VHDL Implementation of UART with Detecting wrong bit position and Error Correction Capability

Sourabh Tiwari
PG-Scholar, II-M.Tech., [VLSI & ES]
Electronics and Communication Engineering
Gyan Ganga Institute of Technology and Sciences
Jabalpur, India
tsourabh9@gmail.com

Dr. Vinod Kapse
Principal
Dept. of Electronics and Communication Engineering
Gyan Ganga Institute of Technology and Sciences, M.P., India
vinodkapse@ggits.org

***Abstract*-UART is important in industrial environment which employs multiprocessor for working and in this process noise is likely to affect the data and due to which control in system is improper. Several existing UART resolve this problem by using error detecting logics which detect error, it detects error and retransmit the data which takes time which is more than two times. This paper presents a novel VLSI implementation of UART designed to include (11,4) extended hamming code that can correct up to one error and detect up to two errors. This improves the noise immunity of the system optimizing the error free reception of data. The whole design is implemented in Xilinx ISE 12.2 simulator targeted to Xilinx Spartan 6 FPGA[1].**

Keywords – Universal Asynchronous Receiver Transmitter(UART), Hamming Code, SEC code, DED code, Xilinx ISE.

1. INTRODUCTION

Microprocessor has peripheral called UART for asynchronous serial communication. Data word/byte taken at a time from microprocessor then adds start, stop, parity bit, etc., to form a frame[3][6]. Frame formed should be transmitted one by one bit. Here start and stop bits decides the synchronization between receiver and transmitter. UART for integration with FPGA has been implemented for reliable transmission of data. UART with control word has been implemented by VHDL and Verilog is proposed with capability that includes stop bit length, parity check, character length, baud rate factor[7].

In this paper, we present a UART with capability that it detect error and without retransmitting the data. It detects the error bit position and simply change that to get the correct data without any error or makes it error free[1].

2. VHDL IMPLEMENTATION

VHDL is a strongly typed language and in digital design it is widely accepted. The design is implemented in VHDL[4] and Verilog as well and targeted towards Xilinx Spartan 6 FPGA. Advantage of FPGA implementation is low cost and time[5].

3. PROPOSED ARCHITECTURE OF UART

UART does not require transmission of clock along with data because of the reason that it works in asynchronous mode[1]. The proposed UART employs a 12bits frame as shown in fig.1. Its working includes two modes first is error detection means detection of bit error position and then in second step correcting that bit. This paper deals with both the technique finding bit having error and correcting that bit[1].

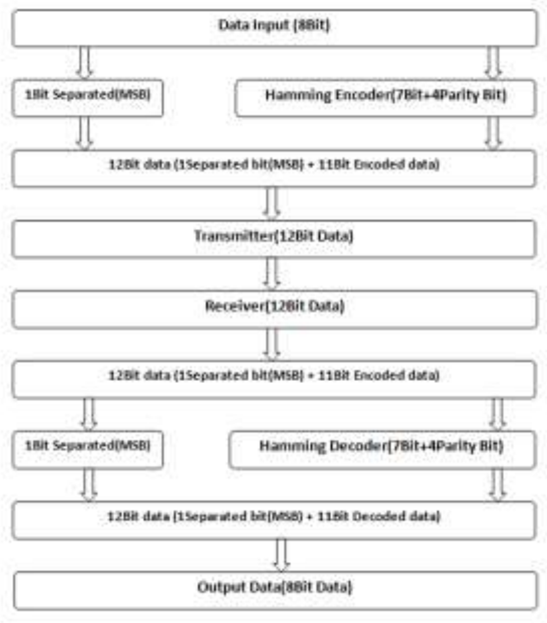

Data Input (8Bit)
1Bit Separated(MSB)
Hamming Encoder(7Bit+4Parity Bit)
12Bit data (1Separated bit(MSB) + 11Bit Encoded data)
Transmitter(12Bit Data)
Receiver(12Bit Data)
12Bit data (1Separated bit(MSB) + 11Bit Encoded data)
1Bit Separated(MSB)
Hamming Decoder(7Bit+4Parity Bit)
12Bit data (1Separated bit(MSB) + 11Bit Decoded data)
Output Data(8Bit Data)

Fig.1Functioning Block Diagram of UART

In detecting wrong bit position mode, (11,4) shown in Fig.1 extended hamming code is employed for detecting error position and correcting that bit. In this mode seven data bits are transmitted per frame and four hamming bits in 2^k are placed in between data bits, forming 11 bit hamming code.

In transmitter, the data frame formed with one start bit including 11 bit hamming data code and two stop bits. Start bit is ‘0’ and stop bit is ‘1’. And frame transmitted by transmitter is one by one means bit by bit. Then transmitted to hamming encoder for encoding and then received by hamming decoder where decoding is done and now 11bit data is separated and hamming decoder decode the data to check the error bit position and by correcting if there is error it is corrected and data is separated by hamming parity bits and actual data is received.

