International Journal of Research in Engineering, Science and Management
Volume 5, Issue 3, March 2022
https://www.ijresm.com | ISSN (Online): 2581-5792

Code-Level Implementation of High Speed Synchronized Data Transmission Technique for Faster Data Transmission

C. S. Jyothilakshmi[1*], Sanjay P. Nambiar[2]
[1]*Department of Electronics and Communication Engineering, Sree Narayana Guru College of Engineering and Technology, Korom, India*
[2]*Technical Engineer, IKK Group of Companies, India*

***Abstract*: It is observed that in the current times the Embedded systems have suddenly achieved more importance and complexity. A large variety of systems are now being designed using Field Programmable Gate Array (FPGA). This happens because of its salient features like size, flexibility, and easy availability of resources. In this paper the design and implementation of a high-speed data transmission protocol with the help of POWER-PC is being implemented. This serial communication protocol is implemented in synchronized mode taking all the advantages of asynchronous as well as synchronous transmissions. The paper implements both synchronous as well as asynchronous modes of data transfers.**

***Keywords*: USART, UART, Synchronized, Asynchronized.**

1. Introduction

Universal Synchronous Asynchronous Receiver Transmitter (USART) is one foremost and best type of the serial communication. During the transmission of data in synchronous communication a common clock signal is used to transmit each data pulse [1]. UART mainly takes a role in computers and computers handles only parallel data [2]. In the beginning end, USRT receives parallel data from the block and turns it into serial data at the receiving end, and then a different USRT takes this converted data and changes it back to parallel form which the computers can easily understand.

USRT rises in popularity due to another important characteristic. That is, it has a high-speed data transmission rate which is configurable. The required configuration can be obtained with the help of registers which can be specialized using another set of registers called control registers. To implement the design both synchronous and asynchronous modes of data transmission can be used.

In this paper the simulation of USART code for implementation of USART-to-USART communication application on Field Programmable Gate Arrays (FPGA) is described and is tested with POWER-PC based Single board computer. To reconfigure the system FPGA is used as the best option to modify the system.

In a UART transmission two signal lines are required for the basic operation namely–RXD and TXD. In USRT a clock signal is present. TXD is located in the transmitting end, if there is no data to transmit; the transmission line will stay idle (active high for UART) until triggered or there is data present. RXD is the receiver line, and this is the input to USART. The signal remains in high state when there is no data and it changes to low when the transmission starts. This means the system has an active low transmission and the receiver uses the information to synchronize the incoming data. The logic '0' represents a voltage level of +3 V to +12 V and logic '1' means it has a voltage level from -3 V to -12 V.

2. Efficient USART Design - Description

The internal architecture of USART mainly consists of two modules namely: Transmitter module and Receiver module. This is sown in below Fig. 1. A RAM, 16-bit Cyclic Redundancy Check (CRC) generator using Linear Feedback Shift Register (LFSR) and also a register used to store the result produced after the calculation, 8-bit Synchronous parallel-serial shift register, a glue logic, and a transmitter logic are the next level modules of USART transmitter. In the similar manner the USART receiving module includes the opposite that is serial-parallel shift register, CRC generator, Synchronous comparator and RAM. RAM is used to store temporary data and a dual port RAM is conveniently used.

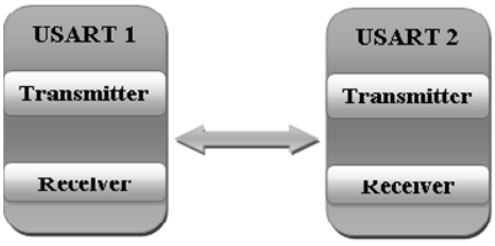

Fig. 1. USART module

USARTs can convert serial data to parallel and parallel data to serial data. This can be explained as follow- "At the transmitter's side the data is transmitted as individual bits in a sequential fashion the parallel to serial converters come into

*Corresponding author: chithrachippy@gmail.com

picture whereas at the receiver's side, USARTs assemble the bits into complete data and thus act as serial to parallel converters."

USARTs are responsible for the basic operation of data format, rate of transfer of data and also few features applicable only to the model. The processor is able to send individual data or data as a block, and the size of this block is decided by the specific model, from the transmitting part of the USART. The data is kept in the temporary storage RAM and then framed based on the mode of transfer and specifications required. In the end, the frame is transferred to the receiving end bit by bit in accordance with the provided clock signal. This data bits are sampled and processed. This processed data from the received frame rests in the temporary storage RAM. The processor reads this data. The job of the receiver is to check this data for any errors and sets the flags accordingly to let the user know about errors. Most USARTs offer a status monitoring mechanism with the help of dedicated status register(s), which is a 16-bit register, using which some of the internal operation features can be monitored. Below block diagram (Fig. 2) shows the internal structure of a USART.

A. Operation Modes

USARTs have two modes of operations- Synchronous and Asynchronous modes. Asynchronous mode is used in most of the applications. Our area of interest lies with the former.

Both the transmitter and receiver are synchronized to the same clock signal in synchronous mode, which is usually based on the data sent from the transmitter and the data is sent along with the clock on a separate link to the receiver. The receiver in turn utilizes the received clock to extract the timing sequence and the data separately and determines the starting and end of the received bit interval.

