Design of Multichannel UART Controller with AXI4.0 Lite Interface

Parth Shah
VLSI & Embedded System Design
GTU PG School
Ahmedabad, India
pdshah77@gmail.com

Mr. Santosh Jagtap
Wipro
Pune, India
sansjagtap@gmail.com

Abstract—UART (Universal Asynchronous Receiver Transmitter) is basically used for long distance slow speed serial communication. When it comes to latest SOC high speed communication design single channel requires more time and lots of interrupt to process the data. To overcome this disadvantage here high speed multi channel UART controller has been implemented which will be used for avoid synchronization error between and high speed master and different slave devices. AMBA AXI4.0-Lite high speed bus is also interfaced with UART Controller. AXI4.0 is reconfigurable and can be updated to any other high speed bus in future. Whole design is simulated using Verilog HDL in Questasim 10.0b software with the help of verilog testbench.

Keywords- *Asynchronous FIFO; Multi Baud Rate Generator; Controller; AXI4.0-Lite*

*****

I. INTRODUCTION

In this era of SOC design, synchronization between master and multiple slave devices working at different speeds is must. Effective Communication is vital factor for any system performance. Two types of communication can be possible one is serial and another is parallel communication. When one wants to transmit the data for longer distance without distortion serial communication is most widely used because parallel communication has more than one channel working at a time which produces distortion in the long distance communication [9]. Sometimes serial communication protocol is very simple and it cannot meet requirements of complex system with different baud rates working at a single time. Universal Asynchronous Receiver Transmitter (UART) is one of the simplest and a cheapest serial communication which is used for serial communication and modem support [1]. A single UART may not be useful for complex system because it provides pre defined specific baud rate at a single time.

To overcome this disadvantage here in this paper multi channel UART controller with AXI4.0-Lite interface has been proposed. Four UART will work simultaneously in different programmable modes and bridge is formed between two serial devices working at different baud rates. Figure 1 below shows the interconnection at SOC level between AXI and UART using south bridge and APB bus system.

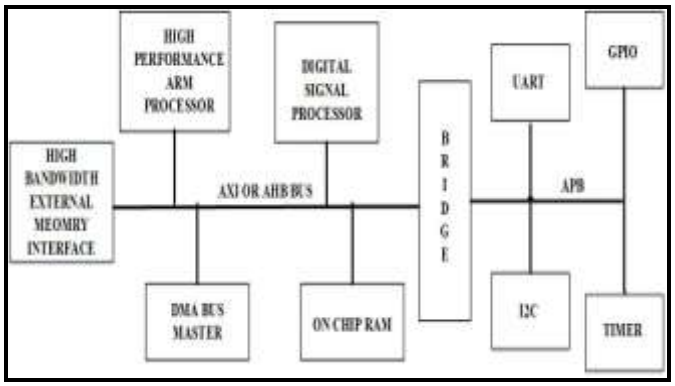

Figure 1. SOC Interconnection of AXI and UART

AXI4.0 is the latest development in field of AMBA on chip protocol [5]. Here direct bridge is tried to design between UART and AXI4.0-Lite protocol [4]. With the help of whole design synchronization error between the Master PC and serial peripheral devices will decrease also interrupt to the microcontroller will be less as on a single interrupt parallel processing can be achieved.

II. BACKGROUND RESEARCH

In this section detail description of all the modules which are used in designing the proposed solution has been mention.

A. UART

The UART is one of the most widely used communication protocol when it comes to serial communication on SoC design. Main advantage of UART is the cost. It is the cheapest communication protocol compare to other serial communication protocols. Primary function of UART is to convert parallel data to serial form and vice versa. UART supports 5 to 8 data bits, 1, 1.5 and 2 stop bits and also extra bit for error correction and detection. UART speed is measure in Bauds per second. Single UART can support speed from 110 to 921600 bauds per second. Single UART consist of separate Transmitter and Receiver for converting parallel data to serial data bits and serial data to parallel data as and when needed [1][2]. Due to complex system multi baud rate support data loss can occur in single UART and bit error can be easily generated. To overcome this Multi channel UART with four UART channels working with Asynchronous FIFO design has been tried to implemented to avoid data loss during the communication.

