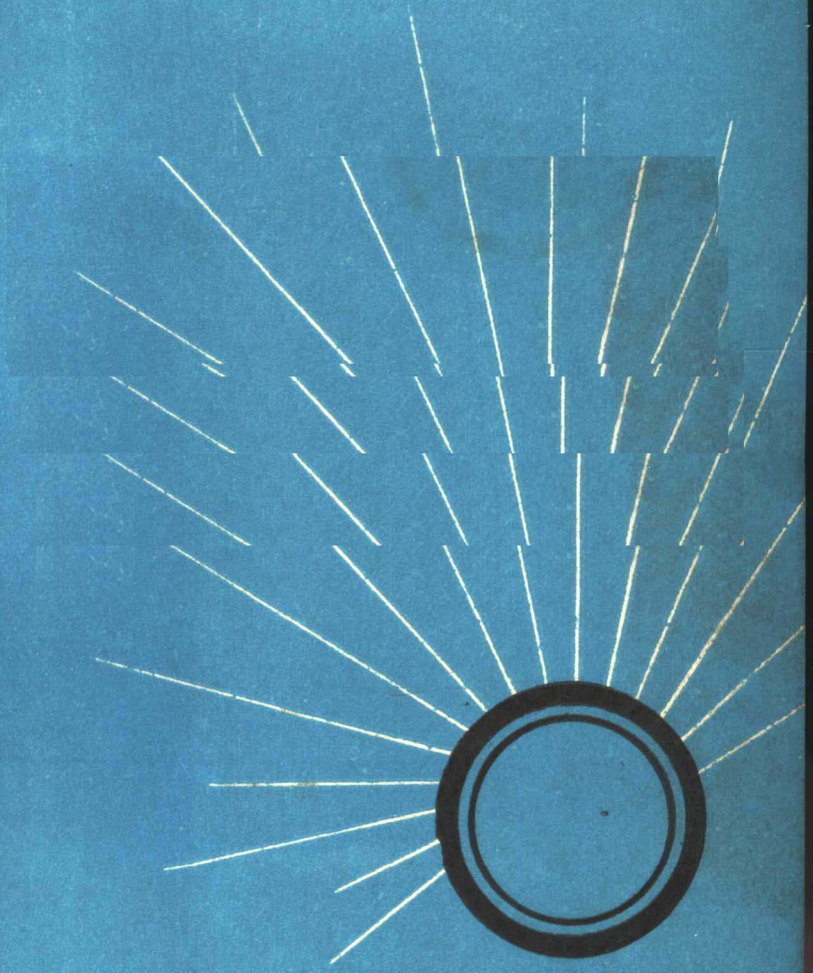
高等学校教材

接口与通信

黎康保　朱宏兴　编著



西北工业大学出版社

高等学校教材

接口与通信

黎康保　朱宏兴　编著

西北工业大学出版社

1989年11月　西安

内容提要

本书主要讲述计算机接口技术的基本概念、基本结构和组成原理。全书共有十二章，分为六个模块，它们是：①计算机的I/O结构组织；②存贮器的组织与通信，磁盘、磁带机；③数据通信与接口；④人机对话接口；⑤模拟信号接口技术；⑥多机互连接口技术。本书论述以硬件为主，硬件和软件相结合，着眼于应用。每章后面附有思考题。

本书可作为计算机科学与工程类各专业的教材，亦可作为自动控制、信息工程、数据通信、仪器仪表及检测、数控技术等学科专业的教材和参考书。同时也可作为有关工程技术人员的参考书。

高等学校教材

接口与通信

编著者　黎康保　朱宏兴

责任编辑　郑永安

责任校对　樊　力　钱伟峰

西北工业大学出版社出版

（西安市友谊西路127号）

陕西省新华书店发行

航空航天工业部〇一二基地印刷厂印装

ISBN 7-5612-0111-0/TP·28（课）

开本787×1092毫米　1/16　27.5印张　674千字

1989年11月第1版　1989年11月第1次印刷

印数1—4500册　定价：5.40元

前　言

电子计算机的飞速发展，特别是微型计算机的普及和广泛应用，使接口技术问题已成为十分重要、十分关键的问题。因为计算机功能的提高及强大功能的体现往往是由接口外围设备的能力和处理外界信息的能力表现出来的。我们知道，只有接口了键盘、显示器、打印机，计算机的处理能力才能得以显现；接口了软磁盘和硬磁盘，计算机工作者才可以采用虚拟技术来极大地扩充计算机的存贮空间；接口多个微型计算机组成分布式多机系统，通过多机并行运算可以达到亿次以上的计算速度和极高的运算能力；也只有将计算机接口了各种各样的自然界模拟信号，计算机的应用才推广到过程控制、测试等领域，实现了机电仪一体化，极大地推动生产力的发展；人们将计算机配上接口组成计算机网，共享信息资源，发展信息资源，使社会信息化，推动社会前进。

