有时称为条件标志或处理机状态字）也是一个重要的项目。在微处理机系统中，条件码是决策过程中最基本的影响因素。并不是所有的指令都对该标志起作用。诸如比较、加、与、或等指令用来设置恰当的条件码以使分支和转跳指令得以执行。

下面给出的条件码是有代表性的例子，在大多数微处理机系统中可以见到。

零——若指令执行的结果使累加器内容为零，则该标志置位。

符号（亦称负号）——该位的值与累加器的最高位相同。该值应保存，虽然它只是在某些指令执行后才改变。

进位（亦称借位）——若指令执行结果因加法导致溢出或因减法导致借位时该标志置位。它也可因执行右环移或左环移指令而被改变（这可用来检测数据字中各位的状态）。

辅助进位（亦称半进位）——若指令执行产生了从第三位到第四位的进位时该标志置位。它通常用于累加器十进制调整指令。

奇偶性——指令执行结果若符合偶校验（等于零的位数可被“2”整除），该标志置位。若符合奇校验，该标志复位。

微处理机中的其它条件可作为处理机状态字的一部分。包括补码溢出、中断状态等。

实 验 准 备

1. 根据上述方法将微处理机现有的各种指令加以分类。
2. 确定你所使用的微处理机的各种寻址方式。
3. 为指令的机器语言编码设计或找一种速查索引。
4. 对各环移指令按照其对进位位或溢出位的作用进行分类。
5. 列举可把数据装入累加器的各种方法。
6. 当累加器的第2位（最低位是第0位）为“1”时，要求程序转向预定地址。至少用两种方法完成这一任务。

8210108

7. 将指令按单字长、双字长和三字长进行分类。
8. 列举将数据送入存储器的各种方法。
9. 哪些指令会改变堆栈的内容?
10. 转移或分支指令与子程序调用指令之间的基本区别是什么?
11. 怎样才能将所有的条件码同时置“0”?
12. 至少列出四种可改变栈指针的指令。
13. 用一条指令或一组指令完成下列指定的操作。
 - a) 累加器清零。
 - b) 对累加器中的数求补。
 - c) 清除进位位或溢出位。
 - d) 将累加器中除第4位外的所有位清零。
 - e) 将栈指针初始化为某预定地址,并将累加器内容入栈。
 - f) 将指定的内存单元加“1”。
 - g) 访问一组连续的内存单元,并将其中数据每次一字地送到累加器中。
 - h) 使用不超过三条的指令将累加器置“2”。
 - i) 用二种方法将累加器的内容乘“2”。
 - j) 把二个内存单元的内容搬到别的单元中去。
14. 写一个短程序,将内存中的两个数相加后,结果仍放回内存。
15. 写一个程序,连续搜寻内存单元,直至找到F4_H为止。

实 验 内 容

1. 熟悉你使用的微机系统的程序输入方法。
2. 设计一个能方便地显示累加器内容和进位位的方法。
3. 检验实验准备步骤13中的各种操作。
4. 验证实验准备步骤14中程序的正确运行。
5. 验证实验准备步骤15中程序的正确运行。

思 考 题

1. 给微程序设计下一定义。
2. 机器语言编程和汇编语言编程的区别是什么?
3. 怎样才能将条件码显示出来?
4. 为你系统中的全部可用内存空间画一分配图。分别标出被RAM、PROM、I/O等占用的地址。
5. 画一个操作流程图,描述输入机器语言程序的过程。
6. 解释下列术语,并说明它们与计算机软件的关系:编译程序,解释程序,翻译程序。

实 验 3

指令定时

实验目的

本实验讨论有关时钟的各种情况、指令定时及延时循环。

预备知识

要正确地执行各种指令和由指令组成的程序，就要定序地进行一系列具体的操作。这一定序工作是由控制单元内部的定时线路完成的。定序所使用的时基是由一种外部线路提供的。用晶体或 R/C 网络连同时基一起决定定时脉冲的频率。晶体是最常用的，因为程序中若使用软件延时的方法，就必须知道每一条指令执行所需用的精确时间。

