

微处理机接口技术

(EE—3402)

[美]—小安德鲁 C. 斯坦嘎德 著

合肥工业大学微型计算机应用研究所 译

合肥工业大学学报编辑部

1984.5.

微 处 理 机 接 口 技 术

(EE—3402)

[美] 小安德鲁 C.斯坦嘎德著

合肥工业大学微型计算机应用研究所 译

合 肥 工 业 大 学 学 报 编 辑 部

1984.5.

译者的话

本书系根据 Professor Andrew C. Staugaard, Jr. 所著《Microprocessor Interfacing》(1982年版本)一书翻译的。

全书共分十一单元。作者以 Motorola 6800系列微处理机及其外围器件为题材,对该系列的各种芯片和运用作了详尽地论述。书中不仅介绍了八位的6800、6809和十六位的68000等微处理机的寄存器结构、寻址方式及其指令系统,还着重以PIA、ACIA、PTM等接口芯片为例,详细叙述了I/O控制、A/D及D/A转换,串行通信,并/串转换等技术。并且讨论了外部I/O设备和各种存贮器,包括大容量外存。特别是用了一个单元的篇幅介绍了6809微处理机,用了三个单元的篇幅论述了68000微处理机,以及它们的指令系统、接口线、外围器件,支持和应用等内容。

译者认为, M6800微处理机是工业控制用优选系列,其外围支持芯片十分齐全。若能通过熟悉这个系列的各种芯片,掌握它们的接口技术和编程方法,就能开展微型计算机的应用,并为进一步学习微型计算机的系统和开发打下基础,而且也很容易去熟悉其它系列的微处理机。因此,就本书的内容来看,对我国从事微型计算机应用的广大工程技术人员来说,是很有益处的。

由于原书是一本教科书,叙述上由浅入深,内容上系统全面,且具有学习指南、试题及答案和十个实验指导,因而尤为适合自学。对大专院校计算机专业的师、生也很有参考价值。

本书由合肥工业大学微型计算机应用研究所组织翻译的。译者按单元顺序为:庞鸿奎(一单元),谢文佩(二单元),冯金益(三单元),唐志兆(四单元),吕正阳(五单元),史斌宁(六、九单元),林锡来(七单元),贺建华(八单元),王建新(十单元)及郭桂银等。全书由朱逸芬副教授审校,旅美华侨赵鑑芳教授给予大力支持。由于译者水平有限,加上时间紧迫,书中难免有欠妥之处,恳请读者指正。

1983年11月

前 言

本教程必须以EE—3401教程为基础。那本教程介绍微处理机的基本知识，使你学习一种典型的第一代八位微处理器的内部寄存器结构，并且通过对其指令系统和寻址方式的研究，学习对它编程；使你熟悉定时、译码、三态缓冲和锁存等基本接口概念；从而使你有能力连接存储器和一些简单的I/O器件，建立一个可以运行的微型机系统。

在这本教程中，将继续介绍微处理机。由于微处理机和微型计算机技术在日新月异地变化，不可能介绍所有内容。但是，本教程能够向你提供工艺技术的状态。你将学习新的接口概念，了解标准外围器件如何操作，以及怎样把它们与微处理机系统相连。另外，还将熟悉某些目前最先进的微处理机。由于第一代八位微处理器，象8080、6800、Z—80等均已生产，许多新的器件也已供应市场，故重要的是，你的微处理机知识要随着技术的发展而提高，直至能精明地应用。

为了完成本教程中的各个实验，需要一台ET—3400微处理机学习机(Microprocessor Trainer)。还需要一只伏—欧表。某些实验中还要使用示波器。其它实验器材，例如集成片、电阻、电容均已提供*，请与印附元件表核对。如有差错，可依据所提供的元件表函索。

如何衡量你的学习？可以“教程目的”为指南。这些仔细结构的教程目的是本教程的主干。当你能够达到所有“目的”的时候，便满足了教程的要求。在本教程中有两种类型的目的。前言之后是“教程目的”。这是概括的。各单元前面是更具体的“单元目的”。当你能够达到各单元目的时，便学到了这些单元的每个内容。看来不那么容易。

在微处理机/微型计算机技术的迷人世界中继续学习吧。

作者

• 美国Heath公司提供本教程时，附有全部实验器材。

——译者注

教 程 目 的

当学完本教程之后,你将能够:

