I2C Slave to SPI Master Bridge

Reference Design

FPGA-RD-02111-1.2

December 2019

Disclaimers

Contents

Figures

Tables

1. Introduction

I^{2}C and SPI are the two widely-used bus protocols in today's embedded systems. The I^{2}C bus has a minimum pin count requirement and therefore a smaller footprint on the board. The SPI bus provides a synchronized serial link with performance in MHz range. As embedded systems are required to support an increasing number of protocols and interfaces, bridge designs targeting popular protocols provide solutions to reduce development time and cost. This reference design implements an I^{2}C slave to SPI master bridge. It serves as an interface between the standard I^{2}C bus of a microcontroller and a SPI bus. This allows the microcontroller to communicate directly with the SPI bus through its I^{2}C bus.

2. Features

The I2C Slave to SPI Master Bridge design includes the features listed below:

- II2C bus slave interface
- Configurable 7-bit I2C slave addressing mode
- 128-byte data buffer
- SPI master interface supporting SPI clocking modes 0, 1, 2, 3
- Configurable SPI serial clock (SCLK) frequency
- Up to five SPI slave select outputs
- Active low interrupt output

3. Functional Description

The functional block diagram of the I^{2}C Slave to SPI Master Bridge design is shown in Figure 3.1. This design operates as an I^{2}C slave and an SPI master. It receives commands from the I^{2}C master and writes or reads data to and from the SPI slave, depending on the command received.

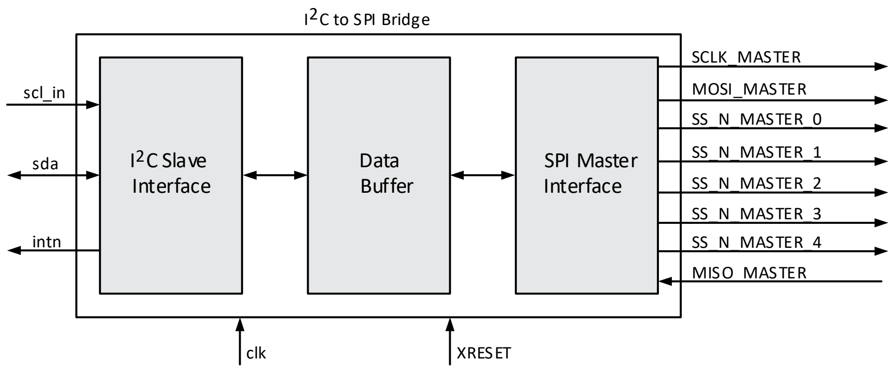


Figure 3.1. I^{2}C Slave to SPI Master Bridge Block Diagram

Table 3.1. I²C Slave to SPI Master Bridge Design Signal Descriptions

Signal Name	Signal	Active State	Description
clk	Input	N/A	System clock signal
XRESET	Input	High	Asynchronous system reset signal
intn	Output	Low	Interrupt signal. When active, this signal indicates that the SPI master has finished the read or write operation.
I²C Slave Interface			
scl_in	Input	N/A	I²C bus clock
sda	Inout	N/A	I²C bus data
SPI Master Interface			
SCLK_MASTER	Output	N/A	SPI clock
MOSI_MASTER	Output	N/A	SPI data bus – master out, slave in
MISO_MASTER	Input	N/A	SPI data bus – master in, slave out
SS_N_MASTER[4:0]	Output	Low	SPI slave select output

4. Design Module Description

The design is implemented in three sections within a single module; the I2C slave interface, the data buffer, and the SPI master interface.

4.1. I²C Slave Interface

The I²C Slave to SPI Master Bridge design has a 7-bit I²C slave address (Figure 4.1.). This address is defined by the parameter I2C_SLAVE_ADDRESS (Table 4.3.). The first seven bits of the first byte sent by the I²C master after a start condition are the I²C slave address. The eighth bit determines the direction of the message. A ‘0’ in the least significant position of the first byte indicates a write operation from the I²C master to the I²C slave. A ‘1’ in this position indicates a read operation on the I²C bus.

