

微 型 计 算 机

—Z-80

上海电子计算机应用服务部

目 录

1. 微型计算机 Z80 简介	1
2. Z80 和 Z80A 中央处理器 (技术手册)	27
3. Z80 和 Z80A 计数器定时器电路 (技术手册)	142
4. Z80 并行入/出电路 (技术手册)	179
5. Z80 串行入/出电路	211
6. Z80 和 Z80A 直接存储存取 (DMA)	267
7. Z80A 微处理器与 16 脚 RAM 的接口	310
8. Z80 系列的程序中断结构	329

编者按：Z80 微型计算机是美国 Zilog 公司生产的第一种八位微型机，也是目前所有八位微型机中功能最强的一种。其特点是指令执行时间短，接口简单，采用廉价的动态存储器和有丰富的指令系统等等，因而在国际市场上较引人注目。本文集着重介绍 Z80 的硬件，包括中央处理器、并行输入/输出控制器、串行输入/输出控制器、计数器/定时器电路、直接访存、中断结构和动态 RAM 的接口等内容。

由于我们水平有限，译文难免有错误不妥之处，敬请读者批评指正。

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) 和一组辅助元件板 (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 脚的双列直插式封装外，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 ——已装好	435 495
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℃, 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

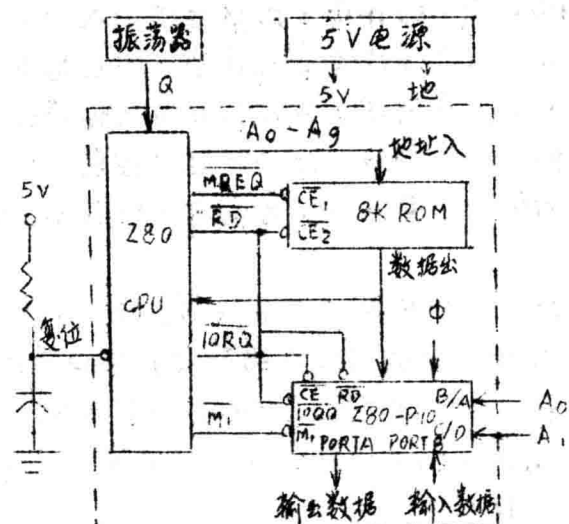


图1 最小 Z80 系统，可由微处理器、振荡器、某些存储器
和 PIO 这样的 I/O 口组成，只要再加上电源和复位
电路。

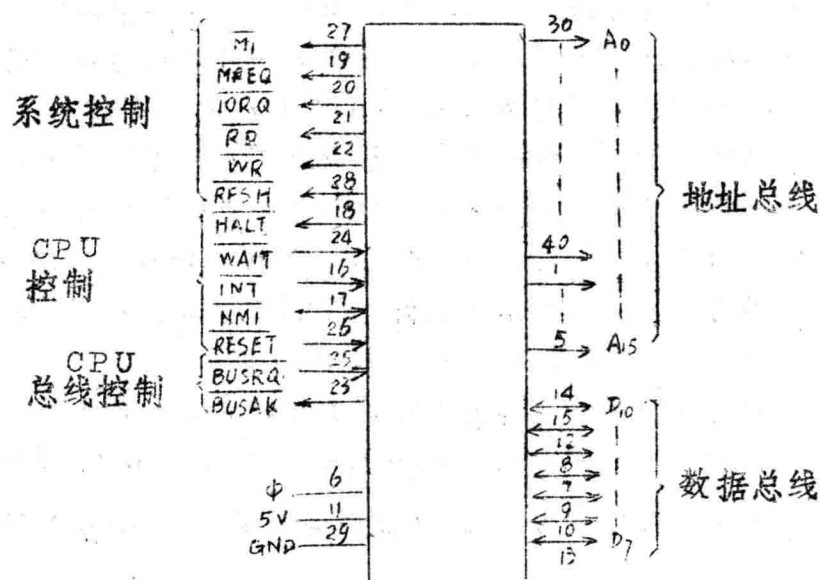


图2 Z80 片上的三条主要总线，即地址总线、数据总线和控制总线。控制总线又可分成三条更小的总线，分别用于
系统控制、处理器控制和总线控制。

起着总线请求线 ($\overline{\text{BUSRQ}}$) 的作用。由这根线输入的信号，不仅要求微处理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 ($\overline{\text{BUSAK}}$) 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。

