Indian Journal of Science and Technology, Vol 9(30), DOI: 10.17485/ijst/2016/v9i30/98724, August 2016
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

FPGA Implementation of High Speed Error Detection and Correction of Orthogonal Codes using Segmentation Method

T. Narendra Babu[1*], Fazal Noorbasha[1], M. Harita[2], N. Tejashree[2] and K. Vamsi Krishna[2]

[1]Department of ECE, K L University, Vaddeswaram, Guntur - 522502, Andra Pradesh, India; tatininarednra@kluniversity.in, fazalnoorbasha@kluniversity.in
[2]Department of ECM, K L University, Vaddeswaram, Guntur - 522502, Andra Pradesh, India; marni.harita@gmail.com, tejashreeneelam@gmail.com, vamsikrishna19950@gmail.com

Abstract

Background: Our main objective is to improve the error detection and correction capability using orthogonal codes with high security and speed. **Statistical Analysis**: In order to achieve high speed and security for error detection and correction, we have used cryptography technique. The concept of segmentation is specifically used, as it gives highly secured signal and also reduces the time complexity. Previous study incorporates mapping technique for error detection and correction. Our proposed methodology uses two decoders in place of mapping at the receiver end. This eases the performance and decreases the clock pulses. **Findings:** The proposed technique will send the k-bit data to encoders and it gets converted into orthogonal codes. The data is then encrypted using encryptor which consists of LFSR. The data is then sent to the receiver and then original data is retrieved using the decoders at the receiver. The multiple bit error correction can be done up to (n/4-1) bits. Here we have compared the delays for 4-bit, 5-bit, 6-bit, 7-bit, 8-bit data. After comparing our technique with the previous study we have found out that the delay time is gradually reduced. Our proposed work is done in n/2+1 comparison, where n represents the bit length of orthogonal codes. Hence this technique achieves 100% multiple bit error detection and error correction rate in the received signal. This technique is simulated in Xilinx software and implement using Field Programmable Gate Array (FPGA). **Application/Improvements:** This technique can be used for efficient transmission of data in the networks. There is also a wide scope for improvement to limit the bandwidth.

Keywords: Comparator, Error Detection and Correction, FPGA, LFSR, Orthogonal Codes

1. Introduction

Information in the signal is transferred from source to destination. If there is some disturbance in the signal or if the data is corrupted and if there is any unwanted noise in the signal then errors might occur and the data is not transferred properly. For this error detection and correction is required to transfer the original data. The existing technique such as Autocorrelation, Solomon code, code convolution doesn't meet high efficiency and band width. To meet this high efficiency in error detection and error correction we have used orthogonal codes with segmentation technique to increase the efficiency of the data. Error detection is the recognition of errors caused by many noises or any other disturbances during transfer of the data from source to the destination end.

The purpose of error detection is to make the probability of receiving error free data while transmitting an error data.

Here we are using the concept of cryptography to implement this Process. A cryptographic system alludes to a suite of calculations expected to execute a specific type of Encryption and Decryption. The term cipher is regularly used to allude to a couple of calculations, one for encryption and one for decoding. Normally, a cryptosystem comprises of three calculations: one for key era, one

*Author for correspondence

for encryption, and one for decoding. It uses encoders at the transmitter for encryption of data and the decoders at receiver for decryption of the original data sent by the sender.

The Existing technique present is 'Mapping'. Mapping is done in both transmitter and receiver block. This technique determines the comparison between the a-bit data and its corresponding b-bit data. In transmitter block mapping is done in encoder block, which takes a-bit data as input and 2^{a-1}bit data is obtained as output which is also called as orthogonal codes. In receiver block mapping is done in decoder block in order to obtain the a-bit data. Here we are proposing a new technique which is known as segmentation. This technique uses two decoders in place of mapping in which the time complexity is reduced. The time delays can be reduced and the efficient output can be obtained. These sources gave us a basic idea and represent the related work of our proposed concept.

In[1] proposed a technique to detect and correct the errors using mapping technique on high security for cryptographic system. Time complexity was been reduced.

In[2] proposed an efficient algorithm on encoding for (n, k) binary cyclic codes. Here in encoder block no need of multiplications because of it reduces time complexity, memory space as well as the battery consumption.

In[3] proposed a novel method based on network coding for optimizing error correction in wireless sensor networks. Here the input data frames are divided into blocks, if the frame contains errors then there is no need of transfer of data to receiver side, further this is to be retransmitted to the transmitter block.

