AN-1248
APPLICATION NOTE

One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106, U.S.A. • Tel: 781.329.4700 • Fax: 781.461.3113 • www.analog.com

SPI Interface

by Miguel Usach

INTRODUCTION

The SPI bus interface is widely used for synchronous data transmission because this interface allows relatively high transmission rates with versatile configurations.

Although the SPI has become a de facto standard, it is not a de jure standard; in other words, it is not officially specified. This can sometimes be considered an advantage because the designer can get the most from a part; however, it complicates the interconnection between different parts.

The SPI bus consists of four unidirectional wires. The names for these wires vary between parts, even within the same range of products.

- Interface enable: $\overline{\text{CS}}$, $\overline{\text{SYNC}}$, $\overline{\text{ENABLE}}$, and so on.
- Data in: SDI, MISO (for master), MOSI (for slave), and so on.
- Data out: SDO, MISO(for slave), MOSI (for master), and so on.
- Clock: SCLK, CLK, SCK, and so on...

It is important to start by defining a convention name to avoid confusion regarding the direction of each pin as shown in Figure 1.

Sometimes only three wires are needed. For example, in a DAC it may not be necessary to read back the data, or, in the case of an ADC, to send data. In those cases, the connection can be defined as a 3-wire interface.

MASTER-SLAVE COMPATIBILITY

The first step is to guarantee the compatibility of the master-slave connection. The SPI interface is not an official specification, so it is important to ensure that data from the master to the slave and/or vice versa fits within both specifications.

The SPI is not a completely synchronous interface because the data is synchronized with the clock, but $\overline{\text{CS}}$ may or may be not synchronous.

In a completely synchronous interface, the edges are divided into a sampling and a driving edge. On the drive edge, the data can be updated in the bus. On the sampling edge, the data in the SDI/DATA IN pin is read in (sampled).

From a practical point, the data in the bus can be updated anytime except in the sampling edge.

The SPI interface defines four transmission modes. The master should be able to support all four modes, but this needs to be confirmed beforehand because sometimes the master is not compatible with a particular mode. This can be overcome by using inverters, as described in the SPI Mode Interconnection section.

For the most part, the slave cannot be configured and can only operate in one mode. However, sometimes it can operate in up to two different modes.

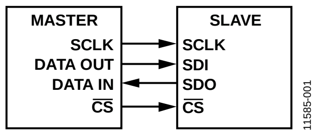

Figure 1. SPI Basic Interconnection

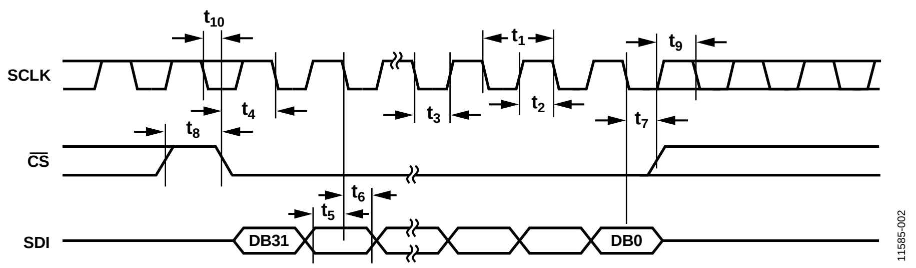

Figure 2. SPI Timing Diagram Example

TABLE OF CONTENTS

REVISION HISTORY

7/13—Revision 0: Initial Version

WHICH TRANSMISSION MODE IS USED BY THE SLAVE?

The timing diagram is a figure with multiple lines and names as shown in Figure 2.

The mode depends on the SCLK level, sometimes called polarity (CPOL), when the transmission is initiated ($\overline{\text{CS}}$ is pulled low) and the sampling edge, called phase (CPHA), as shown in Figure 3. Note that the phase is relative to the polarity and is not an absolute value. The SPI modes are captured in Table 1.

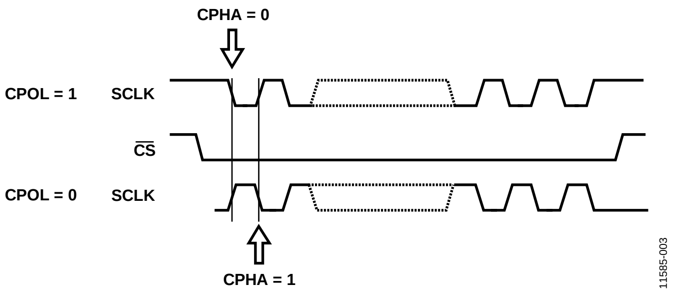

Figure 3. SPI Timing

Table 1. SPI Modes

Mode	Polarity (CPOL)	Phase (CPHA)
0	↓⎍	↑
1	↓⎍	↓
2	↑⊔	↓
3	↑⊔	↑

Identifying the transmission mode is relatively easy. There is a line that connects the $\overline{\text{CS}}$ falling edge with SCLK as shown in Figure 4.