六个系统控制信号都是从 Z80 输出。 $\overline{\text{M}}$ 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 ($\overline{\text{MREQ}}$) 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 ($\overline{\text{IORQ}}$) 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 ($\overline{\text{RD}}$ 和 $\overline{\text{WR}}$) 也是低为有效。 $\overline{\text{RD}}$ 指明微处理器要从存储器或 I/O 设备读数据，而 $\overline{\text{WR}}$ 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 ($\overline{\text{RFSH}}$) 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 $\overline{\text{MREQ}}$ 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 ($\overline{\text{HALT}}$)，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 ($\overline{\text{WAIT}}$) 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， $\overline{\text{RESET}}$ 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 ($\overline{\text{INT}}$) 输入和不可屏蔽中断 ($\overline{\text{NMI}}$) 输入。当 $\overline{\text{INT}}$ 线为低时，如果由软件控制的中断开放触发器 ($\overline{\text{IFF}}$)

~ 4 ~

处于开放状态和如果 $\overline{\text{BUSRQ}}$ 线为高，则将使微处理器在执行完指令之后中断。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 $\overline{\text{IORQ}}$ ）。

$\overline{\text{NMI}}$ 线是由负沿触发的输入，比 $\overline{\text{INT}}$ 线具有较高的优先权。不论 $\overline{\text{IF}}$ 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扞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 DDA

指令时，此标志用来修正二 - 十进制加或减操作的结果。）减标志则帮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 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

图3 甲型代中断结构的互连网可以如任意位置的外围设备中通过二80为基址的地址控制程序。外部由总机或靠近z80,其中浙优先机越前。

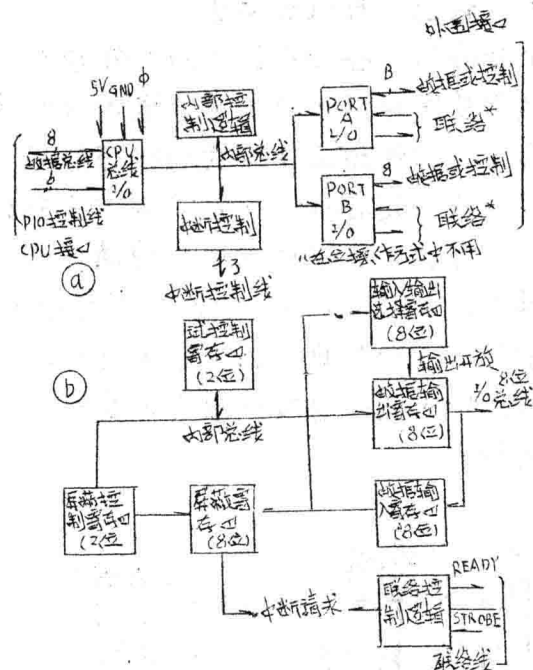


图4 借助二个并行8位I/O口，PIO电路(a)可按并行系统利用每个I/O口，或者在一根接一根基础上用作16根独立的I/O线。在每一I/O口的内部，在操作之前，由Z80向五个控制寄存器送数，使I/O置于初始状态。

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编程定时和计数通道(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，且不论在何种方式，都可以产生可编程的中断。另一些特征是包含可读出的减量计数器，可选择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。

