HCTL Open Int. J. of Technology Innovations and Research
HCTL Open IJTIR, Volume 4, July 2013
e-ISSN: 2321-1814
ISBN (Print): 978-1-62776-132-1

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

Amit Kumar Shrivastava [*] ***and Himanshu Joshi*** [†]
amit0404@gmail.com

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

[*] M.Tech. Student, Department of ECE, Jagan Nath University, Jaipur, India
[†] Asst. Professor, Department of ECE, Jagan Nath University, Jaipur, India

HCTL Open Int. J. of Technology Innovations and Research
HCTL Open IJTIR, Volume 4, July 2013
e-ISSN: 2321-1814
ISBN (Print): 978-1-62776-132-1

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

SPI, I2C, Serial Communication Protocols, FPGA, Reconfigurable Computing.

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).

e-ISSN: 2321-1814
ISBN (Print): 978-1-62776-132-1

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.

HCTL Open Int. J. of Technology Innovations and Research
HCTL Open IJTIR, Volume 4, July 2013
e-ISSN: 2321-1814
ISBN (Print): 978-1-62776-132-1

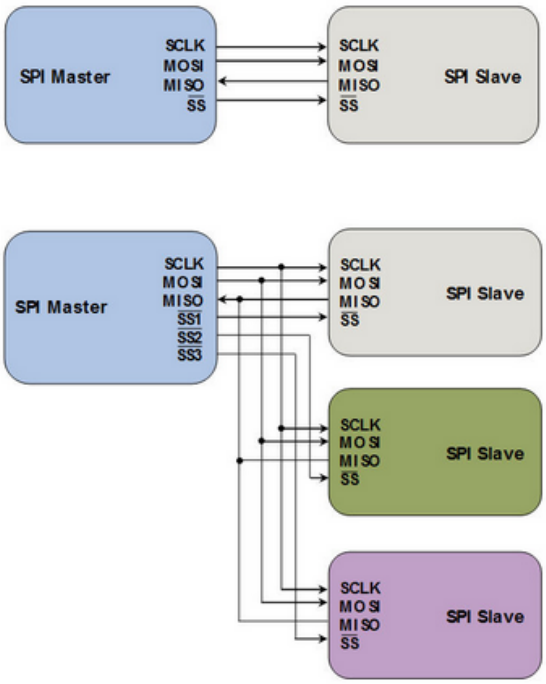
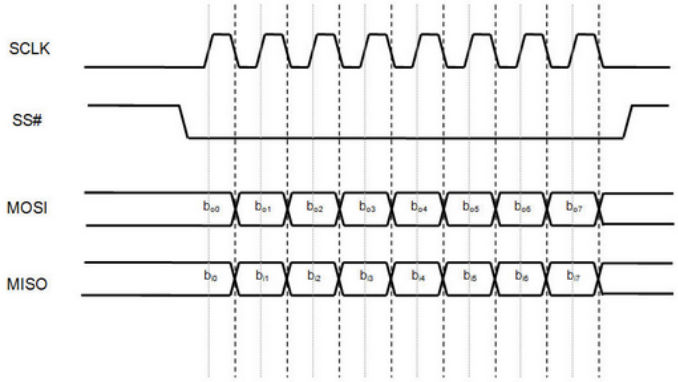


Figure 1: *(SPI) Serial Peripheral Interface Bus Topologies. (a) SPI master connected to a SPI slave (one-to-one connection). (b) SPI master connected to multiple SPI slaves (one-to-many connection) [1]*



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

e-ISSN: 2321-1814
ISBN (Print): 978-1-62776-132-1

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

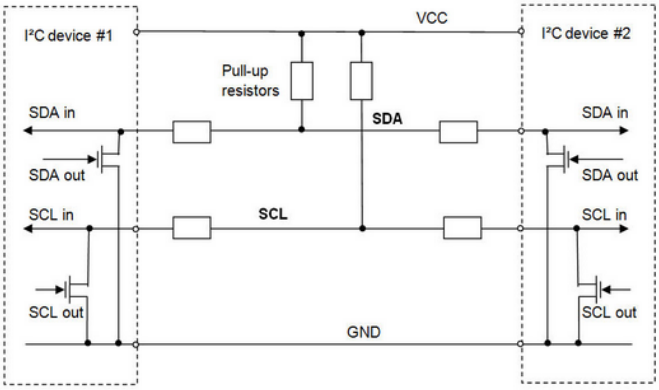
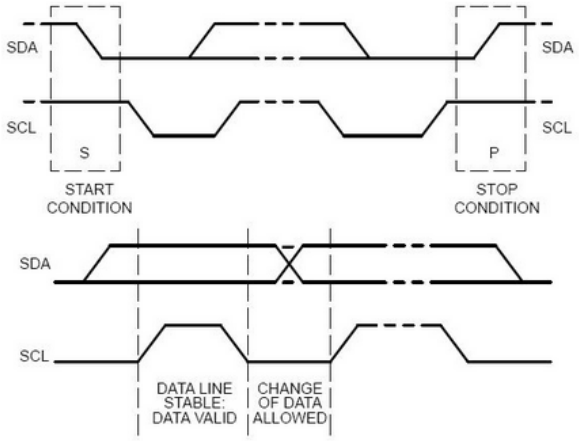


Figure 2: *I2C Bus Connection Topology. Virtually any number of slaves and any number of masters can be connected onto these two signal lines called as Serial Data (SDA) and Serial Clock (SCL) [1]*

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

HCTL Open Int. J. of Technology Innovations and Research
HCTL Open IJTIR, Volume 4, July 2013
e-ISSN: 2321-1814
ISBN (Print): 978-1-62776-132-1



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.

