

Z80

微型计算机用户手册

[美]约瑟夫 杰·卡尔 著
张秀琼 吴定荣 译
杨希武 校



北京科学技术出版社

Z80微型计算机用户手册

[美]约瑟夫 杰·卡尔 著
张秀琼 吴定荣 译
杨希武 校

北京科学技术出版社

内 容 简 介

本书介绍了Z80系列芯片的结构、工作原理、引脚定义和功能,以及由一些芯片组成的实用电路。叙述简要,重点突出。列举了多种常用编码表格,以及按机器码大小顺序和按英文字母顺序排列的Z80指令表,供读者编程使用。

可供大专院校师生及科技工作者作为微型机参考资料,微型机培训班补充教材。

351.4/55

Z80 微型计算机用户手册

[美]约瑟夫·杰·卡尔 著

张秀琼 吴定荣 译

杨希武 校

※

北京科学技术出版社出版

(北京西直门外南路19号)

北京市新华书店发行 各地新华书店经售

北京工业大学印刷厂印刷

※

787×1092毫米 16开本 7.375印张 17.9万字

1985年12月第一版 1985年12月第一次印刷

印数1—6,000册

统一书号15274·025 定价1.50元

译 者 前 言

目前在世界上流行的 8 位微处理器中, Z80 微处理器具有功能较强, 应用较广等优点。以 Z80—CPU 为核心, 组成的 8 位微型计算机产品, 在国内市场中占据了很大的比重。它遍及科技、厂矿、教育、商业……以至家庭、个人各个领域。因此, 普及 Z80 系列产品的知识, 对当前微型计算机的科学技术开发和应用推广有重要意义。本书在介绍 Z80 系列芯片功能, 结构原理、应用等方面, 都有一定特色。它叙述精练, 突出重点, 对一些难予理解的关键问题, 列出典型例子, 深入浅出地进行解析。比如, 对 Z80 中断技术、输出/输入接口电路、存储器的连接以及扩充存储器的电路、一些控制信号功能的介绍等等的叙述, 都是较为周全和实用的。

本书可以作为微型计算机工作者的参考材料, 大专院校学生的微型计算机课程补充教材, 微型机培训班的补充教材, 以及微型机爱好者的自学教材。

我们对已发现的原书中的错漏, 均在译文中予以改正。原书中第 17 章是介绍 Z80 指令系统, 原文叙述过于繁锁, 篇幅很大, 为了便于读者较快了解、应用 Z80 指令系统, 而转译了 Zilog 公司出版的《Microcomputer Components Data Book》(1982/83 年版本) 一书中的 Z80 指令表 (第 10—16 页) 来代替。同时将原书简介 Z8000 的第 16 章略去。这样压缩了篇幅, 读者使用起来, 也比较简便快捷。

在翻译过程中, 得到了北京工业大学副教务长、微型计算机应用与研究协会会长杨希武副教授的大力支持、帮助, 他花费了不少精力和时间对译稿进行了全面的审核、校正, 才使本书得予尽早与读者见面。在此, 我们表示深切的感谢!

全书文稿的抄写, 是由杨小菊同志承担, 图表的绘制工作, 是由杨小玲同志承担, 瞿承忠同志对初稿进行了缮清、初校, 为本书出版做了不少工作, 我们表示十分的感谢。

译文中错漏在所难免, 恳切希望得到广大读者的指正。

译 者

1985 年 1 月

前 言

众所周知，前几年在半导体技术领域里发生了一场炸爆性革命。在七十年代初期出现了集成电路微处理器。开始这些芯片较简单，仅能为全数字可编程计算机提供部份电路，然而，今天以微处理器芯片为基础的计算机已迅速超过了成千上万的小型计算机，并已逐步推广应用到所有国家。芯片也有所改进。早期的芯片如 Intel8008 和 8080 已被后来的产品所取代。Zilog 公司通过改进 8080 芯片，生产出了 Z80 主频为 2.5MHz，后来的 Z80A 其主频加快至 4MHz。

虽然市场上有许多 8 位机，我自愿作为一名 Zilog 公司的“宣传员”。就我本人来说，我喜欢 Z80，这似乎是合情合理的。我自己的微计算机是以 Z80 芯片为基础，具有约 30K 的存储器，使我惊讶的是，这种台式计算机，却比一台能占满机房的 IBM1620 计算机还有更强的计算能力。在六十年代末，我作为大学工程学一年级学生，曾使用过 IBM1620 计算机。

