

527758

5087 作
23025

译文专辑

527758

微处理机与微型计算机

37
25

上海市仪表电讯技术情报所

目 录

结构与工艺

微处理机芯片结构.....	(1)
可编程序的微处理机.....	(10)
采用 SOS 工艺制作的单片微处理机.....	(16)
性能优异的 Z-80 微处理机	(22)
预告第三代微处理机问世的 Z-80 芯片组	(38)
MCS-48 微型计算机系列.....	(44)
关于微型计算机系统设计中的共性问题.....	(59)
双极微处理机结构与应用介绍.....	(65)
12 位 CMOS 微处理机结构与应用.....	(71)
16K 位随机存取存储器.....	(83)

接口

微处理机的外围接口标准.....	(85)
微型计算机的接口元件.....	(99)

软件

研制 8080A 型微处理机用的组合式软件	(103)
用于简化微型计算机程序设计的高级语言	(108)

测试技术

简化微处理机测试的两种新方法	(116)
微处理机芯片测试方案	(123)
微型计算机系列的测试技术	(128)

应用

国外微处理机应用十四则	(141)
微处理机扩展了数字万用表的功能	(169)
采用微处理机集成电路改进仪器性能	(176)
用于化学过程控制的微型计算机系统	(182)

微 处 理 机 芯 片 结 构*

本文谈的是系统结构和芯片结构，并分析了它们对系统性能的影响。此外，文中还叙述了目前通用的几种芯片结构的基本特点与优点，进而为微处理机的应用铺平道路。

现有两种基本的数字计算机结构。一种具有直接存储器存取能力，另一种则没有。究竟使用哪一种，这要看所采用的特定芯片组而定。

没有直接存储器存取的微处理机系统基本结构

图2.1示出目前使用的最普通的结构，这种结构没有直接存储器存取。但是，它却能提供分散控制。由CPU控制存储器和输入/输出通道之间的通信。而这些输入/输出通道则控制外围设备与CPU之间的通信。

具有直接存储器存取的微处理机系统基本结构

图2.2示出具有直接存储器存取的微处理机系统基本结构。这种结构通常需要配备两块类型不同的芯片。一块是中央处理机，另一块是能处理输入/输出任务的特定存储器接口。对需要高输入/输出率的应用来说，如通信系统或微计算机网络，看来直接存储器存取结构最为理想。

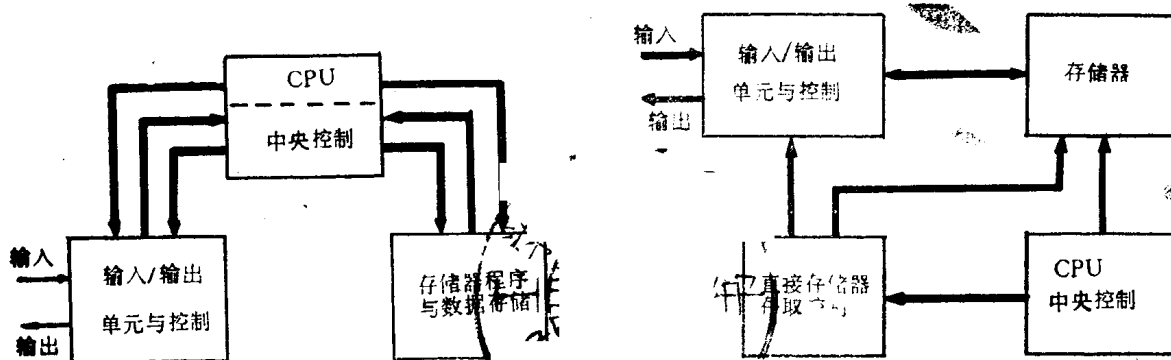


图2.1 没有直接存储器存取的微处理机系统基本结构 图2.2 具有直接存储器存取的微处理机系统基本结构

大多数微处理机常有的单元

不管采用哪一种结构，处理机应能按适当的顺序进行取数、译码和执行指令。因此，大多数微处理机都含有一定的共同单元，如图2.3所示。

必须具备寻找存储器地址的方法。指令一旦从存储器送至处理机，它通过处理机而被保留在指令寄存器里。此外，还需要配备指令译码电路，以确定到底是什么指令，并以产生执行指令所需的信号。某些指令需要对数据进行算术运算或逻辑运算。因此，处理机应具有一个运算部件（ALU）。其他的一些指令还可能包括暂存中间结果，所以还得配备几个各种服务性的寄存器。

此外，必须具备按次序寻找存储器地址的方法，这样就得有一只程序计数器。另外，

* "Chip Architecture", The Microprocessor Handbook, Prepared by The Texas Instruments Learning Center. pp.74-92, 1975. 谢官木译、孙绍凤校

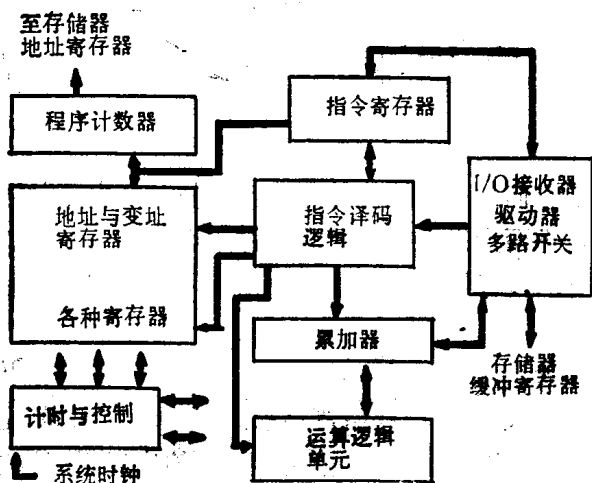


图2.3 大多数微处理机常用的单元

- 减
- 十进制运算
- 乘
- 除
- “异”
- 比较
- 移位
- 串行对并行
- 位长
- 加
- 逻辑“与”
- 逻辑“或”
- 运算移位

图2.4 运算逻辑单元指令系统

还需要进行地址修改，如变址或子程序定位。因此，在处理器中还配备有各种地址寄存器。

运算部件基本指令系统

通过对每只处理机分系统的扼要讨论，揭示了另外的一些细节。图2.4列举了运算部件的能力。这些能力包括加法运算、逻辑“与”、逻辑“或”和一个算术移位。有些处理机具有更大量的运算和逻辑能力。

12位并行输出的管脚布局

当需要确定处理机的寻址时，就会面临折衷考虑的问题。对一个能从4096个字中选择一个字的12位地址来讲，在处理机的管壳上的12个不同的地址输出管脚能提供器件的全部12位并行输出。这种布局示于图2.5。