B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1

Fig.2 Bit Position

Here bit number ‘1’ ‘2’ ‘4’ ‘8’ are bit added for hamming encoding and decoding as shown in fig. 2 which is helpful in error bit position number detecting and correcting.

The proposed UART[2][1] has a transmitter, a control word register, a hamming encoder, a hamming decoder, a receiver.

Transmitter : The transmitter has two sections, a hamming encoder as shown in Fig.4 and a Control Word Register as shown in Fig.3.The serial transmitter section consists Transmitter Hold Register (THR) and Transmitter Shift Register(TSR).

Control word register : The control word register shown in Fig.3 is an 8 bit programmable register. The programming of this register determines the no. of data bits to be transmitted, the baud rate, the character length, the no. of stop bits, whether parity check present and the type of parity.

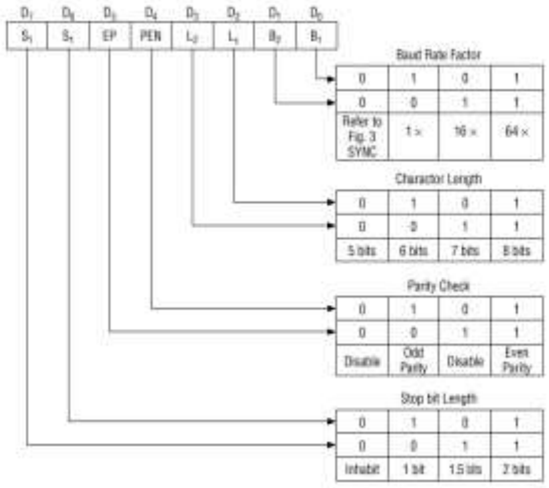


Fig.3 Control word register for UART

Hamming Encoder : Hamming encoder as shown in Fig.4 is used to encode the data to be sent through UART transmitter so that no error containing data to be transmitted[1]. To calculate the numbers of redundant bits (r) required to correct d data bits, let us find out the relationship between the two. So we have (d + r) as the total number of bits, which are to be transmitted; then r must be able to indicate at least d + r + 1 different values. Of these, one value means no error, and remaining d + r values indicate error location of error in each of d + r locations. So, d + r + 1 states must be distinguishable by r bits, and r bits can indicates 2^r states. Hence, 2^r must be greater than d+r+1. The value of r must be determined by putting in the value of d in the relation. For example, if d is 7, then the smallest value of r that satisfies the above relation is 4. So the total bits, which are to be transmitted is 11 bits (d + r = 7 + 4 =11). These redundancy bits are placed in positions 1, 2, 4 and 8 (the positions in an 11-bit sequence that are powers of 2).

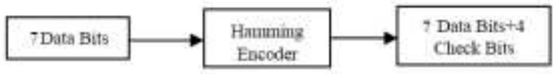
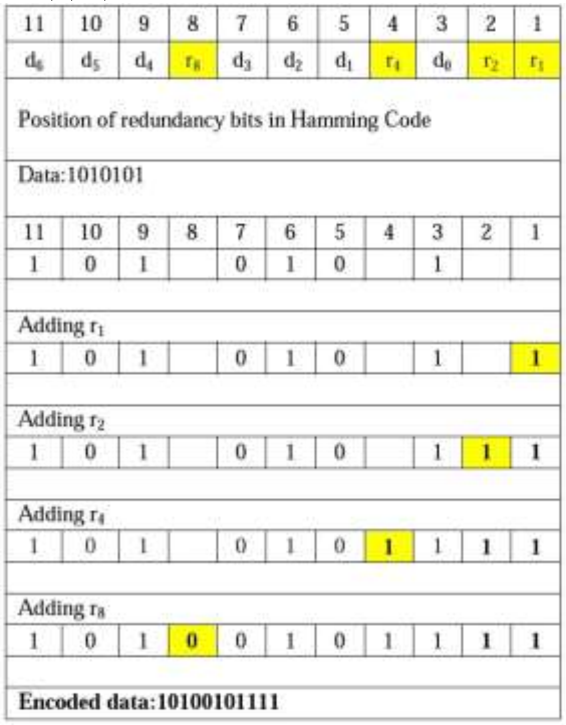


Fig.4 Extended (11,4) Hamming code generation

r1: 1, 3, 5, 7, 9, 11
r2: 2, 3, 6, 7, 10, 11
r4: 4, 5, 6, 7
r8: 8,9,10,11



11	10	9	8	7	6	5	4	3	2	1
d_6	d_5	d_4	r_8	d_3	d_2	d_1	r_4	d_0	r_2	r_1
Position of redundancy bits in Hamming Code										
Data:1010101										
11	10	9	8	7	6	5	4	3	2	1
1	0	1		0	1	0		1		
Adding r_1										
1	0	1		0	1	0		1		1
Adding r_2										
1	0	1		0	1	0		1	1	1
Adding r_4										
1	0	1		0	1	0	1	1	1	1
Adding r_8										
1	0	1	0	0	1	0	1	1	1	1
Encoded data:10100101111										

Fig.5 Positions of bits

Receiver : The serial receiver section also contains a Hamming Decoder Receiver Buffer Register (RBR) and Receiver Shift Register(RSR).