In asynchronous mode, the transmitter and receiver are configured according to the required timing parameters in advance and special bits namely 8-bit Synchronous bit and 16-bit CRC bits are added to the data frame for synchronization purposes. Thus, timing is embedded in the frame and hence there is no need to send a timing signal to the receiver.

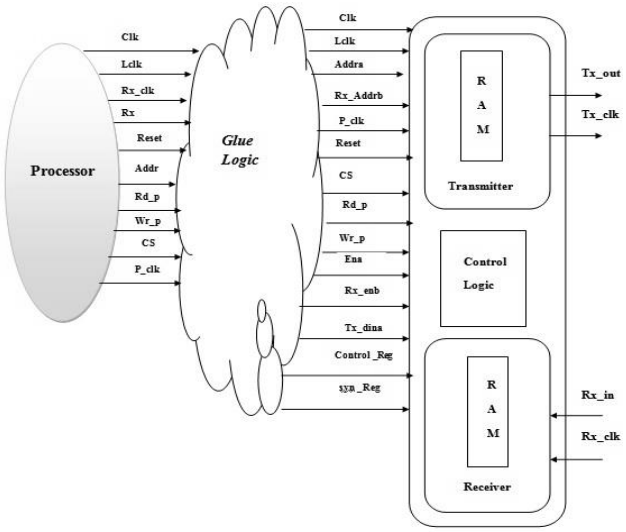

Fig. 2. Block diagram

3. Developed USART- Specifications

High performance and reliability are the requirements of modern serial communication. Along with this the compatibility of the devices should also be considered while considering the design specifications.

To incorporate modern practices, the USART uses 8-bit synchronous mode or asynchronous mode operation, a stop bit which is variable and full duplex mode. But at the same time to keep up the compatibility, this device has five-to-seven-bit transfer and half duplex mode of communications.

This is one of the specific features of USART. The below table gives the detailed specifications of the proposed USART.

Table 1
USART functional specifications

Specification	Justification
Supports the following transmission modes : • Asynchronous (Full/Half duplex modes) • Synchronous (Full/Half duplex modes)	In modern systems, full duplex mode is employed while half duplex mode is retained for compatibility with legacy systems.
Supports a wide range of transmission/reception rates (from 50 Hz to 3MHz). This range is obtained using a baud rate generator that	High frequencies are essential for high speed communication. Lower speeds are needed to communicate with older USARTs. Moreover, lower speeds can be used to minimize cross talk if similar links are adjacent.
Eight-level Transmitter/Receiver Buffer	To account for the high speeds of communication that the USART can reach, blocks of data can be received and buffered until read by the user. Also, this allows the user to issue the transmission of a block of eight-frame size in a single operation. This will also reduce the load on the USART operation in the system.
Parity Supported (Programmable - Enable/Disable parity and Odd/Even parity).	Single error detection techniques might prove beneficial in noisy operation environments.

Variable data lengths supported (Programmable - five to eight bits)	Byte communication is the modem norm. Five to seven bits data length is to retain compatibility with legacy systems.
Variable stop bits supported. (Asynchronous mode) (Programmable – One or two stop bits)	This is to comply with the RS232 standard where two stop bits mode is used to accommodate slightly different clocks in the transmitter and receiver sides when USARTs from different vendors are connected together.
Error Detection of the following errors: • Parity Error • Overrun Error • Framing Error (Asynchronous mode)	Parity error detection provides a measure of the reliability of communication. Framing error detection indicates the necessity of reconfiguring the internal clocking sources at both ends correctly. Finally, overrun error informs that data has been lost and the need to frequently read the received data.
Supports Interrupts (Programmable – with ability of Global Masking)	Most modem systems are interrupt-driven for the reason that interrupt techniques save processing clock cycles in comparison with polling techniques and are essential in real time applications.
Supports Addressable (8-bit Universal – Addresses up to 256 devices	Mainly used in industrial and control applications in multi-drop networks where a master USART can communicate with a certain other slave USART.

4. Design Methodology

In carrying out the USART system design, the methodology used was based on system and software engineering approaches but no specific approach was considered on its own regard. Small variations were incorporated based on the working environment of the specific project.

1. Requirements: It includes defining functionality and essential and required system properties.
2. Design: This phase worked on how the system functionality was to be provided by the components of the system.
3. Implementation and Module Testing: The designing of subsystem was completed and implemented. Later it was mapped to hardware using VHDL. Individual modules were extensively tested for correct functional operation.
4. System Integration: During this phase, the independently developed subsystems were merged together to build the single USART system in an incremental approach.
5. Testing: The overall system was subjected to different testing methodologies to assure correct functionality, reliability and performance.
6. VHDL was used to develop and simulate the USART system and subsystems under Xilinx ISE 13.1 environment.

