

# SYSTEMVERILOG IMPLEMENTATION OF UART WITH SINGLE ERROR CORRECTION AND DOUBLE ERROR DETECTION

T. Yamuna<sup>1</sup>, M.M. Dasu<sup>2</sup>

<sup>1</sup>Student, Department of ECE, GIST college, Andhra Pradesh, INDIA

<sup>2</sup>Assistant Professor, Department of ECE, GIST college, Andhra Pradesh, INDIA

\*\*\*

**Abstract** - A Universal Asynchronous Receiver / Transmitter (UART) is responsible for performing the main task in serial communications with computers. This paper presents Implementation of UART with single error correction using System Verilog. The design is implemented in spartan 6 FPGA. In Communication the cost as well as complexity of the system increases due to simultaneous transmission of data bits on multiple wires so noise is produced in multiprocessor communication to achieve this we are using hamming code method

**Key Words:** FEC (Forward Error Correction), Hamming Code, SEC code, Universal Asynchronous Receiver Transmitter (UART), Xilinx ISE.

## 1. INTRODUCTION

UART stands for "Universal Asynchronous Receiver/Transmitter". The Universal Asynchronous Receiver/Transmitter (UART) controller is the key component of the serial communications subsystem of a computer. It is the chip (on the modem circuit board on internal modems, and on the motherboard for external COM ports) that allows the CPU to share data with the serial device (modem) by converting parallel data format into a serial data stream to be sent over the phone lines as an analog signal. It then receives analog signals in a serial stream and converts them to parallel data to communicate back to the CPU. The UART takes bytes of data and transmits the individual bits in a sequential fashion. At the destination, a second UART re-assembles the bits into complete bytes. It is commonly used in computer serial ports. One of the significant differences between TTL level UART and RS-232 is the voltage level. Valid signals in RS-232 are  $\pm 3$  to  $-\pm 15$ V, and signals near 0V is not a valid RS-232 level.

In this paper, we present a UART with error detecting and correction capability. The inclusion of a hamming encoder in the transmitter section and hamming decoder in the receiver section can correct up to one error by using Systemverilog.

## 1.1. SYSTEMVERILOG IMPLEMENTATION

The design is implemented in Systemverilog and targeted towards Xilinx Spartan-6 FPGA. Advantage of FPGA implementation is low cost and time. Time to market is also short and flexible as well.

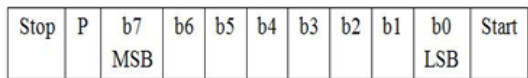
## 2. METHODS AND MATERIAL

### 2.1 Proposed Architecture of UART

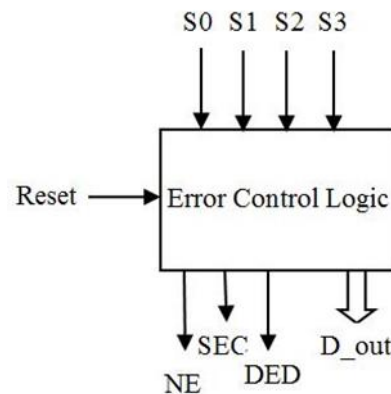
UART works in asynchronous mode which does not require transmission of clock along with the data. The proposed UART employs a 12 bit frame as shown in Fig- 1. The character length can vary from 5 to eight bits and hence the frame length can vary from 8 to 11 bits. This works in the two modes. First one is normal mode and the second one is error correction and detection mode. This paper principally deals with the error correction mode. In the error correction mode, (8, 4) extended hamming code, also called as SEC-DED code, is employed for single bit error correction and double error detection. In this mode four data bits are transmitted per frame. Four hamming bits are concatenated in the LSB position with 4 data bits forming 8 bit hamming code. In the transmitter, the data frame is formed with one start bit followed by 8 bits of hamming code, a „1“, and one stop bit. Start bit is a „0“ and stop bit is a „1“. This frame is transmitted by the transmitter bit by bit. When received by the receiver, the overhead are separated from the frame. The hamming code is decoded to correct the error in the received data. Errors up to one can be corrected in this method and two errors can be detected. The data bits after correction are available in parallel form to be accepted by the microprocessor. The frame formats for the proposed UART

in the error correction mode and also in normal modes are shown in Fig- 1 and Fig- 2 below.

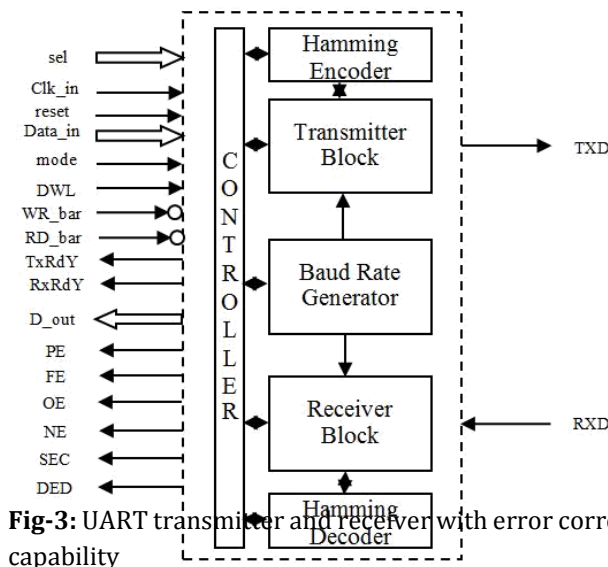
**Fig- 1:** Frame Format – Error Correction Mode



**Fig- 2:** Frame Format – Normal Mode



**Fig- 4:** Hamming Decoder



**Fig-3:** UART transmitter and receiver with error correction capability

Fig- 3 shows the high level block diagram of UART transmitter and receiver with error correction capability. The “clk\_in” signal is the system clock which is used to derive the transmission clock and receiver clock to determine baud rate of transmission and reception using baud rate generator. The “reset” signal resets the UART and put the machine in idle mode clearing all the internal registers. The 8 bit data\_in the input data lines from the processor and the data from these data lines is transmitted over “TXD” line. When “mode” = 1 then the selected mode is error correction mode and if “mode” =0 then the selected mode is normal mode. the transmission clock and receiver clock to determine baud rate of transmission and reception using baud rate generator. “DWL” is data word length, it is a two bit signal which is used to determine the length of character

**2.2 Hamming**

Hamming decoder takes the 9 bit assembled data, discard the 9th bit as it is always “1” and used only for the uniformity of data frame size. The remaining 8 bits are has 4 bit of data and 4 bits of check bits which are used to identify errors. If the error is of only 1 bit then it is corrected and if it is of 2 bit then it can only is identified. Respective flags are generated in this process. Fig- 8 shows the hamming decoder logic used in this design. s3, s2, s1, s0 bits are generated to identify the location and type of error. Table 1 shows the possible values of s3-s0 and corresponding meaning. The “d\_out” is an 8 bit data to the processor, 4 least significant bits are data and remaining 4 bits are always zeros

**Table 1** Error Control Logic

s0	s1s2s3	NE/SE/DE	Error Position	Flag
0	000	no error	-	NE
1	011	single error	D0	SEC
1	101	single error	D1	SEC
1	110	single error	D2	SEC
1	111	single error	D3	SEC

**2.3 Transmitter**

The transmitter block is shown in Fig- 5. When the “mode” input is zero then the selected mode is normal mode in which only parity bit is generated for error detection. Total 9 bits are generated from this unit, 8 data bits and 1 parity bit. When the “mode” input is one then the selected mode is error correction, in this mode 4 LSB data bits are taken from the data bus input “data\_in” and hamming code is applied is to it to from an 8 bit word. Total 9 bits are generated from this unit, 4 hamming bits, 4 data bits and 1 bit is always „1“. The frame format is shown in Fig- 1 and 2 respectively. The transmitter hold register (THR) is used to hold the data from

the parity encoder or from the hamming encoder. The TSR block does the framing i.e. insertion of start and stop bit to the data word and transmits data serially over “TXD” line on rising edge of “clk\_baud”. Also the character length can be varied using DWL only in normal mode. This length is fixed to 8 bits in error correction mode

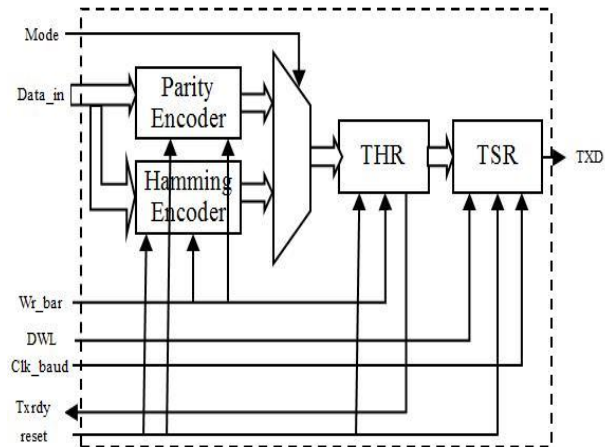


Fig-5: Transmitter with Hamming Encoder

### 2.4 Receiver

The receiver hold register (RHR) starts assembling the data whenever a zero (start bit) is arrived over “rxd” line and checks for stop bit „1“ after the programmed character length and parity bit, if stop bit is not received after programmed word length then a framing error “FE” is generated. The data is sampled at rising edge of bits (variable in case of normal mode depending upon “DWL” signal) and 1 parity bit (always 1 in case of error correction mode) is forwarded to either parity decoder logic or hamming decoder logic depending upon the value of “mode” input signal. The parity decoder logic generates parity error in case of parity error. The hamming decoder logic decodes the input data and checks for any error, if single bit error the error is corrected and data is forwarded to SBUF, also “SED” signal is made high, else if double error are detected then “DED” signal is made high and data is discarded. If no error is detected then the data is forwarded to SBUF and “NE” signal is made high. The SBUF block then generates “rxrdy” signal to inform processor about the arrival of new data byte, the processor reads the SBUF in response to this “rxrdy” signal. Prior to the read cycle if next data byte is arrived then over run error “OE” is generated and UART starts assembling next data byte. As shown in the Receiver block of UART

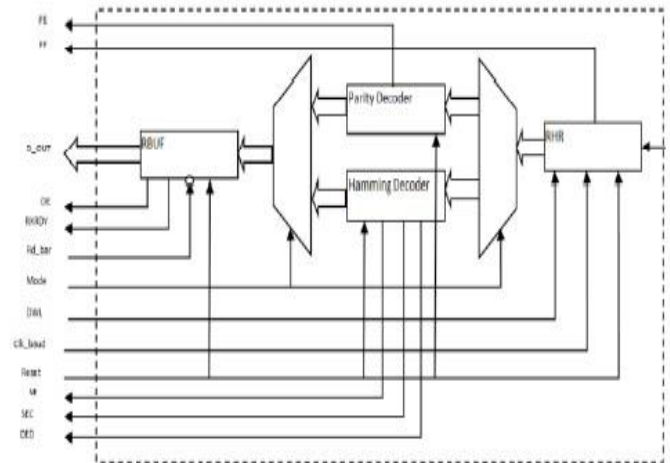


Fig-6 : The receiver block of UART

### 3. Simulation Results

In this section simulation of UART transmitter and receiver is presented. We have implemented our design on Xilinx Spartan 6 FPGA and used Questa simulator 10.3 for simulation, synthesis and implementation. Fig- 11 shows the simulation of UART transmitter in normal mode.

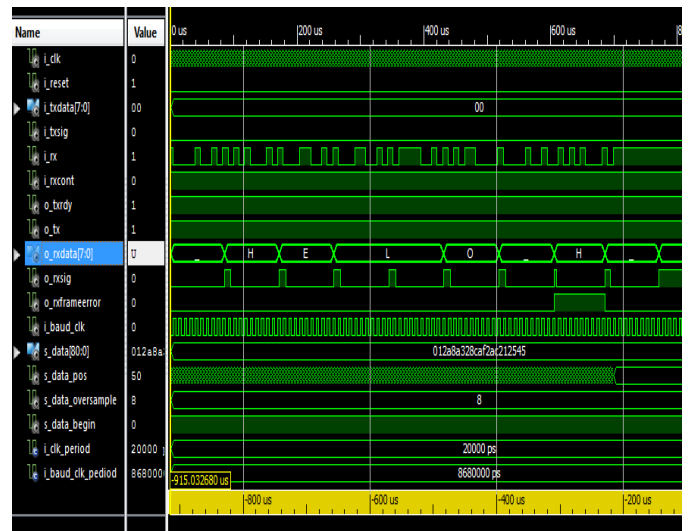


Fig-7. Simulation – Transmitter – Error Correction Mode

After the “wr\_bar” signal goes the “txrdy” goes low and data is transmitted on rising edge of “clk\_baud” over “txd” line serially. The parity bit PB and stop bits are also shown in Fig- 7.

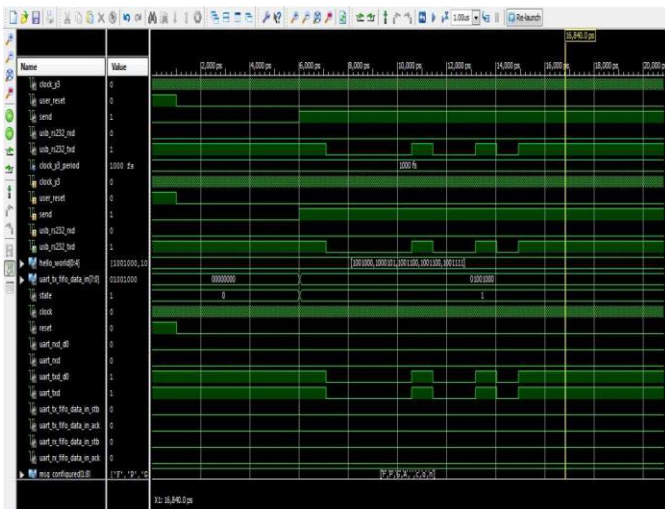


Fig-8.Simulation – Receiver – Error correction Mode

Fig- 8 shows the simulation of UART receiver in Error correction mode. The data collection starts with the arrival of start bit and then on every rising edge of “clk\_baud” data is collected. The parity bit is checked and then after the arrival of stop bit “rxrdy” is signal is generated to processor, in response to this signal the processor initiates read cycle and data is transferred to “d\_out” and the “rxrdy” signal goes low. Fig- 7 shows simulation of UART transmitter in error correction mode the data is loaded when the “wr\_bar” signal goes low, then the incoming data is encoded using extended hamming code, here the encoded data for input bits “0001” is “00010111” and it is transmitted over “txd” line at the rising edge of “clk\_baud”. Fig- 8 shows the simulation of UART receiver in error correction mode. At the arrival of start bit at “rxrd” the receiver block starts assembling the data at each rising edge of “clk\_baud”. After complete assembling of data, it is checked for any error, in this case no error was received and hence the “NE” flag is raised high. And all the other error flags are kept low.

**4. CONCLUSIONS**

The UART with single error correction and double error detection capability is implemented in Systemverilog and tested. This UART design can be implemented in industrial and noisy environment which ensures error free reception upto a likelihood of 10% of errors.

**ACKNOWLEDGEMENTS**

We the authors of this paper would like to gratefully acknowledge with thank M.M. Dasu for explaining the changes made in the specifications we are grateful for many insightful comments about our specifications that helped us

improve our paper. For constant encouragement and support for enabling us to submit this paper & to the management of Gandhiji Institute of Science and Technology, Jaggayyapet, Andhra Pradesh, India to have provided the laboratory facilities for development and execution of this project.

**REFERENCES**

[1] Naresh Patel, Vatsalkumar Patel, Vikaskumar Patel, "VHDL Implementation of UART with Status Register", 2012 IEEE International Conference on Communication Systems and Network Technologies, 2012.

[2] R.W.Hamming, "Error detecting and error correcting codes", The Bell System Technical journal Vol. XXIX, April 1950, Vol. 2, American Telephone and Telegraph company.

[3] Elmenreich W, Delvai M, "Time-triggered communication with UARTs", 4thIEEE International Workshop on Factory Communication Systems, 2002, pp. 97- 104, 2002.

[4] Gallo R, Delvai M, Elmenreich W, Steininger A, "Revision and verification of an enhanced UART", 2004, Proceedings, 2004 IEEE International Workshop on Factory Communication Systems, pp. 315- 318, 22-24, Sept. 2004.

[5] Norhuzaimin J, Maimun H.H, "The design of high speed UART", Asia- Pacific Conference on Applied Electromagnetics (APACE), pp. 20- 21, Dec. 2005.

[6] Himanshu Patel, Sanjay Trivedi, R. Neelkathan, V. R. Gujraty , "A Robust UART Architecture Based on Recursive Running Sum Filter for Better Noise Performance", Held jointly with 6th International Conference on Embedded Systems, 20th International Conference on VLSI Design, pp. 819-823, Jan. 2007.

[7] Fang Yi-yuan, Chen Xue-jun, "Design and Simulation of UART Serial Communication Module Based on VHDL", 2011 3rd International Workshop on Intelligent Systems and Applications (ISA), pp. 1-4, 28- 29, May 2011.

[8] Yongcheng Wang, Kefei Song, "A new approach to realize UART", 2011 International Conference on Electronic and Mechanical Engineering and Information Technology (EMEIT), pp. 2749-2752, 2011.

[9] Idris, M.Y.I, Yaacob M, "A VHDL implementation of BIST technique in UART design", 2003 Conference on Convergent Technologies for Asia-Pacific Region (TENCON), pp. 1450-1454, 2003.

[10] Chun-zhi He, Yin-shui Xia, Lun-yao, Wang, "A universal asynchronous receiver transmitter design", 2011 International Conference on Electronics Communications and Control (ICECC), pp. 691-694, 9-11, Sept. 2011.

[11] Prof. Rami Abielmona, "Project UART Design", for CEG 3150, Digital Systems-II, Fall 2004, November 24, 2004.