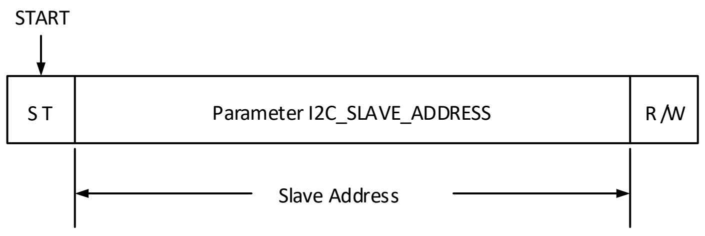


Figure 4.1. I²C Slave Address

4.2. I²C Data Bus Transaction

The command byte is the first byte to follow the 7-bit slave address during an I²C master write transmission (see Figure 4.2. and Table 4.1.). The command byte from the I²C master determines the operation on the I²C bus. Depending on the command, 1 to 128 data bytes will follow.

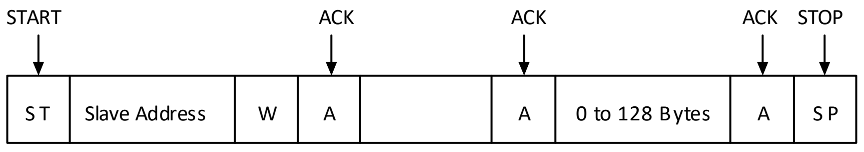


Figure 4.2. Data Format During an I²C Master Write Transmission

Table 4.1. Command Functions

Command ID	Function
0x01	Configures the SPI master interface
0x02	I^{2}C master writes data to the I^{2}C slave to SPI master bridge
0x03	I^{2}C master clears interrupt signal

Configure SPI Master Interface – Command ID 0x01

Before the I^{2}C master communicates with the SPI slave, the I^{2}C master must configure the SPI master interface. The I^{2}C master can change the SPI master interface operation mode, data direction and SPI slave selection by sending a 'Configure SPI master interface' command.

The command ID to configure the SPI master interface is 0x01. If the configure SPI master interface command (0x01) is sent, the following data byte will define the SPI communication interface (see Figure 4.3. and Table 4.2.).

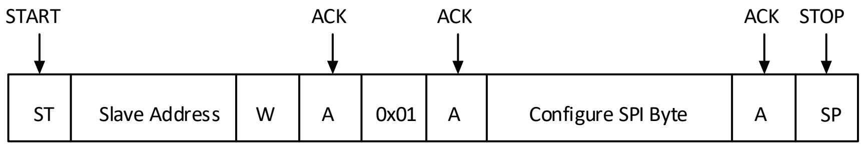


Figure 4.3. Configure SPI Master Interface Operation

Table 4.2. Bit Allocation of Configuration Data Byte

Bit	7	6	5	4	3	2	1	0
Symbol	SS4	SS3	SS2	SS1	SS0	DIRECTION	CPHA	CPOL
Reset Value	1	1	1	1	1	0	0	0

The clock polarity (CPOL) and clock phase (CPHA) determine which edge of the SPI clock signal will be used to drive or receive data. The SPI master and slave must agree to use the same clock polarity and clock phase mode for communication. This design supports all four SPI clock polarity and clock phase modes.

The bit DIRECTION specifies whether the most significant bit or least significant bit is first on the SPI bus. Logic level '0' of this bit indicates the MSB of the data to be transmitted first. Otherwise, the LSB of the data is transmitted first.

SS4 to SS0 contains the Slave Select (SS) signals to be used for transmission on the SPI bus. Five Slave Select lines can be used in this design. SS4 corresponds to pin SS_N_MASTER_4, SS3 corresponds to pin SS_N_MASTER_3, and so on. There is no restriction on the number or combination of Slave Select lines that can be enabled for a SPI transmission. If more than one SS_N_MASTER pin is enabled at a time, the user should be aware of possible contention on the data outputs of the SPI slave devices.

I^{2}C Master Write Data – Command ID 0x02

When the command byte 0x02 is sent, the I^{2}C master can start to write data to the SPI slave. 1 to 128 bytes of data can follow this command byte (see Figure 4.4.).

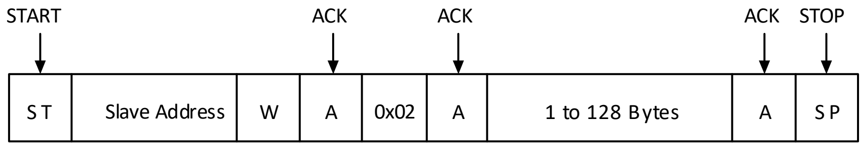


Figure 4.4. I^{2}C Master Write Data

This design places the data received from the I^{2}C master into a data buffer and continues loading the data buffer until a STOP condition on the I^{2}C bus is received. The first data from the I^{2}C master is loaded into the location with address 0 in the data buffer, the second is loaded into address 1, and so on. The data buffer is 128 bytes deep to accommodate the maximum number of data bytes.

