微型计算机

——Z-80

WEIXING JISUANJI

科学技术文献出版社重庆分社

微型计算机——Z-80

中国科学技术情报研究所重庆分所　　编　辑
科学技术文献出版社重庆分社　　出　版
重庆市市中区胜利路91号
四川省新华书店重庆发行所　　发　行
重　庆　印　制　第　一　厂　　印　刷

开本：787×1092毫米 1/16　　印张：11½　　字数：30万
1979年12月第一版　　1979年12月第一次印刷
科技新书目：137—80　　印数：19,000册

书号：15176·445　　定价：1.20 元

目　　录

微型计算机Z80简介

Z80微处理器囊括了8080处理器的全部处理功能，又额外增加了八十条指令。为了使片子数保持最少，8080系统所需的许多外围电路，移到了Z80片上。Z80系统中的各个器件，均用n沟硅栅耗尽负载工艺制造，由4兆赫单相时钟驱动，用单一的5伏电源，且具有TTL兼容的输入和输出。

全套电路包括Z80中央处理器和下列外围电路：计数器-定时器电路（CTC）、并行输入/输出电路（PIO）、直接存储存取控制器（DMA）、串行输入/输出电路(SIO)和一组辅助扦件板（表1）。这些电路都有2.5和4兆赫两种形式，用陶瓷封装，有较宽的温度范围。除计数器-定时器为28脚的双列直扦式封装外，其余电路全用40脚的双列直插式封装。

为达到优先权中断控制，所有的外围电路都可以按链状电路连接。因为系统工作所需的外围电路，大部分已归入Z80片中，所以由Z80、系统时钟和上电复位电路，加上任何必要的存储器和外围电路就可组成最小系统（见图1）。在系统一级上Z80微处理器不用任何额外的硬件就能保证向量优先权中断机构。

Z80的接口简单

虽然Z80在定时和控制信号方面和8080是兼容的，但是脚的分配是不同的。全部输出脚可在0.4伏下吸收1.8毫安电流，这相当于一个标准的TTL负载。

从Z80片出来的三条总线——十六位地址总线、八位双向数据总线和十三根线的控制总线，共占去Z80的四十个脚中的三十七个。余下的三个用于电源、地和单相时钟（图2）。和8080不同的是，Z80不需要状态锁存器或时钟，且中断向量处理和动态存储器再生，完全由微处理器片内的逻辑实现。

表1　　Z80系统的器件

器件名称	说　　明	以一百个发售的单价**（美元）
Z80-CPU	8位中央处理器，2.5兆赫*	26.50
Z80-CTC	计数器/定时器，2.5兆赫*	17.00
Z80-PIO	并行I/O，2.5兆赫*	11.00
Z80-DMA	直接存储器存取，2.5兆赫*	38.00
Z80-SIO	串行I/O，2.5兆赫*	
	辅助插件板	单价（美元）
MCB	微型计算机插件板	
	——全套未装	435
	——已装好	495
MDC	存储器/软盘控制器	795
RMB	RAM存储器板	750
IOB	输入/输出板	350
PMB	PROM/ROM存储器板	395
PPB/EPROM	EPROM写入器（用于2708）	475
PPB/PROM	PROM写入器（用于7620，7640）	475
CPB/ROM	组合写入器	575
VDB	视频显示板	475

* 表示该器件有4兆赫产品

** 额定温度范围0－70°C，塑料封装

十三根控制线实际上分为三条控制总线：系统控制（六根）、微处理器控制（五根）和微处理器总线控制（二根）。一根总线控制线起着总线请求线（BUSRQ）的作用，由这根线输入的信号，不仅要求微处理

JS126/18

器的地址和数据总线变成高阻抗状态，而且也要求系统控制总线中的存储器请求、I/O请求、读数据和写数据等控制线变为高阻抗状态，使其它器件能利用总线。另一根总线控制线用于输出称为总线响应（$\overline{BUSAK}$）的输出信号，它变为高电平时指示上述各线进入高阻抗的第三态。

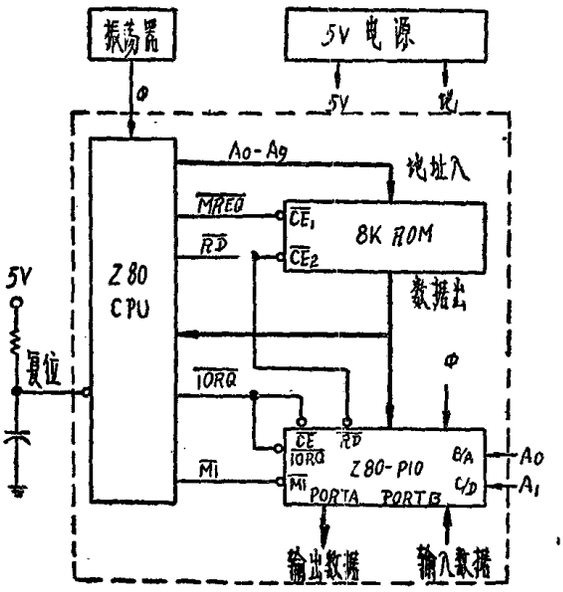


图1　最小Z80系统，可由微处理器、振荡器、某些存储器和PIO这样的I/O口组成，只要再加上电源和复位电路。

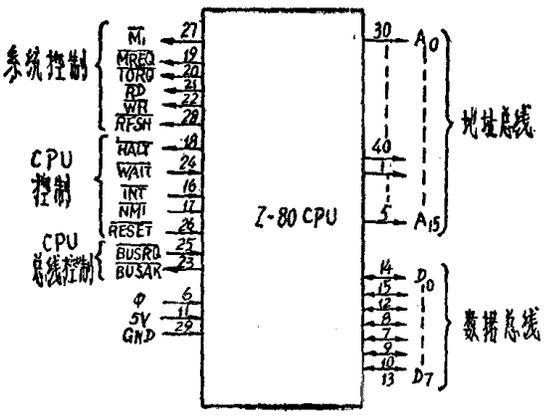


图2　Z80片上的三条主要总线，即地址总线、数据总线和控制总线。控制总线又可分成三条更小的总线，分别用于系统控制、处理器控制和总线控制。

六个系统控制信号都是从Z80输出。$\overline{M_1}$线上（机器周期1）电平之变低指明Z80处于指令的取操作码阶段。存储器请求线（$\overline{MREQ}$）在地址总线上保存着存储器读或写操作需要的有效地址时变低。I/O请求线（$\overline{IORQ}$）变低则指明地址总线的低位字节保存着I/O读、写操作需要的有效I/O口地址。