1. 利用外部接口适配器PIA,提供并行I/O操作的握手控制。
2. 描述数/模、模/数转换器的工作原理。
3. 把数/模、模/数转换器与6800微型机系统相连。
4. 说明几个数/模、模/数转换器的应用。
5. 构成一个微型计算机控制的数字电压表,即DVM。
6. 说明串行通信的基本概念,以及怎样为微型计算机提供软件和硬件的并/串转换。
7. 描述6850异步通讯接口适配器ACIA的工作特性。
8. 把6850ACIA与6800微型计算机系统相连以提供串行通信。
9. 描述电传打字机TTY、打印机、CRT数据终端、MODEM、盒式磁带机和软磁盘的工作特性。
10. 说明如何把各个外部I/O设备与6800微型计算机系统相连。
11. 解释怎样利用6847音频显示发生器即VDG,把彩色TV接收机与6800微型计算机系统相连。
12. 描述动态RAM, ROM, EPROM, E²ROM,磁泡存储器,电荷耦合器件存储器和约瑟夫结存储器的工作原理。
13. 说明怎样连接动态存储器,和提供动态存储器刷新。
14. 连接和编程2708EPROM。
15. 描述6840可编程定时器PTM的工作特性。
16. 连接6840PTM到6800微型计算机系统,产生定时的间隔和输出波形,并测量输入波形。
17. 描述几种新型微处理机和单片微型计算机,如6801、6802、6803、6805、6808、及68701等的内部结构。
18. 说明并且比较上面几种微处理机和单片微型计算机的能力。
19. 描述6809和6809E高性能8位微处理器的内部结构、指令系统、寻址方式和I/O线。
20. 描述68000 16位微处理器的内部结构、指令系统、寻址方式和I/O线。

目 录

前言

教程目的

第一单元 先进的PIA接口和模拟转换

引言	1-1
本单元目的	1-1
本单元学习指南	1-2
PIA接口	1-2
I/O控制技术	1-8
数字-模拟转换器	1-17
模拟-数字转换器	1-25
本单元考试题	1-31
试题答案	1-33

第二单元 串行数据通信

引言	2-1
本单元目的	2-1
本单元学习指南	2-2
串行通信	2-2
并行/串行转换	2-8
异步通信接口适配器 (ACIA)	2-14
本单元考试题	2-24
试题答案	2-25

第三单元 外部设备

引言	3-1
本单元目的	3-1
本单元学习指南	3-2
标准串行转换	3-2
外部I/O设备	3-11
大容量存储外部设备	3-24
本单元考试题	3-29
试题答案	3-30

第四单元 存储器

引言	4-1
本单元目的	4-1
本单元学习指南	4-2

动态RAM	4-2
只读存储器	4-10
磁泡、电荷耦合和约瑟夫逊结存储器	4-17
本单元考试题	4-22
试题答案	4-24

第五单元 可编程定时器

引言	5-1
本单元目的	5-1
本单元学习指南	5-2
可编程定时器及6840PTM	5-3
怎样寻址和初始化6840 PTM	5-12
使用PTM	5-17
本单元考试题	5-34
试题答案	5-36

第六单元 6800系列处理机

引言	6-1
本单元目的	6-1
本单元学习指南	6-2
6800系列的进展	6-2
6800和6802	6-6
6801和6803	6-11
6805	6-24
本单元考试题	6-32
试题答案	6-34

第七单元 6809微处理机

引言	7-1
本单元目的	7-1
本单元学习指南	7-2
6809寄存器结构和指令系统	7-3
6809寻址方式	7-15
6809接口线	7-27

本单元考试题	7-33
试题答案	7-35

第八单元 68000(第一部分)

引言	8-1
本单元目的	8-1
本单元学习指南	8-2
86000微处理机介绍	8-2
68000寻址方式	8-8
68000指令系统	8-16
本单元考试题	8-28
试题答案	8-30

第九单元 68000 (第二部分)

引言	9-1
本单元目的	9-1
本单元学习指南	9-2
8000接口线	9-2
处理状态和意外处理	9-10
本单元考试题	9-18
试题答案	9-20

第十单元 68000 (第三部分)

引言	10-1
本单元目的	10-1

本单元学习指南	10-1
连接6800外围器件	10-2
68000系列,支持和应用	10-12
本单元考试题	10-18
试题答案	10-19

第十一单元 实验

实验一 使用PIA的多路显示器	11-1
实验二 利用PIA进行I/O控制和握手操作	11-6
实验三 模/数转换和ADC接口	11-11
实验四 利用6850 ACIA进行并/串转换	11-19
实验五 标准的串行转换	11-24
实验六 2708EPROM编程	11-27
实验七 ROM/EPROM接口	11-35
实验八 6840 PTM输出	11-37
实验九 6840 PTM输入	11-43
实验十 6809软件操作	11-46

附录A 数据图表(略)

附录B 6800系列摘要(略)

第一单元

先进的PIA接口和模拟转换

引 言

这一单元首先要深入地讨论6821外围接口适配器。你将学会如何利用PIA通过其控制线去控制I/O操作。另外，你也将会看到PIA是怎样的一个多功能器件和它是怎样大大简化了接口技术中的问题。

本单元的第二部分，你将学习D/A和A/D转换是怎样进行的。因为事实上全部“现实世界”（real world）的数据，除金融界的数据外都是模拟量，绝大多数微处理器系统都必须进行某种转换来和“现实世界”进行通讯。因此，你将学习完成这种转换的称做DACs和ADCs的接口器件，如何经过PIA和微型计算机相连接。

