International Journal of Education and Science Research
Review E-ISSN 2348-6457
Volume-3, Issue-2 April- 2016
www.ijesrr.org Email- editor@ijesrr.org

VLSI Implementation of Error Detection and Correction Using Orthogonal Code Convolution

Anjali Sinha, Kritika Pandey, Savita, Rupali Singh
SRM University, NCR Campus
Ghaziabad

ABSTRACT:
When data is transmitted from one end to another through a channel, there are possibilities of addition of noise due to which our data gets corrupted. So, we need to detect and correct this corrupted data as we are aimed at getting the same data at the receiver. In this paper, we are using orthogonal code to detect and correct the corrupted data. We have designed transmitter and receiver. In this paper, FPGA (Field PROGAMMABLE Gate Array) is used for enhancing the error detection and correction capabilities of orthogonal code. FPGA is used basically for the purpose of testing the output.

KEYWORDS: FPGA, Orthogonal code convolution.

INTRODUCTION:
Error detection and correction techniques are the techniques which enable reliable delivery of digital data over unreliable communication channels. Error detection is the detection of errors caused by noise or other impairments during transmission from transmitter to the receiver. Error correction is the reconstruction of original, error-free data. When coding is used, the efficiency of channel becomes more as compared to the case when code is not used. There are various types of error detecting and correcting codes. Error detecting codes are parity check, cyclic redundancy check and checksums in parity check method, an extra parity bit is included with the binary data to make the total number of 1s either odd or even. This results in two methods-even parity methods and odd parity method. Checksum method is used to detect double errors and pinpoint error bits.
Cyclic redundancy check-it is an error detecting code which was invented by W. Wesley Peterson [1]. It is used in digital networks and storage devices to detect sudden changes in raw data. CRC'S are so called because the check (data verification) value is a redundancy i.e.it expands the message without adding information.
Error correcting codes are hamming code, reed Solomon, orthogonal code, etc.
Hamming code-these are linear error correcting codes which was invented by Richard W. Hamming in 1950 [2]. Hamming codes can detect upto two bits of error or correct one bit of error [3]. In contrast, simple parity code can't correct errors and can detect only an odd number of bits. In this method, we need to add some additional bits.
Reed Solomon code-this code is a non-binary cyclic error correcting code which was invented by Irving S. Reed and Gustave Solomon [4]. This method is able to detect and correct multiple symbol errors. By adding t additional bits, a reed Solomon code can detect upto t error bits or correct upto (t/2) error bits.

ORTHOGONAL CODES:
Orthogonal Codes are a kind of error correcting code. It is a binary valued code. There are equal number of 0s and 1s.So n bit orthogonal code have n/2 number of 0s and 1s [5]. So all orthogonal codes generate zero parity bit and when there is any error then parity will be one.

Fig1: -an 8-bit orthogonal code has 8 orthogonal codes and 8 antipodal codes for a total of 16 bi-orthogonal codes.

The corresponding orthogonal code for a k-bit data set is 2^(k-1) [6]. In this transmitter doesn't have to send any parity bit since there is no error at source. But in between the channel if some error occurs then receiver have to detect and correct that error by sending a parity bit. k bit data is mapped into an unique n bit e.g. 8 bit data can be represented into 16 bit similarly 16 bit into 32 bit.

Table I:-Orthogonal codes and corresponding chip error control

N	T
8	1
16	3
32	7
64	15
128	31

PROPOSED WORK:

TRANSMITTER:

The transmitter consists of two blocks: an encoder and a shift register. When a k-bit data is transmitted, the encoder encodes this k-bit data set to n=2^(k-1) bits of the orthogonal code. Then this orthogonal code is passed to a parallel-in-serial out shift register. The PISO transforms this code to a serial data in order to transmit it through the channel. Serial bits of data are sent through the channel to prevent the whole set of data bits from getting corrupted. So, at the output serial bits of data is obtained.

