牛余朋　蔡艳平　成　曙◎编著

- 面向功能复杂的51单片机并行总线设计
- 重点详解，难点剖析，案例导引，全书仿真

单片机并行总线开发及模块设计

清华大学出版社

单片机并行总线开发及模块设计

牛余朋　蔡艳平　成　曙　编著

清华大学出版社

北　京

内 容 简 介

本书以 51 单片机为例，深入探讨了 51 单片机外部并行总线扩展电路的设计原理，并详细介绍了单片机与各种常见外部设备并行总线设计的开发实例。

本书主要解决单片机外围存储器并行总线设计、开关量输入/输出并行总线设计、A/D 和 D/A 并行总线设计、液晶并行总线设计、键盘并行总线设计、实时时钟并行总线设计、可编程并行接口芯片设计等难点问题。在详细讲解单片机并行总线开发原理的基础上，对电子工程师设计产品经常用到的模块实例进行了全面、系统的分析和仿真运行。书中的所有实例都可以直接应用于实际项目开发，从而加快开发者的学习速度和产品设计速度。

本书的工程应用性较强，对于单片机初学者（尤其是在校学生）及单片机工程师都会有较大程度的启发，本书可作为在校学生的学习教材，也可作为从事单片机相关工作人员的参考资料。

图书在版编目（CIP）数据

单片机并行总线开发及模块设计/牛余朋，蔡艳平，成曙编著. —北京：清华大学出版社，2015
ISBN 978-7-302-41014-0

I. ①单… II. ①牛… ②蔡… ③成… III. ①单片微型计算机-电路设计 IV. ①TP368.1

中国版本图书馆 CIP 数据核字（2015）第 169375 号

责任编辑：贾小红
封面设计：刘　超
版式设计：魏　远
责任校对：王　云
责任印制：刘海龙

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：北京鑫海金澳胶印有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　印　张：13.75　　字　数：323 千字
版　　次：2015 年 10 月第 1 版　　印　次：2015 年 10 月第 1 次印刷
印　　数：1～3000
定　　价：36.00 元

产品编号：058935-01

前　言

近年来，随着单片机技术的不断提高，功能的不断完善，其应用也日趋成熟，应用领域也日趋扩大，特别是在工业测控、尖端武器、电子仪器、日用家电领域，更是因为有了单片机而生辉增色。因此，会用单片机也已经成为嵌入式电子产品开发工程师的基本技能。在设计嵌入式系统时有两种不同方案，即总线式设计方案和非总线式设计方案。

所谓总线式设计方案，就是利用单片机的读写外部 RAM 功能，将要设计的外部设备（如键盘、液晶显示器等）统统挂载到单片机总线上，使其统一按类似读写外部 RAM 功能的指令方法进行操作；而所谓非总线式设计方案，则不利用单片机的读写外部 RAM 功能，而是直接利用单片机 I/O 口读写方式进行外部设备的读写，也就是说单片机 I/O 口模拟单片机总线的时序来操作外部设备，所以业界工程师也把这种方式称为模拟 I/O 方式。

对于总线式设计方案和非总线式设计方案，这两种方案哪一种好呢？其实，这两种方案各有利弊。对于总线式设计方案来说，优点就是能够充分发挥单片机的总线读写功能，对于功能复杂的系统开发有利，且易于日后的升级和扩展，缺点是单片机读写总线时必定要产生一定的总线时间延迟，这样对于低速的单片机来说，如果设计的系统非常庞大，则需要设计人员考虑实时性要求；对于非总线式设计方案来说，最大的优点就是灵活性强，这种方案在符合设计原则的前提下，可以根据设计者的开发习惯自由选择单片机端口进行外围设备的设计，缺点是功能复杂的系统开发时，控制复杂，设计困难，且升级需要重新设计电路图。

