

高等学校教材

接口技术

于英民 孙全 莫玮 编著

电子工业出版社

接口技术

于英民 孙全 莫玮 编著

电子工业出版社

内 容 提 要

本教材较全面地介绍微计算机接口技术的原理与应用。全书共十一章。前两章介绍微处理器系统的连接，重点是CPU总线。第三至七章介绍微处理器或微计算机系统与“外部世界”的连接，重点介绍并行接口、GPIB接口、串行接口、定时器接口、模拟器件接口。后四章是系统问题，重点介绍信号处理器接口、底板总线、接口软件设计并简介CAMAC接口。

本教材适合计算机应用、通信与系统、仪表与测量、监测与控制等学科的研究生与本科生。

接 口 技 术

于英民 孙 渝 莫 玮 编著

责任编辑：邓又强

电子工业出版社出版(北京海淀区万寿路)
新华书店北京发行所发行 各地新华书店经售
北京华新印刷厂印刷

开本：787×1092毫米1/16 印张：12.625 字数：307千字

1989年9月 第一版 1989年9月第一次印刷

印数：1—4000册 定价：2.70元

ISBN7-5053-0594-8/TP·88

出版说明

根据国务院关于高等学校教材工作分工的规定，我部承担了全国高等学校、中等专业学校工科电子类专业教材的编审、出版的组织工作。由于各有关院校及参与编审工作的广大教师共同努力，有关出版社的紧密配合，从1978年至1985年，已编审、出版了两轮教材，正在陆续供给高等学校和中等专业学校教学使用。

为了使工科电子类专业教材能更好地适应“三个面向”的需要，贯彻“努力提高教材质量，逐步实现教材多样化，增加不同品种、不同层次、不同学术观点、不同风格、不同改革试验的教材”的精神，我部所属的七个高等学校教材编审委员会和两个中等专业学校教材编审委员会，在总结前两轮教材工作的基础上，结合教育形势的发展和教学改革的需要，制订了1986~1990年的“七五”（第三轮）教材编审出版规划。列入规划的教材、实验教材、教学参考书等近400种选题。这批教材的评选推荐和编写工作由各编委会直接组织进行。

这批教材的书稿，是从通过教学实践、师生反映较好的讲义中经院校推荐，由编审委员会（小组）评选择优产生出来的。广大编审者、各编审委员会和有关出版社为保证教材的出版和提高教材的质量，作出了不懈的努力。

限于水平和经验，这批教材的编审、出版工作还会有缺点和不足之处，希望使用教材的单位，广大教师和同学积极提出批评建议，共同为不断提高工科电子类专业教材的质量而努力。

电子工业部教材办公室

前 言

本教材系按电子工业部的工科电子类专业教材1986~1990年编审出版规划,由无线电技术与信息系统教材编审委员会仪表与测量编审小组征稿、评选、推荐出版的。责任编辑王慧云副教授。

本教材由北京邮电学院于英民、孙全和桂林电子工业学院莫玮编著,天津大学王慧云副教授担任主审。

本教材的对象是宽口径的。适合计算机应用、通信与系统、仪表与测量、信号与电路、监测与控制等学科的研究生和本科生。本教材内容安排还考虑了学科的发展和联系实践,取材主要来自我们的教学和科研工作。本教材对正在从事接口技术的科技人员也是一本合适的参考书。

本教材的参考学时数为50学时,其内容可分为三部分。

第一部分是微处理器系统的连接。它包括第一章和第二章。第一章的重点是CPU总线,包括总线的概念、总线的实现和总线的时序。第二章讨论用CPU总线连接系统可能遇到的进一步的问题。此外,还较全面地讨论了微处理器系统的中断技术。

第二部分是微处理器系统或微型计算机与其“外部世界”的接口。此部分包括第三章至第七章。第三章是并行接口。第四章是标准并行接口,主要是 GPIB 接口。本书还从教学法上作了考虑,并行接口一章的某些写法是为标准并行接口服务,而标准并行接口一章又是并行接口一章的发展,以分散 GPIB 教学上的难点。对本书 GPIB 内容显得过多的专业,使用本书时可跳过 4.7 至 4.10 的四节。第五章串行接口较详细地讨论了异步通信技术。第六章重点是如何在微计算机中加入时间功能。第七章模拟器件接口,较系统地讨论了 A/D、D/A 的原理、特性和应用。

第三部分是系统问题。第八章是信号处理机接口和多微处理器系统。信号处理器是针对数字信号处理技术特点设计的一种较为专用的微处理器,它在应用中常和通用微处理器组成多微处理器系统。第九章底板总线,此章的内容可以在组建系统时给予硬件指导。第十章是接口软件设计,这是正在发展中的内容,越来越重要,因为硬件趋向标准化,而系统又向大型发展。考虑到各学校的条件不同,在应用时可以按各自的条件进行补充,也可以删去此章不作讲授。第十一章 CAMAC 接口,是一种更大更复杂的系统。我国这方面的工作还不普遍,所以仅给出简介。

本教材由莫玮编写第八章,由孙全编写第五章和第十章,其余由于英民编写并统编全稿。成都电信工程学院张世寰教授和北方交通大学蒋焕文教授领导的编审小组对本教材提出许多宝贵意见和建议;主审王慧云副教授认真审阅了全稿,提出了指导性意见,为提高书稿质量付出了辛勤的劳动;本院计算系汪雍教授、教务处和无线电工程系的领导始终关心和支持本书的写作;微机化系统研究室的同志和在本室工作的研究生为本书早日出版付出了坚辛的劳动,在这里对他们表示诚挚的感谢。