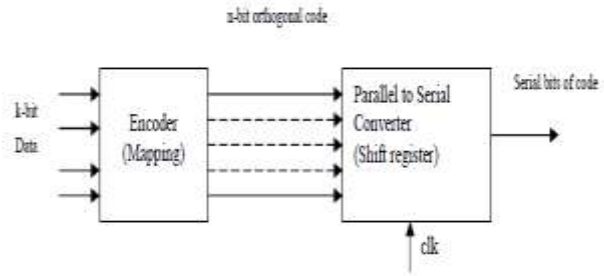

FIG.2-BLOCK DIAGRAM OF TRANSMITTER

RECEIVER:

The receiver consists of SIPO shift register, error detection circuit, error correction circuit and a decoder. The incoming serial data is converted into parallel data with the rising edge of clock. The error detection circuit consists of XOR gate and counter.XOR gate performs its operation on the received code and all combinations of orthogonal code. Counter counts the number of 1's after XOR operation between the received code and all combinations of orthogonal code. A value of zero indicates that there is no corrupted data. The counter also searches for the minimum count. The orthogonal code which is associated with the minimum count is the closest match for the corrupted received code. If the value of minimum count is associated with more than one orthogonal code, then it is not possible to detect which is the correct code. So, a REQ signal goes high in order to request for a retransmission. After this, the matched orthogonal code is the corrected code, which is then decoded to k-bit data. The receiver corrects up to (n/4)-1 bits in the received impaired code.

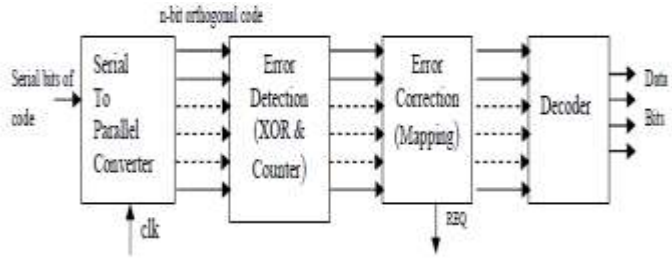

FIG.3-BLOCK DIAGRAM OF RECEIVER

DESIGN METHODOLOGY:

1. The method used in designing this project is based on comparison between the received code and all the combinations of orthogonal code.
2. The received corrupted code is XOR operated with all the combinations of orthogonal codes.
3. The number of 1's is counted in the resultant XOR output.
4. The code which is associated with the minimum no. of 1's is the original orthogonal code.
5. Next, codes are written for encoder, PISO, SIPO, XOR gate, counter, decoder.
6. All the codes are written in verilog.
7. Then, codes for each one are simulated.
8. After this, resulting waveforms are obtained.
9. Lastly, our code is dumped into FPGA kit to check the proper working.

IMPLEMENTATION AND RESULTS:

Code testing/implementation was done on Sparatan-3 hardware board and ISE Xillinx software. ModelSim XE software has been used for simulation of results. For most of 8-bit and some 16-bit orthogonal code simulation was done. Software simulation results:-

Transmitter: For the simulation of transmitter value "0110" was the input labeled as "data" and it was encoded to the associated orthogonal code "00111100" termed as "ortho". For every rising edge of the clock signal 'EN' was used to enable transmission of the serial bits 'txcode'.

Receiver: The serial data received was converted into 8-bit parallel code 'rxcode'. After applying XOR's operation on received code and every combination of the orthogonal code in the lookup table number of 1's are counted by COUNTER. It gives minimum count of 1's. The closest match for the received code was the orthogonal code 'ortho' which was associated with minimum count and was converted/decoded to final data 'data'. Three case:

Match of the received code in lookup table. 'rxcode'="00111100", count='0' and received code is not corrupted and is then decoded to final data "0110".

There was no match in the lookup table of the received code. 'rxcode'="00110100", minimum count is 1 hence an error. The closest match for the received code given by the minimum count is "00111100", and is decoded into final data "0110". Hence the single bit error was detected and rectified by the receiver.

More than one possibility of closest match in the lookup table 'rxcode'="00110000". It was not possible to determine the closest match from the lookup table since the minimum count was associated with more than one orthogonal code. Now a request for retransmission goes to high the label 'REQ'.