根据所用的微处理机，振荡频率可从 1MHz 到 8MHz。实际上，定序脉冲是从基本频率得来的，基本频率只影响程序执行的速度而不影响其运行次序。时钟频率的上限在制造厂商的技术规格中有规定。当处理机定时源由外部供给时，一般在技术规格中指定最高频率、脉冲间隔时间以及脉冲宽度。这些就组成了所谓主控时钟。例如 Intel 8080，需要一个二相的、非重叠的时钟。指定了下列各项的最大值和最小值：时钟周期、上升和下降时间、脉冲宽度（ ϕ_1 和 ϕ_2 都有）、 ϕ_1 和 ϕ_2 之间的间隔以及 ϕ_1 和 ϕ_2 的前沿之间的间隔。可采用各种集成电路芯片作为主控时钟，它们将产生适当的信号。另有一些处理机，在处理机芯片上就包含了主控时钟。

每一个制造厂都有一独特的方法来分解每条指令的周期（执行一条指令所需的总时间）。每一条指令的周期可以分解为各个不同的机器周期，而每一机器周期又是由许多主控时钟周期组成的。因不同的指令所完成的操作不同，故指令周期的长短也不同。大部分系统中都有的三种主要的机器周期是取指、延时和执行。

在取指周期中，从存储器中取来操作码。在控制单元的译码部分对操作码译码。处理机一旦知道了将要执行的指令类型，则其余的机器周期的长度和数目就能决定了。然后从内存中得到其余的字节。

如果在操作码后紧跟着的字节中包含着附加的地址信息，就要进入延时周期。在这种情况下，数据的地址可从内存中得到。当使用间接寻址方式时，由指针地址来获得最终的寻址信息。

执行周期是运行的最后一个周期，这时，操作数已从内存中得到并对数据进行运算，运算包括加、或、减等。

显然，每条指令的执行，并不一定要处理机把各机器周期都用到。一条转跳指令就不需要延时或执行周期，因为修改指令进程所需的所有信息在取指周期中就可得到。下面一系列

操作是一条间接加指令的典型指令周期。

时钟周期	操作	机器周期
1	程序计数器内容送内存	取指
2	从内存取得操作码（间接加）	取指
3	译码	取指
4	建立其尔定时，取得指针地址信息	取指
5	利用指针地址获得数据地址	延时
6	数据地址送内存	延时
7	从内存取数（存入暂存寄存器）	执行
8	与累加器内容相加	执行
9	将结果置于累加器并置相应标志	执行

下面的定时图表示各种定时信号和时钟信号的相互关系。在这个假定的例子中，执行这条指令共需九个时钟周期（在具有二相时钟的系统中，定时仅相对于一相而言）。若本例中基本时钟频率为 2MHz（由晶体控制），则总的执行时间为 4.5μs。

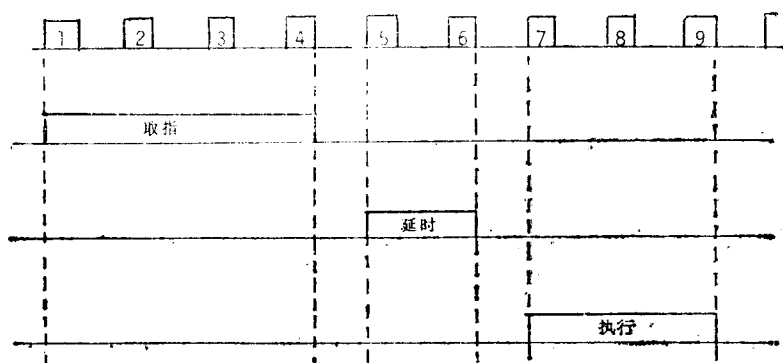


图 1 指令周期

指令系统中执行每条指令所需时钟周期数都是已定的。因此，精确的执行时间是可以决定的。有几条指令可能列出二种时间，如条件转移或条件分支指令。指定的长的执行时间是在进行分支或转移时需要的，短的时间则是当条件不成立不进行分支或转移时需要的。