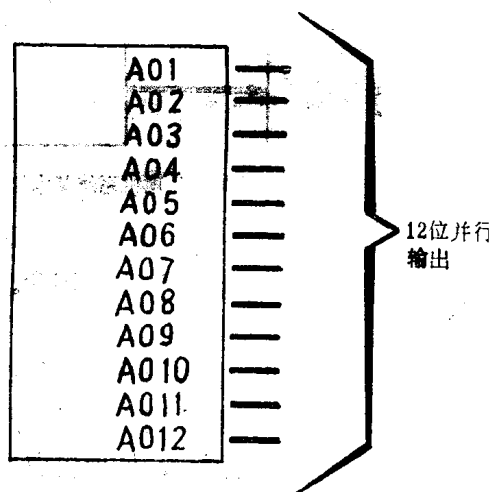


图2.5 12位并行输出的管脚布局

4位并行输出的管脚布局

有些处理机有时只有4个地址输出管脚来提供4位字符组地址，如图2.6所示。当然，这种处理机提供12位所化的时间要比12位并行输出的处理机长三倍。但芯片可以是一个体积较小的管壳封装。

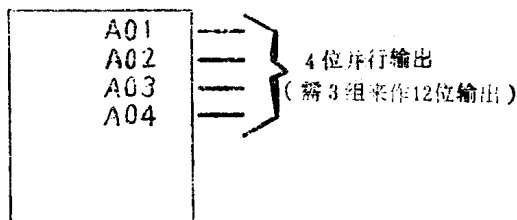


图2.6 4位并行输出的管脚布局

寻址技术

用于微处理机系统的寻址技术有好几种。直接寻址是最普通的一种寻址〔图2.7(a)〕。直接寻址时，指令字的地址部分会告知微处理机数据位于什么地方。相对寻址是另一种寻址方法〔图2.7(b)〕。寻址时，是将某些数据地址位保持于指令中，余下的则保持在某一处理机寄存器中。这两种寻址方法——直接寻址和相对寻址可叫做“平移法”。

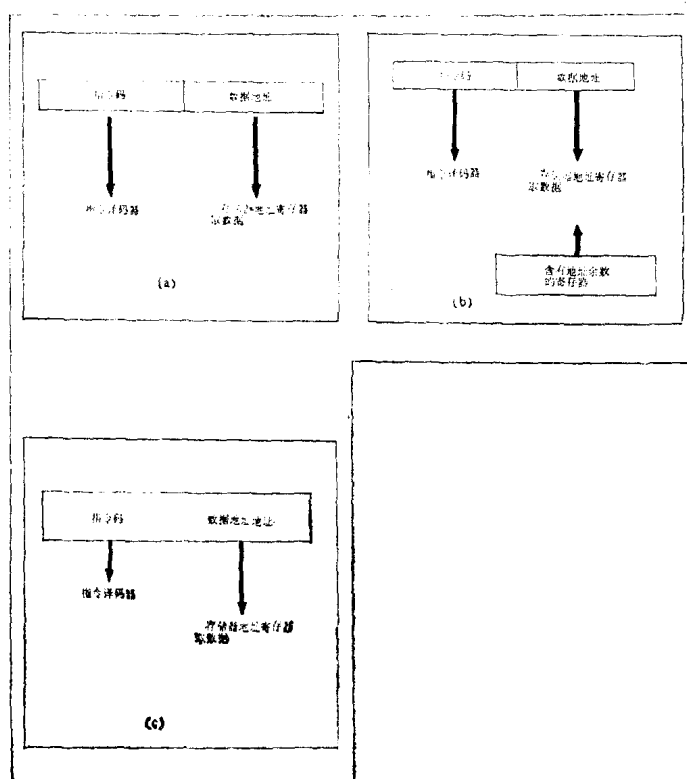


图2.7 寻址技术

除了这些“平移法”以外，还有一类，即间接寻址技术。间接寻址时，指令的地址部分含有存储器中的数据地址的定位(图2.7(c))。

间接寻址——确定CPU寄存器

在微处理器中，通过指令中的地址来确定一只给定的CPU寄存器，这是很普遍的。而这个CPU寄存器则含有数据地址，(图2.8)。

通用于微处理器的一种间接寻址形式是子程序调用程序。它可用于中断和复杂的运算程序中，如多次转移到存储器中存储有子程序所需的微程序的地方。

事实上，储存现存的程序计数器的内容，以及将其转移到新的子程序这样的功能可以通过若干级来完成。当发生这种情况时，通常就称为子程序嵌套，这种子程序嵌套是一种具有多方面功能的软件/硬件特性。

简单的子程序序列

图2.9示出的事件序列恐怕是最简单的子程序序列。

堆栈寄存器

用来执行能保存程序计数器内容的指令的一种普通方法是采用一只寄存器作为临时存储器。这叫做堆栈存储器。顾名思义，这是一堆寄存器，如图2.10所示。这通常是后进先出堆

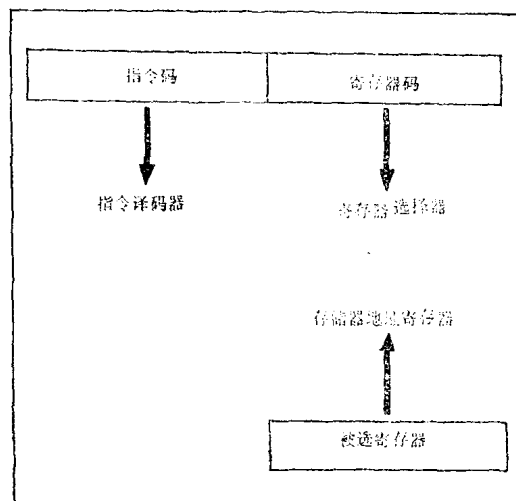


图2.8 间接寻址确定CPU寄存器

子程序调用

事件

A. 把程序计数器现有的内容存储起来

B. 在程序计数器中放置子程序的起始地址

子程序执行

事件

增量程序计数器按顺序通过子程序指令

子程序返回

事件

将原来程序计数器内容重新存储于程序计数器中，并继续通过主程序

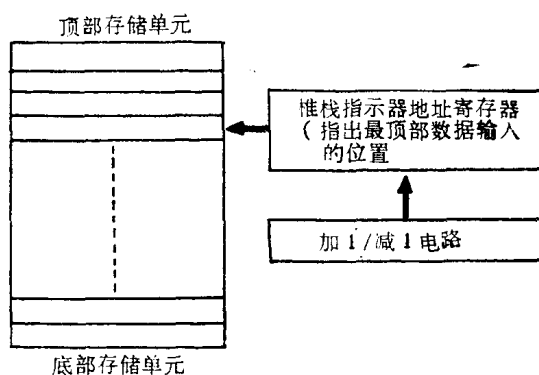


图2.9 简单的子程序序列

图2.10 堆栈寄存器

栈寄存器或后进先出存储器。对这种堆栈进行存取一般是通过称为堆栈指示器的地址寄存器来进行的，这个指示器能指出该堆栈存储器顶端的现时状态。

堆栈本身可以是置在CPU芯片上的一堆寄存器，或者可以是专门用作堆栈存储器的RAM中的某个存储块。

现在让我们看一看某些微处理机的结构。

Burroughs公司Mini-D微处理机结构

这里讨论的是目前流行的某些有代表性的微处理机，并对它们的特殊结构进行分析。Burroughs公司Mini-D微处理机的芯片是属复杂程度较低一种，它具有相当特殊的结构，如图2.11所示。最显著的特点是，程序被存储于芯片上的一只12位×256字ROM中；另外的特点是，在整个单一位的总线上都会发生数据输入/输出流程。它的运标口是个单一串行部件。对累加器，数据存储器和程序地址来说，基本的寄存器字长为8位。

