

Enhancement of Error Control Capability of Orthogonal Code Convolution for Digital Communication

¹ Vijaya K. Shendre, ² Rahul Nawkhare

¹ Mtech Scholar, Department of Electronics Engineering,
Wainganga College of Engineering & Management, Nagpur

² Assistant Professor, Department of Electronics Engineering,
Wainganga College of Engineering & Management, Nagpur

Abstract - Digital communication system, convolution coding is preferred for the channel coding as it facilitates a better error correction as comparison to block coding. Among other techniques such as Cyclic Redundancy and Solomon Codes; orthogonal coding is one of the codes which can detect errors and correct corrupted data in an efficient way. In this project we propose a high efficient combined error detection and correction technique based on the Orthogonal Codes Convolution, Closest Match, and vertical parity. This method will experimentally simulate using Xilinx software. The propose technique will detects 99.99% of the errors and corrects as predicted up to $(n/2-1)$ bits of errors in the received impaired n-bit code.

Keywords - Error detection and correction, Xilinx ISE9.2i, Orthogonal Code Convolution.

1. Introduction

In communication system, day by day, there is an increasing demand of network capacity due to the use of internet and real time transmission of voice and picture. To fulfill these requirements data transmission at high bit rates is essential. With the increase of data transmission and hence sources of noise and interference, engineers have been struggling with the demand for more efficient and reliable techniques for detecting and correcting error in received data.

Although several techniques and approaches have been proposed and applied in the last decade, data reliability in transmission is still problem. When data is stored, compressed, or communicated through a media such as cable or air sources of noise and other parameters such as EMI, crosstalk, and distance can considerably affect the reliability of these data. Error detection and correction techniques are therefore required. Some of those

techniques can only detect errors, such as the Cyclic Redundancy Check (CRC) [8-6]; others are designed to detect as well as correct errors, such as Solomon Codes [10, 5], Hamming Codes [4], and Orthogonal Codes Convolution (OCC) [7, 3]. However, the existing techniques cannot achieve high efficiency in error detection and correction as well as meet bandwidth requirements, especially with the increase in the quantity of data transmitted.

In a previous work, a technique (OCCM) that combines the OCC and a technique, Closest Match using Field Programmable Gate Array (FPGA), has led to the improvement of the detection capabilities of the OCC from 71.88% to 93.57% [2]. In this paper, we present an enhanced technique (OCCMP) that combines Orthogonal Codes Convolution with Closest Match and vertical parity. The results show that the proposed technique enhances both error detection and correction capabilities of Orthogonal Codes Convolution with a detection rate of 99.99%, and $(n/2-1)$ bits correction capability in the received impaired n-bit code.

1.1 Orthogonal Code

Orthogonal codes are binary valued, and they have equal number of 1's and 0's. An n-bit orthogonal code has $n/2$ 1's and $n/2$ 0's; i.e., there are $n/2$ positions where 1's and 0's differ [7,3]. Therefore, all orthogonal codes will generate zero parity bits. The concept is illustrated by 16-bit orthogonal codes as shown in Fig. 1. It has 16 orthogonal codes and 16 antipodal codes for a total of 16 bi-orthogonal codes. The inverse of orthogonal codes is antipodal code; they have the same properties.

[illegible]

Fig1: 16 BIT ORTHOGONAL CODES

[illegible]