本书主要对功能复杂的单片机系统存在的并行总行设计重点和难点进行介绍，主要解决单片机外围存储器并行总线设计、开关量输入/输出并行总线设计、A/D 和 D/A 并行总线设计、液晶并行总线设计、键盘并行总线设计、实时时钟并行总线设计、可编程并行接口芯片设计等难点问题。本书以市场占有率较高的 51 单片机为例展开介绍，深入探讨了 51 单片机的外部并行总线扩展电路的设计原理，详细介绍了单片机与各种常见外部设备的并行总线设计开发实例，介绍了以 ATMEGA128 为例的 AVR 单片机外部并行总线扩展的原理和设计方法，并且全部进行了实验仿真。

本书由牛余朋、蔡艳平执笔，参加编写的人员还有成曙、姚良、林旭泽、曲从善、曾轩、元原，在此一并表示感谢。另外最值得感谢的是清华大学出版社的编辑，没有他们，本书不可能如期顺利出版。

由于编者水平有限，书中难免存在疏漏之处，殷切希望广大读者批评指正。

编　者

目　录

第 1 章 单片机基础知识

1.1 概述

1.1.1 单片机发展历程

单片机专业名称为 Micro Controller Unit（微控制器件），是由 Intel 公司发明的，最早的系列是 MCS-48，后来有了 MCS-51。常说的 51 系列单片机就是 MCS-51（Micro Controller System），这是一种 8 位的单片机。后来 Intel 公司把它的核心技术转让给世界上很多小公司，所以就有许多公司生产 51 系列兼容单片机，例如，飞利浦的 87LPC 系列、华邦的 W78 系列、达拉斯的 DS87 系列、现代的 GSM97 系列等。目前在我国比较流行的就是美国 ATMEL 公司的 89C51，它是一种带 Flash ROM 的单片机，本书的内容就是基于该型号的单片机来开展相关设计和实验仿真的。讲到这里，也许有的读者会有疑问，为什么平时在各种书上看到全是讲解 8031、8051 等型号的单片机，二者又有什么不同呢？其实 8031、8051 与 89C51 同属于一个系列，只是 89C51 的单片机更新型一些。目前 89C51 已经被 89S51 代替，89S51 兼容 89C51，同时近几年也出现了像宏晶公司研发生产的 STC89C 系列的 51 内核的单片机，该单片机最大的特点就是可以利用串口线在线下载程序，使用非常方便，因此该系列单片机近几年占据了国内 51 单片机的大部分市场。

1.1.2 几种常见的单片机

除了上述介绍的单片机，还经常在各种刊物上看到 AVR 系列和 PIC 系列单片机，这么多的单片机，应该先学哪一种呢？在没有学习单片机之前，这是一个令很多初学者非常困惑的问题。

AVR 系列单片机是 ATMEL 公司生产的一种 8 位单片机，采用的是 RISC 精简指令集单片机结构，所以其技术和 51 系列有所不同，开发设备和 51 系列也是不通用的，AVR 系列一条指令的运行速度可以达到纳秒级，即每秒 1000000000 次，是 8 位单片机中的高端产品，由于其出色性能，目前应用范围越来越广，大有取代 51 系列的趋势，所以学完 51 系列，还有必要学习 AVR 系列。PIC 系列单片机则是美国 MICROCHIP 公司生产的另一种 8

位单片机，采用的也是 RISC 的指令集，其指令系统和开发工具与 51 系列更是不同，但由于其价格低和性能出色，目前用户越来越多，国内也有很多公司在推广此系列，不过 PIC 的影响力远没有 51 系列大，所以作为初学者，51 系列仍然是首选。以上几种只是比较常见的系列，日常应用中还有许多其他公司生产的各种各样的单片机，例如，Motorola 的 MC68H 系列、TI 的 MSP430C 系列、德国的西门子 PLC 等，都有各自的结构体系，并不与 51 系列兼容，这里不再深入介绍，感兴趣的读者可在入门后自己研究，下面介绍 51 系列单片机的结构及组成。

