Enhancement of Error Detection and Correction Capability Using Orthogonal Code Convolution

[1,]Mukesh Gholase , [2,]L.P.Thakare, [3,]Dr. A.Y. Deshmukh
[1,2,3,]*Dept. Of Electronics Engineering, G.H.Raisoni College of Engineering. Nagpur, India.*

Abstract

In this paper, The Orthogonal Codes has been developed and realized by means of Field Programmable Gate Array (FPGA). The proposed techniques map a k-bit data block into an n-bit Orthogonal code block (n>k) and transmit the coded block across the channel. Construction of orthogonal coded modulation schemes are realised by means of FPGA. The result shows that the proposed technique enhances both error detection and correction capabilities of Orthogonal Codes Convolution with a detection rate of 99.99%.

Index Terms *- Error detection and correction, FPGA, Orthogonal Code Antipodal Code.*

I. INTRODUCTION

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) [1-3]; others are designed to detect as well as correct errors, such as Solomon Codes [4,5], Hamming Codes [6], and Orthogonal Codes Convolution (OCC) [7,8].

However, the existing techniques are not able to achieve high efficiency and to meet bandwidth requirements especially with the increase in the quantity of data transmitted. Orthogonal Code is one of the codes that can detect errors and correct corrupted data. The objective of this paper is to enhance the error control capability of orthogonal code by means of Field Programmable Gate Array (FPGA) implementation.

II. ORTOGONAL CODES

Orthogonal codes are binary valued, and they have equal number of 1's and n/2 0's; i.e., there are n/2 positions where 1's and 0's differ [7,8]. 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 32-bi-orthogonal codes. Antipodal codes are just the inverse of orthogonal codes; they have the same properties. A notable distinction in this method is that the transmitter does not have to send the parity bit for the code, since it is known to be always zero [8,9]. Therefore, if there is a transmission error, the receiver may be able to detect it by generating a parity bit at the receiving end.Before transmission, a k-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.When received, the data are decoded based on code correlation. It can be done by setting a threshold between two orthogonal codes. This is set by the following equation

$$d_{th} = \frac{n}{4} \qquad (1)$$

Where n is the code length and d_{th} is the threshold, which is midway between two orthogonal codes. Therefore, for the16- 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. The acceptance criterion for a valid code is that an n-bit comparison must yield a good cross correlation value; otherwise, a false detection will occur. This is government by the following correlation process, where a pair of n-bit codes $x_1x_2\ldots x_n$ and $y_1y_2\ldots y_n$ is compared to yield,

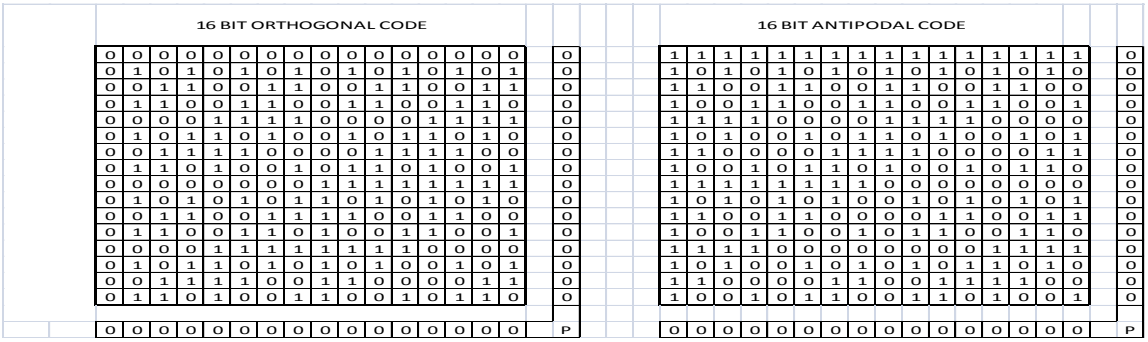


Fig.1. Illustrations of16-bit orthogonal codes.

