Abstract
Universal Asynchronous Receiver and Transmitter (UART) is a circuit that sends parallel data through serial line. It provides a way of serial communication, between two devices but whilst there is a need to provide communication between two devices that are operating at different baud rates, it is intricate to provide communication with an UART. In order to meet the requirements of an UART with multiple channels, this paper prototype a multi-channel UART controller designed with an asynchronous FIFO circuit block. This block can receive data with in a device with UART block of certain baud rate and transmit the data to the sub-equipment with UART block at a baud rate, which may be same or different with that of the receiving devices baud rate. In addition, this reduces the time delay between the sub controllers. Multi-channel UART controller is designed based on XILINX SPATARN-3E FPGA that provide low cost, high performance logic solutions for applications having complex control systems and meet their communication demands quickly and efficiently. This controller is reconfigurable and scalable and it is used to reduce the synchronization error between the subsystems with in a system.

Keywords
MC-UART Controller Spatern, 3E FPGA, BER, DSP, FIFO.

I. Introduction
In many of the devices with in a system, UART is being used. It is a kind of serial communication circuit. It is a feature of the controller useful for communicating serial data to any other device, (for example the communication between the microcontrollers to the PC). The device changes incoming parallel information to serial data and sends the data through the communication line. In some complex control systems or in controllers and processors such as Digital Signal Processors (DSP) the communication demands would affect the performance of the entire system. It is difficult to attain the desired result for various factors affecting the systems in terms of communication. Communications between the master controller and slaver controllers are implemented by serial or parallel port. Parallel communication needs a lot of multi-bit address bus and data bus and it is only convenient for short distance transmission. Serial communication is another way of communication used extensively because of its simple structure and long transmission distance. Nevertheless, sometimes a common serial port could not meet requirements of complex systems with different Baud Rates equipments. Even with some special Baud Rate equipments, it is not possible to meet the desired requirement. The communication parameters such as Baud rate, which is the measure of transmission speed in asynchronous communication, Bit Error Rate (BER), and the synchronization between sub-systems, also engender great effect. Therefore, the device performance will be affected by these parameters. In addition, to make good use of control algorithms with in the complex systems and to improve their precision, it is desirable to look for these communication demands with in that system. In some systems, the communication between the master controller and slaver controller will be through either serial port or parallel port of the device. The communication through parallel port requires a multi-bit address bus and data bus and is only convenient for short distance transmission and the communication through the serial port provides long distance transmission and is simple in its structure. In some systems, a common serial port is used to provide communication between the devices with in which if the devices that have to transfer and receive the data are operating at different Baud Rates, it is difficult to meet the system requirements with a common serial port [2].

A simple example for this is number of devices interface to a common PC and each device be supposed to act as a sub-equipment/slave controller while PC acts as a master controller and each are operating at different Baud Rates. It is no possible to implement this multi-baud rate communication system without a special baud rate converter. Fig. 1, illustrates the example mentioned above.

![Fig. 1: Interfacing of a Master with Slave Controllers at Different Baud Rates](image-url)

Consider an example of a 6-DOF robot, in which there are 6 sub-controllers which are all the same structure to be designed. The PC is used to implement the control algorithm of the robot and send control parameters to sub-controllers and sub-controllers are used to collect feedback signals and send them to the PC. The PC and sub-controllers communicate with each other on a RS485 BUS NET. Each sub-controller has a unique address number and the PC uses this number to identify each sub-controller. When the PC wants to send data to node 6, it has to access front 5 nodes, this engenders time delay and makes performance of the robot’s each DOF not synchronization. So it reduces the control algorithm’s precision and brings difficulties in researching of the control algorithm [2].

In order to solve these difficulties, a Multi-channel UART Controller is prototyped on Xilinx Spatarn-3E FPGA which can receive and transmit data between devices with UART blocks at different Baud Rates also it reduces the time taken for communication or the
time delay between sub-controllers and provides synchronization between the transmitter and receiver.

II. The UART

The Universal Asynchronous Receiver/Transmitter (UART) controller is the key component of the serial communications subsystem of a computer [5]. The UART takes bytes of data and transmits the individual bits in a sequential fashion. At the destination, another UART re-assembles the bits into complete bytes. Serial transmission is commonly used with modems and for non-networked communication between computers, terminals and other devices. There are two primary forms of serial transmission: Synchronous and Asynchronous. They are:

1. UART Universal Asynchronous Receiver/Transmitter
2. USART Universal Synchronous-Asynchronous Receiver/Transmitter

In addition to converting data from parallel to serial (transmission) and from serial to parallel (reception), a UART will usually provide supplementary circuits for signals that can be used to indicate the state of the transmission media. It is used to regulate the flow of data in the event that the remote device is not prepared to accept more data. For example, when the device connected to the UART is a modem, the modem may report the presence of a carrier on the phone line while the computer may be able to instruct the modem to reset itself or to not take calls by raising or lowering one more of these extra signals. The function of each of these additional signals is defined in the EIA RS232-C standard.

Fig. 2: Block Diagram of an UART Controller

III. Multi Channel UART

The Multi-channel UART mainly

- Can be ganged to create multiple UARTs in a single design (Example: multiple instantiations to create multiple UARTs within a design).
- Cost saving by way of multiple UART ASIC and PCB real estate & maintenance cost.
- Customization of number of UART based on customer requirement

Benefits:

A. FIFO Memory Management

As showing in fig. below, when FIFO is full it cannot write any more byte into the FIFO. At this time, the Status Detector will set CS high to indicate that the FIFO is full and stop writing to the FIFO. When FIFO is empty, it is not possible to read from it any more. Then the Status Detector will set Empty high to indicate the status of FIFO and stop reading from it. When FIFO is not full or empty it will be written or read data according the control order. After finishing all write or read operation it will stop until next access is coming.