B. AMBA AXI4.0 Lite

AXI is latest improvement in the field of ARM AMBA on chip communication buses. AXI bus is used for high performance, high clock frequency system design and suitable for high speed sub-modules interconnect. AXI4.0 is available in mainly three types: AXI4.0-Lite, AXI4.0 and AXI4.0 Stream [4][5]. AXI4.0-Lite is used for mainly data transaction purpose.

ISSN: 2321-8169
2617 – 2621

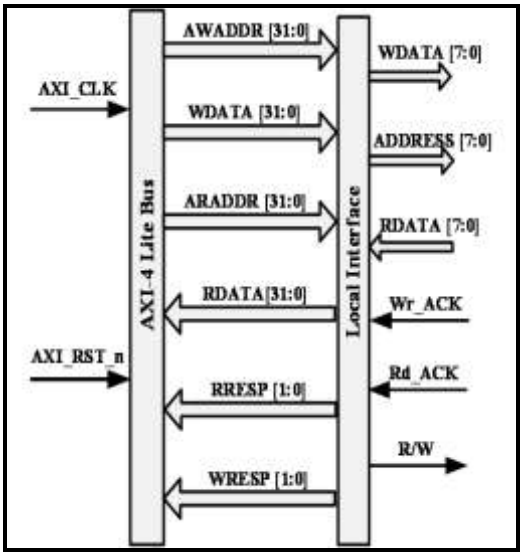

Figure 2. AXI4.0 Lite Bus to Local Bus Design

AXI4.0 supports all the features like Out of Order transaction, Multiple Address Support and Burst Mode support. AXI4.0 Stream is used for mainly Video transfers. AXI4.0-Lite IP Interface provides a point-to-point bidirectional interface between a multi channel UART controllers and AXI Interconnect core. This interface has been optimized for slave operation on the AXI interface. It does not provide support for Master Services.

Fig. 2 shows AXI4.0-Lite bus interfaced with local bus. AXI4.0-Lite supports only basic read and write transaction data and address channels and all other signals are ignored. All the default signals contain two signals named VALID and READY.

C. Asynchronous FIFO

FIFO stands for First In First Out. FIFO is basically used for data storage purpose. When working with two different clock domain Asynchronous FIFO is used to avoid synchronization error and also pass the data from one clock domain to other clock domain without any loss [8]. Detail diagram of Asynchronous design has been shown in Fig. 3. FIFO consists of Dual Block RAM, Read and Write Address pointer, different Read and Write clock frequency and also synchronizers. Write operation occurs on write clock and increments the write pointer by one. Same as read operation is occurs on read clock. This FIFO has four status signal named "Full", "Empty", "Almost Full" and "Almost Empty" for checking status of Asynchronous FIFO. FIFO depth is calculated using (1).

$$D = B - (\alpha/\beta)*B \quad (1)$$

Where D = FIFO Depth, B = Data Burst size,
α = Read Frequency, β = Write Frequency.

Read and Write address value are based on the FIFO depth. One extra flag is used with pointer value for FIFO full and empty condition check. If all bits of read and write pointer are same and read flag is equal to write flag then FIFO is empty as both pointer are on same phase and if read flag and write flag are not same then FIFO is Full as write pointer is faster then read pointer.

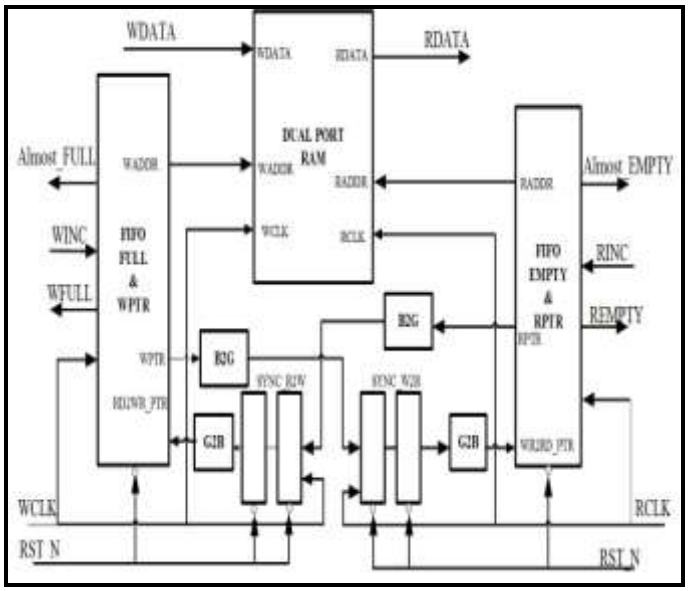

Figure 3. Asynchronous FIFO

Read and write pointers are first converted to gray pointer using binary to gray converter and then it is passed through synchronizer to avoid meta-stability problem. Meta-stability changes output from logic 1 to logic 0 or in between unpredicted state which causes final output to be unpredicted value.

