

概述

SSI208P 接口转换模块主要应用于同步串行接口 (SSI) 光电编码器高速数据采集系统的板级开发。SSI208P 模块将同步串行接口数据转换成并行接口数据, 内部集成了 SSI 同步时钟发生器、脉冲计数器、数据串并转换、接口控制逻辑、输出控制以及收发驱动器 (TTL-RS422 电平转换) 等功能单元, 用户无须了解 SSI 数据格式, 该模块自动将 SSI 数据转换成 8 位并行数据, 简化了 SSI 编码器与 DSP、单片机、PC104 等控制器的接口。

应用方向

- 伺服控制
- 工业控制
-

产品特性

- 单 3.3V 供电, 工作电流小于 100mA;
- 通信速率可配置, 最高达 2MHz;
- 28 脚双列直插封装; 尺寸: 25.4*25.4*6(mm);
- 8 位数据总线, 可接 8~32 位编码器;
- 高数据更新率, 如 16 位数据更新率大于 100KHz;
- 内部时钟, 固定时序;
- 内置 422 差分驱动。

产品信息



目录

1. 产品介绍	1
2. SSI208P 外形	1
2.1 SSI208P 实物图.....	1
2.2 SSI208P 外形尺寸.....	2
3. 管脚定义及特性参数	2
3.1 SSI208P 引脚定义.....	2
3.2 SSI208P 特性参数.....	3
4. SSI208P 控制时序	3
5. 基于 SSI208P 模块的应用实例	4
5.1 TMS320F2812 接口实例	4
5.2 C8051F310 接口实例.....	6
6. 销售与技术服务	7

1. 产品介绍

SSI208P是针对同步串行接口（SSI）编码器开发的高速数据采集模块。内部集成了SSI同步时钟发生器、脉冲计数器、数据串并转换、接口控制逻辑、输出控制以及收发驱动器（TTL-RS422电平转换）等功能单元，其功能框图如图1所示。

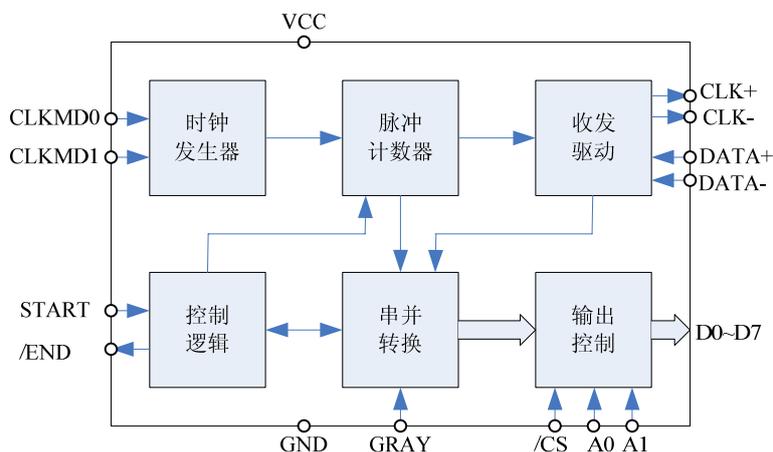


图1 SSI208P功能框图

该模块具有内部时钟，能自动将SSI数据转换成并行数据，对SSI接口数据的读取操作就类似于对A/D、D/A或存储器读取数据的操作一样方便。SSI208P模块通信速率可配置为250KHz、500KHz、1MHz、2MHz，当通信速率配置为2MHz时，对于16位精度的编码器，系统数据更新率不低于100KHz。此外，该模块对采集的数据长度（编码器精度）可以进行配置，最高可以采集32位数据，分4次输出，该模块可以满足高精度高速伺服控制系统的需求。

