International Journal of Electrical, Electronics and Computer Engineering **3**(1): 24-26(2014)

ISSN No. (Online): 2277-2626

Optimal Implementation of UART-SPI Interface with Multiple Slaves in SoC

Pratima Sharma* and Parag Parandkar*

**Department of Electronics and Communication Engineering, Oriental University, Indore, (MP)*

(Corresponding author Pratima Sharma)

(Received 05 January, 2014 Accepted 04 February, 2014)

ABSTRACT: This paper details the design and implementation of SoC's UART -SPI interface. The UART-SPI interface provides usage for the universal asynchronous receiver/transmitter (UART) to serial peripheral interface (SPI). This interface can be used to communicate to SPI slave devices from a PC with UART port. The interface consists of three blocks: the UART interface, the UART-SPI interfacing block and the SPI Master interface and slave is interface with master.

Keywords: SOC, UART, SPI, power optimization.

I. INTRODUCTION

An UART is a device allowing to the transmission and reception of information, in a serial and asynchronous way . universal asynchronus receiver and transmitter are used asynchronus serial data communication between remote communication system. The UART can be used to control the process of breaking parallel data from the pc down in to serial data that can be transmitted. One transmitter module and receiver module UART has been an important input output tools for decades and is still widely used. UART is used for communication between two device SPI is a full duplex serial bus commonly used because its simply hardware interface requirement and protocol flexibility .SPI consists of two block SPI maser and SPI slave. SPI master which is being used in this design implement the design functionality of the SPI protocol .SPI protocol specifies four signal wires MISO- master out slave in (output from master), MOSI- master in slave out (output from slave), SCLK – serial clock (clock output from master), and SS – slave select (active low, output from master).

The SPI master block generates the control signal to interface to external slave device using the serial data out port (MOSI),serial data in port (MISO),output clock (SCLK) and slave select (ss). the ss signal must be used if more than one slave exit in the system .this signal is most often low ,so a low on this line will indicate the spi is active ,while a high will signal ,inactivity . UART to SPI interfacing block that is the middle block joins the UART to SPI master. It helps the interconnection between these two interfaces.

The main advantages is ,the UART-SPI interface can fit in any application where an SPI device has be used .as the UART-SPI interface can be used to communicate to SPI slave device from a PC with UART point it can be used for typical application like interfacing of EEPROM , flash memories.

II. SYSTEM-ON-CHIP

System-on-chip (SOC) is an integrated circuit that includes a processor, a bus, and other elements on a single monolithic substrate. In recent years, there have been great advancements in the speed, power, and complexity of integrated circuits, such as application specific integrated circuit (ASIC) chips, random access memory (RAM) chips, microprocessor chips, and the like. These advancements have made possible the development of system-on-a-chip (SOC) devices.

The empirical law of moore does not only describe the increasing density of transistor permitted by technological advance .it also imposes new requirement and challenges .system complexity increase at the same speed .now a day system could never be designed using the same approach applied 20 years ago . New architectures are and must be continuously conceived. . It is clear now that Moore's law for the last two decades has enabled three main revolutions. The first revolution in the mid-eighties was the way to embed more and more electronic devices in the same silicon die; it was the era of System on Chip. One main challenge was the way to interconnect all these devices efficiently. For this purpose, the Bus inter-connect structure was used for the VLSI subsystem.

A system usually has an embedded user interface as a form of software and encompasses many components inside, not only the hardware but also the software that constitutes the system. Such a complicated entity can be handled only with computer-aided design tools, automatic synthesis of the physical layouts, and sound software engineering knowledge. In addition, the system functions to achieve a specific goal, as a whole, are usually described in algorithms that should satisfy.

A. Need for UART_SPI Interface

The UART_SPI interface a special type of interface because the SPI is a synchronous bus and the UART is an asynchronous bus.

UART can communicate with only one peripheral. This type of interface is necessary where PC is wants to communicate with the SPI slaves through UART port of PC, in this the SPI-UART interface is useful and also like this we need this type of interface in many applications. Using the UART-SPI Interface, UART can communicate with more number of devices.

