

# 单片机两片系统的应用研究

程汉湘 夏勇军 姚齐国 武汉化工学院自动化系(430073)

### Abstract

Many periphery CMOS chips are needed if we design the hardware of single-chip system in the traditional way, which always make the area of the design board become too large, and make wiring complex, such design will reduce the system reliability either. In order to resolve this problem, this paper discusses the dual-chip system, which is made up of PSD302 and 80C196KC. The design clue of this system will also be clearly expressed. As an example, the collection of motor speed data and the application of this dual-chip system in the area of control are explained in this paper.

**Keywords:** chip microprocessors, PSD302, 80C196KC, data collection

### 摘要

传统的单片机硬件电路设计中,使用的外围芯片较多,这往往使得印制电路板的面积过大、布线复杂,使可靠性降低。为克服这一缺陷,本文讨论了用 PSD302 芯片与 16 位微处理器 80C196KC 组成的单片机两片系统,并详细论述了该系统的实现思路,并以调速系统中电机中转速的采集为例说明其在控制系统中的应用。

**关键词:** 单片机, PSD302, 80C196KC, 数据采集

普通的单片微处理器系统常常需要许多的外部存储芯片,如 EPROM、RAM 及译码器和 PLD 等,这样一来,就会使单片机系统硬件设计过于臃肿,布线也变得复杂。美国 WSI 公司(Wafer Scale Integration Inc)生产的现场可编程外围逻辑芯片 PSDXXX [1],[2] (Programmable System Device)片内集成了 EPROM、RAM 和逻辑译码电路,可以对两个端口进行功能编程,使系统的设计、修改和扩展都变得十分灵活方便,而且具有片内程序加密功能。另外,它还可以很好地与大多数通用的 8 位和 16 位微处理器接口。本文就 Intel 公司的 16 位单片机 80C196KC 和 PSD302 所组成的两片系统进行了分析,并说明了该两片系统在调速系统中数据采集的应用。

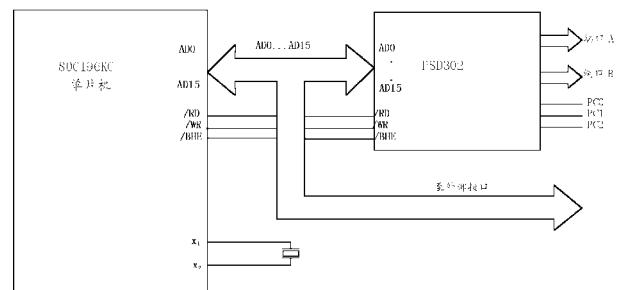
## 1 系统设计

### 1.1 系统的组成及原理说明

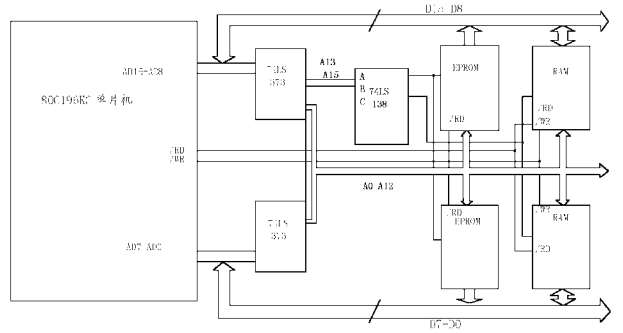
单片机两片系统指的是只由一个微处理器和可编程外围逻辑芯片组成的一个系统,它不需要其它外加的 EPROM、RAM 和逻辑译码单元等器件。它与常规的单片系统是不同的,一般的单片机系统存储器芯片通常是 ROM 和 RAM 分离的,而且对于 16 位机,这两种存储芯片各需要两片,这样会使外部芯片过多,从而使电路板的功耗增加,给硬件的调试也带来诸多不便。PSD 各系列芯片的应用很好地解决了

这一问题。

16 位单片机系统组成原理框图如图 1 所示,(a)和(b)分别为两片系统和传统单片机系统的原理图。



(a)



(b)

图 1 单片机两片系统与常规单片机系统的原理框图比较  
从上面给出的两个原理框图中可以看到,在两片

系统中,PSD302 可以直接与 80C196KC 和外设接口,很方便地实现与微处理器间的数据交换和对外设的片选逻辑译码。而在传统的 16 位单片机系统中,外围存储芯片(EPROM 和 RAM 各两片)必不可少,而且,微处理器的数据、地址总线常常是多路复用的,这就要求用锁存器(74LS373)将地址信号锁存,当然还得有译码单元(74LS138)实现对外设的片选逻辑译码,这样,使整个系统变的非常庞杂。由此可知,单片机两片系统与传统的单片机系统在硬件设计上的优势。

