FPGA implementation of high speed UART with Automatic Address Identification

Mohamed nisar MP[1], Ganesan R[2]

[1]*PG Scholar, Dept. of ECE, Noorul Islam Center for Higher Education, Noorul Islam University, India.*
nisarpaikatil9@gmail.com

[2] *Professor, Noorul Islam Center for Higher Education, Noorul Islam University, India.*
Dr.ganesh.jass@gmail.com

***Abstract*—Universal asynchronous receiver transmitter (UART) is an integrated circuit using for serial communications. Mostly used for short/long-distance, high speed, low-cost data exchange. Drawback of the conventional UART is the wastage of processing time for Slave devices. Slave devices process both address byte and data byte to get the destination address of data byte. There is no necessity to process the data byte at the time of addresses matching/calculation. The proposed high speed UART design will be driven with a clock speed of 25MHz. FIFO (First-In-First-Out) will be designed to hold the data given from CPU to transmitter end/to CPU from the receiver. FIFO holds data to provide synchronization and to prevent loss of data if CPU is not free at that time. FIFO helps to store data temporarily during high speed transmission. The write frequency using in this design is 25MHz and we can choose read frequency according to our need. In this proposed design, the chance to loss the data is very less. This design also features automatic address identification. It will be achieved by introducing an extra bit in the address/data character to differentiate between an address or a data character. Hence Slave devices can easily distinguish an address byte or data byte. So it processes only the address byte (not processing the data byte) to match with its unique address. It reduces the processing time of the slave devices. A parity bit is also used to detect the error occurred while transmission. Proposed work will be modeled using VHDL and synthesized in ModelSim software.**

***Key words* - modified 10 bit UART, Parity bit, Overflow error, Frame error.**

I.INTRODUCTION

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 and other serial devices [1]. In asynchronous transmission protocol, a start signal is sent prior to each byte, character or code word and a stop signal is sent after each code word. The start signal serves to prepare the receiving mechanism for the reception and registration of a symbol and the stop signal serves to bring the receiving mechanism to rest in preparation for the reception of the next symbol [2].The number of data and formatting bits, and the transmission speed, must be pre-agreed by the communicating parties. Fig.1 shows Asynchronous serial transmission data frame format.

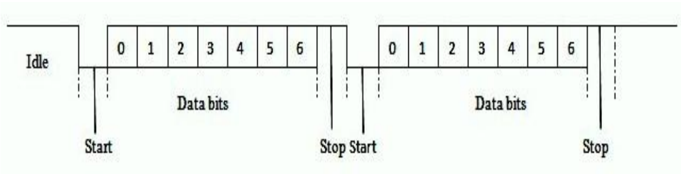


Fig.1 Asynchronous serial transmission

Baud Rate represents the number of bits that are actually being sent over the media, not the amount of data that is actually moved from one DTE device to the other.

II.IMPLEMENTATION OF UART

Every data terminal equipments process each receiving data byte and address byte to detect the coming data belongs to that device. Processing of data bytes to find the destination address is leads to wastage of time since there is no necessity to process the data byte at the time of addresses matching/calculation.

In this design, an extra bit is using to differentiate address byte and the data byte. This bit is setting to 1 to indicate the address byte and setting to 0 to indicate data byte. So that every slave devices can process this bit and check the byte representing the data or address. The receiver processes the coming byte only if it is address byte. If the address is matching, then it receives the following data byte. If the address matching fails, it will ignore the coming data stream. The receiver will not process the data byte, so that slave devices can save a large amount of processing time.

A parity bit is used to check the occurred error while transmission. UART transmitter finds the parity of sending data and setting the parity bit. The parity bit will be 1 for odd parity and 0 for the even parity. The transmitter sends this parity bit along with the data byte. The receiver will find the parity of received data and compare with the parity at the sending time. It can fine if the error occurred while transmission time. The data frame format have a start bit 0 ,8-bit address or data byte, data/address identifier bit, a parity bit and a stop bit.Fig.2 show the address character Fig.3 shows data character.