Joseph J. Carr

目 录

第一章	Z80—CPU 结 构	(1)
§1	算术逻辑单元 (ALU)	(3)
§2	标志寄存器 (F和F')	(4)
第二章	Z80—CPU 引脚	(5)
§1	Z80—CPU 引脚 定义.....	(5)
第三章	Z80 系列 的支 援芯 片	(7)
§1	Z80—PIO.....	(7)
§2	Z80 — SIO.....	(9)
§3	Z80 — DMA	(12)
§4	Z80 — CTC	(14)
第四章	Z80—CPU 时 序 和 接 口 控 制 信 号	(16)
§1	数据/地址 总线.....	(16)
§2	输 入/输 出 (I/O) 操 作.....	(16)
§3	存 储 器 控 制 信 号.....	(17)
§4	CPU 控 制 信 号.....	(17)
§5	中 断 信 号.....	(18)
§6	$\overline{\text{BUSRQ}}$ 和 $\overline{\text{BUSAk}}$	(18)
§7	基 本 CPU 时 序.....	(18)
第五章	Z80 寻 址 方 式	(23)
第六章	Z80 指 令 系 统 概 述	(28)
§1	传 送 指 令.....	(28)
§2	交 换 指 令.....	(28)
§3	成 组 传 送 和 分 块 检 索 指 令.....	(28)
§4	算 术 和 逻 辑 指 令.....	(29)
§5	循 环 和 移 位 指 令.....	(30)
§6	位 操 作 指 令.....	(30)
§7	跳 转、调 用 和 返 回 指 令.....	(30)
§8	输 入/输 出 指 令.....	(32)
§9	CPU 控 制 指 令.....	(32)
第七章	Z80 标 志	(33)
第八章	存 储 器 与 Z80—CPU 的 连 接	(34)
§1	存 储 器 操 作 控 制 信 号.....	(34)
§2	地 址 译 码.....	(36)
§3	动 态 存 储 器.....	(39)

§4	插入等待状态·····	(39)
§5	存储器映象器件·····	(41)
第九章	Z80 I/O ·····	(42)
§1	Z80 I/O控制信号·····	(42)
§2	Z80 I/O 指令·····	(43)
§3	I/O 口地址译码器·····	(43)
§4	地址译码器的应用·····	(45)
§5	Z80 — PIO·····	(46)
第十章	外围设备的连接 ·····	(48)
§1	直接 I/O 方法 ·····	(48)
§2	电传打字机·····	(50)
§3	RS—232 接口·····	(53)
第十一章	中断 ·····	(55)
§1	什么是中断·····	(55)
§2	Z80 中断类型·····	(56)
§3	中断的硬件·····	(56)
§4	中断请求·····	(57)
§5	中断响应·····	(57)
第十二章	中断服务 ·····	(62)
§1	非屏蔽中断·····	(62)
§2	屏蔽中断·····	(65)
§3	方式—0 ·····	(65)
§4	方式—1 ·····	(67)
§5	方式—2 ·····	(67)
第十三章	算术运算 ·····	(69)
§1	ADD 指令·····	(70)
§2	ADC 指令 ·····	(73)
§3	SUB 指令 ·····	(73)
第十四章	逻辑运算 ·····	(73)
第十五章	Z80 所使用的几种信息编码 ·····	(75)
	· 按机器码大小顺序排列的 Z80—CPU 指令表·····	(76)
	· 按记忆符(英文字母)顺序排列的 Z80—CPU 指令表·····	(82)
	· ASCII码(American Standard Code Information Interchange)··	(88)
	· 博多电传打字机编码(BAUDOT TELETYPEWRITER CODE)··	(90)
	· 扩充的二—十进制交换码(EBCDIC) ·····	(91)
	· 8080/Z80 操作码相同、功能相同的指令·····	(92)
第十六章	Z80 指令系统 ·····	(95)

第一章 Z80—CPU 结构

Z80 集成电路微处理器是由 Zilog 公司设计和制造的。Mostek 公司也生产这种产品。Z80 与 Intel8080 微处理器相似，但比它更先进。实际上，Z80 芯片的设计者，就是原来在 Intel 公司设计 8080 芯片的设计者。

如果你对 8080 芯片熟悉的话，那么掌握 Z80 芯片将是很容易的。Z80 的指令系统不仅包括了全部 8080 指令而且还增加了一些。通常我们公认 Z80 有 158 条指令，而 8080 仅有 78 条。要特别指出，从不同角度来计算指令条数，是有不同指令条数的。在十六章里我们将给出 600 多条 Z80 指令，这些指令是 158 条指令的扩展表而已。（例如：BIT b, r 指令可测试七个不同寄存器的八位中任意一位，这就形成了 56 条指令）。

通常，除了那些依赖循环程序来定时的程序以外，任意一个能在 8080 系统上运行的程序，都能在 Z80 系统上运行。这是由于两者的时序不同。所以，那些依赖 8080 时序来定时的程序就不能在 Z80 上运行。

两者除了指令系统的区别外，Z80 与 8080 还有其他的区别。Z80 的程序员比 8080 程序员可以使用更多的内部寄存器和更多的寻址方式。

在硬件上还有几点不同。首先，在时钟上 Z80 废除了 8080 的双相时钟，而采用单相时钟。Z80 时钟为 2.5MHz，而较快的 Z80A 达到 4MHz，Z80B 达到 6MHz。其次在电源上，Z80 使用 +5V 单电源，而 8080 则需要 5V，-5V 和 +12V 三种电源。