有了对执行时间的认识，结合晶体控制的时钟就能产生精确的定时循环，可用于程序延时、精确的输出定时以及各种其他编程使用。实际上可研制子程序以供后面程序中特定的延时用。

在写定时程序时，无操作指令（NOP）常用来凑准延时时间以产生特定的延时。虽然无操作指令并不完成什么操作，但它确实需要时间来执行。此外，NOP 指令还常用于机器语言编程的场合，这样可以保留空间以备填写遗漏的指令。

一种简单的延时程序以流程图的方式示于图 2。程序写成子程序形式，在每一框的右边指令执行时间简单表示为 t_1 、 t_2 ……等。内循环程序提供延时时间的主体部分。假定计数器 N 包含了延时程序的长度。则总的延时时间可表示为：

$$T_D = t_1 + (t_2 + t_3 + t_4 + t_5) N + t_6$$

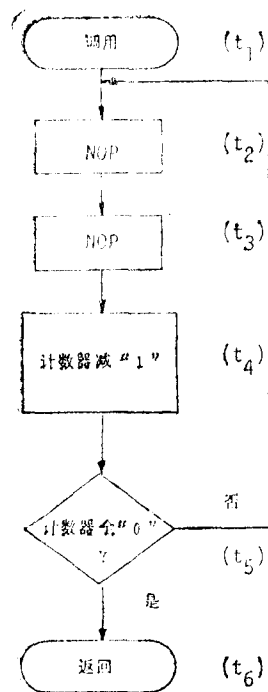


图2 简单的延时程序

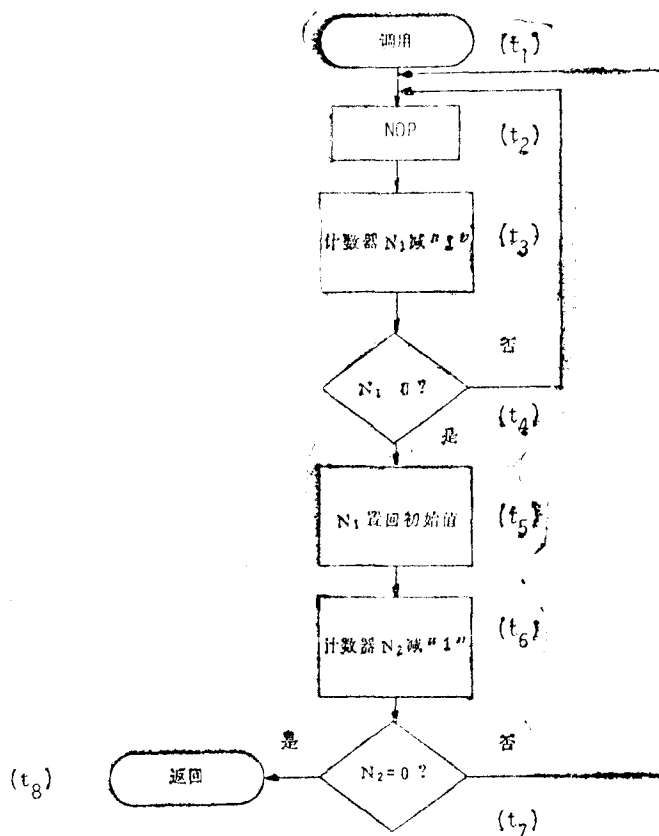


图3 双重定时循环

为增加总的延时时间，NOP 指令提供了必需的附加延时。 t_1 和 t_6 是固定的，不作为循环过程的组成部分。该程序可提供的最短延时为 $N = 1$ ，延时时间为 $t_1 + t_2 + t_3 + t_4 + t_5 + t_6$ 。最长延时时间决定于 N 的最大值（一般为一字长）。该公式还可用于计算产生指定延时所需的 N 值。因 t_1 和 t_6 是已知的，则：

$$N = \frac{T_0 - t_1 - t_6}{t_2 + t_3 + t_4 + t_5}$$

附加的 NOP 指令可以加在循环内有效地增加所需延时的时间 (Nt_{NOP})，也可加在循环外以凑准时间。必须指出，当该程序重复使用时，在每次调用前应使计数器 N 的值复位为初值。