In[4] presented a novel architecture design for forward error correction technique based on reed solomon coding scheme for wireless applications.

In[5] proposed a technique on error correction and detection using orthogonal code convolution. Here a k-bit data converts into an n-bit Orthogonal code and then transmits the coded block in the channel.

In[6] proposed an enhanced technique which combines OCCMP with the closest match and Vertical parity because of it errors are predicted up to (n/2-1).

In[7] proposed a technique related to orthogonal code convolution in which input data can be applied to any encoding system, which can also be used mainly in digital transmission.

In[8] designed and implemented on FPGA which includes the novel enhancement of error correction and detection.

In[9] proposed a coding and decoding technique using Reed-Solomon codes for an ADSL modem which mainly concentrates on the part of the modem that exploits the single latency operation of the device.

In[10] concentrated in error correction and Detection, theoretically explained all the existing methods which are to be used for the detection and correction of errors.

In[11] proposed a technique on reed Solomon codes for efficient way of detection and correction of burst errors.

In[12] proposed a technique on Detection and correction of errors using Hamming codes; it includes the identification of redundancy bits which improves the scalable property.

In[13] research for a k-bit data, the corresponding n-bit orthogonal code it is used to detect orthogonal code which is present in the lookup table. It also meet high bandwidth requirement.

In[14] proposed a method on Forward Error Correction Coding which transmits the error correction information along with the message. Here the error rate is ½ and rate ¾ orthogonal coded modulation schemes are realized by using FPGA and gives efficient bandwidth.

In[15] proposed for the digital communication which includes transmitter and the receiver by using verilog for the detection and correction of the errors.

2. Materials and Methods

2.1 Linear Feedback Shift Register

LFSR is a linear feedback shift register which is an n-bit counter and is designed for generating pseudo-random sequence. It generates random sequence which may be uniform or non-uniform. LFSR is used for communication purposes for network security i.e., it acts as a key in cryptographic process, encoder and the decoder. Here in the LFSR, input is the feedback of the previous bit. For the linear operation of the single bit data XOR bit is used. The starting input signal is of the LFSR is called as seed. If the starting input signal value of the seed is altered then the output will also be altered. Therefore LFSR is nothing but the shift register where state output depends on the feedback polynomial. We can specify a LFSR with a polynomial equation and produce pseudo-random number at the output. The feedback from the shift register effects the input is determined as taps. By changing the polynomial equation the value at the output also changes

i.e., sequence also changes. The "one" present at the end of the polynomial equation is given as the input to the first bit and the powers of the polynomial equation gives us the tapped bits. The first bit and the last bit are connected to the tap. The powers of the polynomial equation are counted from the left side. The general characteristic polynomial of the LFSR is represented as

$$g(x) = g_r x^r + g_{r-1} x^{r-1} + \text{----} + g_2 x^2 + g_1 x + g_0$$

LFSR consists of D flip flop. The maximum length of the LFSR is 2^{k-1} bits. Here we used only the D flip flop because to reduce the complexity i.e., the input follows the output according to the reset.

2.2 Transmitter

Generally a Transmitter mainly consists of three important components. They are Encoder, Encryptor and shift register. The input which is k-bit data and is given to the encoder. The encoder which will encodes this k-bit data into 2^{k-1} orthogonal codes. The orthogonal codes which are predefined and available in lookup table. Look up table consists of n-combinations of the given data. The data in which n/2 combinations are taken as orthogonal codes and other n/2 combinations are antipodal codes. Then these orthogonal codes which are taken from look up table are applied to the Encryptor. The encryptor module mainly constitutes of two key components. They are Linear Feedback Shift Register (LFSR) and XOR. The LFSR is used to generate Pseudo random numbers. The orthogonal codes are XOR'ed with the pseudo random numbers which are generated in the LFSR. Now, to transfer this code it must be converted into serial data mode. To enhance this operation a shift register is necessary. This shift register transmits the orthogonal codes by rising edge of the clock. This can be clearly observed in Figure 1.