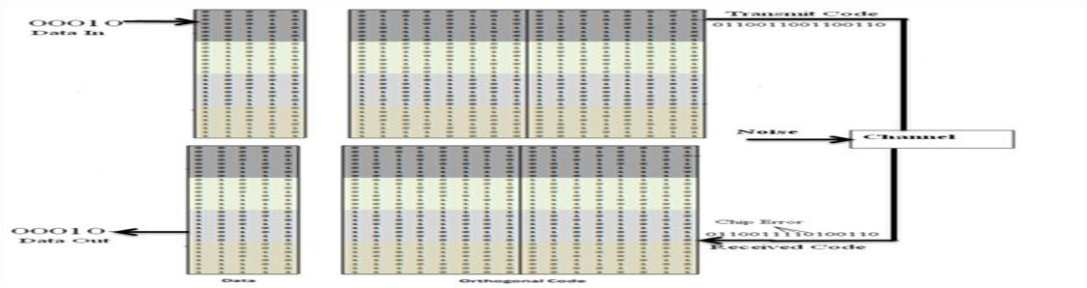


Fig.2. Illustration of OCC Enconding and Decoding

$$R(x, y) = \sum_{i=1}^{n} x_i \, y_i \leq d_{th} \qquad (2)$$

Where R(x, y) is the cross correlation function, d_{th} is the threshold defined in (1). Since the threshold (d_{th}) is between two valid codes, an additional 1-bit offset is added to (2) for reliable detection. The number of errors that can be corrected by means of this process can be estimated by combining (1) and (2), yielding.

$$n_c = R(x,y) - 1 \leq \frac{n}{4} - 1 \qquad (3)$$

In this equation, n_c is the number of error that can be corrected by means of an n-bit orthogonal code. For example, a single error in 8-bit orthogonal code can be corrected. Similarly, 3-bit errors in 16-bit orthogonal code can be corrected, and so on Table 1 give the error correction capabilities and the detection rates, respectively, for four lengths of orthogonal codes.

In a previous work, these detection rates were improved. This method performs XOR operation between the received code and each code in the look-up table that contains all the codes.

TABLE I

Orthogonal Codes and the Corresponding Error Correction Capabilities.

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

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 in zero, it means there is no error. Otherwise, the code is corrupted. The corrected code is associated with 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.

III. METHODOLOGY

An orthogonal code is having equal number of 1's and 0's so it will generate zero parity all time except that code received is erroneous. We need not to send any extra parity bit while transmitting data. In case if, after the transmission the data received is erroneous then it will generate parity error. In this process of detecting and correcting the errors code received is split into two equal parts. Each part will be checked for parity bit, if generated parity is zero then code is error free and if one then the received code is considered to be the erroneous. By orthogonal code method we can detect the part of the incoming or received data that is containing the errors and along with the correction of code we can also improve trans-reception system by using effective means to reduce the error in that particular area and over all reception system will become more effective.

TABLE II
Orthogonal Codes and the Corresponding Detection rate.

Code length *(n)*	Detection Rate (%)
8	93.57
16	99.95
32	99.99
64	100.00

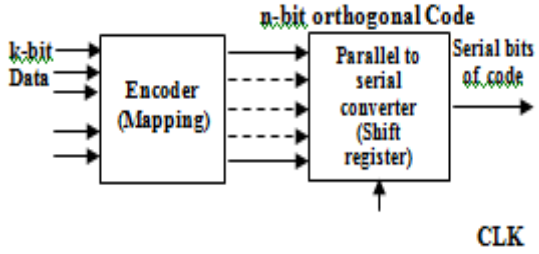


Fig.3 Block diagram of 5-bit transmitter

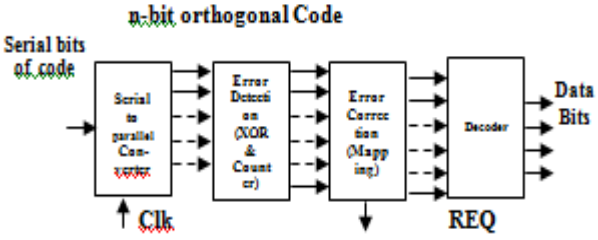


Fig.4. Block Diagram of Receiver

3.1 Transmitter

The Transmitter consists of two blocks (a) encoder (b) shift register

(a) The encoder encodes a k-bit data set to $n=2^{k-1}$ bits of the orthogonal code.