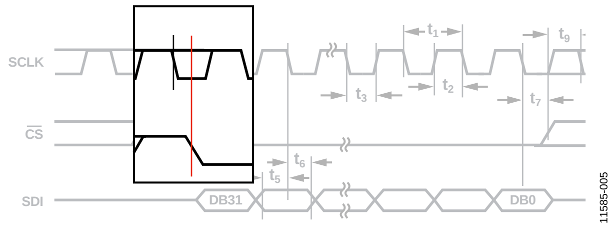

Figure 4. Polarity Mode

In this particular case, SCLK can be high or low; there is no restriction.

The SDI diagram should have a bit that is enclosed by two timings, setup and hold. The two timings refer to the time that the data should be present in the bus before and after the sample edge, so both timings use the sampling edge for reference as shown in Figure 5.

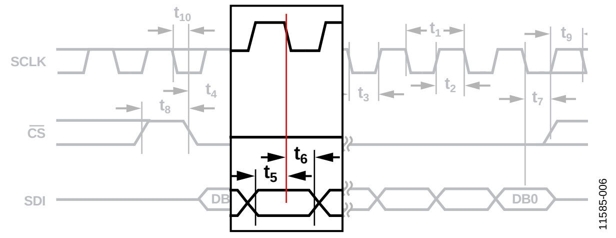

Figure 5. Phase Mode

In this case, the sample edge is falling.

Correlating the findings with Table 1, the slave part is compatible with Mode 1 and Mode 2.

WHEN IS THE BUS DATA UPDATED?

The SDO is used to transfer data from slave to master as well as transferring data out from master to slave. Although the data can be updated anytime, typically two strategies are employed.

One strategy is to update the SDO/DATA OUT pin in the driving edge as shown in Figure 7.

The other strategy is to update the SDO/DATA OUT pin several nanoseconds after the sampling edge as shown in Figure 8.

There are technical reasons behind both strategies, but it is important to understand the tradeoffs.

Masters use the first strategy because the SDO drivers are designed to support fast transitions.

Slaves implement an internal SDO driver weaker than the master, thus the strategy implemented is dependent on the data transfer rate.

If the SDO signal is updated in the driving edge, the pin has only one-half (or even less) of a clock period to update the signal because a signal should be stable several nanoseconds before the sampling edge.

To guarantee a correct readback, the SCLK should be reduced to guarantee that the signal is stable before the sampling edge.

For this example, assume a maximum transition time of 36 ns.

SDO Data Valid from SCLK Rising Edge	t_9	36 ns

This mean that the maximum cycle time is 36 ns + master setup time (assume 10 ns) = 46 ns, so the maximum SCLK frequency for reading back is 10 MHz.

If the pin is updated several nanoseconds after the sampling edge, the slave has almost the full SCLK period to guarantee a stable value of the signal in the bus so that the readback can be done without reducing the SCLK frequency.

The main trade-off is for slow masters because the data only is stable in the pin several nanoseconds after the sampling edge and the master hold time can be violated. This problem occurs because the hold time is high, that is, >15 ns. If this is the case, the recommendation is to use a logic gate to delay the new data in the DATA IN pin as long as it needed as shown in Figure 6.

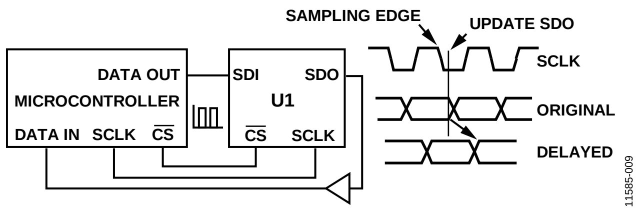

Figure 6. Enable Time

Several gate technologies and typical propagation delays for a NOR gate are shown in Table 2.