Z80 还提供了一个 8080 所没有的非屏蔽中断和动态存储器刷新逻辑。

Z80 采用 N 沟道 MOS 技术，在触摸时应小心，以免因静电放电而损坏芯片。

图 1—1 为 Z80—CPU 芯片内部电路框图。它包括如下部件：算术逻辑单元（ALU），CPU 寄存器，指令寄存器，加上指令译码器和地址控制器。

Z80 有八位数据总线和十六位地址总线。用十六位地址总线，意味着 Z80 可以寻址 2^{16} 个不同的内存单元。

Z80 内部寄存器相当于 208 位可以由程序员访问的读写存储器。这 208 位构成 18 个八位寄存器和 4 个十六位寄存器，图 1—2 表示了 Z80 内部寄存器的结构。

主寄存器组由累加器（寄存器 A）和标志寄存器（寄存器 F），以及六个通用寄存器（B, C, D, E, H, L）组成。辅助寄存器组是主寄存器的“复制品”，它有累加器（A'）和标志寄存器（F'），以及通用寄存器 B', C', D', E', H' 和 L'。在任一时刻这些寄存器仅有一组起作用。例如，除了寄存器组交换指令以外，在任一条指令中同时使用 B 和 B' 是不可能的。

通用寄存器可以配对组成三对 16 位寄存器对 BC, DE 和 HL。辅助寄存器总是配成 16 位寄存器对 BC', DE' 和 HL'。

Z80 的专用寄存器包括中断矢量寄存器 I 和存储器刷新寄存器 R（两者均是 8 位），以及四个 16 位寄存器：变址寄存器 IX，变址寄存器 IY，堆栈指示器 SP 和程序计数器 PC。

