

Implementation of Encoder for (31, k) Binary BCH Code based on FPGA for Multiple Error Correction Control

Samir Jasim Mohammed
College of Engineering, University of Babylon.
Babylon, Iraq

ABSTRACT

This paper describes the design and implementation of (31,k) binary BCH (Bose, Chaudhuri, and Hocquenghem) encoder using a Field Programmable Gate Array (FPGA) reconfigurable chip. BCH code is one of the most important cyclic block codes. Designing on FPGA leads to a high calculation rate using parallelization (implementation is very fast), and it is easy to modify. BCH encoder has been designed and simulated using Xilinx-ISE 10.1 Web PACK and implemented in a xc3s700a-4fg484 FPGA. In this implementation, A 31 bit-size code word has been used. The BCH code encoders of (31, 26, 1), (31, 21, 2), (31, 16, 3), (31, 11, 5) and (31, 6, 7) have implemented on FPGA. The results show that the systems work quite well.

Keywords

Error correcting codes, BCH codes, BCH encoder, FPGA

1. INTRODUCTION

The reliable transmission of information over noisy channels is one of the basic requirements of digital information and communication systems. Error correction coding is the means whereby errors which may be introduced into digital data as a result of transmission through a communication channel can be corrected based upon received data [1,2]. Error correcting codes have a wide range of applications in different fields like digital data communications, memory system design, and fault tolerant computer design among others [3].

BCH codes are one of the most powerful random-error correcting cyclic codes. BCH codes can be defined by two parameters that are code size n and the number of errors to be corrected t . BCH codes are being widely used in mobile communications, computer networks, satellite communication, as well as storage systems such as computer memories or the compact disc [1,4].

BCH codes are polynomial codes that operate over Galois fields (or finite fields). The generator polynomial of this code is specified in terms of its roots from the Galois field $GF(2^m)$. Let α be a primitive element in $GF(2^m)$. The generator polynomial $g(x)$ of the t error-correcting BCH code of length $2^m - 1$ is the lowest-degree polynomial over $GF(2)$ which has $\alpha, \alpha^2, \alpha^3, \dots, \alpha^{2t}$ as its roots [i.e., $g(\alpha^i) = 0$ for $1 \leq i \leq 2t$]. Let $\phi_i(x)$ be the minimal polynomial of α^i . Then $g(x)$ must be the least common multiple of $\phi_1(x), \phi_2(x), \dots, \phi_{2t}(x)$, that is,

$$g(x) = \text{LCM} \{ \phi_1(x), \phi_2(x), \dots, \phi_{2t}(x) \} \quad (1)$$

For any positive integers m ($m \geq 3$) and t ($t < 2^{m-1}$), there exists a binary BCH code with parameters of code words length $n = 2^m - 1$, number of parity check bits $n - k \leq mt$, and minimum distance ($d_{min} \geq 2t + 1$) [5].

2. BCH CODES

BCH codes are polynomial codes that capable of correcting any combination of t or fewer errors in a block of $n = 2^m - 1$ digits. To know the encoding and decoding of the BCH code, the knowledge of finite fields is necessary [5].

An (n,k) binary BCH code encodes k -bits message into n -bits code word. The message vector can be expressed in a polynomial form as follows [6]:

$$m(x) = m_0 + m_1x + m_2x^2 + \dots + m_{k-1}x^{k-1} \quad (2)$$

The message digits are utilized as a part of the codeword. The systematic encoding can be implemented by:

$$c(x) = p(x) + x^{n-k} m(x) \quad (3)$$

Where $p(X)$ is the remainder and can be expressed as

$$p(X) = x^{n-k} m(x) \text{ mod } g(x) \quad (4)$$

It follows from the definition of a t -error-correcting BCH code of length $n = 2^m - 1$ that each code polynomial has $\alpha, \alpha^2, \dots, \alpha^{2t}$ as roots, $c(\alpha^i) = 0$, for $(1 \leq i \leq 2t)$ [7].

The encoding of a cyclic BCH code in systematic form can be expressed by the circuit shown in Fig.(1). The systematic encoding procedure can be described with the following steps [7]:

- i. Switch 1 is closed during the first k shifts, to allow transmission of the message bits into the $n-k$ stage encoding shift register.
- ii. Switch 2 is in down position to allow transmission of the message bits directly to an output register during the first k shifts.
- iii. After transmission of the k th message bit, switch 1 is opened and switch 2 is moved to the up position.
- iv. The remaining $n-k$ shifts clear the encoding register by moving the parity bits to the output register.
- v. The total number of shifts is equal to n , and the contents of the output register is the codeword polynomial $p(x) + x^{n-k} m(x)$.