2. SSI208P 外形

2.1 SSI208P 实物图



图2 SSI208P实物图

2.2 SSI208P 外形尺寸

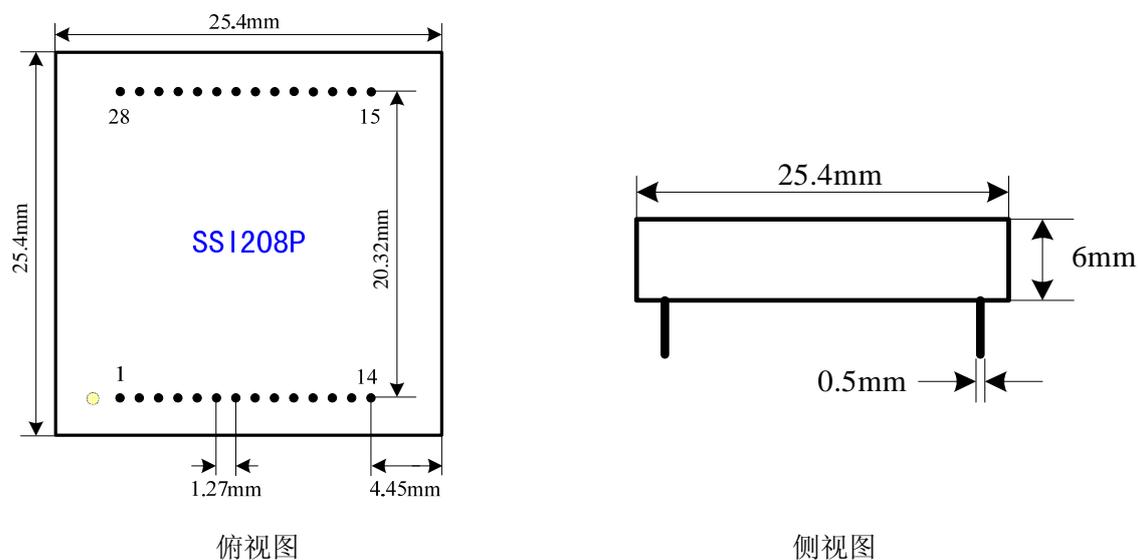


图3 SSI208P外形尺寸

3. 管脚定义及特性参数

3.1 SSI208P 引脚定义

编号	引脚名称	方向	引脚含义	备注
1-4	D0-D3	输出	数据输出	接处理器数据线
5	NC		空管脚	使用时悬空
6-9	D4-D7	输出	数据输出	接处理器数据线
10-12	NC		空管脚	使用时悬空
13,14	A0,A1	输入	数据高低字节选择	A1A0: 00-最低八位, 01-次低八位, 10-次高八位, 11-最高八位
15	DATA+	输入	编码器数据线+	
16	DATA-	输入	编码器数据线-	
17	CLK-	输出	编码器同步时钟-	
18	CLK+	输出	编码器同步时钟+	
19	NC		空管脚	使用时悬空
20	GRAY	输入	格雷码转换控制	0-原码输出

				1-将格雷码转换为 BCD 码
21	/CS	输入	编码器数据输出选择	0: 输出选择
22	/END	输出	转换结束	1: 正在转换, 0: 转换结束
23	START	输入	启动转换	上升沿启动转换
24	NC		空管脚	使用时悬空
25	GND		电源地	
26	CLKMD1	输入	同步时钟设置	00: 250KHz 01: 500KHz 10: 1MHz 11: 2MHz
27	CLKMD0	输入	同步时钟设置	
28	VCC		电源	3.3V 供电

3.2 SSI208P 特性参数

电源 VCC	最小值 3.0V, 典型值: 3.3V, 最大值: 3.6V	
功耗	静态电流<70mA, 最大电流<100mA	
温度范围	-40~85℃	
I/O 电气特性	输出高电压 (V _{OH})	最小值 VCC-0.4V
	输出低电压 (V _{OL})	最大值 0.8V
	输入高电平 (V _{IH})	最小值 2V, 能承受 5V 输入
	输入低电平 (V _{IL})	最大值 0.8V

4. SSI208P 控制时序

SSI208P控制时序如图4所示。START上升沿启动一次SSI编码器数据收发过程,START电平升高后,125ns内SSI208P模块开始向编码器发送一帧同步时钟脉冲信号,同时转换结束管脚/END变高。发送脉冲期间管脚/END保持高电平状态,转换结束/END管脚电平变低后,即可从D0~D7并行读取编码器数据,每次读取八位,由A1A0控制输出数据高低位,0低八位、1表示读取高八位,对于16位编码器需读取两次。编码器并行数据读取结束后将START管脚置低,准备启动下一次转换。

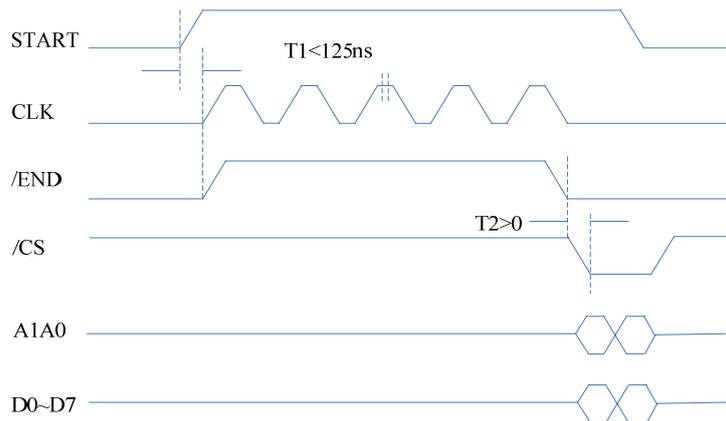


图 4 SSI208P 控制时序

5. 基于 SSI208P 模块的应用实例

5.1 TMS320F2812 接口实例

采用SSI208P模块可以大大简化单片机、DSP、PC104等控制器扩展SSI编码器接口的软硬件设计，下面给出一种基于DSP处理器TMS320F2812的典型应用。 DSP处理器TMS320F2812与SSI208P模块的硬件连接原理图如图5所示。

