

MDT10F1823 SPI 通讯程序范例

文件编码： AN001808Y

简介

SPI 串行外围设备接口是 Motorola 首先在 MC68HCXX 系列处理器上定义的，主要应用在 FLASH、实时时钟、数字信号处理和数字信号解码之间。SPI 是一种高速的、全双工同步通讯总线，在芯片的引脚上只需占用四根线，节约了芯片的管脚，同时为 PCB 的布局上节省了空间，提供方便。正是出于这种简单易用的特性，现在越来越多的芯片集成了这种通讯协议，如 MDT10F1823。

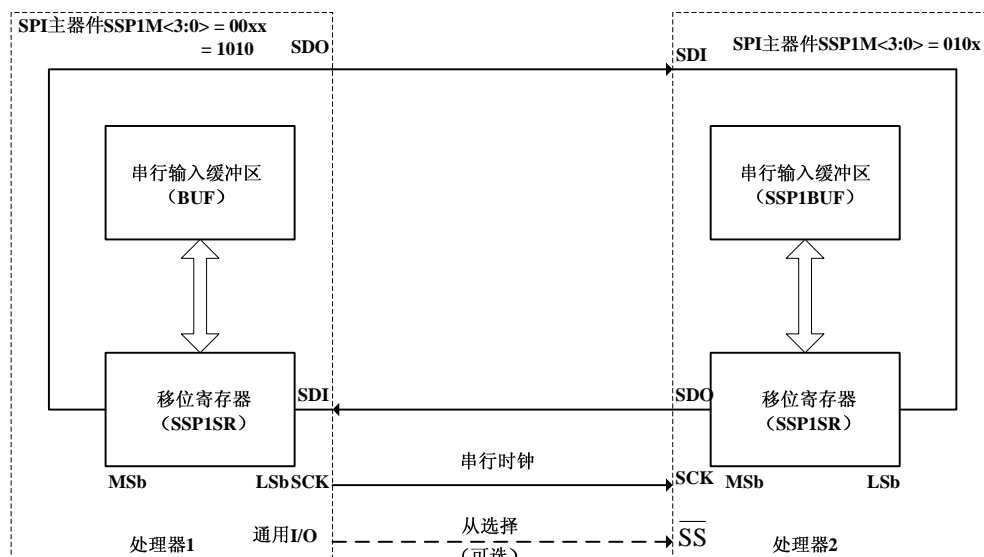
工作原理

SPI 通讯原理很简单，它以主从方式工作，这种模式通常有一个主设备和一个或多个从设备，需要至少4根线，事实上3根也可以（单向传输时）。它们是SDI(数据输入)、SDO（数据输出）、SCK（时钟）、CS（片选）。

- 1)、SDO：主设备数据输出，从设备数据入；
- 2)、SDI：主设备数据输入，从设备数据输出；
- 3)、SCK：时钟信号，主设备输出，从设备输入；
- 4)、CS：从设备片选信号，输入，主设备控制。

SPI 串行数据在 SCK 时钟脉冲的作用下一位一位的传输，数据在时钟上升沿或下降沿时改变，在紧接着的下降沿或上升沿被读取，完成一次数据传输，输入也使用同样的原理。这样，在至少8次时钟信号的改变（上沿和下沿为一次）就可以完成8位数据的传输。

SPI 允许数据一位一位的传输，甚至允许暂停，因为 SCK 时钟线由主设备控制，当没有时钟跳变时，从设备部采集或传送数据。也就是说，主设备通过对 SCK 时钟线的控制可以完成对通讯的控制。因为 SPI 的数据输入和输出线都是独立的，所以允许同时完成数据的输入和输出。





```
void SPI_init(void)
{
    CPIOC    |= 0x02;
    SSP1STAT = 0x40;    //时钟边沿选择
    SSP1CON1 = 0x32;    //主机模式
    SSP1CON3 = 0x00;
}

unsigned char SPI_comm(unsigned char data_spi)
{
    unsigned char data_rd;
    SSP1BUF = data_spi;    // 将需要传送的数据写入缓冲区，启动传输
    while((SSP1STAT & (1<<BF)) != (1<<BF)); //等待数据传输完成
    data_rd = SSP1BUF;    // 读取接收到的数据
    return data_rd;
}
```