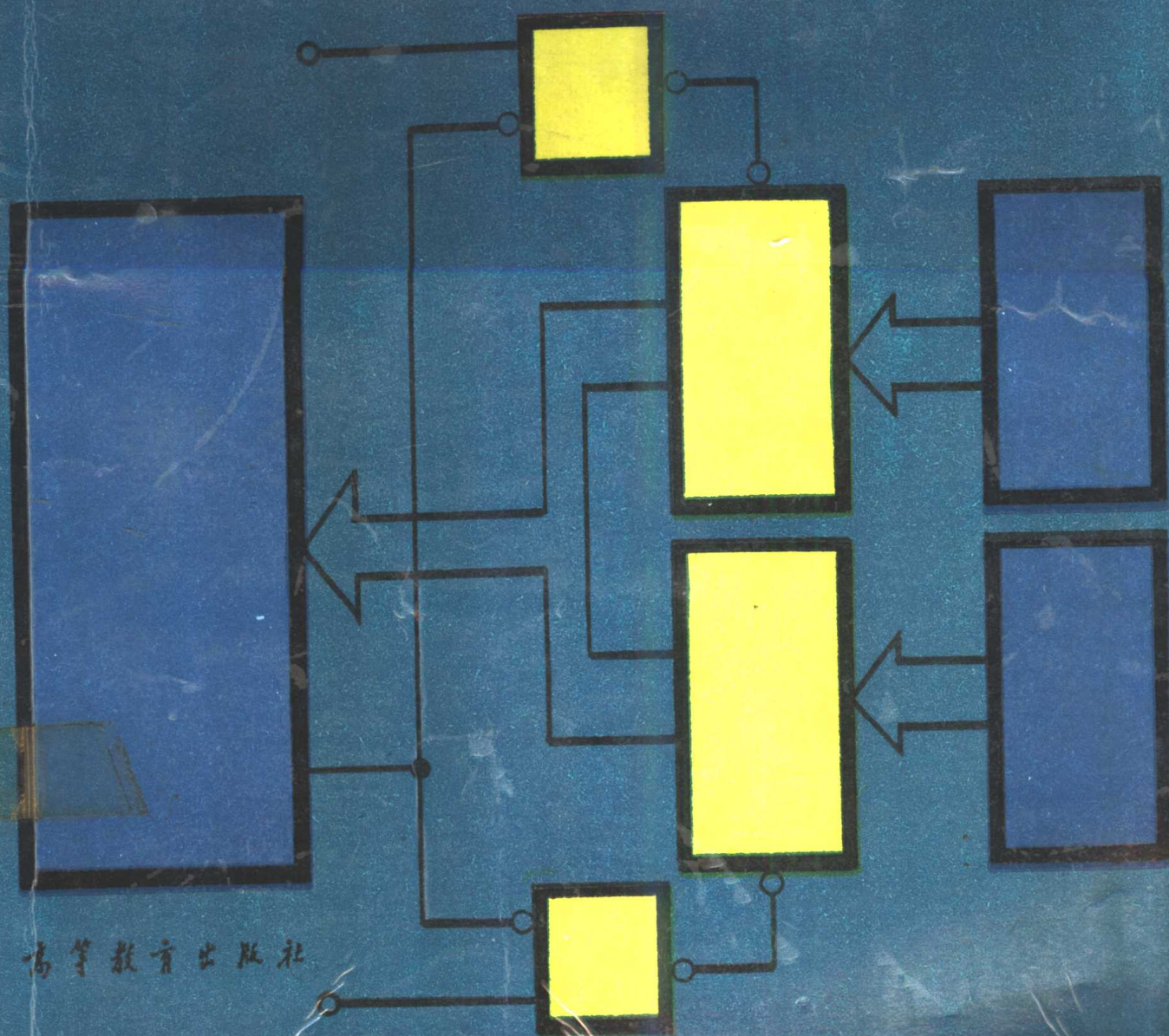


高等学校教学用书



微型计算机 接口技术

● 黎康保 朱宏兴 编



高等教育出版社

高等学校教学用书

微型计算机接口技术

黎康保 编
朱宏兴

高等教育出版社

本书是根据我国微型计算机应用开发的需要和编者长期从事这方面科研、教学的经验体会编写而成,全书以阐明最基本的微型计算机接口硬件和软件工作原理、接口方法为主,并结合实际应用。主要内容有:微型计算机结构与接口基础,存储器的通信接口,总线,数据通信接口,人机通信接口,模拟信号接口技术,接口连接的抗干扰问题。全书共分九章,每章后附有习题。

本书可供各部门从事微型计算机应用开发的科技工作者参考,也可作为大专院校及电大、职大等的选修课教材或专业人员培训教材。

高等学校教学用书

微型计算机接口技术

黎康保 编
朱宏兴

*

高等教育出版社出版

高等教育出版社照排中心照排

新华书店总店北京科技发行所发行

天津新华印刷厂印装

*

开本787×1092 1/16 印张19.5 字数450 000

1990年10月第1版 1990年10月第1次印刷

印数0001~3 160

ISBN 7-04-002908-1/TP·68

定价4.25元

前 言

微型计算机借助接口技术来处理各种各样的外界信息,同外界交换信息和传递信息。如何接口外界信息,如何高效高速与外界交换信息和传递信息,是微处理器本身设计中的重大课题。接口技术已成为微机应用系统的关键和重要组成部分,而且它直接体现着微型计算机的能力和功能力。随着微型计算机的空前普及和深入应用,对接口技术的要求就更加迫切,内容更加多样,更加丰富。对接口技术的研究已成为广大计算机科学与工程领域的学者、工程技术专家的热门课题。美国 I-EEE 计算机科学与工程协会和我国计算机学会教育和培训专业委员会提出在大学的计算机科学与工程系建立“接口与通信”课程,并作为专业的核心课程之一。各行各业从事微型计算机开发应用的工作者无疑都应该学习和掌握这门技术知识。