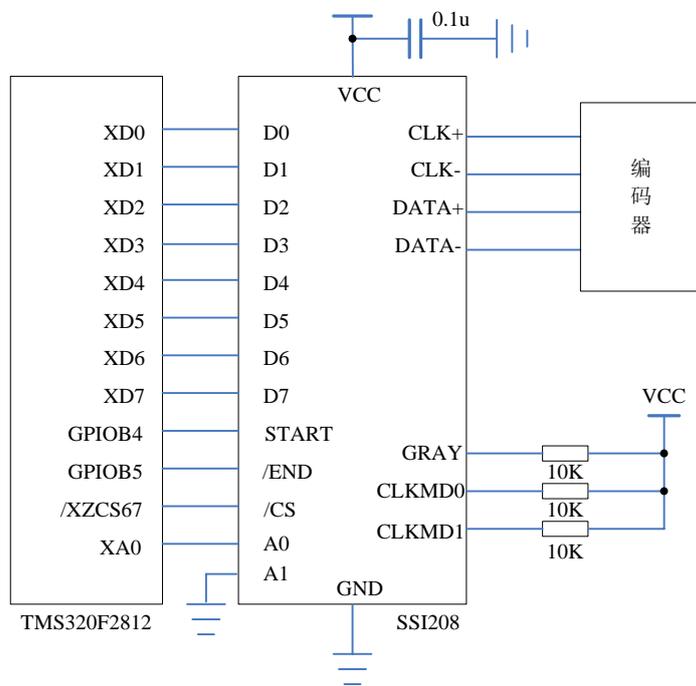


图 5 TMS320F2812 与 SSI208P 连接硬件设计原理图

上例中编码器为单圈16位绝对式角度编码器，编码器输出数据形式为格雷码；SSI208P模块的八位数据总线与TMS320F2812的低八位数据线相连；因使用的编码器为16位，所以

仅需一位地址线即可区分编码器数据高八位和低八位；使用外部地址片选管脚/XZCS67作为 SSI208P 的外部片选信号；使用通用 IO 口 GPIOB4 控制 SSI208P 模块启动；使用通用 IO 口 GPIOB5 进行 SSI208P 模块转换结束状态查询；CLKMD0、CLKMD1 接上拉电阻拉高，将 SSI208P 模块同步时钟频率配置为 2MHz；将 GRAY 管脚拉高，SSI208P 模块将编码器输出的格雷码数据转换成 BCD 码。其它控制器的硬件电路依次类推。

由 SSI208P 模块控制时序图可以看出 SSI208P 模块的启动转换控制和数据读取操作比较简单，软件流程如图 6 所示。

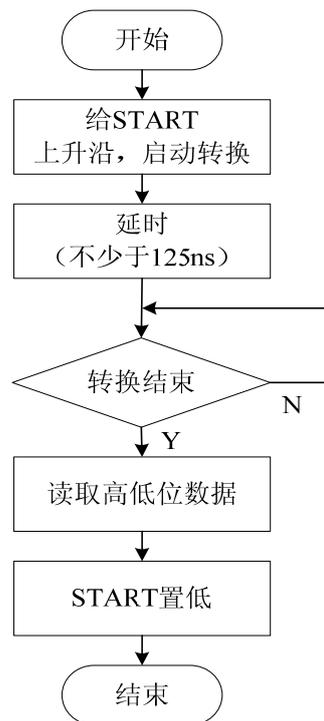


图6 SSI208P软件流程图

对应图 5 中硬件设计，DSP 处理器 TMS320F2812 相应的软件代码如下所示：

```

#include "DSP28_Device.h"
#define SSIL      *(int *)0x00100000      //SSI 模块低八位地址
#define SSIH      *(int *)0x00100001      // SSI 模块高八位地址
int  SSI_VAL;      //编码器数据
main()
{
    .....
    EALLOW;
    //配置 GPIO4、GPIO5 引脚为 IO 引脚
    GpioMuxRegs.GPBMUX.bit.GPIOB4=0;
    GpioMuxRegs.GPBMUX.bit.GPIOB5=0;
  