Fig. 3: Asynchronous FIFO

IV. SPARTAN-3E FPGA

Xilinx has two main FPGA families: the high-performance Virtex series and the high-volume Spartan series. The Virtex series of FPGAs have integrated features such as wired and wireless infrastructure equipment, advanced medical equipment, test and measurement, and defense systems. The Spartan series targets applications with a low-power footprint, extreme cost sensitivity and high-volume such as displays, set-top boxes, wireless routers and other applications. The Spartan-6 family is built on a 45-nanometer (nm), 9-metal layer, and dual-oxide process technology. It is a low-cost solution for automotive, wireless communications, flat-panel display and video surveillance applications. The Spartan-3A consumes more than 70-90 percent less power in suspend mode and 40-50 percent less for static power compared to standard devices.
In addition, the integration of dedicated DSP circuitry in the Spartan series has inherent power advantages of approximately 25 percent over competing low-power FPGAs. Of these two at present, the Spartan-3e of Spartan series is used to implement the Multi-Channel UART Controller.

The Spartan-3E family of Field-Programmable Gate Arrays (FPGAs) is specifically designed to meet the needs of high volume, cost-sensitive consumer electronic applications. The Spartan-3E family builds on the success of the earlier Spartan-3 family by increasing the amount of logic per I/O, significantly reducing the cost per logic cell [1]. New features improve system performance and reduce the cost of configuration.

Features of Spartan-3E
1. Very low cost, high-performance logic solution for high-volume, consumer-oriented applications
2. Proven advanced 90-nanometer process technology, Multi-voltage, multi-standard Select IO™ interface pins
3. Enhanced Double Data Rate (DDR) support, Abundant, flexible logic resources
4. Efficient wide multiplexers, wide logic, Fast look-ahead carry logic
5. Eight global clocks plus eight additional clocks per each half of device, plus abundant low-skew routing Configuration interface to industry-standard PROMs
6. Fully compliant 32-/64-bit 33 MHz PCI support (66MHz in some devices)

V. Algorithm
1. Start.
2. Initialize.
3. Check for the mode.
4. Depending on type of mode, it will start working.
5. Master and slave equipments are set at different baud rates.
6. The controller can be reconfigurable using different registers.
7. Implementing UART using FIFO’s to reduce the synchronization error between subsystems in a system with several subsystems.
8. FIFO checks for the read/write condition.
9. If it is write it a check for FIFO full, if is not full, it will start writing.
10. If it is read, it checks for FIFO empty if it is not empty, it will start reading.
11. Depending on the enable values it will either transmit or receive.
12. Stop.

A. Applications
• Multi-channel serial IO boards, Industrial controllers
• Multi-port RS-232/RS-422/RS-485 cards, Factory automation and process control
• Supervisory/maintenance terminals

VI. Results
Functional and timing simulation has been performed on the Multi-Channel UART using a VHDL Test Bench with test cases. Read and Write cycles have a recovery time due to time slicing latency. Read or write accesses during the recovery time are not permitted. The recovery time is relaxed during back-to-back write cycles to the txdf. After initiating a reset to the core, allow a minimum of (8 * NC+3)*tCY (the period of the system clock) after the trailing edge of reset prior to using the device. This delay is needed for the synchronous clearing of state in each of the channels.

The controller sends data at the same time but at different Baud Rate. When sub-controllers are required to receive data at the same Baud Rate, the controller can set all channels at the same Baud Rate to transmit data. There are no time differences between sub-controller when the controller transmits data to sub-controllers at the same time.

In fact, there are hardware delays in FPGAs and these delays may cause sub controllers cannot receiver data from the controller at the same time precisely. So using this controller can greatly improve synchronization of sub-controllers. The fig. 4 and fig. 5 shows the receiving and transmitting sequences with same and different baud rates.

VII. Conclusion
In this paper, the brief description of a UART is shown, in traditional Universal Asynchronous Receiver Transmitter (UART) controller, the data transmission is inefficient (for the transmission and reception of the data between the devices with different baud rates) and the data bus utilization ratio is low. In this paper the brief description about the Spartan-3e FPGA by Xilinx, regarding its features and small introduction to the Multi-Channel UART Controller about its features and applications are mentioned. In this Multi-Channel UART Controller mainly by making use of time slicing a UART core the area efficiency is seen i.e. it provides effective utilization of the area. Multi-channel UART controller is designed based on XILINX SPARTAN-3E FPGA which provide low cost, high performance logic solutions for applications having complex control systems and also meet their communication demands quickly and efficiently. This controller is reconfigurable and scalable and it also can be used to reduce the synchronization error between the subsystems with in a system and will also reduce the time delays between the sub
controllers of a complex system and improve the synchronization of each sub controller.

VIII. Future Scope
A Multi-Channel UART Controller implementation on an FPGA is cost effective. Due to the reprogrammability feature of FPGA, it is possible to rewrite or modify the code for a Multi-Channel UART. If the complete usage of chip area is of concern then it is better to go for an ASIC where the required amount of chip area is used and it is mainly used, where there is no need to rewrite the code for a specific application. Hence depending upon the application requirements i.e. where there is chip utilization and no need of reprogram-ability it is better to go for ASIC devices.

References