2.3 Receiver

A Receiver block consists of decryptor, decoders, comparator and a shift register. The serial bit is converted into parallel code with the help of the shift register. The received data is then applied to the decryptor. The received orthogonal code is XOR'ed with the random number, which is generated by the LFSR in synchronous with clock and reset. This generates an output which is decrypted one. This decrypted code is then processed through the decoders. We used two decoders namely decoder 1 and decoder 2. Decoder1 is for orthogonal codes and decoder 2 is for antipodal codes. Decoder 1 has n/2 combinations and decoder 2 has other n/2 combinations. Here we have used two decoders in order to decrease the time complexity. Thus, the numbers of clock pulses are reduced to n/2 and hence it reduces the time complexity. The decoders here compare all the combinations which are present in the lookup table. The decoder also consists of counter which counts the number of 1's in the orthogonal and antipodal codes which is obtained from the XOR operation which occurs between the received code and each and every combination of the orthogonal and an antipodal codes in the look up table.

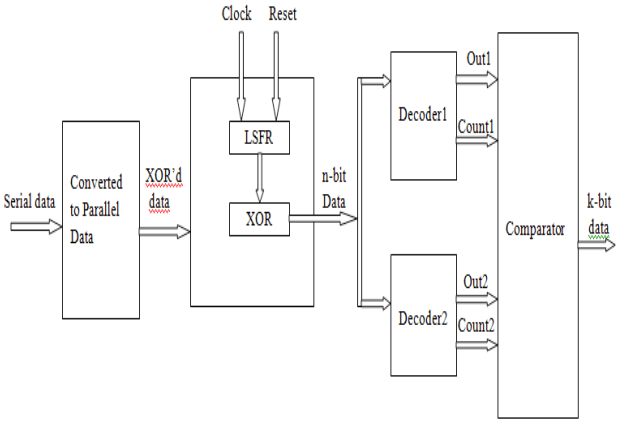

Figure 1. Block diagram of transmitter.

Then it searches for the minimal count in each decoder. Here we will compare the both count values of both decoder 1 and decoder 2. If Decoder1 gives the minimal count than decoder 2 then we get the output as the count value in the decoder1 .If the counter shows zero count then it is a error free code else, there is a error in the code. Here we consider the value other than zero represents there is a error in the received code. The minimum count in the decoder1 and the minimum count in the decoder are then compared using comparator. Then minimum count obtained from comparator is taken and it is matched with orthogonal codes or the antipodal codes which is present in the lookup table. The error correction takes place if there is any mismatch in these codes. The corrected from the received code, is then decoded to the original k-bit data and is obtained at the receiver side. Here the receiver can detect and correct upto (n/4)-1 bits in the received code. In this way multiple errors can be

detected and corrected and it takes place in the decoder of the receiver. If the minimum count has more than one combination of orthogonal code then the signal REQ goes high in which represented as 1 that is request for the retransmission of the data. This can be clearly observed in Figure 2.

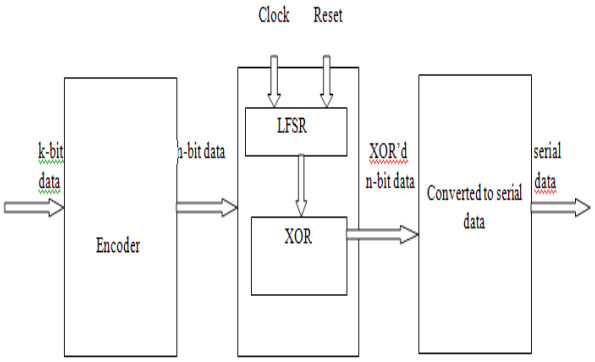

Figure 2. Block diagram of receiver.

3. Results and Discussion

Here we have used ISE Xilinx 14.1 software and a hardware board of Spartan-3 to test the output The Modelsim XE software is used for the simulation purpose. The output of the simulation i.e., the final results are checked for 4-bit combinations of input and 8-bit orthogonal code. The working of the clock cycles and the input data taken is briefly specified for both transmitter and receiver.

3.1 Transmitter Module

After simulating code which is present in the encoder, encryptor and parallel to shift register, the working and the output of the transmitter can be observed clearly in Figure 5. Here the encoder encodes a k-bit data to n= 2^{k-1} bits of corresponding orthogonal codes. This orthogonal code is then applied to the encryptor. This encrypted code which is then generated from shift register will be converted into serial data, for the transmission.