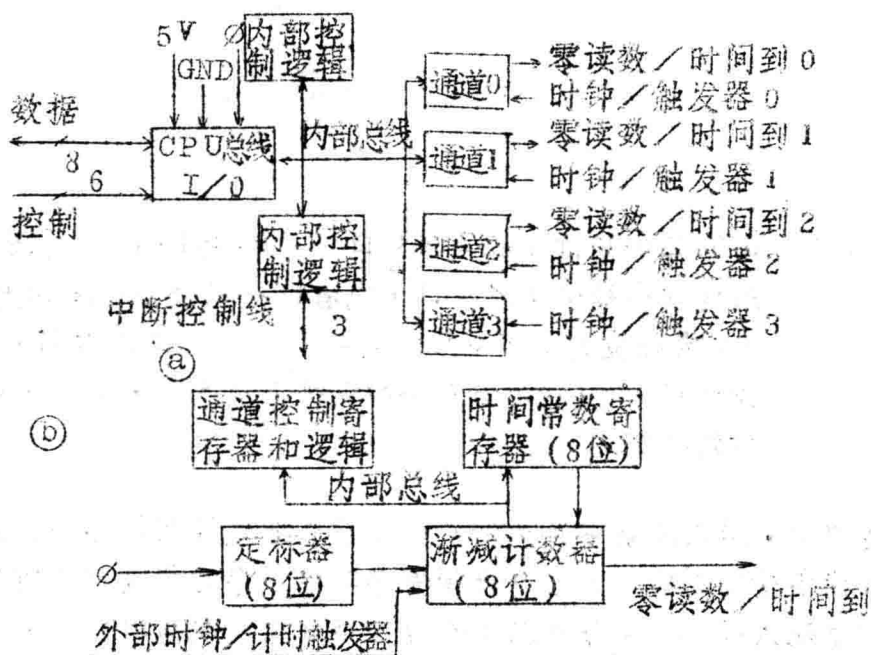


图5 每一CTC提供四个有计数/定时能力的通道，每个通道有一个8位计数器(a)。此外，每个通道有一个控制寄存器和一个可编程序的8位定标器(b)。

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（ $\overline{\text{BUSRQ}}$ ）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 INT 或产生 $\overline{\text{BUSRQ}}$ 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程序通路，可以处理异步、同步和同步位规约（IBM Asynchronous、

HDLG 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。

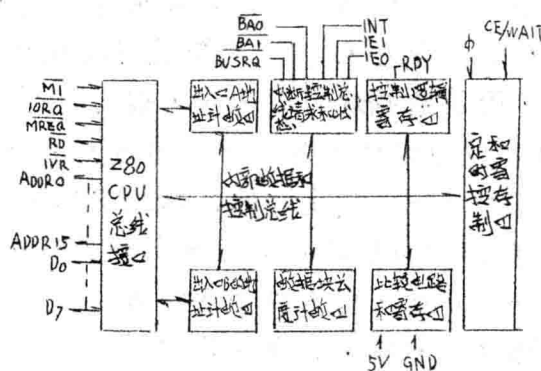


图 6 DMA 控制器有三类操作：只传送、只搜索和搜索传送相结合。挂到系统总线上的任何器件，都可由 DMA 来控制；内部计数器保持对源和目标地址跟踪。

接口电路具有 $\times 1, 16, 32$ 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二一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。另外有二一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。

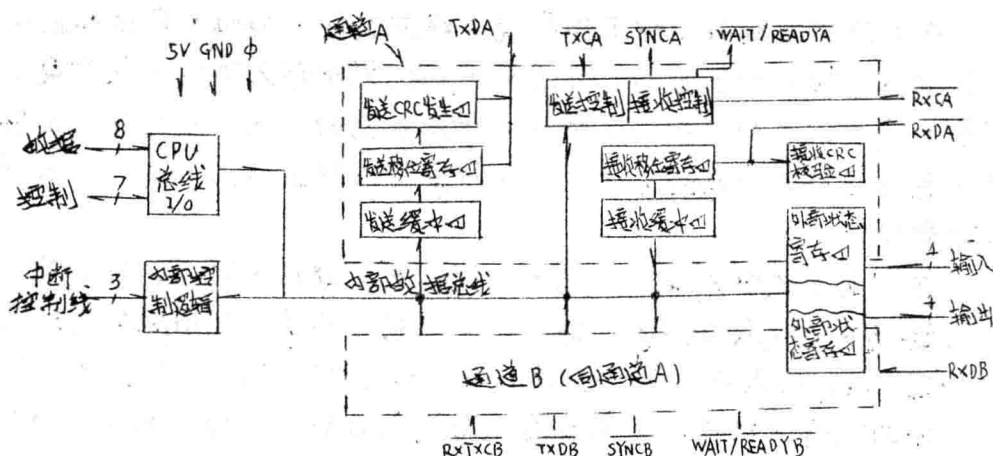


图7 二个独立的全双工串行 I/O 通道制在 SIO 器件上。每一通道都可以通程序序的安排以异步或同步方式工作，包括 Bisync 和 HDLC/SDLC 方式。

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 8B₁₆ 出入口的第六位置 1 或把地址 21C8₁₆ 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/I 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在一起；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; 16 K 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;
驻留内存的软件	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 扩件板 (如 MCB、MDC 等) 的任意组合的一个机箱内有双面的软盘系统
	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;