一种双重定时循环的流程图如图 3 所示。这一过程使延时时间显著加长。

单循环延时的延时时间可以从几微秒到几毫秒。双循环程序可得到几秒的延时，在不使用大量内存的情况下几小时的延时也可达到。使用这些延时程序的重要性在下一例子中可明显看出。实际上有些处理机甚至没有内部的延时指令。

用计算机的语音合成法长期以来使许多人感兴趣。市场上流行的设备可以产生象人说话一样的声音。它们用于供盲人用的计算器、象棋游戏甚至可用于某些家庭计算机系统。这类设备利用存在内存中的信息定时输出信号，这只是对某一位进行开或关的操作。其他的系统还可以由计算机系统选择，把预先录在磁带上的信息再放出来。

作为一种语音合成法的方式的介绍, 也为了熟悉指令定时, 将研究一个单音的产生。虽然计算机不能很方便地产生正弦波, 但是只要简单地控制一位输出的开或关, 就能产生某一频率的方波。

累加器的开关作用可由连续的增量来实现。因为与其余各位无关, 累加器的计数实际上可从 0 到 255₁₀。这将使最低位在 0 和 1 之间“振荡”。延时程序可用来设置开和关的时间长短。还要假定输出口有锁存, 因此输出数据仅在重写后才改变。

这一过程的流程示于图 4。延时时间可利用前面讨论过的过程进行计算。

延时时间可按下式决定:

$$T_0 = (t_2 + t_3) N + (t_1 + t_4 + t_5)$$

这就是脉冲开(或关, 两者必居其一)的时间。因为循环是对称的, 因此波形总周期为 $(T_0) \times 2$ 。时间 t_1 、 t_2 、 t_3 ……是根据微处理机的时钟频率决定的。利用以上这些关系, 输出音调的频率

$$f = \frac{1}{2(T_0)}$$

单音发生的频率可由下表所列数据来决定:

音符	频率
C	261.63
D	293.66
E	329.63
F	349.23
G	392.00 (低八度时为196)
A	440.00
B	493.88

利用秒延时循环来决定每一音符的持续时间, 可以把各种延时时间存于内存中, 这样就能产生曲调。必须指出, 上表所列音符的高八度或低八度可以简单地用表中的频率乘以“2”或除以“2”来得到。

