International Journal of Mechanical and Industrial Engineering

Volume 1 | Issue 3 Article 6

January 2012

Multi-UART Controller with Programmable Modes of Operation

Rohit Athavale
PES Institute of Technology, Bangalore,100 ft Ring Road, Bangalore, Karnataka, India,
rohit.athavale89@gmail.com

Tejas N
PES Institute of Technology, Bangalore,100 ft Ring Road, Bangalore, Karnataka, India,
tejas5130@gmail.com

HV Jayashree
PES Institute of Technology, Bangalore,100 ft Ring Road, Bangalore, Karnataka, India,
jayashreehv@pes.edu

Multi-UART Controller with Programmable Modes of Operation

Rohit Athavale, Tejas N & HV Jayashree

PES Institute of Technology, Bangalore,100 ft Ring Road, Bangalore, Karnataka, India
E-mail : rohit.athavale89@gmail.com, tejas5130@gmail.com, jayashreehv@pes.edu

Abstract **-** A Multi-UART controller is presented in this paper which will serve the purpose of communication between different systems or sub-systems of large system. The different systems or sub-systems of a large system will be able to operate at different baud rates and will use the UART (Universal Asynchronous Receiver Transmitter) to perform communication. This controller has different modes of operation—Normal, Bridge and Hub modes. The controller uses an asynchronous FIFO (First In First Out) block for communication purpose. This controller will reduce the synchronization errors between different systems or sub-systems which are communicating with each other. Since communication between sub-systems or different systems is handled by this controller it will not burden the master processor or main system with communication related operations. This paper improves over [2] as the modes of operation being incorporated in the design itself and are completely reconfigurable.

Key words **-** *Asynchronous FIFO, UART, programmable modes.*

I. INTRODUCTION

In most communication or control systems there are highly evolved and specialised computer architectures that take care of inter-system communication or communication between sub-systems of a large system. These systems do not have any generalized architecture, in fact they have architecture which is very specific to its requirements and is customized to meet the demands of that particular system.

However, communication of data between sub-systems of a large system will need synchronization between the sub-systems themselves. These sub-systems may operate at different baud rates. Also in the scenarios where communication of data is required between distant but independent systems, there is great need for synchronization to improve the overall efficiency of the systems.

In this paper we propose a Multi-UART controller that will use a serial communication circuit UART. Universal Asynchronous Receiver Transmitter (UART) circuits are popular and widely used in several systems. The advantage of serial communication being used is that the distortion of a signal will be lesser. Longer distances between systems or sub-systems may be tolerated [1].

The Multi-UART controller being discussed here will reduce the duties of inter-system communication of the master controller of each independent system. In case of communication between sub-systems of a large system, this Multi-UART controller will minimize operational overheads of the master controller while performing inter sub-system communication.

The Multi-UART controller will not only allow communication at different baud rates [B], but will allow the communication in three pre-set modes. These modes are the Normal mode, Hub Mode and the Bridge mode.

II. TOP LEVEL ARCHITECTURE

The architecture consists of a controller which manages operations of the four UARTs. At a given point of time,any UART can either receive data or transmit data.Which means that if UART1 is receiving data then SIN1(Serial In 1) is activated but SOUT1(Serial Out 1) is deactivated.

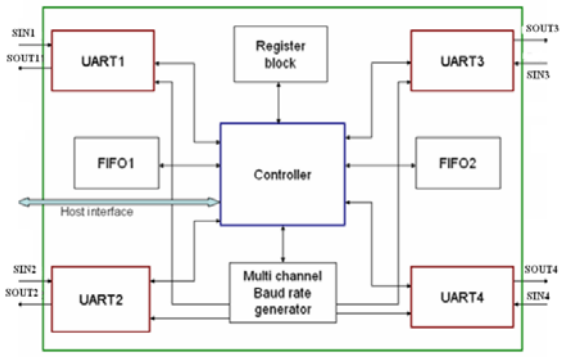

Fig. 1 : Top Level Architecture

There are two FIFO blocks being used in this architecture.The FIFO block is a 64x8 in size.FIFOs are often used to safely pass data from one clock domain to another[2].A linear FIFO with read and write pointers is designed.Special emphasis is laid on read pointer empty and write pointer full conditions as per [4].

The register block consists of different kinds of registers which stores differrent data.The various registers are Mode register, Status register, Configuration registers, Clock Division Registers, Transmitter registers, and Receiver register

The Multi-channel Baud rate generator interacts with all four UARTs and generates independent clock signals for each UART.The master clock frequency is used to derive the individual UART clock frequencies.Each UART has its own clock_div_values register which holds a value.These set of registers are in the register block. The master clock frequency is divided by the value in this register to obtain the indivdual UART clock frequency.The clock_div_values registers of each UART can be reprogrammed to obtain different baud rates for each individual UART.

III. DESIGN OF CONTROLLER

The controller is a circuit consisting of sequential and combinational circuits.The controller acts as an interface between the four UARTs,the FIFO blocks,the baud generator block and the status registers.