Hamming Decoder : A Hamming decoder Shown in Fig.6 with an example of 11 bit of data can be received with reset as inputs. 7 bits of output data which has error corrected information bits. The 7 bit of data as output to show the position of error if any.

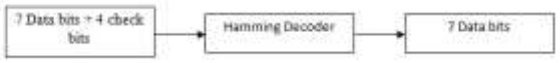


Fig.6 Hamming decoder output

At the receiving end the parity bits are recalculated. The decimal value of the k parity bits provides the bit-position in error, if any. Hamming code is used for correction for 11-bit numbers (d11 d10 d9 d8d7 d6 d5 d4 d3 d2 d1) with the help of four redundant bits (r4 r3 r2 r1).with error positions (C8 C4 C2 C1) For the example data 100100101111.

First r1 is calculated considering the parity of the bit positions, 1, 3, 5, 7, 9, 11. Secondly the parity bits r2 is calculated considering bit positions 2, 3, 6 7,10,11 then, the parity bits r4 is calculated considering bit positions 4, 5, 6 and 7, finally the parity bits r8 is calculated considering bit positions 8,9,10,11 as shown. If any corruption occurs in any of the transmitted code 10000101111, the bit position in error can be found out by calculating r4 r3 r2 r1 at the receiving end. For example, if the received code word is 1000010111, the recalculated value of r4 r3 r2 r1 is 1001, which indicates that bit position in error is 9, the decimal value of 1001.

This is decoded to identify the error position if any, and the error is corrected. In case of single error, the error is corrected.

If the syndrome is obtained as ‘0000’ no error occurred, else indicates errors and error correction takes place in case of single error by inverting corresponding bit. As shown in fig. 5 Encoded Data is formed with calculations of error bit position.

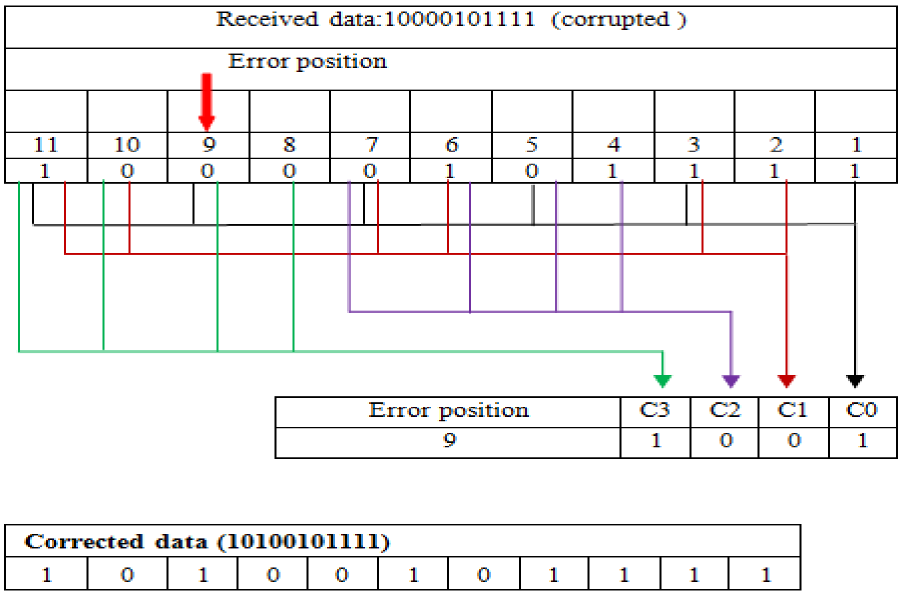


Fig.7 Error Detection And Correction Using Hamming code.

4. SIMULATION RESULT

The test bench waveforms for the transmitter and for the RSR and RHR of the receiver part are given in Fig.8.