## 1.2 系统的微处理器 80C196KC

在 16 位单片机中,Intel 公司的 8XC196KX 是国内比较流行的一种,本文采用 80C196KC 作为系统的微处理器。与 8 位单片机相比,它们主要实用于复杂的、适时性要求较高的自动控制、数据采集系统、信号处理系统以及高级智能仪器中。近年来,随着国内 16 位单片机的应用正不断普及,它们正成为越来越多的科技工作者进行系统开发和应用研究的对象。80C196KC 集成了丰富的系统资源<sup>[9]</sup>,符合本系统的设计要求,而且,80C196KC 的价格也并非难于接受,所以选择它来进行本系统的开发。

## 1.3 可编程外围芯片 PSD302 的编程

PSD302 是美国 WSI 公司生产的现场可编程外围芯片 PSDXXX 中的一种,它内部集成了 19 个可单独配置的 I/O 接口、两个可编程逻辑门阵列、512K 位的 EPROM、16K 位的静态 RAM,此外它还可以方便地实现片内程序的加密。它的两个可编程地址译码端口可方便地实现片内存储器 and 片外接口芯片的片选。其数据总线可方便地配置为 8 位或 16 位的工作方式,能很好地与多种不同配置的微处理器芯片接口。正是由于 PSD 内部集成了如此多的外围器件,才使得单片机两片系统的实现成为可能。

PSD 芯片的编程,采用 WSI 公司提供的专用软件 PSD-Lite 和硬件描述语言 PSDlabel-HDL。程序主要是限定片内存储器的地址范围和 PSD 各扩展端口逻辑,并实现单片机的汇编程序在 PSD 内部地址上的映像。

在程序中,首先要对总线接口信号、端口和其它内部节点进行声明,即为所用到的 PSD 的管脚取一个便于识别的别名,如下所示:

```
rd      pin 22; " Bus Interface signals declaration
inst    pin 40; "Port C pin declaration
x8279_cs pin 11; "Port B pin declaration
```

```
.....
rs0,csiop,es7,es6, ... es1,es0 node; " other internal
node and PAD A Output declaration
```

为便于编程,将一些相关的信号组合在一起定义为一个新的变量:

```
address = [a15,a14,a13,a12,a11,X,X,X, X,X,X,X, X,X,X,X];
```

上面一句表示将各地址信号定义在一起,其中“X”说明可为任意值。从该定义可以看出,在后续程序中我们只关心 address 变量的前五位。

接下来是对各输出端口进行编程。对于端口 A,一般用作 PSD 内 EPROM 的片选。PSD302 内部 256KB 的 EPROM 被分成 8 个存储块,每块 32KB,如图 2 所示。8 个存储块由 PAD 的 8 根输出线 ES0、ES1...ES7 进行选择。

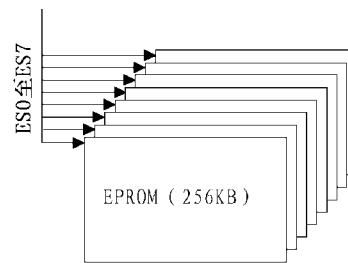


图 2 内部的 EPROM 分块

对各片选信号可编程如下:

```
.....
es2 = inst& ((address >= ~h0000) & (address <=
~h1FFF));
.....
```

```
es7 = inst&((address >= ~hE000) & (address <=
~hFFFF));
```

另外,PSD 内部 16K 位的 SRAM 由 PAD 输出的 rs0 来选择:

```
rs0 =!inst&((address >= ~hE000) & (address <=
~hE7FF));
```

端口 B 一般用作外设的片选信号,如外设芯片 8279 可编程如下:

```
x8279_cs= !inst&((address >= ~hF800) &
(address <= ~hFFFF))&!ale;
```

当然,编程中所涉及的地址要依具体硬件电路的设计而定,以上只是说明了几个重要环节编程的过程及格式。当硬件电路修改时,程序中的部分地址应作相应的变化。

整个 PSD 芯片的编程设计过程可由图 3 的编程框图表述。首先,在 PSD 的编程环境中先后生成扩展名为 abl、tt2、glc 等文件,然后与 80C196KC 汇编编程