存储器读和存储器写二根线（$\overline{RD}$和$\overline{WR}$）也是低为有效。$\overline{RD}$指明微处理器要从存储器或I/O设备读数据，而$\overline{WR}$则指明数据总线保存着数据，待存入所寻址的地点。当第六根系统控制线即再生信号（$\overline{RFSH}$）变低时，它指明地址总线较低七位含有动态存储器的再生地址，因而现在的$\overline{MREQ}$信号应该用作全部动态存储器的再生读出。

五根微处理器控制线中，有四根是输入线，一根是输出线。它们全是低为有效。唯一的输出线是暂停线（$\overline{HALT}$），它指明Z80何时执行完暂停指令和在等待不可屏蔽或可屏蔽中断。在暂停期间，微处理器自动执行无操作指令，以保持动态存储器再生。等待输入（WAIT）给微处理器指明，所寻址的存储器或I/O设备尚未作好数据传送的准备（此时微处理器将进入等待状态，直到此控制线为低）。这根控制线使任何速度的存储器或外部设备都能和Z80同步。

为在接通电源之后使微处理器复位或置于初始状态，$\overline{RESET}$线可以拉到低电位。此时将迫使Z80的程序计数器置成00_{16}，封锁中断开放触发器，把I寄存器置为00_{16}，把R寄存器置为00_{16}，和把中断置成0。

最后二根控制线是中断请求（$\overline{INT}$）输入和不可屏蔽中断（$\overline{NMI}$）输入。当$\overline{INT}$线为低时，如果由软件控制的中断开放触发器（IFF）处于开放状态和如果$\overline{BUSRQ}$线为高，则将使微处理器在执行完现指令之后中断。每当微处理器接受中断时，在下一个指令周期的开始应发出响应信号（在M_1时发$\overline{IORQ}$）。

$\overline{NMI}$线是由负沿触发的输入，比$\overline{INT}$线具有较高的优先权。不论IFF的状态如何，这个输入在现指令执行完时总得到响应。此时它将迫使Z80把程序计数器的内容存入外

部堆栈，然后从0066_{16}存储单元开始执行。

中断和状态标志增加了灵活性

有三种中断方式可供程序员选择。方式0容许中断设备把任何指令扦到数据总线上和让微处理器执行。方式1使微处理器自动转移到固定地址0038_{16}开始执行，无需借助外部硬件（程序计数器的内容压入Z80内部堆栈）。

方式3是最强有力的一种中断，可以用间接寻址调用任何存储单元的内容。此时Z80将I寄存器的内容作为高位字节和中断设备提供的八位作为低位字节，组成间接地址。

二个相同的八位状态标志寄存器（F和F′）也是Z80的组成部分。每个寄存器的六个位可用于转移、调出或返回指令；它们的置位或复位由微处理器的各种操作决定。F和F′寄存器都有四位可测试的状态标志和二位不可测试的状态标志。四个可测试的位是：进位、零、负号和奇偶/溢出。

进位标志包含累加器最高位来的进位。加、减、移位和循环移位指令都可以改变其状态。如果一个操作把0装入累加器，则零标志将置位，否则将复位。当数是带符号的数时，如果操作结果为负（累加器的第七位是符号位），则负号标志置位。奇偶/溢出位有双重用途，这一位的置位，在逻辑操作时表示累加器中的操作结果是偶数，在执行带符号的对2补码运算时用于指明溢出的情况。

半进位和减标志是二个不可测试的标志。半进位是对四个最低有效位操作结果所产生的BCD（二-十进制）进位或借位。（当用DAA指令时，此标志用来修正二-十进制加或减操作的结果。）减标志则邦助识别先前的指令是加还是减操作，借以修正BCD操作；加和减的修正是不同的。

移位操作不仅可对累加器执行，而且还可以对任何一个寄存器或存储单元执行。此外，I/O操作同样如此，即不仅可以对累加器进行，也可以对任何寄存器进行。十六位的直接送数和存数可以送往BC寄存器对、DE寄存器对或IX或IY寄存器，而不是象8080那样只送往HL寄存器对。因此，交换和寄存器传送操作量大为减少。此外，应用HL寄存器对的十六位算术运算也比8080有所扩充，包括了带进位加和带借位减。

软件给Z80加添“马力”

Z80有许多独有的指令，可用于保证多字节数据块操作，这在数据通信和文本处理中是极大的方便。例如，数据块传送指令从HL寄存器对指定的存储单元取出数据，放入DE寄存器对指定的存储单元，并使HL和DE寄存器对内容加1，BC寄存器对内容减1。这里假定BC寄存器对存放操作字节数。这条指令可以一次或重复执行。HL和DE内存放的地址也可减1。

利用数据块传送指令，Z80可以5.25微秒/字节（对4兆赫时钟）传送数据。数据块操作也可以用于存储器搜索和I/O操作。移位和循环操作得到了加强。例如，通过累加器的十进制算术四位移位大大加速了BCD乘和除，位操作指令则容许对外存储器或内部寄存器的任何一位进行快速存取。

指令系统的另一个得到加强的方面是包括了十进制调整指令，这条指令在减和加操作之后工作。求补指令和循环指令也是指令系统的一部分。循环指令使B寄存器逐次减1，如果其内容不为0，则作相对转移。其它的操作示于附录2（Z80软件）。