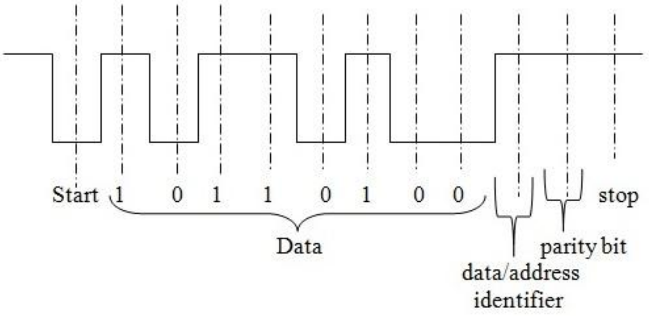


Fig.2 Address character

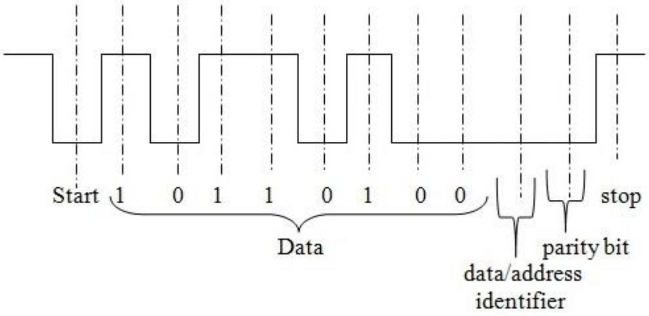


Fig.3 Data character

The architecture of the UART is shown the Fig.4. It consist of baud rate generator, a transmitter section and a receiver section. Baud rate generator gives the enable signal to both the transmitter and the receiver. The proposed architecture of baud rate generator is shown in fig.5. It consists of delay elements and multiplexer. Delay elements are used to provide delays or we can say to generate lower data rate clocks. User can select any clock signal according to the practical application with the help of this multiplexer.

Transmitter contains one transmitting buffer and transmitting register to store the data. Load signal is using to indicate loading the data in to the receiver. We are uploading a 9 bit data indicates data0, which contains a 8bit data and a single bit data/address identifier. First the data is loading in to the transmitting buffer then it copied in to the transmitting register for the transmission. When the transmitting register is sending the data, second data will be uploading to the transmitting buffer at the same time. So that we can increase the speed of the process. TxD signal is the output data frame sending to the transmitter which will be contained the start bit, parity bit and the stop bit along with the 9 bit data. TregE signal indicates transmitting register is empty and TbufE signal indicates transmitting buffer is empty. Transmitter block has shown in Fig.6.

The receiver sets Read signal to start the reading of received data. And the receiver finds the frame error, parity error and the overflow error.

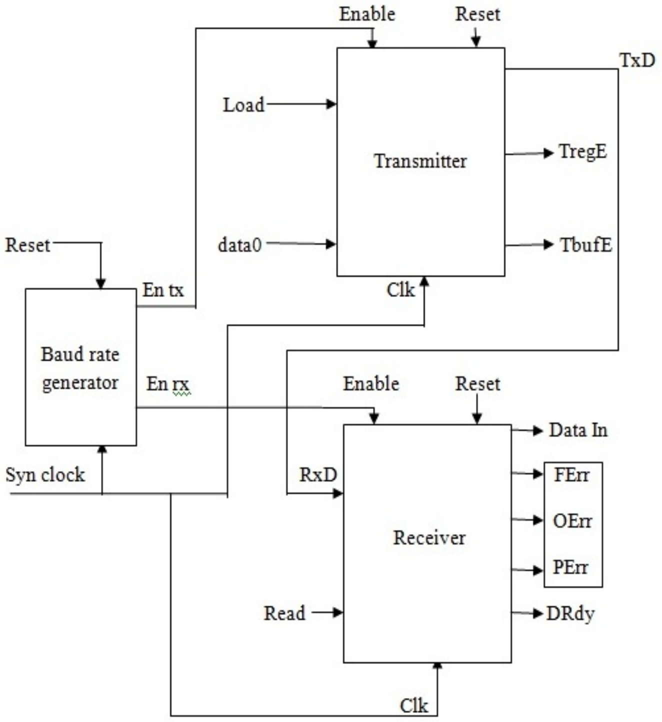


Fig.4 Proposed UART architecture

The parity of 9bit received data will be calculated and it compares it with the 10th bit of received frame to find the parity error. PErr will be 1 if there is no parity error. If the register over flows, then over flow error comes and OErr become 1. Frame error is error occurring with the stat or stop bit. FErr is setting to high if there is any frame error. DRdy signal indicated that output data is ready to come. Data In is the final 9bit data coming out of the receiver. Receiver section is shown in Fig.7.