由于计算机的发展，特别微型计算机的发展和应用，要接口的设备和外界信息越来越多，越来越复杂，接口本身已不是一些逻辑电路的简单组合，而是强大的硬件和软件的综合技术，称之为“接口技术”。接口技术本身直接体现着计算机的体系结构，直接关系着计算机的推广和应用。目前“接口技术”很重要的作用就是要进行设备间的信息通信处理，解决计算机之间、计算机与外围设备之间的信息流通、信息控制与信息管理等问题。这就使“接口与通信”联系在一起了，它是目前十分重要、十分关键的技术领域，为国内外计算机应用学科所注重，并成为计算机应用学科的组成部分，是目前很热门的课题。

微型计算机与外围设备或各微型计算机之间进行信息通信处理，在简单的输入/输出中只用少数IC逻辑电路组成。在复杂的应用场合，则需应用专门设计的LSI接口芯片，它们可以初始化编程以实现各种各样的功能，提供数据信息通信通道，提供对数据通信的控制和管理，并在CPU的直接控制下运行。有的“接口与通信”芯片本身就是一个微型计算机系统。如IEEE-488总线的接口芯片Intel 8291/8292，MC68488（GPIA）适配器的内部就含有微处理器来组成总线的各种操作功能。有的芯片比通常的微处理器还要复杂的多，价格也昂贵的多，例如军标MIL-STD-1553B总线的接口芯片COM1553B就是这样的芯片。近年来出现了专门的I/O处理器芯片。如8089IOP，它包括两个独立的DMA输入/输出通道，是16位的处理机，标准时钟5MHz，通道传输率高达1.25兆字节/秒。它把处理器功能和直接存贮器存取（DMA）控制结合在一起，除了通常的DMA功能之外还能对数据进行翻译、比较、装配和拆卸，可设置多种结束数据传送的条件，具有经过优化处理的基本指令50条。

1978年将CPU、RAM、ROM和I/O集成于一个芯片的单片微机，体积大为缩小，价格更加低廉。如8位的Z8系列和MCS-51系列单片微机，有4K×8的内部ROM，有124和128个内部通用寄存器，有128K字节的寻址空间，片内集成有定时/计数器，并行端口和串行全双工端口。16位的单片微机如MC-96（8096）具有更高的性能，在单片上有16位的CPU，8KROM，232字节RAM，10位的A/D转换器，16位定时/计数器以及并行接口，全双工串行接口等。由单片微机组装起来的接口器件将使“接口与通信”技术的进步提高到一个崭新的阶段。

美国 I-EEE 计算机科学与工程协会在1983年提出建立“接口与通信”课程作为计算机科学与工程类专业的核心课程，并给出教学大纲。中国计算机学会教育与培训专业委员会在1985年12月提出在我国计算机科学与工程类专业建立《接口与通信》核心课程。

根据我国计算机科学与工程专业的基本教学要求和作者长期从事这一领域的实际科研、教学工作的经验和体会，参考 I-EEE 的教学大纲及中国计算机学会教育与培训专业委员会关于在我国计算机科学与工程类专业建立“接口与通信”核心课的精神撰编此书作为教材以及有关工程技术人员的参考书。本书以阐述基本计算机接口的硬件和软件的工作原理、接口方法为主。硬件软件相结合而以硬件为主，着眼于应用。学习中以掌握基本原理为重点，通过目前芯片实例深入掌握应用方法和技能。芯片在不断更新换代，而原理则是相对固定的。机型结合则以我国最流行的8位微机Z80和16位微机Intel 8086为主。

全书分为六个模块共十二章，每模块相对独立以利于取舍，每章末尾附有习题。六个模块是：

一、计算机的I/O结构组织（包括第一、二章的内容）。

二、存贮器的组织与通信，磁盘、磁带机（第三章内容）。

三、数据通信与接口（包括第四、五、六章）。

四、人机对话接口（第七章内容）。

五、模拟信号接口技术（包括第八、九、十和第十一章）。

六、多机互连接口技术（第十二章内容）。

本书除作为计算机科学与工程类专业教材外，还适用于自动控制与工业自动化、信息工程、数据通信、仪器仪表及检测、数控技术等各学科专业的教材和参考书。

全书编写大纲由黎康保提供，朱宏兴负责编写第一、二、三、十一章和附录Ⅴ。黎康保负责编写第四、五、六、七、八、九、十和第十二章以及书中的其他内容，并负责全书的统稿工作。全书插图由林树根同志绘制。

承蒙西安交通大学胡正家教授十分详细地审阅了本书，提出了许多极为宝贵的意见，对我们的教益至深；西北工业大学康继昌教授提供了他在国外学习期间的部分有关A/D、D/A转换器方面的笔记和产品参考资料对我们很有帮助，对此我们一并表示最衷心的感谢。此外，我们还要感谢大力支持和关心本书编写的西北工业大学韩兆轩教授。

黎文楼、张德英同志为本书作了大量的资料整理和文字工作，在此也深表感谢。

计算机科学技术发展十分迅速，作者受学识水平所限，书中难免会有许多缺点和错误之处，敬请批评指正。

编 者

1988年10月于西北工业大学

目　录

第一章　微处理器与系统结构

§1-1　8位微机结构

微处理器是数字电子系统中一个具有算术和逻辑运算功能的器件，它是微型计算机的核心部件——中央处理器（CPU）。通常典型微处理器的内部结构至少包括以下三个基本功能部件：