III. PROPOSED MC UART CONTROLLER

Fig. 4 shows the proposed multi channel UART controller top diagram. It consist of mainly four UART blocks interconnected with the help of controller and also one register block is there to control the status of operation. Controller consists of three asynchronous FIFO, Clock divider and multiplexing logic. Controller can work in different programming modes like Normal Mode, Bridge Mode, Hub Mode and Bridge Hub Mode. Multiplexing Logic consist of simple 2x1 multiplexer and de-multiplexers.

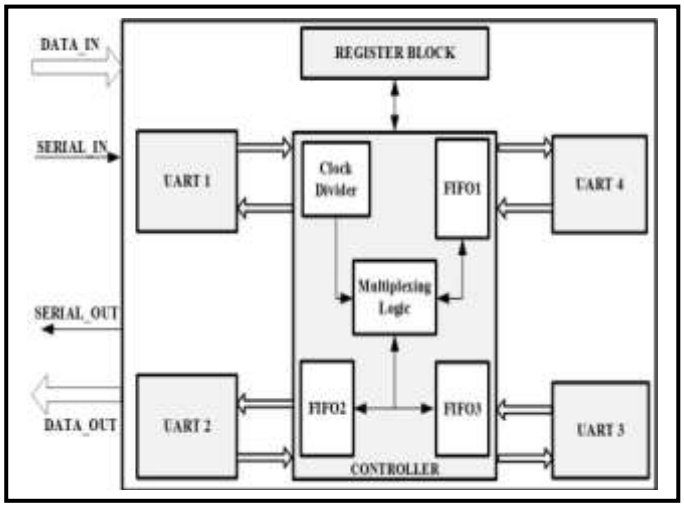

Figure 4. Proposed Design

ISSN: 2321-8169
2617 – 2621

A. Normal Mode

In this mode any two UART will receive serial data and transmit to remaining two UART at same baud rates.

B. Bridge Mode

In this mode any two UART will receive serial data at different baud rate and transmit it at other specified baud rate.

C. Hub Mode

In this mode any one UART will receive serial data at other baud rate and transmit the same data at same baud rates to other three devices.

D. Bridge Hub Mode

In this mode we are implementing bridge in Hub mode means single UART will transmit the data to other three serial devices at three different baud rates.

Clock Divider is used for generating specified clock from the system clock according to baud rate needs. Clock divider value has been calculated using (2).

$$Đ = Ƒ / (16*B) \quad (2)$$

Where Đ = Divisor Value, Ƒ = System Clock Frequency
B = Desired Baud Rate

This divisor values are stored in 16 bit baud register and timers are used according to need of divisor value. Other than controller separate UART module is consist of Transmitter and Receiver logic. According to mode selection Transmitter and Receiver will turn on and off and data will be transferred. 8 bit mode selection register has been used to define which UART will transmit and which UART will receive the data.

Other than serial data bridge is formed between AXI4.0 and UART controller so AXI4.0 transfers parallel data and which goes into controller directly and any two UART can transmit that parallel data inform of serial data at two different baud rates. So with the help of this whole controller one can bridge between serial data transmission working at different baud speeds also it can easily convert parallel data which is coming from AXI4.0 bus to serial data. Register block will consist of all the status and control register information about UART and asynchronous FIFO design.

IV. ALGORITHM

Fig 5 shows software algorithm of the whole project. As shown in fig 5 first mode of operation will be selected than FIFO full and empty condition will be checked and according to full and empty transaction will start and stop.

Whole Project has been implemented using Verilog HDL using Questasim 10.0b. It has been synthesized using Xilinx software for Automative Spartan 6 Family.