For example, the 4-bit data 0001 which is taken as the input data for the encoder and we get the output as 8-bit, where k=4 and the respective orthogonal code as 2^{k-1} =2^3 =8 is obtained. The encoder signal as shown in the Figure 3 represents the output of the encoder 01010101 which is applied as input to the encryptor. Then the processed output from the encryptor 00110000 is obtained for the 4-bit input as shown in the Figure 4. The generated orthogonal code is then transmitted to the receiver block serially by using a shift register with the rising edge of the clock.

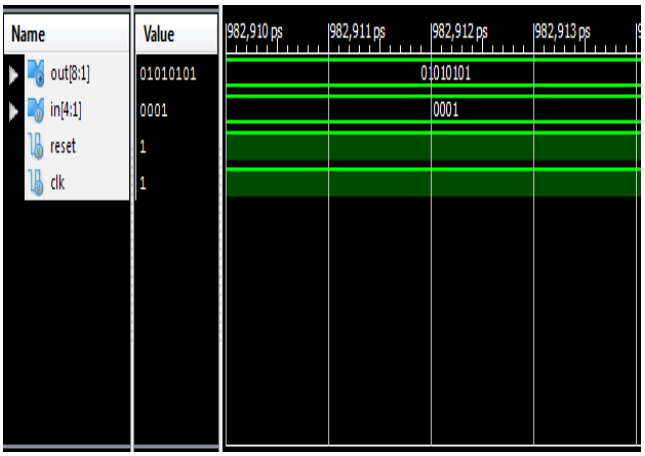

Figure 3. Simulation output of encoder.

Figure 4. Simulation output of Encryptor.

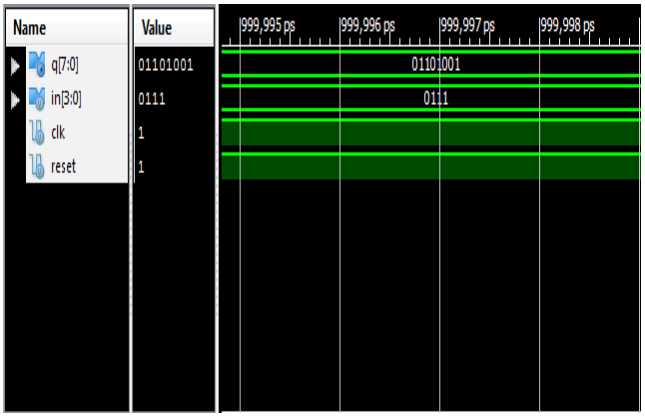

Figure 5. Simulation output of transmitter.

3.2 Receiver Module

The transmitted data is applied to the receiver and then the serial data is converted into parallel data at the receiver side to recover original data. The received signal represents the input signal of receiver. Then this data is decrypted using decryptor and then we obtain the orthogonal codes for this decrypted data. Then it is applied to both the decoders i.e., to orthogonal decoder1 and antipodal decoder 2. Here in the decoders a counter variable is used in which it is used to count the number of 1's in the result and this is noted as the counter value. When the received code is XOR'ed with each and every possible combination in the orthogonal code in the decoder1 and each and every possible combination in the antipodal code in the decoder2, the count value of the decoder1 and the count value of the decoder 2 are obtained. The correct data is obtained by comparing the minimum count of the received data in the decoder1 and the minimum count of the received data in the decoder 2. Therefore the minimum count in the decoder1 and the minimum count of the decoder 2 are compared using the comparator, and then the minimum count of the received data is obtained. Here we get 3 cases which we specified below.

In the first case, the received code is then applied to the decryptor in which the received data is XOR'ed the LFSR, and then we get decrypted output data i.e., 10101010. This is given to both the decoders here in which it is XOR'ed with each and every combination in the lookup table. Then it checks for the closet match in both the decoders and gives the minimum count in decoder1 and also gives minimum count in decoder2.Now the minimum counts from both the decoders will be compared in comparator. This count value represents the number of errors in the received data. In this case if both the count values are in high impedance state i.e., ZZZZ then out value and count value be ZZZZ or else the value will be count1 as shown in Figure 6.

In the second case, the received code is 10010111, this received code is then applied to the decryptor in which the received data is XOR'ed with the LFSR then we get decrypted output data i.e., 10010111. This is given to both the decoders here in which it is XOR'ed with each and every combination in the lookup table. And it checks for the closet match in both the decoders and gives the minimum count from decoder1 and also gives minimum count from decoder 2.Now the minimum counts from both the decoders will be compared in comparator. This count represents the number of errors in the received data. . In this case if both the count values are in high impedance state i.e., ZZZZ then out value and count value will be ZZZZ or else the value will be count2 as shown in Figure 7.