Fig.2: 16 Bit Antipodal Codes

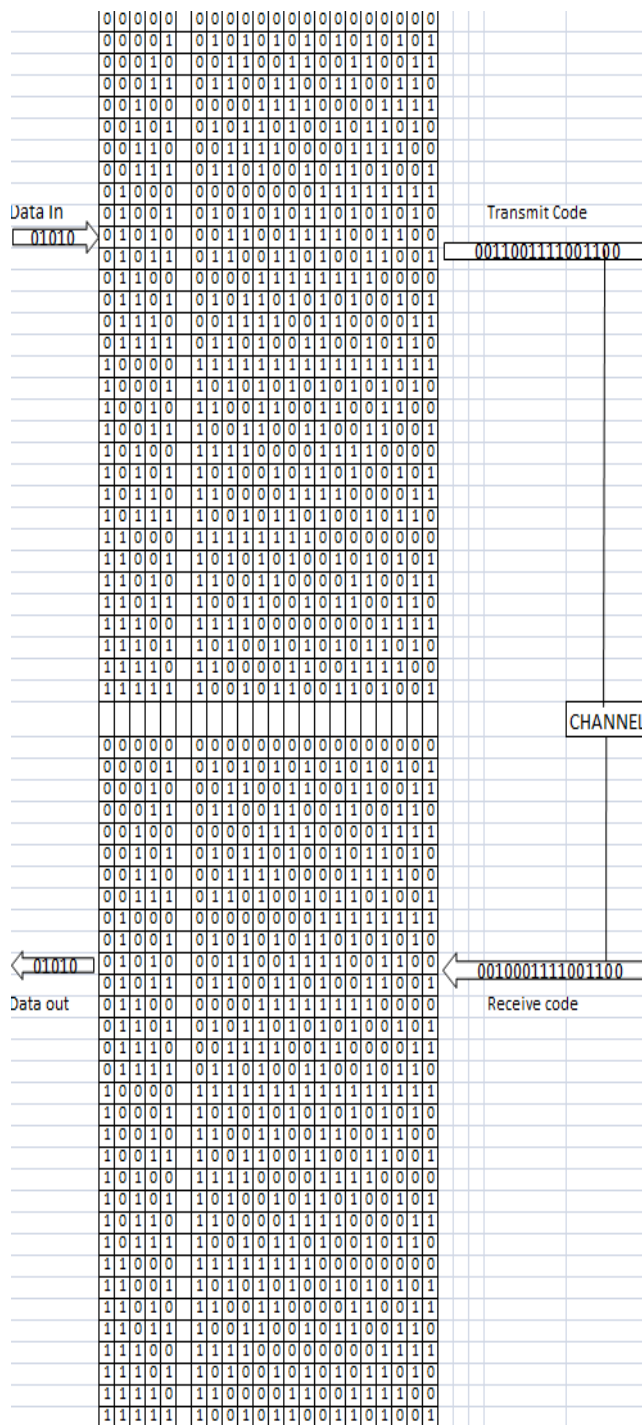


Fig.3: Illustration Of 16 Bit Occ Encoding And Decoding

Before transmission, a m-bit data set is mapped into a unique n-bit orthogonal code. For example, a 5-bit data set is represented by a unique 16-bit orthogonal code, which is transmitted without the parity bit. Based on code correlation, the received data is decoded. It can be done by

setting a threshold between two orthogonal codes. The following equation is used for setting threshold:

$$d_{th}=n/4 \quad (1)$$

Where n is the code length and d_{th} is the threshold, which is midway between two orthogonal codes. Therefore, for the 16-bit orthogonal code (Fig. 2), we have $d_{th} = 16/4 = 4$. This mechanism offers a decision process in error correction, where the incoming impaired orthogonal code is examined for correlation with the codes stored in a look-up table, for a possible match.

A counter is used to count the number of 1's in the resulting signal. For example, for 16-bit orthogonal code, the operation will lead to Thirty Two counter results. If one of the results is zero, it means there is no error. Otherwise, the code is corrupted. The corrected code is associated with the minimum count. If the minimum count is associated with one combination, the received and corrected code will be this combination. However, if the minimum count is associated with more than one combination of the orthogonal codes, it is not possible to correct the corrupted code.

Table I: Error correction capabilities of orthogonal codes.

Code length(n)	Correction Capability (nc)
8	1
16	3
32	7
64	15

Table II : Error detection capabilities of orthogonal codes.

Code length(n)	Detection Capability (nc)
8	71.88%
16	80.35%
32	86.01%
64	90.07%

2. Methodology

2.1 Design Methodology

To improve the detection and correction capabilities of the OCCM technique, the advance method, OCCMP(orthogonal code convolution with Closest Match and vertical parity) is proposed. This technique allows the transmission of several successive codes (16 codes in this paper) followed by their vertical parity byte. As the orthogonal codes has property of closure, the parity byte,

which is the exclusive sum (XOR) of Sixteen orthogonal codes, is also an orthogonal code, and thus, the same technique used to detect and correct errors of each code in the block of data and in the parity byte. After a byte is received, the system checks and corrects errors if the code is corrupted using the Closest Match technique. After checking and correcting the received 17 bytes by receiver, the new vertical parity byte is calculated corresponding to the 16 data bytes and compares the new byte to the received parity byte in order to detect and correct more errors.

Table III: Detection rate with closest match techniques

Code length(n)	Detection Capability (nc)
8	93.57%
16	99.95%
32	99.99%
64	99.99%

2.2. Transmitter

The transmitter consists of four blocks: an encoder, a memory, a vertical parity generator and a shift register. A 5-bit data is encoded by an encoder to 16-bit orthogonal code using the look-up table (Fig. 2). After encoding, that 16 bit orthogonal code is stored in a 17 by 16 memory and a parity byte is calculated representing the columns; for each column, a parity bit is calculated., a shift register is used to convert these parallel 272 bits in serial data and then that data is transmitted in serial form.