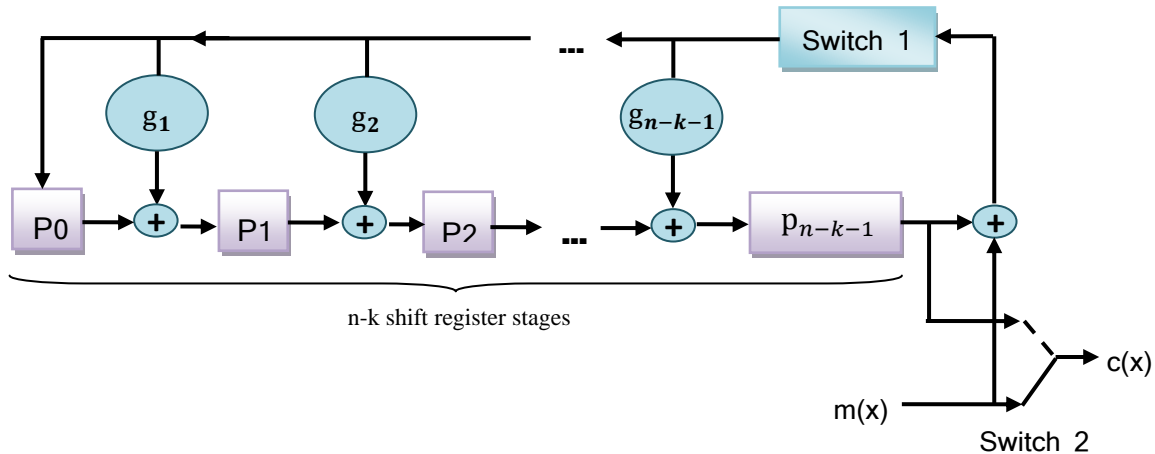


Fig 1: Encoding with an (n-k) stage shift register.

3. FIELD PROGRAMMABLE GATE ARRAY (FPGA)

Field-Programmable Gate Arrays (FPGAs) are pre-fabricated silicon devices that can be electrically programmed to become almost any kind of digital circuit or system [8].

FPGAs provide a number of advantages over fixed-function Application Specific Integrated Circuit (ASIC) technologies such as standard cells. ASICs are designed for specific application, and once manufactured, they cannot be modified, while FPGAs are configured in a relatively short amount of time, and often be reconfigured if a mistake is made [8,9].

An FPGA consists of an array of uncommitted configurable logic blocks (CLBs), programmable interconnects and Input Output blocks (IOBs). The basic architecture of an FPGA is shown in Fig.(2). FPGA architecture is dominated by programmable interconnects, and the configurable logic blocks which are relatively simple. This feature makes these devices far more flexible in terms of the range of designs that can be implemented with these devices [10].

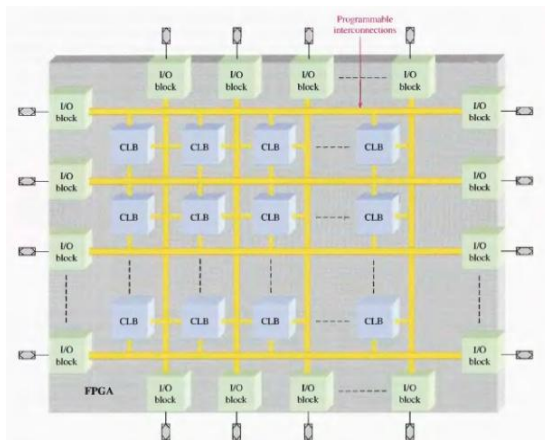


Fig 2: FPGA architecture

FPGA can be configured anytime is needed, having a structure based on RAM technology that allowing the interconnectivity of the components to be changed as required. On the other hand, they allow parallel structures implementation, with response time less than a system with microprocessor [11].