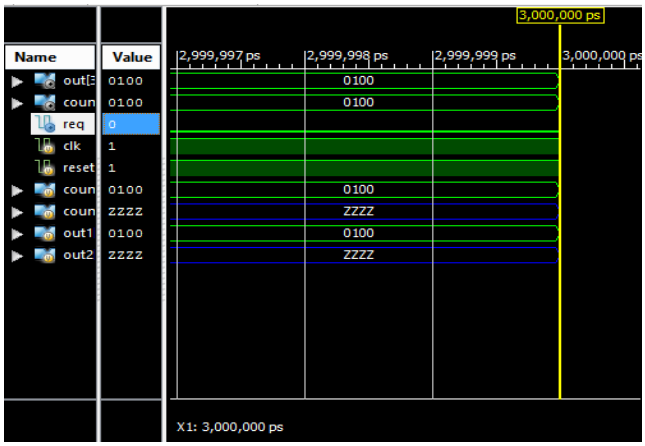

Figure 6. Simulation output of case1.

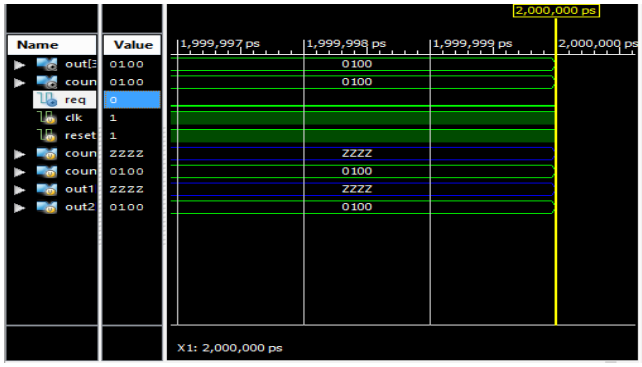

Figure 7. Simulation output of case 2.

In the third case, the received code is 001100000, this received code is then applied to the decryptor in which the received data is XOR'ed with the LFSR then we get decrypted output data i.e., 00110000. This is given to both the decoders here in which it is XOR'ed with each and every combination in the lookup table. Then it checks for the closet match in both the decoders and gives the minimum count from decoder1 and also gives minimum count from decoder 2.Now the minimum counts from both the decoders will be compared in comparator. This count represents the number of errors in the received data. In this case if count1 value is less than the count 2 value, then the out value and the count value is count1 and out1, if count2 value is less than the count1 value than the out value and the count value is count 2 and out 2, if count1 is equal to the count2 then the received

code is error one. And the req will be high it requests the sender to resend the data as shown in Figure 8. Table 1 shows the results of error detection & correction rate. As the input bit size increases the detection and correction capability also increases. For k-bit data the error detection and correction rate is calculated using as the number of bit size increases correction rate approaches to 100%. Table 2 Delay times of transmitter and receiver obtained from the simulation reports of 4-bit, 5-bit, 6-bit, 7-bit and 8-bit data. The delay time is been simulated and synthesized using Xilinx ISE 14.1. These are also implemented on sparton-3E FPGA.

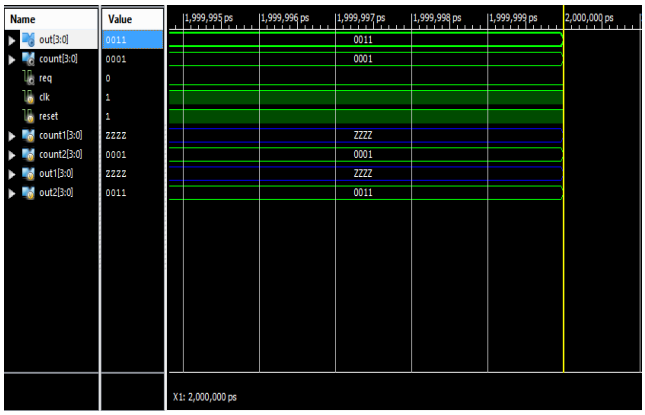

Figure 8. Simulation output of case 3.