Fig. 4 shows the block diagram of the transmitter. The input signal is parallel 5-bit data and the output is serial codes that take 272 periods of clock to transmit all the information.

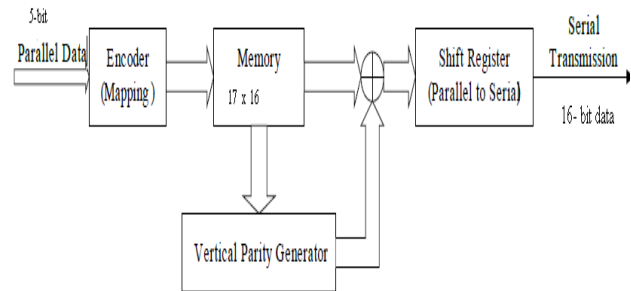


Fig.4: Block Diagram of Transmitter

2.3 Receiver

As shown in Fig. 5, when a block of 16-bit code is received by receiver, it first converts it into parallel form by serial to parallel converter. XOR in the bit of data is

carried out to detect the error and the Closest Match technique is used to correct a one bit error or multi-bit errors in the received code by comparing that received code with each code in the look-up table. By using counter the number of 1's in the resulting signal is calculated. If that number is any number, except zero, shows an error or errors in the received code and the closest match code is used to correct it if only one bit error is detected. If there is an error in more than one bit, the received code is retransmitted. The indication for the retransmission is given by the flag. After matching the received sixteen codes, a new parity byte (Pc) is calculated using these matching codes and "XOR" performance is conducted between the new and corrected received parity byte (Pr). The 1's in the resultant signal show the column with an odd number of error or errors. The location of the error or errors is easily found by using flag which will become on for the block of corrupted code. Decoding is performed after correcting the error or errors, and these 256 corrected bits are decoded to Sixteen 5-bit data in order.

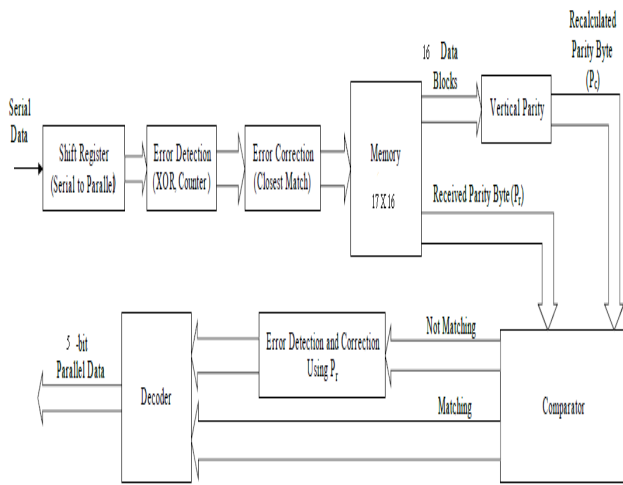


Fig:5 Block Diagram of Receiver

3. Implementation and Results

The system has been going to implement using a the software Xilinx ISE 9.2i .For error detection, in a previous work [2], it was shown that if an orthogonal code changes to another orthogonal code, it is not possible to detect the problem. However, with OCCMP, when one block in the package changes to another orthogonal code, the error can be detected using the parity byte. Let's assume that the number of blocks in the memory is k (including the parity byte). When more than one block in this memory change to other orthogonal codes and there is an even number of errors in one column, the errors cannot be detected.

The number of combinations that cannot be detected is given as follows:

$$n_i = \begin{cases} C_k^i \cdot (C_{2n-1}^1)^{\frac{i}{2}} & \text{for } i \text{ even} \\ C_k^i \cdot C_{2n-1}^1 \cdot (C_{2n-1}^2)^{\frac{i-1}{2}} & \text{otherwise} \end{cases}$$

where $i=2, 3, \dots, k$ is the number of corrupted blocks, and n_i is the amount of combinations that cannot be detected.