4. PROPOSED BCH ENCODERS DESIGN

The system proposed in this paper is based on the use of reconfigurable FPGA circuit for hardware implementation of encoders. FPGA board is connected with a computer in order to download the software of the system into an FPGA chip. Five BCH encoders have been designed which are (31, 26, 1), (31, 21, 2), (31, 16, 3), (31, 11, 5) and (31,6, 7).

4.1 Design of Encoder for (31,26) BCH Code

The encoding circuit calculates the parity bits using the LFSR (Linear Feedback Shift Register). The feedback connections of the LFSR are formed in a way that depends on the generator polynomial of the code. The generator polynomial of the (31,26) BCH code is:

$$g(x) = 1 + x^2 + x^5$$

The message digits are utilized as a part of the codeword. The message digits are shifted into the rightmost 26 stages of a codeword register, and then appending the parity digits by placing them in the leftmost 5 stages. The input data of the encoding circuit is 26 bits and the output is a serial of 31 bits. The proposed encoder of (31,26) single error correcting BCH code has been implemented on FPGA target device as shown in Fig.(3).

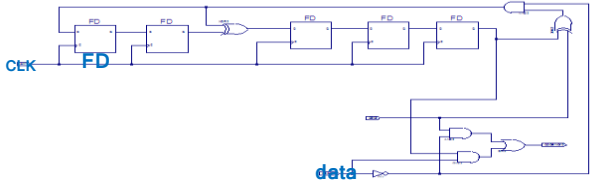


Fig 3: Encoding logic circuit of (31,26) BCH code implemented on FPGA.

For the BCH encoding circuit, a control signal (cecode) is needed to allow the data signal to enter the encoding circuit and pass to the output at the same time. Also control signal gives a delay in order to make the encoding circuit able to prepare the parity bits. As a result the control signal is necessary to control the operation of switch 1 and switch 2 shown in Fig.(1). The complete encoding circuit with control signal is shown in Fig.(4) below

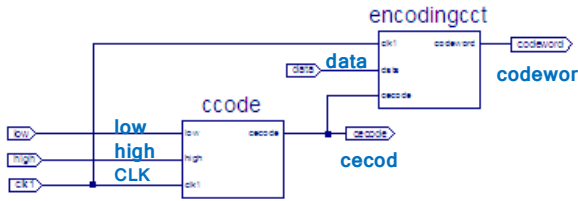


Fig 4: Schematic of (31,26) BCH Encoder

4.2 Design of Encoder for (31,21) BCH Code

The encoding circuit calculates the parity bits using the LFSR. The feedback connections of the LFSR are formed in a way that depends on the generator polynomial of the code. The generator polynomial of the (31,21) BCH code is:

$$g(x) = 1 + x^3 + x^5 + x^6 + x^8 + x^9 + x^{10}$$

The input data of the encoding circuit is 21 bits and the output is a serial of 31 bits. The proposed encoder of (31,21) double error correcting BCH code has been implemented on FPGA target device as shown in Fig.(5).

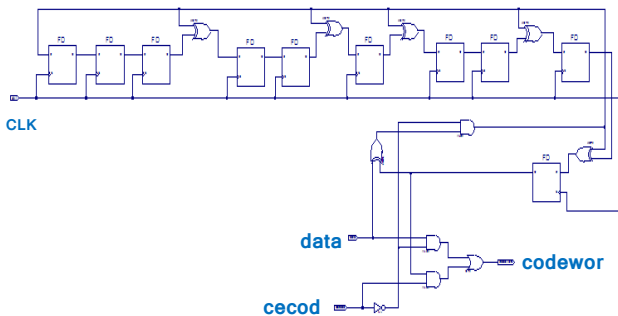


Fig 5: Encoding logic circuit of (31,21) BCH code implemented on FPGA.

4.3 Design of Encoder for (31,16) BCH Code

The parity bits of the (31,16) BCH code have been calculated by using the LFSR. The feedback connections of the LFSR are formed in a way that depends on the generator polynomial of the code. The generator polynomial of the (31,16) BCH code is:

$$g(x) = 1 + x^3 + x^5 + x^6 + x^8 + x^9 + x^{10}$$

The input data of the encoding circuit is 16 bits and the output is a serial of 31 bits. The proposed encoder of (31,16) triple error correcting BCH code has been implemented on FPGA target device as shown in Fig.(6).