由于作者水平有限,书中难免存在错误和不足之处,殷切希望广大读者批评指正。

作者

一九八八年八月

目 录

第一章 微处理器接口原理	(1)
1.1 总线的概念.....	(1)
1.2 输入口.....	(3)
1.3 输出口.....	(4)
1.4 状态口.....	(4)
1.5 控制口.....	(5)
1.6 中断向量输入口.....	(6)
1.7 存贮器接口.....	(6)
1.8 地址译码器.....	(8)
1.9 CPU控制总线及 CPU 定时.....	(10)
思考题和练习题	(15)
第二章 微处理器接口实际	(16)
2.1 总线的负载效应.....	(16)
2.2 总线冲突的避免.....	(17)
2.3 动态存贮器接口.....	(20)
2.4 输入输出控制方法.....	(23)
2.5 中断操作技术.....	(25)
2.6 直接存贮器存取 (DMA)	(30)
思考题和练习题	(31)
第三章 并行接口	(32)
3.1 握手联络线.....	(32)
3.2 并行接口集成电路.....	(34)
3.3 并行接口应用.....	(44)
思考题和练习题	(50)
第四章 标准并行接口	(52)
4.1 打印机接口“标准”	(52)
4.2 GPIB 接口概述.....	(53)
4.3 GPIB 系 统描述.....	(54)
4.4 GPIB 数 据字节的传送控制.....	(56)
4.5 GPIB消息规定和接口管理.....	(57)
4.6 GPIB 接 口功能.....	(60)
4.7 接口功能状态图.....	(61)
4.8 接口功能子集.....	(73)

4.9	GPIB接口实现和GPIB集成电路	(76)
4.10	仪器GPIB接口及远控工作	(90)
4.11	GPIB控制器实例	(91)
	思考题和练习题	(93)
第五章	串行接口	(94)
5.1	串行通信基本概念	(94)
5.2	串行接口电路	(95)
5.3	调制解调器	(100)
5.4	异步串行通信接口连接标准	(100)
5.5	串行接口实现举例	(105)
5.6	异步串行通信的可靠性	(106)
	思考题和练习题	(111)
第六章	定时器接口及CRT显示器接口	(112)
6.1	CTC的组成及引脚信号	(112)
6.2	CTC的编程预置	(115)
6.3	编程和工作过程说明	(117)
6.4	CTC应用举例	(119)
6.5	CRT显示器接口	(123)
6.6	软磁盘接口	(127)
	思考题和练习题	(127)
第七章	模拟器件接口	(129)
7.1	D/A转换器	(129)
7.2	A/D转换器	(131)
7.3	采样保持器和多路转接器	(136)
7.4	隔离和接地技术	(139)
7.5	模拟器件接口实例	(140)
	思考题和练习题	(146)
第八章	信号处理器接口和多微处理器系统	(147)
8.1	多微处理器系统的基本概念	(147)
8.2	几种典型的通用信号处理器	(149)
8.3	信号处理器系统的组建	(155)
8.4	信号处理器接口	(156)
8.5	信号处理器系统的应用与开发特点	(164)
第九章	底板总线	(168)
9.1	引论	(168)
9.2	STD总线	(169)
9.3	S100总线	(172)
9.4	IBM PC 总线	(176)
第十章	接口软件设计	(179)

10.1	软件开发环境及开发工具.....	(179)
10.2	接口软件设计.....	(180)
第十一章	CAMAC 标准接口系统简介	(187)
11.1	CAMAC系统概况.....	(187)
11.2	CAMAC标准接口系统.....	(187)
11.3	CAMAC系统的应用和发展.....	(191)
参考资料		(192)

第一章 微处理器接口原理

从微计算机原理课中知道,微处理器是通过其总线(数据总线、地址总线、控制总线)和存贮器、I/O接口电路等连接成各种微处理器系统或微型计算机而进行工作和应用的。本书从研究和解决这类系统中“连接”或接口的共性问题开始。

1.1 总线的概念

什么是总线?总线是一种公用导线,一些数据源中的任一个都可以利用它传送数据到另一个或多个目的。例如,在图1.1中,假定数据从器件A传送到器件C,此时器件A是数据源,器件C是目的;在另一时刻,数据从器件D传送给器件A,此时器件D是源,器件A变成目的。要使数据传送无错误,要有一定的时序,第一个事件结束后,才能开始第二个事件。此外,在给定时间周期内,源只能有一个,目的则可以多个。换言之,在特定时间周期内,总线仅能为一个源专用,从属于一个源。图1.1给出一条导线的连接图,实际微处理器有8条或16条数据总线,其总线的性质和这里分析的情况一样。

1.1.1 集电极开路电路

图1.1示出几个源的输出通过总线连在一起,标准TTL门电路不允许进行这种连接。假定作了这种连接,当一个门电路输出高电平而另一个输出低电平时,在它们的输出将产生很大的电流,结果导致输出电平不确定。为了说明这个问题,图1.2(a)示出标准TTL门电路的原理图,图1.2(b)示出两个这种电路连在一起的输出。假定A电路企图输