1.1.3 单片机的结构及组成

单片机到底是什么，究竟能做什么呢？其实单片机就是一种能进行数学和逻辑运算，根据不同使用对象完成不同控制任务的面向控制而设计的集成电路，就像在计算机中可以用不同的软件在相同的硬件上实现不同的功能。单片机其实也是如此，同样的芯片可以根据不同的要求做出截然不同的产品，只不过计算机是面向应用的，而单片机是面向控制的，例如，控制一个指示灯的亮和灭，控制一台电机的启动和停止等。那么单片机的内部究竟由哪些部件组成呢？大家都知道计算机中有很多的部件，如 CPU（中央处理器）、RAM（内存条）、ROM（程序存储器）、输入/输出设备（并行口/串行口）等，在单片机中这些部件也都具备，并且全部集成在一块芯片上，这就是单片机名称的由来。如图 1-1 所示为单片机的典型组成结构。单片机提供静态逻辑操作功能，工作频率可以低至 0Hz，支持两种软件编程的节约电源管理模式，一种是空闲模式，一种是电源关闭模式。在空闲模式下，单片机的 CPU 停止工作，而内部数据存储器、定时器、串口和中断系统仍然工作。在电源关闭模式下，单片机保存数据存储器的内容，晶体振荡器工作，其余的部分停止工作，直到有中断或硬件复位时，单片机芯片的其他部分才能工作。如图 1-2 所示是 AT89S51 的内部结构图。