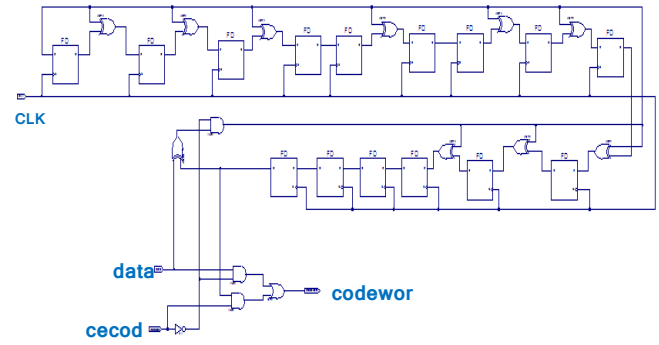


Fig 6: Encoding logic circuit of (31,16) BCH code implemented on FPGA.

4.4 Design of Encoder for (31,11) BCH Code

The parity bits of the (31,11) BCH code have been calculated by using the LFSR. The feedback connections of the LFSR are formed in a way that depends on the generator polynomial of the code. The generator polynomial of the (31,11) BCH code is:

$$g(x) = 1 + x^2 + x^4 + x^6 + x^7 + x^9 + x^{10} + x^{13} + x^{17} + x^{18} + x^{20}$$

The input data of the encoding circuit is 11 bits and the output is a serial of 31 bits. The proposed encoder of (31,11) five error correcting BCH code has been implemented on FPGA target device as shown in Fig.(7).

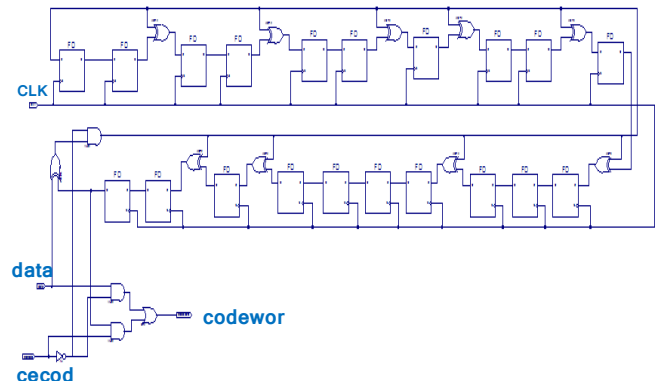


Fig 7: Encoding logic circuit of (31,11) BCH code implemented on FPGA.

4.5 Design of Encoder for (31,6) BCH Code

For the (31,6) BCH code, the feedback connections of the LFSR of the encoding circuit are formed in a way that depends on the

generator polynomial of the code. The generator polynomial of the (31,6) BCH code is:

$$g(x) = 1 + x + x^2 + x^5 + x^9 + x^{11} + x^{13} + x^{14} + x^{15} + x^{16} + x^{18} + x^{19} + x^{21} + x^{24} + x^{25}$$

The input data of the encoding circuit is 16 bits and the output is a serial of 31 bits. The proposed encoder of (31,6) seven error correcting BCH code has been implemented on FPGA target device as shown in Fig.(8).

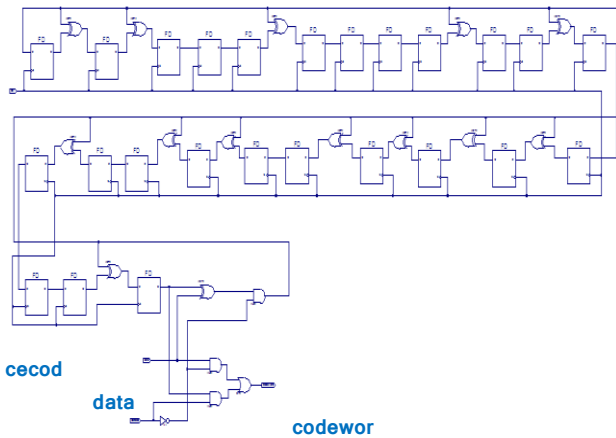


Fig 8: Encoding logic circuit of (31,6) BCH code implemented on FPGA.

5. RESULTS AND DISCUSSION

Five BCH encoders have been implemented on Spartan 3a-xc3s700a FPGA. Fig.(9) shows the experimental system. The FPGA is connected with a computer in order to download the software of each system into an FPGA chip. The system is implemented in 1.5625 MHz clock frequency such as in [12] and [13]. The simulation results show that the circuits work quite well.