Results: From the above simulation deduction can be made that for k-bit data the corresponding n-bit orthogonal code is able to detect any faulty combination other than the combinations in the lookup table. There are 2^k number of combinations and hence percentage of detection is ((2^n)-(2^k))/2^n %. The number of clock cycle required to process the received code was (2n+2) by the system to correct upto (n/4)-1 bit error. Example encoding a 4-bit data to 8-bit orthogonal code, will have 2^4=16 combinations of orthogonal code. Receiver will not be able to detect error in codes which are one of the combinations of the orthogonal code of out of 256 possible combinations of the 8-bit received code. Hence, detection percentage

of 8-bit code (2^8-2^4)/2^8=93.57 % , also able to correct single bit error. Likewise, percentage detection for 16-bit orthogonal code = 99.99% with 3-bit of error detection.

TABLE II:-Summary table of 8-bit, 16-bit and N-bit orthogonal code:

	Number of combinations	Nf Number of undetected combinations	Detection Rate
8-bit orthogonal codes	256	Nf = 16	93.57%
16-bit orthogonal codes	65535	Nf = 32	99.95%
N-bit orthogonal code	2^N	Nf = 2N	(2N-Nf)/ 2N

CONCLUSION:

The implementation of the orthogonal code is proven to elevate the likelihood of error detection from 50% to 93% and 99.9% for 8-bit and 16-bit respectively. Also for any digital transmission system the technique can be applied to any encoding system. Correction capabilities of orthogonal code and parallel implementation to speed up the data processing are needed to be further improved.

SIMULATION RESULTS:

FIG. 4-SIMULATION RESULT OF ENCODER:

FIG. 5-SIMULATION RESULT OF PARALLEL IN SERIAL OUT (PISO) SHIFT REGISTER:

FIG.6-SIMULATION RESULT OF TRANSMITTER:

FIG. 7-RTL SCHEMATIC OF TRANSMITTER

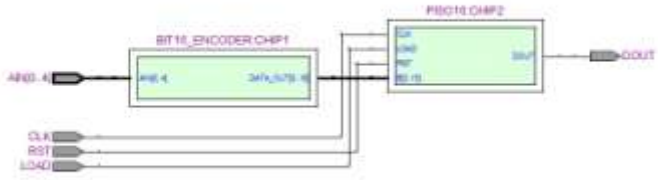

FIG. 8-SIMULATION RESULT OF SERIAL IN PARALLEL OUT (SIPO) SHIFT REGISTER:

FIG. 9-SIMULATION RESULT OF ERROR DETECTION AND CORRECTION:

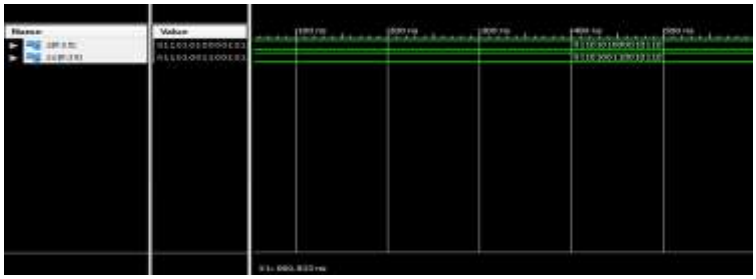

FIG. 10-SIMULATION RESULT OF DECODER:

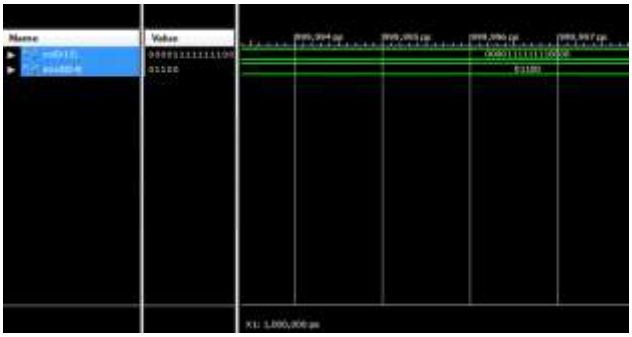

FIG. 11-SIMULATION RESULT OF RECEIVER:

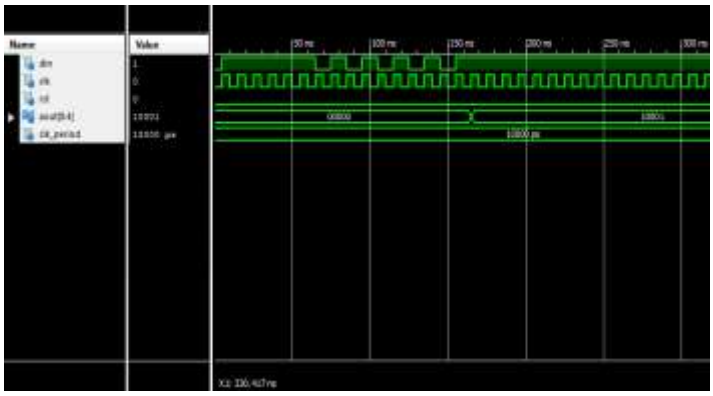

FIG. 12-RTL SCHEMATIC OF RECEIVER

TABLE III:-FPGA OUTPUT OF ORTHOGONAL CODE

PARAMETERS	TRANSMITTER	RECEIVER
AREA	275252 KB	329780 KB
DELAY	5.531 NS	39.022 NS
POWER	0.038 WATTS	0.038 WATTS

TABLE IV:-FPGA OUTPUT OF HAMMING CODE

PARAMETERS	TRANSMITTER	RECEIVER
AREA	365432 KB	498321 KB
DELAY	6.894 NS	45.987 NS
POWER	0.067 WATTS	0.067 WATTS

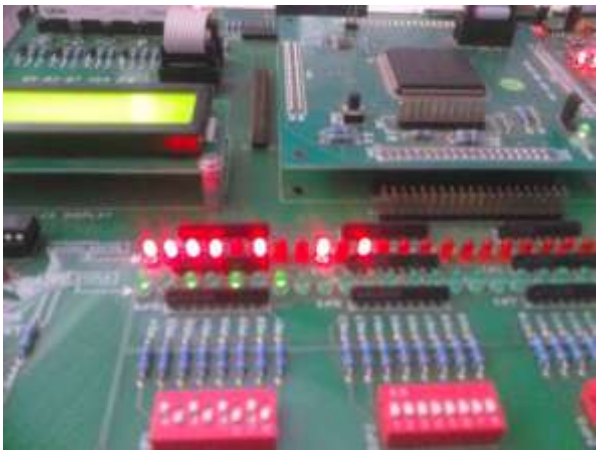

FIG.13-IMPLEMENTATION RESULTS ON FPGA KIT

REFERENCES:

1. Baicheva, T., S. Dodunekov, and P. Kazakov, "Undetected error probability performance of cyclic redundancy- check codes of 16-bit redundancy," IEEE Proc. Comms., Vol. 147, No. 5, Oct. 2000, pp. 253-256.
2. Moon, Todd K. (2005). Error Correction Coding. New Jersey: John Wiley & Sons. ISBN 978-0-471-64800-0.
3. D.K. Bhattacharryya, S. Nandi. "An efficient class of SEC-DED-AUED codes". 1997 International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN '97)
4. Reed, Irving S.; Solomon, Gustave (1960)," Polynomial codes over certain finite fields", Journal of the Society for Industrial and Applied Mathematics(SIAM).
5. Saleh Faruque, "Error Control Coding Based on Orthogonal Codes", Wireless Proceedings, Vol. 2, pp. 608-615, 2004.
6. S. Faruque, A. Dhirde, N. Kaabouch "Forward error control coding based on orthogonal code and its implementation using FPGA", 7th IASTED International Conference, Montreal, Quebec, Canada, May 30 – June 1, 2007, Unpublished.