本书是根据我国微型计算机开发应用的需要和编者长时间从事这方面科研、教学工作的经验体会,参考计算机科学与工程专业的“接口与通信”课程的有关精神和美国 I-EEE 的建议大纲编写而成。全书以阐明最基本的微型计算机接口硬件和软件工作原理、接口方法为主,结合应用实际,照顾必要的知识面;以硬为主,硬软结合;着眼于应用。学习中以搞通原理为主,学习芯片的应用在于巩固和掌握基本原理和基本方法。因为芯片是在不断更新换代,而原理则是相对固定的。

本书主要内容有:

1. 微机结构与接口基础。特别介绍了近年来兴起的单片微型计算机及接口基础。
2. 存储器的通信接口。
3. 总线。特别介绍了目前广为流行应用的 STD 总线。
4. 数据通信接口。介绍数据通信的一些必要知识及串行接口和并行接口。
5. 人机通信接口。
6. 模拟信号接口技术。
7. 接口连接的抗干扰问题。

全书分为九章,每章后附有习题。

本书可作为各部门,各专业从事微型计算机应用开发的工程技术人员,大、专院校师生的参考书。特别适合于学时要求在 60 学时左右的本科、专科、电大、职大和专业人员培训班的教材。学习本课程要求“微型计算机原理”、“汇编语言程序设计”等作为先行课。

本书承蒙湖南大学黄义源副教授作了精心、细致的审核,提出了大量极其宝贵的意见,我们表示衷心的感谢。

黎文楼、张德英同志为本书作了很多文字和资料的整理工作,付出辛勤的劳动,在此表示衷心的感谢。

本书在编撰过程中韩兆轩教授、宋玉田副教授曾给我们很多的协助和鼓励,我们表示衷心感谢。本书编写大纲由黎康保提供,并编写第一章 §1-1、§1-4,第四章,第五章,第六章,

第七章、第八章以及附录一、二、三,朱宏兴编写第一章 §1-2、§1-3,第二章,第三章,第九章及附录四、五,最后由黎康保作了全书的统稿工作,全书插图由林树根绘制。

由于微型计算机接口技术这一应用学科发展迅速,应用广泛,内容庞杂,涉及知识面很广,以及编者水平和工作范围所限,不当和错谬之处在所难免,恳请读者、专家批评指正。

编 者

一九八八年十二月于西北工业大学

目 录

第一章 接口技术与微型计算机结构

.....	(1)
§ 1-1 概述	(1)
§ 1-2 输入/输出和中断系统	(2)
输入/输出的寻址方式	(2)
数据传送的控制方式	(4)
§ 1-3 16位微型机结构	(14)
一、Intel 8086	(14)
二、Z8000	(17)
三、M68000	(19)
§ 1-4 单片微型计算机	(20)
一、概述	(20)
二、MCS-51 系列单片微型计算机	(22)
三、Z8 系列单片微型计算机	(26)
四、MC 6801 单片微型计算机	(34)
五、单片微机的掉电保护	(36)
习题	(38)
第二章 存贮器、磁带、磁盘接口	(39)
§ 2-1 存贮器接口	(39)
一、存贮器模块的组成	(39)
二、存贮器地址空间的选择方式	(39)
三、存贮器的扩充	(41)
四、RAM 与 CPU 的连接	(46)
五、随机存贮器连接中常见故障及判别方法	(49)
§ 2-2 音频盒式磁带机接口	(50)
一、磁带记录的标准	(50)
二、磁带机与 CPU 的接口	(51)
§ 2-3 磁盘机接口	(53)
一、软磁盘接口	(53)
二、硬磁盘接口	(62)
习题	(64)
第三章 微型计算机的总线	(65)
§ 3-1 概述	(65)

§ 3-2 S-100 总线和 MULTIBUS 总线	(66)
一、S-100 总线	(66)
二、MULTIBUS 总线	(68)
§ 3-3 STD 总线	(70)
一、STD 总线简介	(70)
二、STD 总线模块化工业控制机的应用系统模式	(74)
§ 3-4 IEEE-488 总线	(77)
§ 3-5 RS-232C 接口	(81)
习题	(84)
第四章 串行通信接口	(85)
§ 4-1 串行通信基本概念和规程	(85)
一、基本概念	(85)
二、串行通信规程及 CRC 校验	(86)
§ 4-2 串行接口的实现	(89)
§ 4-3 串行接口器件	(92)
一、Intel 8251A	(92)
二、Z80-SIO	(98)
习题	(114)
第五章 并行通信接口	(115)
§ 5-1 并行通信接口原理	(115)
一、直接传送并行接口	(115)
二、异步数据传送握手	(115)
三、并行 I/O 端口结构	(117)
§ 5-2 并行输入/输出接口器件	(118)
一、MC 6820/6821 PIA	(118)
二、Intel 8255 A(PPI)	(123)
三、Z80 PIO	(128)
§ 5-3 IEEE-488 总线接口器件	(134)
一、Intel 8921(GPIB)	(134)