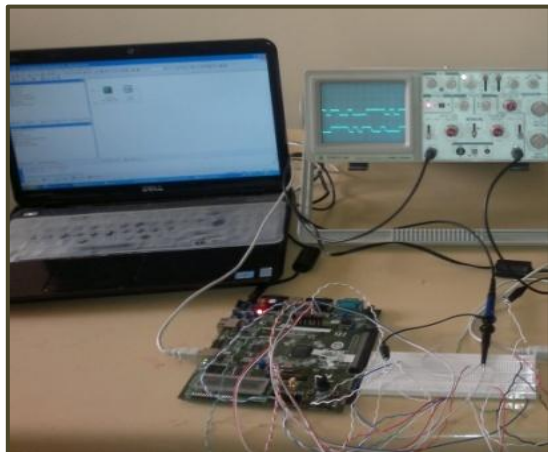


Fig 10: The experimental system

Fig.(10) shows the simulation result of (31,26) BCH encoder using Xilinx-ISE 10.1 simulator and illustrates the date word

(10011101001011000110101001) which entered to the FPGA encoder to be encoded. After adding the parity bits to the data, the codeword are performed to be (0001110011101001011000110101001) as shown in the figure.

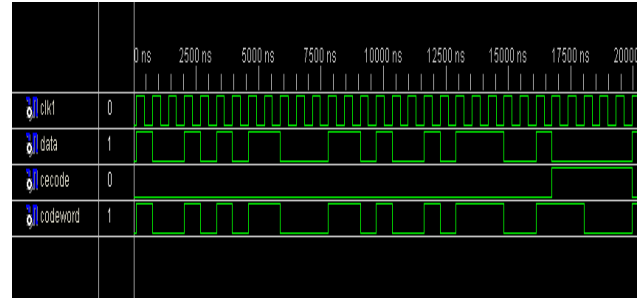


Fig 10: Simulation results of (31,26) BCH encoder

The simulation result of (31,21) BCH encoder is shown in Fig.(11) below. The figure illustrates the date word (100111010010110001101) which entered to the FPGA encoder to be encoded. After adding the parity bits to the data, the codeword are performed to be (1110011010100111010010110001101) as shown in the figure.

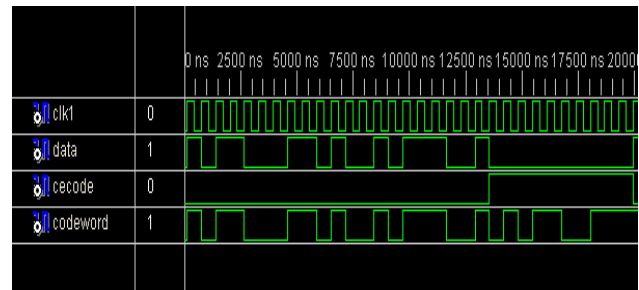


Fig 11: Simulation results of (31,21) BCH encoder

For the (31,16) BCH encoder, the Xilinx-ISE 10.1 simulation result shows the date word (1001110100101101) which entered to the FPGA encoder to be encoded. After adding the parity bits to the data, the codeword are performed to be (0100001011110101001110100101101) as shown in the figure.

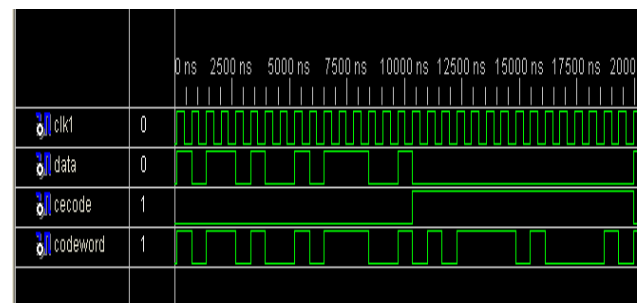


Fig 12: Simulation results of (31,16) BCH encoder

Fig.(13) shows the simulation result of (31,11) BCH encoder using Xilinx-ISE 10.1 simulator and illustrates the date word (01011101001) which entered to the FPGA encoder to be encoded. After adding the parity bits to the data, the codeword

are performed to be (0100001011110010100101011101001) as shown in the figure.

