

家用电器单片机 控制系统的 制作与检修



主编：李建事 陈 刚

上海交通大学出版社

家用电器单片机控制系统 的原理与检修

李建事 陈 刚 主编
张志鸣 姚战民 邓小志 编

- 全自动洗衣机控制系统的检修
- 空调器控制系统的检修
- 电冰箱控制系统的检修
- 微波炉控制系统的检修
- 电饭煲控制系统的检修
- 抽油烟机制系统的检修
- 电消毒碗柜控制系统的检修
- 全功能电风扇控制系统的检修

上海交通大学出版社

**家用电器单片机
控制系统的原理与检修**

上海交通大学出版社出版、发行

上海市番禺路 877 号 邮政编码 200030

全国新华书店经销

上海交通大学印刷厂·印刷

开本:787×1092(毫米)1/16 印张:13 插页:2 字数:320000

版次:1998 年 3 月 第 1 版 印次:1998 年 3 月 第 1 次

印数:1—6000

ISBN 7—313—01946— 7/TP·347

定价:13.80 元

序 言

单片机在家用电器中的应用已经很普及,从摄像机、影碟机、电视机,到最简单的电风扇,都可以用单片机处理信号,并对机器实现自动化、智能化的控制。

本书面向广大无线电爱好者及家电维修工,故本书不涉及单片机对声像信号的处理,因为这部分电路相当复杂。本书只介绍如何利用单片机系统对家用电器进行控制,并介绍控制系统的检修方法。

本书是《家用电器单片机控制系统的原理与检修》的第一分册。该书的第二分册将介绍CD机、LD机、VCD机的原理与检修。第三分册将介绍彩电遥控系统的原理与检修。

本书由福州艺通电器有限公司李建事和西安交通大学计算机系微机教研组陈刚主编。参加编写的有福州洪山科技园区的张志鸣、福州艺通电器有限公司网络室姚战民、科技情报研究室邓小志。

限于水平,谬误之处,敬请指教。

编 者

1998年1月20日

目 录

第一章 微型计算机简介	1
第一节 存储器.....	2
第二节 微处理器 CPU	4
第三节 总线.....	6
第四节 输入/输出接口(I/O 口)	8
第五节 定时系统	11
第二章 单片机简介	13
第一节 单片机指令系统概述	13
第二节 程序设计概述	14
第三节 单片机的开发	17
第三章 全自动洗衣机内单片机系统的原理与检修	26
第一节 原理与检修综述	26
第二节 全自动洗衣机中 LY360 型电脑程控器的原理与检修	44
第三节 申花电脑全自动洗衣机原理与检修	51
第四节 全自动洗衣机中的单片机模糊控制系统的原理与检修	61
第四章 空调器内单片机系统的原理与维修	69
第一节 单片机对空调器的控制	69
第二节 空调器中 MC6805R3 单片机控制系统的原理与检修	70
第三节 空调器中 NEC75028CW-015 单片机控制系统的原理与检修	75
第四节 空调器中 μ PD75328 单片机控制系统的原理与检修	77
第五节 空调器中 8051 单片机控制系统的原理与检修.....	80
第六节 空调器中 8031 单片机控制系统的原理与检修.....	88
第五章 电冰箱内单片机系统的原理与检修	94
第一节 电冰箱中 8031 单片机控制系统的原理与检修.....	94
第二节 电冰箱中 8098 单片机控制系统的原理与检修.....	99
第三节 电冰箱中 MC6805R2/R3 单片机控制系统的原理与检修	104

第六章 厨房卫生间电器内单片机控制系统的原理与检修	107
第一节 微波炉内 MC6805R3 型单片机控制系统的原理与检修	107
第二节 飞跃 WP600 型微波炉内单片机控制系统的原理与检修	112
第三节 国产“新宝”牌(SAMPO)微波炉内单片机控制系统的原理与检修	119
第四节 电饭锅中 8031 单片机控制系统的原理与检修	122
第五节 抽油烟机内单片机控制系统的原理与检修	129
第六节 自动型抽油烟机 FEC242C 单片机控制系统的原理与检修	130
第七节 消毒碗柜单片机控制系统的原理与检修	132
第八节 玉环 CDR-30A 电热水器内单片机控制系统的原理与检修	134
第九节 电暖器内单片机控制系统的原理与检修	136
第七章 全功能电风扇内单片机系统的应用与维修	138
第一节 BA 系列电风扇控制集成电路	138
第二节 MC 系列电风扇控制集成电路	150
第三节 WT 系列电风扇控制集成电路	154
第四节 PC 系列电风扇控制集成电路	155
第五节 PT 系列电风扇控制集成电路	158
第六节 Z8600 在电风扇上应用	168
第八章 彩电内单片机控制系统的原理与检修	170
第一节 电视机控制原理简介	170
第二节 彩色电视机内的单片机控制系统简介	177
第三节 彩色电视机内单片机系统的控制原理	185
第四节 故障排除	192
第九章 家用录像机内单片机控制系统的原理与检修	195
第一节 录像机内单片机控制系统的工作原理	195
第二节 录像机控制系统故障检修	198

第一章 微型计算机简介

数字计算机由五个基本部分组成,即运算器、控制器、存储器、输入和输出设备。其中运算器与控制器通常合称为中央处理单元或中央处理器(CPU)。而把具有 CPU 功能的大规模集成电路称为微处理器。整个微机系统如图 1-0-1 所示。