Once the I²C slave interface receives the command 0x02 and the STOP condition on the I²C bus is detected, the SPI master interface will read data beginning with address 0 of the data buffer and write to the SPI slave through the pin MOSI_MASTER. Meanwhile, the SPI master interface reads data from the SPI slave through the pin MISO_MASTER and loads to the location with the corresponding data buffer address. After the STOP condition of the I²C bus is detected, further communication on the I²C bus will not be acknowledged until the communication is complete. Once the SPI communication is complete, the pin intn is active. At this time, the I²C master can resume communication on the I²C bus.

I²C Master Clear Interrupt – Command ID 0x03

An interrupt is generated after the SPI transmission is complete. This interrupt can be cleared if the I²C master sends a 'Clear Interrupt' command. This command ID is 0x03 (Figure 4.5.).

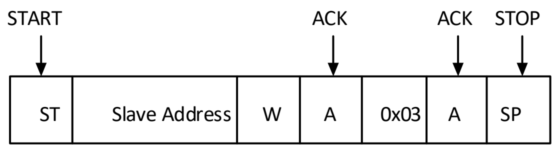


Figure 4.5. I²C Master Clear Interrupt

I²C Master Read Data – No Command ID

I²C read data requires no Command ID. The I²C slave address with the read/write bit set to a '1' will cause the I²C slave interface to send the data buffer contents to the I²C master (see Figure 4.6.). The data buffer contents are not modified during the I²C master read process.

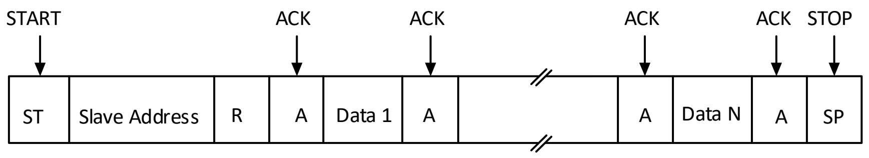


Figure 4.6. I²C Master Read Data

The first data sent to the I²C master is the content of address 0 in the data buffer, followed by the content of address 1, and so on, until the STOP condition is detected. If the number of I²C master read data is more than 128, then the 129th data sent to the I²C master is the content of address 0 in the data buffer. This is repeated until a STOP condition is detected on the I²C bus.

4.3. SPI Read and Write

Data in the data buffer will be sent to the SPI slave if the Command ID is 0x02 and the STOP condition of the I²C bus is detected. The data on the MOSI_MASTER pin is the same information as the I²C bus data without the I2C slave address and Command ID. For example, if the message shown in Figure 4.7. is transmitted on the I²C bus, the SPI master will send data to the SPI slave shown in Figure 4.8.

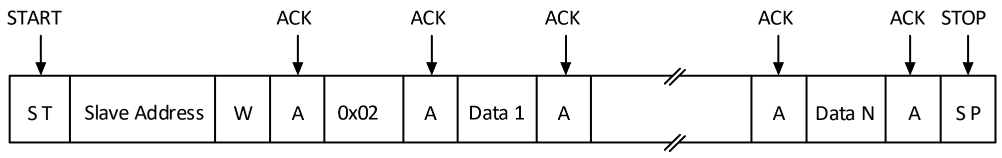


Figure 4.7. I²C Master Write Data

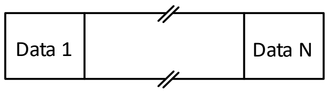


Figure 4.8. Data Appears on the MOSI_MASTER Pin

This design counts data bytes received from the I^{2}C master and sends the same number of bytes to the SPI slave. As the SPI master interface sends data through the MOSI_MASTER pin, it also reads data from the MISO_MASTER pin and saves it in the data buffer. Therefore, the data in the buffer is overwritten by the data received from the MISO_MASTER pin. The data in the data buffer can then be read back by the I^{2}C master.

The SPI master interface generates SCLK to synchronize the SPI data transfers and SCLK is only available during the data transfer. SCLK is derived from the system clock, CLK, by dividing the frequency. The divisor is determined by the parameter CLOCK_SEL (see Table 4.3.).