中断矢量寄存器 I I 寄存器与外部设备共同提供中断服务程序入口的间接内存单元地

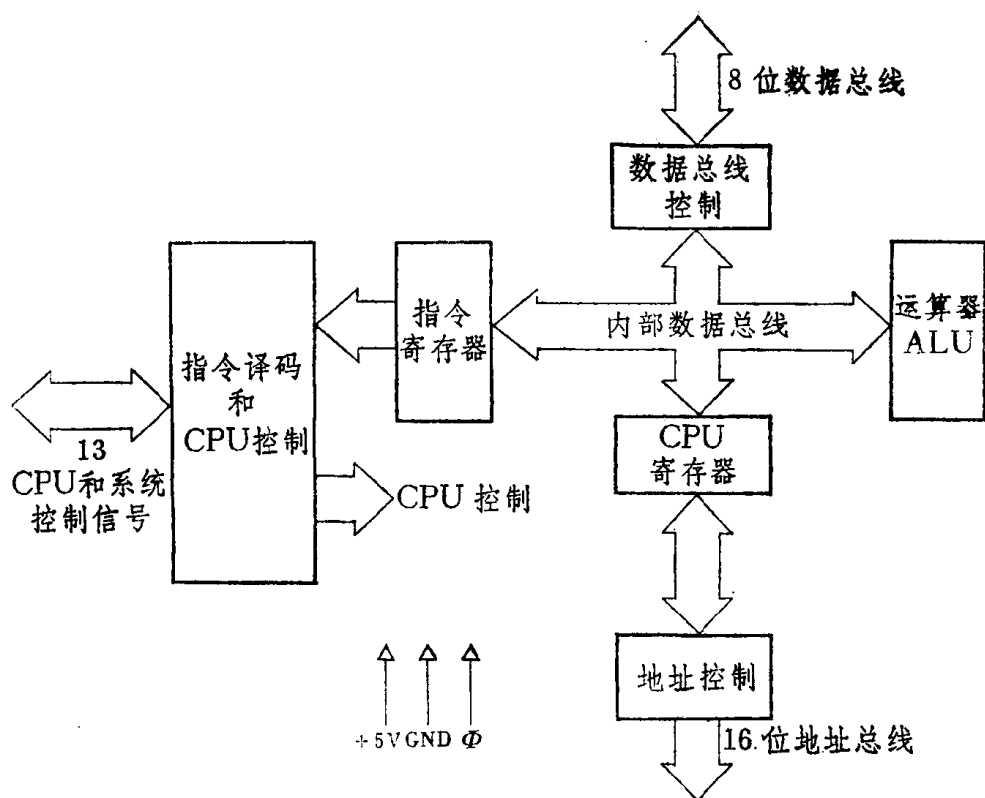


图1—1 Z80—CPU框图

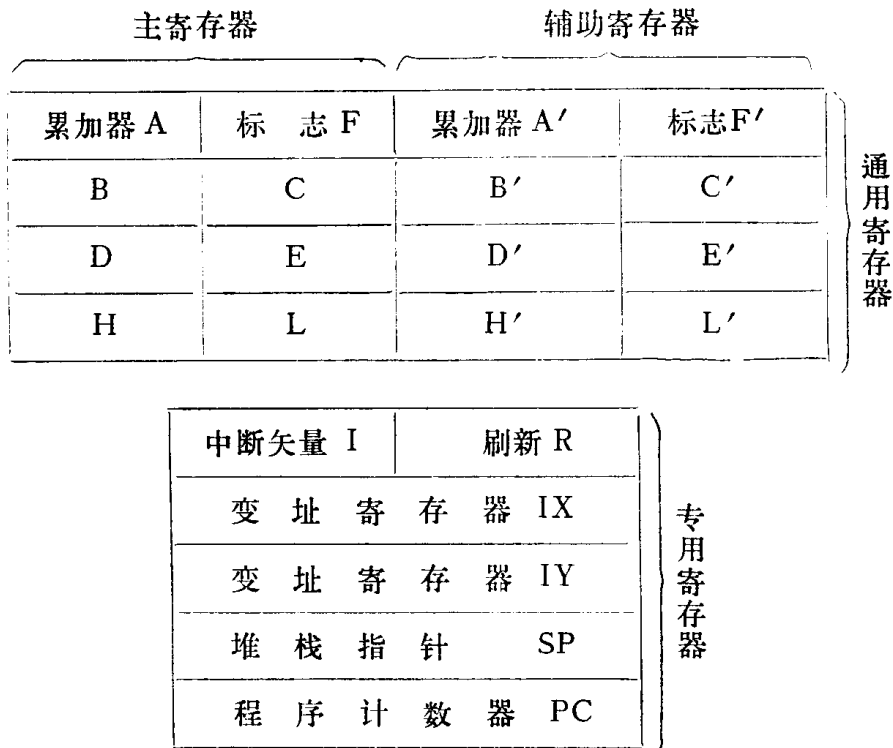


图1—2 寄存器结构

址，即 CPU 将转到该内存单元，而该单元内存放着中断服务程序的入口地址，由外设提供这16位地址的低8位，而 I 寄存器提供其高8位。

存储器刷新寄存器 R 当 CPU 从存储器取出指令，在对指令进行译码和执行指令期间，由 R 寄存器提供对动态存储器刷新的地址。每次取指令后，该寄存器中的7位自动增一，而第8位则是通过指令 LD R, A 来编程控制的。在刷新期间，刷新信号有效，R 寄存器的内容放到地址总线的低8位，而 I 寄存器提供其高8位。

变址寄存器 IX 和 IY 这两个寄存器在间接寻址指令时，作为存储器单元的指示器。存储器单元的实际地址为某个变址寄存器内容与位移量 d 的和，位移量 d 是整数（用2的补码表示）。两个寄存器 IX 和 IY 是互相独立的，请注意，有许多微处理器没有变址寄存器。

堆栈指示器 (SP) 该寄存器是一个两字节的寄存器，用来存放设置在存储器中的“后进先出”（LIFO）堆栈的栈顶地址。数据通过 PUSH 和 POP 指令，推入或弹出堆栈。

程序计数器 (PC) 在所有计算机里，程序计数器都是用来存放正要从存储器中取出的指令的地址。Z80 的程序计数器是16位的寄存器。在每一条指令取出后，PC 的内容自动地增加适当的数值（例如：单字节指令增1，PC+1；双字节指令 PC+2，等等）当发生跳转操作时，程序计数器将内含着转向存储器单元的地址。当程序返回时，如果没有发生跳转操作，则 PC 内含着下一条要取出的指令的存储器地址。

图1—3表示了调用操作时，程序计数器是怎样工作的。我们假定程序的起始地址为0200（16进制），到0206结束。当程序执行到0202单元时，遇到一条无条件调用指令，转向0612单元，为了说明起见，设存放在0612单元的子程序是一条返回指令（实际上也许没有这种情况，但为了说明问题却是很有用的），程序就跳回到下一个单元0205。注意，在这种情况下，0202的下一条指令不是存放在0203单元而是存放在0205单元。这是由于调用指令是一条三字节指令。在0202单元存放调用指令的操作码CD，在0203单元存放要转向的存储器单元地址的低字节，而在0204单元存放其高字节。

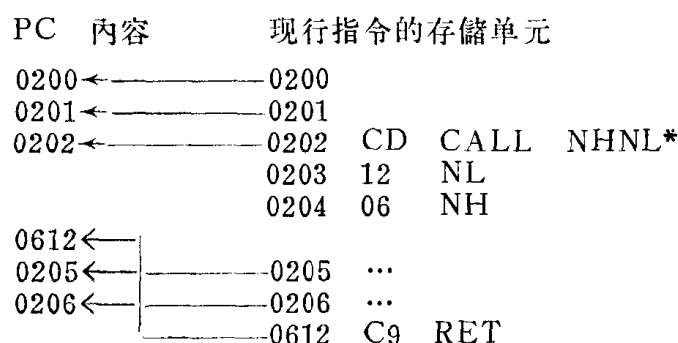


图1—3

* 原文为 C3 JP 0612

§1 算术逻辑单元 (ALU)

算术逻辑单元 (ALU) 是任何计算机和微处理器的心脏。计算机与其它数字电路的差

别就在于此。微处理器利用该电路来完成数据处理。在 Z80 微处理器里可完成的运算有：加、减、比较、逻辑与、逻辑或、逻辑异或、逻辑左移、算术左移、逻辑右移、算术右移、增量、减量、置位（即使该位为 1）、复位（即使该位为 0）和位测试，检查被测试的位是 1 还是 0。

§ 2 标志寄存器(F和F')