```

```

GpioMuxRegs.GPBDIR.bit.GPIOB4=1;           //GPIOB4 配置为输出
GpioMuxRegs.GPBDIR.bit.GPIOB5=0;           //GPIOB5 配置为输入
EDIS;
//启动数据发送及转换
GpioDataRegs.GPBDAT.bit.GPIOB4=0;
GpioDataRegs.GPBDAT.bit.GPIOB4=1;
for(i=0;i<1;i++);           //延时约 200ns
while((GpioDataRegs.GPBDAT.bit.GPIOB5)!=0); //查询转换是否结束
SSI_VAL=(SSIL&0xFF)|((SSIH<<8)&0xFF00); //读取 SSI 编码器高低八位数据并组合

GpioDataRegs.GPBDAT.bit.GPIOB4=0;
.....
}

```

5.2 C8051F310 接口实例

例中编码器为单圈16位绝对式角度编码器，编码器输出数据形式为BCD码；SSI208P模块的八位数据总线与单片机C8051F310的P1端口相连，单片机IO口P2.0控制模块启动转换，P2.1查询转换是否结束，P2.2作为模块片选信号，P2.3区分编码器数据高八位和低八位；CLKMD0、CLKMD1接地，将SSI208P模块同步时钟频率配置为250KHz；GRAY管脚接地，编码器输出的BCD码数据不做转换。原理图如图7所示。

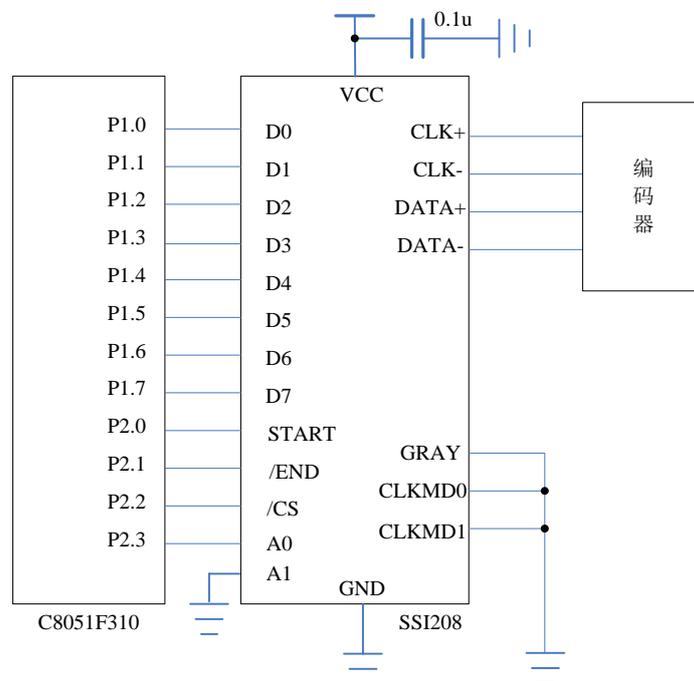


图 7 C8051F310 与 SSI208P 连接硬件设计原理图

对应图 7 中硬件设计，单片机 C8051F310 相应的软件代码如下所示：
sbit START =P2^0; //P2.0 与 START 管脚对应

```

sbit END_N   =P2^1;           //P2.1 与 END 管脚对应
sbit CS_N    =P2^2;           //P2.2 与 CS 管脚对应
sbit A0      =P2^3;           //p2.3 与 A0 管脚对应
.....
unsigned int SSIL;
unsigned int SSIH;
unsigned int SSI_VAL;
.....
void main (void)
{
    .....
    CONVST=0;
    for(i=0;i<n;i++);
    CONVST=1;                 //启动转换
    for(i=0;i<n;i++);         //延时
    while(END_N!=0);         //查询转换是否结束
    CS_N=0;                   //SSI 模块片选
    A0=1;                     //读取编码器数据高字节
    for(i=0;i<n;i++);         //延时* (1)
    SSI_H =P1;                //从 P1 端口读取编码器高八位数据
    A0 =0;                    //读取编码器数据低字节
    for(i=0;i<n;i++);         //延时* (2)
    SSI_H =P1;                //从 P1 端口读取编码器低八位数据
    SSI_VAL=(SSIL&0xFF)|((SSIH<<8)&0xFF00); //合并成编码器数据
    CONVST=0;
    CS_N=1;
    .....
}

```

注：在程序加入延时（1）和延时（2）是因为 IO 口响应有滞后，需一定的稳定时间。

6. 销售与技术服务

联系方式：13141393549（靳先生） htjin@asee.buaa.edu.cn
13683263781（陈先生） shelfer@163.com
13552091081（赵先生） 208servo@163.com