使Z80工作

有了上述四种基本的Z80外围电路，实际上可以构成任何高性能微型计算机。例如用Z80可以组成一个过程控制系统，如图3

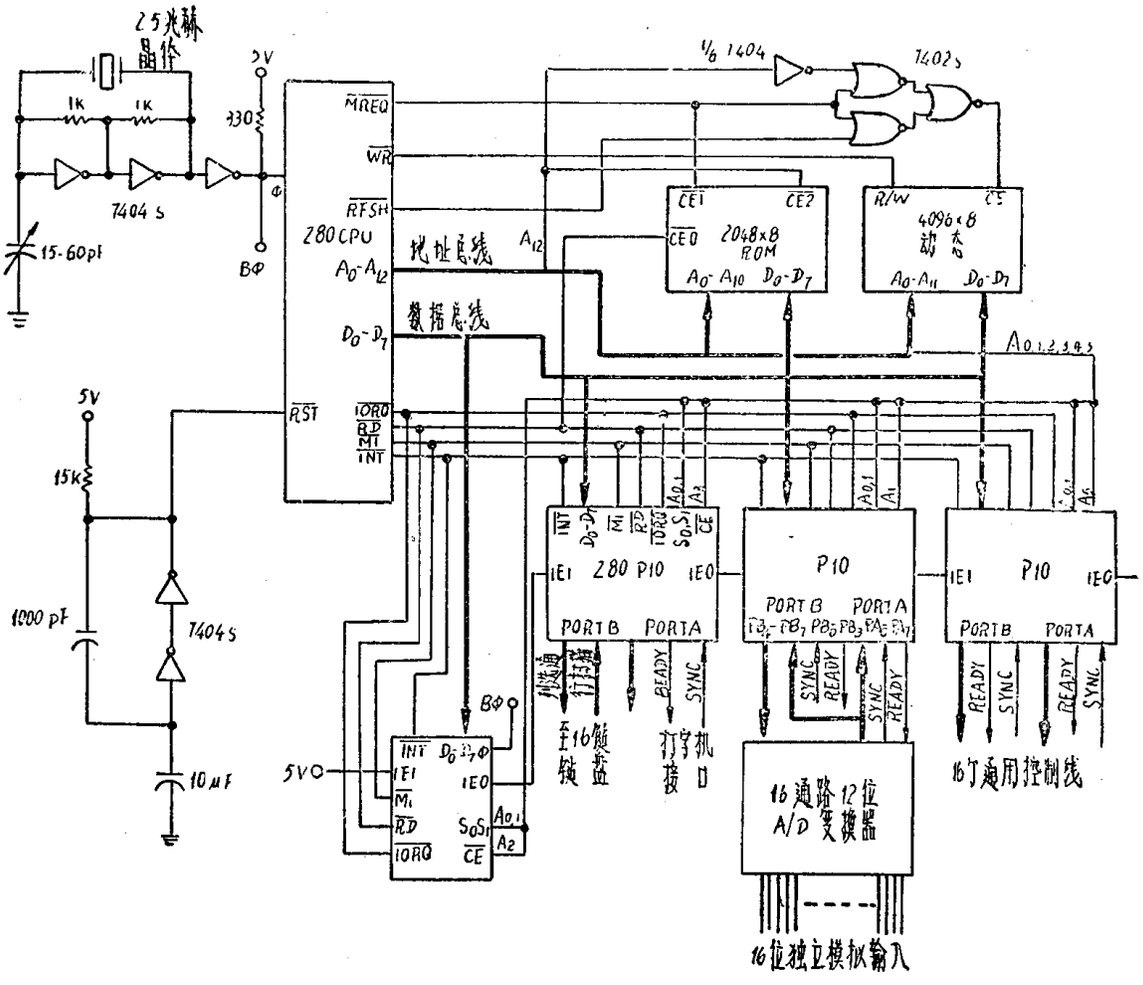


图 3　用链状中断结构的方法可以把任意数量的外围电路片加到这个以Z80为基础的过程控制系统。外围电路片越靠近Z80，其中断优先权越高。组成此数据采集子系统仅需16块集成电路，其中九块是存储器。

所示。Z80 控制器处理的外围电路包括三个并行输入/输出电路(PIO)和一个计数器/定时器。三个PIO分别处理十六键的键盘和打字机、多通道模拟/数字变换器和十六根控制线。因为外围电路片可以链接，所以只需少量或甚至不需软件或硬件开销，就可以实现优先权中断结构。申请中断的PIO利用中断方式使微处理器转向中断服务例程，在此例程执行完后，一条特殊的指令——从中断返回指令——回到PIO，并使微处理器能对优先权较低的中断服务。

全部辅助电路片都有两根链接线，一为中断开放入(IEI),另一为中断开放出(IEO)。因为计数器-定时器电路(CTC)用在控制器中使Z80免做定时循环，故软件开销减至最少。图3所示的控制器需要十四块集成电路，其中九块是存储器（2048字节 ROM 和4096 字节 RAM）。

Z80-PIO（并行接口控制器）具有二个 8 位的I/O口，提供TTL兼容的接口（图4a）。A口有四种可能的操作方式：字节输出、字节输入、字节双向总线和位控。B口除没有字节双向方式外，其余操作方式都有。I/O 口逻辑由联络控制逻辑和六个寄存器组成（图4b）。六个寄存器是8位输入寄存器、8位输出寄存器、2位方式控制寄存器、8位屏蔽寄存器、8位I/O选择寄存器和2位屏蔽控制寄存器。最后三个寄存器仅用于I/O在程序的控制下按位方式工作的时候。PIO 的四十脚中，24 个用于 I/O 口和 CPU 总线，六个用于微处理器接口，三个中断控制，四根用于联络，另三个用于电

源、地和单相时钟。

六个内部寄存器中的四个是由Z80装入信息，供特征程序设计用。二位的方式控制寄存器的内容确定在PIO的四种工作方式中准备用哪一种。类似地，二位的屏蔽控制寄存器指明应加监视的任何外围接口线的有效状态（高或低）。在所有未屏蔽引脚为有效（逻辑“与”状态）或在任一未屏蔽引脚为有效（逻辑“或”状态）时它还可以容许产生中断。送入屏蔽寄存器的码确定了在给定状态条件下应监视哪些外围设备接口引脚。I/O选择寄存器中保存的码确定位控方式时哪些脚是输入或输出。另外二个寄存器存放输入或输出数据。