对输出波形的监听可采用各种方法。在有些情况下, 可以在输出位简单地接一喇叭。程序停在输出位为逻辑“1”时喇叭不应当出现对地短路现象, 为此可接入一电容。

在音调发生系统中, LM 380 音频功率放大器可用来驱动喇叭。图 6 表示了其接法。

对于有二个输出的处理可采用图 7 二种接法中的任一种。它们可用来产生二个同时发生的音调。

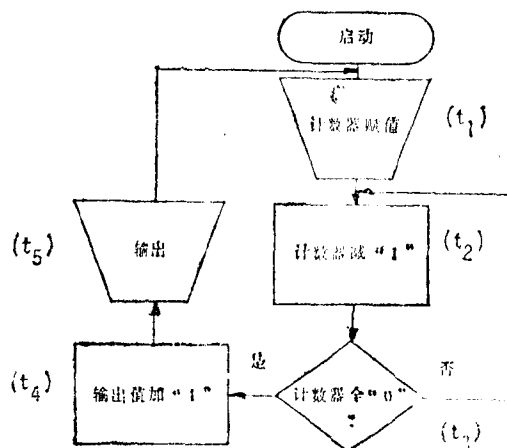


图 4 单音发生程序

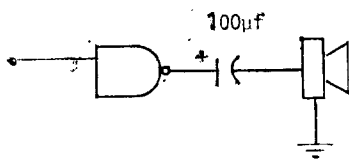


图5 喇叭连接

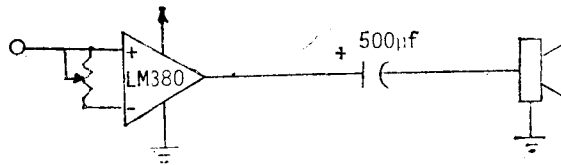


图6 音频功率放大器

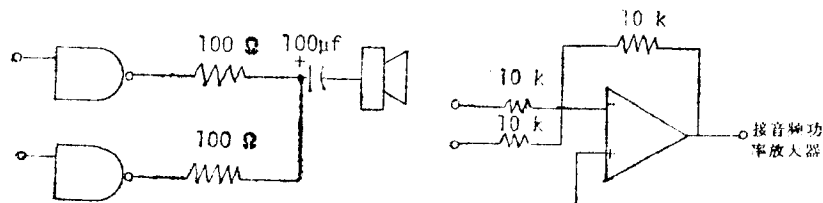


图7 二个单音发生器

实验准备

1. 按照前面讨论过的方式编制作子程序的单循环延时程序。应提供一输入口以便在调用时可调整延时时间。
2. 重复上述步骤，编制一个双循环的延时程序。
3. 根据你所用微处理机的时钟频率计算延时程序（单循环和双循环的）的最大和最小延时时间。
4. 利用前面讨论的单音发生方法写出产生音阶的程序。
5. 许多人都熟悉威斯敏斯特（Westminster）钟声，编制程序使处理机演奏这一曲调（提示：所需音符为E、C、D、G（低八度）、G、D、E、C）。
6. 在单音发生程序中，累加器的初值是多少，这一点一定要知道吗？

实验内容

1. 对实验准备步骤1中写的单循环子程序，用实验方法确定其最小与最大延时时间。
2. 对双循环延时子程序重复步骤1。
3. 用示波器或频率计检验音阶中各音符频率的正确性。
4. 因为图4程序中累加器是每个定时循环后加“1”的，请你来确定数据位 $d_0 \sim d_7$ 间的关系。

思考题

1. 用于电话的音调发生器实际上是用两个不同的音调合成的。用滤波器和频率计分解这些音调的两个合成频率（提示：音调发生器的每一行和每一列都有其固有的频率，每个按键是某二个频率的合成）。
2. 根据以上步骤1测得的数据，用本实验的程序产生这些音调。
3. 写一个短程序，用本实验的音调发生及延时程序产生HELLO这个词。
4. 怎样才能将输出信号改为正弦波输出？

实 验 4

二进制加法和减法

实 验 目 的

本实验的目的是熟悉一下二进制数、“2”的补码记数法以及二进制加法和减法。

预 备 知 识

用于表示数的位置记数法是大家熟知的。通常应用这种方法将其它进制的数转换成十进制数。实际上，十进制采用下述位置记数法：

$$546_{10} = 5 \times 10^2 + 4 \times 10^1 + 6 \times 10^0$$

二进制采用：

$$10111_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 23_{10}$$

八进制：

$$374_8 = 3 \times 8^2 + 7 \times 8^1 + 4 \times 8^0 = 252_{10}$$

十六进制：

$$10FD_{16} = 1 \times 16^3 + 0 \times 16^2 + 15 \times 16^1 + 13 \times 16^0 = 4349_{10}$$

以上全都采用了位置记数法的形式。

二进制加法基本上按照十进制加法的一套规则。但也有一些特殊情况：

$$0 + 0 = 0$$

$$1 + 0 = 1$$

$$0 + 1 = 1$$

$$1 + 1 = \underline{10} \text{ (特殊情况)}$$

下面划线的进位表示这种特殊情况，在这里不允许出现结果为2。采用十进制同样的规则来进行多位二进制数的加法是很方便的。

$$\begin{array}{r} 10011101 \\ + 01011011 \\ \hline 11111000 \end{array}$$

这种运算所需的逻辑线路不太复杂，所有的微处理机系统都有此功能。这一运算由称之为半加器的线路完成。如果有可能进位到下一个高位时，例如上例，则要使用一个全加器。在电子计算机系统中，两个八位数相加用八个全加器就够了。

如果不计负数，单一的八位数可表示的十进制数的范围是0到255₁₀。当采用带符号的二进制数时，要保留一位用来作符号指示。最高有效位为零表示正值，为“1”表示负值。采用这种约定时，必须记住：零在系统中被认为是正数。