由于它具有ROM程序存储器，芯片功能是在制造期间由ROM掩模的程序设计来确定的。这种ROM程序存储器限制了这种结构在大量的专门用途中的应用。

这种芯片不适用于需要高输入/输出率或高处理机速度的应用场合。但是，在输入/输出和运算速度不太重要的大量应用中，它则是一种低成本的理想芯片。

Intel 8008和8080微处理机

为了获得具有较强通用能力的芯片，就必须配备外部程序存储器、中断控制、直接存储器存取能力、和并行数据输入/输出。具有这些能力的几种微处理机在市场上现已有出售，

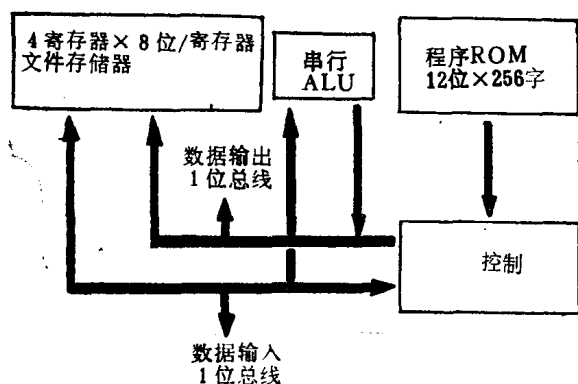


图2.11 Burroughs公司Mini-D微处理机结构

	8008	8080
ALU 字长	8位	8位
地址总线 字长	8位	16位
堆栈	7个 内部寄存器	外部 RAM
BCD 运算	无	有
I/O	中断	中断与直接 存储器存取

图2.12 Intel 8008和8080微处理机

其中两种是Intel 8008和8080微处理机。

这两种芯片具有许多共同点，但它们之间存在着很大的差异，见图2.12。

8008和8080基本上都能构成一个可解决各种应用问题的微型计算机，因为它们速度快，输入/输出能力良好，并且有一组可扩充的指令系统。

这些能力究竟如何完成的呢？下面让我们看一下8080的结构。

Intel 8080的基本结构

图2.13示出Intel 8080的基本结构。注意在8080的运算器部分中，具有采用二-十进制或二进制运算的选择。另外，运算器有快速先行控制进位加法。输入/输出是通过一只缓冲锁存器来控制的。输入/输出的目的地是累加器、一只用来寄存数据的暂时寄存器或中间结果寄存器、或用来寄存指令的指令寄存器。中间结果存储器是由若干个8位子寄存器组合而成的三只16位寄存器所构成。每只子寄存器可分别被选择。在这个结构中还有一只16位程序计数器和一只具有有关增量和减量电路的16位堆栈指示器。存储器地址通过16只管脚同时发送。

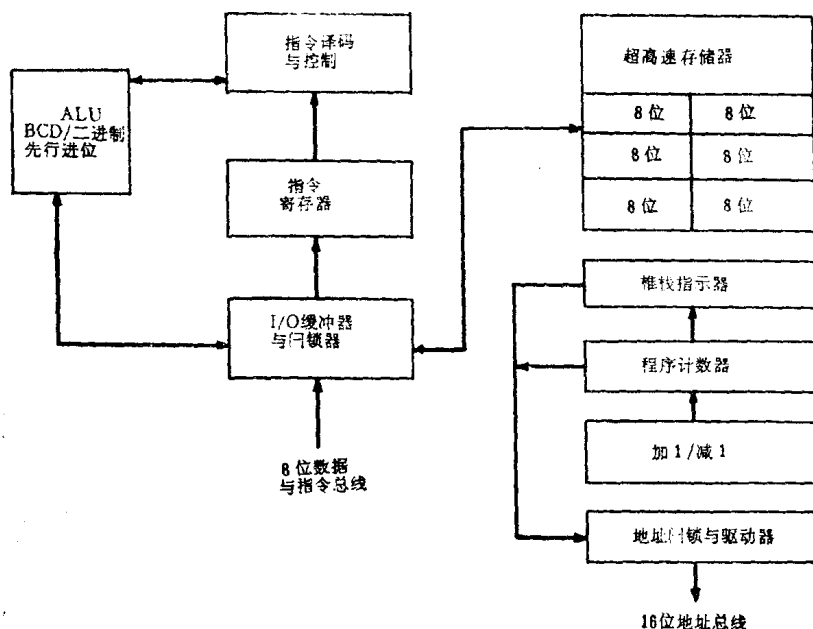


图2.13 Intel 8080的基本结构

Intel 8008的基本地址、指令和数据

为了获得全部数据和地址并行能力，8080被封装在一只具有40根管脚的管壳内。而8008则封装在一只具有18根管脚的管壳内。18根管脚的8008的主要局限性在于，地址、指令和数据全部都得通过同一个8根管脚的输入/输出总线，如图2.14所示。因此，8008比8080在速度上要慢得多。同时，由于这种布局提供一个内部堆栈，8008的子程序嵌套仅限于7级。

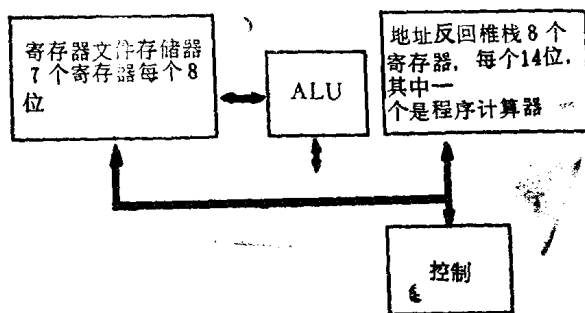


图2.14 Intel 8008的基本地址、指令和数据

这两种Intel器件是良好的通用器件。在专用系统方面它们的唯一不足之处是，恐怕用户得购买的微处理机比他们所需的来得多。

Intel芯片的另一特点是，它们是芯片系列的一部分，能通过较为简单的方法而互连成操作系统。这是通用芯片的共同特点，至于象Mini-D这类专用芯片通常就没有这个特点。

Motorola 6800 CPU的基本结构

能构成一个完整的芯片系列的一种更诱人的芯片组合是Motorola器件，即M6800微处理机系列。这个系列由微处理机、只读存储器、随机存取存储器、外围接口转接器，以及异步通信接口转接器所组成。6800CPU结构与8080完全相似，只有一点是例外：即6800CPU芯片上有一只变址寄存器，而8080没有，如图2.15所示。6800具有直接存储器存取能力和两只累加器。

6800芯片组的外围接口转接器（MC6820）简化了输入/输出问题，因为转接器能在处理机与16个其他的外围设备和/或微处理机之间提供一个双向的8位数据传输器件。因此，由于接口转接器之故，对在通信系统和微型计算机网络中使用6800就会带来很大的方便。利用6800一个部分的异步通信分隔式接口转接器通过一只解调器（MODEM）解决了跟微型计算机和电话线连接有关的一些问题。

