基于 CPLD 的 DSP2812 SPI,SCI 外部中断扩展机制的研究

□ 龙 也[1] 李 洁[2] 钟文斌[3]

([1]武汉大学动力与机械学院自动化系 湖北·武汉 430072;
[2]武汉大学国际软件学院软件工程系 湖北·武汉 430072;
[3]武汉大学动力与机械学院机械制造及其自动化系 湖北·武汉 430072)

摘 要：设计了基于基于 CPLD 的 DSP2812 SPI,SCI 外部中断扩展方案。利用 CPLD 的可编程逻辑特性编写 SCI 及 SPI 接口。并通过 CPLD 与 2812 的外部中断引脚相连传回 SCI,SPI 数据接收完成信号,触发 2812 的外部中断并在相应中断函数内通过读取指定端口识别码来确定当前外部中断申请设备。达到了同时接收发送多路 SCI、SPI 数据并保证无故障、无丢失工作的目的。

关键词：外部中断 SCI 扩展 SPI 扩展 CPLD 资源扩展 2812

中图分类号：TP39 **文献标识码**：A **文章编号**：1007-3973(2010)02-105-01

TMS320F2812 强大数字信号处理能力使其在控制系统开发领域备受关注和重视。但是,TMS320F2812 的 SCI(串行通信接口)数量仅为两路,SPI(高速同步串行口)数量仅为一路,这大大限制了其在数据通信量较大场合的应用。基于 TMS320F2812 与 CPLD 的 SPI, SCI 扩展方案可用于增加可用从模式 SPI, SCI 数据接收端口数量。

1 SCI 中断扩展实现原理

2.1 外部中断触发原理

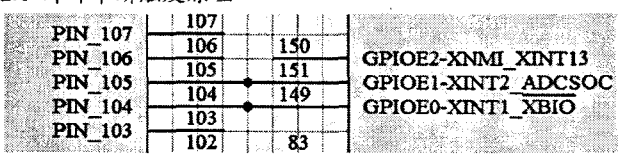

图 1

如图所示,CPLD 的 PIN_104,PIN_105 引脚分别与 TMS320F2812 的 GPIOE0-XINT1_XBIO 引脚及 GPIOE1-XINT2_ADCSOC 引脚硬件相连。

当 CPLD 接收完一个 SCI 或 SPI 数据并需传回时,CPLD 分别拉低 PIN_104, PIN_105 引脚以触发 TMS320F2812 的 XINT1, XINT2 外部中断。TMS320F2812 根据中断类型代码调用相应的中断处理函数。

2.2 TMS320F2812 设备识别及数据接收

在 XINT1 与 XINT2 的中断处理函数中,TMS320

F2812 读取 GPIOA0-PWM1,GPIOA1-PWM2,GP

IOA2-PWM3 三个 IO 口的状态以获取该次中断的中断源设备代码(如:000 为设备 0,001 为设备 1),再结合当前中断类型代码便可唯一确定当前外部中断申请的设备。(如图 2 所示)

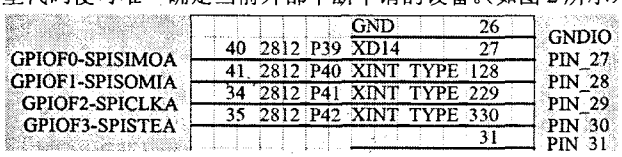

图 2

随后根据当前中断申请设备代码,TMS320

F2812 向 16 位地址总线写入对应的数据访问地址,并拉低 CS(片选信号),RD(读信号),拉高 R/W(读写使能)三个引脚。CPLD 在检测到相应的写信号后读入地址总线数据,并将对应存储单元内的数据放到 16 位的数据总线上,由 TMS320F2812 取出接收到的 SCI(SPI)数据。

2.3 CPLD16 级深度中断扩展实现代码

```
always @(posedge SCI_XINT1 or posedge SCI_XINT2 or
posedge SCI_INTFINISH_FLAG or posedge SCI_XINT_SEND)
begin if(SCI_XINT_SEND)
//当检测到已开始一次外部中断请求时将 SCI_XI
NT 置为忙碌状态
begin SCI_XINT_BUSY <= 1; end
if(SCI_INTFINISH_FLAG)
//当一个 SCI 外部中断请求已经被响应时,清除该中断请
求信息,堆栈值减一
begin SCI_INTSTACK_B3 = SCI_INTSTACK_B3>>1;
SCI_INTSTACK_B2 = SCI_INTSTACK_B2>>1;
SCI_INTSTACK_B1 = SCI_INTSTACK_B1>>1;
SCI_INTSTACK_NUM = CI_INTSTACK_NUM-1; SCI_
XINT_BUSY <= 0;
//允许新的 SCI 中断请求
end
if(SCI_XINT1)
//SCI1 产生了外部中断请求信号
beginif(SCI_INTSTACK_NUM<=15)
begin SCI_INTSTACK_B3[SCI_INTSTACK_NUM] = 0;
      SCI_INTSTACK_B2[SCI_INTSTACK_NUM] = 0;
      SCI_INTSTACK_B1[SCI_INTSTACK_NUM] = 1;
SCI_INTSTACK_NUM = SCI_INTSTACK_NUM
+1;end end end
always @(posedge CLK)
begin
SCI_XINT = 0;SCI_XINT_SEND = 0;
//如果 SCI_XINT 模块空闲,并且仍有数据待发送则启动
发送过程,并提交发送行为报告
if(~SCI_XINT_BUSY)
begin if(SCI_INTSTACK_NUM>0)
begin
SCI_XINT_TYPE[2] = SCI_INTSTACK_B3[0];
SCI_XINT_TYPE[1] = SCI_INTSTACK_B2[0];
SCI_XINT_TYPE[0] = SCI_INTSTACK_B1[0];
SCI_XINT_SEND = 1; SCI_XINT = 1; end end end
```

3 结论

本文通过利用 CPLD 编写 16 级深度的 SCI,SPI 中断数据、类型存储队列,很好的解决了在利用 CPLD 实现多端口 SCI 数据无互扰并行接收时,CPLD 如何无丢失,无错误,及时的向主控芯片 TMS320F2812 发出相应 SCI 数据接收中断信号并送出已接收通信数据的问题。在实际应用中取得了良好的效果。