Z80 设置了两个状态寄存器：F 和 F'。在任一时刻只有一个有效，这取决于程序员所选择的是主寄存器组还是辅助寄存器组。这两个寄存器均为 8 位，它的每一位表示不同的状态条件。因此，F 和 F' 寄存器的各位也称为条件位。

通过对数据进行一定的算术运算或其他处理，来确定标志寄存器 F 和 F' 的各位是置位还是复位，程序可判别其操作结果。各位分配如下：

位(F/F')	名 称	意 义
0	C	进位标志。表示从累加器最高位（B7）的进位。
1	N	减标志。用于BCD减运算。
2	P/V	奇偶/溢出位。
3	X	不确定（未用）。
4	H	BCD半进位标志（BCD运算的第四位）。
5	X	不确定（未用）。
6	Z	零标志。运算结果为零时置 1。
7	S	符号标志。运算结果为负时置1,结果为零或正时置0。

第二章 Z80—CPU 的引脚

Z80 芯片是一种标准的 40 条引脚双列直插式封装 (DIP) 的集成电路组件。因为 Z80 芯片采用 NMOS 技术, 注意在使用这种芯片之前应熟悉它的使用规则, 否则, IC 芯片将会被损坏。(译注: 目前已有 CMOS 工艺的 Z80 芯片)

Z80 的引脚功能如图 2—1 所示, 关于引脚的定义在下面说明。

§ 1 Z80 引脚的定义

A0~A15	地址总线 (16 位)。允许对内存直接寻址达 64K (即 65 536 字节), 外加 256 个不同的 I/O 口。地址总线为三态, 输出, 对内存寻址需要使用全部 16 位, 而对 I/O 口寻址只使用其低 8 位 (A0~A7)。
D0~D7	8 位数据总线。与地址总线一样, 数据总线也是三态, 不过, 数据总线是双向 (输入/输出) 总线。
$\overline{M1}$	机器周期 1, 当该引脚为低电平时, CPU 处于取指令周期中的取操作码阶段。
$\overline{MREQ}$	存储器请求信号。低电平有效, 输出。当该引脚为低电平时, 在地址总线上的地址是要访问 (读或写) 存储器的有效地址。
$\overline{IORQ}$	输入/输出请求。低电平有效, 三态, 输出。该引脚有效时, 表示要对某个 I/O 口进行操作。地址总线的低字节 (A0~A7) 包含着所选中的口地址 (0 到 255 之间)。在这期间, 累加器 (或 B 寄存器) 的内容送到地址总线的高字节。此外, 在中断请求响应周期, 利用 $\overline{IORQ}$ 通知中断源, 将中断矢量送到数据总线上 (即中断服务程序入口地址表的低字节地址)。
$\overline{RD}$	读信号。低电平有效, 三态, 输出。用来表示 CPU 正在从内存或 I/O 口进行读操作。
$\overline{WR}$	写信号。三态, 输出, 低电平有效。表示 CPU 正在对内存或 I/O 口进行写操作。
$\overline{RFSH}$	动态存储器刷新信号。输出, 低电平有效。表示地址总线的低 7 位, 是动态存储器的刷新地址。
$\overline{HALT}$	暂停信号。输出, 低电平有效。指出 CPU 正在执行暂停指令。在暂停状态时进行空操作。
$\overline{WAIT}$	等待信号。输入, 低电平有效。指出被寻址的存储器或 I/O 设备, 还未准备好将数据传送到数据总线。
$\overline{INT}$	可屏蔽中断信号。输入, 低电平有效。通知 CPU 某个外设有中断申请。如果中断触发器 (软件控制) 为置位状态时, CPU 将在现行指令周期结束时响应中断请求。
$\overline{NMI}$	非屏蔽中断信号。输入。负跳变触发有效。非屏蔽中断请求输入信号, 不论中

断触发器处于什么状态，只要在该引脚输入有效负跳变信号。就使 CPU 在现行指令周期末了，响应非屏蔽中断请求，即强迫程序从 0066H 单元自动重新起动。

RESET 复位信号。输入，低电平有效。当信号有效时，使中断触发器复位，清除程序计数器（即将 PC 置为 0000H），以及清除 I 和 R 寄存器。该引脚可作为硬件控制程序跳转到 0000H 单元。

BUSRQ 总线请求信号。输入，低电平有效。表示请求 CPU 将地址总线、控制总线和数据总线置于高阻（三态）状态，以便其他外设获得对这些总线的控制权。它总是在现行指令机器周期末响应。

BUSAK 总线回答信号。输出，低电平有效。它与总线请求信号一起使用，通知申请使用总线的外设，CPU 的三总线已处于高阻状态。当 **BUSAK** 变为低电平时，申请的外设可以控制这些总线。

ϕ 系统时钟，输入信号。与 TTL 电平相容，其最高频率为 2.5MHz (Z80) 或 4MHz (Z80A)