1. MC 68488 (GPIA)	(136)
习题	(141)
第六章 人机通信接口	(149)
§ 6-1 LED 显示器及与微机接口	(143)
一、用硬件段译码的多位显示接口	(144)
二、硬件段、位译码、软件多位显示	(146)
三、全软件段、位译码和多位显示	(146)
四、存储器周期窃取显示	(149)
五、点阵式显示器	(150)
§ 6-2 键盘及接口	(151)
一、键盘连接及颤抖问题	(151)
二、键盘状态识别软件	(152)
三、键译码及命令转移程序	(156)
§ 6-3 专用显示器/键盘接口器件	(160)
一、Intel 8279 键盘显示控制器	(160)
二、键盘编码器	(168)
§ 6-4 控制面板(控制台)接口	(169)
一、显示	(169)
二、开关与键盘	(169)
三、故障报警	(171)
四、控制面板实例	(172)
§ 6-5 CRT 接口	(173)
一、CRT 显示器接口结构组成	(173)
二、家用电视机作 CRT 显示终端的接口	(177)
三、CRT 图形显示	(177)
§ 6-6 操纵杆、超声笔和光笔	(179)
一、操纵杆	(179)
二、超声笔	(180)
三、光笔	(180)
§ 6-7 打印机接口	(182)
一、打印机的组成原理	(182)
二、打印机的参数	(183)
三、微计算机与打印机接口实例	(185)
四、汉字打印	(186)
习题	(187)

第七章 模拟信号接口及基本部件	(189)
§ 7-1 模拟信号与计算机接口	(189)
§ 7-2 实时时钟	(191)
一、Intel 8253 可编程定时/计数器(PIT)	(191)
二、Z80-CTC	(195)
§ 7-3 集成运算放大器	(199)
一、跟随器与求和器	(199)
二、积分器	(203)
三、模拟电压比较器	(203)
四、集成运算放大器的主要技术指标	(205)
§ 7-4 产生过零脉冲	(206)
§ 7-5 模拟开关	(206)
一、双极晶体管开关	(207)
二、结型场效应管(J-FET)开关	(208)
三、MOS 场效应管开关	(209)
四、多路开关	(209)
§ 7-6 采样/保持电路	(210)
一、原理与参数	(210)
二、集成电路采样保持器	(211)
§ 7-7 交流信号变换	(212)
一、相位不变只幅值变化的交流信号变换	(212)
二、相敏交流信号变换	(213)
三、闭路相敏全波解调器	(213)
§ 7-8 微弱信号变换	(215)
一、微弱信号的特殊要求	(215)
二、一般运算放大器不能放大微弱信号	(216)
三、微弱信号放大器	(216)
习题	(219)

第八章 模拟/数字转换技术及接口	(220)
§ 8-1 D/A 转换及接口	(220)
一、D/A 转换器结构原理	(220)
二、CMOS 网络电阻开关	(222)
三、双极晶体管电流开关网络	(223)

四、双极性问题及 D/A 芯片	(224)	三、交流电压峰值的采集和测试	(266)
五、交流信号 D/A 转换器	(225)	习题	(268)
六、D/A 转换器与微机接口及应用编程	(226)	第九章 屏蔽、接地及连接技术	(269)
§ 8-2 A/D 转换技术	(231)	§ 9-1 屏蔽与接地技术	(269)
一、逐位比较式 A/D 转换器	(231)	一、屏蔽	(269)
二、双斜坡积分 A/D 转换器	(233)	二、接地	(271)
三、高速 A/D 转换器	(235)	§ 9-2 传输线与连接技术	(275)
四、集成 A/D 芯片的特性及调整	(237)	一、点到点的传输	(275)
§ 8-3 A/D 转换器与 CPU 接口		二、机器内板的连接	(280)
及应用程序设计	(240)	三、机壳间的连接	(282)
一、硬件接口	(240)	§ 9-3 常见干扰的抑制	(282)
二、用汇编语言程序设计	(241)	一、电源电压的脉动干扰	(282)
三、用软件控制的 A/D 转换器接口	(249)	二、开关、继电器等的脉动干扰	(283)
四、用高级语言对 A/D、D/A 应用编程	(256)	三、各种机械触点开关的抖动干扰	(285)
§ 8-4 用 DMA 控制器接口快速 A/D 转换器	(261)	四、安装、布线等考虑不周引起的干扰	(285)
§ 8-5 交流信号的 A/D 转换及接口	(264)	习题	(286)
一、交流信号周期(频率)的采集、测试接口	(264)	附录一 ASCII 码表	(288)
二、交流相角(可控硅导通角)测试接口	(265)	附录二 Z80 指令系统	(289)
		附录三 8080/8085 指令系统	(293)
		附录四 S-100 总线的信号及其定义	(295)
		附录五 MULTIBUS 总线的引脚功能	(297)
		参考文献	(300)