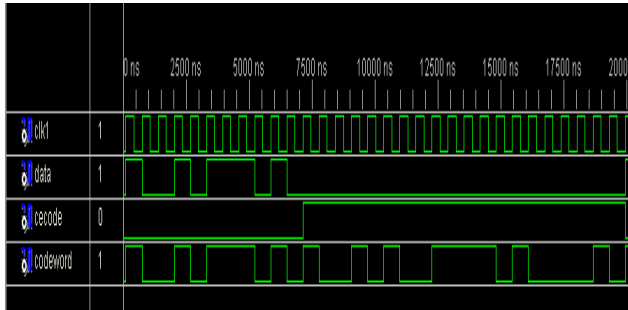


Fig 13: Simulation results of (31,11) BCH encoder

In Fig.(14), the simulation result of (31,6) BCH encoder using Xilinx-ISE 10.1 simulator is shown. The figure illustrates the data word (011001) which entered to the FPGA encoder to be encoded. After adding the parity bits to the data, the codeword are performed to be (1111000110111010100001001011001) as shown in the figure.

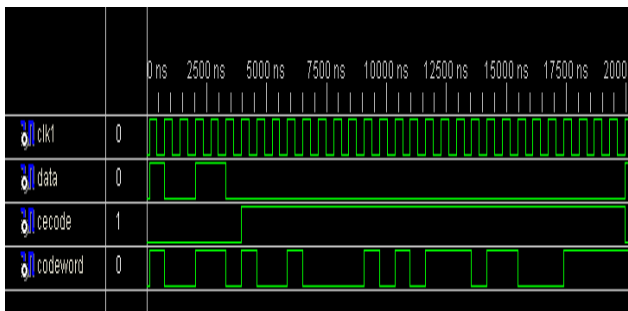
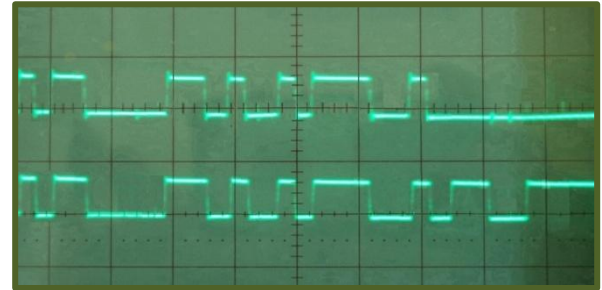
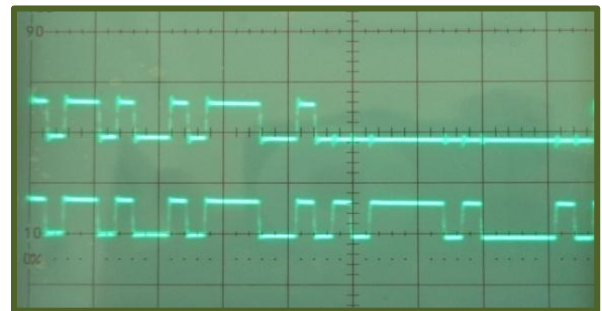


Fig 14: Simulation results of (31,6) BCH encoder

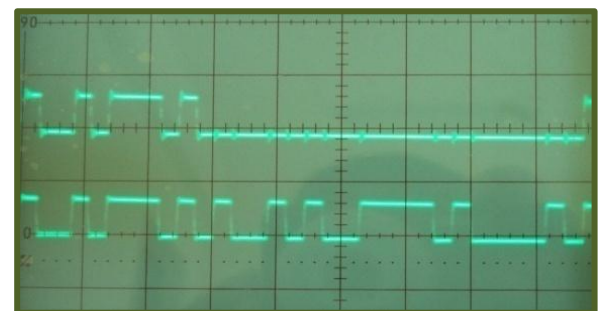
The synthesis process produces a bit stream file that can be downloaded to the FPGA board. The bit stream file of the BCH code has been successfully downloaded to XC3S700A-4FG484 FPGA board. The results of encoder are displaced on an oscilloscope. The experimental results of the five BCH encoders are shown in Fig.(15). The results of encoder decoder on the FPGA are displayed on oscilloscope.