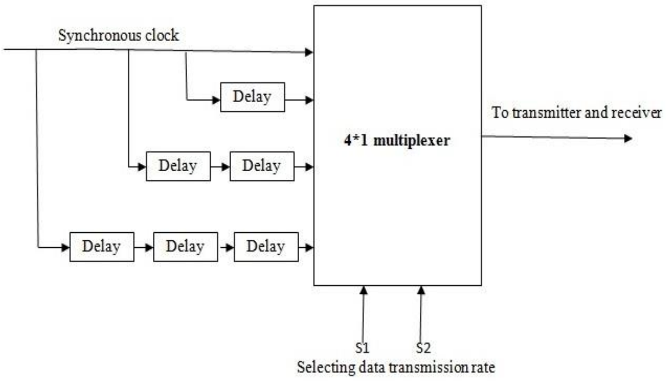


Fig.5 Baud rate generator

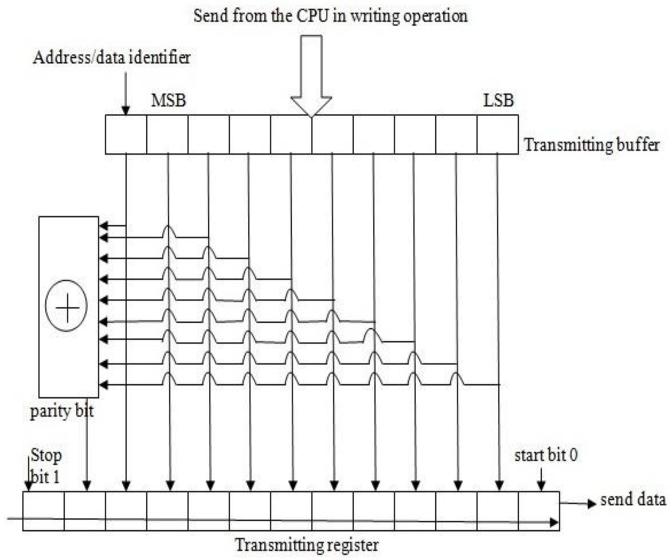


Fig.6 Transmitter

Transmitter sends 8 bit data / address character and an address identifier bit. It finds the odd parity of these 9 bits and it will be considered as the 10th bit. A start bit 0 and a stop bit 1 will be implementing at the beginning and ending of this 10 bit frame.

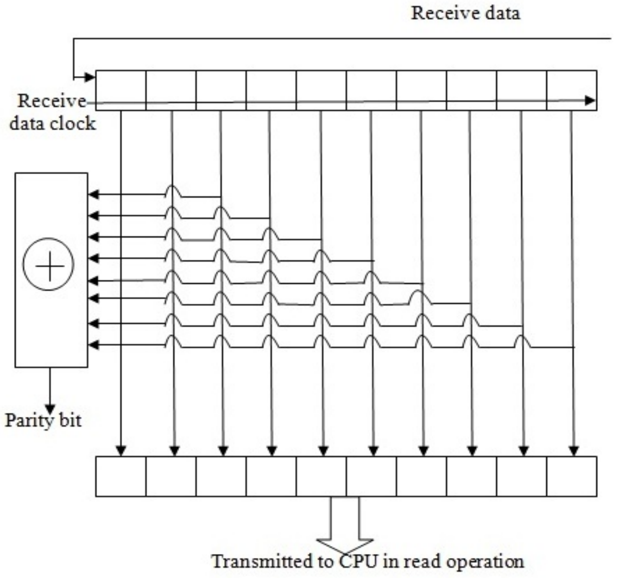


Fig. 7 Receiver

The receiver receives frame coming from the transmitter and it expel the star and stop bits. Then it finds the odd parity of first 9 bits and comparing it with the 10th bit to find if the parity error is present. The receiver also finds the overflow error and the frame error.