为了了解本单元将要学习些什么，下一节列出有关学习目的，按照学习指南完成所需各步，保证学习成功。并且检验在学习步骤中画空格的每一步完成的情况，再记下你每完成一步所花费的时间。

本单元目的

当学完本单元之后，你将能够：

1. 把PIA接口到6800上，使PIA的A口和B口寄存器安置在一个连续的存储器地址中。
2. 使用6800的变址寄存器编程来予置PIA。
3. 利用PIA的控制线和外部设备进行通讯。
4. 通过编程来予置PIA可得到几种I/O控制操作（查询，完全握手，部分握手）。
5. 利用PIA控制线执行完整的输入和输出握手。
6. 提供D/A转换的两种方法。
7. 通过PIA把D/A转换器与微型计算机连接。
8. 提供A/D转换的三种方法。
9. 通过PTA把A/D转换与微型计算机连接。
10. 利用ET-3400学习机，把PIA和ADC构成一个数字电压表（DVM）。

本单元学习指南

	完成时间
阅读PIA接口这一节。	_____
完成复习题1—10。	_____
做第十一单元中的实验一。	_____
阅读I/O控制技术这一节。	_____
完成自我测试复习题11—25。	_____
做第十一单元中的实验二。	_____
阅读D/A转换器这一节。	_____
完成自我测试复习题26—29。	_____
（选做）EE-3401微处理器基础教程的实验九， 如有这本教程且有所需要元件的话。	_____
阅读A/D转换器这一节。	_____
完成自我测试复习题40—50。	_____
做第十一单元中的实验三。	_____
完成本单元考试。	_____
核对考试答案。	_____

PIA接口

大多数微处理器都有一系列的支持芯片，用以简化同外界的接口问题。外围接口适配器（PIA）6821就是最通用的接口芯片之一，PIA是为支持6800MPU而发展起来的，但也正被用在许多别的MPU微处理器的基本设计中。

PIA用来简化MPU到外部设备之间接口的问题。当然，任何器件利用常规组合逻辑都能同MPU连接。然而这种常规逻辑的方法，通常需要许多集成电路芯片，这就失去了微处理器的一个主要优点——仅需少数集成电路芯片的简单而直接了当的设计。PIA的优点是在许多情况下，只要一片或二片集成电路就能完成整个接口任务。

由于PIA能完成多种常规外围控制的任務，因此MPU可空出来去处理更重要的任务。PIA还使MPU能把外围设备当成存贮单元一样来处理。另外，PIA在高速MPU和低速I/O设备之间起着缓冲器的作用，因为PIA芯片内已有一些地址译码功能因此在许多实际应用中不需要单独的地址译码器。

PIA还在另一方面优于组合逻辑，因为PIA是可编程的，因而它非常灵活。也就是说它的组态可通过正在执行的程序从一种变为另外一种。例如，在一个程序当中，输出口可变为输入口。下面将会看到这是怎样做到的，但首先要学习PIA的内部结构。

PIA的基本概念

PIA按其功能可分为两边——MPU一边和外围设备一边。图1-1示出了PIA的功能图。在MPU这一边，包括连接到MPU的数据、地址和控制总线上的数据、地址和控制线。而外围的这一边包括和外部设备连接的2个8位I/O口（A口和B口）。每一个口都有8根数据线和两根控制线，每根数据线都可被配置为独立的输入线或输出线，控制线用做中断处理和握手。PIA包含了6个用户可寻址的8位寄存器，3个用于A口，3个用于B口。除后面要讨论的一种情况外，每个3寄存器组，相应于它们连接的口完成的功能相同。如图1-1中所示，每个3寄存器组都包括一个输出寄存器（ORA或ORB），数据方向寄存器（DDRA或DDRB）和控制寄存器（CRA或CRB）。

通过PIA从MPU到外设之间进行数据传输时，输出寄存器起到8位锁存器或暂存寄存器的作用。传输操作被认为是一个输出操作，因为，数据传输的公共参考点是MPU而不是外设。MPU通过写输出寄存器操作把数据从MPU经输出口线输送到外设。对数据输入操作，输出寄存器对用户是透明的（transparent），意即你可不用管它。当通过PIA进行数据输入时，内部三态缓冲器连接输入口线到MPU数据总线。因此，输入数据是不被锁存的，这可能是一个缺点，因为在PIA上的数据有效时，MPU必须准备好立即接收输入数据。MPU读输出寄存器地址，将通过输入口线把外设的数据传输到MPU中去。

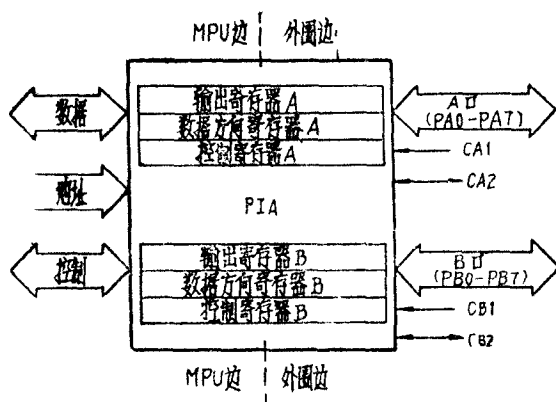


图1-1 外围接口适配器 (PIA)