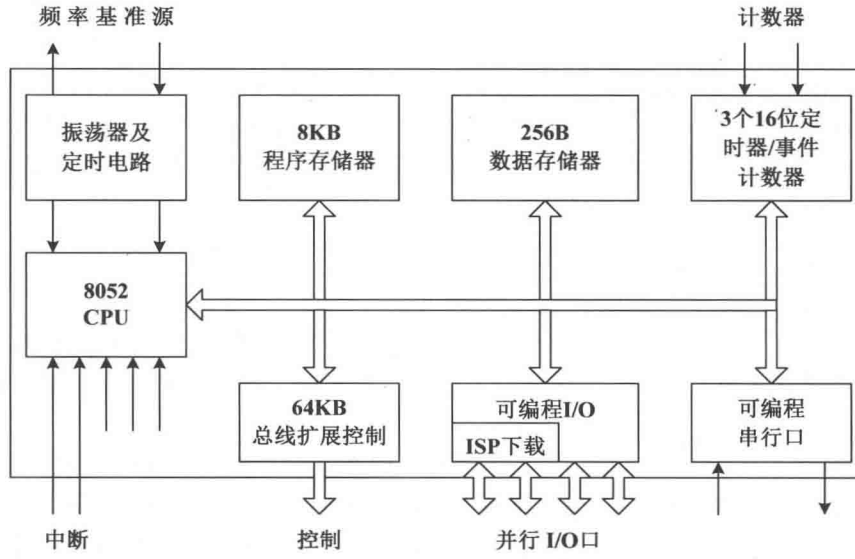

图 1-1 单片机的典型组成结构

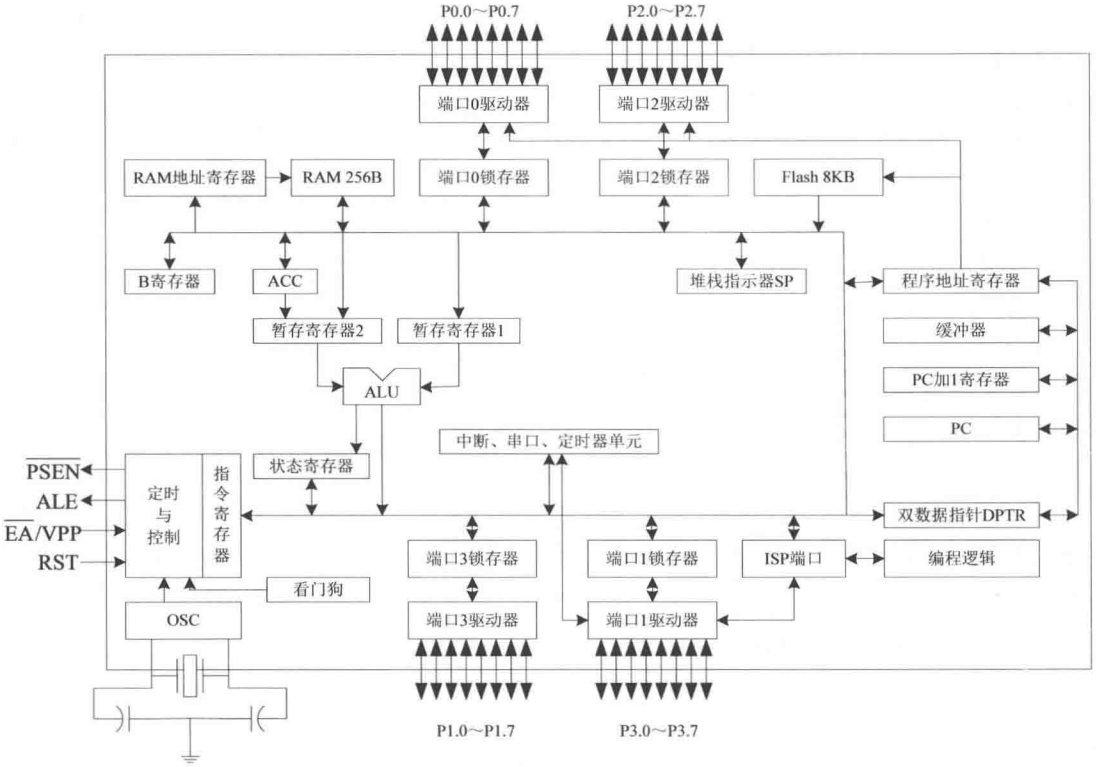

图 1-2　单片机 AT89S51 的内部结构图

1.2　数的进制及位和字节的含义

1.2.1　数制及其转换

1．3 种基本数制

数制指的是数据的表现形式。在计算机中常用的数制有二进制数、十进制数和十六进制数。无论是什么数制，将这个数制所包含的数码个数称为基，将各个数字所具有的数值称为权。

（1）十进制数（Decimal）

十进制数是日常生活中最为常用的数制形式。十进制数的基为 10，其包含 10 个数码，即 0、1、2、3、4、5、6、7、8、9。十进制数的权是以 10 为底的幂，每个数所处位置不同，则其代表值不同。前一个数的权值为其相邻后一个数权值的 10 倍。

例如，一个十进制数为 2345.678D，则其代表的数值为：

$2345.678D=2\times10^3+3\times10^2+4\times10^1+5\times10^0+6\times10^{-1}+7\times10^{-2}+8\times10^{-3}$

从左到右各位代表的权值为：

10^3　　10^2　　10^1　　10^0　　10^{-1}　　10^{-2}　　10^{-3}

也就是平常所说的千、百、十、个、0.1、0.01、0.001。

（2）二进制数（Binary）

二进制数是计算机中的基本数据形式，所有数据在计算机中都以二进制数形式进行存储和运算。二进制数的基为 2，在二进制数中只包含 0 和 1 两个数码。二进制数的权是以 2 为底的幂，每个数所处位置不同，则其代表值不同。前一个数的权值为其相邻后一个数权值的 2 倍。

例如，一个二进制数为 1011.1001B，则其代表的十进制数值为：

$$1001.1011B=1\times 2^3+0\times 2^2+0\times 2^1+1\times 2^0+1\times 2^{-1}+0\times 2^{-2}+1\times 2^{-3}+1\times 2^{-4}$$
$$=8+0+0+1+0.5+0.25+0.125+0.0625$$
$$=9.9375$$

从左到右各位代表的权值为：

2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
8	4	2	1	1/2	1/4	1/8	1/16

二进制数之所以能够成为计算机工作基本数据形式，这是由计算机中的电路工作原理决定的。计算机中的电路通常只有两种状态，在二进制数中可以用 1 代表电路中的高电平，用 0 代表电路中的低电平；或者用 1 代表电路中的导通，用 0 代表电路中的截止。采用二进制数可以方便地对电路进行计数工作。