GND 直流电源信号的地端。

+5 电源端。由稳压电源供给 +5 伏的直流电压。

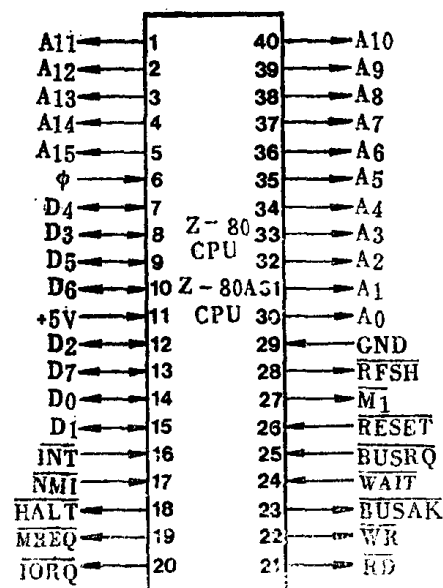


图2-1 Z80-CPU引脚图

第三章 Z80 系列的支援芯片

Z80 不是单片计算机 (Single-Chip Computer)，为了使 Z80 微处理器成为真正的计算机，我们还需附加外部电路。在某些计算机商品中，这些外部电路可以由 TTL 或 CMOS 芯片连接成能完成所需要功能的电路。为了比较容易组成计算机，Zilog 公司生产了专用的接口电路。

这些专用的接口芯片中，有两种用来作为串行和并行输入/输出接口。其中 Z80—SIO 是串行 I/O 接口芯片，而 Z80—PIO 是并行 I/O 接口芯片。若是 Mostek 公司生产的产品，这两种芯片的型号则分别为 MK3884(Z80—SIO)和 MK3881 (Z80—PIO)。

还有一种直接存储器存取 (Direct Memory Access) 接口芯片，称为 Z80—DMA (Mostek MK3883)。在计算机中，直接存储器存取芯片，能使外围设备不要通过 CPU 而直接把数据写入存储器或从存储器读出数据。这就使数据的存储操作更快，并且能节省 CPU 的时间。CPU 的时间在某些应用场合是极为宝贵的。

Z80—CTC(Mostek MK3882)是一种具有四个通道的多功能的计数器/定时器芯片，在 Z80 基本微型计算机系统 (Z80—Based microcomputer) 中，它提供计数器和定时器的功能。

§ 1 Z80—PIO

Zilog 公司生产的 Z80—PIO(Mostek MK3881)用来作为并行 I/O 端口控制器，它有两个可由用户编程的端口。Z80—PIO 两个完全独立的 8 位双向端口，每个口都有一对联络线来实现联络功能，所以这种芯片可以用来进行同步传送。

Z80—PIO 可以通过编程，以四种不同即：字节输出，字节输入，字节双向(只有 A 口)和位控工作方式。

字节输出也称为方式 0。它用来实现 CPU 经过数据总线写(传送)数据到外围设备。如果选择方式 0 工作，当进行写数据操作时，接口会发出联络信号 (ready)，用这个信号去通知外部设备，现在数据已可以利用了。在数据保持可用状态下，ready 信号始终保持高电平，直到收到从外部设备送来的选通 (strobe) 信号为止。

字节输入方式，也称为方式 1。若用 PIO 作为输入口工作，当 CPU 进行该操作时，PIO 将会送出一个 ready 信号通知外部设备，PIO 现在可以接受数据了。在外部设备送来选通 (strobe) 信号时，就把数据送入 PIO 的数据输入寄存器。

字节双向方式也称为方式 2。它来作为 8 位的双向 I/O 口。方式 2 要用四根联络线，由于这个原因，所以只有 A 口能工作在双向方式。(借用 B 口的一对联络线)。

位控方式，也称为方式 3。它用来作为反映外设的状态和对外设进行控制。方式 3 不用联络线，这种工作方式需要定义口的数据总线，哪些位是输入，哪些位是输出。所以，在程序中送完选择工作方式 3 的控制字之后，紧跟着要送入 I/O 选择控制字(即定义哪些位是输