PIA用来规定通过口的数据传输方向的寄存器是数据方向寄存器，每个口都已连接有一个8位数据方向寄存器，对A口是DDRA，而对B口是DDRB。在数据方向寄存器中（0—7）的每一位都形成各个口的I/O线（0—7），每一位或做输入或做输出。在某一个DDR中的某些位置成逻辑1时，对应位的口线就做为输出线，而逻辑0对应的口线就为输入线。例如，如果DDRA的位5是1，A口的线5（PA₅）将配置为输出线，如DDRB位3为0时，B口线3（PB3）就配置成输入线。因此，一个口用写一个字节到数据方向寄存器来配置。

控制寄存器是使PIA具有多种控制功能的8位寄存器。另外，控制寄存器也被用来表明状态信息。回想一下，对数据传输操作和予置PIA程序时，每一个控制寄存器的第2位是被用来选择输出寄存器还是数据方向寄存器的。控制寄存器的其它位连同PIA的四条控制线被用来进行中断处理和握手操作，这些操作将在下面详细介绍。

PIA寻址。

PIA到MPU各总线的连接如图1-2所示。PIA的数据线 (D_0-D_7) 直接接到MPU的数据总线 (D_0-D_7)。PIA的使能 (E)、读/写 ($R/\overline{W}$) 复位 ($\overline{RESET}$) 和中断 ($\overline{IRQA}$, $\overline{IRQB}$) 线用作控制和定时, 它们直接接到MPU相应的控制线上去。

为了PIA内部各寄存器的寻址, 必须利用PIA的寄存器选择 (RS) 线和片选 (CS) 线, 这些线可直接或间接连接到MPU的地址总线上。PIA的三根片选线用于访问PIA, CS_0 和 CS_1 必须为高电平, CS_2 必须为低电平。一旦PIA通过PIA片选线被选中, 两根寄存器选择线就被用来访问PIA的6个内部寄存器。

回忆一下 RS_1 用来选择A口或B口, 当 RS_1 是低电平就选中A口, 而当 RS_1 为高电平则选中B口。一旦一个口被选中, RS_0 和相应口的控制寄存器一齐用来选择该口的三个内部寄

