Effective Communication Protocols for Verification on SoC Using FPGA

K.S.Gowthaman*, L.Jubair Ahmed**,S.Janani***
*M.E Applied electronics, Sri Eshwar college of engineering, Coimbatore
** Assistant Professor, Sri Eshwar college of engineering, Coimbatore
***M.E VLSI Design, Sri Eshwar college of engineering, Coimbatore

ABSTRACT
A System on Chip (SoC) is an integrated circuit that integrates all components of a computer or other electronic system into a single chip. It may contain digital, analog, mixed-signal, and often radio frequency functions all on a single chip substrate. Traditionally, Engineers have employed simulation acceleration, emulation and/or an FPGA prototype to verify and debug both hardware and software for SOC design prior to tape out. So that design effective communication protocols are important for efficient verification on SOC.A serial communication interface based on FPGA (Field Programmable Gate Array) has been designed in this paper, used for data communication with other equipment. It guarantees the realization of the serial communication function under the condition of without any increasing in hardware resources.

Keywords *- SOC- System on Chip, I2C-Inter IC Protocol, UART-Universal Asynchronous Receiver Transmitter*

1. Introduction

FPGA is a general-purpose user programmable device, which has the high integration and generalization of the Gate Array devices, and the programmable flexibility of programmable logic device user. FPGA is composed of the programmable logic unit array, the wiring resources and the programmable I/O unit array. A FPGA includes rich logic gates, registers and I/O resources. A FPGA chip is equivalent to the system which is realized by hundreds and even more standard digital integrated circuits [6].

The structure of the FPGA is flexible. The FPGA's logic unit, programmable interval connection and the I/O unit can be programmed by user to realize any logic functions and to meet various design requirements. The speed of FPGA is quick; the power consumption is low and the universality is strong. FPGA is especially suitable for the design of complicated system. Using FPGA can also realize dynamic configuration, online system reconstruction (according to the need, the function of the circuit is changed in the different time of the system running to make the system have the multiple tasks of spatial correlation or time correlation) and hardware softening, software sclerosis and so on. In view of the function of serial communication, this paper makes use of the above characteristics of FPGA, using logic to extend the function of hardware standardization module, which keep the unity of the hardware interface and meet the user's diverse requirements.

FPGA demonstrates good performance and logic capacity by exploiting parallelism. At present single FPGA platform can play multi-functions, including control, filter and system. FPGA design flow is a three-step process consisting of design entry, implementation, and verification stages. The full design flow is an iterative process of entering, implementing, and verifying the design until it is correct and complete. The key advantage of HDL when used for systems design is that it allows the behavior of the required system to be described (modeled) and verified (simulated) before Synthesis tools translate the design into real hardware (gates and wires). HDL describes hardware behavior. There are main differences between traditional programming languages and HDL. Traditional languages are a sequential process whereas HDL is a parallel process. HDL runs forever whereas traditional programming language will only run if directed.

FPGA consists of Programmable logic unit, the wiring resources and I/O unit which are programmable. FPGA is suitable mainly for complex designs. The design of FPGA is flexible. The logic blocks, routing resources and I/O unit can be programmed by the user to realize any logic function and to meet various requirements.

2. Related Work

Digital data transmission can occur in two basic modes: serial or parallel. Data within a computer system is transmitted via parallel mode on buses with the width of the parallel bus matched to the word size of the computer system. Data between computer systems is usually transmitted in bit serial mode. Consequently, it is necessary to make a parallel-to-serial conversion at a computer interface when sending data from a computer system into a network and a serial-to-parallel conversion at a computer interface when receiving information from a network. The type of transmission mode used may also depend upon distance and required data rate.