（3）十六进制数（Hexadecimal）

十六进制数的基为 16，在十六进制数中包含了 16 个数码，各个数码表示如下：

0　1　2　3　4　5　6　7　8　9　A　B　C　D　E　F

十六进制数的权是以 16 为底的幂，与二进制数相同，每个数所处位置不同，则其代表值不同。前一个数的权值为其相邻后一个数权值的 16 倍。

例如，一个十六进制数为 F72.C4H，则其代表的十进制数值为：

$$F72.C4H=15\times 16^2+7\times 16^1+2\times 16^0+12\times 16^{-1}+4\times 16^{-2}$$
$$=3954.765625$$

从左到右各位代表的权值为：

16^2	16^1	16^0	16^{-1}	16^{-2}

相对于二进制数而言，十六进制数能够用较少的位数表现较大的数值，便于书写和记忆。由于二进制数与十六进制数的转换非常简单，所以在计算机中的数值常常采用十六进制数来表示。在汇编语言中，为了将十六进制数中的 A、B、C、D、E、F 与字母区分开来，书写时常常会在其前面加一个 0。例如 F7H 写作 0F7H。

2．数制间的转换

由于不同数制的特性不同，常常需要将一个数转换为十进制数来理解数值大小，将一个数转换为十六进制数来编写程序，将一个数转换为二进制数来理解对应数据在计算机中的含义。下面介绍计算机中不同数制的转换方法。

（1）转换为十进制数

在前面介绍数制的过程中已经介绍了二进制数和十六进制数转换为十进制数的方法。将一个二进制数或十六进制数转换为十进制数只需将该数各个数位的基与权相乘累加即可。例如：

$10110101B=1\times2^7+0\times2^6+1\times2^5+1\times2^4+0\times2^3+1\times2^2+0\times2^1+1\times2^0$

$=181$

$B5H=11\times16^1+5\times16^0=181$

（2）十六进制数与二进制数的转换

由于十六进制数的基 16 是二进制数基 2 的 4 次方，所以二进制数与十六进制数的转换十分简单。

十六进制数转换为二进制数时，只需从右至左将各个位上的数以二进制数表示出来即可。二进制数转换为十六进制数时，整数部分需要从右至左依次将 4 个二进制数表示成一个十六进制数，左侧不满 4 个时在前面补 0 到 4 个即可。小数部分自左向右每 4 位一组，最后不满 4 位的在后面补 0，写出对应的十六进制数即可。如表 1-1 所示为 0～15 的十进制数、二进制数、十六进制数的转换关系表。

表 1-1　不同数制的数据转换表

十进制数 D	二进制数 B	十六进制数 H	十进制数 D	二进制数 B	十六进制数 H
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	10	1010	A
3	0011	3	11	1011	B
4	0100	4	12	1100	C
5	0101	5	13	1101	D
6	0110	6	14	1110	E
7	0111	7	15	1111	F

例如，一个二进制数 10010110101B 要转换为十六进制数，其过程如下：

10010110101B=0100 1011 0101=4B5H

一个十六进制数 A3FBH 转换为二进制数的过程如下：

A3FBH=1010 0011 1111 1011B

（3）十进制数转换为二进制数或者十六进制数

将一个十进制数转换为二进制数时，对于十进制数整数部分采用“除 2 取整”的方法，小数部分采用“乘 2 取整”的方法。十进制数转换为十六进制数时可以采用类似的“除 16 取整”以及“乘 16 取整”的方法。由于十六进制数与二进制数之间的转换十分简单，在十进制数转换为十六进制数时，也可以先转换为二进制数继而转换为十六进制数。