①寄存器阵列，包括通用寄存器和专用寄存器，用以暂时存放数据和地址；

②算术逻辑部件ALU，简称运算器，用来执行算术和逻辑运算；

③控制与定时部件，简称控制器，产生执行指令的各种控制信号。

一、简介

1974年美国Intel公司研制成功8位并行微处理器Intel 8080，随后又制造出8080的改进型产品Intel 8080A。8080A有16位地址总线，8位数据总线；内存寻址范围可达64KB；I/O寻址范围为256个I/O端口；具有一般中断INT和DMA功能。

8080A CPU由4个功能部件组成：寄存器阵列及地址控制逻辑；算术逻辑单元；指令寄存器及控制部件；双向三态数据总线缓冲器。

Intel 8080A系统中的CPU模块由三片电路组成：

(1) 8080A CPU；

(2) 8224时钟发生器驱动器（外接石英晶体）；

(3) 8228系统控制器。

8080A使用的电源种类较多，有+5伏、-5伏和+12伏三种。

同年，美国Motorola公司研制成8位并行MC6800微处理器，它与8080A微处理器并列为第二代微处理器的代表产品，其基本指令72条，寻址方式6种，指令操作总计197种；采用单一电源+5伏；中断功能4级，具有DMA能力。MC6800 CPU包括寄存器阵列、运算器和控制器三大部分。为了与外界交换数据，CPU通过数据缓冲器与双向数据总线D_7～D_0相连。地址通过输出缓冲器送到16位地址总线A_{15}～A_0。由于这两种总线与CPU相连的缓冲器都是三态的，所以实现DMA操作比较方便。在MC6800中，存贮器地址和I/O设备地址是编在一起的，即采用存贮器映象I/O寻址结构。

MC6800处理中断时，CPU内部寄存器的保护（入栈）或恢复（出栈）都是由片内硬件自动完成，这是MC6800的一个优点。

1976年，美国Zilog公司在Intel 8080A微处理器的基础上研制成功高性能的8位微处理器Z80，它在软件上对8080A是向上兼容的。

二、Z80 CPU

(一) 内部结构

Z80 CPU是一种8位的40条引脚的内部单总线结构的微处理器。它的内部结构如图1-1所示。

1．寄存器阵列

（1）通用寄存器组。Z80有两组通用寄存器：主寄存器B、C、D、E、H、L和备用寄存器B′、C′、D′、E′、H′、L′，也可以成对当作16位寄存器使用，它们是BC、DE、HL和BC′、DE′、HL′。

（2）累加器和标志寄存器。Z80 CPU有两个8位累加器（A和A′）和两个8位的标志寄存器（F和F′）。累加器A保存8位算术和逻辑运算结果，标志寄存器F指明8位或16位操作结果的特征。

（3）程序计数器PC（16位）。

（4）堆栈指示器SP（16位）。

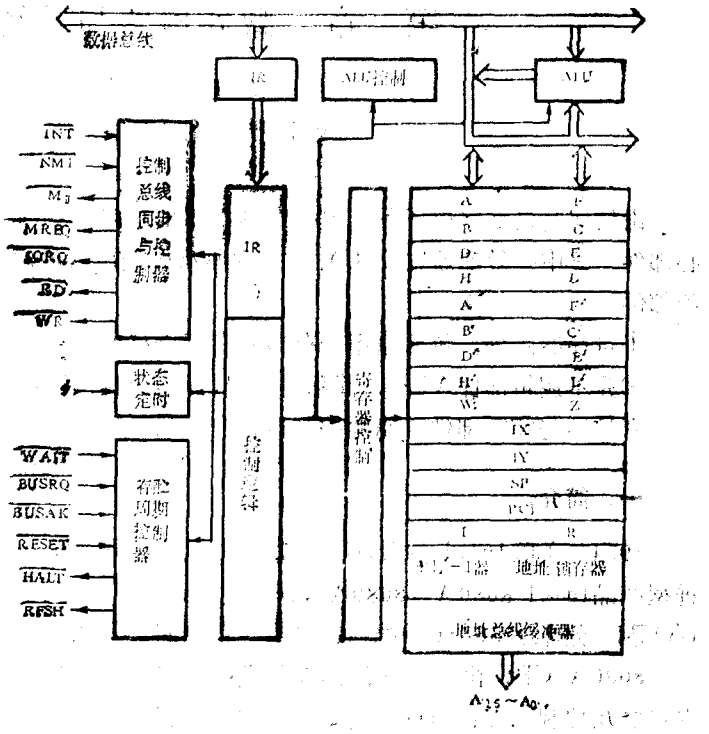


图 1-1 Z80 CPU内部结构框图

（5）变址寄存器。Z80 CPU有两个独立的16位变址寄存器IX和IY，用来存放变址寻址方式中用的变址值。变址指令中有一个附加字节，用来指明相对此变址值的位移量以形成操作数的有效地址。