III. RESULTS AND DISSCUSSION

This architecture is implemented using VHDL and simulated in ModelSim SE PLUS 6.2c. The output wave form is shown in Fig.8.

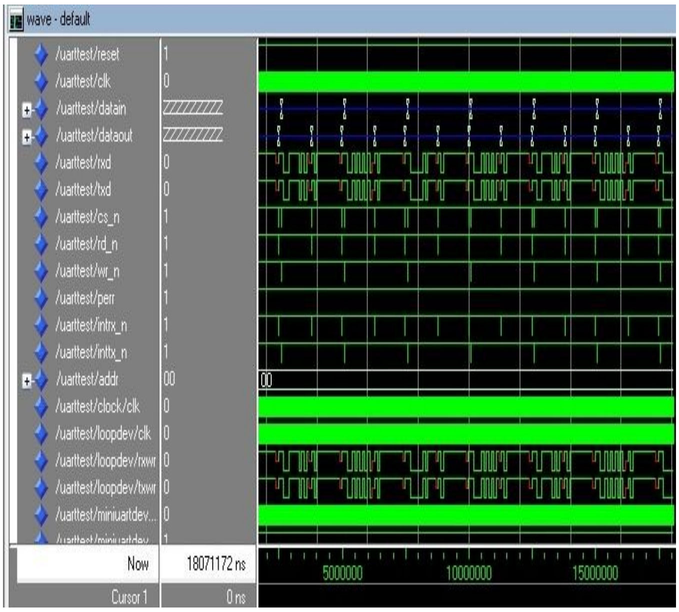
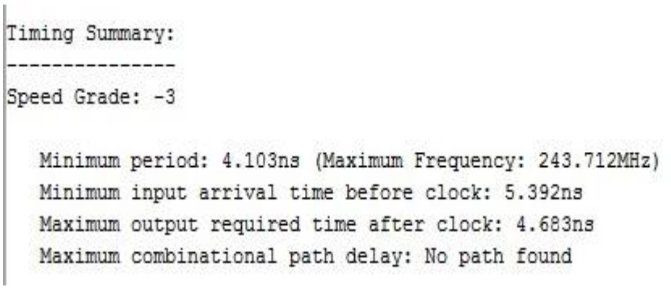


Fig.8 Simulated output waveform

Also an analysis and estimation has been done by using Xilinx software to find out the usage of available resources while doing the FPGA implementation of this design.

Fig.9 gives a clear idea about timing parameters of this design.

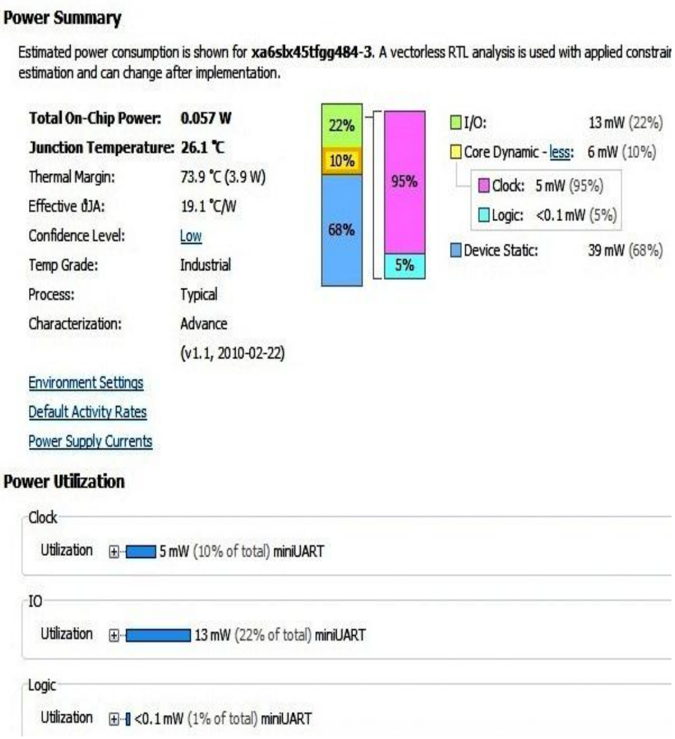


```
Timing Summary:
---------------
Speed Grade: -3

   Minimum period: 4.103ns (Maximum Frequency: 243.712MHz)
   Minimum input arrival time before clock: 5.392ns
   Maximum output required time after clock: 4.683ns
   Maximum combinational path delay: No path found
```