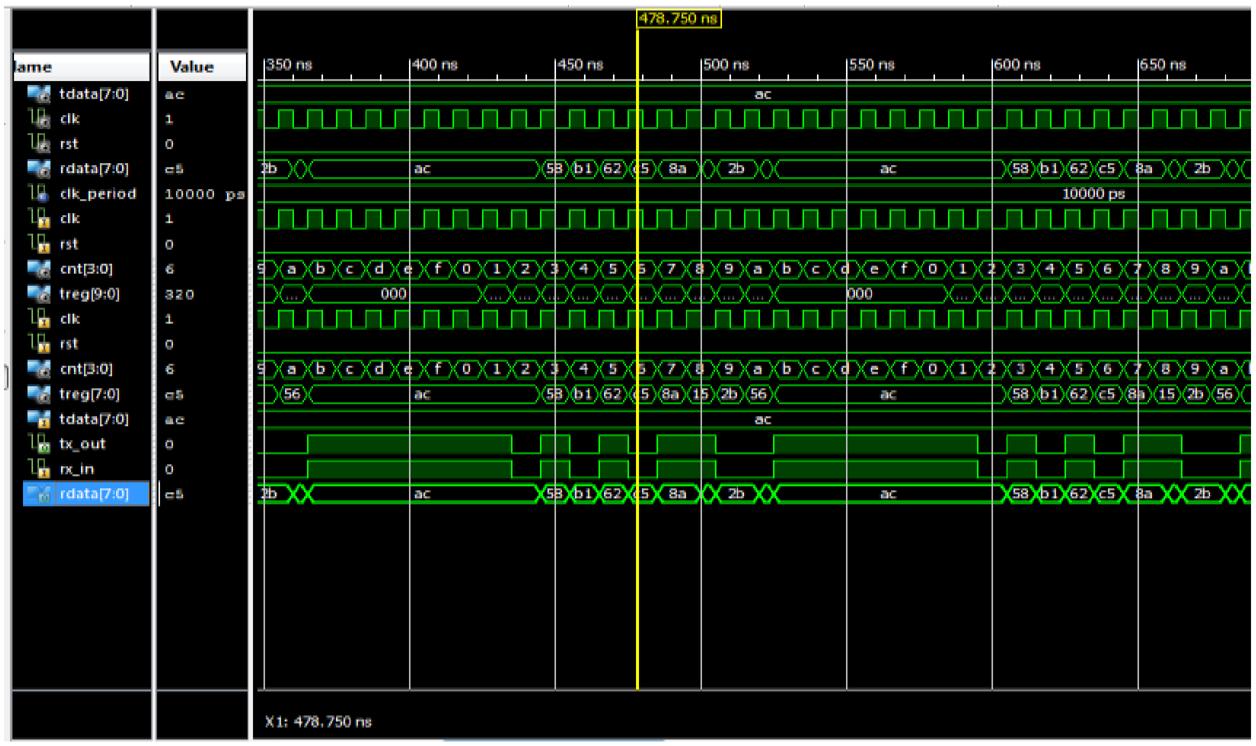


Fig.8 Final Result

Result:

Out of the 11 bits data the 4 bits "r1,r2,r4,r8" are transmitted. The frame is formed starting with start bit '0', hamming bits "0100" (r1,r2,r4,r8) and a '1' followed by two stop bits '11'. The RSR received above data with an error in bit position 9th bit. This error is corrected and the data output from RSR is given with all bits, except the data bits in the frame, replaced with zeros as "000101100000". The RHR extracts the four data bits "1011" concatenated with "0000" in position other than four hamming bits giving out a data byte.

5. CONCLUSION

FPGA Design of Hamming encoder and decoder with error detection and correction capabilities have been simulated and implemented in Xilinx Spartan 6 FPGA. The system has been designed using Verilog HDL and implemented on hardware using Xilinx12.2.

REFERENCES

1. Sindhuaja Muppalla, Koteswara Rao Vaddempudi, "A Novel VHDL Implementation of UART with Single Error Correction and Double Error Detection Capability", SPACES-2015, Dept of ECE, KL UNIVERSITY.
2. Himanshu Patel, Sanjay Trivedi, R. Neelkanthan, 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 onVLSI Design, pp. 819-823, Jan. 2007.
3. J. Norhuzaimin, and H.H. Maimun, "The design of high speed UART," Asia Pac. Conf. on Appl. Electromagnetics (APACE 2005), Johor, Malaysia, Dec. 2005.
4. Y. Fang, and X. Chen, "Design and simulation of UART serial communication module based on VHDL," 3rd Int'l Workshop on Intel. Sys. and App. (ISA 2011), Wuhan, China, May 2011.
5. S. Yu, L. Yi, W. Chen, and Z. Wen, "Implementation of a multichannel uart controller based on fifo technique and fpga," in Industrial Electronics and Applications, 2007. ICIEA 2007. 2nd IEEE Conference on, may 2007, pp. 2633 –2638.
6. Design and simulation of UART serial communication module based on VHDL - Fang Yi-Yuan, Chen Xue, IEEE Explore, may2011.
7. 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.
8. Prof. Rami Abielmona, "Project UART Design", for CEG 3150, Digital Systems-II, Fall 2004, November 24, 2004.