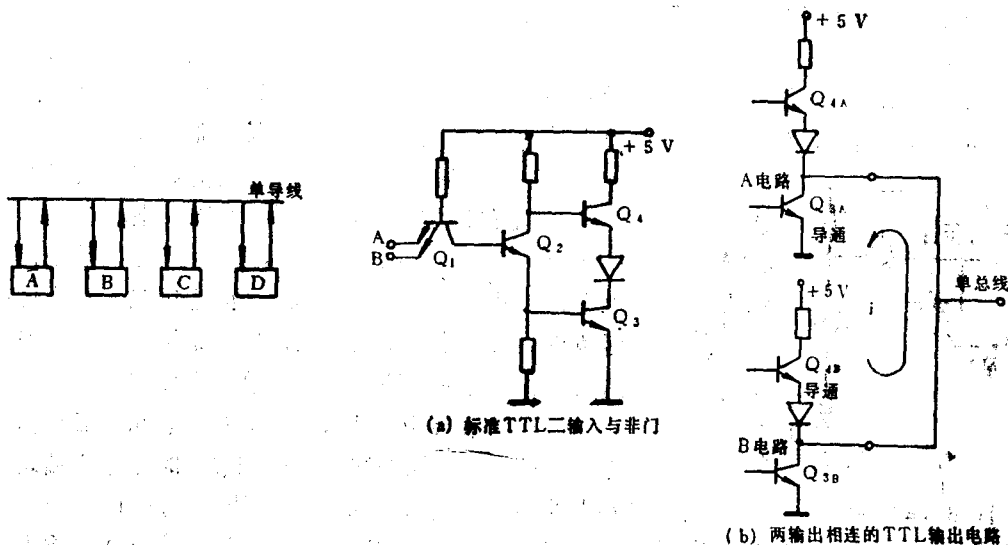


图 1.1 单导线总线

图 1.2 TTL 输出不能互连的说明

出低电平,其 Q_{3A} 导通, Q_{4A} 截止; B 电路企图输出高电平,其 Q_{3B} 截止 Q_{4B} 导通。结

果是 5V 电源将通过 B 电路导通的 Q_{4B} 和 A 电路导通的 Q_{3A} 产生一个大电流, 近似短路。此电流值超过器件的额定值, 因此输出将不是标准的 TTL 高电平, 也不是低电平。

上述问题的解决方法, 是用集电极开路电路。集电极开路需外接一个上拉电阻, 使输出达到高态。图 1.3 示出两个集电极开路输出连在一起, 共用一个上拉电阻, 因此最大电流将会被限制。假定两晶体管或一个晶体管导通, 输出将是低电平, 相反, 则为高电平。

图 1.4 示出如何用集电极开路电路实现总线的概念。例如, 要求信号从器件 A 传送到

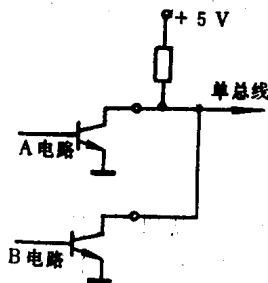


图 1.3 两集电极开路电路的输出电路

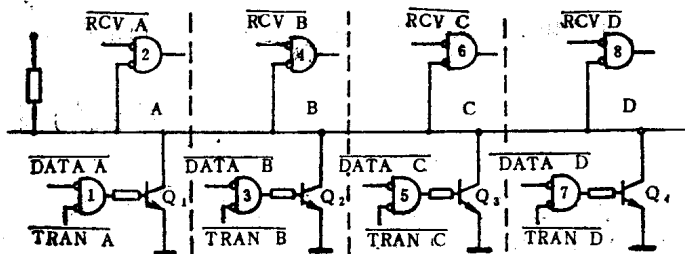


图 1.4 用集电极开路电路实现总线传输

器件 C, $\overline{\text{TRAN A}}$ 必须有效而 $\overline{\text{TRAN B}}$ 、 $\overline{\text{TRAN C}}$ 和 $\overline{\text{TRAN D}}$ 必须无效, DATA A 在门 1 输出, 它的“非”出现在总线上。因为 $\overline{\text{TRAN B}}$ 、 $\overline{\text{TRAN C}}$ 和 $\overline{\text{TRAN D}}$ 无效, 所以 Q_2 、 Q_3 和 Q_4 截止 (与总线分离)。为了实现传输, $\overline{\text{RCV C}}$ 也必须有效, 使门 6 的输入开, 门 6 输出总线数据的“非”。门 6 的输出等于 DATA A。任何其它 $\overline{\text{RCV}}$ 线也可以有效, 使数据传给多个目的。因为, 在给定的时间周期内目的可以有多个, 源仅能有一个。

1.1.2 三态电路

三态电路对利用总线传送数据提供了另一种手段。三态电极和集电极开路电路的主要区别是保留有源上拉电阻, 使输出阻抗可控制。

三态的定义: (1) 低阻抗高电平输出, (2) 低阻抗低电平输出, (3) 高阻抗输出。图 1.5 示出一种三态非门, 当禁止端 $\overline{\text{DISABLE}}$ 高电平时, D_1 截止, 它和图 1.2(a) 的电路相同。所以, 当输入端为高电平时, 因 Q_3 导通 Q_4 截止而输出低电平, 并且是低阻抗的。当输入低电平时, 因 Q_3 截止 Q_4 导通而输出高电平, 输出阻抗因 Q_4 导通, 仍是低阻抗的。但当 $\overline{\text{DISABLE}}$ 端为低电平时, D_1 导通将 Q_4 的基极钳在低电平, 使其截止。另一方面, Q_2 也因输入端为低电平而截止, 所以输出呈高阻态, 这就是第三态。正因为有这第三态, 才使它适合挂载在总线上使用。