微处理器中的运算器(CPU)能直接进行各种操作。它在控制器的控制下,通过算术逻辑单元(ALU)完成各种算术运算和逻辑运算。也就是说,它是计算机数据的变换部分。

控制器指挥和控制着计算机各个部分的工作。整个计算机系统是在“指令”的要求下,有步骤地运行。每个计算机都有自己所能接受的基本指令,这些指令构成指令系统。程序是由一条一条基本指令组成的,其指令顺序就决定了某特定问题的具体细节,这些指令存放在存储器(ROM)内。控制器按照程序所给出的顺序,依次取出指令,一步一步地分析每条指令,并相应地发出各种控制信号,使运算器、存储器等各部分自动而又协调地执行这些指令所规定的操作,从而使计算机能自动地完成各种运算任务。

存储器用来存放数据与指令,它由许多存储单元组成,每一个存储单元用以存放一个数据代码,为了区分不同的存储单元,把全部存储单元按一定顺序编号,这个编号就称作单元的地址。存储器可以按要求将外界送来的数据存(写)入指定的单元,也可以按要求将所指定单元内的内容读出,送到有关的电路。

计算机内的存储器分为半导体随机存储器(RAM)与只读存储器(ROM)两种,后者在计算机运行过程中只能读出它原来写入的信息,但断电后存储器内的信息仍保存着;而前者在计算机运行过程中可读可写,但断电后存储器内的信息就消失了。

计算机内部 CPU 与外部设备必须通过输入/输出接口(I/O)电路建立联系。因为外部设备输入到计算机的信号,计算机未必能识别,而计算机输出的信号未必能对外部设备进行控制,两者之间必须通过接口电路对信号进行变换才能达到互相识别的目的。

单片机芯片之间、部件之间,各部件与外部设备之间采用总线进行连接。总线实际就是许多信号线的集合,它能使数据的流通与信息的交换按一定的要求定向地进行。

总线分为内部总线、系统总线与外部总线。内部总线用于连接微机内部 CPU 与外围芯片;系统总线用于连接微机之间各插件板与系统板;外部总线用于连接微机与外部设备。不管是内部总线,还是系统总线、外部总线,其中每一部分还可以分为地址总线、数据总线与控制总线。

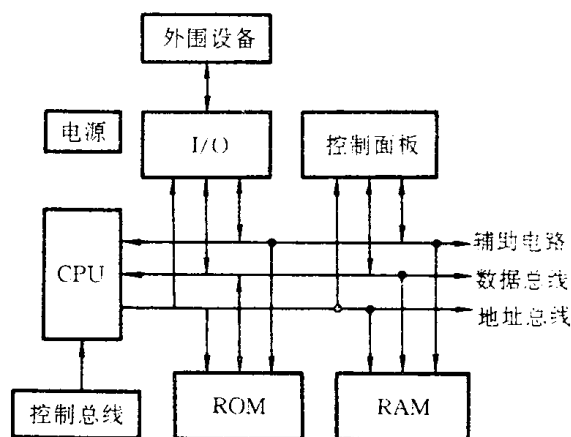


图 1-0-1 微机系统框图

第一节 存贮器

存贮器是存放数据与程序的部件。其数据或指令都用二进制代码表示。存贮器的基本功能是保存大量的代码,按需要取出来(读出)或把新的代码存进去(写入)。存贮器由许多存贮单元组成,每个存贮单元有一个编号称为单元地址,每个存贮单元通常存放在一个有独立意义的代码称为字,其代码的位数称字长。例如,某存贮器共有 4069(简称 4k)个单元,字长为 8 位,则说明这个存贮器容量为 4k,或表示为 4096×8 位。当某一代码写入某一单元时,首先要指出该单元的地址,然后在地址所指出的单元中写入代码。向一个存贮器存入一个数或取出一个数的时间叫做存取周期,它表示存贮器的工作速度。存贮器的容量与工作速度是微机性能的两个重要指标。

计算机内部的存贮器一般是半导体存贮器,而外部存贮器一般由磁介质组成。计算机内部的半导体存贮器分只读存贮器(ROM)与读写存贮器(RAM)两大类。

一、只读存贮器(ROM)

半导体只读存贮器(ROM)是指计算机在运行过程中只能读出,不能写入的存贮器。这种存贮器的内容在断电后不会消失,因此常用来存贮不变的程序与不变的数据。

ROM 的内部框图如图 1-1-1 所示,它主要有由地址译码器、存贮矩阵、输出缓冲器以及芯片选择逻辑组成。

图 1-1-1 中 $N=2^p$,地址译码器的输出端共 N 个,可表示 2^p 个 p 位二进制数。每一个 p 位二进制数对应着译码器 N 个输出端子中的某个高电位端子(其余均为低电位),所以输出端共有 2^p 个,即 $N=2^p$ 。

$M \times N$ 个存贮矩阵中可存 $N(2^p)$ 个字,每个字有 m 位。存贮器存入或读出哪一个存贮单元中的字,由地址译码器输出端的相应端子的高电位决定,存贮矩阵输出的就是存贮于其中的数据(m 位)。输出缓冲器则是用来缓存某个输出字的 m 位输出。选择逻辑电路用来决定存贮器的工作状态。

计算机的字长可以指存贮器的可寻址单元的宽度,也可指数据总线的宽度。如果这两种定义对于某一机器来说不一致,一般以数据总线的宽度为准。

单片机中的 ROM 一般可分为以下三种:

(1)可擦除可编程只读存贮器(EPROM)