（6）中断矢量寄存器I（中断页地址寄存器）。当Z80 CPU以方式2响应中断时，中断向量寄存器I用来提供16位中断矢量地址指针的高8位地址，即中断矢量地址指针的页地址（若以256个字节为一页，64K内存就分为256页）。

（7）动态RAM刷新寄存器。为了防止动态存贮器存贮电容中的信息被泄漏就需要定时（一般为2ms）对动态存贮器进行刷新。Z80是利用取指周期的后两个T状态（此时CPU对指令进行译码和执行内部操作，不使用存贮器）来刷新的。每一个取指周期对一部分存贮器进行刷新，以保证在2ms内对整个64K内存刷新一遍。每次刷新内存中的一行，这个行地址就由R寄存器中的低7位提供，而在每刷新一行后，R的内容自动加1，以指向下一行。

（8）两个不可编程的暂存寄存器W、Z。

2．算术逻辑单元ALU

ALU是微处理器的核心部件，它主要包括一个加法器、两个独立的8位累加器、两个8位的标志寄存器、若干个暂存寄存器以及十进制调整电路。ALU能执行加、减；逻辑与、逻辑或、逻辑异或、比较；位置位、位复位、位测试；加1、减1以及算术和逻辑的左移、右移或循环移位等。

3．控制器

控制器包括指令寄存器、指令译码器和定时与控制电路共三个部分。每条指令从存贮器取出后便送到指令寄存器，然后再送到指令译码器译码，通过定时与控制电路，在规定的时刻产生和提供相应的控制信号，以执行所需要的操作。

Z80 CPU引脚图如图1-2所示。

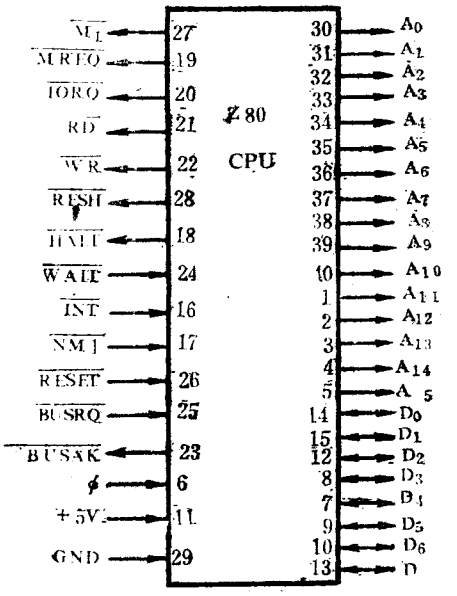


图 1-2 Z80 CPU引脚图

（二）Z80 CPU芯片引脚说明

（1）A_{15}～A_0 地址总线（单向、三态输出）。A_{15}～A_0组成16位地址总线，用来为CPU与内存贮器及I/O设备之间交换数据提供地址。当I/O寻址时，使用地址总线的低8位。在刷新期间，低7位地址由R寄存器提供。

（2）D_7～D_0 数据总线(双向、三态）。D_7～D_0组成8位双向数据总线，用于CPU和存贮器或I/O设备之间进行数据交换。

（3）$\overline{M_1}$ 机器周期(输出,低电平有效)。$\overline{M_1}$指明现行机器周期是取指令操作码周期。当指令操作码是两个字节时，则每个取操作码周期都发出$\overline{M_1}$信号，这种双字节指令的操作码经常用CBH、DDH、EDH、FDH开始。

（4）$\overline{MREQ}$ 存贮器请求（三态输出，低电平有效）。它表明地址总线上保持有一个供存贮器读或写操作的有效地址。

（5）$\overline{IORQ}$ 输入输出请求（三态输出，低电平有效）。它表明地址总线的低8位中保持有一个供I/O读或写的有效的I/O端口地址。$\overline{M_1}$与$\overline{IORQ}$信号一起表明一中断响应周期，通知外设把中断矢量放到数据总线上。

（6）$\overline{RD}$ 读控制信号（三态输出，低电平有效）。它表明CPU要从存贮器或一个I/O装置读入数据。

（7）$\overline{WR}$ 写控制信号（三态输出，低电平有效）。它表明CPU要向存贮器或一个I/O装置写入数据。

（8）$\overline{RFSH}$ 刷新（输出，低电平有效）。它表明地址总线的低7位是动态存贮器的刷新地址。它与$\overline{MREQ}$信号一起用于刷新动态寄存器。

（9）$\overline{HALT}$ 暂停（输出，低电平有效）。它表明CPU已经执行了一条HALT指令，CPU处于暂停状态。在暂停期间，CPU执行空操作指令，以保持存贮器刷新操作照常进行。直到CPU接受不可屏蔽或可屏蔽中断时，才脱离暂停状态，重新往下执行程序。

（10）$\overline{WAIT}$ 等待（输入，低电平有效）。它告诉CPU所寻址的存贮器和I/O装置尚未准备好数据传送，CPU插入等待状态，直至此信号变为无效。

（11）$\overline{INT}$ 中断请求（输入，低电平有效）。由I/O设备产生的可屏蔽中断请求信号。

（12）NMI 非屏蔽中断请求（输入，低电平有效）。它比$\overline{INT}$优先权高，即不管中断允许触发器IFF的状态，NMI总是在现行指令执行完毕时得到响应。NMI信号自动迫使Z80