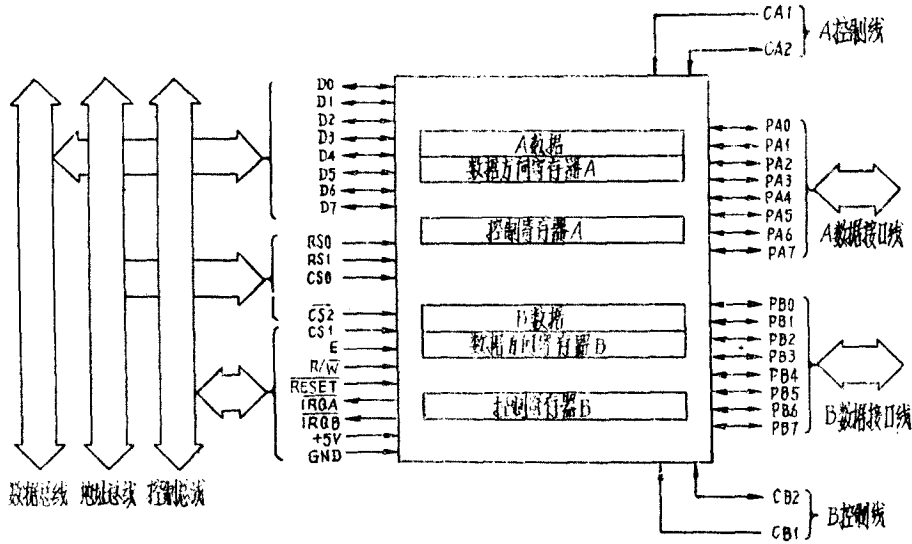


图1-2 MPU和PIA的外设接口线

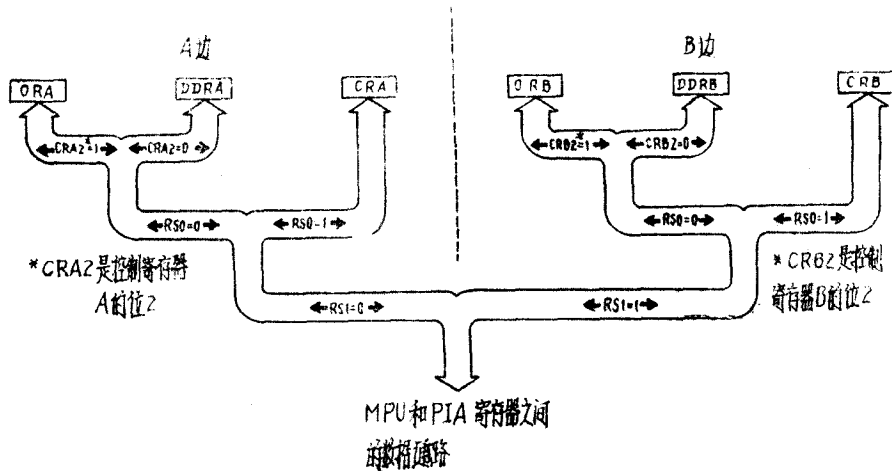


图1-3 PIA寄存器的选择

寄存器之一。如果 RS_0 为高, 该口控制寄存器被选中, 而如果 RS_0 为低, 不是选中输出寄存器

就是选中数据方向寄存器。在 RS_0 为低的情况下，实际哪个寄存器被选中是取决于相应口控制寄存器的第二位。若控制寄存器的第二位(dit_2)是低，就选中数据方向寄存器，若控制寄存器的第二位(bit_2)是高则输出寄存器被选中。例如，假设 RS_1 ， RS_0 和控制寄存器A的第二位都是低，在这些条件下，A口数据方向寄存器(DDRA)将被选中。PIA寄存器选择的过程被概括在图1-3中。

图1-4是PIA如何连到6800MPU的一种可能的电路图。PIA的数据线(D_0-D_7)、读写($R/\bar{W}$)和复位线($RESET$)都直接连到6800相对应的线上。PIA的使能线(E)为了定时直接连到6800的 ϕ_2 时钟线上。6800的地地总线通过外部逻辑的局部译码由片选线选择PIA。译码电路图示出PIA地址是5000到50FF，实际上仅使用地址5000——5003来访问PIA寄存器。在你的6800内存图中任何不用的区域都能被使用。我们已选择的这个内存映象是由于ET-3400微处理器学习机系统没有用到内存的这个区域。在第十一单元ET-3400 PIA的实验中就使用了这个线路。

对本教程，我们总是将PIA寄存器选择线(RS_0 和 RS_1)连到MPU地址线 A_0 和 A_1 上。怎样连接它们都无甚区别。不管怎样，如果 RS_0 连到 A_1 ， RS_1 连到 A_0 ，则PIA的两个输出寄存器将被置位在两个连续的地址单元中。对数据方向寄存器和控制寄存器的情况也是一样。在后面将体会到看到，在多字节数据(16位)工作的时候，这是一个优点。

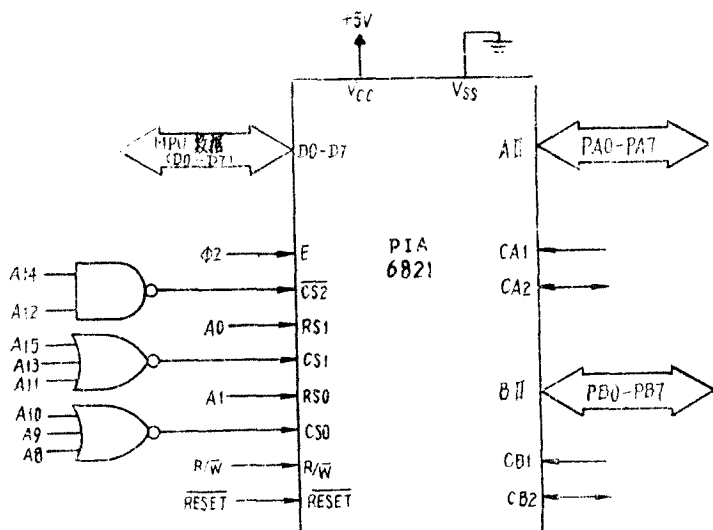


图1-4PIA到6800MPU的连接

利用图1-4的电路，PIA的寄存器被访问的情况如图1-5所示。注意，输出寄存器，数据方向寄存器和控制寄存器是被分配在连续地址上。此外地址，5000和5001不是选择输出寄存器就是选择数据方向寄存器。具体选择哪一个寄存器则取决于各自控制寄存器的第2位。

PIA的初始化

当你用PIA进行数据传输前，必须规定哪个口用做输入，哪个口做输出，这一操作称PIA的初始化(予置)。为初始化PIA，必须完成一个短的程序，把一个数值存

地 址	CS0	CS1	CS1	RS1	RS0	寄存器的选择
5 0 0 0	1	1	0	0	0	DDRA 或 ORA*
5 0 0 1	1	1	0	1	0	DDRB 或 ORB*
5 0 0 2	1	1	0	0	1	CRA
5 0 0 3	1	1	0	1	1	CRB