这种存贮器的内容可由用户抹去,再用电脉冲重新编写程度。开发单片机时,用户在编程调试阶段,常采用含有 EPROM 的单片机,但这种存贮器的单片机价格较高,它常用于产品的研究与开发阶段。

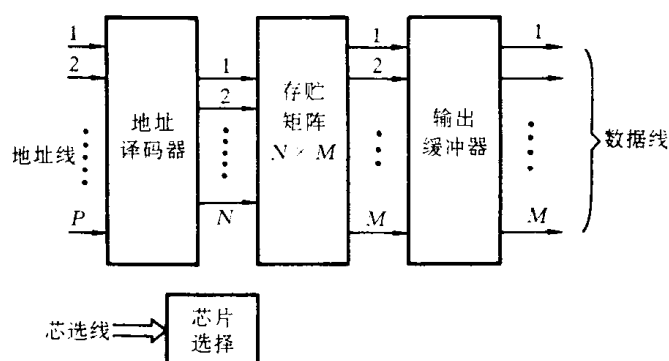


图 1-1-1 ROM 内部框图

(2) 可编程只读存储器 PROM

用户可以根据自己设计的程序,用电或光的方法在 PROM 中写入指令代码。但一经写入后,就不能再更改,用户只能一次编程写入。PROM 的价格低于 EPROM,适合用小批量试生产。

(3) 掩膜 ROM

这是由生产厂家按一定的信息模式生产的,存有固定信息的 ROM。用户可以设计好固定程序后交由厂家大批量掩膜。这种 ROM 的成本最低,适合生产厂家大批量生产。

二、读/写存储器(RAM)

读/写存储器也称随机存储器。在计算机运行过程中,它既可读出其中的内容,也可存入其中的内容。但是一断电,存储器的内容就立即消失了,所以它常用来存储计算机运行过程中的中间结果。

RAM 有静态和动态两种,静态 RAM 是用一位触发器存入一位信息,只要给它提供电源,它的信息可以一直保留。动态 RAM 的信息是以充电的形式将电荷存入 MOS 管的栅极——衬底电容上。这些电荷,由于漏电的缘故,只能保持约若干毫秒的时间。因此,必须对元件予以周期地刷新(即给电容再充电)。动态 RAM 存入一位信息所需的元件较少,故它在一定的面积上能组装较多的位数,同时它比静态 RAM 速度快,但它需要外部附加刷新电路。静态 RAM 不需刷新,便于应用。较小容量的存储器一般用静态 RAM,而容量较大时,一般用动态 RAM。

三、堆栈

堆栈存储器是用来作为数据暂时存储的一组寄存器或存储单元。在暂存数据时,将新输入的数据依次地堆放在旧的数据上,这样最先存放的数据就在堆栈存储器的底部,而最后存放的数据则在存储器顶部。当取数据时,则从堆栈存储器的顶部先取,因此第一个取出的数是最后一个存进去的数,这种工作方式称之为“后进先出”。堆栈存储器的寄存器(或存储单元)数目称之为堆栈的深度。

实现堆栈存储器的功能有两种方式:

(1) 硬件堆栈

它是由一组直接位于微处理器芯片的内部寄存器组成。当需要将累加器 A 中的数据推入堆栈时,则先将堆栈存储器中原先所存的数据依次下推一个位置。然后将新输入的数据存在堆栈顶部,见图 1-1-2 所示。

当一个数据需要从堆栈存储器中弹出时,它总是将堆栈顶部的数首先弹出,然后再将堆栈存储器内部所存的数据依次上推一个位置,见图 1-1-3 所示。

总之,不论是堆入或弹出操作都是从堆栈顶部这个固定的位置开始的。它的优点是速度快,然而硬件堆栈深度受到限制,一般用 8 个或 16 个寄存器作为堆栈存储器。另外还要设置“栈空”及“栈满”的标志,这些都对使用者带来不便。因此,使用这种方法的比较少。

(2) 软件堆栈

大部分微处理器使用这种堆栈存储技术。它是用微处理器外部的存储器 RAM 中任意个连续存储单元来作为堆栈存储器,而在微处理器内部设置一个堆栈指示器,用它来保存与指示作为堆栈存储器的 RAM 区栈顶的地址。由于这时堆栈的深度不受限制,存储器 RAM 中任何一个区域都可以由程序员来指定作为堆栈存储器。堆栈存储器的写入与读出是通过指令来实

现的,但是栈顶由堆栈指示器自动管理。每执行一次数据入栈操作,堆栈指示器便自动执行栈地址减 1。相反地,当从堆栈中取出一个数据,则堆栈指示器中的栈地址自动加 1,这样就保证了堆栈指示器所指示的是栈顶的地址。

第二节 微处理器 CPU

一、中央处理器概述

中央处理器(CPU)又称微处理器(见图 1-2-1),它是单片的指挥与执行机构,由它读入用户程序,并逐条地予以解释、执行。通常微处理器的内部硬件结构可分成二大部分:一部分为寄存器组、运算器;另一部分为控制器。寄存器组、运算器部分包括若干个可编程序的工作寄存器(工作寄存器主要由地址寄存器与数据寄存器组成)以及算术逻辑单元。寄存器组与运算器之间的所有数据和地址的传送都是通过内部数据总线、地址总线进行的。各寄存器及运算器的输入及输出均挂在总线上,由相应的门电路进行控制。控制器是使整个系统按时序协调操作的功能部件。控制器在运算器,I/O 接口和存贮器之间发送同步信号。它对指令实行读、取、译码、执行等进行操作。