为减轻定时状态中的一些软件开销，CTC提供了四个可用软件置值的可程控定时和计数通道（图5）。每个通道不是在定时器方式就是在计数器方式下工作，且不论

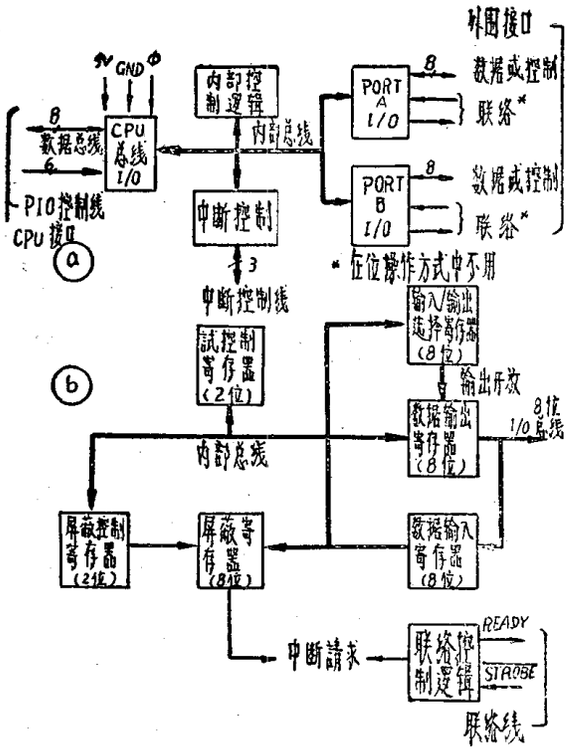


图4　借助二个并行8位I/O口，PIO电路(a)可按并行系统利用每个I/O口，或者在一根接一根基础上用作16根独立的I/O线。在每一I/O口的内部，在操作之前，由Z80向五个控制寄存器送数，使I/O置于初始状态。

在何种方式，都可以产生可程控的中断。另一些特征是包含可读出的减量计数器、可选择的16或256时钟定标器（每个定时器有一个）、供定时器启动用和给计数器或定时器自动再装入常数用的可选择正沿或负沿触发器。此外，三个通道都有“零读数/时间到”输出，可以驱动达林顿晶体管。

每个通道有二个寄存器，各长八位，由微处理器装入信息。其中一个寄存器，叫时间常数寄存器，将预置值送入减量计数器。另一个称为通道控制寄存器，存放通道工作方式和状态信息。每一通道中还有八位的减量计数器和八位定标器。在定时器方式中计数器的内容用定标器减小；在计数器方式中，其内容用时钟触发器输入减小。

计数器-定时器电路（CTC）的二十八个引脚中，八个连接数据总线，七个连接控制线，三个处理中断控制，三个用于电源、地和单相时钟。四个输入通道中，三个各有一根输入线和一根输出线，第四个通道仅有一根输入线。

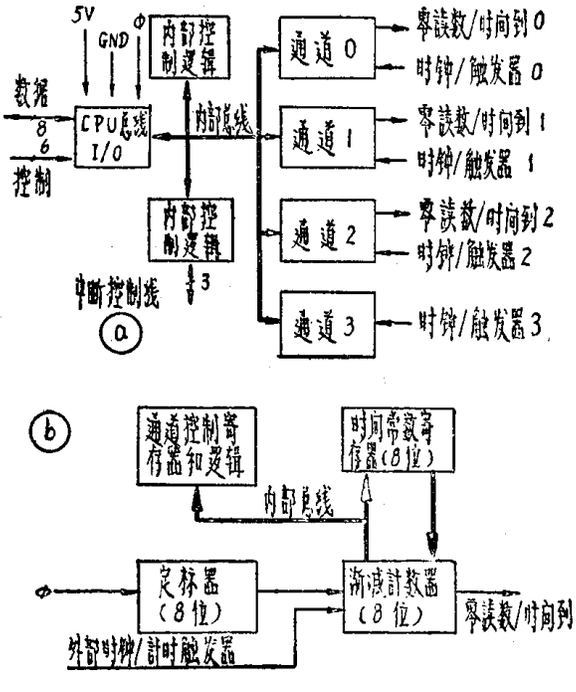


图5　每一CTC提供四个有计数/定时能力的通道，每个通道有一个8位计数器(a)。此外，每个通道有一个控制寄存器和一个可编程序的8位定标器（b）。

直接存储器存取(DMA)加速了数据传送

接口电路之一的DMA控制器，是为在Z80微型机系统的任何二个口之间高速传送数据块而设计的，但也可以用于其它微处理器。该电路是可编程序的单通道器件，它为数据传送提供全部地址、定时和控制信号（图6）。此外，DMA电路还可以从数据块搜索特定的可按位屏蔽的字节，而该数据块可以传送，也可以不传送。此电路可以高达1.2兆字节/秒的速度执行单一的传送、单一的搜索或同时搜索和传送操作，可自动从程序规定的起始地址中减去或加上出入口地址。

DMA片上有四种通信方式。1.一次一字节方式，即每次请求就送一个字节；2.群方式，出入口准备好多少就传送多少；3.连续方式是把微处理器封锁起来，直至操作完了；4.透明方式，它窃用再生周期。当电路发现字节一致或结束传送时，它可以通过程序的控制产生中断。或者可通过程序来安排完全的重复周期，以实现自动重复或按命令重复。片上有一个数据块计数器，当传送了一定数量的字节之后便发出一个信号，但传送并不停止。

DMA控制器中有数据和地址总线用的总线接口电路、控制电路参数的逻辑和寄存器和产生出入口地址的字节计数电路。还有地址增、减电路、调整被寻址的二个出入口的读-写定时的定时电路和容许字节匹配操作的比较逻辑（如发现字节一致，则DMA的状态寄存器中的标志置位）。中断和中断请求($\overline{BUSRQ}$)逻辑也做在片上，此逻辑包括控制寄存器、全部优先权编码逻辑和中断矢量寄存器。控制寄存器规定了片子产生中断的条件，优先权编码逻辑则在产生$\overline{INT}$或产生$\overline{BUSRQ}$输出之间作出选择，中断矢量寄存器用于自动转向中断服务例程。

DMA控制器有四十根引脚，其中二十四根用于地址和数据总线，五根用于微处理器控制总线。另外八根处理中断控制和定时，三根用于电源、地和时钟输入。

对于串行通信，串行输入/输出电路(SIO)有二个全双工可编程序通路，可以处理异步、同步和同步位规约(IBM Bisync、HDLC和SDLC)。它在任何同步方式中还可以产生循环冗余检错码。SIO有四个独立的串行口——二个用于发送，二个用于接收（图7）。可以处理有1、1.5或2个停止位和有奇偶或无奇偶校验的5、6、7或8位异步数据。