*选择取决于PIA各自控制寄存器的第2位

图1-5 按图1-4电路图选择PIA各寄存器地址

放在口的数据方向寄存器中。在数据方向寄存器中的这个数值规定了什么口用作输入，什么口用作输出。PIA两个口配置所需要步骤如下：

第一步：清两个控制寄存器第2位，选取数据方向寄存器。

第二步：按两个口的配置要求，把数据存放在数据方向寄存器中。

第三步：两个控制寄存器的第2位置1，以访问输出寄存器，接着就可进行数据I/O传输操作。

图1-6的程序是按A口作为输入口，而B口作为8位输出口予置PIA。我们假定按图1-4连接PIA到6800，并且根据图1-5指定寄存器。由于我们已拟定了A口和B口寄存器是连续地址单元，因此6800的16位变址寄存器就能被用来予置PIA，然后我们映象A口和B口寄存器到连续的地址单元。最后一点，如果在第一次初始化以前，系统是复位的，则第一步可以不要，因为在PIA复位以后所有的内部寄存器都清零。

回想一下，在PIA被初始化以后，在6800和外设之间可用任意的存储器访问指令进行数据的传输。6800各种取(LD)和存(ST)指令常被用于此目的。一条取指令可被用来做数据输入即读操作。而一条存指令可被用来做数据输出即写操作。

在做完自我测验复习题后，做第十一单元的实验一。

PIA提供了两种类型的接口驱动器。当它们被定义为输出线时，A口线是集电极开路的，而B口是提供三态输出。回忆一下，三态逻辑是用内部图腾柱电路结构以达到低阻抗快速输出的目的。PIA的口线和TTL兼容，并且每一口线都能够驱动两个标准的TTL负载。其它输出端线也需要外部缓冲。此外，B口线在1.5伏电压下能提供1mA的电流以直接驱动复合晶体管开关的基极，而A口线不用外接上拉电阻能直接驱动CMOS器件。

然而PIA的输出驱动能力是有限的。例如，你必须提供外部缓冲器来驱动诸如小的白炽灯和继电器之类的外围设备。此外，从PIA到外部设备之间有一定距离的时候，也需要外部缓冲。

在许多情况下，为此目的可使用集电极开路的缓冲器。集电极开路器件能提供比PIA所提供的三态输出更大的电流和更高的电平。而且集电极开路器件还能用来通断接地的电路，当连接到小继电器时这一特性尤其需要。如果必须利用集电极开路器件做为PIA口线的缓冲器时，要知道，因为集电极开路器件需要一个相当大的上拉电阻，结果使开关速度变慢。

源码程序	注 释
LDX #	
0 0	清两个控制寄存器的位2 (第一步)
0 0	
STX \$ \$	
5 0	
0 2	
LDX #	
0 0	配置A口为输入和B口为输出 (第二步)
FF	
STX \$ \$	
5 0	
0 0	
LDX #	
0 4	两个控制寄存器位2置位 (第三步)
0 4	
STX \$ \$	
5 0	
0 2	

无符号 = 隐含

= 立即

\$ = 直接

\$ \$ = 扩展

X = 变址

图1-6PIA的初始化程序

自我测验复习题

1. 在PIA内部, 用户可存取的是什么寄存器?
2. 在微型计算机系统中, 数据传输的参考点总是_____。
3. 要配置B口为数据输出, B口的数据方向寄存器内十六进制值是_____。
4. 数据传输中, 控制寄存器的第_____位用来选择OR或DDR
5. 选择A口或B口是用PIA的_____线。
6. I/O数据在什么时候被PIA锁存?
7. 为把输出寄存器, 数据方向寄存器, 控制寄存器置成连续地址单元RS₀被接到_____而RS₁接到_____。
8. 按题7这样连接的优点是什么?
9. PIA两个口初始化所要求的步骤是什么?
10. 假设按图1-4所示连接PIA到6800, 地址寄存器写一个配置A口为输出, B口为输入的程序。

答案

1. 对两个口 (A和B), 用户可存取的是输出寄存器, 数据方向寄存器和控制寄存器。
2. MPU。
3. FF16。
4. 第2。
5. RS₁。
6. 仅在输出操作的时候, 数据进到PIA不用锁存, 但必须立即被MPU读走。
7. A₁, A₀。
8. 在A口和B口各寄存器置成连续地址单元以后, 用十六位寄存器指令可同时访问两个口。
9. 第一步: 清两个控制寄存器的第2位。
第二步: 在数据方向寄存器中寄存数值。
第三步: 置两个控制寄存器的第2位为1。
10.

```
LDX  *0000
STX  $ 5002
LDX  #FF00
STX  $$ 5000
LDX  #0404
STX  $ 5002
```

I/O控制技术

利用程序控制或中断控制这两种基本的控制技术之一,可实现MPU和外部设备之间的通讯。回想一下,在程序控制期间,所有I/O传输都是被程序控制的。不管I/O设备是否需要服务,程序都以一定的时间间隔从输入设备读取数据和向输出设备写数据。在这里,明显的优点是简单。然而,你的简单却牺牲了效率,因为有很多时间MPU被占用去执行一些不必要的操作。

程序控制I/O一个较为有效方式是查询,用这种I/O控制方式,外设要激励一个状态标志。当它被外设激励时,这个状态标志通常是将一个位单元置位或清零。MPU周期地读状态标志确定是否需要服务。假设标志被激励,于是将发生数据传输。因为MPU必须周期地间断它正在做的工作去检验状态标志,因而也就要牺牲一些效率。外设必须等待直至MPU轮到检测它的状态标志位为止。因此,就有一个响应时间的问题。