Table 2
Selection of Baud Rate

S0	S1	S2	BAUD RATE
0	0	0	115200
0	0	1	57600
0	1	0	38400
0	1	1	19200
1	0	0	9600
1	0	1	4800
1	1	0	1200
1	1	1	300

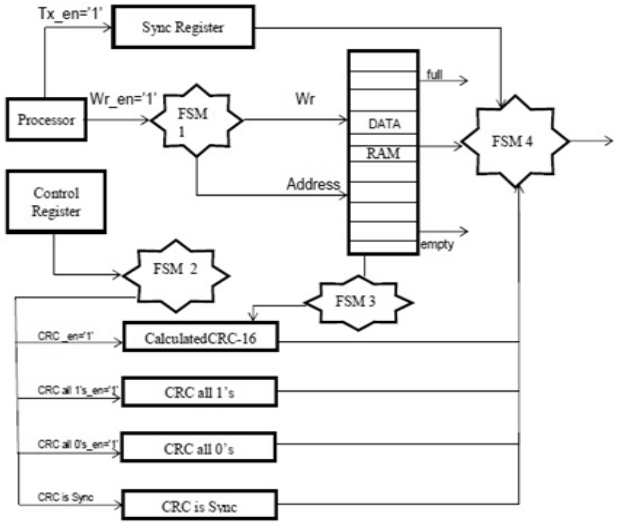

Fig. 3. Transmitter subsystems block diagram

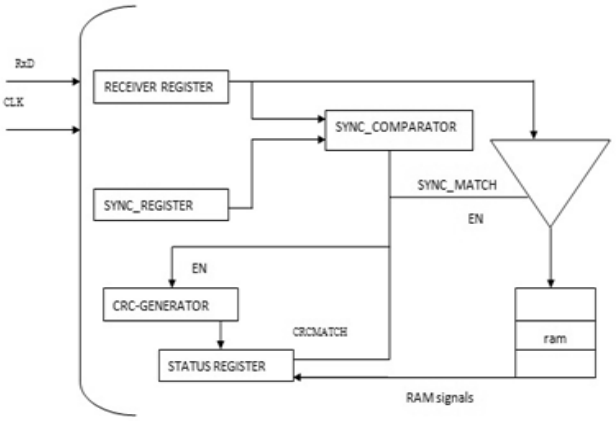

Fig. 4. Receiver subsystems block diagram

5. Testing and Verification

In general, the testing process has two separate goals:

1. To demonstrate that the system meets its requirements.
2. To find any faults or errors that will make the whole system faulty.

For the first goal a set of tests are given where the tests are similar to the expected working requirements. The second is the defect testing, where the test cases are designed to expose defects.

In general, the project went through several phases during the testing process as illustrated in Fig. 5.

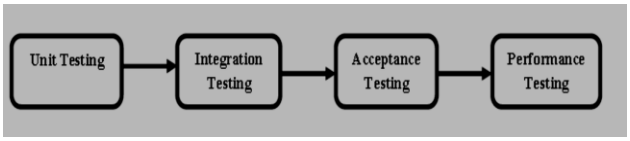

Fig. 5. Systems testing phases

6. Results

The implementation of a USART interface ideal for the application in FPGA based systems is done. Each subsystem is tested separately and compared to the test results after integration.

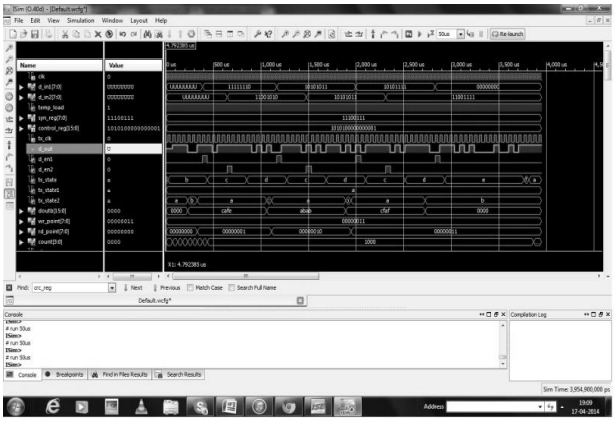

Fig. 6. USART transmission

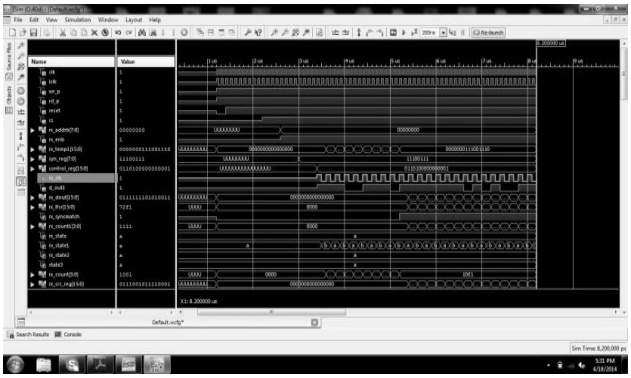

Fig. 7. USART receiver

7. Conclusion

Here both synchronous and asynchronous operations can be operated uniquely using 9-bit address. The USART is capable of recovering from the basic and common serial communication errors. The design uses a VHDL programming.

References

[1] M, Ehlert. (2004). Different approaches of high-speed data transmission standards. Advances in Radio Science, 2004.