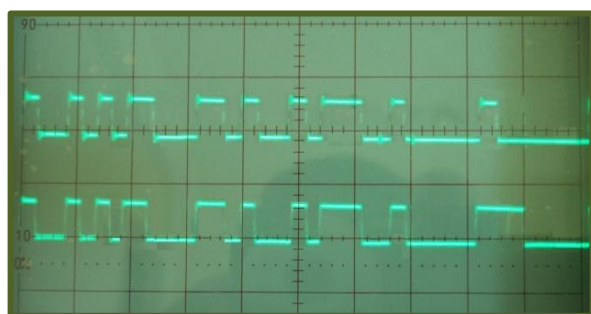
**Fig 15b: Data word (21 bits) and codeword(31 bits).
 Voltage: 5V/DIV, Time: 2µsec/DIV**



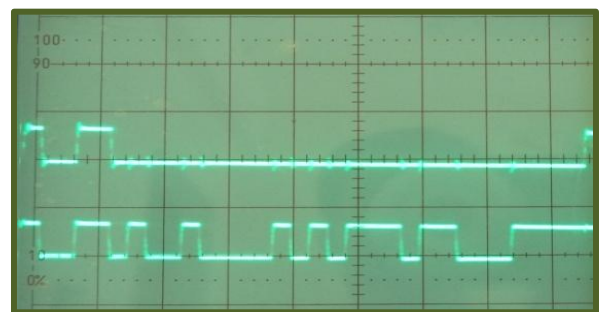
**Fig 15c: Data word (16 bits) and codeword(31 bits).
 Voltage: 5V/DIV, Time: 2µsec/DIV**



**Fig 15d: Data word (11 bits) and codeword(31 bits).
 Voltage: 5V/DIV, Time: 2µsec/DIV**



**Fig 15a: Data word (26 bits) and codeword(31 bits).
 Voltage: 5V/DIV, Time: 2µsec/DIV**



**Fig 15e: Data word (6 bits) and codeword(31 bits).
 Voltage: 5V/DIV, Time: 2µsec/DIV**

The behavior of multiple error correcting of (31, k) BCH encoder has been studied and compared by implementing on FPGA. After the hardware synthesis of the five BCH encoders, the following device utilization summary was obtained:

Table 1. BCH Encoders area summary.

Components (utilization)	(31,26,1) Encoder	(31,21,2) Encoder	(31,16,3) Encoder	(31,11,5) Encoder	(31,6,7) Encoder
Number of Slices	19 out of 5888	22 out of 5888	26 out of 5888	27 out of 5888	30 out of 5888
Number of slices flip flops	37 out of 11776	42 out of 11776	47 out of 11776	52 out of 11776	57 out of 11776
Number of 4 input LUTs	34 out of 11776	38 out of 11776	42 out of 11776	42 out of 11776	46 out of 11776
Number of bonded IOBs	6 out of 372	6 out of 372	6 out of 372	6 out of 372	6 out of 372
Number of GCLKs	1 out of 24	1 out of 24	1 out of 24	1 out of 24	1 out of 24

Table (1) shown above illustrates that the chip area occupied in a xc3s700a-4fg484 FPGA is very small. The (31, 6) BCH encoder occupied more area than the other codes since it has greater parity bits (25 bits) than the other codes, and corrects seven bits error

6. CONCLUSION

The reliable transmission of information over noisy channels is one of the basic requirements of digital information and communication systems. Because of this requirement, modern communication systems rely heavily on error control coding.

In this work, the design and implementation of five BCH encoders using a Field Programmable Gate Array (FPGA) have been presented. The codes used in this work are (31, 26, 1), (31, 21, 2), (31, 16, 3), (31, 11, 5) and (31, 6, 7).

The five proposed BCH encoders have been designed and simulated using Xilinx-ISE 10.1 Web PACK and implemented

in a xc3s700a-4fg484 FPGA and the results show that the system works quite well.

BCH codes have been shown to be excellent error correcting codes among codes of short lengths. They are simple to encode and relatively simple to decode. Due to these qualities, there is much interest in the exact capabilities of these codes.

7. REFERENCES

- [1] Neubauer, J. Freudenberger and V. Kuhn “Coding Theory Algorithms, Architectures and Applications” John Wiley & Sons, 2007.
- [2] T. K. Moon, “Error Correction Coding”, John Wiley & Sons, 2005.
- [3] A. S. Das, S. Das, and J. Bhaumik “Design of RS (255,251) Encoder and Decoder in FPGA”, international journal of soft computing and engineering, Volume- 2, Issue-6, January 2013.
- [4] J. G. