Abstract

Today, at the low end of the communication protocols we find the inter-integrated circuit (I2C) and the serial peripheral interface (SPI) protocols. Both protocols are well suited for communications between integrated circuits for slow communication with on-board peripherals. The two protocols coexist in modern digital electronics systems, and they probably will continue to compete in the future, as both I2C and SPI are actually quite complementary for this kind of communication. I2C and SPI are the most commonly used serial protocols for both inter-chip and intra-chip low/medium bandwidth data-transfers. This paper contrasts and compares physical implementation aspects of the two protocols through a number of
recent Xilinx’s FPGA families, showing up which protocol features are responsible of substantial area overhead. This valuable information helps designers to make careful and tightly tailored architecture decisions. For a comprehensive comparative study, both protocols are implemented as general purpose IP solutions, incorporating all necessary features required by modern ASIC/SoC applications according to a recent market investigation of an important number of commercial I2C and SPI devices. The RTL code is technology independent, inducing around 25% area overhead for I2C over SPI, and almost the same delays for both designs.

Keywords

Introduction

Today, for the low end from the communication protocols look for the inter-integrated circuit (I2C) and also the serial peripheral interface (SPI) protocols. Both protocols are very well best for communications between integrated circuits for slow communication with on-board peripherals. Both I2C and SPI protocols coexist in electronics communication systems, and they will probably continue to compete. Mostly, I2C and SPI are the serial protocols for both inter-chip and intra-chip data-transfers. This paper contrasts and compares physical implementation areas of both protocols via a quantity of recent Xilinxs FPGA families, listed which protocol features account of substantial area overhead. This informative information helps designers to produce careful and tightly tailored architecture decisions. To get a comprehensive comparative study, both protocols are implemented as general purpose IP solutions, incorporating all necessary features essential to modern ASIC/SoC applications based on a recent market investigation of the important quantity of commercial I2C and SPI devices. The RTL code is technology independent, inducing around 25% area overhead for I2C over SPI, and almost similar delays for both designs.

Serial Communication Protocols

SPI Protocol

Serial peripheral Interface (SPI) allows full duplex synchronous data communication between the FPGA and other peripheral devices (including Sensor modules).
There are two modes of operation: Master and Slave, here the focus is on SPI slave. The SPI slave could possibly be in Run mode, Wait mode and forestall mode. You could find four different external signals of SPI slave MISO (Master In Slave Out), MOSI (Master Out Slave In), Clock (System Clock) and SS (Slave Select). Multiple slave products are allowed with individual slave select lines. To start with a communication, the master first configures the hands of your energy. The master then transmits the best slave select bit to your desired chip into a logic 0. A logic 0 is transmitted because the slave select line is active low.

During each SPI clock cycle, a full duplex data transmission occurs:

1. The master sends a bit on the MOSI line; the slave reads it from that same line,
2. The slave sends a bit on the MISO line; the master reads it from that same line.

Transmissions involve shift registers of eight bits each. Data are usually shifted out with an increase of significant bit first, while shifting a different least significant bit in to the same register. Next register have been shifted out, the master and slave have exchanged register values. Then this device takes that value and writes it to memory. If there are other data to exchange, the shift registers include things like new data plus the process repeats. Transmissions may involve numerous clock cycles. When there’s no more data for being transmitted, the master deselects the slave.

Transmissions involve shift registers of eight bits each. Data will often be shifted out with significant bit first, while shifting a new least significant bit into your same register. There after register have been shifted out, the master and slave have exchanged register values. After that your device takes that value and writes it to memory. If there are numerous data to modify, the shift registers include new data along with the process repeats. Transmissions may involve many clock cycles. When there isn’t a more data being transmitted, the master deselects the slave.
I2C Protocol

The I2C protocol was handed by Philips Semiconductors to allow faster devices to say with slower devices plus allow devices to communicate together.
spanning a serial data bus without data loss. I2C enabled microcontrollers like PIC18F452 from Atmel and TMS470 from Texas Instruments needs a many programming and knowledge of the register structures for configuring it. Hence they may not be portable. We here present a sort of I2C bus controller.