控制器内常见的电路有程序计数器(PC)、指令寄存器(IR)、指令译码器(ID)、控制逻辑部件、堆栈指示器(SP)、处理机状态寄存器(PSW)等。它们均有各自 CPU 不可缺少的作用。

二、算术/逻辑运算部件

算术/逻辑运算部件与累加器、专用寄存器以及内部数据 RAM 相连,完成 8 位数据的相加、相减,加 1、减 1、与、或、异或、清零等算术逻辑运算,还能进行左移、右移等操作,算术/逻辑运算部件还可以对累加器和任一寄存器进行位操作,运算的结果经过内部的数据总线送至累加器或数据 RAM 中,同时此结果还影响状态寄存器。

三、专用寄存器组

(1)累加器

累加器也是一个多位寄存器,它不进行加法运算,而是用来临时存放运算器在运算过程中的中间结果。累加器除了能装入与输出数据外,还能将存贮于其中的数据向左或右移位。所以,它又是一种移位寄存器。

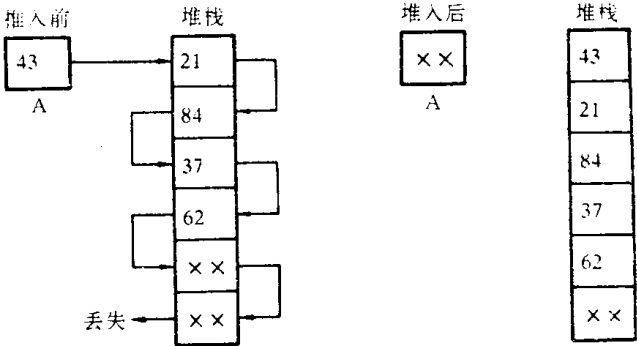


图 1-1-2 数据推入堆栈时

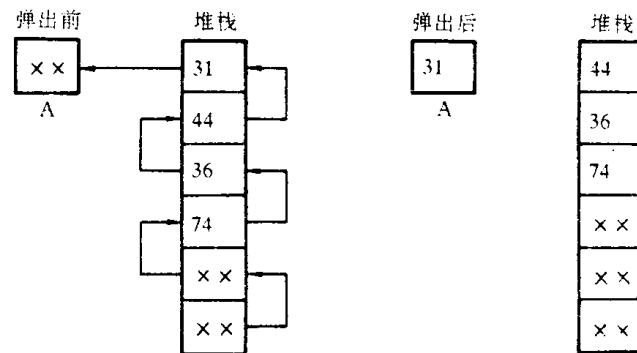


图 1-1-3 数据弹出堆栈时

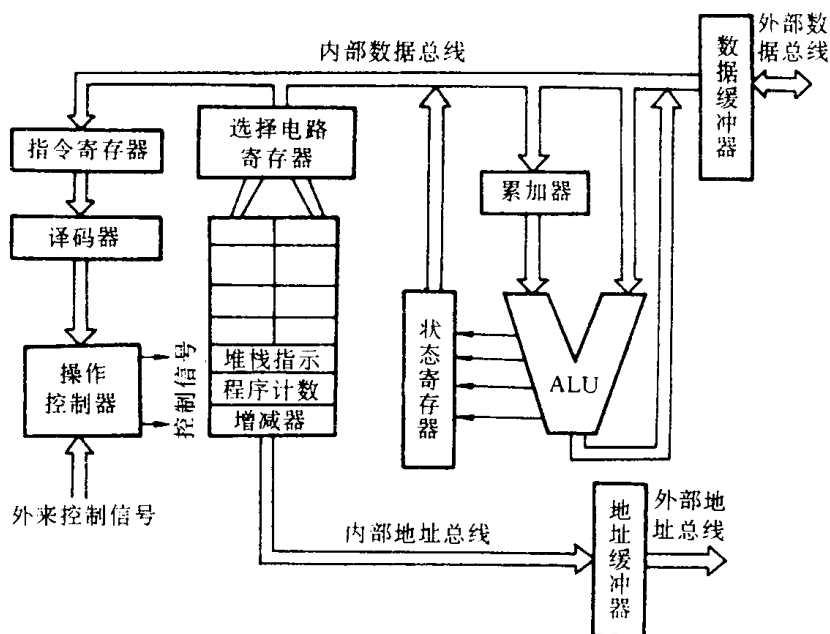


图 1-2-1 CPU 框图

(2) 状态寄存器

数据进行算术或逻辑运算时,可能会发生进位、溢出、全零、符号及奇偶性状态的变化。运算后往往需要保存这些状态的变化。为此,设置了一个状态码寄存器,有时亦称为条件码寄存器。

(3) 程序计数器

它是指令地址码的寄存器,用来指示下一个要执行指令的地址。由于程序的执行一般是按顺序的,因此在微处理器执行了取指令的动作后,程序计数器即自动加 1,为取下一条指令作好准备。

(4) 堆栈指示器

它是用来指示 RAM 中堆栈顶的地址,每次推入或弹出一数据时,它能自动减 1 或加 1,这个指示器对中断处理及子程序转移特别有用。

(5) 变址寄存器

在程序编制中常常要修改地址,在微处理器的指令系统中常采用一种称为变址寻址的工作方式。因此,要用一组触发器来保存要修改的基准地址,此即为变址寄存器。当然也可以使用变址寄存器来暂存数据。