就前面所讨论的情况而言，看来有两种截然不同的芯片结构：一种是能力有限的专用微处理机，如具有内部ROM程序存储的Mini-D微处理机。另一种是通用8位微处理机，如8008、8080和Motorola 6800。后面将进一步谈谈16位或16以上的微处理机芯片，它具有更快的速度和更佳的精度。最通用的字长是16位，但16位并不是极限。

另一方法是仍然采用8位芯片，但要改进这种芯片的通用性。如前所述的将8008改革成8080就是采用这种方法的一个实例。

RCA COSMAC微处理机的基本结构

8位COSMAC是具有6800变址特点的一种微处理机，如图2.16所示。它还提供了如下的选择，即可以把内部堆栈用作地址堆栈，或把内部寄存器用作外部RAM堆栈的堆栈指示器。但是，采用COSMAC时，程序员可以存取文件存储器中的任何一个16位寄存器，并可将数据从这一寄存器传输到另一寄存器，另外还可指定它们中的任一个作为程序计算器，堆栈指示器，变址寄存器等。这种灵活性提供了程序设计的多样化。

此外，由于文件寄存器是16位字长，所以在寄存器-间接寻址方式中可被用来存储数据的地址。注意，对寄存器是程序计数器

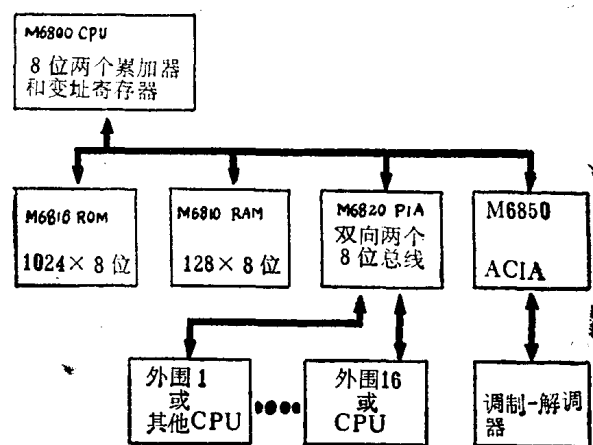


图2.15 Motorola 6800基本结构

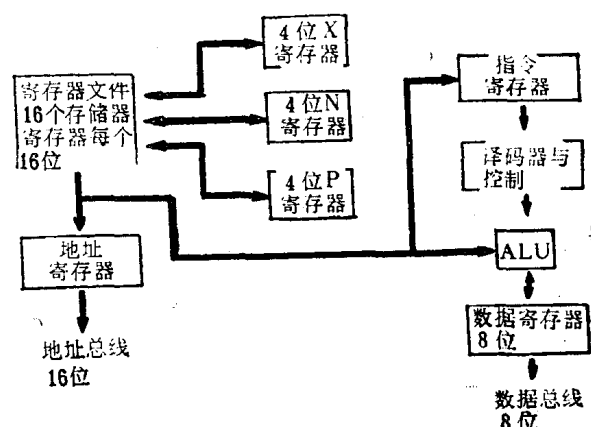


图2.16 RCA公司COSMAC微处理机的基本结构

的调用仍以4位代码方式被保存在P寄存器中。同样,在N和X4位寄存器中的代码能表示出16位寄存器中哪些是数据位置,哪一个为变址寄存器。

Rockwell PPS-8 CPU的基本结构和芯片组

Rockwell公司PPS-8中央处理机和芯片组的组合提供了高的输入/输出率,见图2.17。芯片组中的一块芯片是直接存储器存取控制器。这块芯片控制着直接存储器存取,其办法是将存储器地址和接口信号提供给输入/输出控制器。对使中央处理机跟系统级功能件相接口来说,该芯片组恐怕是最完整的有关辅助芯片了。

Fairchild公司PPS-25微处理机的基本结构

图2.18示出的Fairchild公司PPS-25是一种字长较长的微处理机。这个微处理机的实际精度是25位。字长的实现是这样的:采用一只4位加法器,接着使数据以4位字节传送给CPU,然后将结果累加在A寄存器中。但是,PPS-25不仅是一种25位处理机,而且其主要程序ROM也置于CPU芯片外部。因此,编程序和微处理机的制造是分开进行的。

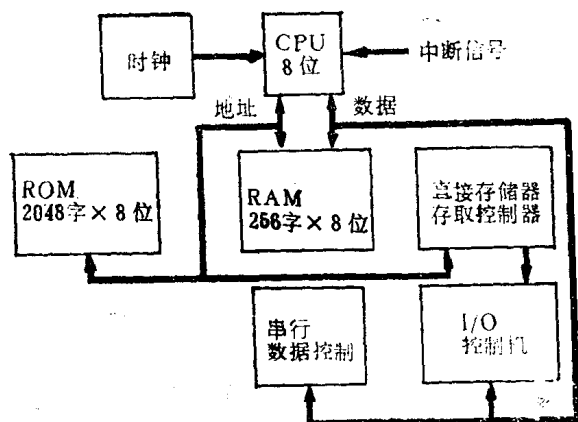


图2.17 Rockwell公司PPS-8 CPU和芯片组的基本结构

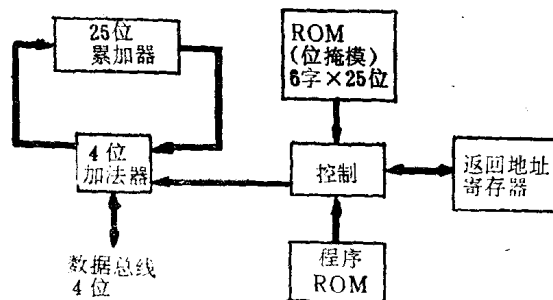


图2.18 Fairchild公司PPS-25微处理机的基本结构

一般来说,PPS-25微处理机适用于计算精度要求较高的应用场合,如科学计算。

提高计算精度的另一方法,是采用由并行结构形成的16位微处理机,如National公司的IMP-16, Raytheon公司的RP-16和Texas仪表公司的SBPO400微处理机。要获得一个给定精度的话,这些微处理机比起采用4或8位微处理机来要快得多。此外,这些微处理机能提供并行寻址,堆栈能力,和真正的通用机的输入/输出装置。

在这些位数较长的微处理机中,有着将微控制ROM置在芯片上的趋向,以获得较高级的指令,如乘和除。

National公司IMP-16微处理机的基本结构

National公司IMP-16微处理机的基本结构示于图2.19。具有地址控制的控制ROM起着有点类似于指令寄存器和具有100个25位控制字芯片的译码器的作用。另外,还有一只运算器,它具有16位容量和为子程序调用、中断和直接存储器存取所用的16个字16位的下推堆栈。