The controller receives data from any of the UART at a given baud rate which is synchronized by the baud rate generator block.If data is received from UART1 or UART2 then that data is written into FIFO1. Similarly when we are operating UART3 or UART4 then FIFO2 is utilised.At a given point of time both FIFO1 and FIFO2 can be used simulataneously.

When started the controller checks the status registers.Most of the status registers are set to zero and assume appopriate values once the inititaliasation is complete.The controller obtains various values from the status register block,depending on which the baud generator block,FIFO1 and FIFO2 operate.

The controller receives or transmits data in terms of bytes.Which means that if data is received serially from UART1,then the processed data is converted to a byte of the data and then is sent to the controller.Similarly if wished to transmit data the controller writes a byte of data into the buffer register of the UART.The transmit algorithm within each UART is capable of transmitting that byte of data serially to its destination. For example if we serially received data from UART1 then a byte of data will be sent as input to the controller through Ext_out1.If we wished to send data serially through UART4,then a byte of data must be sent to UART4 through Ext_Data4.

The master clock(Clk) is the input provided to the controller.The individual operating frequencies of each UART is derived from this master clock.The controller has the capability to enable or disable each of the individual UART clocks.

There are two FIFO blocks as mentioned above.Each of the FIFO block can be written into in terms of bytes.Also the controller can read bytes of data from each FIFO block.Data in terms of bytes can be written or read into each FIFO block. independantly.However at given point of time we cannot write and read into the same memory location of any of the FIFO blocks.FIFO11_out and FIFO22_out can read data from FIFO1 and FIFO2 blocks respectively.Similarly FIFO11_Data and FIFO22_Data can write into FIFO1 and FIFO2 blocks respectively.

The controller has two more inputs which we provide.They are the reset(rst_n) and chip select(CS) inputs.The reset input resets all the entire controller and its operations.Only when the CS signal is high the controller will operate normally.

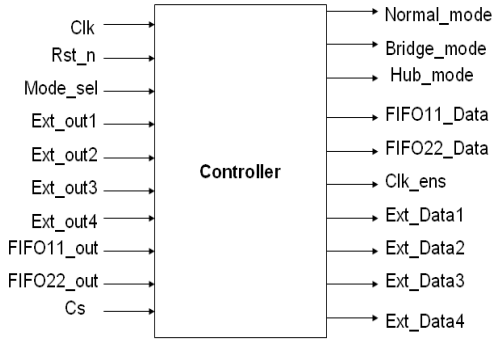

Fig. 2 : Structure of Controller

If CS is low the controller will enter the IDLE state. The individual clocks of UART1, UART2, UART3 and UART4 are disabled in the IDLE state. If CS is high then the controller will enter the RUN state. The Multi-UART controller can operate normally in the RUN state. The individual clocks of each of the UART are enabled in the RUN state.

Now we discuss the different modes of operation of this controller. In the RUN state the controller can operate in three different modes. The three different modes are—the normal mode, the hub mode and the bridge mode. Mode_sel is an input signal which determines the mode in which the controller is working.

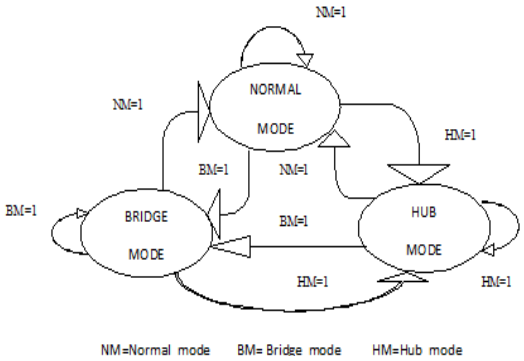

Fig. 3 : State Diagram of Controller in RUN State .

If the input to the Mode_sel is "00" then the Normal Mode is activated and the Normal_mode which is an output signal is made high. Once the controller is in the normal mode, the controller sets the Normal Mode register in the Status Register block. All other mode registers are reset. In the normal mode UART1 will receive data and UART3 will transmit that data. Similarly in the normal mode UART2 will receive data and UART4 will transmit that data.UART1 and UART2 can independently receive data at different baud rates.UART3 and UART4 can independently transmit data at different baud rates.

If the input to the Mode_sel is "11"in binary or "03" in hexadecimal then the Hub Mode is activated and the output Hub_mode is made high while Normal_mode and Bridge_mode outputs go low. Once the controller is in the Hub Mode the controller sets the Hub Mode register in the Status block. The Normal Mode and Bridge Mode registers are reset. In the Hub Mode UART1 will receive data while UART2, UART3 and UART4 transmit that data.

If the input to the Mode_sel is "111" in binary or "07" in hexadecimal then the Bridge Mode is activated and the output Bridge_mode is made high while Normal_mode and Hub_mode outputs go low. In the Bridge Mode, the controller sets the Bridge Mode register in the Status Register block. The two other mode registers are reset. In the Bridge Mode, UART1 will receive data which is transmitted by UART2 at different baud rates. Simultaneously in the Bridge Mode UART3 will receive data and UART4 will transmit that data at different baud rates.

IV. SOFTWARE STRUCTURE

The various parts of the Multi-UART controller is shown in the Fig1.The design and analysis of FIFO is as per [2][3]. In [2] circular FIFO was implemented but we have opted for a simple linear FIFO which is easily incorporated in the top level architecture proposed in this paper.