Table 1. Results of error detection and correction

Input bits(k)	Orthogonal code O/P bit length(n)	**No of Errors(e)**	**% of errors detected and corrected**
4	8	1	93.75
5	16	3	99.95
6	32	7	99.99
7	64	15	99.99
8	128	31	100
N	$n=2^{(k-1)}$	e=(n/4)-1	$\%=(2^e-2e)/2^e$

4. Conclusion

Our technique to implement Error detection and correction has reached 100% efficiency. If there are any errors present in the data then this technique can be capable to detect and correct in the receiver and the original data can be recovered. Here we have compared the delay for the 4-bit, 5-bit, 6-bit, 7-bit, 8-bit data. Therefore the delay is increased when we increase the number of bits and it gets corrected up to (n/4-1) bits for n-bit orthogonal codes. This is implemented using Xilinx software and Verilog is used as the technique to detect and correct data bits by using cryptographic technique. Finally, this paper has a future work to limit the bandwidth for error detection and correction.

5. References

1. Narendra Babu T, Noorbasha F, Gunnam LC. Implementation of high security cryptographic system with improved error correction and detection rate using FPGA. International Journal of Electrical and Computer Engineering. 2016 Apr; 6(2):602-10.
2. Qamar RA, Maarof MA, Ibrahim S. An efficient encoding algorithm for (n, k) binary cyclic codes. Indian Journal of Science and Technology. 2012 May; 5(5): 1-5.
3. Azari L, Ghaffari. Proposing a novel method based on network coding for optimizing error recovery in wireless sensor networks. Indian Journal of Science and Technology. 2015 May; 8(9):859–67.
4. Pasricha S, Sharma S. FPGA based design of Reed Solomon codes. Indian Journal of Science and Technology. 2009 Mar; 2(4):1-5.
5. Gholase M, Thakare LP, Deshmukh AY. Enhancement of error detection and correction capability using orthogonal code convolution. International Journal of Computational Engineering Research. 2013 Apr; 3(4):66-71.
6. Jirapure SD, Balwani S. Design and implementation of orthogonal code convolution using enhanced error control technique. International Journal of Engineering Research

Table 2. Delay times of transmitter and receiver obtained from the synthesis reports for 4-bit, 5-bit, 6-bit, 7-bit and 8-bit data

Input data	**Encoder delay(ns)**	**Encryptor delay(ns)**	**Transmitter delay(ns)**	**Decryptor delay(ns)**	**Decoder delay(ns)**	**Receiver delay(ns)**
4 bit	4.010	5.597	5.597	5.597	15.604	13.493
5 bit	4.010	6.209	6.209	6.209	18.329	17.398
6 bit	4.010	7.308	7.308	7.308	21.453	20.501
7 bit	4.010	7.419	7.419	7.419	34.591	39.689
8 bit	4.010	7.823	7.823	7.823	51.816	52.564

and Applications. International Conference on Industrial Automation and Computing; 2014 Apr; p. 1-5.
7. Kaabouch N, Dhirde A, Faruque S. Improvement of Orthogonal Code Convolution Capabilities using FPGA Implementation. IEEE Electro/Information Technology Proceedings; 2007. p. 337-9.
8. Wang A, Kaabouch N. FPGA based design of a novel enhanced error detection and correction technique. IEEE International Conference on Electro/Information Technology; 2008 May. p. 25-9.
9. Stylianakis V, Toptchiyski S. A reed-Solomon coding/decoding structure for an ADS modem. IEEE International Conference on Electronics, Circuits and Systems. 1999, pp. 430 – 434.
10. Gupta V, Verma C. Error detection and correction. International Journal of Advanced Research in Computer Science and Software Engineering. 2012 Nov; 2(11):1-7.
11. Shrivastava P, Singh UP. Error detection and correction using reed solomon codes. International Journal of Advanced Research in Computer Science and Software Engineering. 2013 Aug; 3(8):1-5.
12. Kumar UK, Umashankar BS. Improved hamming code for error detection and correction. 2nd International Symposium on Wireless Pervasive Computing; 2007 Feb.
13. Panse Mrunal S, Meshram S, Chaware D, Raut A. Error detection using orthogonal code. IOSR Journal of Engineering. 2014 Mar; 4(3):7-11.
14. Faruque S, Kaabouch N, Dhirde A. Wireless and optical communication proceeding. ACTA Press; 2007 Jun. p. 565-630.
15. Meshram VP, Tawade NR, Deshmukh SM, Nandanwar SV, Borkar DD. Implementation of error correction technique using OCC on FPGA. IJSTE. 2016 Mar; 2(9):12-6.