第一章 接口技术与微型计算机结构

§1-1 概 述

1971 年 4 位微机问世至 1981 年 32 位微机诞生, 产品已经历四代, 由 4 位微机发展至 8 位微机, 现在 16 位微机已成熟, 32 位微机在发展、完善, 64 位微机也已研究出来。微机的功能在迅速发展, 主要有两个方面: ① 在数据处理能力和处理速度上的发展, CPU 本身已打破冯·诺曼结构, 一个微处理器由多个传统的 CPU 功能结构构成, 结构面向高级语言, 面向多任务操作系统, 用重叠并行操作, 流水作业, 把大型机、巨型机的一些技术移植入微机。② 接口外界能力方面的发展。计算机功能的提高及强大功能的体现往往要由接口外围设备的能力和外界信息的能力表现出来。例如只有接口了键盘、显示器、打印机, 计算机的功能才能得以显现; 接口了软磁盘和硬磁盘, 可以采用虚拟技术来极大地扩充计算机的存贮空间; 可以通过接口多个微型机来达到亿万次的计算速度和极高的运算能力; 例如接口了自然界的模拟信号计算机的应用才推广到控制、测试领域等等。前者是微型计算机结构组成和体系结构中要研究的课题, 后者是本书要讨论的课题。

由于微型计算机的发展和应用, 要接口的设备和要接口的外界信息越来越多, 越来越复杂, 而且接口本身直接体现着微机的体系结构, 直接关系着微型计算机的推广应用, 它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”。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重要组成部分。美国 I-EEE 计算机科学与工程协会及我国计算机学会计算机教育与培训专业委员会将计算机的“接口与通信”列为计算机科学与工程专业的核心课程之一。

“接口技术”研究的领域大体上可概括为:

1. 计算机与存贮器的通信接口, 如存贮器扩展接口, 软磁盘机接口, 硬磁盘机接口, 磁带机接口, 直接存贮器通信 DMA 接口等等。

2. 人机通信接口, 包括数码显示器、键盘连接接口, 控制台(板)的设计和接口, CRT 显示器接口, 光笔输入、语音输入、打印机接口等。

3. 模拟信号接口, 计算机对模拟信号处理早在 1946 年第一台数字计算机出现以前就作了很高级的应用。第一次世界大战就应用模拟计算机来控制火炮; 第二次世界大战中, 高射炮、航空武器、舰艇武器、海岸炮等的控制系统(指挥仪)都应用了计算机, 计算机对模拟信号的处理达到十分高级的程度。数字计算机的出现, 特别是今天微型计算机的发展, 数字计算机对模拟信号的处理已远不只于军事上的应用, 更重要的是应用于工业过程控制、检测, 用于实现机电仪一体化, 用于国民经济, 用于社会组成的各领域, 用于人们的日常生活中, 成为推动社会前进的巨大力量。

4. 多机通信接口, 多机连接打破了计算机传统的体系结构, 极大地提高了计算机的能力,

使社会进入信息时代的高级阶段。

微型计算机与外围设备接口进行信息通信处理,在简单的输入输出中只用少数 IC 逻辑电路组成,对复杂的应用场合,则需应用专门设计的 LSI 接口芯片,它们可以初始化编程为各种功能方式,提供数据信息的通信通路,提供对数据通信的控制和管理,并在 CPU 的直接控制下运行。

随着微型计算机的发展,出现了 I/O 处理机,传输 I/O 数据高速高效,使 I/O 接口功能提高到新阶段。I/O 接口处理机有自己的 CPU,去掉了主 CPU 对 I/O 的干预,使主 CPU 和 I/O 设备之间的关系由直接管理转化为间接管理的不关系,和 CPU 并行工作,这就使 I/O 设备和 CPU 的功能都获充分发挥和提高。8089 IOP 就是这样的单片通用输入输出处理器,它包括两个独立的 DMA 输入/输出通道,每个通道都有自己的寄存器组,根据各自的优先权,两个通道交替工作。8089 IOP 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。IOP 与 CPU 之间以存储器为基础进行通信,把处理器功能和直接存储器存取(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。大约有 50 种基本指令,且都经过优化处理。

1978 年将 CPU、RAM、ROM 和 I/O 集成于一个芯片出现单片的微机系统,体积大为减小,价格低廉。如 8 位的 Z8 系列和 MCS-51 系列单片机,有 4KB 的内部 ROM,有 124 和 128 个内部通用寄存器,具有 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 位单片机如 MC-96(8096)具有更高性能,在单片上有 16 位 CPU,8K ROM,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。在微型计算机的外围设备(如 CRT、打印机、绘图仪等)上装上单片机,将使这些设备在操作上具有多样性,具有编辑和逻辑处理能力,使各种功能大大提高,使外围设备智能化,并能与主微机进行多机通信等。这些将使计算机的外围设备的水平提高到一个新的阶段。

§1-2 输入/输出和中断系统

一个微型计算机系统通常带有多个输入/输出设备,它们通过各自的接口与总线相连。中央处理器和这些设备通信是通过输入/输出端口进行寻址来实现的。

一、输入/输出的寻址方式

通常有两种 I/O 寻址方式:存储器映像 I/O 寻址方式和专用 I/O 指令寻址方式。

(一) 存储器映像 I/O 寻址方式

存储器映像 I/O 寻址又称为存储器对应 I/O 寻址方式。I/O 接口与存储器共用一个地址空间,即指定存储器地址空间内的一个区域供 I/O 设备使用。一般设定 I/O 端口占用地址线 $A_{15}=1$ 的 32K 空间内,而存储器占用 $A_{15}=0$ 的 32K 地址空间。因此,中央处理器可用存储器读、写控制信号 $\overline{MEMR}$ 、 $\overline{MEMW}$,并经 A_{15} 状态控制来确定是存储器还是 I/O 设备,如图 1-2-1 所示。

存储器映像 I/O 寻址方式的优点是:所有存储器指令都可以用来实现中央处理器与外设端口之间的数据传送。而其缺点是:第一, I/O 设备占用了内存单元,使内存容量减小。第二,存储器指令通常要求三个字节,其中两个字节用来对 I/O 接口进行寻址,译码所需硬件比较