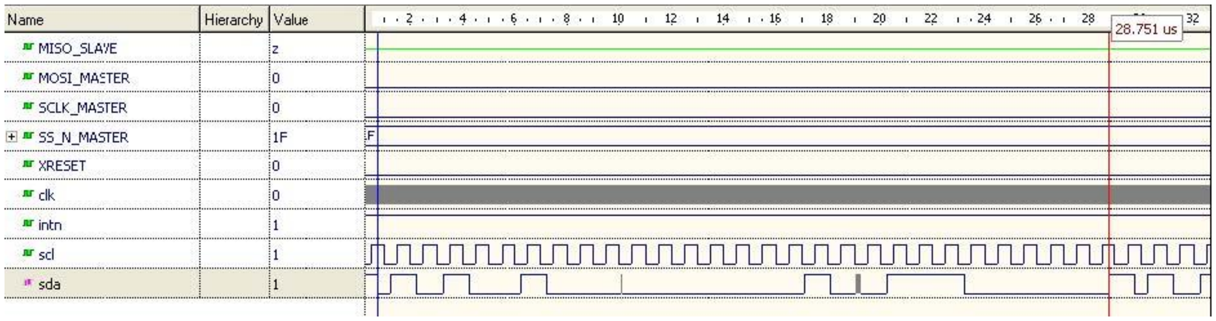
Table 4.3. Parameters I2C_SLAVE_ADDRESS and CLCOK_SEL

Parameter	Description	Value
I2C_SLAVE_ADDRESS	Specifies the I^{2}C slave address	
CLOCK_SEL	Specifies the factor for deriving SCLK from the system clock, clk. SCLK is derived using the following equation: SCLK=clk/2*(CLOCK_SEL+1)	0 to 255

The SCLK polarity and SCLK phase depends on the setting of CPOL and CPHA. Whether the most significant bit or least significant bit is first on the SPI bus depends on the value of the DIRECTION bit. During the SPI data transfer, pins SS_N_MASTER_4 to SS_N_MASTER_0 are set to the corresponding values of SS4 to SS0, otherwise these pins are set to be inactive.

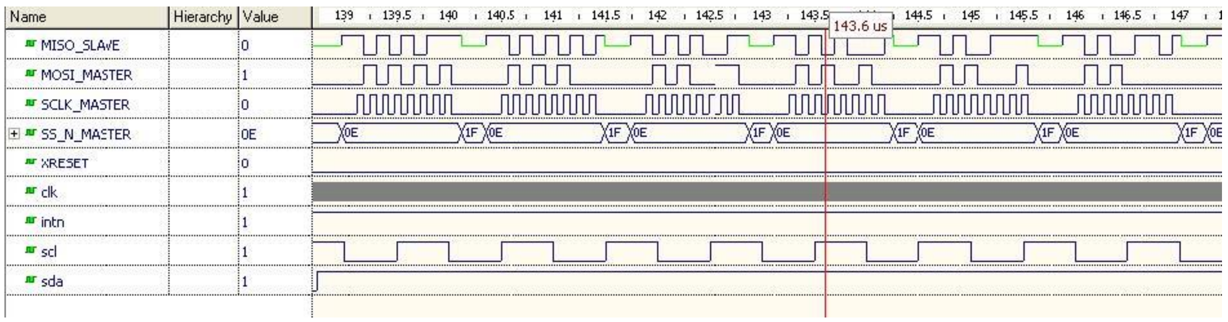
5. Test Bench Description

The I^{2}C Slave to SPI Master Bridge design is simulated using an I2C master module and a SPI slave module. The I^{2}C master module contains several tasks to emulate the Configure SPI Master Interface command, I^{2}C Master Write Data command, I^{2}C Master Clear Interrupt command and I^{2}C Master Read command. The SPI slave module emulates the responses of a SPI slave device. The simulation result shown below is based on the RTL simulation of the design.