III. UART DESIGN

An UART (Universal Asynchronous Receiver/Transmitter) is the microchip with programming that controls a computer's interface to its attached serial devices. UART is an integrated circuit designed for implementing the interface for serial communications. It provides the computer with the RS-232C Data Terminal Equipment (DTE) interface so that it can "talk" to and exchange data with modems and other serial devices. As part of this interface, the UART also:

- Converts the bytes it receives from the system along parallel circuits into a single serial bit stream for outbound transmission
- On inbound transmission, converts the serial bit stream into the bytes that the system handles
- Adds a parity bit (if it's been selected) on outbound transmissions and checks the parity of incoming bytes (if selected) and discards the parity bit
- Adds start and stop delineators on outbound and strips them from inbound transmissions
- May handle other kinds of interrupt and device management that require coordinating the on-chip communication of operation with high speed devices. The UART includes both transmitter and receiver. The transmitter is a special shift register that loads data

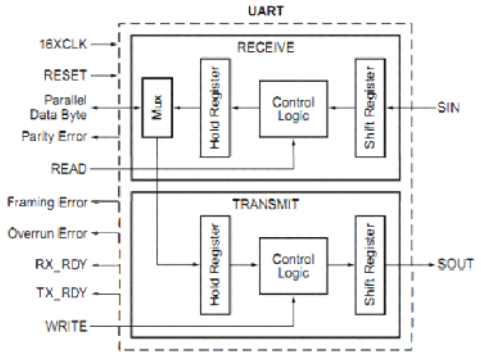

Figure1. UART Block diagram

- reassembles the data byte.
- Wait until the incoming signal becomes ' 0 ' (the start bit) and then start the sampling tick center in parallel and then shifts it out bit-by-bit. The receiver shifts in data bit-by-bit and. When the center reaches 7, the incoming signal reaches the middle position of the start bit. Clear the center and restart.
- When the center reaches 15, we are at the middle of the first data bit. Retrieve it and shift into a register. Restart the center. Repeat the above step N-I times to retrieve the remaining data bits. If optional parity bit is used, repeat this step once more. Repeat this step M more times to obtain.

IV. SPI DESIGN

SPI stands for Serial Peripheral Interface.SPI is a synchronous protocol that allows a master device to initiate communication with a slave device. Data is exchanged between these devices. SPI is implemented by a hardware module called the Synchronous Serial Port or the Master Synchronous Serial Port. It allows serial communication between two or more devices at a high speed and is reasonably easy to implement.

SPI is a Synchronous protocol. Only the master device can control the clock line, SCK. Often a slave select signal will control when a device is accessed. This signal must be used for when more than one slave exists in a system, but can be optional when only one slave exists in the circuit. As a general rule, it should be used. This signal is known as the SS signal and stands for "Slave Select." It indicates to a slave that the master wishes to start an SPI data exchange between that slave device and itself. The signal is most often active low, so a low on this line will indicate the SPI is active, while a high will signal inactivity. It is often used to improve noise immunity of the system. Its function is to reset the SPI slave so that it is ready to receive the next byte.

SPI is a Serial Interface and uses the following signals to serially exchange data with another device:

SS - This signal is known as Slave Select. When it goes low, the slave device will listen for SPI clock and data signals.

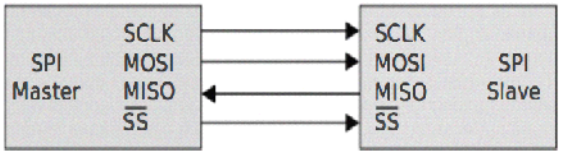

Figure2. SPI Block Diagram

SCK - This is the serial clock signal. It is generated by the master device and controls when data is sent and when it is read.

MOSI - The signal is generated by Master, recipient is the Slave.

MISO -The signals are generated by Slaves, recipient is the master.

SI - Serial Data Input (used to transfer data into the SPI device).

SO - Serial Data Output (used to transfer data out of the SPI device).

CS - Chip Select Input (for enabling device operation).

W- Write Protect Input (used to guard against program/erase instructions).

V. INTERFACING

The UART-to-SPI interface can be used to communicate to SPI slave devices from a PC with a UART port.