序编译产生的十六进制文件(.hex)在 PSD-Lite 编程环境中进行代码映像,生成目标文件(.obj)。该目标文件便可作为最终代码写入 PSD 芯片。

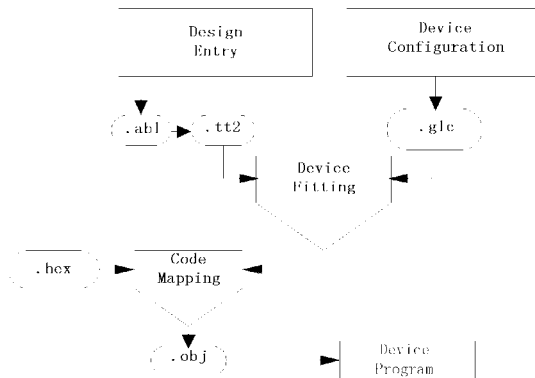


图3 PSD编程过程框图

## 2 单片机两片系统在调速系统中的应用

在的调速系统中,转速和电流的采集是控制系统的重要环节。本文重点并不是研究调速系统理论,因此下面仅以电机转速的采集为例来分析两片系统在调速系统的数据采集中的应用。

在 80C196KC 中有四个 HSI 高速输入部件 (High Speed Input), 它们以定时器 T1 为时间基准记录外部事件发生的时间及状态。并且 HSI 脉冲信号的输入不需由 CPU 干预, 而由 HSI 部件自动完成。

程序是在对 HSI 初始化后再进行数据读取:

```

LOOP: LDB  HSI_MODE,#00H
        ;设定每 8 个正跳变为一个事件
    CLRB  INT_PENDING
        ;清中断悬挂寄存器
    LDB  IOC0,#01H
        ;允许 HSI.0 输入
    .....
READ_1: JBC  IOS1,7,READ_1
        ;查看 HSI 保持寄存器是否有数据
    LD   AX,HSI_TIME
        ;读取时间发生的最初时间
    LDB  IOC1,#04H
    LDB  INT_MASK,#01H
        ;允许 T1 溢出中断
    EI
    .....
READ_2: JBC  IOS1,7,READ_2
        ;查看 HSI 保持寄存器是否有数据
    LD   BX,HSI_TIME
        ;读取时间发生的最初时间
    .....
    
```

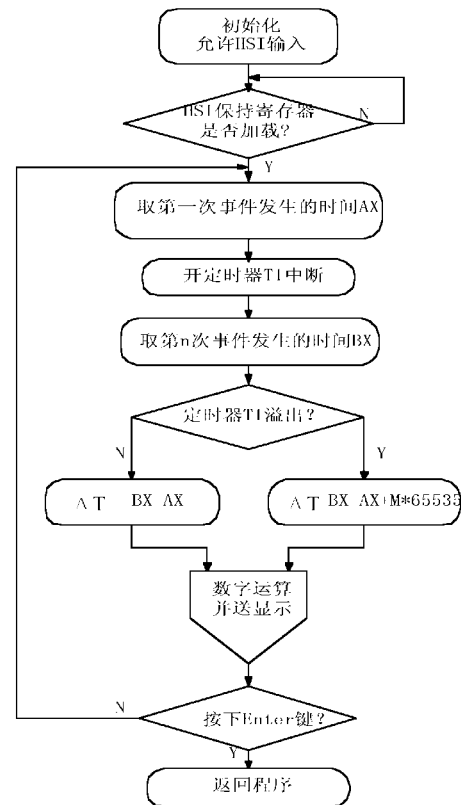


图4 电机转速采集程序设计框图

我们采用光电编码盘将脉冲信号输入 HSI, 并假设电机每转可由光电编码盘产生 2500 个脉冲。如上面的程序将 8 个正跳变定为 HSI 的一个事件, 记下 3 次事件内的 T1 的变化值, 并分别装入 AX 和 BX 中。在电机运行过程中一定要注意定时器 T1 溢出的情况, 如有溢出中断产生, 记下其溢出次数 m。则在光电编码盘产生  $3 \times 8 = 24$  个脉冲期间, T1 的变化值为:  $\Delta T = (BX - AX)$ ; (T1 未溢出时) 或  $\Delta T = (BX - AX) + m * 65535$ ; (T1 溢出时) 由此, 光电编码盘每产生一个脉冲, T1 的变化值为:  $\Delta T / 24$ 。

在此系统中, 我们采用 16M 晶振, 这样, 晶振的振荡周期  $T_{osc} = 1 / (16 \times 1000000) S$ 。与 8098 不同, 80C196KC 的晶振时钟被 2 分频为一个状态周期 T, 即  $T = 2 * T_{osc} = 2 / (16M) = 125ns$ 。

因为 16 位计数器 T1 的记数脉冲来自芯片内部的时钟发生电路, 而且每 8T 其值加 1, 这样, T1 每变化一次, 所需要的时间是:  $8 \times 125ns = 1\mu s$ 。

由上面的分析可知, 光电编码盘每产生一个脉冲所花的时间为:

$$T_{bm} = 1 \times (\Delta T / 24) \mu s$$

$$\text{电机的转速为: } 60 \times 10^6 / (2500 * T_{bm}) \text{ r/min}$$

完成上述计算后,将转速送显示,再回到转速采集程序的开始,重复上面的工作。

该程序用到了定时器 T1 和键盘中断,关于中断优先级的控制编程可参阅相关文献<sup>[4]</sup>。

当然,上面只是说明了实现转速采集及计算的原理,适当地选取每次转速采集过程完成后的时间延时,便可以取得很好的适时显示效果。另外,通过改变所记录的时间发生次数,能获得更高的显示精度。