Transmission mode is based on following information's

- Amount of data
- Speed of processor
- Hardware or software implementation
- Number of available pins
- Number of sensor(devices)

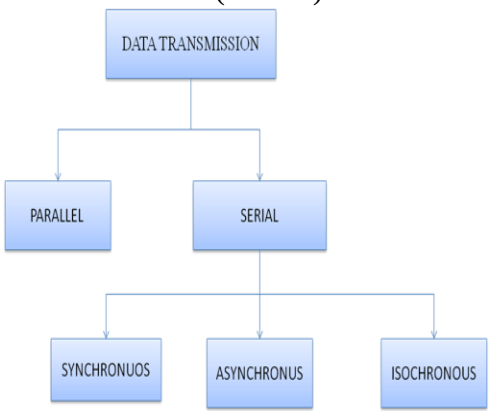

Fig.1.Data Transmission

2.1 Parallel Vs Serial Data Transmission

Many communication systems were generally originally designed to connect two integrated circuits on the same printed circuit board, connected by signal traces on that board (rather than external cables). Integrated circuits are more expensive when they have more pins. To reduce the number of pins in a package, many ICs use a serial bus to transfer data when speed is not important.

Parallel and serial data transmissions are most widely used data transfer techniques. Parallel transfer has been the preferred way for transfer data. But with serial data transmission we can achieve high speed and with some other advantages.

In parallel transmission n bits are transferred simultaneously, hence we have to process each bit separately and lineup them in an order at the receiver. Hence we have to convert parallel to serial form.

Signal skewing is the problem with parallel data transmission.In the parallel communication, n bits leave at a time, but may not be received at the receiver at the same time, some may reach late than others. To overcome this problem, receiving end has to synchronize with the transmitter and must wait until all the bits are received. The greater the skew the greater the delay, if delay is increased that speed.

Another problem associated with parallel transmission is crosstalk. When n wires lie parallel to each, the signal in some particular wire may get attenuated or disturbed due the induction, cross coupling etc

Serial communication is full duplex where as parallel communication is half duplex. This means that, in serial communication we can transmit and receive signal simultaneously, where as in parallel communication we can either transmit or receive the signal.

Serial Cables can be longer than Parallel cables. The serial port transmits a '1' as -3 to -25 volts and a '0' as +3 to +25 volts where as a parallel port transmits a '0' as 0v and a '1' as 5v. Therefore the serial port can have a maximum swing of 50V compared to the parallel port which has a maximum swing of 5 Volts. Therefore cable loss is not going to be as much of a problem for serial cables as they are for parallel.

3. Effective Serial Communication Protocols And Comparison

Embedded system mainly uses serial communication to communicate with peripherals. Therefore serial communication plays vital role in embedded system design. Many serial communication protocols are used like Universal Asynchronous Receiver Transmitter (UART), Control Access Network (CAN), Universal Serial Bus (USB), Serial Peripheral Interface (SPI) and Inter IC Protocol (I2C).Serial communication protocol have characteristic of high speed and low data loss and it simplifies system level design and ensure data transfer. Inter IC protocol satisfy the requirement of embedded system of less number of wires for interconnection. Implementation of inter IC protocol on FPGA gives flexibility to use it according to application.

3.1 I2C

An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it. In an embedded system physical size of device get reduce due to reduction in size of transistor but as number of components or ICs increases on devices, interconnects also increases which creates problem. To overcome this problem Philips Electronics design protocol for communication between different integrated IC called as Inter IC protocol. Many electronics appliances such as Liquid crystal display (LCD), memory, keyboards, cell phones, PCs, TVs use I2C protocol for communication. I2C bus physically consists of two active wires namely serial clock line (SCL) and serial data line (SDA). SCL and SDA both are active high bidirectional and half duplex in nature. Each device which is connected to these wires has unique address. Each device can act as transmitter or receiver depending on its nature.

The I2C bus is a multi-master bus. This means that more than one IC capable of initiating a data transfer on the bus is considered to it. I2C provides low cost and efficient link between ICs. It is synchronous protocol in which master initiate data communication and data get exchanged between master and slave. All communication is control by master using SCL line. All slave connected with SCL line. Slaves are controlled by same SCL line. SCL and SDA line avoids collision of data by using clock stretching.