复杂。第三，存储器指令比 I/O 指令的执行时间要长。第四，用存储器指令来处理 I/O 时，在程序清单中不易区别，给程序设计和调试带来了一定的困难。

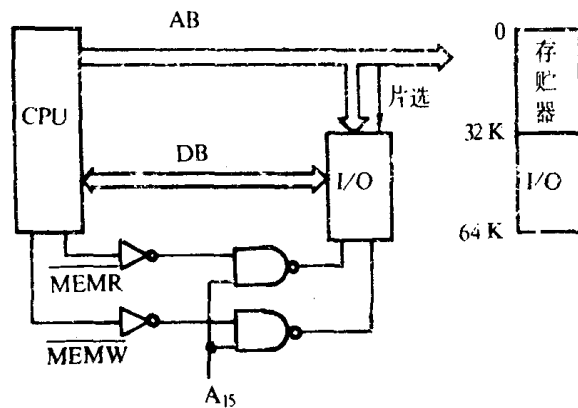


图 1-2-1 存储器映像 I/O 结构

(二) 专用 I/O 指令

专用 I/O 指令寻址方式又称端口寻址方式。这种寻址方式，中央处理器访问 I/O 接口必须使用专用的 I/O 指令。Z80 CPU 就是采用这种寻址方式的。Z80 的 I/O 指令有：

1. 直接寻址的 I/O 指令

IN A, (n)

OUT (n), A

其中 n 为所要寻址的一字节的端口地址，执行指令时，它出现在地址总线的低 8 位 ($A_7 \sim A_0$) 上。输入指令把地址为 n 的端口数据或状态寄存器的内容送至累加器 A；输出指令把累加器 A 的内容送至地址为 n 的端口的数据或控制寄存器中去。

2. 用寄存器 C 间接寻址的 I/O 指令

IN r, (C)

OUT (C), r

输入指令的功能是由 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；输出指令的功能是把 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) 单个的数据 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, 以便实现由同一个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

INDR

OTIR

OTDR

其功能与单个传送的情况相似, 只是执行这类指令时, 若 B 减 1 的结果不为零, 则自动重复执行数据块传送, 直至 B-1 的结果为零而结束。

专用 I/O 指令寻址方式的优点是: 第一, 只用单字节低 8 位进行 I/O 接口寻址, 译码所需的硬件比较少。第二, I/O 指令执行时间较短。第三, 专用的 I/O 指令与存储器指令书写标志明显不同, 有利于程序设计和调试。而其缺点是: 第一, 在进行算术逻辑运算时, 输入和输出的数据都必须经过累加器 A 暂存, 因此, 处理能力不如存储器映象 I/O 寻址方式。第二, 由于采用了专用的 I/O 周期和专用的 I/O 控制线, 从而增加了 CPU 本身控制逻辑的复杂性。

二、数据传送的控制方式

一个微机系统常带有多种外部设备, 由于它们的工作速度极不相同, 因此就产生了一个复杂的定时问题, 即 CPU 何时应当从某外设取得数据, 而又何时才能向某外设送出数据, 这就涉及到 CPU 同外设(通过接口)间如何传送信息的问题。通常有三种数据传送的控制方式: 程序控制方式、中断方式和直接存储器存取(DMA)方式。

(一) 程序控制方式

这类传送是在程序控制下进行信息交换, 它又分为无条件传送(同步传送)和条件传送(查询传送)两类。

1. 无条件传送

这种传送主要用于外部设备各种动作的时间是固定和已知的情况。因为 CPU 在传送信息时, 确认外设已经作好了准备, 所以不必查询外设的状态。在输入时, 就给输入指令, 在输出时, 就给输出指令。因此, 这种传送的优点是程序简单, 软件和硬件很省。值得注意的是, 输入时, 要保证输入的数据确实已经准备好; 输出时, 外设已将上次送出的数据取走, 可以接收新的数据。否则就可能读入不正确的数据, 或外设收到的是不完全的数据。

2. 条件传送

一般情况下, 外设状态和 CPU 的工作要求不同步, CPU 必须先检测每个外设的状态, 根据它们所处的状态来实现传送信息。因此, 在相应的接口电路里, 除了应具备一般的数据缓冲器外, 还要设置状态标志位, 用来反映外设处理数据的准备情况。输入时, 若数据已准备好, 则将标志位置位; 输出时, 若外设已取走数据, 也把标志位置位。一般的过程是:

- ① 从外设接口读入状态字。
- ② 检查状态字中的准备就绪标志位, 判断能否传送数据。
- ③ 如果外设未就绪, 则重复循环①、②步, 直到该外设准备就绪, 且 CPU 取到置位

标志为止。

i CPU 从外设读取数据或把数据写入到外设,同时把外设的就绪标志位复位。

下面用 Z80 指令系统分别写出了输入与输出的查询部分程序:

查询输入程序

```
TEST: IN    A, (STATUS)
      AND    80H
      JR     Z, TEST
      IN     A, (DATA)
```

查询输出程序

```
TEST: IN    A, (STATUS)
      BIT    7, A
      JR     Z, TEST
      LD     A, (STORE)
      OUT    (DATA), A
```