Table 2. Gate Technologies and Propagation Delays

Technology	Propagation Delay
AHC	4.4 ns
HC	9 ns
HCT	11 ns

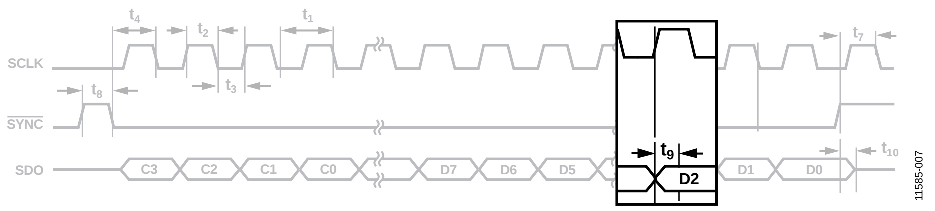

Figure 7. SDO Driving Edge Update

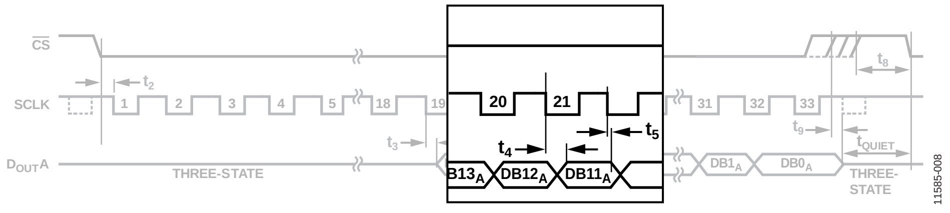

Figure 8. Delayed SDO Update

ARE THERE ADDITIONAL CONSIDERATIONS?

Enable Time

Enable time defines how fast the SPI interface is enabled and ready to receive or transmit data. This is typically referred to as the SCLK sampling edge as shown in Figure 9.

Disable Time

Disable time defines how fast the SPI is disabled to ignore any new generated sampling edge transitions as shown in Figure 9.

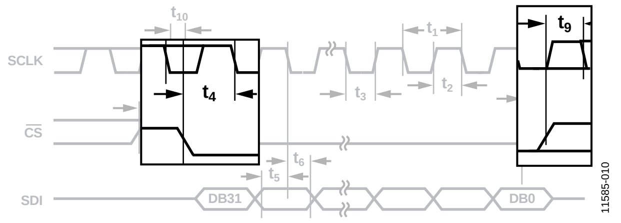

Figure 9. Enable and Disable Time

$\overline{CS}$ as Start Conversion Signal

Some ADCs, in order to reduce the pin count, and fit in small packages, or just to reduce the routing complexity, offer multiple functionality in a single pin.

When the $\overline{CS}$ is used to generate the internal start conversion signal, there are two different case scenarios.

First Scenario

The SCLK signal is used as an internal clock, and continuous SCLK is needed. In this case, the SCLK is limited between a maximum and minimum value as shown in Table 3.

Table 3. Example of SCLK Frequency Limitation

Parameter	Min	Max	Unit	Description
f_{SCLK}	0.01	20	MHz	SCLK frequency

There is timing, restriction, similar to the enable time described below, in which the master cannot generate a sampling edge, or the conversion will be corrupted, as shown in Figure 10.

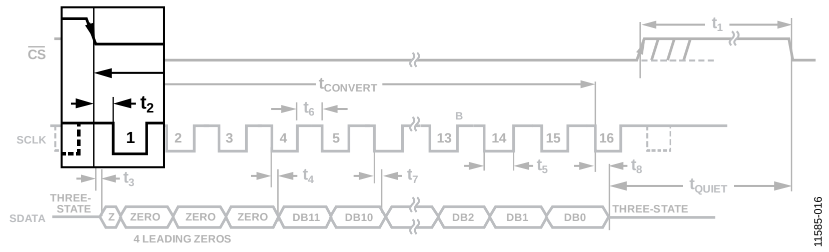

Figure 10.Continuous SCLK During Conversion

Second Scenario

The part includes an internal conversion clock. In this case, the recommendation is to not generate SCLK pulses to reduce the digital feedthrough impact in the LSB bits conversion as shown in Figure 11.

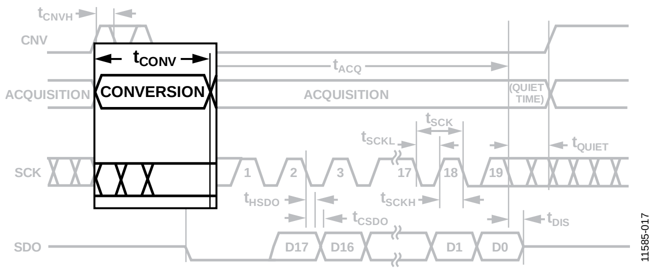

Figure 11. Quiet SCLK During Conversion

If the SPI interface is implemented by the hardware, rather than an FPGA, it is not possible to have accurate control of the SCLK and $\overline{CS}$ pin. If this is the case, the recommendation is to use a GPIO as $\overline{CS}$, to accurately control the relation between $\overline{CS}$ and SCLK.

SDO as Conversion Ready Pin

In some ADCs, the SDO provides double functionality. This is typically noted as SDO/$\overline{RDY}$. The SDO pin is disabled with $\overline{CS}$ and remains at high impedance until the conversion is completed at which point the pin is pulled low, indicating the end of the conversion.