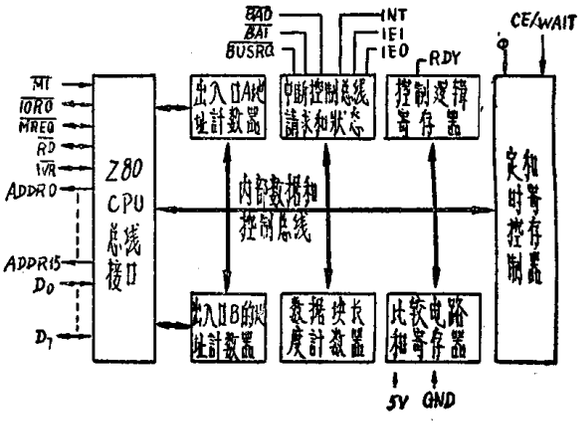


图6 DMA控制器有三类操作：只传送、只搜索和搜索传送相结合。挂到系统总线上的任何器件，都可由DMA来控制；内部计数器保持对源和目标地址跟踪

接口电路具有×1、16、32和64时钟方式，数据速率在0到600兆赫之间。发送器部分具有八根调制解调器控制线、四个接收器数据缓冲器和出错标志寄存器和二个发送器部分的缓冲器。总线-I/O控制部件包括通道和寄存器选择、读/写控制和中断响应周期特殊定时控制等用的逻辑。中断逻辑包括链形电路和二个专用八位控制寄存器（处理各种各样的中断）以及用于中断响应的八位向量寄存器。

三个接收缓冲器为高速数据传输的中断服务提供充裕的时间。接收器-移位寄存器受接收控制逻辑的控制，接收控制逻辑包括二个八位寄存器，分别用于选择接收方式和

规定选中的接收方式。另外有二个供可编程序同步字符用的八位寄存器。外部状态寄存器是八位的只读寄存器，它指示调制解调器控制引脚的状态和几个内部状态条件。内部状态寄存器还指示SIO的状态。每一通道自备接收、发送和状态寄存器组。

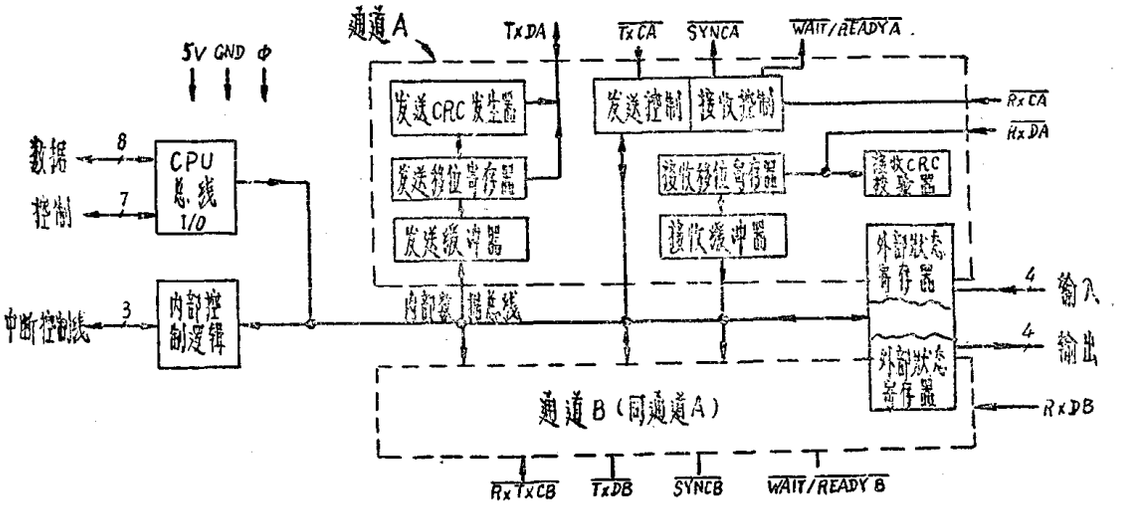


图7　二个独立的全双工串行I/O通道制在SIO器件上。每一通道都可以通过程序的安排以异步或同步方式工作，包括Bisync和HDLC/SDLC方式。

在熟悉了系统的所有基本组成部分之后，就可以通过软件把它们组织成工作系统。因为Z80的指令系统很丰富，用人工的办法对软件程序进行汇编，对大多数应用来说可能是太复杂了。应利用专门的开发系统或分时系统。

开发系统加速软件的研制

Zilog公司提供的Z80开发系统和软件，包括若干可用于研制硬件或研制软件或二者相结合的大型专用系统（见表2）。该公司还提供汇编程序、编译程序、分时服务、Basic和PLZ(即将提供Cobol和Fortran)。

开发系统中的全部程序语句，用文本编辑程序处理后存入双面的软盘文件管理系统。一旦形成文件，就可对程序调试，并可用汇编程序或编译程序译成机器码，以备Z80用。机器码可用具有询问、控制和跟踪能力的硬件/软件调试程序包测试。

在管态中系统有四种工作环境：建立文件、编辑、调试和汇编。文件建立包括若干出色的标准功能，如把记录存入磁盘、从磁盘取出记录、修改记录和存新的结果等。开发系统的调试和汇编程序提供某些出色的强有力的功能。利用调试命令可以建立断点，比较存储器块和跟踪操作。

在调试方式中，举例来说，系统的更新数据，可以在实时执行程序的过程中，存入专用的存储器中。一旦发生了任何用户定义的条件（如把$8B_{16}$出入口的第六位置1或把地址$21C8_{16}$的内容读出等），则程序的执行可被挂起，系统可重新进入管态。程序结束前的最后256个更新数据的全部记录，可存入系统存储器，供用户使用。

开发系统中的主汇编程序具有下列功能：宏汇编、条件汇编、大型文件分类符号表的相互参照汇编。所有这些可选用的功能以及打印和列表功能，都可在汇编时通过置参量值来获得。带I/O管理的浮动汇编程序可汇编出浮动目标程序块，并有一个连接引导程序。这样就可以指定应该包括在正在被汇编的现文件中的其它文件，从而可使不同的程序连接起来。

系统中的文本编辑程序包含了许多命令（甚至比许多完善的小型机的编辑程序的还

多），有助于对源文件进行处理。尽管它是行编辑程序（指示字总是指向行的开头），但仍具备若干字符串处理命令。借助于自动分页功能，可对超过存储器工作区间容量的文件进行编辑。Put（放）和Get（取）命令有助于将文件的一部分从一个磁盘文件复制到另一磁盘文件，或把它们嵌入程序。借编辑程序中的二十多个命令，可实现文本重复、修改、存储、行号打印和宏编辑功能。