Name	Hierarchy	Value
MISO_SLAVE		z
MOSI_MASTER		0
SCLK_MASTER		0
SS_N_MASTER		1F
XRESET		0
clk		0
intn		1
scl		1
sda		1

Figure 5.1. I^{2}C Master Sends Configure SPI Master Interface Command



Name	Hierarchy	Value
MISO_SLAVE		0
MOSI_MASTER		1
SCLK_MASTER		0
SS_N_MASTER		0E
XRESET		0
clk		1
intn		1
scl		1
sda		1

Figure 5.2. I^{2}C Master Issues STOP, Data Transfer on SPI Bus

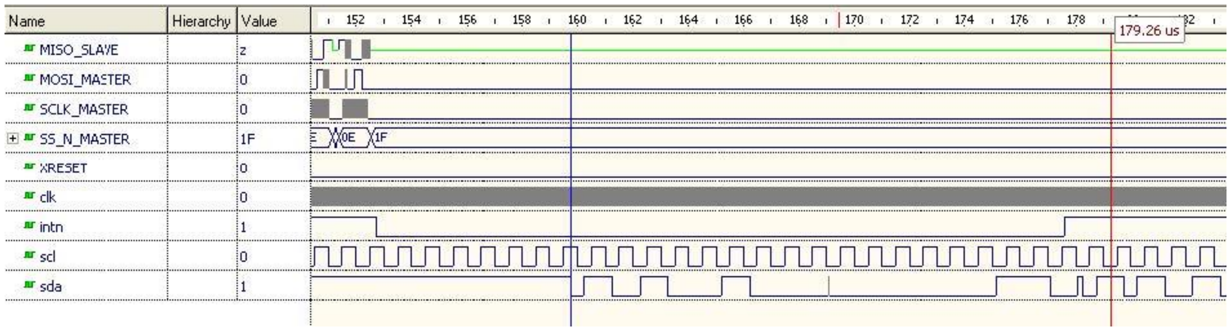


Figure 5.3. I^{2}C Master Clear Interrupt

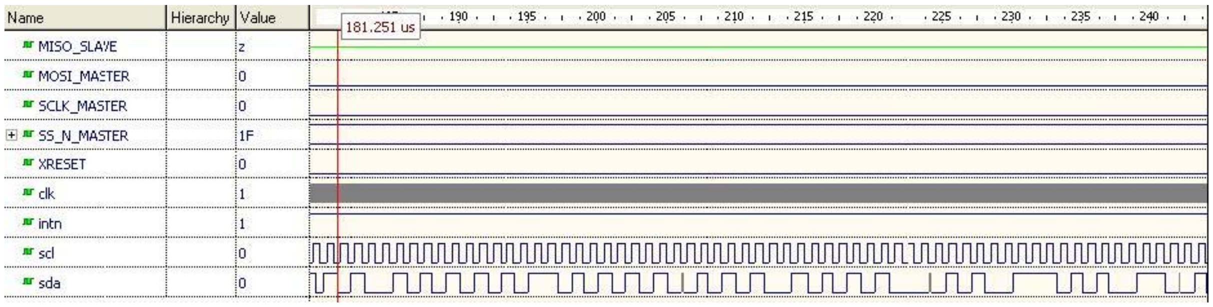


Figure 5.4. I^{2}C Master Read Data

6. Implementation

Table 6.1. Performance and Resource Utilization

Device Family[1]	Language	Speed Grade	Utilization (LUTs)	f_{MAX} (MHz)	Architecture Resources
MachXO™ [2]	Verilog	-3	215	>40	1 EBR
	VHDL	-3	213	>40	1 EBR
Platform Manager™ [3]	Verilog	-3	421	>40	N/A
	VHDL	-3	424	>40	N/A

Notes:

1. The MachXO implementation uses embedded RAM whereas the Platform Manager implementation uses distributed RAM.
2. Performance and utilization characteristics are generated using LCMXO1200C-3T100C with Lattice Diamond™ 1.1 or ispLEVER® 8.1 SP1 software. When using this design in a different device, density, speed, or grade, performance and utilization may vary.
3. Performance and utilization characteristics are generated using LPTM10-1247-3TG128CES, with Lattice ispLEVER 8.1 SP1 software. When using this design in a different device, density, speed, or grade, performance and utilization may vary.

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

Revision History

Revision 1.2, December 2019

Section	Change Summary
All	• Changed document number from RD1094 to FPGA-RD-02111. • Updated document template.
Disclaimers	Added this section.

Revision 1.1, December 2010

Section	Change Summary
Implementation	• Added support for the Platform Manager device family. • Added support for Lattice Diamond design software.

Revision 1.0, June 2010

Section	Change Summary
All	Initial release.

www.latticesemi.com