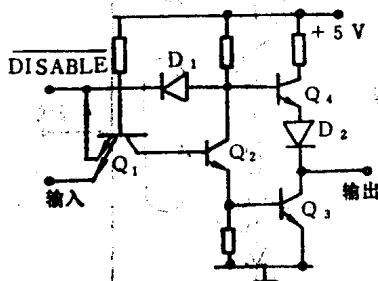


图 1.5 三态反相器

三态反相器的符号示于图 1.6。它和一般反相器相比, 多一个三态控制端。此控制端有时用禁止端表示, 如图 1.6 (a) 所示。也有时用使能端 $\overline{\text{ENABLE}}$ 表示, 如图 1.6 (b)。对图 1.6 (a), 当 $\overline{\text{DISABLE}} = 0$ 时, 输出为高阻态, 对图 1.6 (b) 来说, 当 $\overline{\text{ENABLE}} = 0$ 时输出为低阻抗状态。

用三态电路实现总线数据传输的说明示于图 1.7。假设要求把数据 A (DATA A)

传送给器件 D, $\overline{\text{TRAN A}}$ 应当有效, 使 DATA A 送上总线, $\overline{\text{RCV D}}$ 也应当有效, 将总线上的数据选入 D 器件, 使它在 D 输出出现。因为只允许有一个源利用总线, 所以, 在作

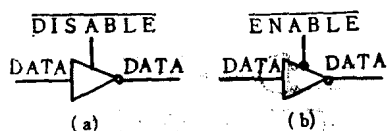


图 1.6 三态反相器的符号

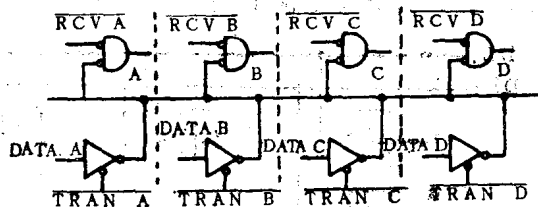


图 1.7 用三态电路实现总线数据传输

上述传输时, $\overline{\text{TRAN B}}$ 、 $\overline{\text{TRAN C}}$ 和 $\overline{\text{TRAN D}}$ 都必须无效, 使不用电路呈高阻, 与总线分离。

1.1.3 总线驱动器

微处理器一般是 MOS 单片集成电路。这种电路的驱动能力很弱, 仅能驱动一个标准 TTL 负载。因为总线上挂接的电路可能很多, 所以在微处理器引脚和总线间, 或总线和电路间需要缓冲器或驱动器。从减轻处理器或总线的负载考虑, 可以叫缓冲器, 从提高处理器或总线的驱动能力考虑, 又叫驱动器。驱动器和缓冲器是同一电路。驱动器分两种电路: 单向和双向。单向驱动器基本上就是驱动能力强的门电路, 在此不需讨论, 见 2.1.3 节。双向驱动器有特殊性, 图 1.8 示出双向驱动器电路。它由两个三态门组成, 仅有一个使能控制信号。从图中看出, 在任何时候仅有一个门被使能。假设处理器是数据的源, 控制信号 CONTROL 应当是高电平, 电路 A 被使能。假设处理器是目的, 控制信号应为低电平, 电路 B 被使能。

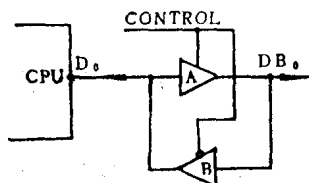


图 1.8 双向驱动器传输方向控制说明

图 1.9 示出 8 位双向总线驱动器 74LS245 的简化图。G 引脚是使能控制端, 它有效时, 三态门的传输方向决定于 DIR 引脚。DIR 是方向控制引脚。DIR 有效时, 数据的传送方向为 A 到 B, DIR 无效时, 方向由 B 到 A。

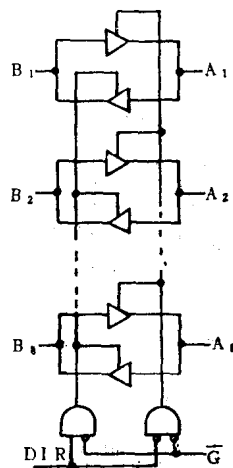


图 1.9 74LS245 8 位双向驱动器简化图

1.2 输入 口

输入口是外部设备通过总线向 CPU 输入数据的端口。它是 CPU 从外部设备输入数据的接口电路。图 1.10 示出简化的 8 位数据总线和三个输入口。

我们从 0^号 端口开始分析外设数据如何输入 CPU。端口由三态门组成, 0^号 端口的

三态门使能端连接门A的输出, 由门A控制。当不进行数据交换时, 门A输出低电平, 0^* 号端口高阻态与总线分离。当输入数据时, 三态门被使能, 外设的数据通过 0^* 号端口送上总线。A门输入受端口选择信号 $\overline{PS}_0$ 和I/O读控制信号 $\overline{IOR}$ 的控制。微处理器执行

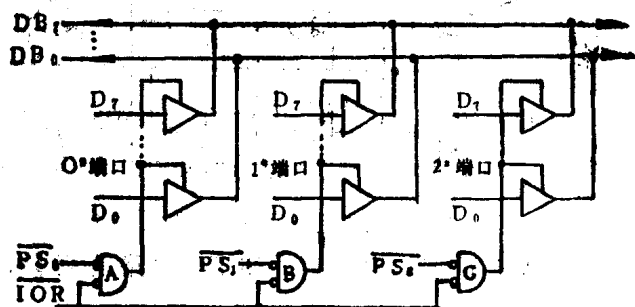


图 1.10 三态门输入端口

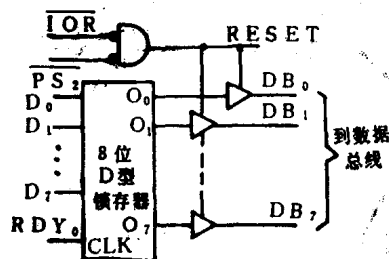


图 1.11 锁存型输入

输入指令, $\overline{PS}_0$ 被有效, 接着 $\overline{IOR}$ 有效, 数据从 0^* 号端口送上总线, CPU从其总线引脚取入内部。在图1.10中, $\overline{IOR}$ 对各端口是公共的, 但微处理寻址时仅有一个端口被选中。在上述情况下, 应当 0^* 号端口选中, 即 $\overline{PS}_0 = 0$, $\overline{PS}_1$ 和 $\overline{PS}_2$ 都应当为“1”。

以上分析中实际上已假定外设的 0^* 号口提供的数据是长时间存在的。实际微处理器的读操作是在一瞬间完成。有些外部设备提供数据的存在时间也是短暂时, 不可能和微处理器的读瞬时同步, 这就要求在输入端口中有锁存器, 而且外设还必须提供一个时钟脉冲, 将输入数据锁存住, 以备微处理器按它自己的定时读入。图1.11示出带锁存器的输入口(图中RESET信号的作用见1.4节)。

1.3 输 出 口

输出口是微处理器通过总线向外部设备输出数据的端口, 也是微处理器向外部设备输出数据的接口。

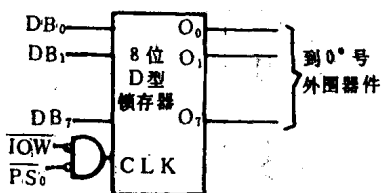


图1.12 8位输出口

微处理器向外输出数据的时间间隔是很短的(约 $0.5 \sim 1.0 \mu s$)。外部设备所需数据的持续期总是大于此间隔, 所以输出口必需用锁存器。图1.12示出用8位D锁存器组成的输出口。和输入口相反, 这里输入是数据总线, 输出引线和外设相连。微处理器执行输出指令时, 先使端口选择信号 $\overline{PS}_0$ 有效, 接着

写控制信号 $\overline{IOW}$ 有效, 此两信号“非与”将微处理器放到总线上的数据进行锁存。锁存的数据提供外设使用。输出口中不需要三态电路, 因为总线允许同时有多多个目的。图1.12和图1.11使用相同的端口选择信号 $\overline{PS}_0$, 虽然端口地址相同, 是输出操作还是输入操作, 由不同的控制信号($\overline{IOW}$ 和 $\overline{IOR}$)进行区别, 不会造成传输上的混乱。

1.4 状 态 口

外设数据准备好, 可以输入给微处理器, 微处理器如何能知道呢? 外设准备好接受微

处理器的输出数据，微处理器又如何知道呢？要解决上述问题，在数据传送中得进行状态信息交换。图1.13示出状态输入端口，从外设向微处理器输入状态信息。状态口也是输入口，不过输入的是状态信息，而不是外设的数据。状态口（图1.13）和输入口（图1.11）结合使用，可以解决较复杂的输入问题。现以0*端口为例说明全过程如下：

（1）外设提供的数据送到输入口输入（图1.11中的 $DB_0 \sim DB_7$ ）。

（2）外设提供钟控RDY₀脉冲，使数据锁存入D型锁存器（图1.11）。

（3）同一钟控脉冲RDY₀使状态口的RS触发器置位，记存外设的状态，在图1.13中用置“1”表示“准备好”。

（4）微处理器使 $\overline{IOR}$ 和 $\overline{PS}_0$ 有效，输入状态信息。

（5）微处理器识别输入的状态信息。

（6）微处理器使 $\overline{IOR}$ 和 $\overline{PS}_0$ 有效，从端口读入已锁存的外设数据（图1.11）。

（7）数据读入微处理器后，RESET（图1.11）信号有效，使RS触发器复位（图1.13），以准备下次记存状态信息。

假若外设提供的状态信号的持续期足够长，由微处理器直接检测不成问题，RS触发器可以省去，如图1.13中的RDY₀~RDY₇。

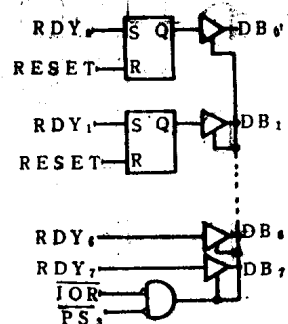


图1.13 状态口

1.5 控制口

有些外围设备需要控制信号，微处理器可以输出它所要求的控制信号。输出控制信号给外设的接口电路称控制口。图1.14示出多数情况适合的控制口。从图看出，它和图1.12的输出口电路相同，只是输出连接8条控制线。控制和传输的着眼点不同，前者要求对每条控制线提供相应的时间函数，而后者要求8条线的编码（8位码）。对一条控制线来说（例如CONTROL₀），微处理器先输出 $DB_0 = 1$ ，再输出 $DB_0 = 0$ ，则在CONTROL₀上产生一个脉冲。若输出“1”后等待一段时间再输出“0”，或连续输出“1”再输出“0”，即能产生具有一定宽度的脉冲。对8条控制线来说，微处理器连续输出与8个时间函数对应的8位字，就能在控制口输出产生所要求的8个控制信号。