为研制高级语言程序，可以利用Basic解释程序，以会话方式编写和调试程序。此外，PLZ语言（面向过程的语言）也可供常驻内存应用，这种语言的语法和语义融合了Algol、PL/1和Pascal等语言的风格。用它可以利用Z80的体系结构，可以编译出高效率的目标码和易于转换成机器码。共有二级。第一级PLZ把汇编语言和产生浮动程序块所需的语句结合一起；第二级类似于高级系统语言，其中的单个语句可以代替一系列汇编语言语句。

表2　　硬件和软件支援

类　型	单价（美元）	名　称	说　明
系　统	8990	Z80硬件和软件开发系统	3千字节ROM、1千字节RAM供系统管理程序； 16K字节RAM；实时调试程序块； 双面软盘；做成电路的模拟器； RS-232或电流回路接口； 软件和用户手册；额外的卡片槽； 双机箱系统；打印机、PROM写入器等的通用接口；
	6990	Z80软件开发系统	同上，但无做成电路的模拟器
	6990	Z80硬件开发系统	同第一种系统，但无通用接口
	5990	Z80微型机系统	在可包含Z80扦件板（如MCB、MDC等）的任意组合的一个机箱内有双面的软盘系统
驻留内存的软件	未标价	OS Z80-Z80开发系统和MCB系列的操作系统	汇编程序：将汇编语言记忆符译成机器语言。包括宏汇编、条件汇编、实际上为任意长的程序和分类符号表汇编（带完全的相互参照表）。 浮动汇编程序和连接装配程序： 将各自单独汇编好的程序连接起来，以供执行； 编辑程序环境：容许用户输入和修改文本，如汇编语言源程序等。 文件环境：对用户在写、调试和执行程序的过程中产生的磁盘文件进行控制和处理。 调试环境：使用户能利用各类调试工具来输入、测试和存放程序。
	未标价	Basic解释程序	此程序提供一种翻译语言，可在执行时在逐个语句的基础上译成机器码。
	未标价	PLZ-Zilog 驻内存程序设计语言	从浮动汇编程序到高级系统程序设计； · 可利用Z80体系结构 · 编译出高效率的目标码 · 易译成机器语言 二级语言可适应程序设计任务的需要
交叉软件	未标价	Z80交叉汇编程序 Z80-PLM语言编译程序	有ASCII 16位Fortran和PLI形式 全PLM语言编译程序产生Z80目标码

附录Ⅰ　Z80微处理器结构

除了八个通用的16位寄存器和运算器外Z80微处理器还包括所有的总线控制、存储器控制和定时信号。Z80和Intel 8080A和8085微处理器是向上兼容的。

8080[1]所备的全部寄存器，在Z80中有二套，除了8080的八个8位寄存器（A、F、B、C、D、E、H和L）外，还有另一组寄存器（A′、F′、B′、C′、D′、E′、H′和L′）和若干其它的专用寄存器。增加的寄存器包括二个16位变址寄存器（IX和IY）、一个八位中断向量寄存器（I）和一个八位存储器再生寄存器（R）。十六位的栈顶指示器和十六位的程序计数器(PC)是从8080的寄存器组转过来的。

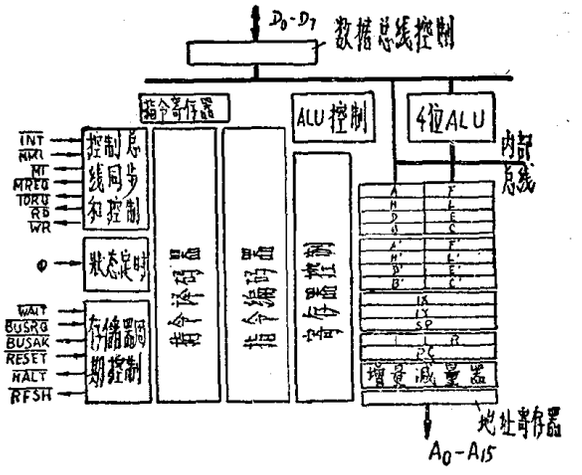
一般说来，所有的指令全访问主寄存器组内的寄存器，而对另一组寄存器的存取则通过二条交换指令实现。第一条交换指令交换累加器和标志寄存器（A、F和A′、F′）的内容，而另一条指令交换其它六个通用寄存器（如B、C和B′、C′等）。二条指令都是单字节的执行时间最短的指令，所以用四个时钟周期（4兆赫时钟时为1微秒）就可完成整个交换过程。这些指令和寄存器极便于高速处理单级中断。

Z80的二个变址寄存器是独有的，在8080结构中没有相应的寄存器。从其作用来说，它们和6800微处理器[2]中的一个变址寄存器相似。利用这种（变址）方式的指令，如取到累加器指令[LD A,（IX＋7）]，包含了单字节的偏移字段（在这里是＋7），操作数的有效地址等于偏移值和IX寄存器内容的和。这种寻址方式特别适用于数据表访问、多字节输入或将指示字送往子程序参数组。Z80把偏移量字节当作以2的补码表示的数，这样既可以作正的变址，也可以作负的变址。

Z80的一个非同一般的特征是其自动再生动态存储器的能力。它的存储器再生寄存器作为一个七位计数器工作，每次取出指令码后计数器内容自动加1，然后放到地址总线的低七位上，同时处理器的状态线变低，指明有有效再生计数存在。因为这整个过程发生在Z80内部对操作码译码的时候，所以决不会与总线上的其它处理器活动发生冲突。

I寄存器形成地址的高八位。当有中断产生和Z80处于向量方式时，地址的低八位由产生中断的外设供给。作为对中断的响应，微处理器执行一条带复合地址的间接调用指令。全部支援器件具有相应的寄存器，其中存放着地址的低八位，在中断得到响应时便把它们提供给Z80。

运算器能完成十二种基本操作：加、减、与、或、异或、比较、位测试、位置位、位复位、递增、递减、左或右移和循环（算术或逻辑），它通过有缓冲的内部总线跟寄存器和外部数据总线通信。每条指令从存储器取出后就放入指令寄存器，由控制部分进行译码，并向Z80的子系统发送所有的控制信号。



附录II　Z80的指令系统

Z80能执行150种不同的指令，包括8080A指令系统的全部78条指令在内。Z80有七类基本的指令：取数和交换、数据块传送和搜索、算术和逻辑、位操作（置位、复位和测试）、转移、调用和返回和输入/输出，还有基本的微处理器控制指令。总括起来，Z80能识别696个操作码，其中244个是8080A中有的。