I/O控制最有效的方式是用中断。用这种控制方式是MPU主程序被外设所中断,而中断仅发生在外设确实需要MPU服务时,而且I/O设备通常能从MPU得到一个相当快的响应时间。让我们观察一下如何用PIA执行这些I/O控制操作的。

I/O操作的程序控制

象在以前做过的那样,对于PIA的I/O简单程序控制,可经PIA口的读和写即数据存取来完成。而为了完成一个查询操作,PIA的状态标志必须是有效的。这就需要把状态标志置于PIA的控制寄存器中。因此,让我们对控制寄存器作一较仔细地了解。

控制寄存器各位的功能如图1-7所示。暂时我们假定A口和B口两个控制寄存器的位功能是相同的。注意,控制寄存器有两个基本的功能,即状态和控制。位0到位5是控制位,而位6和位7是状态位。现在回想一下,每一个PIA口都有两条联合的控制线,这些线也示在图1-7中,用CA₁, CA₂和CB₁, CB₂表示。控制线CA₁和CB₁总是作为各自对应口的输入线。而要指出的是CA₂和CB₂既可做输入控制线也可做输出控制线,在图1-7中是用双向箭头表示的。

控制寄存器的状态位表明了口上控制线的有效作用,也就是选通。如果CA₁是有效的,A口控制寄存器的第7位将置位(逻辑1)。同样地,如果CB₁是有效,B口控制寄存器将位7置位。当CA₂或CB₂被用作输入线时,相应口的控制寄存器第6位就指明在这些线上的有效作用。因而控制寄存器的第6位和第7位被规定为状态标志位。

在上面的讨论中,当一根控制线提出服务请求时,我们用了专门的术语“启动”或“选通”。对一个有效跳变是怎样定义的呢?在这里使用了控制寄存器的位1和位4。如果位1置位(逻辑1),那么在CA₁/CB₁上的一个上跳变(0→1)将使状态标志位7置位。

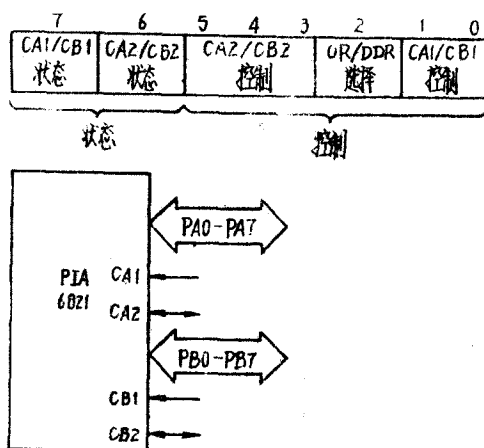


图1-7 PIA控制寄存器位功能和控制线

而如果位1是被清0(逻辑0),则CA₁/CB₁上的下跳变(1→0)将启动状态标志位7。同样方法,当CA₂/CB₂线作输入线时,位4定义这些线的有效状态。如果位4置位,那么在CA₂/CB₂上的上跳变将启动该标志位。

现在,如果一个外设要求MPU服务,它必须启动(选通)一根PIA的控制线。因此,可以周期查询控制寄存器及检测位6或位7状态来测定什么时候I/O设备需要服务。一旦标志位置位,接着就用读相应口输出寄存器来清除此标志。

图1-8可当成一个短小的程序来使用,叫做周期测试CA₁标志子程序,该标志是A口控制寄存器的位7。在这个查询子程序中,假设PIA已分配的地址如图1-5所示5000到5003,并且指出使用了BMI(如是负就分支)指令。虽然控制寄存器中不是算术数据,但6800将仍然把在累加器中最高位(位7)看做是符号位。因此,启动CA₁的标志将表示在累加器中为一负值,并且BMI指令能识别这一状态。如输入选通线是CA₂,应该怎样改变图1-8中的子程序去测试CA₂的标志呢?一用一条算术左移(ASLA)指令,加在BMI指令的前面即可。因为CA₂的标志是控制寄存器的位6,这一标志在取到累加器后,在用BMI指令测试该状态之前,必须左移到符号位的位置上。

I/O操作的中断控制

现在假设时间对MPU是最关键的,且当I/O设备需要服务时,希望能迅速地做出响应。在这种情况下必须采用中断控制。当任何口的控制线被一个

源程序码	注 释
LDA \$5002	读A口控制寄存器
BMI XX	测试CA ₁ 标志,如置位就分支到I/O程序
RTS	如标志位没置位就返回

图1-8 测试CA₁状态标志的一个查询子程序

I/O设备启动(选通)时,PIA就能够向MPU产生一个中断,中断的产生是由PIA控制寄存器的位0或位3来完成的。如果控制寄存器位0置位,若CA₁/CB₁线有效,将使PIA向MPU发中断。当位0被清除时,就不产生中断。因此,必须采用象上节所讨论过的I/O程序控制。同样,当控制寄存器位3置位时,由于CA₂/CB₂上的有效输入,也将产生中断。通过在PIA上标定为IRQA和IRQB的两根输出线向MPU发中断。这些线在图1-9中都有说明。注意,一个有效的CA₁或CA₂将导致IRQA由高到低的下跳变。同样一个有效的CB₁或CB₂也将导致IRQB产生一个由高到低的下跳变。对于任何通过IRQA或IRQB产生的中断,都必须使控制寄存器的位0或位3置位。PIA的这些中断线能和任何MPU的硬件中断引出脚相连。在6800中,IRQ, NMI或RESET都能被使用。实际上PIA的这两条中断线能连接在一齐(线或)接到MPU的同一个中断线上。