以上均为程序员可用的工作寄存器,亦即程序员可通过指令来使用这些寄存器。

另外还有一些程序员不能使用的内部工作寄存器,如指令寄存器,它用来保存指令操作码、地址缓冲器、数据缓冲器,它将在微处理器的内部地址总线、数据总线与外部地址总线及数据总线之间建立联系,并通过外部控制信号可使总线输出三种状态。

四、CPU 工作过程概述

在微处理器内部,执行一条指令先由程序计数器将要取指令的地址经由地址缓冲器输出

到外部地址总线;与地址总线相连的存贮器某一字节被访问,同时对存贮器发出一读指令的信号;经过几百毫微秒的读取时间后,在外部数据总线上出现了指令的第一字节,即指令的操作码,它经由数据缓冲器送入指令寄存器,再经译码器译码,由操作控制器发出执行此指令的一系列控制信号;而最后完成此指令的执行。当然对于双字节或三字节指令,控制器还会发出再去存贮器取第二或第三字节的信号,以便正确地执行这一条指令。

为了对计算机的运算过程作简单描述,先对日常生活中的数据计算进行描述与分析。

假定有一张纸,记有下列若干题目:

- ① $3+5=?$;
- ② $8+7=?$;
- ③

用袖珍计算器对上述题目计算的过程如下:

- ①选定被运算题目,先选第①题。
- ②从纸上读取第一个数据,送到计算器暂存,即将第①题中的 3 敲入计算器。
- ③从纸上读运算符号;即将第一题中的加号敲入计算器。
- ④再敲入数据 5。
- ⑤要求计算器执行加法运算,即按一下“=”号。
- ⑥将计算器的运算结果写在纸上。

计算机处理问题的过程与上述相仿。计算机中的存贮器相当于上述的纸,它记录着数据、运算类型及题目序号。计算器相当于计算机内的寄存器与运算部件,它可以暂存数据,并对数据进行计算。人相当于计算机的控制器。人按已定的顺序选择题目,输入规定好的数据与算符,相当于计算机内的控制器向存贮器送出地址,再将存贮器内被地址选中的单元内的指令与数据有序地读入 CPU。人按计算器的“=”号,使计算器得出结果,相当于计算机内的控制器命令运算器按输入的数据与指令操作。人将结果写到纸上,相当于控制器指令将运算器运算得出结果送到指定的存贮器内的某个单元。

第三节 总线

将多个装置或功能部件连接起来,并传送信号的公共通道称之为总线。总线实际上是一组通讯导线。在微型计算机中,各部件的联系都是通过总线进行的,总线分系统总线或外部总线。对于微处理器内部,各逻辑功能单元也是通过总线实现互联的,也称为内部总线。

通常微处理器的外部总线可分为三种:

(1)数据总线

数据总线一般都是双向的,即数据流动方向可取正向或反向。通过它实现微处理器,存贮器和 I/O 部件三者之间的数据交换。例如,它可将微处理器输出的数据传送到相应的存贮单元或 I/O 部件,同时又可将外部的数据输入到微处理器。对于 8 位微处理器来说,它有一条 8 位宽的双向数据总线,而且是具有三状态控制的总线。容易实现微处理器的数据总线与系统数据总线的“脱开”或连接。

(2)地址总线

地址总线是微处理器输出地址用的总线。用它来确定存贮器中所存信息的地址及 I/O 部

件的地址。例如一条 16 位宽的单向总线,可允许输出 $2^{16}=65536$ 个外部地址。同样它也是具有三状态控制的总线。地址总线常与数据总线结合使用,以确定在数据总线上传输数据的来源或目的地。根据系统大小的不同,地址线可能有几条不用,也可能不够用,需要外加电路来扩展地址线数。

(3)控制总线

通过控制总线传输控制信号使微型计算机各个部件动作协调。有些控制信号是由微处理器向其他部件输出,如读、写等信号;而另外一些控制信号是其他部件输入到微处理器中,如中断请求、复位等。

微处理器内部使用的数据总线,可以采用单总线结构,即各部件的输入和输出公用一条一定宽度的数据总线。从一个寄存器将数据输入到运算器的输入端,或者从运算器输出运算结果到某一寄存器都是通过这条单一的内部数据总线来传输。当然内部数据总线上的数据信息是由定时信号控制。单总线的结构图如图 1-3-1 所示。

为了提高微处理器的运算速度,也有采用双总线或多总线结构的。例如,所有寄存器的输入用一条总线,用它来作运算器的输入总线。而运算器的输出则为另一条总线,它接到各寄存器的输入端,这就是双总线结构,如图 1-3-2 所示。

在微型计算机系统结构中,有一种称为共总线的连接方式。在这种连接方式中,微处理器将输入/输出接口都当成存贮单元来对待。也就是说,对所有使用的输入/输出接口都预先分配好了地址。因此,从某一输入接口输入数据到微处理器或微处理器输出数据到输出接口时,微处理器都把它当作对相应的某一存贮单元进行读和写操作。这样从微处理器来看,存贮器与输入/输出接口仅是地址的不同。因而,指令系统中就不必再设置专门的 I/O 指令。因为对外部设备相应的地址发出取的指令,即相当于输入指令;而对外部设备相应的地址发出存的指令,即相当于输出指令。由于这样的安排,存贮器与 I/O 接口可使用同一的数据总线、地址总线及控制总线,因此称这种系统结构是共总线连接方式,或称存贮器编址的 I/O 结构。这种结构虽然减少了一些存贮空间,但却使得系统的连接更为简单。

PIC16C5X 单片机内部采用哈佛结构,程序存贮器(ROM)和数据存贮器(RAM)是分开的,两种存贮器具有各自独立的寻址空间与独立的地址总线。这样可使指令的提取和执行周期性重叠,以便对字节、位和寄存器进行高速操作,即在一条指令被执行时,下一条指令已正从程序存贮器中读出。

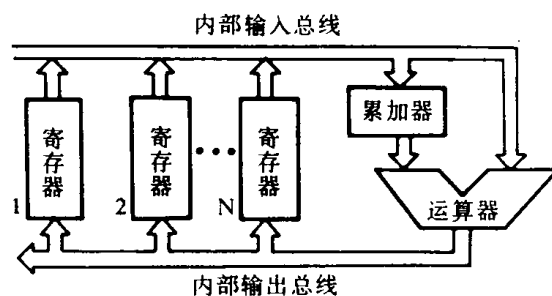


图 1-3-1 单总线结构图

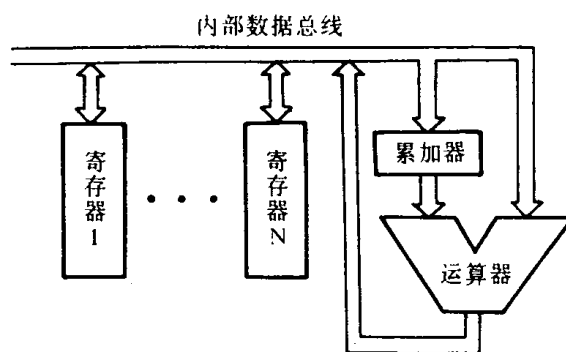


图 1-3-2 双总线结构图

第四节 输入/输出接口(I/O口)

输入/输出接口是微型计算机的重要组件。微处理器有它自己的内部特性,用户不能更改。而单片机的输入/输出设备又是各式各样的,它们也都有自己的特性,特别是单片机工作速度一般比微处理器低得多,因此要使两者能结合在一起工作,就需要有适当的接口。

通过输入/输出接口,就能把微处理器和外部设备连接起来,实现输入和输出。

输入/输出接口分单向口与双向口。单向和双向的意义详见本章第三节。

一、接口的分类方法

接口电路有以下几种分类方式:

(1)按数据传递的方式分类

微型计算机系统的接口,按数据传递方式的不同可分为并行接口和串行接口。

①并行接口。能实现将信息单元的字符同时在一组通路上传输的接口电路,叫并行接口。其通道数目等于同时传送的数据字符的位数(如图 1-4-1 所示)。8 位微机的数据,通过 8 根数据线同时从 CPU 经并行接口传送到被控制的对象。

构成一个并行接口至少要求有锁存器和总线驱动器。它一般是由输入缓冲器、输出缓冲器、地址译码器、内部状态寄存器和对寄存器或端口进行读/写的控制电路等部分组成,如图 1-4-2 所示。

输入缓冲器把来自外部设备的信息锁存起来,保持其稳定状态,直至 CPU 需要该信息为止;输出缓冲器锁存来自 CPU 的数据,并保存它们,直至外设取走为止;内部状态寄存器用来表示在缓冲器内有无输入或输出的数据;地址译码器用来选择接口。

②串行接口。能将信息流的字符在单一通道上按序传送,完成数据逐位传送功能的接口电路,叫串行接口。采用串行接口可大大减少传输线的数目。

串行接口在实际使用中还分为同步串行数据传送接口和异步串行数据传送接口。所谓同步串行数据传送接口,是指其数据的发送端和接收端共用同一时钟信号,因而数据的发送和接收是严格地被时钟信号所同步的。

不需同步时钟的串行数据传输接口称为异步串行数据传输接口。其特点是只有一根数据传送线和地线,而无时钟信号线。但接收端和发送端的时钟频率要求一致。

(2)按输入输出信息的性质分类

在一个实际问题中,微型计算机系统要处理两种性质完全不同的基本量——模拟量和数字量。按照微机系统所输入或输出信息的性质不同,相应的接口可分为数字接口和模拟接口。

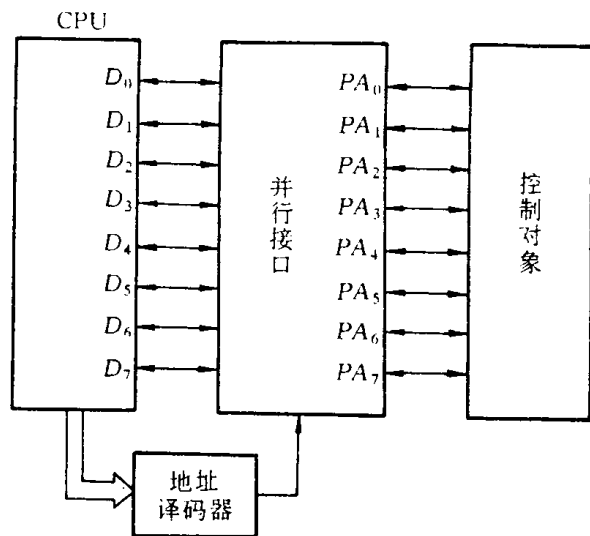


图 1-4-1 典型的并行接口连接图

①数字接口。如果通过某接口传递的信息都是二进制代码形式的数字量或控制命令,则称该接口为数字接口,如键盘接口等。

②模拟接口。完成把某一连续变化的模拟量转换成相应的数字量(简称 A/D 转换),或把某一数字量转换成相应的模拟量(简称 D/A 转换)的接口,称为模拟接口。

一个模拟量要输入计算机,首先要进行 A/D 转换(如果该信号变化速度较快,还需用采样保持电路,才能得到正确的转换值),把它转换成相应的数字量后,才能被计算机接收,然后才可能对该信号进行数据处理。若与计算机连接的是模拟量控制的外设,则必须先把计算机输出的数字量,经过 D/A 转换,把该数字量转化为相应的电压值后,才能进行控制。D/A 转换器的输出还需经过电压或电流放大(输出驱动)才可能驱动外设工作。

(3)按接口的作用分类

微型计算机的接口,按照其作用进行分类可分为内务操作接口、用户通信接口、数据采集接口和控制接口四类。

①内务操作接口。内务操作接口是指微机的处理和控制功能处于最基本水平上所必须的接口。它包括各类总线驱动器、数据锁存器及三态缓冲器等。有时也把存贮器和外存贮器设备的接口包括在内务操作接口内。

图 1-4-3 是一个典型的微机系统的内务操作接口。图中采用了总线驱动器和总线接收器来把微处理器连至系统总线。

当前,一般微处理器的每根输出线只能提供约几 mA 的驱动电流。但是,也有一些微处理器的内部驱动电路可直接驱动少量接口和存贮器芯片,而不需外加总线驱动器。但对于使用大量存贮器或接口的大型系统来说,就需要总线驱动器。

总线驱动器是一种放大器,用来增加 CPU 对数据总线、地址总线和控制总线的驱动功率。因此,从微处理器送出的信息一般都经过总线驱动器才与各类总线相连。

总线接收器则用来减轻总线的负载,并起到滤波和阻抗匹配的作用。特

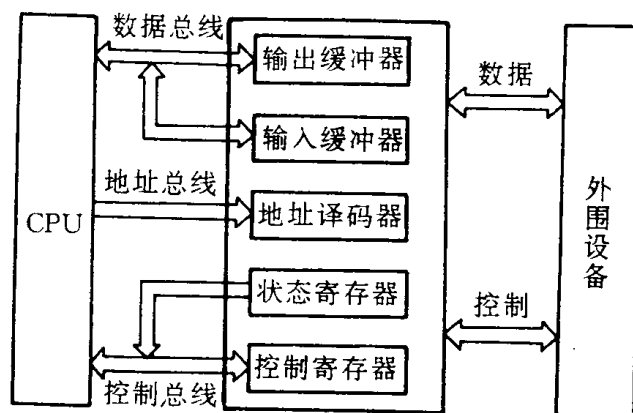


图 1-4-2 基本的并行接口框图

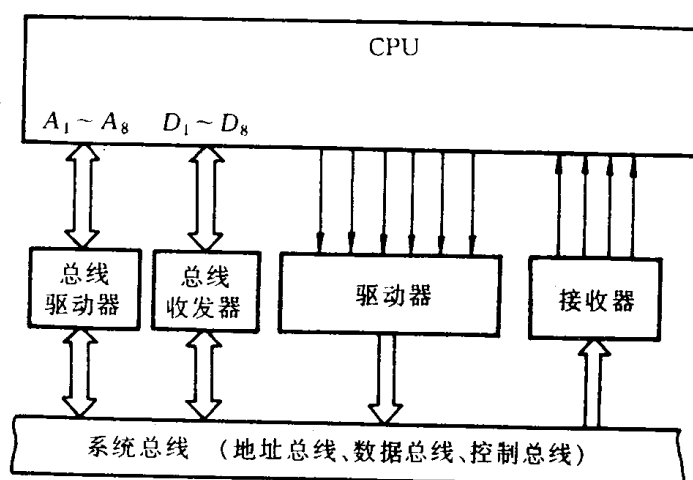


图 1-4-3 内务操作接口

别是在总线较长时,信号传送会受到噪声的干扰,波形可能畸变,信号幅度也可能减小,这就更需要通过总线接收器来增加抗干扰能力。

②用户通信接口。用户通信接口,是指微机系统把数据传送给用户指定的外设或接收用户输入信息的那些接口电路,如键盘接口、显示器接口等。

用户通信接口需要解决信息的表示方法和传送速度两大问题。信息的一般表示方法由具体的外部设备来确定,而传递的速度往往受微处理器控制。

如果外部通信设备的工作速度较低时,为了减少线路连接的复杂程度,可采用串行接口传送。当外部设备要求高速率的数据传送时,则应采用高速并行传送接口。

③数据采集接口。当单片机用于过程控制时,必须监视控制对象的变化,如现场的温度、压力和速度等参数。通常要把从现场获得有关参数信息传送给微机系统的接口电路,这种接口称为数据采集接口。实际上,也就是指那些和传感器直接相连的接口。因此,有时也把数据采集接口称作传感器接口。

因为模拟信号易受干扰影响,特别当信号电平较低时更是如此。所以在数据采集接口的设计中,应该尽早地把传感器的模拟信号转换成数字信号,然后再进行长距离的传送。

④控制接口。一个应用于实际生产过程的单片机,仅有数据采集接口显然是不够的,它往往还需要对某些执行部件进行控制,这就要用到控制接口。所谓控制接口是指把来自微机的控制数据或命令,经过适当处理后,传递给执行部件的接口电路。