具有7个通用可寻址存储寄存器,其中4个可作累加器、还有两个累加器可用作变址寄存器。另外三个16位通用寄存器可用于主程序的程序计数和地址的存储。

除了芯片上具有微程序ROM之外,这种结构与先进的8位微处理机特别相似。显然,这种在芯片上具有控制ROM的优点在于,通常需要的微处理,如乘、除就不必再去精心设

计和存储在主程序中了。因此，复杂的微处理可以通过单指令来调出象加、减移位或交换单时钟操作那样，很容易地就可调出。

Texas Instruments 公司 SBP 0400微处理机的基本结构

现在让我们以 TI SBP0400微处理机的分析为例来归纳一下有关芯片结构的讨论。这是一种4位微处理机，但是它能组合联结成以4位为增量的任何位字长的处理机。它的指令系统与很流行的运算部件SN74181特别相似。但是，与某些需要软件实现的微处理机相比，要把0400微处理机扩展至任何字长只不过是一个简单的芯片互连问题。图2.20示出了0400的基本结构。

指令译码由指令可编程序逻辑阵列(PLA)来完成。这个指令译码能选择512个单时钟操作中的任一个。这些时钟操作可由任一个寄存器完成，并将结果存贮于任一个寄存器内。地址通过4位总线传送。运算部件中采用先行进位，并备有进位输入和进位输出端，以级联和扩展微处理机字长。这个结构基本上与具有并行输入和输出数据总线和地址总线的系统相似。但是采用0400微处理机时也有运算选择输入。0400的工作寄存器可用作堆栈本身，或将其中一个规定为堆栈指示器，因为堆栈中的任一个寄存器可以被加1或减1。

4只SBP0400微处理机组成的16位并行微处理机

根据其扩展能力，要实现任何理想字长的微处理机，如16位并行微处理机问题不大，见图2.21。这种扩展只要将进位输入和输出连起来，并将工作寄存器的移位线连接一起就行了。这就提供了外部进位输出发生电路。然后，将指令和输入数据传送入16位系统。这不仅在逻辑上是简单措施，而且这个电路可能是目前速度最高的，因为它是I²L逻辑。

考虑这种类型的微处理机结构有两个方法。一、可把0400视作为功能块，并通过其芯片系列来组成一个系统；二、充分利用0400双极速度，并用之装配一个芯片组以复合成另一个微处理机。总而言之，0400微处理机能对微型计算机的许多问题提供简便的解决方法。

芯片结构的讨论就此结束。其中有好多芯片结构十分相似，它们在特定规格方面的差异比在基本结构方面来得明显。专用结构与通用结构之间的差异则例外。

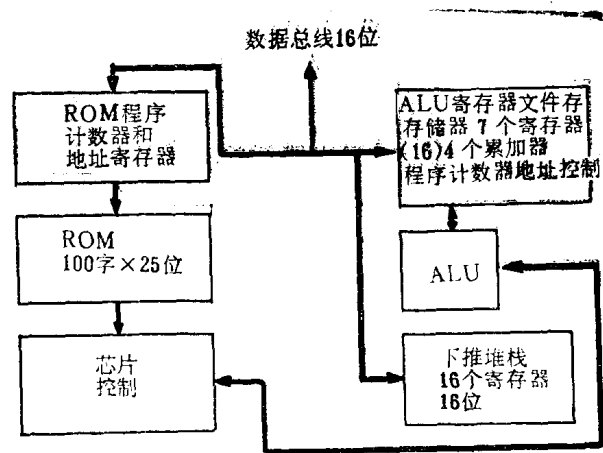


图2.19 National IMP-16微处理机的基本结构

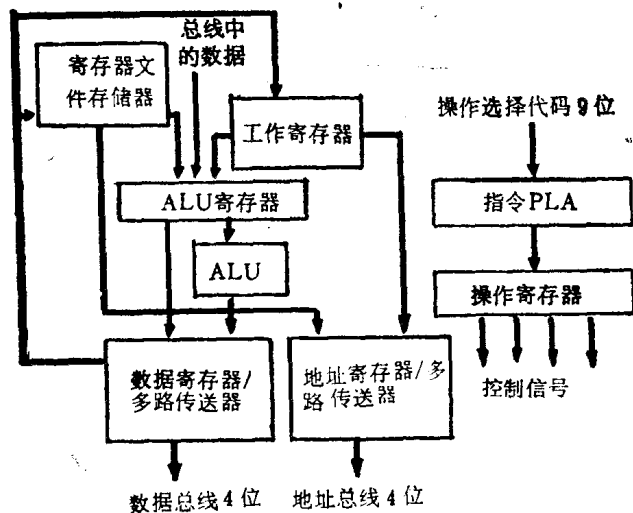


图2.20 Texas Instruments SBP0400微处理机的基本结构

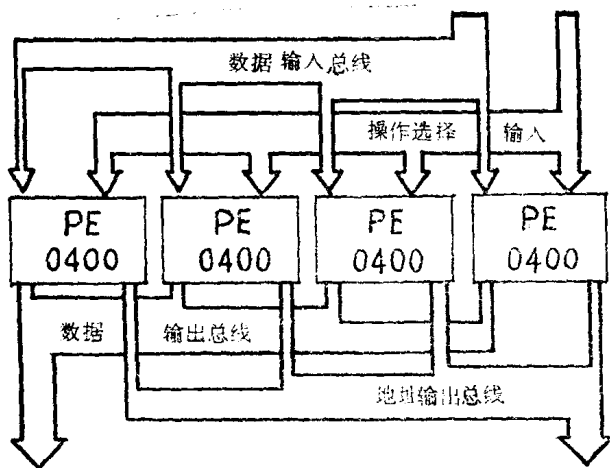


图2.21 由4只SBP 0400微处理机结合构成的16位并行微处理机

的应用是少量的，那么可以选择适当的通用芯片，构成专门的应用。

通信应用的芯片结构

如图2.23所示那样，如果象通信系统要求的高输入/输出率，那么就可选择具有直接存储器存取和良好中断结构的微处理机。最好是，厂商提供的微处理机有一套完整的芯片系列，能方便地与I/O接口配套。

通用芯片结构

如图2.24所示，如果应用要求较复杂，那么就应选择具有良好输入/输出能力，良好指令系统和良好辅助芯片组的微处理机。同时，这种微处理机应能提供微程序设计和外部堆栈能力或内部大的堆栈能力，并且应是易于编程序的一个系统。

- 要求最佳I/O处理能力
- 要求直接存储器存取
- 良好中断结构
- 最好是包括有接口芯片的芯片组

例子:

编码器
前端处理机
数据输入终端

图2.23 通信应用的芯片结构

- 芯片上编有程序的ROM
- 有限速度
- 有限I/O
- 短字长

例子:

计算器
控制机

图2.22 用于专门应用的芯片结构

专用芯片结构

如图2.22所示，对使用量大，速度和输入/输出有限的芯片来说，专用的编有程序的ROM片可能是最好的。如果所需

- 良好I/O能力
- 多样的指令组
- 微程序设计的堆栈能力
- 便于设计良好程序软件的辅助工具

例子:

POS终端
微型计算机

图2.24 通用的芯片结构

可程序的微处理机*

所需处理能力有限、费用较低的那一类应用可以采用本文辑^[1]于前面文章所论述的那种固定指令组微处理机。但是,比起这一类简单的应用来说,大多数应用则要求更快的运算速度、更灵活的输入/输出、以及经过更加仔细剪辑的指令系统。可编微程序的微处理机,尤其是那些由位片元器件系列所构成的微处理机,提供了这些较为复杂的应用所需的更大的灵活性和解题能力。

此处,在名词术语上存有一个明显的混淆,必须立即予以澄清。微处理机系指小型计算机中的中央处理单元,而微程序设计指的是,以控制信号的顺序来实现处理系统控制部分的功能,控制信号组成字(称作微指令),并存贮在控制存贮器中^[2,3,4]。这两个听起来差不多的术语之间没有必要的联系。当然,微程序设计可以在诸如IBM之类的大型计算机上进行,或者在诸如Burroughs 1700、Hewlett-Packett 2100、Varian 73之类的小型计算机上进行。某些微处理机,如Intel 4040,既不是可由用户进行的可编微程序的微处理机,也不是已由制造厂商把微程序编制好的微处理机。大多数的微处理机,如Intel 8080或Motorola 6800,它们的微程序都是编好的,但用户不能更换控制存贮器。非可编微程序的微处理机的程序设计是普通的程序设计,而不是微程序设计。术语“微程序”(根据控制级而编写的一种完整的程序)和“微代码”(根据控制级而编写的一个指令或指令序列)应该保留下来,以便归属于为实现采用微程序之系统的控制部分而编写的那些程序。

除此混淆之外,我们从微程序设计中还能得到些什么呢?答案是速度与灵活性。可编微程序处理机可以把微程序编制得能直接执行指令,否则的话,这些指令在软件中执行的速度就要缓慢得多。这类指令的典型例子是乘、除、浮点运算、三角函数、文件研究和文本分析。一个单一的可编微程序的处理机,只要改变一下微程序,就能重新配置以完成各种不同的任务。这样,微程序设计就为硬件的控制级提供了软件的灵活性。

当然,微程序设计也有其缺点。由于涉及的细节太多,所以微程序设计是相当困难的。由于现有的软件或硬件的辅助系统很少,故大多数的微程序必须用汇编语言或机器语言写成,并用人工调试。此外,微程序经过专门设计的处理机无法使用不是专门为它编写的软件。为经过专门微程序设计的处理机而编写的程序也不能在任何其它的机器上使用。

显然,在对处理机进行微程序设计时,必须兼顾到灵活性与方便性,以便能执行现有合适计算机的某些指令或全部指令——即对此计算机进行仿真。经过微程序设计的仿真器将能采用专为仿真计算机所编写的软件。通过省去某些特点和增添其它一些特点,恐怕会使采用微程序设计的处理机(比起原来的计算机来)更适于某一特定的应用。

可编微程序的微处理机已能够化相当低的成本而利用微程序设计的灵活性。这一类微处理机中有些是完整的功能单元,只须再加一只控制存贮器。DEC公司用来生产LSI-11型微型计算机时所采用的Western数字公司的MPS-1600型微处理机就是一个典型的例子。但

Lance A. Leventhal, "Microprogrammable Microprocessors", Simulation, Vol.26, No.6, April 1976, pp.193-

197. 汤杰译, 孙绍凤校

是，多数可编微程序的微处理机通过把中央处理机分成几块处理口片及有关的组件，能带来新的灵活性。这些处理口片（亦称作位片）由寄存器的对称部分和中央处理机运算的部分所组成。其复杂性介于多数先进计算机所采用的中规模集成电路与微型计算机所采用的单片中央处理机的集成电路之间。用户可以用相当少的这类组件构制一个为专门用途而设计的解算口片〔5〕。

目前推广这些中间器件的原因之一是因为双极工艺的电路片复杂性有限。这类高速工艺不可能生产出单片微处理机。因此，要达到较高的速度，就需要对处理机的功能作某些划分。事实上，国家半导体公司在几年以前就已经在推广IMP系列的MOS位片。但是，这些器件速度较低，使得它们无法适用于其微程序设计能够得到合理调整的大多数应用。对多数能够容忍速度慢的用户来说，他们所称心的是单片微处理机的简单性。

通常，位片本身只不过是计算机结构中的一个部分，图1示出了一个典型的位片结构。

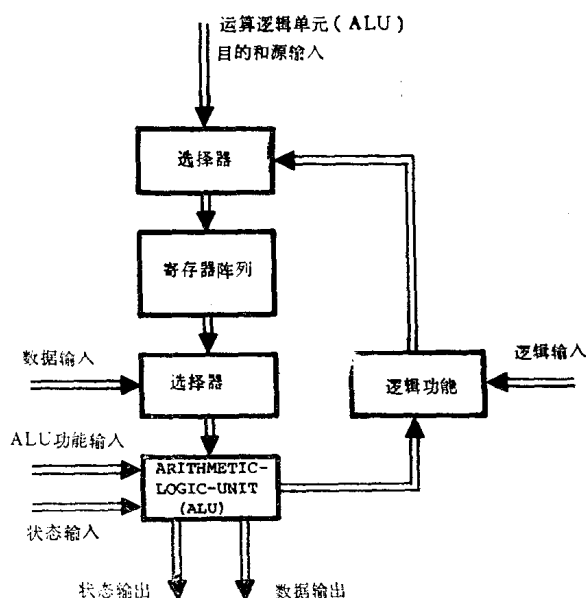


图1 典型的位片微处理机

这种微处理机包括一个能够完成运算和逻辑功能的运算—逻辑部件、一个能存贮数据的寄存器阵列，以及若干只选择器。这些选择器能够选择用来存贮操作数和结果的寄存器和其它源，以及选择在把结果存贮起来以前所要完成的逻辑操作。需要外部输入，以便提供数据并确定所要执行的特定功能。

表1 双极型微处理机

处理机型号	片宽度	采用工艺	周期时间(毫微秒)
Advanced Micro Devices AM2900	4	小功率肖特基TTL	100
Fairchild Macrologic	4	小功率肖特基TTL	100
Intel 3000	2	肖特基TTL	150
Monolithic Memories	4	肖特基TTL	200
Motorola	4	发射极偶合逻辑	55
Signetics 3000	2	肖特基TTL	150
Texas Instruments SAP0400	4	集成注入逻辑	1000
Texas Instruments 74S481	4	肖特基TTL	100

位片上端的选择器能够选择用什么样的寄存器来获取运算—逻辑部件所需的数据，并能存贮结果。下端的选择器能够选择在运算—逻辑部件中（ALU）将采用的是寄存器阵列的输出、还是数据输入、或者还是0。逻辑功能可以包括ALU输出的移位和涉及逻辑输入的逻辑操作（如逻辑“与”或“或”）。因此，表1所示出这类双极型位片将包括下列部分：