下面以一个十进制数 78.8125D 为例来介绍“除 2 取整”以及“乘 2 取整”方法的使用。

78.8125D 整数部分为 78，转换为二进制数时将此数不停除 2 直到商为 0 为止，然后将所有余数逆向整合即可得到想要的二进制数。

2∟ 78 ……余数为 0　低位

2∟ 39 ……余数为 1

2∟ 19 ……余数为 1

2∟ 9 ……余数为 1

2∟ 4 ……余数为 0

2∟ 2 ……余数为 0

2∟ 1 ……余数为 1　高位

如上所示将余数逆向整合得到 78D=1001110B。

小数部分 0.8125 转换为二进制数时将该数的纯小数部分不停乘以 2 直至乘积为 1 为止，将所有结果整数部分一次组合即可得到想要的二进制数。

0.8125×2=1.625 ……整数部分为 1　高位
0.625×2=1.25 ……整数部分为 1
0.25×2=0.5 ……整数部分为 0
0.5×2=1 ……整数部分为 1　低位

如上所示将整数顺次整合得到 0.8125D=0.1101B。

将小数部分以及整数部分整合得到 78.8125D=1001110.1101B。

1.2.2 数和物理现象的关系

在数字电路中，可以用一盏灯的亮和灭来表示电平的高和低，即用“1”表示高电平（亮），用“0”表示低电平（灭），如果现在有两盏灯，那么会有几种状态呢？如表 1-2 所示为两盏灯的亮灭与二进制数之间的对应关系。

表 1-2 两盏灯的亮灭与二进制数之间的对应关系

第 一 盏 灯	第 二 盏 灯
0	0
0	1
1	0
1	1

两盏灯的组合可以表示 4 种状态，即 00，01，10，11。这样看来，灯的亮和灭这种物理现象同数字确实有着某种联系，如果把各状态按一定的规律排好，那么电平的高或低就可以用数字来表示了，换句话说，不同的数字可以代表不同数量灯的电平高或低，例如，0000，0001，0010，0011，0100，0101，0110，0111，1000，1001，1010，1011，1100，1101，1110，1111，这 16 种组合就可以代表 4 盏灯的不同状态。

1.2.3 位和字节的含义

在单片机中一盏灯（实际上是一根线）可看作一位，有 0 和 1 有两种状态，分别对应电平的低和高，是单片机最基本的数量单位，用 bit 来表示。8 盏灯（8 根线）有 256 种状态，这 8 盏灯（也就是 8 位）可称为一个字节，用 B（即 BYTE）表示。那么单片机是如何来储存这些数字所代表的字节的状态的呢？后文介绍存储器时将做详细说明。

1.3 51 单片机基本硬件结构

1.3.1 硬件结构

前面提到过单片机的内部结构是由 CPU、ROM、RAM 等组成，现在介绍外部引脚。如图 1-3 所示为单片机的引脚图，这就是实验中要用的 89C51 单片机的外部引脚图。如