若外设要求微秒级的控制脉冲，图1.14中的D型锁存器可用与门电路代替，如图1.15。

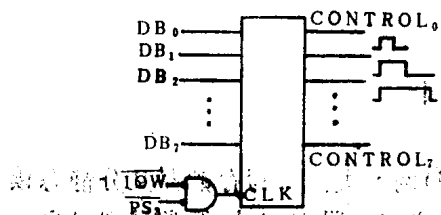


图1.14 锁存型控制口

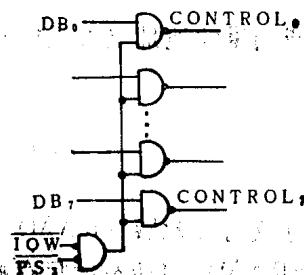


图1.15 非锁存型控制口

现举一例，说明复杂的外设如何通过控制口、状态口和数据口在微处理器控制下进行

工作。图 1.16 示出微处理器控制的模-数转换器 (A/D) 电路图, 它的工作顺序如下:

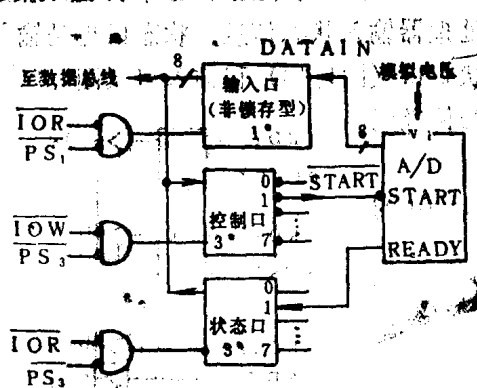


图 1.16 从 A/D 输入数据

(1) 微处理器通过控制口输出, 控制第一位线 (图中 3[#] 端口的 1)。这使控制线 START 成为有效, 启动 A/D 开始转换。转换有一过程, 长短取决于 A/D 的型号, 总之, 微处理器要等待。

(2) 微处理器读入状态位 READY, 检查是否完成, 若未完成, 继续检测。

(3) 当转换过程结束, A/D 发出 READY 有效信号。

(4) 微处理器检测到 READY 信号后, 通过端口 1[#] 读入数据 (对应输入直流电压值的数字量)。

1.6 中断向量输入

一种和微处理器数据总线挂接的输入口如图 1.17 所示, 它是按照 CPU 的中断操作时序, 在中断响应信号 INTA 有效时将一组数据送到数据总线上, 由微处理器读入, 作为中

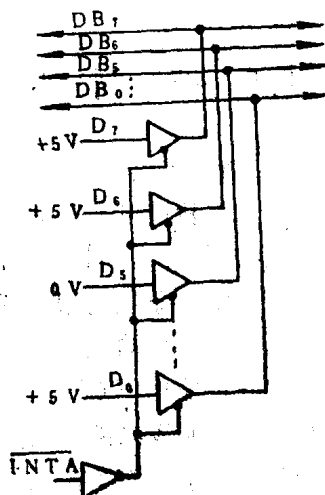


图 1.17 中断向量输入

断向量或中断向量的地址 (参见 1.9 节)。

1.7 存储器接口

存储器分读写存储器 (RAM) 和只读存储器 (ROM) 两大类。读写存储器又分静态读写存储器 (SRAM) 和动态读写存储器 (DRAM)。只读存储器有多种, 但常见到的也分两种: 掩膜 ROM 和可擦可编程 ROM (EPROM)。DRAM 留在下章研究, 本章讨论 ROM、EPROM 和 SRAM 的接口。

1.7.1 ROM 接口

掩膜 ROM 在工厂制造过程中已将信息存入，封装后不能改变，用户只能读出，不能修改。ROM 芯片对外的引脚有数据线、地址线和片选 $\overline{CS}$ 控制。为了传送数据，数据线引脚和 CPU 数据总线连接。为了选取存储单元，地址线引脚和 CPU 地址总线连接。ROM 在数据传输中是源，不能总占据数据线，只有当 CPU 需要它将数据送上数据总线时它才占据数据总线，所以有片选控制端 $\overline{CS}$ 。它控制 ROM 内部的输出三态门，当 $\overline{CS}$ 有效时，三态门被使能，当 $\overline{CS}$ 无效时它的数据总线引脚呈高阻态。当接口时，此 $\overline{CS}$ 应该等于存储器选择信号 $\overline{MS}$ 和存储器读控制信号 $\overline{MEMR}$ 的“或”，如图 1.18（图中地址线省略）。 $\overline{MS}$ 由地址线译码决定（参见下节）。图 1.18 的连接可以实现：当微处理器将存储器芯片选中并进行读操作时，存储器芯片才将数据送上总线，其余时间它与数据总线分离（高阻态）。