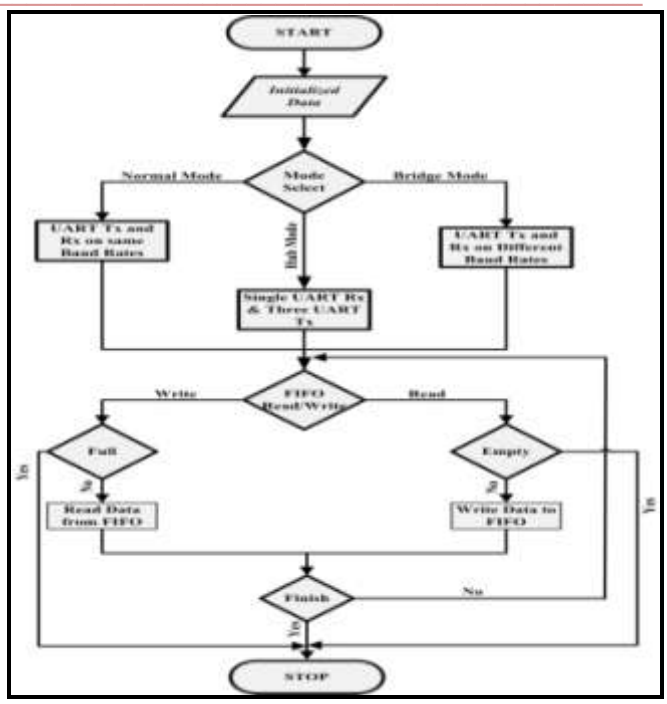

Figure 5 Design Flow Chart

V. RESULTS

Simulations results are shown for the entire MC-UART controller including single component simulation results. Fig 6 shows the simulation result for single asynchronous FIFO design. This asynchronous FIFO has been used twice to get synchronization in different modes. There is no data loss during the FIFO data out when output is taken at different clock frequency.

Fig 7 shows the simulation result of the multi clock generator from system clock. Simple clock divider has been verified in which divisor value needs to set to generate divisor clock from system clock. Divisor value has been calculated using (2).

