SPI to I2C Using Altera MAX Series

AN-486
2014.09.22

101 Innovation Drive
San Jose, CA 95134
www.altera.com

Contents

SPI to I^2C Using Altera MAX Series..... 1-1
Serial Peripheral Interface..... 1-2
I^2C Interface..... 1-3
Implementing the Design..... 1-4
Document Revision History..... 1-4

SPI to I²C Using Altera MAX Series

1

2014.09.22

AN-486 Subscribe Send Feedback

Altera® MAX® II, MAX V, and MAX 10 FPGA devices serve as a bridge between a host that has serial peripheral interface (SPI) to communicate with devices connected through an I²C bus.

The I²C is a serial, two-wire, low-bandwidth, industry standard protocol used in embedded systems to communicate with various low-speed peripheral devices. The SPI is a widely used, fast, four-wire, full duplex, serial communication interface.

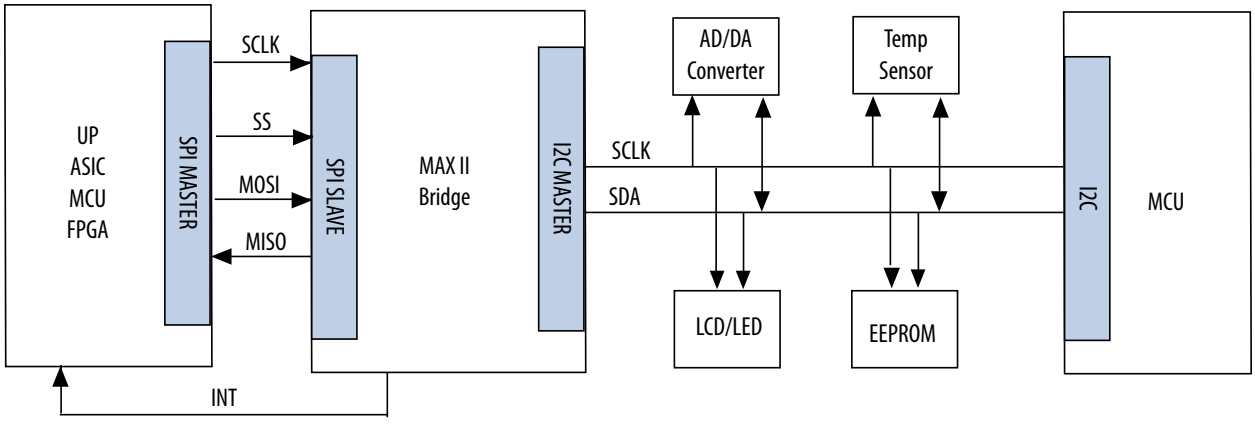
Many embedded systems today have SPI interfaces, making it difficult to connect them with peripheral devices in an I²C fashion. You can make the connection by modifying the system, but this is economically inefficient. The best solution is to use Altera devices as a bridge to connect the two interfaces.

You can use the MAX II, MAX V, or MAX 10 FPGA devices to implement the bridge. Altera devices provide greater flexibility, consume less power, and can be economically integrated into the embedded system. The MAX II, MAX V, or MAX 10 FPGA device acts as an SPI slave to the host (SPI master) and acts as a master to the I²C bus.

The provided designs enable an SPI-equipped host to control data flow to other devices such as Analog-to-Digital (AD) converter, LED controller, audio processor to read temperature sensors, hardware monitors, and diagnostic sensors that are on an I²C interface.

Figure 1-1: Implementing an SPI to I²C Interface Using a MAX II CPLD

The figure below shows a block diagram of a design example that uses a MAX II device.



The bridge interfaces with the SPI host as an SPI slave using four wires, SS and SCLK signals for control, and MISO and MOSI signals for data. The side interfacing with the I^2C bus has two wires, and SCLK and SDA signals.

Related Information

- **Design Example for MAX II**
  Provides the design files for this application note (AN 486).
- **Design Example for MAX 10**
  Provides the MAX 10 design file for this application note (AN 486).

Serial Peripheral Interface

The SPI bus has only one master, which is connected to many slaves.

Altera device acts as one of the slaves to the SPI master device.