入, 哪些位是输出)。

Z80—PIO 的引脚图如图 3—1 所示, 下面介绍各个引脚的功能。

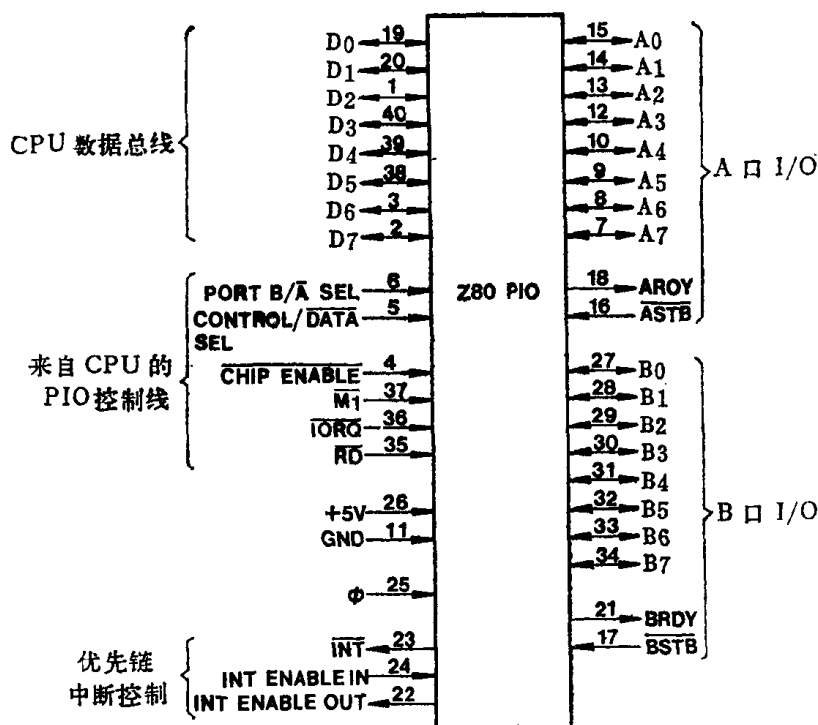


图3—1 Z80—PIO引脚

- D0—D7** 这些引脚连接到 Z80—CPU 的数据总线上, 它们是双向和三态的引脚。在 CPU 和 PIO 之间传送的控制字和数据都需从这些引脚上通过。
- B/ $\overline{A}$ SEL** 这是输入引脚, 用来选择 A 口或 B 口。当 $\overline{B/A}$ SEL 为低电平时, 选中 A 口; 而当它为高电平时, 选中 B 口, (这根引脚通常接 CPU 地址总线 A0)。
- C/ $\overline{D}$ SEL** 这是输入线, 用来选择 CPU 和 PIO 之间传送信息的类型。若这根引脚是低电平, 则告诉 PIO 从 Z80—CPU 数据总线上送来的信息是 I/O 数据, 若这根引脚是高电平, 则通知 PIO 送来的信息是要写入接口的控制字。数据或控制字写入哪个口则由 $\overline{B/A}$ SEL 来决定 (它通常是接地址总线的 A1)。
- $\overline{CE}$** 低电平有效的输入引脚。这根引脚用来作为片选线。当它为低电平时, 在 I/O 写周期, 允许 PIO 接受来自 CPU 的控制字或数据。而在 I/O 读周期, 会把 PIO 口的数据送到 CPU。
- $\overline{M1}$** 这根引脚用来使 PIO 和 CPU 同步。通常将它接到 CPU 的 $\overline{M1}$ 引脚。当它有效时, 表明 CPU 正处于 M1 机器周期。
- $\overline{IORQ}$** 向 Z80—CPU 请求输入/输出的引脚。它是使系统同步的另一根引脚, 通常连接到 CPU 的 $\overline{IORQ}$ 端。
- $\overline{RD}$** 低电平有效的输入引脚。当它有效时, 表明 Z80—CPU 处于读周期。
- IEI** 中断允许输入, 它是高电平有效。

IEO	中断允许输出，它是高电平有效。它和 IEI 一起构成中断优先链结构*。
INT	低电平有效，开漏极的输出引脚，用来向 CPU 请求中断。
A0~A7	三态双向引脚，是A口的输入/输出引脚。
ASTB	低电平有效输入引脚，由外部设备送来的选通A口的信号线。
ARDY	高电平有效的输出引脚。它有效，表明A口的数据寄存器准备就绪。
B0~B7	三态双向引脚，是B口的输入/输出引脚。
BSTB	低电平有效的输入引脚。它有效，是表示外部设备选通B口。
BRDY	高电平有效的输出引脚。它有效，表示B口的数据寄存器已准备就绪。

§ 2 Z80—SIO

Z80—SIO 是一种串行 I/O 接口芯片，用来直接与 CPU 连接。它与 PIO 相似，也是一种可以编程序的两个通道的接口芯片。然而，SIO 是以串行的方式传送数据，即每次传送一位。当然，在大多数情况下，并行传送数据速度比较快，但是在计算机和通讯设备之间传送数据时，往往喜欢采用串行传送的方式。这种方式，可以简化它们之间传输线的硬设备。即串行传送数据只要一对传输线，或一根电话线，或一个无线电通讯联络通道，甚至短距离的传送也喜欢用串行传送，更何况长距离的传输呢。Z80—SIO 是按照刚好能处理适当的串行数据规程设计的。像其它 Z80 系列芯片一样，它采用单电源和单相时钟的芯片。

除了与电源和与 CPU 总线的连接，两个通道是共同的以外，两个通道是互相没有联系的。和 PIO 一样，两个通道分别以 A 和 B 来标定。Z80—SIO 数据的传输率是从 0—550,000 位/秒。

接受和发送均有充分缓冲器。发送部分的寄存器是双重缓冲，另一方的接受寄存器是四重缓冲。

Z80—SIO 能异步操作（它的作用很像 UART，但它又具有 Z80 系统的特点）；二进制同步操作和 HDLC/IBM—SDLC 操作。SIO 具有 8 个调制—解调器来控制输入/输出，它还具有优先中断逻辑，能自动提供中断矢量字，且具有 CRC—16 和 CRC—CCIT(—0/—1) 校验。