Fig. 9 Timing parameters

Power estimation has done to find out the power utilization of the entire system and it is shown in Fig.10. Total On-chip power is 0.057W. 22% of this power is utilized for the I/O devices and 10% is used for the core dynamic elements such as clock, logic devices. Remaining 68% of the total power is used for device static. Power utilization summary has clearly shown in the Fig.10.



Power Summary

Estimated power consumption is shown for **xa6slx45tfgg484-3**. A vectorless RTL analysis is used with applied constrai
estimation and can change after implementation.

Total On-Chip Power:	**0.057 W**
Junction Temperature:	**26.1 ℃**
Thermal Margin:	73.9 ℃ (3.9 W)
Effective ϑJA:	19.1 ℃/W
Confidence Level:	Low
Temp Grade:	Industrial
Process:	Typical
Characterization:	Advance (v1.1, 2010-02-22)

I/O:	13 mW (22%)
Core Dynamic - less:	6 mW (10%)
Clock:	5 mW (95%)
Logic:	<0.1 mW (5%)
Device Static:	39 mW (68%)

Environment Settings

Default Activity Rates

Power Supply Currents

Power Utilization

Clock

Utilization ⊞ 5 mW (10% of total) miniUART

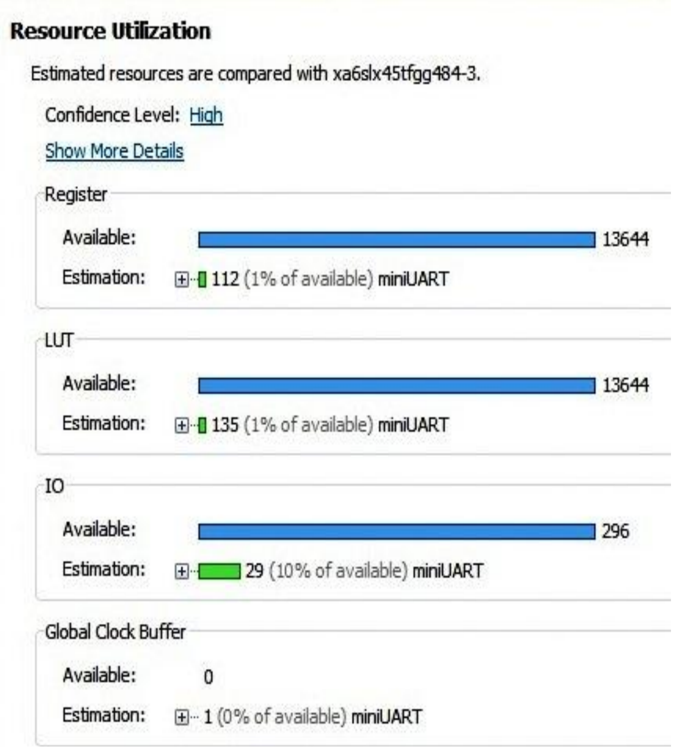
IO

Utilization ⊞ 13 mW (22% of total) miniUART

Logic

Utilization ⊞ <0.1 mW (1% of total) miniUART

Fig: 10 Power utilization summaries



Resource Utilization

Estimated resources are compared with xa6slx45tfgg484-3.

Confidence Level: High

Show More Details

Register

Available: 13644

Estimation: ⊞ 112 (1% of available) miniUART

LUT

Available: 13644

Estimation: ⊞ 135 (1% of available) miniUART

IO

Available: 296

Estimation: ⊞ 29 (10% of available) miniUART

Global Clock Buffer

Available: 0

Estimation: ⊞ 1 (0% of available) miniUART

Fig. 11 Resource utilization

Resource utilization is the amount of registers, look up tables and IOs used to implement this design which is shown in the Fig: 11. Total112 registers are used to fulfill this design, that is 1% of the available registers.135 look up tables are used which is also 1% of the total available. 29 IOs are used among 296 IOs. There is no clock is available within the FPGA so an external crystal oscillator has to be used to generate the clock pulse.