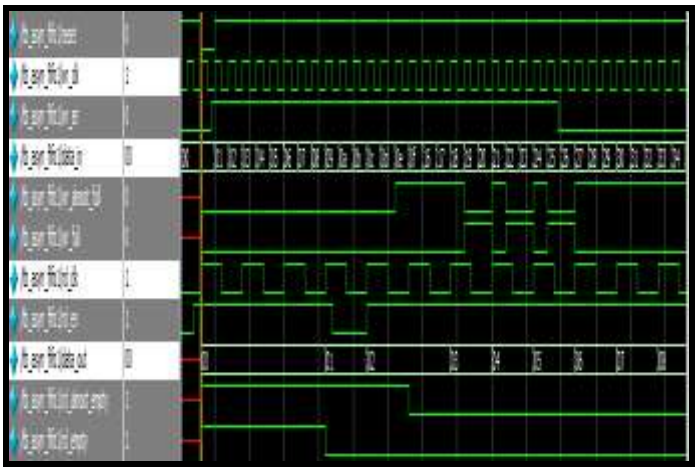

Figure 2 Simulation Result of Asynchronous FIFO

 ISSN: 2321-8169
2617 – 2621

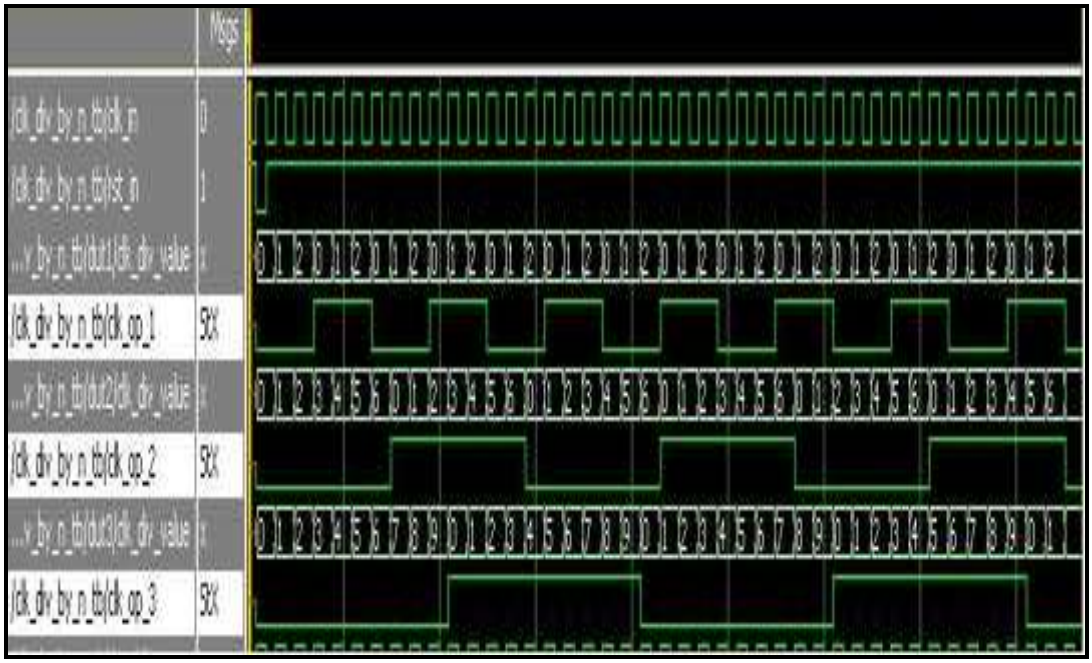

Figure 7 Simulation Result of Clock Divider

Fig 8 shows simulation result for separate UART transmitter. Fig 9 shows simulation result for single UART receiver. Separate Transmitter and receiver are verified using verilog testbench and four instances have been used for top design. Figure 10 shows simulation result for UART controller Normal mode here parallel data has been received from AXI4.0-Lite bus and this data has been transmitted serially with the help of any two UART at same baud rates. Also other than serial data this parallel data can also be available at output of controller. So from all this simulation result we can conclude that controller is working properly and there is no data loss when communicating on different clock rates.