- (1) 一个能够进行各种算术和逻辑功能的2位或4位运算—逻辑部件。

(2) 一个具有八个至十六个 2 位或 4 位寄存器的寄存器结构；这些寄存器中的某一些寄存器可以组成一个可寻址阵列，而另一些则是分开的。

(3) 为运算—逻辑部件的操作数与结果选择源和目的、以及允许移位和其他逻辑操作所需要的多路转换器与选择器。

(4) 允许跟系统的外部器件进行各种不同的总线连接的 2 位或 4 位数据通道。

运算—逻辑部件还能接收诸如进位输入之类的状态信息，产生诸如进位输出和可以用来进行位片间信息传递的零与溢出指示符之类的状态输出。其次，这些位片可以组合起来，形成一个具有字长较长的处理单元，即八个 2 位的位片可以组成一个 16 位的处理机，或八个 4 位的位片可以组成一个 32 位的处理机。

大量的信息必须由外部的来源提供给处理机位片。所要进行的运算和逻辑操作，ALU 所要用的源与目的，输入数据总线的内容，以及输入状态信息等均必须在外部确定好。对处理机位片来说，输出总线的内容与输出状态信息也必须是由外部来控制和管理。但是，处理机位片本身将取代 50 至 200 个中规模的集成器件，这些器件目前用来执行相同的功能。所用器件数目的减少导致了成本的降低和可靠性的提高，因为元件，电路板和连接数更少了。

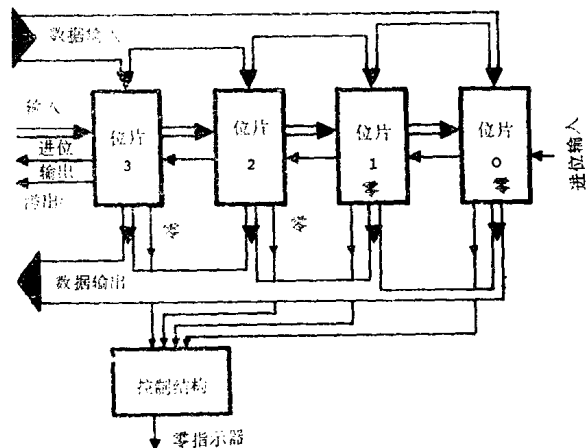


图2 处理机位片的结构阵列

处理机位片并联组成阵列，各位片所承担的功能相同，可产生输出数据的一部分。状态输入与输出允许各位片之间的通信。

处理机位片阵列的某些外部连接仅仅用来使此阵列成为能够获取数据、进行操作和传递结果的一个完整的单元。这些阵列连接示于图 2 中。对所有的位片来说，所要进行的运算操作或逻辑操作以及 ALU 所需的源与目的是一样的。输入数据总线将分成长度适当的几截，进入处理机的位片，输出数据总线将在位片的出口处重新组合。（请注意图 2 中总线尺寸大小的变化）。外部的进位输入被提供给最低有效位片，然后由一块位片向下一块位片进位，并从最高有效位片那儿获取外部进位输出。来自位片的零指示符必须组合在一选通结构中，因为只有当所有位片的结果为“0”时，阵列的总结果才为“0”。整个阵列的溢出指示符产生于最高有效位片；而其他的溢出指示符均无意义。可能会需要一些附加连接，以便使 ALU 产生的结果能在阵列的长范围内进行移位。

计算机的其他功能可以通过采用构成双极型处理机系列的有关元件而得以简化，除 Monolithic Memories 公司外，其余生产双极位片的厂家均能提供此类器件系列。当然，这些器件系列可以互相兼容（尤其是如果它们是用同一种半导体工艺制成的话），而且也可用标准元件。各种元器件系列中的典型产品包括有：

(1) 序列发生器，能控制执行微指令的顺序。这种序列发生器甚至可以在无处理机位片的场合下，用来控制一个采用微程序设计的系统。无运算能力并担负控制任务的采用微程序设计的系统被称为微控制器。

如图 3 所示，序列发生器能够允许有若干个地址源，以便确定下一步执行哪一个微指令。新地址可以取自微程序计数器，以便按照顺序执行微指令，也可取自外部地址源或微子程序返回栈，以便能够使用微子程序。其他的选择可包括现行微指令的重复，采用逻辑输入

而由逻辑操作来导出一个地址，或用一个清除或“0”输入来强制产生一个“0”地址。选址输入将确定应作出哪一种选择，以及确定返回地址是否能保存在微子程序返回栈中。序列发生器还可以处理状态信息，如外部进位输入与输出，以及“0”与溢出标记。

(2) 双极只读存储器 (ROM) 与可编程程序只读存储器 (PROM)，这些存储器能保存微子程序的最终型式。这类器件所见的典型存取时间为60至100毫微秒 (而多数主存储器的存取时间约为1微秒)。

(3) 高速寄存器，可以用来存贮指令、地址、数据、状态、或其它的信息。在获取新指令或新地址的同时，此类寄存器可以用来存贮老的指令或地址，从而能够进行“流水线”操作或使指令周期的若干部分重叠。此类寄存器还可具备专门的特征，从而使其能够充当选通缓冲器、高速锁存器、多路转换器、总线驱动器与接口，或中断输入/输出口。

(4) 总线驱动器，能够用来使处理机跟外部总线相接口。此类驱动器可以是双向的，能为较长的线路提供缓冲和足够的驱动电流。可以采用好几种格式，来处理不同类型的总线。可以包括进奇偶发生与检验。

(5) 中断控制装置，能确定中断的优先，保持中断状态，并能为每一个中断提供一个识别码 (有时称作向量)。这类器件通常组合起来可以具有编码器、优先识别电路与状态寄存器的功能。它们甚至还能可编程序，从而获得更大的灵活性。

(6) 直接存取存取 (DMA) 控制装置，跟中断控制装置处理中断的情况极其相似，它们能处理DMA请求。这类装置还能够为DMA操作提供定时和控制。

(7) 先行进位发生器，能够使几个处理机位片相连接，而无需由位片阵列的一端传送到另一端这样的缓慢的行波进位。

(8) 双极随机存取存储器，可以作为外加的寄存器或高速的局部存储器。此类读/写存储器所具有的存取时间通常小于100毫微秒。它们在研制的初期被用作微程序存储器，特别是借助于被称作为只读存储模拟器这样的专用研制工具^[6,7]。

要制成一个能工作的计算机系统，除处理机位片及其相应系列中所含的器件外，还得需要另一些元器件。图4示出了这样的一种计算机系统。需要有一些标准的中规模集成器件，以便提供一个系统时钟，完成某些条件功能，充当选择器或多路转换器，以及处理复位之类的系统管理任务。一个完整的计算机系统可以有100至200块电路片，在这些电路片中只有20至30片为处理机位片或其他的系列元件。例如，在Carnegie-Mellon大学所作的一次对PDP11/40型计算机的模拟，用了132块芯片，其中，只有10块芯片是大规模集成器件^[8]。