A comparative study is shown in the bellow table. Table gives the differences in parameters used for some former designs and novel design. It clearly shows that the novel design occupies in less area in FPGA.

TABLE I
Comparative study of different parameters related to FPGA

Selected device / **Parameters**	xc3s500e5fg320[3] and 3s500efg320-4 [4]	xa6slx45tfgg484-3
	Normal design	**Novel design**
Number of I/0 s	16	29
Look up tables (LUP)	773	135
Flip Flops	458-626	112
Global clocks	1- 4	1
Speed grade	-4	-3
Minimum period (Maximum Frequency)	10.592ns (94.411MHz)	4.103ns (243.712MHz)
Minimum input arrival time before clock	6.232ns	5.392ns
Maximum output required time after clock	7.661ns	4.683ns
Maximum combinational path delay	No path found	No path found

IV CONCLUSION

A modified UART is designed with automatic address indication. One parity bit is used to check the error occurred during transmission. The 10-bit UART design is implemented using VHDL and simulated using ModelSim software to see the functionality of each sub-modules. Xilinx software is also used for the analysis and to find different estimations to realize FPGA implementation of this design. Usage of data/address identification bit gives the advantage of saving the UART processing time by comparing the address and decides whether to receive or ignore the incoming data packets. And the baud rate selection is possible for different applications by using a multiplexer at the baud rate generator. With all the features mentioned, it adds to the flexibility, stability and reliability to the conventional UART design that is widely being used.

REFERENCES

[1] NennieFarinaMahat "Design of a 9-bit UART Module Based on VerilogHDL" IEEE- ICSE2012 Proc., 2012, Kuala Lumpur, Malaysia

[2] J. Norhuzaimin, and H.H. Maimun, "The design of high speed UART," Asia Pac.Conf. on Appl. Electromagnetics (APACE 2005), Johor, Malaysia, Dec. 2005.

[3] Nagaraju. A, S. Nagi Reddy, "A Robust UART Implementation for Industrial Applications on FPGA". International Journal of Science and Research (IJSR), Volume 2 Issue 11, November 2013.

[4] Bibin M C, Premananda B S, "Implementation of UART with BIST Technique in FPGA". International Journal of Inventive Engineering and Sciences (IJIES)ISSN: 2319–9598, Volume-1, Issue-8, July 2013

[5] Ritesh Kumar Agrawal, VivekRanjan Mishra, "The Design of High Speed UART", Proceedings of 2013 IEEE International Conference on Information and Communication Technologies (ICT 2013)

[6] FANG Yi-yuan, CHEN Xue-jun, Design and Simulation of UART Serial Communication Module Based on VHDL

[7] YongchengWang, KefeiSong , "A New Approach to Realize UART" , 2011 International Conference on Electronic &Mechanical Engineering and Information Technology

[8] He Chun-zhi , Xia Yin-shui, Wang Lun-yao, "A Universal Asynchronous Receiver Transmitter Design".

AUTHORS PROFILE

Mohamed nisar MP born on 1991 and received the B.E. degree in electronics and communication engineering from Visvesvaraya technological university, Belgum, India, in 2012. He is currently pursuing the Masters degree in Communication systems from Noorul islam university, Kanyakumari, India. His current research interests include VLSI and Mobile communication networks.

Ganesan R born on 1976 and received bachelor, Master and Doctorate degree in engineering in 1999, 2001 and 2011 respectively. He has been working as Professor in the department of ECE, Noorul Islam University, India. He has published ten international journals, three international conference papers. His area of research is VLSI architecture for network security and digital image processing algorithms. He is a life member of ISTE and IETE. He is a review committee member for many peer review journals. He is a deputy project director for NIUSAT payload design and steering committee member of NIUSAT which is an innovative program is being developed by NIU, India.