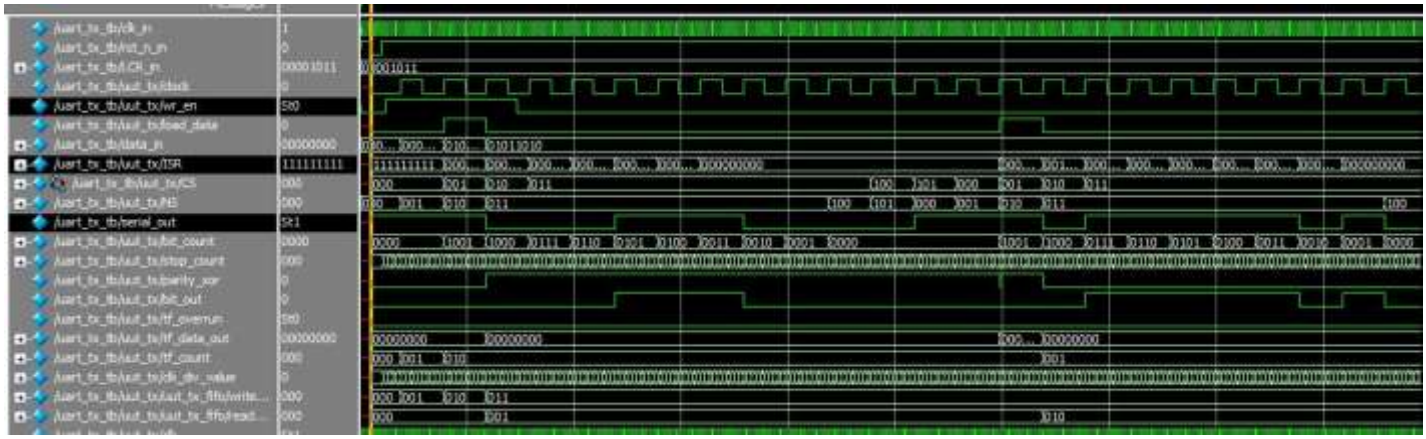

Figure 8 Simulation Result of Single UART Transmitter

Figure 9 Simulation Result of Single UART Receiver

ISSN: 2321-8169
2617 – 2621

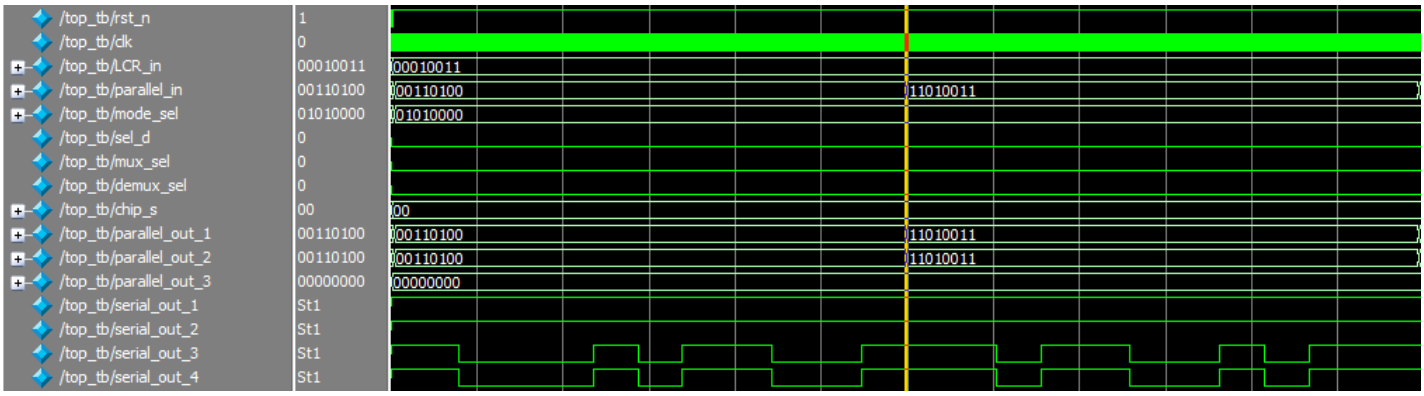

Figure 10 Simulation Result of Normal Mode

VI. CONCLUSION

In this paper with the help of Asynchronous FIFO design Multi-channel UART controller with AXI4.0 lite interface is simulated using Questa Sim 10.0b. This controller will be useful for avoiding synchronization error between the master PC and slow serial peripheral devices connected with the high speed microcontroller. Controller is reconfigurable and scalable. With different modes of operation parallel processing can be easily achieve and interrupt can be reduce to main microcontroller or master PC. As more task will occur in single interrupt service routing throughput of the overall system increases.

REFERENCES

[1] Agrawal, R.K. and Mishra, V.R., "The Design of High Speed UART", IEEE conference on Information & Communication Technologies(ICT), April 2013, pp. 388-390

[2] Mahat, N.F., "Design of a 9-bit UART module based on Verilog HDL", IEEE International conference on Semiconductor Electronics(ICSE), September 2012, pp. 570-573

[3] Shouqian Yu, Lili Yi, Weihai Chen and Zhaojin Wen "Implementation of Multi-channel UART controller Based on FIFO Technique and FPGA", IEEE conference on Industrial Electronics and Application (IEA), May 2007, pp. 2633-2638

[4] Srinivasa Rao.B, Srinivas.B, Srinivas.K "High Speed System-On-Chip communication using AXI4 Protocol", International Journal of Innovative Research & Studies (IJIRS), October 2013, pp. 201-210

[5] Math, S.S., Manjula, R.B., Manvi, S.S. and Kaunds, P. "Data Transactions on system-on-chip bus using AXI4 protocol", IEEE conference on Recent Advancements in Electrical, Electronics and Control Engineering (RAEeCE), December 2011, pp. 423-427

[6] Samir Palnitkar, Verilog HDL- A Guide to Digital Design and Synthesis, SunSoft Press, 1996, pp. 280-305

[7] Deepchand Jaiswal, Dr. Rita Jain, M. Zahid Alam "Design of Multi-channel UART Controller based on FIFO and FPGA", International Journal of Advanced Research in Computer Engineering & Technology, June 2012, pp. 419-426

[8] Xinrui Zhang, Jian Wang, Yuan Wang, Dan Chen, Jinmei Lai "BRAM- based Asynchronous FIFO in FPGA with Optimized Cycle Latency ", Solid-State and Integrated Circuit Technology (ICSICT), November 2012, pp. 1-3

[9] Serial Protocol Comparison,[Online] Available at http://www.embedded.com/design/connectivity/4023975/Serial-Protocols-Compared