There are two bidirectional wires SDA and SCL connected to a positive supply voltage via a pull-up resistor as shown in Figure 1. When the bus is free, both lines are HIGH. The output stages of devices connected to the bus must have an open-drain or open-collector in order to perform the wired-AND function. They are pulled-up to the logic „high" level by resistors connected to a single positive supply, usually +3.3 V or +5 V. All the connected devices have open collector driver stages that can transmit data by pulling the bus low and high impedance. [6] Active pull up resistor SDA Master IC IC1 IC2 R1 R2 VCC (3.3 to 5 V) SCL [4]. The I2C is governed by certain frame structure which is byte oriented. Slaves gives start signal to the master which then follows slave address and data direction. Data direction means write or read. The slave addressing can be 7 or 10 bit and data direction means receiving (read) or transmission (write) data. Write operation is also term as transmission of bytes to particular receiver.

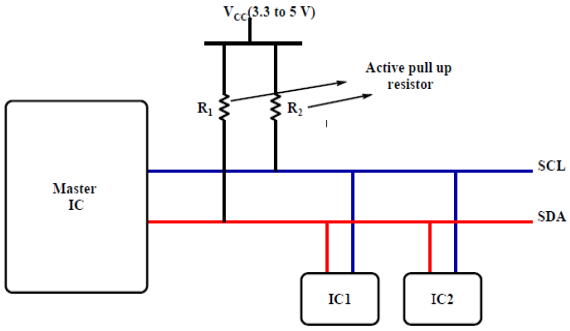

Fig.2. I2C general block diagram

This operation depends on frame structures third bit whether it is read or write bit. The „0" bit means write and „1" bit means read. START condition is when SCL is at logic high and SDA has transition from high to low while STOP means SCL is at high position and SDA transition from low to high. Initially both SCL and SDA are active high due to active pull up resistor.

For start condition SCL should be active high while SDA is low. At each clock cycle frame structure contains address, read or write bit and ACK is received. Reception of ACK means precede otherwise it restarts again. The process of data reading and writing will continue till stop signal arrives. After STOP signal SCL and SDA will be again high.

This protocol uses serial communication protocol which reduces the requirement of wires in embedded system. It is useful in terms of number of pin count and with easier controlling. Implementation of I2C on FPGA is beneficial for data processing, minimal coding and faster operation. Implementation of protocol on FPGA aids in development of dedicated hardware for various applications.

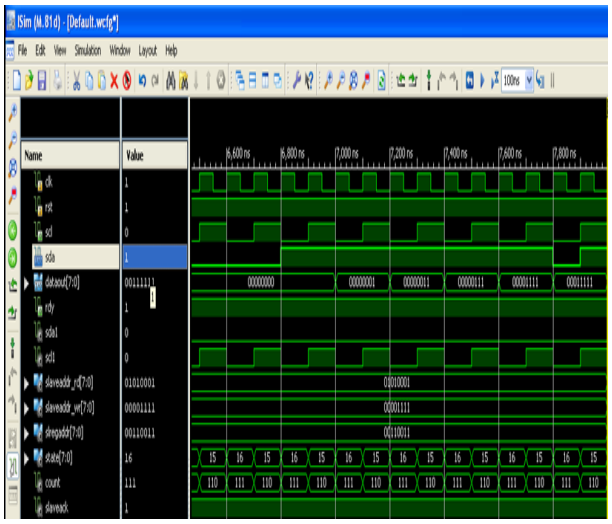

Fig.3.Simulated waveform of the I2C bus protocol.