(b) The Shift Register transforms this code to a serial data in order to be transmitted as shown in Fig.3.

For example, 5-bit data is encoded to 16-bit (2^4) orthogonal code according to the lookup table shown in Fig.2. The generated orthogonal code is then transmitted serially using a shift register with the rising edge of the clock.

3.2 Receiver

The received code is processed through the sequential steps, as shown in Fig.4. The incoming serial bits are converted into n-bit parallel codes. The received code is compared with all the codes in the lookup table for error detection. This is done by counting the number of ones in the signal resulting from 'XOR' operation between the received code and each combination of the orthogonal codes in the lookup table. A bit signal and also searches for the minimum count. However a value rather than zero shows an error in the received code. The orthogonal code in the lookup table which is associated with the minimum count is the closest match for the corrupted received code. The matched orthogonal code in the lookup table is the corrected code, which is then decoded to k-bit data. The receiver is able to correct up to (n/4-1) bits in the received impaired code. However, if the minimum count is associated with more than one combination of orthogonal code then a signal, REQ, goes high

IV. IMPLEMENTATION AND RESULTS

A Spartan-3 hardware board and ISE. Xillinx software has been used for code testing. The simulation has been performed using modelSim XE software. The Simulation Result were verified for most of the combination 16-bit Orthogonal Code, ISE Xillinx Software have use for synthetics.

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

$$(2^n - 2^k)/2^n\%$$

This gives for 8-bit detection

$$\frac{2^8 - 2^4}{2^8} \times 100 \qquad \frac{128 - 8}{128} \times 100 = 93.75$$

This gives for 16-bit detection

$$\frac{2^{16} - 2^5}{2^{16}} \times 100 \qquad \frac{655536 - 32}{65536} \times 100 = 99.95$$

This gives for 32-bit detection

$$\frac{2^{32} - 2^6}{2^{32}} \times 100 \qquad \frac{4294967296 - 64}{4294967296} \times 100 = 99.99$$

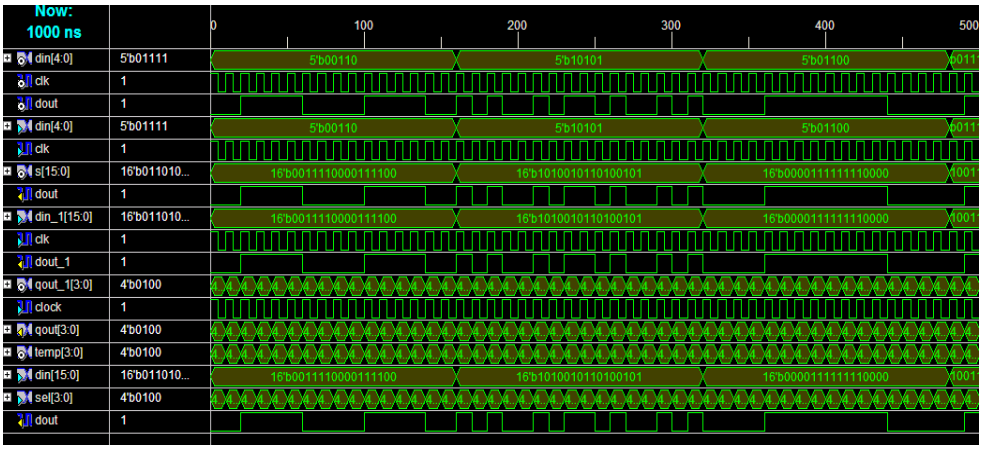
This gives for 64-bit detection

$$\frac{2^{64} - 2^7}{2^{64}} \times 100 \qquad \frac{1.84 \times 10^{19} - 128}{1.84 \times 10^{19}} \times 100$$