取数指令是在Z80内部寄存器之间或寄存器和外存之间传送数据。所有这些指令全要指定数据源和目标。数据块传送指令又将任何一块存储器的内容送到另一处。搜索指令可从任何二块外存储器中找出任何一个八位字符。一旦找到了字符，指令便告结束。

算逻指令对累加器另一通用寄存器或外存中的数据进行操作，结果保存在累加器中，相应的状态标志置位。位操作指令可使累加器、任何通用寄存器或任何外存单元中的任何一位置位、复位或对它测试。转移、调用和返回指令用于程序内不同存储单元间的传送。

I/O指令是在外部存储器的存储单元或通用寄存器和外部I/O设备之间传送数据。不论在哪一种情形里，出入口的号码是由任何I/O操作期间地址总线的低八位提供。基本的微处理器控制指令包括中断开放触发器的置位或复位指令或建立中断响应方式的指令。

除8080的七种寻址方式（直接、寄存器、寄存器间接、修改0页、扩充、蕴含和立即）外，Z80还增加了三种寻址方式：相对、变址和位寻址。

专用的字节调用指令使Z80程序可以调出存储器0页的八个单元中的任何一个，在修改0页寻址方式中，用一个字节就可以指明完全的十六位地址，从而节省了存储空间。

相对寻址是用操作码后的一个字节规定相对于现程序计数器值的位移量。位移量以2的补码形式出现，因而最多可位移+127或-128字节。扩充寻址是在指令中包含二个地址字节。

变址寄存器也可用作地址的一部分。在变址方式中操作码后面的一个数据字节是位移量。这个位移量必须加到操作码指定的变址寄存器，以求得存储器地址字。还有蕴含寻址方式，就是操作码把Z80的一个或多个寄存器的内容作为操作数。这种寻址方式可使Z80能对任何存储单元和寄存器进行存取，还可使任何一位置位、复位或对之测试。

Z80的指令系统

记忆符	说明
八位取数指令	
LD r,r'	r'寄存器的内容取到r
LD r,n	n取到寄存器r
LD r,(HL)	寄存器对HL指定的存储单元内容取到r
LD r,(IX+d)	IX+d指定的存储单元内容取到r
LD r,(IY+d)	IY+d指定的存储单元内容取到r
LD(HL),r	r的内容放到HL指定的存储单元
LD(IX+d),r	从r放到IX+d指定的存储单元
LD(IY+d),r	从r放到IY+d指定的存储单元
LD(HL),n	数n放到HL指定的存储单元
LD(IX+d),n	数n放到IX+d指定的存储单元
LD(IY+d),n	数n放到IY+d指定的存储单元
LD A,(BC)	BC指定的存储单元内容取到累加器
LD A,(DE)	DE指定的存储单元内容取到累加器
LD A,(nn)	nn存储单元的内容取到累加器
LD(BC),A	累加器内容放到BC指定的存储单元
LD(DE),A	累加器内容放到DE指定的存储单元
LD(nn),A	累加器内容放到nn指定的存储单元
LDA,I	从I取到寄储器A
LDA,R	从寄存器R取至累加器
LDI,A	从累加器取到寄存器I
LD R,A	从累加器取到寄存器R
十六位取数指令	
LD dd,nn	把nn取到寄存器dd
LD IX,nn	把nn取到寄存器IX
LD IY,nn	把nn取到寄存器IY
LD HL,(nn)	存储单元nn内容取到寄存器L,(nn+1)取到H
LD dd,(nn)	存储单元nn内容取到寄存器dd
LD IX,(nn)	存储单元nn内容取到寄存器IX
LD IY,(nn)	存储单元nn内容取到寄存器IY
LD(nn),HL	从HL放到存储单元(nn)
LD(nn),dd	从寄存器对dd取到存储单元(nn)
LD(nn),IX	同上,但用IX
LD(nn),IY	同上,但用IY
LD SP,HL	从HL取到栈指示器
LD SP,IX	从IX取到栈指示器
LD SP,IY	从IY取到栈指示器
PUSH qq	从寄存器对qq取到栈内
PUSH IX	从IX取到栈顶
PUSH IY	从IY取到栈顶
POP qq	从栈顶取到寄存器对qq
POP IX	从栈顶取到IX
POP IY	从栈顶取到IY
交换、传送和搜索指令	
EX DE,HL	交换DE和HL的内容
EX AF,A'F'	交换AF和A'F'的内容
EXX	交换二组(各六个)通用寄存器的内容
EX(SP),HL	交换栈顶指示器和HL的内容
EX(SP),IX	同上,但用IX寄存器
EX(SP),IY	同上,但用IY寄存器
LDI	从(HL)取到DE,DE和HL内容加1,BC减1
LDIR	同上,但循环到(BC)=O
LDD	从存储单元(HL)取到(PE),DE、HL和BC减1
LDDR	同上,但循环到(BC)=O
CPI	比较累加器和(HL)的内容,如相等则将Z置1,HL加1,BC减1
CPIR	同上,但重复到BC=O
CPss	比较操作数S和累加器内容
CPD	同CPI,但HL减1
CPDR	同CPIR,但HL减1
八位算逻指令	
ADD A,r	r内容加到累加器
ADD A,n	字节n加到累加器
ADD A,(HL)	HL内容加到累加器
ADD A,(IX+d)	存储单元(IX+d)加到累加器
ADD A,(IY+d)	同上,但(IY+d)
ADCA,s	操作数S和累加器进行带进位加
SUBs	从累加器减去r、n、HL、IX+d或IY+d的内容
SBCs	同上,但还减去进位标志
ANDs	操作数S和累加器逻辑与
ORs	操作数S和累加器逻辑或
XORs	操作数S和累加器异或
INCr	寄存器r加1