表 1-3 所示为 89C51 单片机引脚分配表。

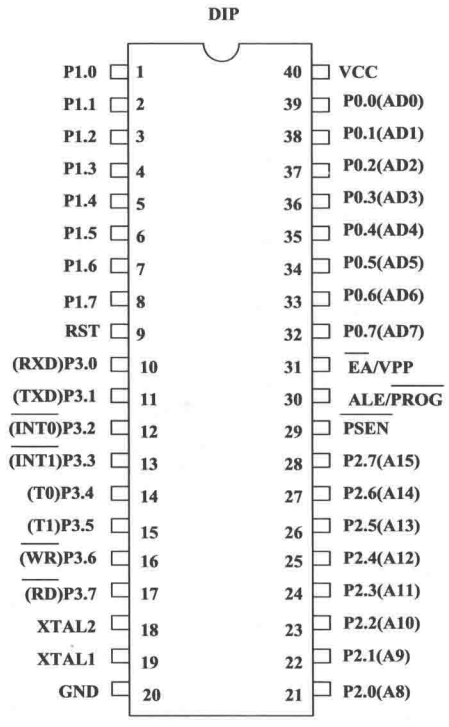

图 1-3　89C51 单片机的引脚图

表 1-3　89C51 单片机引脚分配表

引　脚　号	引 脚 名 称	说　　明
1～8	P1.0～P1.7	端口 P1
9	RST	复位信号输入端
10～17	P3.0～P3.7	端口 P3，该端口具备第二功能，详见 1.3.2 节
18	XTAL2	时钟振荡器输出端，内部振荡器输出端
19	XTAL1	时钟振荡器输入端，内部振荡器输入端
20	GND	电源地
21～28	P2.0～P2.7	端口 P2
29	$\overline{\text{PSEN}}$	外部程序存储器从程序存储器中取指令或读取数据时，该信号有效
30	ALE/$\overline{\text{PROG}}$	地址锁存信号访问外部存储器时，该信号锁存低 8 位地址；无 RAM 时，此引脚输出晶振的 6 分频信号
31	$\overline{\text{EA}}$/VPP	程序存储器有效地址，EA=1 时从内部开始执行程序；EA=0 时从外部开始执行程序
32～39	P0.7～P0.0	端口 P0
40	VCC	电源正

1.3.2　端口结构分析

从 1.3.1 节的硬件结构中可以看出，89C51 单片机总共有 4 组端口，P0、P1、P2 和 P3，

了解这 4 组端口的结构原理对于日后的编程会有很大的帮助，由于这 4 组端口结构不尽相同，下面分别介绍单片机总的 4 组端口。由于每组端口都是由 8 位组成，故在下面的讲解中，只以每组端口的其中一位来解释。

1．P0 口的结构及工作原理

P0 口字节地址为 80H，位地址 80H～87H。P0 端口 8 位中的一位结构图如图 1-4 所示。

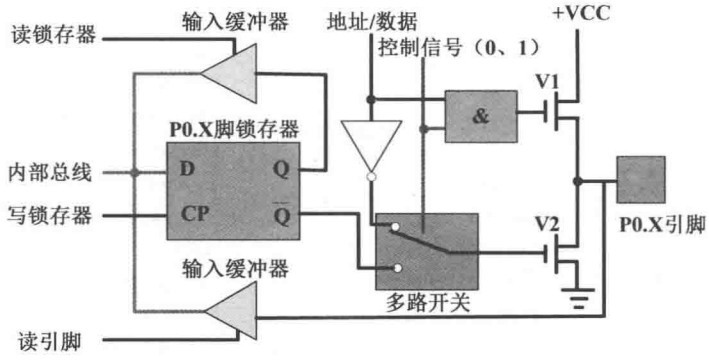

图 1-4　P0 端口位结构图

由图 1-4 可见，P0 端口由锁存器、输入缓冲器、多路开关、一个非门、一个与门及场效应管驱动电路构成。图 1-4 中标号为 P0.X 引脚的图标，表示引脚可以是 P0.0～P0.7 的任何一位，即在 P0 口有 8 个与图 1-4 所示相同的电路组成。下面先介绍组成 P0 口的每个单元部分。

（1）输入缓冲器

在 P0 口中，有两个三态的缓冲器，学过数字电路的读者都知道三态门有 3 个状态，即在其输出端可以是高电平、低电平，同时还有一种高阻状态（或称为禁止状态），图 1-4 中，上面一个是读锁存器的缓冲器，也就是说，要读取 D 锁存器输出端 Q 的数据，需要使读锁存器中这个缓冲器的三态控制端（图 1-4 中标号为“读锁存器”端）有效，下面一个是读引脚的缓冲器，要读取 P0.X 引脚上的数据，也要使标号为“读引脚”的三态缓冲器的控制端有效，引脚上的数据才会传输到单片机的内部数据总线上。

（2）D 锁存器

构成一个锁存器，通常要用一个时序电路（时序的单元电路内容请参考数字电路相关知识），一个触发器可以保存一位二进制数（即具有保持功能），在 51 单片机的 32 根 I/O 口线中，都是用一个 D 触发器来构成锁存器的。图 1-4 中的 D 锁存器，D 端是数据输入端，CP 是控制端（即时序控制信号输入端），Q 是输出端，$\overline{Q}$ 是反向输出端。