The EEPROM, ADC and RTC will demand an interface for communication totally. So I2C bus is utilized as a possible interface totally. It really is helpful to minimize system-level interconnect. This simplifies it level design plus the design with the mother-board and associated chip-boards. Moreover transmitting information above the I2C bus will improve system performance since transmission of digital information is less be subject to interference from environmental noise sources.

The I2C Controller Bus is really a two-wire, bi-directional serial bus that

![I2C Bus Connection Topology](image)

Figure 2: I2C Bus Connection Topology. Virtually any number of slaves and any number of masters can be connected onto these two signal lines called as Serial Data (SDA) and Serial Clock (SCL) [1]

delivers an efficient and efficient technique of data transmission within the short distance between many devices. I2C provides good support for communication with a few other slow, on-board peripheral devices which are accessed intermittently, while being extremely modest in their hardware resource needs. That is a simple, low-bandwidth, short-distance protocol. I2C is straightforward to make use of to link multiple devices together as it possesses a built-in address [1, 2].

Each I2C signals are serial data (SDA) and serial Clock (SCL) as shown in figure 2. The device that initiates a transaction throughout the I2C bus is
termed the master. The master normally controls the clock signal. A pc device being addressed from the master is known as a slave [1, 2]. The I2C protocol supports multiple masters, but a majority of system designs include only one. There could often be many slaves from the bus. Both masters and slaves can receive and transmit data bytes. Standard I2C devices operate up to 100Kbps, while fast-mode devices operate at around 400Kbps. A lot of the I2C devices on the market today support 400Kbps operation. Higher-speed operation may allow I2C to keep up with increasing interest on bandwidth in multimedia along with other applications [2].

**Comparison between SPI and I2C Protocols**

Figure 5 shows the comparison between SPI and I2C Protocols.

**Simulation Results**

**SPI Master Controller**

Table 2 shows the synthesis summary of hardware utilization and speed for the algorithm of SPI Master Controller on different FPGA processors. Figure 5 shows the simulation results for the algorithm of SPI Master Controller implemented using verilog HDL on Xilinx ISE Project Navigator Ver. 13.4 and ISE Simulator [3]. Algorithm of SPI Master Controller have been tested and verified on Digilent NEXYS2 FPGA board [4] containing Spartan3E [5] XC3S1200E FPGA processor.
Table ?? shows the synthesis summary of hardware utilization and speed for the algorithm of I2C Master Controller on different FPGA processors. Figure ?? shows the simulation results for the algorithm of I2C Master Controller implemented using verilog HDL on Xilinx ISE Project Navigator Ver. 13.4 and ISE Simulator [3]. Algorithm of I2C Master Controller have been tested
Table 1: Synthesis Summary for SPI Master Controller.

<table>
<thead>
<tr>
<th>FPGA Processor</th>
<th>Speed Grade</th>
<th>Number of Slices Used</th>
<th>Number of Slice Flip Flops Used</th>
<th>Number of 4 input LUTs Used</th>
<th>Number of bonded IOBs Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spartan 3E XC3S500E -5</td>
<td>51 out of 4656</td>
<td>65 out of 9312</td>
<td>92 out of 9312</td>
<td>21 out of 232</td>
<td></td>
</tr>
<tr>
<td>Spartan 3E XC3S1200E -5</td>
<td>51 out of 8672</td>
<td>65 out of 17344</td>
<td>92 out of 17344</td>
<td>21 out of 250</td>
<td></td>
</tr>
<tr>
<td>Spartan 6 XC6SLX25 -3</td>
<td>44 out of 30064</td>
<td>63 out of 15032</td>
<td>89 out of 486</td>
<td>21 out of 226</td>
<td></td>
</tr>
<tr>
<td>Virtex 4 XC4VFX100 -12</td>
<td>43 out of 42176</td>
<td>52 out of 84352</td>
<td>81 out of 84352</td>
<td>21 out of 576</td>
<td></td>
</tr>
<tr>
<td>Virtex 5 XC5VFX100T -3</td>
<td>51 out of 64000</td>
<td>63 out of 64000</td>
<td>89 out of 339</td>
<td>21 out of 680</td>
<td></td>
</tr>
<tr>
<td>Virtex 6 XC6VCX130T -2</td>
<td>47 out of 160000</td>
<td>60 out of 80000</td>
<td>90 out of 408</td>
<td>21 out of 240</td>
<td></td>
</tr>
</tbody>
</table>


