DESIGN OF SPI IP TO COMMUNICATE WITH I2C BUS OF A MICROCONTROLLER

1BHAGYASHRI G.ALLAWAGOL, 2VIDYASHREE AC, 3NARENDRA KUMAR

1,2PG Scholar, RNSIT Bangalore-98
3Assistant Professor, RNSIT Bangalore-98
E-mail: 1pattuallawagol@gmail.com, 2vidyashree0102@gmail.com, 3nkrnsit@gmail.com

Abstract- Serial peripheral interface is one of the most commonly used protocols for medium band-width data transfer. Primarily developed by Motorola, it is used worldwide in many peripheral devices. With a focus on hand-shaking between master and slave, this is an attempt to implement SPI protocol on FPGAs. The purpose of this paper is to provide a full description of an up-to-date SPI Master/Slave FPGA implementations. All related issues, starting from the elaboration of initial specifications, till the final system verification, are comprehensively discussed and justified. The whole design code, either for synthesis or verification, is implemented in Verilog 2001 (IEEE 1365). The RTL code is technology independent, achieving a transfer rate of 71 and 75MBPS for the Master and the Slave, respectively, when mapped onto Xilinx Virtex 5 FPGA devices.

Index Terms- Serial Peripheral Interface (SPI), Intellectual Property(IP), Inter-Integrated circuit(I2C).

I. INTRODUCTION

In the world of communication protocols, SPI is often considered as little communication protocol compared to Ethernet, USB, SATA, PCI-Express and others that present throughput in the 100 Mb/s range, if not Gb/s. It is important to not forget the purpose of each protocol. Ethernet, USB, and SATA are meant for outside the box communications and data exchanges between whole systems. While SPI, as well as others serial protocols such as I2C and 1-wire for instance, are well suited for communications between integrated circuits for low/medium data transfer speed with on-board peripherals.

Consequently, when there is a need to implement a com-munication between an integrated circuit such as a microcon-troller and a set of relatively slow peripherals, there is no point in employing any excessively complex protocols. In this case, SPI stands among the best candidates. This is why it becomes a worldwide standard for modern digital electronics systems and it will probably continue to compete in the future [5][6][7][8].

In our attempt to implement universal SPI and I2C IP cores according to the design-reuse methodology, we first made a market study of an important number of recent commercial SPI and I2C devices (datasheets) from different vendors to look at the requirements and what features are to be included to satisfy modern ASIC/SoC applications.

The key features required for SPI Master/Slave IP cores as a result of the market investigation are summarized in Table I, and their translation into architectures are depicted by figures 1 and 2, respectively.

I2C and SPI are the two widely-used bus protocols in todays embedded systems. The I2C 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 SPI slave to SPI master. This allows the microcontroller to communicate directly with the SPI bus through its I2C bus as shown in fig (3).
II. METHODOLOGY

The Serial Peripheral Interface bus is a synchronous, serial communication link that operates in full duplex, meaning that a device transmits and receives data simultaneously. The devices communicate as a Master/Slave, where the Master initiates communication by selecting a Slave device with a hardware line and also provides the synchronous clock used to shift data bits in and out of the Slave. The signals required for communication are the Slave Select (SS), Master In Slave Out (MISO), Master Out Slave In (MOSI), and Serial Clock (SCK). The advantages of SPI over other communication protocols is that the addressing is performed in hardware with the SS line, making it faster to address a device, and that communication is full duplex, allowing for faster transfers of data as shown in figure(4).

SPI communication begins with the Master asserting the SS line. Depending upon the device, the SS line might be active high or active low. The Master must then wait at least one clock period before starting communication. Much like the active polarity of the SS line, the waiting period after SS activation varies from device to device. As an example, an analog-to-digital converter might require that the Master wait for a conversion to be completed after its SS line has been asserted. Next, the Master will begin shifting data out of the MOSI line, and it will shift data in on the MISO. Data is always transferred as full duplex, even when that data is not meaningful. As an example, for a Master to receive 24 bits of data from a Slave device, it must also transmit 24 bits to the Slave device.
III. IMPLEMENTATION RESULTS

The whole design code, either for synthesis or functional verification, is implemented in Verilog 2001 (IEEE 1365). The synthesis design code is technology independent and was simulated at both RTL and gate level (post place and route net list) with timing back annotation using Model Sim SE 6.3f and mapped onto Xilinx FPGAs using Foundation ISE 10.1 version.

The RTL Code size of SPI-Master is about 1.33 times the code size of SPI-Slave. The 33 percent extra code size is mainly due to the additional logic required by the SPI-Master to handle the clock stretching feature (wait-states insertion periods) to cope with different unavailability situations either for the Master or for the Slave.

The RTL Code size of I2C-Slave is about 1.33 times the code size of SPI-Slave. The 44 percent extra code size is due to additional logic required by I2C Slave to handle software addressing, control flow and clock stretching fault. In this paper we are implementing on chip communication using SPI which helps us in communication with I2C Bus of a microcontroller.

CONCLUSION

This Paper has shown the results of an up to date FPGA implementation of Master/slave sides of standard SPI Protocol which are

Utilization ratio of 4 percent and 2 percent for SPI-Master and SPI-Slave, respectively, when mapped on to the smallest Virtex-5 FPGA device.

A maximum transfer rate of 71 and 75 MBPS for SPI-Master and SPI-Slave respectively.

FUTURE SCOPE

I2C (Inter-Integrated-Circuit) is a multi-master serial communication bus protocol invented by Philips (now NXP). It is used to connect low speed peripherals on a motherboard, embedded system, or other systems with wired connections. Serial Peripheral Interface (SPI) is a synchronous high speed serial data link that operates in full duplex mode. PSoC's capability to manage both is used to make an I2C to SPI bridge.

REFERENCES