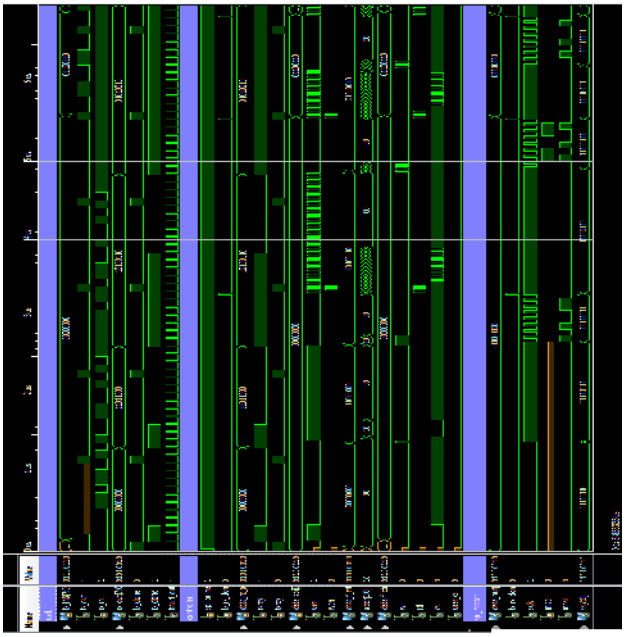

Fig. 3.

SPI is a full duplex, serial bus commonly used in the embedded world because of its simple hardware interface requirements and protocol flexibility. SPI devices are normally smaller in size (low 110 counts) when compared to parallel interface devices. The interfacing diagram is shown below (Fig. 3).

It consists of three blocks, the UART interface, the UART-to-SPI control block, and the SPI master interface. The internal UART-to-SPI control blocks stitches the Core UART and SPI master. The SPI master block generates the control signals to interface to external slave devices. This interface communicates with the slave devices using the serial data out port (MOSI), serial data in port (MISO), output clock (SCLK), and slave select ports (SS_N [7:0]). There are three internal registers in the design: control register, transmit register, and receive register. The control register sets the different control bits, the transmit register sends the TX data to the SPI bus, and the receive register receives the Rx data from the SPI bus. After every reset, data received from the external UART go to the control register. The control bit position are given in table 1.

Table 1: control bit position.

7	6	5		3	2	1	0
SS				CPHA	CLKDIV		

When the UART-to-SPI communicates to any of the slave devices, it enables only the corresponding slave select signal. Only one slave device should be transmitting data during a particular data transfer. Slave devices that are not selected do not interfere with SPI bus activities during that period.

Once a particular slave have selected then it should take the data from master that will transfer to its After the slave interface will take this particular data and send it to master.

At the same time, slave interface is forward the data to the spi master, spi master will take the data from spi slave through miso pin after completion of 8-bits it will set the tx_rx_done pin. On activating the tx_rx_done pin interface will send the RD, addr values and it will take the data from the spi master.

At the same time, slave interface is forward the data to the spi master, spi master will take the data from spi slave through miso pin after completion of 8-bits it will set the tx_rx_done pin.

VI. SIMUATION RESULTS

The Interface of UART - SPI in SOC has been synthesized using the Xilinx 9.2i .

SIMULATION RESULT UART_SPI INTERFACING

VII. CONCLUSION AND FUTURE SCOPE

The Interface of UART - SPI in SOC will come very effective in many applications. The communication in the SOC architecture makes easy as they have been connected with a bus. In future as of more applications will add into the subsystem the routing architecture plays a vital role in the system and it can be implemented in NOC.

REFERENCES

[1]. Design and simulation of UART serial communication module based on VHDL -Fang Yi-Yuan, Chen Xue, IEEE Explore, may 2011.
[2]. IEEE paper “optimal implementation of uart_spi interface in Sohttp://www.intersil.com/content/dam/Intersil/documents/an13/an1340.pdfC”.
[3]. rrin.us –“Hayes ESP 8-port Enhanced Serial Port Manual
[4]. Spi block Guide v03.06, “free scale semiconductor”.
[5]. Serial pheripheral Interface (SPI) flash memory Backgrounder “spansion”.
[6]. UART TO SPI design example of “actel pdf”
[7]. www.nxp.comldocuments/datasheet!SCI6IS752SCI6IS762.pdf.