![Simulation Results for the I2C Master Controller Algorithm](image)

**Figure 5:** Simulation Results for the I2C Master Controller Algorithm

## Conclusion and Future Work

Floating Point Real-Time Sensor Data Acquisition on an FPGA based Processor System is an important project in the field of microelectronics and is broadly

Amit Kumar Shrivastava and Himanshu Joshi

Design, Implementation and Functional Verification of Serial Communication Protocols (SPI and I2C) on FPGAs.
Table 2: Synthesis Summary for I2C Master Controller.

<table>
<thead>
<tr>
<th>FPGA Processor</th>
<th>Speed Grade</th>
<th>Number of Slices Used</th>
<th>Number of Slice Flip Used</th>
<th>Number of Slice Flops Used</th>
<th>Number of 4 Input LUTs Used</th>
<th>Number of Bonded IOBs Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spartan 3E</td>
<td>-5</td>
<td>45 out of 4656</td>
<td>60 out of 9312</td>
<td>81 out of 9312</td>
<td>29 out of 232</td>
<td></td>
</tr>
<tr>
<td>XC3S500E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spartan 3E</td>
<td>-5</td>
<td>45 out of 8672</td>
<td>60 out of 17344</td>
<td>81 out of 17344</td>
<td>29 out of 250</td>
<td></td>
</tr>
<tr>
<td>XC3S1200E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spartan 6</td>
<td>-3</td>
<td>41 out of 30064</td>
<td>57 out of 15032</td>
<td>78 out of 486</td>
<td>29 out of 226</td>
<td></td>
</tr>
<tr>
<td>XC6SLX25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Virtex 4</td>
<td>-12</td>
<td>37 out of 42176</td>
<td>48 out of 84352</td>
<td>75 out of 84352</td>
<td>29 out of 576</td>
<td></td>
</tr>
<tr>
<td>XC4VFX100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Virtex 5</td>
<td>-3</td>
<td>45 out of 64000</td>
<td>57 out of 64000</td>
<td>78 out of 339</td>
<td>29 out of 680</td>
<td></td>
</tr>
<tr>
<td>XC5VFX100T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Virtex 6</td>
<td>-2</td>
<td>40 out of 160000</td>
<td>55 out of 80000</td>
<td>81 out of 408</td>
<td>29 out of 240</td>
<td></td>
</tr>
<tr>
<td>XC6VCX130T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Applicable in a number of automatic test and measuring equipments. Such a system/device can be used to collect the required data from any peripheral input devices, such as meters, sensors and etc. via HDL based controlling software. The measured data then further can be stored in the memory device such as SD card and can be retrieved with the help of a PC for further analysis. The recorded values can be shown numerically whereas their relationship can be displayed graphically as a curve on the screen. This work proposed a design of the data acquisition system using FPGA interfaced to an external memory card. The system will have capability to receive the digital signals from multi-channels sensors.

The scope of this thesis work was to develop reconfigurable hardware modules on FPGA, that can together form a Real-Time Sensor Data Acquisition System. We have accomplished the major two reconfigurable modules that are required to develop an FPGA based Floating Point Real-Time Sensor Data Acquisition System.
As a future work, we would like to continue this project to make it a complete, functional, low-cost, robust hardware implementation, which includes verification of this module in a real-time commercial/industrial application.

References


This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution 3.0 Unported License (http://creativecommons.org/licenses/by/3.0/).

©2013 by the Authors. Licensed and Sponsored by HCTL Open, India.

Amit Kumar Shrivastava and Himanshu Joshi

Page 10 of 10

Design, Implementation and Functional Verification of Serial Communication Protocols (SPI and I2C) on FPGAs.