### 3 实验结果分析

上面实验采集的数据可以通过 80C196KC 的串口,用 RS-232C 标准电缆送到 PC 机上显示出来。图 5 为电机起动和停机时的转速变化曲线,该实验曲线从整体上反映了两个动态过程,与理论值基本相符。

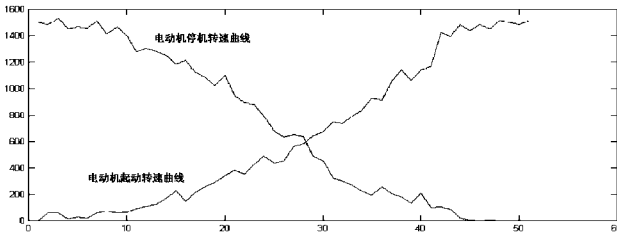


图 5 电动机起动和停机的转速变化曲线

### 4 结束语

在单片机的两片系统中,就微处理器而言,可以

选用 Intel 8098 系列中的专门用于电机高速控制的 80C196MC,甚至 Intel 80286 也可以很好地与 PSD 系列芯片接口。就 PSD 芯片而言,该芯片已发展到 PSD9XX 系列,它们具有更大的存储容量和更加灵活、丰富的接口功能,其中的 PSD4XX 和 PSD5XX 系列都可以与 16 位机直接接口,并且,随着产品的不断更新,其价格差距也不断缩小,大有取而代之。

通过以上分析可以看出,单片机两片系统结构简单明了,使用灵活方便,不失为一很好的单片机设计思路和发展方向,在今后单片机开发中具有很好的应用前景。

### 参考文献

- 1 颜荣江,朱元清,侯文军.PSD3 系列可编程单片机外围接口芯片原理编程及应用.北京:人民邮电出版社,1995
- 2 孙涵芳.单片机现场可编程外围芯片 PSD 的原理及应用.北京:北京航空航天大学出版社,1998
- 3 孙涵芳.Intel 16 位单片机.北京:北京航空航天大学出版社,1998
- 4 汪建,章育群.MCS-96 系列单片机的中断优先级的控制方法.微处理机,1999(1)

[收稿日期:2001.10.14]

(上接第 42 页)

### 067 基地变电站一次系统

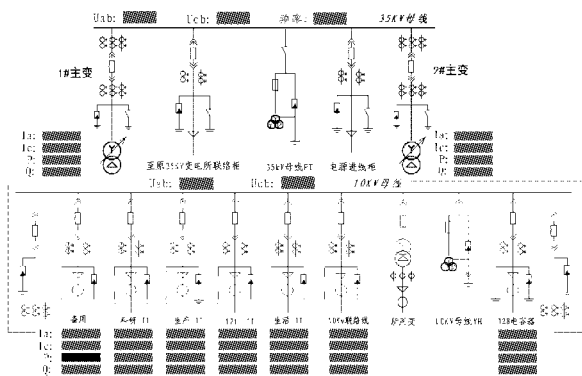


图 1 电力监控界面图

### 2.2 实时图像的绘制

图像的实时性是通过线程来实现的,每次线程运行时,对数据库重新查询的到新数据,并根据查得的新数据对图像进行重画,这样就能将时刻变化的一个电力一次系统反映给用户。线程可以用 Thread 类实现,也可以通过 Runnable 接口来实现。不论哪种实现方法你都需在线程的 run()方法中利用数据库对象

实现对数据库的查询,用 repaint()实现图像的重画。需要注意的是,应在线程的在 Java 小程序的 Paint()函数中编写与实时数据相关的图像,这样字符的显示代码,这样才可以利用 repaint()重画。另外,线程的休眠时间可以由 Sleep(int time)函数决定(单位为毫秒),这决定了图像的刷新速率。关于这部分代码的编写,其中 paint()函数的编写应由具体的要求来定,调用 API 函数很容易实现。线程的定义也很容易,线程 run()函数的编写主要是对数据库的访问,这部分代码在前面已经给出。

### 3 结束语

本文具体讨论了对工业监控环境中对实时数据在 Web 页面上的自动定时刷新问题的一种实现方法,希望对该问题的解决起到抛砖引玉的作用。

### 参考文献

- 1 Richard 著.VJ++宝典.电子工业出版社,1999
- 2 闪四清著.Microsoft SQL Server 2000 实用教程.人民邮电出版社,2000

[收稿日期:2001.10.9]