控制接口不但要解决信号转换的问题,而且要解决控制信号的功率放大问题。将低电压弱电流信号放大到可以驱动控制对象的执行元件所需的电压值和电流值。

实际上,CPU 送出的控制信号功率很弱(在毫伏安数量级)。因此,在控制接口中必须加以功率放大,才能驱动执行部件。

二、传送方式的分类

微型计算机与外围设备之间传送数据的方式按传送条件可归结为二大类型:

1. 程序控制输入/输出

这时微型计算机和外围设备之间的数据交换均由微型机内的程序来控制,其过程见图 1-4-4。

程序控制输入/输出可分为以下几种:

(1)无条件传送

根据固定的和已知的定时,可将 I/O 指令插入程序中,当程序执行至该 I/O 指令时,外围设备已经为数据交换作好了准备,并在此指令时间内接收或发送数据。这是一种最简单的传送方式。

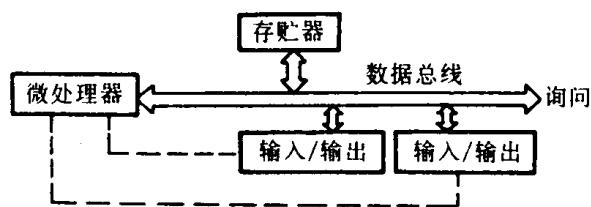


图 1-4-4 程序控制输入/输出过程

(2)条件输入/输出传送方式

这种传送方式的过程见图 1-4-5 所示。它分为以下几个步骤:

①测试外围设备状态。先执行一条指令,即读入所选择的外围设备状态。然后,基于该设备状态,转至第二步,即作条件转移。若设备处于“忙”状态,则重复第一步。

②当准备就绪时,启动设备。

③根据 CPU 发出的 I/O 指令进行输出/输入数据传送。

④传送结束后,该设备暂停。条件输入/输出传送方式的主要优点是能较好地协调外围设备与 CPU 之间的定时差别,且用于接口的硬件较少,不要求专门的转输线。主要缺点是 CPU 必须等待循环,很费时间。

(3)中断

为了节省上述等待循环的时间,达到快速反应的目的,可采用中断方式进行数据传送。当主程序在执行过程中,若外围设备在某种原因不能执行,则向 CPU 提出中断请求,经过中断响应、现场保护、中断服务、现场恢复等 9 个步骤,返回到主程序,中断过程见图 1-4-6 所示。

当外围设备需要中断服务时,它首先向 CPU 的中断线发出一中断请求信号,微处理器就在现行指令周期结束时,发出“中断响应”信号或简单地开始一个中断周期予以响应(这是第 2 步)。通常在执行第 2 步时,就把中断关闭,直至此中断处理完毕才再允许新的中断,以保证在处理一个中断过程中,不致又接收或承认另一个新的中断(否则,要处理多重中断)。

下一步是寻找和确定请求服务的设备和保存程序计数器的内容(如压入堆栈)。找到了中断设备后,就可以转移到相应的中断服务子程序中去,这是第 3 步。

通常子程序的开始部分是保存 CPU 内部寄存器内容(入栈)。接着执行所要求的数据交换,这是第 4 步。数据交换后,第 5 步就是恢复 CPU 内寄存器的内容(出栈)并要重新开启中断(使用开中断指令)。然后程序回到原来的断点。继续执行正常程序。图 1-4-7 为中断程序的流程。

2. 直接存储器存取(DMA)

这种数据交换方式是由外围设备或 DMA 控制器来控制的。它直接在外围设备与存储器之间进行数据传送,而不需要 CPU 干预,其特点是数据交换速度快,交换方式见图 1-4-8 所示。

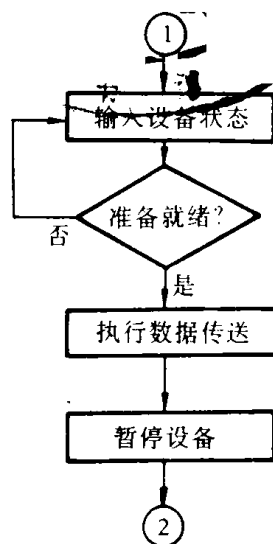


图 1-4-5 条件输入传送过程

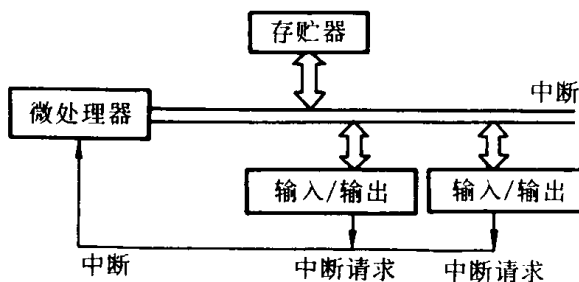


图 1-4-6 中断过程

第五节 定时系统

单片机内微处理器执行指令的时间顺序(时序)都是事先规定好的。微处理器为了连续地执行指令序列,需要有时钟脉冲。时钟脉冲可以内部供给,也可以外部供给。为了获得可靠稳定的频率,常采用晶体振荡器来产生脉冲,再经分频,整形后得到所需的时钟脉冲。机器一通电,则时钟信号连续不断地产生。

如图 1-5-1 所示,环形计数器的 CLK 端子输入矩形时钟脉冲 K ,按环形计数器工作原理,其输出端 Q_0 、 Q_1 …… Q_5 的输出波形分别为 T_0 、 T_1 …… T_5 。当第一个时钟脉冲上升沿到来之时, T_0