另外一点,当PIA产生一个中断时,PIA相应的中断线(IRQA或IRQB)将保持低电平,直到在相应口的输出寄存器执行一个读操作为止。

现在你已经熟悉了除位5外的控制寄存器的所有各位。当I/O设备在使用PIA执行握手操作时,就使用位5这一位。握手操作在下一节进行讨论。在我们往下讲以前,用一些例子讨论一下前面的控制寄存器,这有助于我们复习。

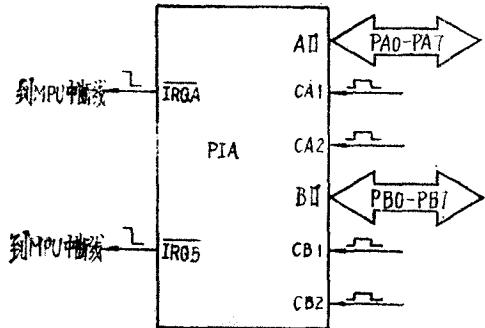


图1-9 PIA的中断线

假设在控制寄存器B中存储了十六进制值1C, 在存储操作之后, 控制寄存器B配置成图1-10所示的那样, 这一结构将控制B口工作如下:

——位0为零, 将对一个有效CB₁线禁止IRQB中断。

——位1为零, 规定CB₁线上的下跳变为有效状态。

——位2置1, 对接着进行的数据传输操作选择B口输出寄存器。

——位3置1。对于CB₂线上的一个有效跳变将使能IRQB中断。

——位4置1, 规定CB₂线上的上跳变为有效状态。

——位5为零, 指定CB₂为输入控制线(下面立即讨论)。

——位6为零, 用作CB₂的状态标志。

——位7为零, 用作CB₁的状态标志。

注意, 我们用CB₂来启IRQB中断线, 而用CB₁来禁止。因此, 中断控制应使用CB₂, 而程序查询则必须使用CB₁。查询程序应周期地检测CB₁状态标志(位7)。

现在假设在控制寄存器A中存储了十六进制数04, 在存储操作后, 控制寄存器A将被配置如图1-11所示的那样。这一结构将控制A口工作如下:

——位0为零, 表示在CA₁线上的有效跳变将禁止IRQA中断。

——位1为零, 规定CA₁线下跳变为有效状态。

——位2置1, 对接着进行的数据传输操作选择A口输出寄存器。

——位3为零, 表示在CA₂线上的有效跳变将禁止IRQA中断。

——位4为零, 规定CA₂线下跳变为有效状态。

——位5为零, 指定CA₂为输入控制线。

——位6为零, 用作CA₂的状态标志。

——位7为零, 用作CA₁的状态标志。

在这里两个中断使能位都为零, 所以查询程序必须测定位7和位6, 以确定何时CA₁和CA₂为有效。

握手操作

计算机作I/O操作时, 术语“握手”指的是协调MPU与I/O设备之间通讯的一种技术。作为一个完全握手操作, 必须有双向通讯。一般说来, 使用双向同步通讯信号: 其一问“准备好了吗?” 另一个回答“是”, “没有”, 或“操作完成”。包含一根信号线的单向通讯有时叫做部分握手操作。此外, 通常有两种形式的握手: 一是包含一个数据输入设备的输入握手和包含一个输出设备的输出握手。例如, 一个完全的输入握手操作大体的过程如下:

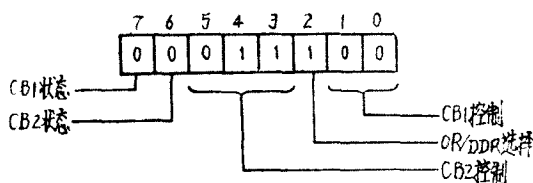


图1-10 通过提供I/O程序控制和通过CB₂提供中断控制所配置的控制寄存器B

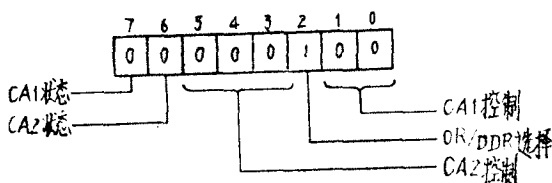


图1-11 通过CA₁和CA₂两者提供的I/O控制所配制的控制寄存器A