其中 STATUS 和 DATA 分别为状态端口和数据端口的符号地址; STORE 为要输出的数据的内存符号地址。

(二) 中断控制方式

在查询传送方式中, CPU 要不断地询问外设, 当外设没有准备好, CPU 就要等待, 不能干别的操作, 而且许多外设的速度是较低的, 它们输入或输出一个数据的速度是很慢的, 这样就浪费了 CPU 的许多时间。

为了提高 CPU 的效率, CPU 在输入或输出数据时就先执行一条指令, 发出启动外设工作的命令, 然后 CPU 就继续执行主程序。当 I/O 装置准备就绪时, 由外设(中断源)向 CPU 发出中断请求, CPU 就暂停执行原来的主程序(即实现中断), 转去执行输入或输出操作(即中断服务), 待输入输出操作完成后即返回, CPU 继续执行原来的主程序。这样就大大地提高了 CPU 的效率, 而且允许 CPU 与多个外设同时工作。下面讨论 Z80 的中断系统。

1. 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 $\overline{\text{NMI}}$ 和可屏蔽中断输入线 $\overline{\text{INT}}$, 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 $\text{IFF}_1 = 1$ 时, 允许中断; 当 $\text{IFF}_1 = 0$ 时, 禁止中断。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 EI 指令来开放中断, EI 指令使 IFF_1 和 IFF_2 都处于开中断状态。当 CPU 接受中断后, IFF_1 和 IFF_2 自动地被硬件置 0, 禁止中断, 直到程序员用新的 EI 指令来开放中断为止。

当 Z80 CPU 进行非屏蔽中断处理时, IFF_2 用来保存 IFF_1 的状态。当 $\overline{\text{NMI}}$ 请求被 CPU 接受时, 为了防止发生新的中断, IFF_1 将被置 0, 为了保存 CPU 响应 $\overline{\text{NMI}}$ 以前 IFF_1 的状态, 就设置 IFF_2 来暂存 IFF_1 的状态。待 CPU 处理 $\overline{\text{NMI}}$ 完毕之后, 自动地将 IFF_2 的状态送回 IFF_1 , 使中断系统恢复原来的状态。

2. Z80 供中断系统使用的专用指令

EI —— 允许中断指令, 使 $IFF_1 = 1$ 来开放可屏蔽中断.

DI —— 禁止中断指令, 使 $IFF_1 = 0$ 来封锁可屏蔽中断.

RETI —— 中断返回指令, 用它来表示现行中断服务程序已经结束.

RETN —— 非屏蔽中断返回指令, 除了表示非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.

LD A, I —— 它将中断矢量寄存器 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.

LD I, A —— 它将累加器 A 的内容送入中断矢量寄存器 I.

IMn —— 设置可屏蔽中断方式指令. 有三种可屏蔽中断方式, 当 n 为 0、1 或 2 时分别表示方式 0、方式 1 或方式 2.

3. Z80 的不可屏蔽中断方式

Z80 CPU 按照总线请求、不可屏蔽中断、可屏蔽中断优先顺序响应外部请求. 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, 采样 $\overline{NMI}$, 若为低电平有效, 则 CPU 立即予以响应. 在 $\overline{NMI}$ 响应周期, CPU 将 PC 内容压入堆栈, 再将地址 0066H 置入 PC, 即转入 $\overline{NMI}$ 服务程序的首地址 0066H 单元开始执行程序, 其流程图如图 1-2-2 所示.

4. Z80 的可屏蔽中断方式

在没有总线请求和不可屏蔽中断请求情况下, 即只有当 CPU 的中断允许触发器 $IFF_1 = 1$ 时, CPU 才能响应可屏蔽中断 $\overline{INT}$.

(1) 可屏蔽中断方式 0

当 CPU 复位时, 自动处于中断方式 0. 可屏蔽中断方式 0 也可用一条中断方式指令 $IM0$ 来设置. Z80 CPU 以方式 0 响应中断请求的过程如下:

① 外设发生中断请求 $\overline{INT}$, 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 接受中断并进入响应周期.

② 在中断响应周期, CPU 向 I/O 口发出中断响应信号 ($\overline{M_1}$ 和 $\overline{I/O RQ}$ 信号同时低电平有效), 以通知该外设, 它的中断请求已被响应.

③ 该外设接口收到响应信号后, 将一条单字节的 $RST P$ 指令代码送到数据总线.

④ CPU 在中断响应周期把 $RST P$ 指令的代码送入指令寄存器.

⑤ CPU 在结束中断响应周期后, 执行 $RST P$ 指令, 把当前的 PC (即程序的断点) 值压栈, 同时把 $RST P$ 指令中 0 页的 8 个入口 (00H、08H、10H、18H、20H、28H、30H 和 38H) 之一的地址送入 PC.

中断方式 0 通常只允许有 8 级中断, 每一级只有 8 个单元, 但通常中断服务程序总会超过 8 个字节, 因此, 可在每级的最初三个单元内安排一条三字节的 $JP nn$ 指令代码, 再转移到相应的中断服务程序的首址 nn . 中断方式 0 的流程图如图 1-2-3 所示.