$$= 1 \times 100 = 100$$

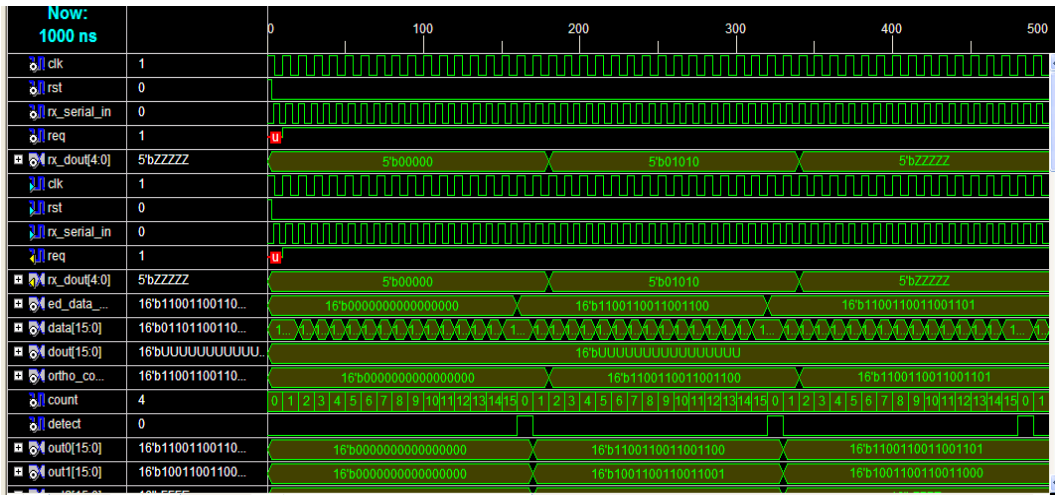
4.1 Transmitter

Fig. 5 shows an example of the results of the transmitter simulation corresponding to the input data value 00010 labelled as 'data'. This data has been encoded to the associated orthogonal code "0110011001100110" labelled 'ortho'. The signal 'EN' is used to enable the transmission of the serial bits ' txcode' of the orthogonal code with every rising edge of the clock.



4.2 Receiver

Upon reception, the incoming serial data is converted into 16-bit parallel code 'rxcode'. Counter is used to count the number of 1st after XOR operation between the received code and all combinations of orthogonal code in the lookup table. Fig.6, the received code is rxcode="0110011001100110", count='0' and hence the received code is not corrupted. The code is then decoded to the corresponding final data 00010.



The results of the simulation show that for a k-bit data, the corresponding n-bit orthogonal code is able to detect any faulty combination other than the combinations of orthogonal code in the lookup table.

TABLE III

Correction Capabilities Between Orthogonal Codes Convolution

Codes length (*Bit*)	**Techniques** OCC
8	1
16	3
32	7
64	15
N	(n/4-1)

V. CONCLUSION

FPGA implementation of orthogonal code convolution is presented to ensure the efficient digital communication. This work involved the implementation of the transmitter and receiver using VHDL to detect and correct the errors. A fully synthesizable HDL code was written to ensure that the design was feasible. This orthogonal code implementation has improved the error detection up to 99.9% for 16-bit coding. It is noted that with this method, the transmitter does not have to send the parity bit since the parity bit is known to be always zero. Therefore, if there is a transmission error, the receiver will be able to detect it by generating a parity bit at the receiving end. Finally, this work has the future scope of further improvement in orthogonal coding for large digital data processing.

REFERENCES

[1] 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.

[2] 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, PP 565-630, ACTA Press, June 2007.

[3] 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.

[4] 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.

[5] S. Faruque, " Error Control Coding based on Orthogonal Codes", Wireless Proceedings, Vol. 2, pp. 608-615, 2004.

[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] 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.

[8] A. Hokanin, 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.

[9] 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.