BR0101固件开发——模数/数模部分

（第十六周）

本周主要进行了AD9286的调研。

# AD9286的调研

## SPI接口控制

AD9286的SPI接口允许用户对转换器的功能进行配置，或者对内部提供的一些寄存器进行操作。

### 通过SPI进行配置

AD9286采用的是三线SPI串口，具体定义如下：



图 1 AD9286 SPI接口信号定义

AD9286的SPI接口时序如下图所示[1] ：

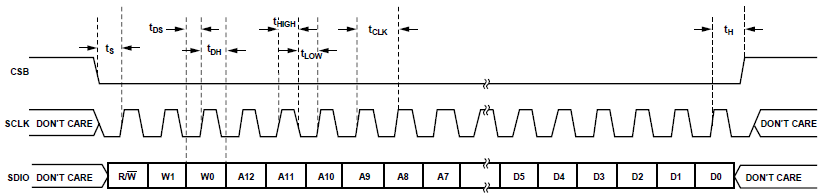


图 2 AD9286 SPI接口时序

#### 传输开始

CSB下降沿和SCLK的上升沿标志着传输的开始，启动时间。

#### 流传输模式

如果把CSB一直拉低，并且指令中W0和W1代表的值为3，就可以开启流传输模式（Streaming）[2] 。这种状态下器件的SPI接口一直有效，会不断接收新的数据，从起始地址开始逐一把接收到的数据写入后续的地址。

#### SPI无效模式

如果把CSB一直拉高，SPI就处于高阻状态，并且启用所有SPI引脚上的第二功能。

#### 指令阶段

每次SPI的传输，首先要通过上位机向AD9286传送16比特的指令，指令中包括1比特的读写控制位、2比特的字长控制位和13比特的地址位。

#### 读写控制位

读写控制位是指令的第一位，决定了后续的数据是读还是写。如果读写控制位为1，则可以从AD9286读出数据；如果读写控制位为0，则可以向AD9286写入数据。由于AD9286的SDIO是复用的数据输入/输出口，所以需要根据这一位调整管脚的输入/输出方向。

#### 字长控制位

字长控制位W1和W0决定了传输数据的字节数，具体定义如下所示：

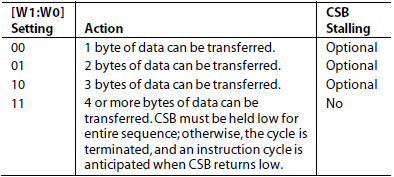


图 3 字长控制位的定义

[W1:W0] = 00, 01, 10的情况下可以在字节边界上拉CSB，使传输暂停；但是[W1:W0] = 11的情况下不能使用这个功能。

#### 地址位

剩下的13比特是地址位，表示数据传送开始的地址。

#### 数据阶段

指令阶段后面是数据阶段，数据的数量由前面的字长控制位确定，所有数据都是由8比特的字构成的。

#### 比特顺序

数据传输的顺序可以是MSB优先也可以是LSB优先，上电之后，默认的配置是MSB优先。在LSB优先的模式下，指令的比特顺序也会相应地颠倒（即16比特中读写控制位在最后）。

总的来说，AD9286和AD9715的SPI控制方式差不多，只是数据的格式有些区别，可以在AD9715固件的基础上进行修改。

需要注意的是，在ADC以最高性能工作时，SPI接口最好不要处于活动状态，因为SCLK、SDIO、CSB等信号常常和ADC的时钟信号是异步的，这些信号产生的噪声可能使模数转换的性能降低。

### 不通过SPI进行配置

如果不通过SPI接口，也可以对AD9286进行一些简单的配置。这种模式下，需要将CSB信号拉高，可以利用SDIO/PWDN信号的第二功能，控制ADC掉电。

## 寄存器结构

AD9286的寄存器地址为8位，寄存器可以分为三类：芯片配置寄存器（地址0x00到0x02）、器件索引和传输寄存器（地址0x05和0xFF）和程序寄存器（地址0x08到0x38）。

### 传输寄存器映射

地址0x08到0x38的寄存器在收到传输指令之前不会影响器件的运行。传输指令是在寄存器0xFF中写入0x01，将传输控制位置为1，这样能够使得这些寄存器的值在内部同时被更新，之后传输控制位会被自动清零。可以理解为这些寄存器需要经过一个锁存器来控制内部的功能。

### 指定通道寄存器

有些功能可以两个通道分开配置，芯片内部自动将控制这些功能的寄存器分为两个。这些寄存器也被称为局部寄存器。可以通过寄存器0x05的通道1控制位和通道2 控制位分别访问控制两个通道的局部寄存器。

如果通道1控制位和通道2 控制位都置为1，则写入操作会同时影响两个通道；读出操作会返回通道1的数据。

全局寄存器会影响整个器件或者它所控制的功能不允许对两个通道分别配置。寄存器0x05的通道控制位不会影响全局寄存器的行为。

# 参考资料

1. ADI, [AD9286 Datasheet](../references/AD9286/AD9286.pdf)
2. ADI, [Interfacing to High Speed ADCs via SPI](../references/AD9286/Interfacing%20to%20High%20Speed%20ADCs%20via%20SPI.pdf)