HCTL Open Int. J. of Technology Innovations and Research
HCTL Open IJTIR, Volume 4, July 2013
e-ISSN: 2321-1814
ISBN (Print): 978-1-62776-132-1

	I²C	SPI
Originator	Philips (1982)	Motorola (1979)
Plug & Play	Yes	No
Interface type	Serial (2 wires)	Serial (3+N wires)*
Distance	Short (In-box communication)	
Application	Multi-master register-access	Transfer of data-streams
Protocol Complexity	Low	Lower
Design Cost	Low	Lower
Transfer rate	Limited (100 & 400 KHz and 3.4 MHz)	Free (n x MHz to 10n x MHz)
Power Consumption	Low (2 pull-up resistors)	Lower
Transfer type	Half Duplex	Full Duplex
Time Constraint	Synchronous	
Multi Master+	Yes	No
Multi Slave	Yes	Yes
I/O constraints	Open-drain with pull-up resistors	No constraint
Addressing+	Software (7/10 bits)	Hardware (Chip Select)
Flow Control+	Yes	No
Clock Stretching+	Yes	No

*: N is the number of devices connected to a single master on the bus.
+: Feature inducing substantial area overhead.

Figure 3: *Comparison between SPI and I2C Protocols [2]*

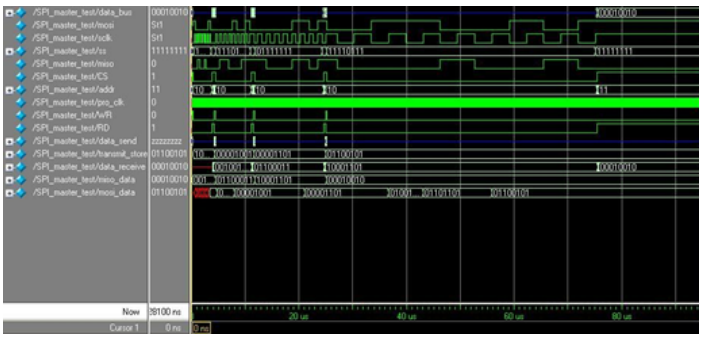


Figure 4: *Simulation Results for the SPI Master Controller Algorithm*

I2C Master Controller

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

HCTL Open Int. J. of Technology Innovations and Research
HCTL Open IJTIR, Volume 4, July 2013
e-ISSN: 2321-1814
ISBN (Print): 978-1-62776-132-1

Table 1: *Synthesis Summary for SPI Master Controller.*

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

and verified on Digilent NEXYS2 FPGA board [4] containing Spartan3E [5] XC3S1200E FPGA processor.

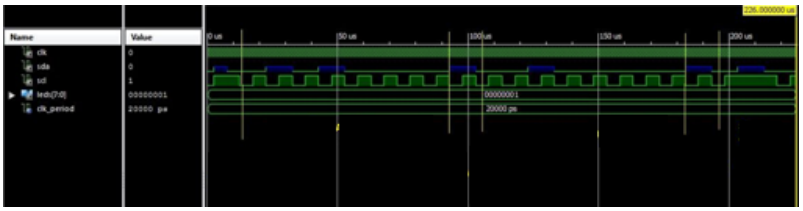


Figure 5: *Simulation Results for the I2C Master Controller Algorithm*

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

e-ISSN: 2321-1814
ISBN (Print): 978-1-62776-132-1

Table 2: *Synthesis Summary for I2C Master Controller.*

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

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

[1] Frederic Leens, An Introduction to I2C and SPI Protocols, IEEE Instrumentation & Measurement Magazine, Feb 2009.

[2] A.K. Oudjida, M.L. Berrandjia, R. Tiar, A. Liacha, K. Tahraoui, FPGA Implementation of I2C & SPI Protocols: A Comparative Study, IEEE Conference, 2009.

[3] The I2C Bus specifications version 2.1 January 2000.

[4] UM10204 I2C -bus specification and user manual Rev. 4 - 13 February 2012.

[5] Xilinx ISE Project Navigator Ver.13.4 and Xilinx ISE Simulator, Available online at: http://www.xilinx.com/products/design-tools/ise-design-suite/index.htm (last accessed on 26-May-2012).

[6] Digilent NEXYS2 FPGA board, Available online at: http://www.digilentinc.com/Products/Detail.cfm?Prod=NEXYS2 (last accessed on 26-Oct-2012).

[7] Xilinx Spartan3E FPGA Datasheet, Available online at: http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf (last accessed on 03-Oct-2012).

[8] Raj Gaurav Mishra and Amit Kumar Shrivastava, Implementation of Custom Precision Floating Point Arithmetic on FPGAs, HCTL Open International Journal of Technology Innovations and Research, Volume 1, January 2013, Pages 10-26, ISSN: 2321-1814, ISBN: 978-1-62776-012-6.