SPI MODE INTERCONNECTION

Sometimes because the controller cannot be configured in a particular SPI mode used by the slave or because there is a need to operate all the devices with the same SPI mode, that is, daisy-chain mode, the mode needs to be modified externally.

Consider these two cases:

- The mode is complementary where MODE 1 = $\overline{\text{MODE 2}}$ or MODE 0 = $\overline{\text{MODE 3}}$
  Using a inverter gate in the SCLK line the problem is fixed
- The modes are not complementary.
  The solution becomes a bit more elaborate, and involves the use of inverters and flip-flops, thus the recommendation is to avoid this because timing issues may arise.

TOPOLOGIES

The SPI interface permits different topologies allowing the master to control one or several slaves.

Standalone Topology

In this configuration, there is only one slave and one master, as shown in Figure 12.

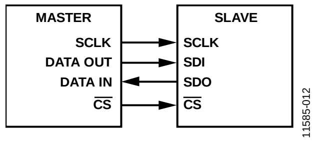

Figure 12. Standalone Configuration

Daisy-Chain Topology

In this configuration, there is one master and multiple slaves connected in series as shown in Figure 13.

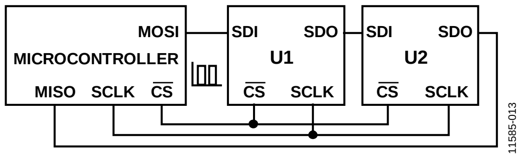

Figure 13. Daisy-Chain Configuration

The main benefit of this configuration is the reduced number of connections required.

Operating in this mode, the clock period may need to be increased because the propagation delay of the line between subsequent devices. In addition, the number of clocks should be increased because the required clocks are the sum of U1 and U2.

Typical transmission in a daisy-chain configuration is shown in Figure 14. The first data-word is assigned to the last slave connected and the last data-word is assigned to the closest slave.

There are parts that can be configured in daisy-chain mode but, by default, the part power-up in stand-alone mode, that is, the SDO pin does not clock out data.

In this case, the recommendation is to place the part in the first place in the chain and enable daisy-chain mode by writing directly to the part. Because the SDO is in high impedance before enabling the mode, it is recommended to connect a pull-up (or pull-low) resistor in the SDO pin to control the data that is transferred to the second device in the chain.

Similar problems occur when the SDO pin is used for multiple functionality, SDO/$\overline{RDY}$. The recommendation is to place a pull-up resistor to avoid electrical issues and continue using the $\overline{RDY}$ functionality.

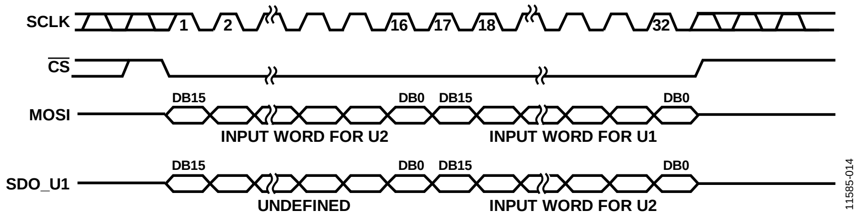

Figure 14. Daisy-Chain Timing Diagram

Parallel Configuration

In this configuration, there is one master with multiple slaves connected in parallel as shown Figure 15.

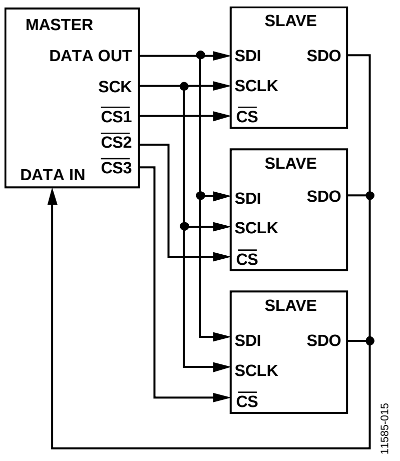

Figure 15. Parallel Configuration

In this configuration, SCLK and SDI are shared within all the parts. Due to the parasitic net (or track) capacitance, it is recommended to increase the clock period slightly.

As a precaution, in this configuration, the SDO may be not disabled synchronously with $\overline{\text{SYNC}}$ in some parts, for example, if the part is configured in daisy-chain mode.

In this case, to avoid electrical issues, the recommendation is to not connect the SDO pin to the bus. Alternatively, if it is possible to disable the SDO pin, place a serial resistance with the SDO pin to minimize electrical problems in the first transmission and disable the SDO pin at the beginning.

NOTES