The software structure involved in the design of the following blocks- UART block, FIFO blocks, Status Register Block, Baud Generator block. The controller which interacts with all of the above was designed and its design was discussed earlier in III.

A structural design approach was followed. Some components like UART and FIFO blocks are used more than once. A single UART was designed and verified. Then UART component was instantiated four times to obtain four independent UARTs. Similarly once FIFO block was designed and verified, it was instantiated twice to obtain FIFO1 and FIFO2.Codes in Verilog HDL were used to design the architecture of the Multi-UART controller.

V. SIMULATION AND VERIFICATION

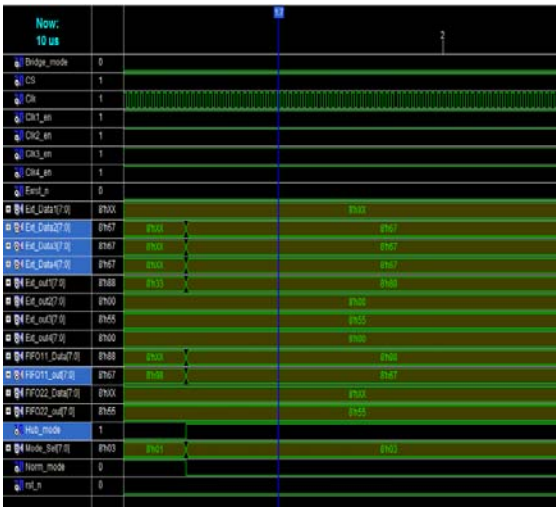

Fig. 4 : Controller Waveform in Hub Mode

Xilinx ISE 9.1i was used to compile the codes in Verilog HDL[5], check syntax, generate RTL and analyse synthesis reports. The ISE Xilinx simulator was used to study the waveforms.

Test benches were written to verify each component like UART[6], Status Register block, FIFO block, Baud Generator block .After component level verification was complete, verification for the entire Multi-UART controller is carried out.

In Fig 4, the controller enters the Hub Mode. Signals which are mentioned in our discussion are blue in Fig 3.View the waveform from when the Hub_mode signal goes high. The FIFO11_out [7:0] signal takes value of "67" in hexadecimal. The Hub Mode ensures that data from UART1 enters the FIFO1.That data is to

be transmitted by UART2, UART3 and UART4 at different baud rates. It is seen that Ext_Data2 [7:0], Ext_Data3 [7:0] and Ext_Data4 [7:0] all take the value of "67" in hexadecimal. Therefore the operation of the Controller in the Hub Mode is verified.

Fig 5 shows the complete Multi-UART controller and its operation. It is verified that how the inputs (SIN1, SIN2, SIN3 & SIN4) and outputs (SOUT1, SOUT2, SOUT3 & SOUT4) operate at different baud rates.

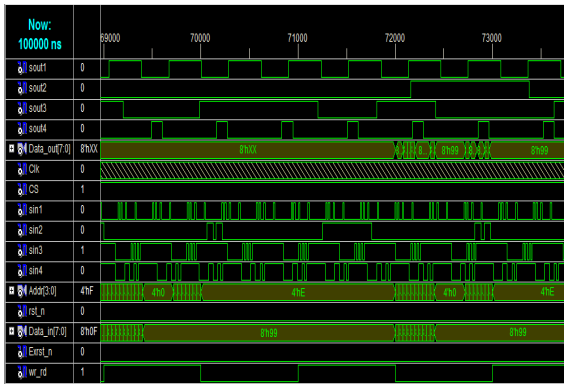

Fig. 5 : Multi-UART Waveform

VI. CONCLUSION

This paper demonstrates a method to design a Multi-UART controller. The multi-UART controller involves the design of asynchronous FIFO for its operation. The controller operates at different baud rates and reduces synchronization error between two systems or sub-systems of a large system.

The controller is designed to operate in three modes—the Normal Mode, the Hub Mode and the Bridge Mode. These modes are reprogrammable to meet different system requirements. Hence it will not burden the Master Controller of the system with communication overheads and procedures.

REFERENCES

[1] L. K. Hu and Q.CH. Wang, "UART-based Reliable Communication and performance Analysis", Computer Engineering, Vol 32 No. 10, May 2006, pp15-21

[2] Shouqian Yu, Lili Yi, Weihai Chen and Zhaojin Wen,"Implementation of a Multi-channel UART Controller Based on FIFO Technique and FPGA", Second IEEE Conference on Industrial Electronics and Applications, 2007.

[3] Vijay A. Nebhrajan, "Asynchronous FIFO Architectures", www.eebyte.com

[4] C. E. Cummings, "Simulation and Synthesis Techniques for Asynchronous FIFO Design with Asynchronous Pointer Comparisons", SNUG San Jose 2002

[5] Samir Palnitkar,"A guide to Digital Design and Synthesis", SunSoft Press, 1996.

[6] Application Notes: UARTs in Xilinx CPLDs (XAPP341 v1.2), Oct 1, 2002.

◈◈◈