工作在异步操作方式时，SIO 非常像一般的 UART，可以通过编程序来选择数据字长是 5，6，7 或 8 位，也同 UART 一样，在每传送一帧数据字的末尾有 1， $1\frac{1}{2}$ 或 2 位的停止位。CPU 发送数据时，没有这些停止信息，当送入 SIO 时，再由 SIO 插进去。和 UART 一样，SIO 在数据中还加入奇偶校验位（偶、奇，或无校验），用来判断帧出错或不出错。然而，SIO 还提供发生器和中断检测，时钟频率可以分别使用 $1\times$ 、 $16\times$ 、 $32\times$ 和 $64\times$ 数据的传输率。

图 3—2 是 Z80—SIO 的结构图。图 3—2(a) 是芯片的总体框图，而图 3—2(b) 是通道框图。

* 此处为译者所加。

输入侧的 8 根数据总线接 CPU 的数据总线，当工作在发送器时，用来接收 CPU 送来的数据。还有 6 根与 CPU 连接的控制信号线。在内部，有和外界不直接沟通的内部数据总线，两个通道逻辑，内部控制逻辑，还有中断控制逻辑和离散控制部分（包括调制—解调和其它

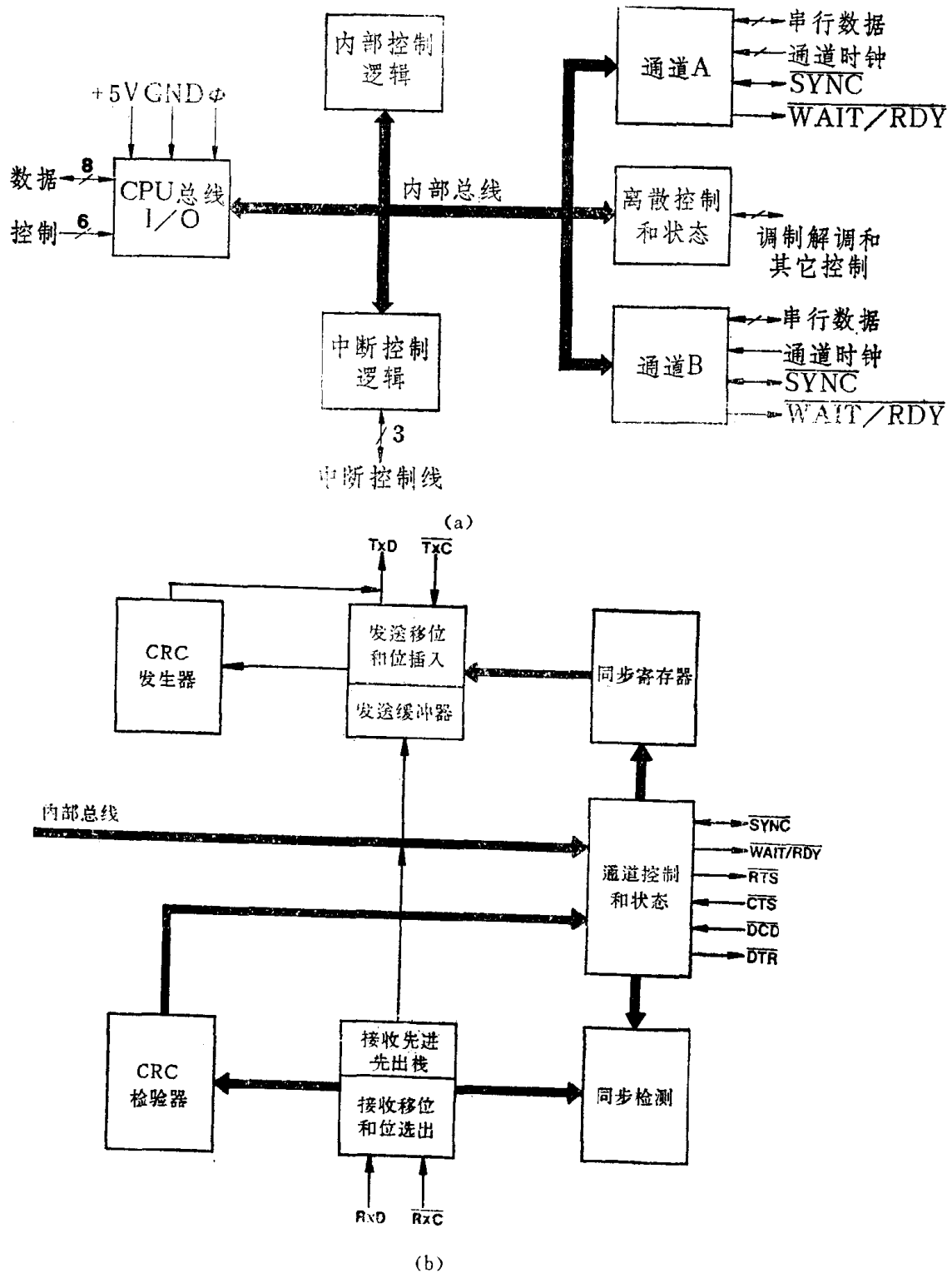


图3-2 Z80-PIO结构框图

(a) 内部总体框图

(b) 通道框图