CPU转向0066H存贮单元开始执行中断服务程序。

(13) $\overline{\text{RESET}}$ 复位（输入，低电平有效）。它迫使程序计数器和指令寄存器清零，并且使CPU初始化：

①使中断允许触发器IFF复位；

②置I寄存器为00H；

③置R寄存器为00H；

④置CPU为中断方式0。

在复位期间，地址总线和数据总线都进入高阻状态，所有控制信号都处于无效状态，此时停止刷新操作。

(14) $\overline{\text{BUSRQ}}$ 总线请求（输入，低电平有效）。当$\overline{\text{BUSRQ}}$有效时，CPU在现行机器周期结束时立即响应，CPU将所有三态总线处于高阻状态，而由DMA控制器控制这些总线。它用于存贮器与快速的外设之间直接传送信息。

(15) $\overline{\text{BUSAK}}$ 总线响应（输出，低电平有效）。它告诉总线请求装置，CPU的三态总线已处于高阻状态，此时该外部设备可以控制使用总线。

(16) ϕ 单相系统时钟。Z80时钟频率为2.5MHz；Z80A时钟频率为4MHz；Z80B时钟频率为6MHz。

(17) 电源 V_{cc} = + 5伏。

(三) Z80 CPU的主要特点：

(1) Z80 CPU有两组通用寄存器、两个累加器和两个标志寄存器。可用交换指令对两组寄存器进行快速交换信息，这样可以提高数据处理操作的效率。

(2) 增加了两个16位变址寄存器IX和IY，使Z80具有比8080A更加灵活的多种寻址方式。

(3) 增加了一个中断矢量寄存器I，用来形成矢量中断的入口地址。

(4) 增加了一个刷新寄存器R，用来控制MOS动态RAM的刷新，相应地增加了一个控制信号RFSH。

(5) Z80有两个中断请求信号 $\overline{\text{INT}}$和$\overline{\text{NMI}}$，比8080A具有更强的中断处理能力。

(6) Z80采用单一电源+5V供电；又采用单相时钟，工作频率可提高到4MHz以上，使Z80具有明显的速度优势。

(7) Z80指令系统不仅包括了8080A的全部78条指令，而且又增加了80条指令，因而使处理能力大大增强，程序长度大为缩短。

(8) Z80 CPU把8080A模块中三片LSI电路的大部分功能都集成在Z80芯片上，而且还扩大了其他功能。

(9) Z80的标志寄存器有六个状态标志。

1977年Intel公司在8080A基础上研制成功了高性能的8位微处理器8085A。它的指令系统与8080A微处理器兼容，但功能更强，速度更快，集成度更高，而且允许有三片集成电路：8080A CPU、8155（RAM）和8355/8755（ROM/EPROM）组成最小规模的MCS-85微型计算机系统。

§1-2 Z80的中断系统和I/O结构

一、输入/输出的寻址方式

通常有两种I/O接口结构：一种是专用I/O指令结构，另一种是存贮器映象I/O结构。因此，与之对应的就有两种I/O结构寻址方式。

（一）专用的I/O指令寻址方式

专用的I/O指令寻址方式也称端口寻址方式，它有以下三个特点：

(1) I/O设备的地址空间与存贮器地址空间是独立的，即I/O接口地址不占用存贮器的地址空间。Z80 CPU用地址总线的低8位对I/O设备寻址，因此寻址范围为256。

(2) CPU要用专门的I/O指令来实现与外设端口之间的数据传送。

(3) CPU对I/O设备的定时控制是用I/O读写控制信号 $\overline{IORQ}$、$\overline{RD}$、$\overline{WR}$ 来进行定时控制的。

因为一个外设不仅有数据寄存器，还有状态寄存器和控制寄存器，它们各需要一个端口才能加以区分，故一个外设往往需要几个端口地址。

（二）存贮器映象I/O寻址方式

存贮器映象I/O寻址又称为存贮器对应I/O寻址方式，它也有三个特点：

(1) I/O接口与存贮器共用同一地址空间，即指定存贮器地址空间内的一个区域供I/O设备使用。故每一个I/O设备占用存贮器空间的一个地址。

一般设定I/O端口占用地址线 $A_{15}=1$ 的32K地址空间，而存贮器地址占用 $A_{15}=0$ 的32K地址空间。因此，当 $A_{15}=1$ 时，I/O接口工作；当 $A_{15}=0$ 时，存贮器工作。