Figure 1-2: Timing Diagram for SPI

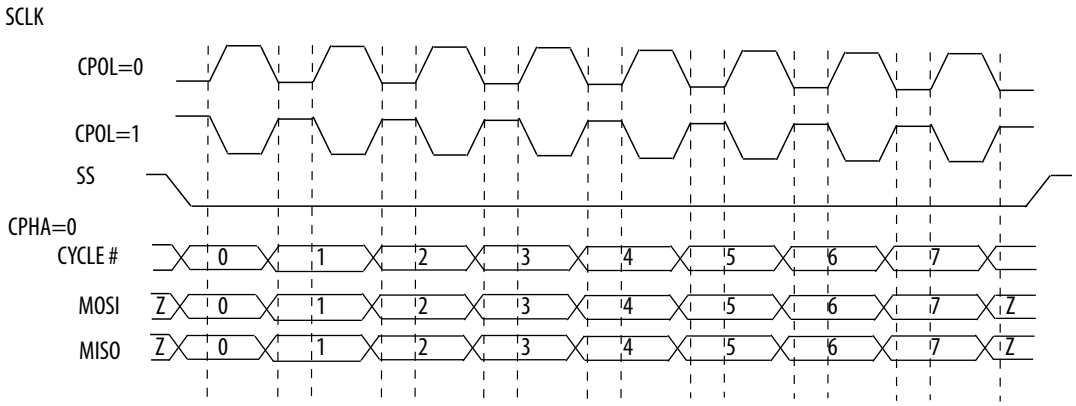


Table 1-1: SPI Interface Pins

The table below describes the SPI pins.

Signal	Direction	Function
SS	Input (active low)	Slave select
SCLK	Input	SPI cloc
MISO	Output	Master-in Slave-out
MOSI	Output	Master-out Slave-in

The SPI sends:

- command register (8 bits)
- data in (8 bits)

The SPI receives:

- status register (8 bits)
- data out (8 bits)

The SPI word length is fixed at 16 bits.

In every SPI word, the command register dictates the functions on the I²C bus, and the data in holds the data to be sent by the I²C bus. Similarly, the last bit of the status register is the acknowledge bit and the data out is the data received over the I²C line in the previous I²C cycle.

At the end of every SPI bus:

- The slave select line goes high; indicating a word complete.
- The master executes an I²C bus as per the value of command register at that time.

After a fixed delay, depending on the frequency of the I²C SCL, another SPI word can be sent. The minimum delay between two SPI words is the I²C SCL clock frequency.

I²C Interface

Altera device acts as a master to the I²C bus.

Because the designs are meant to provide an interface between an SPI master and an I²C device, multi-master support is not provided on the I²C bus.

Table 1-2: I²C Interface Pins

The table below describes the I²C interface pin.

Signal	Direction	Function
SCLK	Output	I²C serial clock
SDA	Bidirectional	I²C data bus

The I²C functions are carried out based on the command register value received from the SPI side.

Table 1-3: I²C Commands

The table below describes the significance of the value stored in the command register.

Command Register	Data In Register	Function on the I²C Line
10000000	Slave address + R/W	Start/repeat start
01000000	Data to be written	Write a byte
00100000	Don't care	Read a byte
00010000	Don't care	Stop
00000000	Don't care	Null, wait state

The data read in a particular I²C transaction is stored in the data out register and is read by the SPI master in its next SPI transaction. The last command word, 00000000 (b), is required for the SPI master to read the value of status and data out registers without doing anything on the I²C bus.

Figure 1-3: I²C Command Format

Implementing the Design

You can implement the design by using the source code and allocating the appropriate signal and control lines to the general purpose I/O (GPIO) lines of the Altera devices. You require an SPI master and an I²C slave as additional resources to demonstrate this implementation.

The MAX II design uses an EPM240 device. You can also implement this application in MAX V and MAX 10 devices.

Note: The MAX II design has been implemented in Verilog and successful operation has been demonstrated using the MDN-B2 demo board. The source code, testbench, and the complete Quartus II project are available in the provided design example files.

Document Revision History

Date	Version	Changes
September 2014	2014.09.22	• Added MAX V and MAX 10 devices. • Removed outdated information from the implementation section.
December 2007	1.0	Initial release.