Features of I2C

- It is multi master serial ended computer bus
- I2C has two wire bi-directional serial bus
- It is a simple and efficient method of data exchange
- I2C protocol have low bandwidth
- It is a Short distance protocol

Advantages of I2C

- Used for security sensitive applications like sensor connections, RFID, biometric devices, etc
- Common communication standards between microcontrollers and sensors
- Each device is recognized by its unique address and can operates as either a transmitter or receiver, depending
- upon the function of the device
- It Provides enhance security system
- Compatible with FPGA

3.2 SPI

The Serial Peripheral Interface (SPI) is a synchronous serial bus developed by Motorola and present on many of their microcontrollers. The SPI bus consists of four signals: master out slave in (MOSI), master in slave out (MISO), serial clock (SCK), and active-low slave select (/SS). As a multi-master/ slave protocol, communications between the master and selected slave use the unidirectional MISO and MOSI lines, to achieve data rates over 1Mbps in full duplex mode.

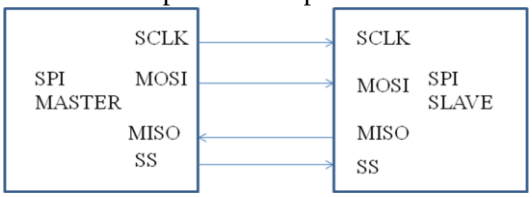

Fig.4. SPI Master/Slave Communication

The data is clocked simultaneously into the slave and master based on SCK pulses the master supplies. The SPI protocol allows for four different clocking types, based on the polarity and phase of the SCK signal. It is important to ensure that these are compatible between master and slave. In addition to the 1Mbps data rate, another advantage to SPI is if only one slave device is used, the /SS line can be pulled low and the /SS signal does not have to be generated by the master. (This capability is, however, dependent on the phase selection of the SCK).

SPI is one of the most commonly used serial protocols for both inter-chip and intra-chip low/medium speed data-stream transfer. SPI IP with one Master One Slave configuration with that of 8-bit data transfer which incorporates all necessary features required by modern ASIC/SoC applications. A disadvantage to SPI is the requirement to have separate /SS lines for each slave. Provided that extra I/O pins are available, or extra board space for a demultiplexer IC, this is not a problem. But for small, low-pin-count microcontrollers, a multi-slave SPI interface might not be a viable solution.

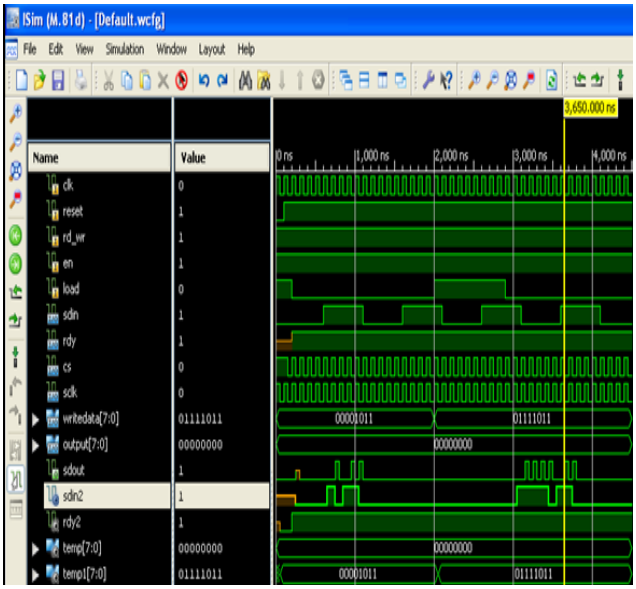

Fig.5. Simulated waveform of the SPI bus protocol

3.3 Comparison between I2C and SPI

Selecting between I2c and SPI, the two main serial communication protocols, requires a good understanding of the advantages and limitations of I2C, SPI, and your application. Each communication protocol will have distinct advantages which will tend to distinguish it as it applies to your application. The key distinctions between I2C and SPI are[3]:

- I2C requires only two wires, while SPI requires three or four
- I2C draws more power than SPI
- I2C supports multiple devices on the same bus without additional select signal lines through in-communication device addressing while SPI requires additional signal lines to manage multiple devices on the same bus
- I2C ensures that data sent is received by the slave device while SPI does not verify that data is received correctly
- I2C can be locked up by one device that fails to release the communication bus
- SPI cannot transmit off the PCB while I2C can, albeit at low data transmission speeds
- I2C is cheaper to implement than the SPI communication protocol
- SPI only supports one master device on the bus while I2C supports multiple master devices
- I2C is less susceptible to noise than SPI
- SPI can only travel short distances and rarely off of the PCB while I2C can transmit data over much greater distances, although at low data rates
- The lack of a formal standard has resulted in several variations of the SPI protocol, variations which have been largely avoided with the I2C protocol