(2) CPU可用全部的存贮器指令来实现与外设端口之间的数据传送。

(3) CPU可用存贮器读写控制信号MEMR、MEMW，并经 A_{15} 状态控制来确定是存贮器还是I/O设备，如图1-3所示。

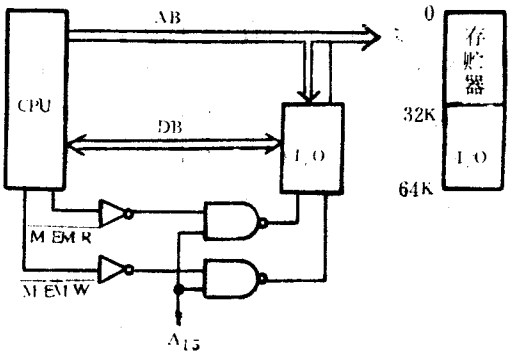


图 1-3 存贮器映象I/O结构

（三）两种I/O结构及寻址方式之比较

1．端口结构寻址的优点是：

① I/O设备不占用存贮器地址空间；

② 编制程序时，易把I/O指令与存贮器指令分开；

③ I/O指令一般只需两个字节，所以寻址速度较快。此方式的缺点是需要一些专门的I/O指令。

2．存贮器映象I/O结构寻址的优点是：

① 扩大了I/O设备的寻址范围（可超过256个I/O端口）；

② CPU对外设的操作可使用全部的存贮器指令，故指令多，使用灵活、方便。此方式的缺点是I/O设备占用了内存地址，使工作内存容量减小。

二、Z80的I/O指令和时序

(一) Z80的I/O指令

Z80输入输出指令共有24条。

1. 直接寻址的I/O指令

IN　　A, (n)；　　A←(n)

OUT　　(n), A；　　(n)←A

其中，n为所要寻址的一字节端口地址。执行指令时，端口地址出现在地址总线的低8位(A_7～A_0)上。输入指令的功能是把地址为n的端口的数据或状态寄存器的内容送至累加器A；输出指令的功能是把累加器A的内容送至地址为n的端口的数据寄存器或控制寄存器中去。

2. C寄存器间接寻址的I/O指令

IN　　r, (c)；　　r←(c)

OUT　　(c), r；　　(c)←r

输入指令是由c寄存器的内容构成8位地址，寻址所需要的外设端口，把这端口的内容送至CPU中的任一内部寄存器A、B、C、D、E、H或L中；输出指令是把CPU中某一寄存器的内容送至所寻址的外设端口。因此，这类指令可使外设端口直接与任一CPU内部寄存器交换信息。

3. 数据块输入输出指令

(1) 单步增减址的I/O指令

INI　；　(HL)←(C)，　HL←HL+1，B←B−1

IND　；　(HL)←(C)，　HL←HL−1，B←B−1

OUTI　；　(C)←(HL)，　HL←HL+1，B←B−1

OUTD　；　(C)←(HL)，　HL←HL−1，B←B−1

这类指令是在由寄存器对HL所指定的内存单元和由寄存器C所选定的I/O端口之间传送数据，而用寄存器B作为数据字节计数器。此指令执行一次输入或输出一个数据，且自动修改地址指针，计数器减1，以便实现由同一外设把一组数据送入到内存中的一个相邻区域，或从内存的一个区域，把一组数据通过同一外设输出。

(2) 成组的数据块I/O指令

INIR　；　(HL)←(C)，HL←HL+1，B←B−1，直至B=0

INDR　；　(HL)←(C)，HL←HL−1，B←B−1，直至B=0

OTIR　；　(C)←(HL)，HL←HL+1，B←B−1，直至B=0

OTDR　；　(C)←(HL)，　HL←HL−1，B←B−1，直至B=0

其功能与单个传送的情况相似，只是执行这类指令时，若B−1≠0，则自动重复执行数据块传送，直至B−1=0而结束。

上述指令中数据块的最大容量为256个字节，它与数据块传送或数据块搜索指令是有区别的（最大容量为64K字节）。

(二) Z80 CPU的I/O时序

Z80 CPU的I/O时序如图1-4所示。

Z80 CPU在执行I/O指令期间产生I/O周期。I/O过程是以地址有效开始的，在T_1的上升沿后，所选择的I/O端口地址有效（出现在地址总线的低8位A_7～A_0）。在T_2的上升沿后，$\overline{IORQ}$和RD或WR信号都变为有效。为了有充足的时间对I/O端口的地址进行译码寻址，在T_2结束后，将自动地插入一个等待时态T_W。

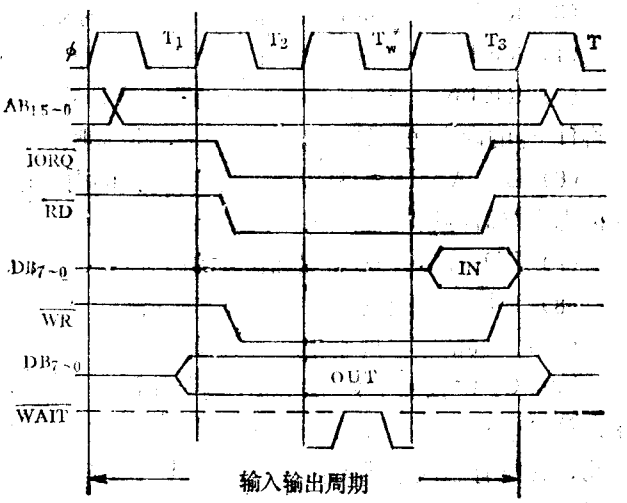


图 1-4 Z80 CPU I/O时序

在输入周期，CPU在T_3时态的下降沿采样数据总线，获取由外设输入的数据，同时$\overline{IORQ}$和RD变为高电平无效，输入周期即告结束。

在输出周期，CPU要输出的数据，自T_1时态的下降沿后就出现在数据总线上，并且在整个输出周期保持不变。在T_2上升沿后，IORQ和WR信号同时低电平有效，表示允许进行I/O写操作。在T_3时态前半周结束前，把数据总线上保持的数据输出到指定的外设中去。