对于 D 锁存器来讲，当 D 输入端有一个输入信号，如果这时控制端 CP 没有信号（即时序脉冲没有到来），这时输入端 D 的数据是无法传输到输出端 Q 及反向输出端 $\overline{Q}$ 的。如果时序控制端 CP 的时序脉冲到达，这时 D 端输入的数据就会传输到 Q 及 $\overline{Q}$ 端。数据传送过来后，当 CP 时序控制端的时序信号消失时，输出端还会保持着上次输入端 D 的数据（即把上次的数据锁存起来）。如果下一个时序控制脉冲信号到来，这时 D 端的数据才再次传送到 Q 端，从而改变 Q 端的状态。

（3）多路开关

在 51 单片机中，当内部的存储器够用时（即不需要外扩展存储器时，这里讲的存储器包括数据存储器及程序存储器），P0 口可以作为通用的输入/输出端口（即 I/O）使用，对于 8031（内部没有 ROM）的单片机，或者编写的程序超过了单片机内部的存储器容量需要外扩存储器时，P0 口就作为地址/数据总线使用。那么这个多路选择开关就是用于选择是作为普通 I/O 口使用还是作为地址/数据总线使用的选择开关了。从图 1-4 可知，当多路开关与下端接通时，P0 口作为普通的 I/O 口使用；当多路开关是与上端接通时，P0 口作为地址/数据总线使用。

（4）输出驱动

从图 1-4 中可看出，P0 口的输出是由两个 MOS 管组成的推拉式结构，也就是说，这两个 MOS 管一次只能导通一个，当 Vl 导通时，V2 截止，当 V2 导通时，Vl 截止。

上面已对 P0 口的各单元部件进行了详细的讲解，下面研究一下 P0 口作为 I/O 口及地址/数据总线使用时的具体工作过程。

（1）作为 I/O 端口使用时的工作原理

P0 口作为 I/O 端口使用时，多路开关的控制信号为 0（低电平），如图 1-4 所示，多路开关的控制信号同时和与门的一个输入端相接，与门的逻辑特点是“全 1 出 1，有 0 出 0”，那么控制信号如果是 0，这时与门输出的也是一个 0（低电平），此时 Vl 管就截止，在多路控制开关的控制信号是 0（低电平）时，多路开关是与锁存器的 $\overline{Q}$ 端相接的（即 P0 口作为 I/O 口线使用）。

P0 口用作 I/O 口线，其由数据总线向引脚输出（即输出状态 Output）的工作过程：写锁存器信号 CP 有效，数据总线的信号的输出流程为锁存器的输入端 D→锁存器的反向输出 $\overline{Q}$ 端→多路开关→V2 管的栅极→V2 管的漏极→输出端 P0.X。前面已经介绍过，当多路开关的控制信号为低电平 0 时，与门输出为低电平，Vl 管是截止的，所以作为输出口时，P0 是漏极开路输出状态，类似于 OC 门，当驱动上接电流负载时，需要外接上拉电阻。如图 1-5 所示就是由内部数据总线向 P0 口输出数据的流程图。

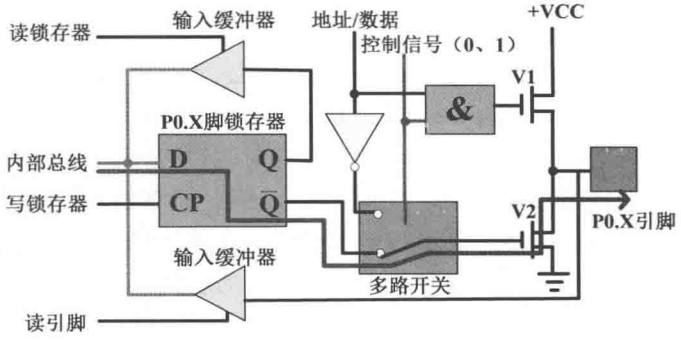

图 1-5　P0 口内部数据总线向引脚输出时的流程图

P0 口用作 I/O 口线，其由一引脚向内部数据总线输入（即输入状态 Input）的工作过程，数据输入时（读 P0 口）有以下两种情况：

第一种情况是读引脚，即读芯片引脚上的数据。读引脚数时，读引脚缓冲器打开（即