These distinctions between SPI and I2C should make selecting the best communication option for your application easier. Both SPI and I2C are good communication options, but each has a few distinct advantage and preferred applications. Overall, SPI is better for high speed and low power applications while I2C is better for suited to communication with a large number of peripherals and dynamic changing of the master device role among the peripherals on the I2C bus. Both SPI and I2C are robust, stable communication protocols for embedded applications that are well suited for the embedded world.

4. Universal Asynchronous Receiver Transmitter

It is observed that many researchers have designed UART by applying different techniques like algorithms, logical relations. Researchers have undertaken different phenomena with regards to design UART and attempted to find better result. Today in real world the actual applications, usually needed only a few key features of UART [1].

Specific interface chip will cause waste of resources and increased cost. Particularly in the field of electronic design, SOC technology is recently becoming increasingly mature. This situation results in the requirement of realizing the whole system function in a single or a very few chips.

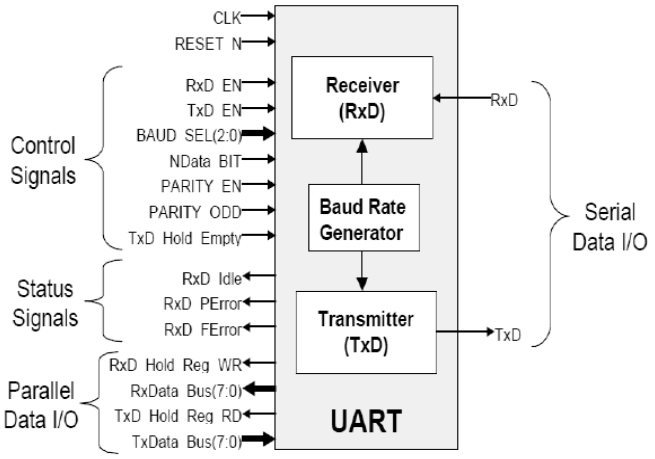

Fig.6. Complete UART blocks

From the survey it is observe that the implementation of UART basically uses the on-chip UART IP hard core because it has high performance but it has poor flexibility and poor transportability, hence it is usually unable to meet the high requirements of the customer. With the rapid development of FPGA soft core plays an increasingly important role in embedded system depending on the high performance, high flexibility, transportability and configuration. A UART is the microchip with programming that controls a computer's interface to its attached serial devices. Specifically, it provides the computer with the RS-232C Data Terminal Equipment (DTE) interface so that it can "talk" to and exchange data with modems.

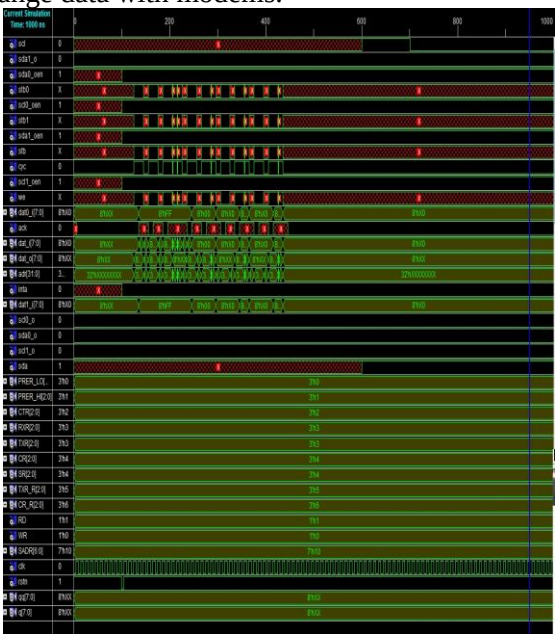

Fig.7. Simulated waveform of the UART protocol

• Converts the bytes it receives from the computer 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 computer 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