这种完整的计算机系统由一个进行运算与逻辑操作，并将暂时结果贮存在内部寄存器中的处理机位片阵列，一个含有微指令的微程序存储器 (通常为只读存储器)，一个能对执行

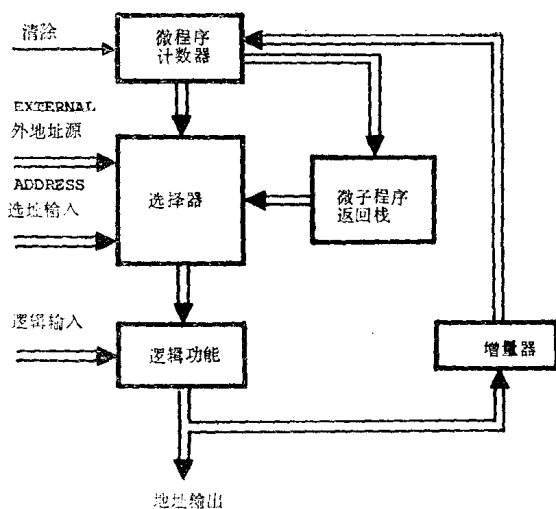


图3 典型的序列发生器

此序列发生器对要执行的下一个微指令的地址允许有几种选择。新地址可以取自微程序计数器，可以取自外部地址源，或取自微子程序返回栈。选址输入能确定采取哪一种选择，现行的微程序计数是否能保存在返回栈中。

微指令顺序进行控制的序列发生器, 以及能提供所需的时钟脉冲、外部总线接口状态和信息的分立逻辑所组成。

目前, 处理机位片已大量采用, 其目的有两个:

(1) 对现行的计算机设计进行仿真, 用这种方法进行仿真的最简单的计算机是 Data General 公司的 Nova 机。Nova 机的仿真方法目前可从 Monolithic Memories 公司与 Digital Computer Controls 公司那儿获得。采用处理机位片来进行仿真的计算机设计有 Intel 公司的 8080 和 DEC 公司的 PDP-11。

(2) 设计高速的控制机。诸如磁盘, 视频显示器和过程控制系统这样的装置所要求的高速度比起采用固定指令组的微处理机所能获得的速度要来得快。休斯飞机制造公司已经采用由位片设计成的微型计算机来作为信息转换装置、自动导航仪、以及扫描转换器的控制机^[9]。

最有前途的应用将是为之能够研制出一种专用的解算元件的那些应用。这种解算元件将是能够编微程序的, 从而能以高速度来执行单一的专项任务。Grumman 宇航公司已制成一种专用装置, 可用来模拟空战演习; 据该公司报道把几台这样的装置并联起来能够在 2 至 20 个小时内解决原先在大型计算机上要化几百个小时才能解决的问题。这种专用系统的硬件成本约为 80,000 美元左右^[12,13]。看起来能借助于采用处理机位片的微程序设计方法的其它任务包括信号处理、图象显示、分类、文件检索以及数据的编排和检查。设计用来完成这些任务的装置基本上还是“未知框”, 这些“未知框”将用硬件取代软件。

可编程程序的微处理机为解算元件的设计开辟了新的领域。可以创造出专用的元件, 以完成高速的任务。当然, 今后的研制将导致系统的价格更便宜, 而且更易于设计。可编程程序的微处理机的灵活性将导致更便宜、更简单的计算。

潜在的应用:

- 灵活的终端设备 •

微处理机能够使普通的终端设备变成能够局部地执行多项任务的灵活的终端设备。为此目的, 某些厂家甚至还将提供计算机的整套另件。

- 混合接口 •

微处理机可以作为模一数系统数字部分的预先处理机。微处理机可以对输入和输出进行格式编排, 进行检误和定标, 而且通常能够减轻数字式计算机的负担。一台廉价的微处理机通过把数字计算机从琐细而又费时的任务中解脱出来, 从而大大地提高了数字计算机的解题能力。

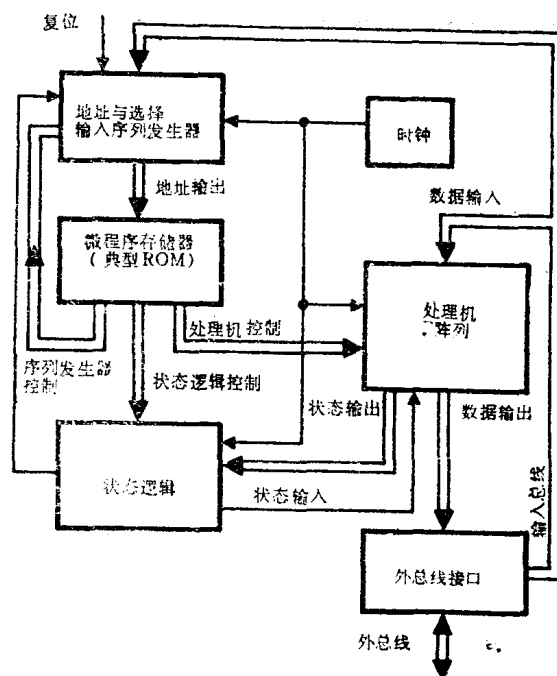


图4 用处理机位片构成的计算机系统

参 考 文 献

- (1) LEVENTHAL, L. A. "Fixed-Instruction-Set Microprocessors", Simulation, May 1976, pp. 162-165.
- (2) HUSSON, S. S. "Microprogramming: Principles and Practices", Prentice-Hall Cliffs, New Jersey, 1970.
- (3) AGRAWALA, A. K., RAUSCHER, L. G. "Microprogramming: Perspective and Status", IEEE Transactions on Computers, August 1974, pp. 817-837.
- (4) FLYNN, M. J. "Microprogramming—Another Look at Internal Computer Control", IEEE Proceedings, November 1975, pp. 1554-1567.
- (5) MYLAND, D. C. "How to Design Your Own Microprocessor", Electronic Design, September 27, 1975, pp. 72-78.
- (6) BLAKESLEE, L. R. "Digital Design with Standard MSI and LSI", Wiley, New York, 1975.
- (7) DAVIDOW, W. H., SLAYMAKER, H. W. "Debugging Microprogrammed Systems", Scientific Micro Systems Application Note Mountain View, California, July 1973.
- (8) SCRUPSKI, S. E. "High-Density Bipolars Spur Advances in Computer Design", Electronics, October 30 1975, pp. 81-86.
- (9) VAHEY, M. "Bit Sliced Architectures for Microcomputers", Paper Presented at Microprocessors III Conference, Fullerton, California, 1976.
- (10) MYLAND, D. C. "Increase Microcomputer Efficiency", Electronic Design, November 8, 1975, pp. 70-75.
- (11) ALTMAN, L. "The New LSI", Electronics, July 10, 1975, pp. 81-92.
- (12) SCRUPSKI, S. E. "Coming: Cheap, powerful Computers", Electronics, December 11, 1975, pp. 67-68.
- (13) "Micros Can Beat the IBM 370/168", Electronics, February 5, 1976, pp. 41-42.
- (14) CUSHMAN, R. H. "Second Annual Microprocessor Directory", 1975, pp. 29-42.
- (15) BORGERSON, B. R. "The Viability of Multimicroprocessor Systems", Computer, January 1976, pp. 26-30