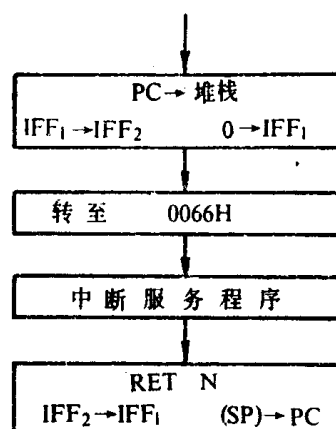


图 1-2-2 不可屏蔽中断流程图

(2) 可屏蔽中断方式 1

可用一条中断方式指令 IM1 来设置中断方式1,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 指令,而是在CPU 内部自动形成一条转移到0038H 的RST 单字节调用指令,控制程序到0038H 单元去执行中断服务程序,若有多个中断源时,就必须从0038H 单元为起始的存贮区域内存放一段排队程序,用查询方式来识别中断源以及确定它们的优先权,转去执行优先级最高的中断服务程序。

可见,中断方式1 的硬件电路最简单,一是不要中断硬件排队电路,二是外设的接口电路不需要配备中断矢量电路,中断方式1 的流程图如图1-2-4 所示。

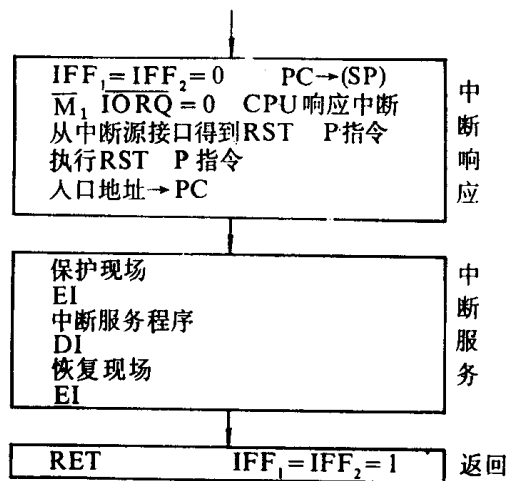


图 1-2-3 中断方式0 的流程图

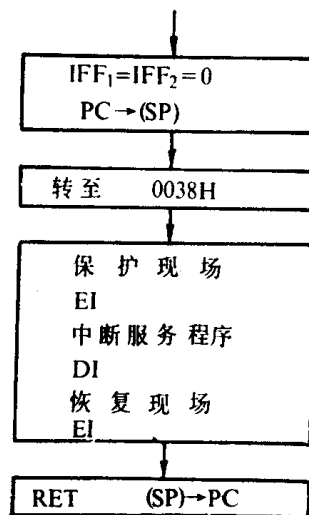


图 1-2-4 中断方式1 的流程图

(3) 可屏蔽中断方式 2

可用一条中断方式指令 IM2 来设置中断方式2,方式2 是三种可屏蔽中断方式中功能最强、最灵活的一种,方式2 的中断服务程序可放在内存的任意区段,并把它们的入口地址编成一个地址表,每一服务程序入口地址在该表内占两个单元,最多可有128 个中断源,就形成一个256 个存贮单元的地址表,它占用一页的内存区,这个地址表可以用程序放置于内存的任意页内,在中断响应时,中断优先级硬件排队电路选中当前中断请求优先级最高的外设接口,由该接口提供页内地址(低8 位),与CPU 的中断矢量寄存器所提供的页面地址(高8 位)组合起来形成地址表指针,以便从入口地址表中获得所需要的中断服务的入口地址。

可用如下指令组分别将地址表指针送入CPU 和I/O 接口电路的中断矢量寄存器。

LD A,n₁

LD I,A

LD A,n₂

OUT (PORT),A

PORT 是接口中的控制寄存器的端口地址。

中断方式2 的流程图如图1-2-5 所示。

(三) 直接存储器存取(DMA 方式)

利用中断方式来传送数据,对于高速I/O设备,以及成组地交换数据的场合就显得速度太慢了,所以希望能用硬件在外设与内存间直接进行数据交换而不通过CPU,因而可大大提高传送速度,这就是DMA方式。

1. Z80-DMA的操作功能:

Z80-DMA有以下三种操作类型:

(1) 传送。数据块可以从一个外设写到另一个外设;或者从主存贮器的一个区域写到另一个区域;或者从外设写到存贮器或者从存贮器写到外设。

(2) 检索。在单一端口内(存贮器或外部设备)查找一个特殊字节,该字节可按位加以屏蔽。

(3) 既检索又传送。在没有找到匹配的字节时,数据块一直在传送,找到匹配的字节后,传送工作停止,并同时产生中断请求信号。

Z80-DMA又有三种操作方式:单字节方式、字组方式、连续方式。

(1) 单字节方式。每次只传送一个字节,在两次传送操作中间CPU收回总线的控制权,要第二次传送时,DMA必须再次请求总线。

(2) 字组方式。在CPU响应DMA请求后,DMA操作一直进行到该端口的就绪信号RDY变为无效或字组传送结束或已找到匹配的字节为止。

(3) 连续方式。在CPU响应DMA请求后,DMA操作就连续进行,直到程序规定的块长度或数据块检索后才恢复CPU对总线的控制。连续操作期间,如果RDY信号变为无效时,DMA只是简单地处于暂停状态,直到RDY变成有效时,再继续操作。

2. Z80-DMA的结构

Z80-DMA的结构如图1-2-6所示,除 $\overline{\text{CE}}/\overline{\text{WAIT}}$ 引脚外,其它引脚与Z80 CPU相连时都不需要加缓冲器。

DMA的内部数据总线与系统的数据总线相连接,对于源口与目标口,有各自的起始地址寄存器(16位)和地址计数器(16位)。地址计数器用来存放即将传送的字节数,并且当地址变化时,自动进行加1或减1。