• Handles interrupt from the keyboard and mouse (which are serial devices with Special ports)

• May handle other kinds of interrupt and device management that require coordinating the computer's speed of operation with device speeds Serial transmission is commonly used with modems and for non-networked communication between computers, terminals and other devices. The communications links across which computers or parts of computers talk to one another may be either serial or parallel. A parallel link transmits several streams of data 8 (perhaps representing particular bits of a stream of bytes) along multiple channels (wires, printed circuit tracks, optical fibers, etc.); a serial link transmits a single stream of data. At first sight it would seem that a serial link must be inferior to a parallel one, because it can transmit less data on each clock tick. However, it is often the case that serial links can be clocked considerably faster than parallel links, and achieves a higher data rate. A number of factors allow serial to be clocked at a greater rate.

• Clock skew between different channels is not an issue (for un clocked serial links)

• A serial connection requires fewer interconnecting cables (e.g. wires/fibers) and
Hence occupies less space. The extra space allows for better isolation of the Channel from its surroundings

• Crosstalk is less of an issue, because there are fewer conductors in proximity. In many cases, serial is a better option because it is cheaper to implement. Many ICs have Serial interfaces, as opposed to parallel ones, so that they have

In most computer systems, the UART is connected to circuitry that generates signals that comply with the EIA RS232-C specification. There is also a CCITT standard named V.24 that mirrors the specifications included in RS232-C. 2.2 RS232-C Bit Assignments (Marks and Spaces) .In RS232-C, a value of 1 is called a Mark and a value of 0 is called a Space. When a communication line is idle, the line is said to be “Marking”, or transmitting continuous 1 value.

5. Conclusion

Today, at the low end of the Communication Protocols there are mainly Two Protocols: Inter- Integrated circuit (I2C) and the Serial Peripheral Interface (SPI) Protocols. Both the protocols are well suited for communications between Integrated Circuits for communication with ON-Board Peripherals. SPI is one of the most commonly used serial protocols for both inter-chip and intra-chip

low/medium speed data-stream transfer**.** From the simulation results of the all bus protocol, we can compare speed and pin count of all protocols. As parallel bus having highest speed data transmission but alongside it is having more pins. I2C having advantage of slave acknowledgement feature but SPI does not support this feature.

Finally, we may conclude that based on the application requirement bus protocols can be chosen. Even more types bus protocols can also be implementing on FPGA and compare with these bus protocols & even system can be interfaced to PC through USB interface.

References

[1]. Chun-zhi, He; Yin-shui, Xia; Lun-yao, Wang ,"A universal asynchronous receiver transmitter design," Electronics, Communications and Control (ICECC), 2011 International Conference on , vol., no., pp.691-694, 9-11 Sept. 2011.

[2]. S.Thane, S.Somkuamanit, S.Khuntawee. "Implementation of Multi-Protocol, Data Acquisition with High Speed USB interface, Using FPGA", The International Multi Conference of Engineers and Computer Scientists (IMECS)- Vol I, March 17-19, 2010. Hong Kong.

[3]. Frédéric Leens, "An Introduction to I2C and SPI protocols". IEEE Instrumentation & Measurement Magazine February 2009.

[4]. Bollam Eswari, N.Onmagal, K.Preethi, S.G. Sreejeesh, "Implementation of I2C Master Bus Controller on FPGA". International Conference on Communication and Signal Processing, April 3-5, 2013, India.

[5]. Bernhard Linke, "Overview of 1-Wire Technology and Its Use", Jun 19, 2008, Maxim Integrated Products, Inc.

[6]. Volnei A.Pedroni, "Circuit Design with VHDL", MIT Press, England. [6]. Douglas L. Perry, "VHDL: Programming by Example" 4th Edition.