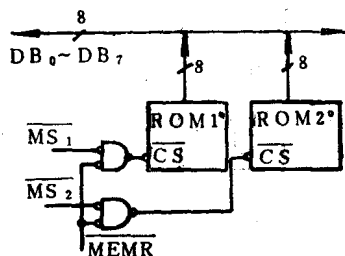


图 1.18 ROM 接口

1.7.2 EPROM 接口

EPROM 是这样一种存储器芯片，即：它在微处理器系统中完全起掩膜 ROM 的作用，但是它存储的信息可由用户用编程器（或称 EPROM 写入器写入）。用户也可用一种紫外线擦抹器将已存的信息抹去，进行改写，使用比较灵活方便。表 1.1 给出 EPROM

表 1.1 EPROM 2716 模式选择

模式	脚	$\overline{CE}/\text{PGM}$ (18脚)	$\overline{OE}$ (20脚)	V_{PP}	输出
读		低	低	+5V	输出
未选中		无关	高	+5V	高阻
功率下降		高	无关	+5V	高阻
编程		TTL电平正脉冲50ms	高	+25V	输入

2716 的 4 种工作模式，2716 有三个控制引脚 $\overline{CE}/\text{PGM}$ ($\overline{CS}/\text{PGM}$)、 $\overline{OE}$ 和 V_{PP} 。

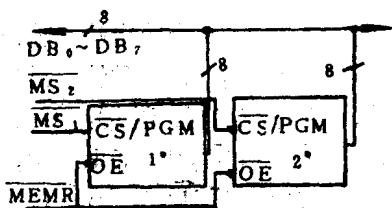


图 1.19 EPROM 2716 接口图

在微处理器系统中应用时 V_{PP} 固定为 5V。当微处理器进行读操作时， $\overline{CE} = 0$ 和 $\overline{OE} = 0$ ，EPROM 输出数据，在其余时间它应当处于高阻状态，与数据总线分离。后一种状态有两种模式可选，一般选功率下降模式，功率下降 75%（从 525mW 降至 132 mW）， $\overline{CE} = \text{TTL}$ 高电平。按这种考虑的接口方法如图 1.19 所示。 $\overline{CE}$ 引线接存储器选择信号 $\overline{MS}$ ， $\overline{OE}$ 接存储器读控制信号 $\overline{MEMR}$ 。

以上 EPROM 接口方法不是唯一的，因为不同型号的 EPROM 的控制信号可能不同，需要进行具体分析，上例仅提供考虑的原则。

1.7.3 SRAM接口

RAM 是读写存储器,当被微处理器读时,它是数据源,当被写时,它变成数据的目的。因此,它除了片选 $\overline{CS}$ 控制信号外,还应当有一个控制数据方向的信号 $\overline{WE}$ (WRITE)

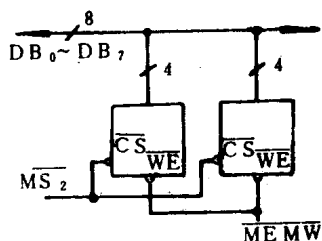


图 1.20 RAM 2114 接口

或 $\overline{RD}/\overline{WE}$ (READ/WRITE), 总共有两个控制引脚。图 1.20 示出 RAM 2114 的接口电路。 $\overline{CS}$ 接由处理器地址决定的存储器选择信号 $\overline{MS}$, $\overline{WE}$ 接处理器的存储器写控制信号 $\overline{MEMW}$ 。当微处理器寻址此 RAM 时, $\overline{MS} = 0$, 若是进行读操作, $\overline{MEMW} = 1$, RAM 内部输出三态门被使能, 数据送上 CPU 数据总线, 若是进行写操作 $\overline{MEMW} = 0$, 输入三态门被使能, CPU 总线上的数据送入 RAM 内部。2114

是 4×1024 位, 它只有 4 个数据线引脚。在图 1.20 中, 用二片 2114 作 8×1024 位存储器, 一片 2114 接高 4 位数据线, 另一片接低 4 位数据线。

1.8 地址译码器

地址译码器监测 CPU 的地址总线和控制总线, 产生所需要的端口选择信号和存储器选择信号。已有多种集成译码器, 再加适当的门电路进行组合, 以上功能是不难实现的。我们用 74LS138 为例来说明地址译码器的实现。表 1.2 给出了 74LS138 的真值表。

表 1.2 74LS138 真值表

G_1	$\overline{G_2A}$	$\overline{G_2B}$	C	B	A	输出
1	0	0	0	0	0	$\overline{Y_0} = 0$ 其余为 1
1	0	0	0	0	1	$\overline{Y_1} = 0$ 其余为 1
1	0	0	0	1	0	$\overline{Y_2} = 0$ 其余为 1
1	0	0	0	1	1	$\overline{Y_3} = 0$ 其余为 1
1	0	0	1	0	0	$\overline{Y_4} = 0$ 其余为 1
1	0	0	1	0	1	$\overline{Y_5} = 0$ 其余为 1
1	0	0	1	1	0	$\overline{Y_6} = 0$ 其余为 1
1	0	0	1	1	1	$\overline{Y_7} = 0$ 其余为 1
其它状态			x	x	x	全输出为 1

图 1.21 示出 74LS138 的引脚图, 它有 A、B、C 三个输入端, $\overline{Y_0} \sim \overline{Y_7}$ 8 个输出端。输出