控制逻辑由CPU的编程设定,控制寄存器存放各种操作的控制信号,状态寄存器存放Z80-DMA的各种状态信息。

匹配字节寄存器用来存放待查找的字节,即比较数据,掩码寄存器用来存放掩码字节,掩码字节允许对匹配字节的某些位作比较,如果被检索的数据字与匹配字节一致,则匹配电路将使状态寄存器中相应特征位置“1”。

脉冲产生器可以定时发出脉冲,使外部设备知道已经传送了多少字节。

中断和总线请求电路包括控制寄存器、优先编码逻辑电路以及中断矢量寄存器。根据控制寄存器规定的中断条件,优先编码电路产生 $\overline{\text{INT}}$ 输出或对 $\overline{\text{BUSRQ}}$ 进行选择,实现中断优先和通道优先控制。

3. 编程

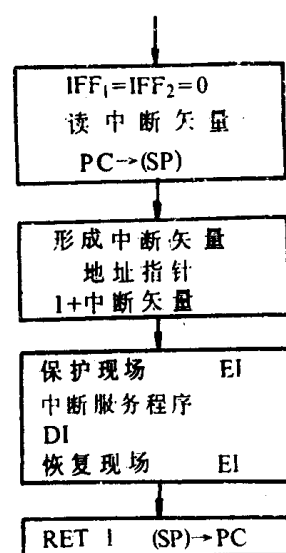


图1-2-5 中断方式
2的流程图

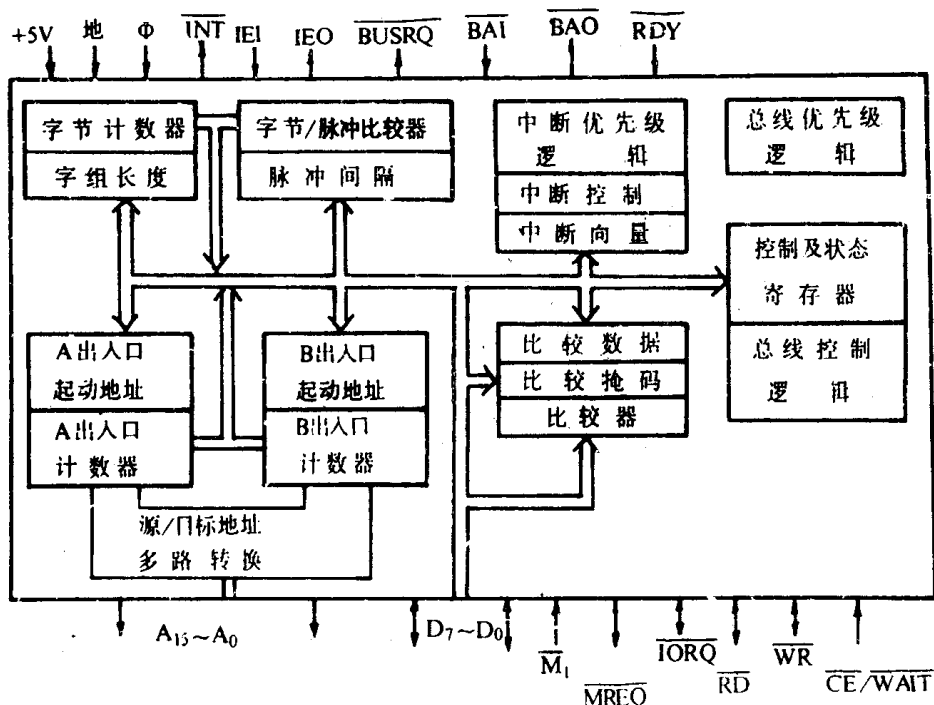


图 1-2-6 Z80-DMA 内部方框图

(1) Z80-DMA 命令字

Z80-DMA 有两个可编程的基本状态：允许状态和禁止状态。Z80-DMA 在接通电源或通过别的方法复位时，便自动置于禁止状态。此时，既不能起动总线请求，也不能传送数据，但是允许 CPU 对 DMA 进行编程或读状态字。一旦命令字写入其中时，也自动置于禁止状态。此状态维持到 CPU 给 DMA 一个允许命令为止。

CPU 必须在进行数据传送或检索之前，对 Z80-DMA 进行编程。它是通过输出命令在数据总线上给 DMA 送一连串 8 位的命令字。这些命令字包含有装入控制寄存器和其它寄存器的各种控制信息和参数。由于 Z80-DMA 的功能比较灵活，内部寄存器又多，因此编程比较复杂。其命令字的结构有下列两个特点：

1) 在命令字中规定了特征位，用于对内部寄存器进行寻址。Z80-DMA 共有六种不同的命令字，分为两组。第一组有两种命令，它们的名称为 1A 和 1B。第二组有四种命令，它们的名称为 2A、2B、2C、2D。命令字的 D_7 用来规定组别， $D_7=0$ 为第一组命令字， $D_7=1$ 为第二组命令字。而 D_1 和 D_0 用来规定命令字的名称。命令字写入 DMA 时，DMA 将根据这些特征位来选择对应的内部寄存器。

2) 在某些命令字中，规定某些位置“1”时，下一个紧跟写入的字节就被写入某一特定的内部寄存器。这些特定的内部寄存器有中断矢量寄存器、中断控制寄存器、比较寄存器、掩码寄存器、字组长度的寄存器、出/入口起始地址寄存器、脉冲寄存器、定时寄存器等。

Z80-DMA 的命令字如下：

① 命令字节 1A

命令字节 1A 用来设定 DMA 操作类型、DMA 操作的字节数、DMA 传送的方向和 A