记忆符	说明
INC (HL)	由 HL指定地址的存储单元内容加 1
INC(IX+d)	由 IX+d指定地址的存储单元内容加 1
INC(IX+d)	由 IY+d指定地址的存储单元内容加 1
DECm	操作数m减 1
十六位算术指令	
ADD HL, ss	寄存器对SS加到 HL
ADC HL, ss	同上，但带进位标志
SBC HL, ss	从HL减去SS 和进位标志的内容
ADD IX, pp	寄存器对 PP 加到IX
ADD IY, rr	寄存器对 rr 加到IY
INC SS	寄存器对SS 加 1
INC IX	IX 寄存器加 1
INC IY	IY 寄存器加 1
DEC	寄存器对 SS 减 1
DEC IX	寄存器 IX 减 1
DEC IY	寄存器 IY 减 1
通用算术和控制指令	
DAA	累加器十进制调整
CPL	累加器内容求反
NEG	累加器内容求反和加 1
CCF	进位标志求反
SCF	进位标志置 1
NOP	空操作
HALT	暂仃，等待中断或复位
DI	封锁中断
EI	开放中断
IM0	把微处理器置于中断方式 0
IM1	" 1
IM2	把微处理器置于中断方式 2
循环和移位指令	
RLCA	累加器向左循环移位
RLA	同上，但包括进位标志
RRCA	累加器向右循环移位
RRA	同上，但包括进位标志
RLCr	寄存器 r 向左循环移位
RLC(HL)	HL 指定存储单元向左循环移位
RLC(IX+d)	IX+d 指定的存储单元向左循环移位
RLC(IY+d)	IY+d 指定的存储单元向左循环移位

记忆符	说明
RLm	同任一RLC 指令，但包括进位标志
RRCm	同 RLC 但为向右移
RR m	同 RLm 但为向右移
SLAs	左移 (任一RLC 寄存器)
SRAs	同上，但右移，并保持最高有效位不变
SRLs	同 SLA，但右移
RLD	同时从AC_L移四位到L，L到H和H到AC_L
RRD	同时从AC_L移四位到H，H到L和L到AC_L
置位、复位和测试指令	
BIT b, r	测r寄存器的位b
BIT b, (HL)	测HL 指定地址的存储单元的位b
BITb,(IX+d)	测存储单元(IX+d)的位 b
BITb,(IY+d)	测存储单元(IY+d)的位 b
SET b, r	寄存器r的位b置 1
SET b, (HL)	HL 指定的存储单元内的位b置1
SETb,(IX+d)	同上，但 IX+d
SETb,(IY+d)	同上，但 IY+d
SET b, S	操作数m 的位b 置 0
转移、调用和返回指令	
JPnn	无条件转移到存储单元nn
JPcc, nn	如条件cc 为真，转nn；否则继续往下执行
JRe	无条件转移到PC+e
JRC, e	如C= 0 则继续，如C= 1 则做 JR e
JR NC, C	JR c,e 的反
JR Z, e	JRZ, e 的反
JP(HL)	从(HL)取到PC
JR(IX)	从(IX)取到 PC
JP(IY)	从(IY)取到 PC
DJNZ, e	寄存器 B 减 1，如B= 0 则相对转移
CALLnn	无条件调用起始地址为 nn 的子程序
CALLcc, nn	如条件 cc 为真，则调用存储单元nn内的子程序
RET	从子程序返回
RET cc	如cc为假则继续往下执行，否则执行RET
RETI	从中断返回
RETN	从不可屏蔽中断返回
RSTp	PC 存入栈内，0 送 PC_H，重新启动矢量送PC_L。

续

记忆符	说明	记忆符	说明
输入/输出指令			
IN A, n	把外设n的输入数据取到累加器	OUT(C), r	寄存器r的内容放到输出口(C)
INr, (C)	把外设(c)输入的数据取到寄存器r	OUTI	由HL指定地址的存储单元的内容放到输出口(C)，HL加1，B减1
INI	地址由C规定的存储单元的内容存入HL规定的地址内，B减1，HL加1	OTIR	同上，但重复到B=0
INIR	同上，但重复到B=0	OUTD	同OUTI，但HL减1
IND	同INI，但HL也减1	OUDR	同OTIR，但HL减1
INDR	同INIR，但HL也减1		
OUT n, A	累加器内容放到输出口(n)		

注：

b为三位代码，指明要修改的位的位置
cc为三位代码，指明8个条件码中应该用哪一个
d为8位的位移量
dd指寄存器对BC、DC、HL或栈指示器
e表示带符号的对2补码表示的数，其值在−126和+129之间
m是8位的数
n是8位的数
nn为二个8位字节
p指明0页上存放八个重新启动矢量的存储单元之一
pp指明寄存器对BC、DE、IX或栈指示器
qq指明寄存器对AF、BC、DE或HL
r或r′指明寄存器A、B、C、D、E、H或L或其对应寄存器
rr指明寄存器对BC、DE、IX或栈指示器
s指明r寄存器，数据字n，或HL、IX+d或IY+d寄存器内容指定的存储单元的内容
ss指明寄存器对BC、DE、HL或栈指示器

[薛家政　译]

参　考　文　献

1. Mckenzie K. 和 Nichols A. J.,"Microprocessor Basic Part 2: The 8080",«Electron Design», May 10, 1976, pp. 84—92.
2. Mazur T., "Microprocessor Basics Part 4: The 6800", «Electronic Design», July 19, 1976, pp.66—77.

Z80和Z80A中央处理器(技术手册)

1.0 引　　言

“微型计算机”这个术语已被用来描述近几年来设计的各类小计算装置。从由TTL中规模集成电路(MSI)构成的简单“微程序控制”的控制器直到中央处理器 (CPU)的一部分是由 TTL 大规模集成电路 (LSI) “位片”构成的低档小型计算机，都用这个“词儿”来称呼。但是近年来对 LSI工艺有重大影响的还是 MOS 工艺。用这种工艺，仅由几块MOS LSI电路就可以构成完善的、功能极强的计算机系统。

Zilog 公司的 Z80 元件系列是现时微型计算机技术发展中的重大进展。这些元件可以和任何类型的标准半导体存储器相结合，组成具有各种各样功能的计算机系统。例如只要二块 LSI 电路片和三个标准的 TTL MSI组件，就可以组成一个简单的控制器。加上额外的存储器和输入/输出 (I/O) 设备，计算机的功能可以达到和小计算机一样。由于计算功能可在宽广的范围内变化，用户便可以建立种种不同的标准模块，以满足各种各样应用的需要。

微型计算机市场上 MOS 大规模集成工艺占优势的主要原因是其价格较低。例如 MOS LSI微型计算机已取代了终端控制器、外设控制器、交通信号控制器、销售点终端、智能终端和测试系统等应用中的 TTL 逻辑。事实上， MOS LSI 微型计算机正在闯入几乎每一种应用电子技术的产品中，甚至取代了许多机械系统，如秤量装置和汽车控制器等。

MOS LSI 微型机算机市场业已形成，应用这些计算机的新产品层出不穷，其发展