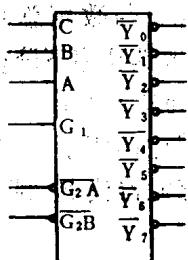


图 1.21 74LS138 引脚图

低电平有效。A、B、C 三端有 8 种组合, 每一种情况对应于一个输出端为低电平。此外, 还有三个控制端: G_1 、 $\overline{G_2A}$ 、 $\overline{G_2B}$, 只有 G_1 为高电平、 $\overline{G_2A}$ 和 $\overline{G_2B}$ 为低电平时, 芯片才允许工作, 否则芯片输出为高电平, 与 A、B、C 端无关。一般称它为 3-8 译码器。用 3-8 译码器实现地址译码器的例子如图 1.22 所示。

图 1.22(a) 是存储器地址译码器, 图 1.22(b) 是 I/O 端口地址译码器。它们所用的控制信号不同, 这是因存储器和 I/O 公用地址总线进行寻址, 为

了区别 CPU 当前是和存储器交换数据还是和 I/O 端口交换数据, 必需用不同的控制信号参加译码。当 Z-80 执行存储器访问时, $\overline{\text{MREQ}} = 0$ ($\text{IORQ} = 1$), 而执行 I/O 访问时, $\text{IORQ} = 0$ ($\overline{\text{MREQ}} = 1$)。

图 1.22 (a) 的译码器可以产生 8 个存储器选择信号, 每个存储器选择信号在 2 K 的

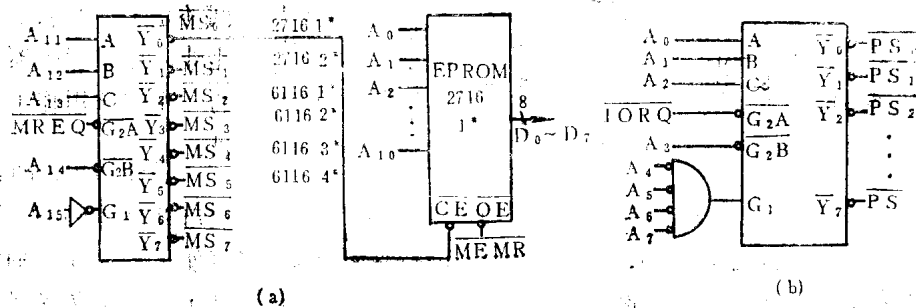


图 1.22 用 74LS138 实现的地址译码器

地址变化范围内, 都可保持低电平。换句话说它可作 2 K 字节存储器芯片的片选信号。EPROM 2716 的存储容量为 2 K 字节, RAM 6116 的存储器容量也是 2 K 字节。表 1.3 示出 0000H 至 07 FFH 2 K 地址区间的地址线状态表。从表看出, 在区间 A_{15} 至 A_{11} 都为零, 而 A_{10} 至 A_0 由全“0”变至全“1”。 A_{15} 至 A_{11} 保持零, 使 $\overline{\text{MS0}} = 0$, 选中 2716 1*。而 A_{10} 至 A_0 由全“0”变至全“1”, 正好寻址 2716 1* 的 2 K 个存储单元。

表 1.3 2K 地址范围 0000~07FFH 内地址线状态说明

十六进制 地址范围	地址线状态											
	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4
0 0 0 0	0	0	0	0	0	0	0	0	0	0	0	0
0 7 FF	0	0	0	0	0	1	1	1	1	1	1	1

由图 1.22 (a) 地址译码器产生的存储器地址空间示于表 1.4。图 1.22 (b) 的地址译码器产生 8 个 I/O 口选择信号, 可以选择 8 个 I/O, 其端口地址也见表 1.4。

表 1.4 图 1.22 地址译码器的地址空间

存储器地址空间			I/O 端口地址空间		
地址范围	选择信号	芯片连接例	端口地址	选择信号	端口连接例
0000~07FFH	$\overline{\text{MS}}_0$	2716 1*	00H	$\overline{\text{PS}}_0$	输入/输出
0800~0FFFH	$\overline{\text{MS}}_1$	2716 2*	01H	$\overline{\text{PS}}_1$	输入
1000~17FFH	$\overline{\text{MS}}_2$	6116 1*	02H	$\overline{\text{PS}}_2$	锁存型输入
1800~1FFFH	$\overline{\text{MS}}_3$	6116 2*	03H	$\overline{\text{PS}}_3$	状态口/控制口
2000~27FFH	$\overline{\text{MS}}_4$	6116 3*	04H	$\overline{\text{PS}}_4$	备用
2800~2FFFH	$\overline{\text{MS}}_5$	6116 4*	05H	$\overline{\text{PS}}_5$	备用
3000~37FFH	$\overline{\text{MS}}_6$	备用	06H	$\overline{\text{PS}}_6$	备用
3800~3FFFH	$\overline{\text{MS}}_7$	备用	07H	$\overline{\text{PS}}_7$	备用

到目前为止, 我们把 I/O 端口和存储器分开处理。这是 Z-80 的标准工作方式——隔离 I/O 结构。微处理器今天有多种, 按对 I/O 处理的方法不同分为两大类: 隔离 I/O 结构和存储器映象 I/O 结构。存储器映象 I/O 的处理方法是把 I/O 端口看作一个存储单元, 用存储器访问指令去访问 I/O 端口。R6502 就是这类微处理器。Z-80 属隔离 I/O 结构,