因为在存贮器读写时，$\overline{RD}$和$\overline{WR}$也同样有效，所以不能单独用$\overline{RD}$或$\overline{WR}$信号作为I/O选通信号，必须同时用MREQ或$\overline{IORQ}$来区别是存贮器的读写操作，还是I/O的读写操作。

三、Z80的中断系统

（一）Z80中断系统的组成

Z80中断系统主要包括两个部分：

（1）具有两条中断请求线：非屏蔽中断输入线$\overline{NMI}$和可屏蔽中断输入线$\overline{INT}$。两者均为低电平有效。

（2）两个中断允许触发器IFF_1和IFF_2。

IFF_1用于控制开中断或关中断；IFF_2用于暂存IFF_1的状态。当$IFF_1=1$时，允许可屏蔽中断；当$IFF_1=0$时，禁止可屏蔽中断。当CPU复位时，将迫使IFF_1和IFF_2都处于“0”状态，即禁止中断状态。程序员可在任何时候用一条开中断指令EI来开放中断。

当CPU接受可屏蔽中断后，IFF_1和IFF_2自动地被硬件置“0”，禁止中断请求，直到程序员用新的EI指令来开放中断为止（通常在EI指令后是一条返回指令，为了保证断点的正确返回，必须在返回指令执行完成以后，才允许开放中断）。

当Z80 CPU进行不可屏蔽中断处理时，IFF_2用来保存IFF_1的状态。当$\overline{NMI}$请求被CPU接收时，为了防止发生新的中断，IFF_1将被置“0”。为了保存CPU响应$\overline{NMI}$以前IFF_1的状态，就设置IFF_2来暂存IFF_1的状态，待CPU处理$\overline{NMI}$完毕之后，自动将IFF_2的状态送回IFF_1，使中断系统恢复到原来的状态。

（二）Z80供中断系统使用的专用指令

(1) EI——允许中断指令，使$IFF_1=1$来开放可屏蔽中断。

(2) DI——禁止中断指令，使$IFF_1=0$来封锁可屏蔽中断。

(3) RST P——重新启动指令，供机器内部使用。

(4) RETI——中断返回指令，用于表示现行中断服务程序已经结束。

(5) RETN——不可屏蔽中断返回指令，除了表示不可屏蔽中断服务程序已经结束，还将IFF_2的内容送回IFF_1，使中断系统恢复原来的状态。

(6) LD A, I——将中断矢量寄存器I的内容送入累加器A，同时也将IFF_2的内容送入标志寄存器的P/V位。

(7) LD I, A——将累加器A的内容送入中断矢量寄存器I。

(8) IMn——设置可屏蔽中断方式指令。

(三) Z80的不可屏蔽中断方式

Z80 CPU按照下列优先顺序响应外部请求：总线请求→不可屏蔽中断→可屏蔽中断。因此，在没有总线请求的情况下，CPU在执行现行指令的最后一个机器周期的最后一个T时态后采样NMI，若为低电平，则CPU立即予以响应。在NMI响应周期，CPU相当于执行一条RST指令，将PC内容压入堆栈，再将地址0066H置入PC，然后转入NMI服务程序的入口地址开始执行程序。其流程图如图1-5所示。

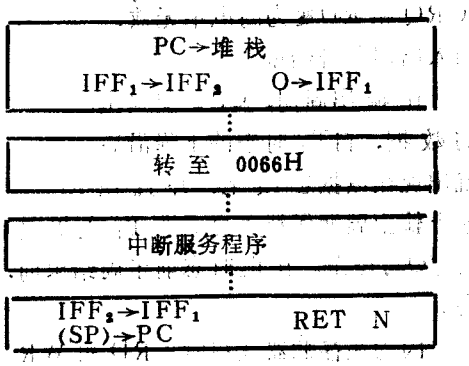


图 1-5 不可屏蔽中断的流程图

当CPU响应NMI时，PC→堆栈，$IFF_1\rightarrow IFF_2$，$0\rightarrow IFF_1$和0066H→PC等均由硬件完成。不需请求中断的外设提供RST指令。同样，执行RET N指令的操作也由硬件来完成。

不可屏蔽中断的时序如图1-6所示。第一个机器周期是不可屏蔽中断请求响应周期，这类似于取指周期，但是并不将数据总线上的数据送入指令寄存器，而是在CPU内部自动形成一条转移到0066H的RST单字节调用指令。第二和第三个机器周期是存贮器写周期，即分别将PC_H和PC_L压入堆栈。

(四) Z80的可屏蔽中断方式

在没有总线请求和不可屏蔽中断的情况下，只有当CPU的中断允许触发器$IFF_1=1$时，CPU才能响应可屏蔽中断INT。可以由程序员用两条指令EI或DI来确定开中断或关中断。CPU响应可屏蔽中断有三种方式：方式0、方式1、方式2，这可用中断方式指令IM0、IM1、IM2来选择。CPU复位时，自动把中断方式置为方式0。