The total number of combinations that cannot be detected for k blocks of n -bit code is N given by:

$$N = \sum_{i=2}^k n_i$$

For 17 block of 16 bit code

$$N = \sum_{i=2}^{17} n_i$$

$$= 17C2 \cdot 31C1 + 17C4 \cdot (31C1)^2 + 17C6 \cdot (31C1)^3 + 17C8 \cdot (31C1)^4 + 17C10 \cdot (31C1)^5 + 17C12 \cdot (31C1)^6 + 17C14 \cdot (31C1)^7 + 17C16 \cdot (31C1)^8 + 17C3 \cdot 31C1 \cdot 30C1 + 17C5 \cdot 31C1 \cdot (30C1)^2 + 17C7 \cdot 31C1 \cdot (30C1)^3 + 17C9 \cdot 31C1 \cdot (30C1)^4 + 17C11 \cdot 31C1 \cdot (30C1)^5 + 17C13 \cdot 31C1 \cdot (30C1)^6 + 17C15 \cdot 31C1 \cdot (31C1)^7 + 17C17 \cdot 31C1 \cdot (30C1)^8$$

$$= 215556182.9 \times 10^6$$

The detection rate for k block of n -bit code is

$$\frac{2^{k \cdot n} - N}{2^{k \cdot n}} \times 100\%$$

Detection rate for OCCMP-16

$$((2^{17 \cdot 16} - 215556182.9 \times 10^6) / 2^{17 \cdot 16}) \times 100\%$$

$$= 99.99\%$$

4. Conclusion

In this work, we are going to develop and implement a real-time high-efficiency technique to detect and correct errors in digital communication. The experimental results will show that the propose technique can detect 99.99% of errors and correct up to $(n/2-1)$ -bit of errors for n -bit orthogonal codes.

Table IV: Comparison of Detection and Correction Capability of different Method

Technique	Detection Rate	Correction Capability nc
OOC-8	71.88%	1
OCCM-8	93.57%	1
CRC-32	99.99%	0
OCCMP-8	99.99%	3
OCCMP-16	99.99%	7

Table V: Comparison Of Correction Capabilities Between Occ, Occm And Occmp.

Code length(Bit)	Techniques		
	OCC	OCCM	OCCMP
8	1	1	3
16	3	3	7
32	7	7	15
64	15	15	31
n	$(n/4)-1$	$(n/4)-1$	$(n/2)-1$

References

- [1] Mukesh Gholase¹, L.P.Thakare², Dr. A.Y. Deshmukh³ International Journal of Engineering Sciences Paradigms and Researches, Vol. 05, Issue 01, June 2013 ISSN (Online): 2319-6564.
- [2] N. Kaabouch, A. Dhirde, and S. Faruque, "Improvement of the Orthogonal Code Convolution capabilities using FPGA implementation", *IEEE Electro/Information Technology Proceedings*, pp. 380-384, 2007.
- [3] S. Faruque, N. Kaabouch, and A. Dhirde, "Forward error control coding based on orthogonal code and its implementation using FPGA", *Wireless and Optical Communication Proceedings*, Paper # 565-630, ACTA Press, June 2007.
- [4] U. K. Kumar, and B. S. Umashankar, "Improved hamming code for error detection and correction", *2007 2nd International Symposium on Wireless Pervasive Computing*, pp. 498-500.
- [5] Z. Cai, J. Hao, S. Sun, and F. P. Chin, "A high-speed reed-solomon decoder for correction of both errors and erasures", *2006 IEEE International Symposium on Circuits and Systems*, pp. 281-284.
- [6] S. Shukla, N.W. Bergmann, "Single bit error correction implementation in CRC-16 on FPGA", in *Conf. Rec. 2004 IEEE Int. Conf. on Field-Programmable Technology*, pp. 319-322.
- [7] S. Faruque, "Error Control Coding based on Orthogonal Codes", *Wireless Proceedings*, Vol. 2, pp. 608-615, 2004.
- [8] T. Baicheva, S. Dodunekov, and P. Kazakov, "Undetected error probability performance of cyclic redundancy-check codes of 16-bit redundancy", *IEEE Proceedings. Communications*, Vol. 147, No. 5, pp. 253-256, Oct. 2000.
- [9] A. Hakanin, H. Delic, and S. Sarin, "Two dimensional CRC for efficient transmission of ATM Cells over CDMA", *IEEE Communications Letters*, Vol. 4, No. 4, pp. 131-133, April 2000.
- [10] V. Stylianakis, S. Toptchiyski, "A Reed-Solomon coding/decoding structure for an ADS modem", in *Conf. Rec. 1999 IEEE Int. Conf. on Electronics, Circuits and Systems*, pp. 473 – 476.



Vijaya K. Shendre received BE degree in Electronics Engineering from RTMNU in 2010. She is Currently Pursuing her M.tech from WCEM, Nagpur in the stream of Electronics Engineering from RTMNU. Working as an researcher her research interest includes technological support to learning and work processes, VLSI, VHDL, Digital Communication as well as Digital Signals.

Rahul Nawkhare Currently working as an Asst. prof. in Department of Electronics Engineering at WCEM, Nagpur. His research interest include Image Processing, Digital Communication etc.