1. 可屏蔽中断方式0

中断方式0是Z80为兼容8080而设置的。Z80 CPU以方式0响应中断请求的过程如下：

(1) 外设发出中断请求$\overline{INT}$，CPU在执行现行指令的最后一个机器周期的最后一个T时态采样$\overline{INT}$，若这时$IFF_1=1$，则在这条指令执行完后，CPU接受中断并进入中断响应周期。

(2) 在中断响应周期，CPU向I/O发出中断响应信号（$\overline{M_1}$和$\overline{IORQ}$同时低电平有效），

以通知该外设其中断请求已被响应。

(3) 该外设接口收到$\overline{M_1}$和$\overline{IORQ}$同时低电平有效信号后，将一条单字节的RST P指令代码（或三字节的CALL nn指令代码，但很少使用）送到数据总线上。

(4) CPU在中断响应周期T_3的上升沿把RST P的指令代码送入指令寄存器。

(5) CPU在结束中断响应周期后，转入执行RST P指令。即把当前的PC(即程序的断点)值压栈，把RST P指令中0页的8个入口之一的地址送入PC。在RST P指令所对应的8个入口地址段存放着8个不同的中断服务程序。因此，根据外设提供的不同RST P指令的代码，就可以直接转到相应的中断服务程序首址。

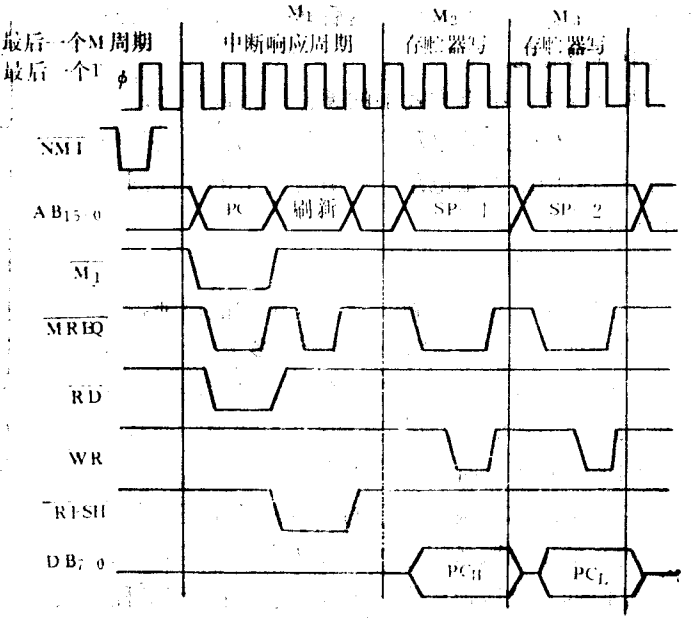


图 1-6 不可屏蔽中断时序图

可见，要使CPU按方式0响应中断请求，除了要对CPU设置方式0外，还要求请求中断的外设接口电路在CPU响应中断时能将RST P的指令代码送到数据总线，CPU在执行这一指令之后，控制程序跳转到0页的8个入口地址之一中去。

中断方式0通常只允许有8级中断，每一级只有8个单元，但中断服务程序通常总会超过8个字节。因此，可在每级的最初三个地址单元里安排一条三字节的JP nn指令代码，再转移到相应中断服务程序的首址。中断方式0的流程图如图1-7所示。

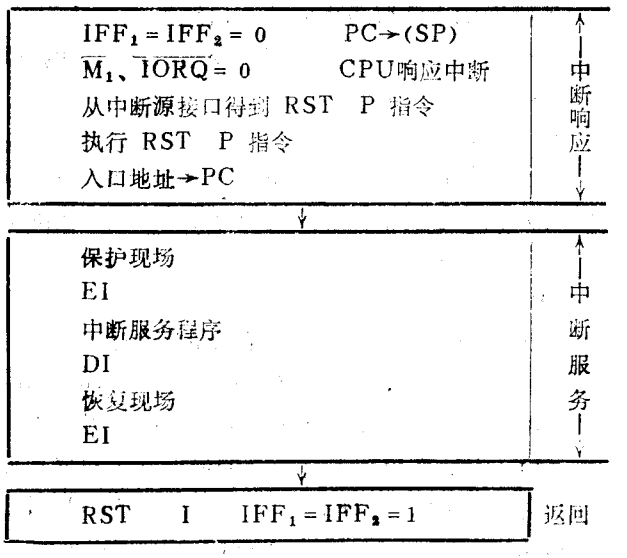


图 1-7 中断方式0的流程图

2. 可屏蔽中断方式1

中断方式1也是Z80为兼容8080而设置的。方式1的中断响应过程基本上同方式0，所不同的是它与不可屏蔽中断一样，不需要由外设接口提供一条RST P指令，而是在CPU内部自动形成一条转移到0038H的RST 38单字节调用指令，控制程序到0038H单元去执行中断服务程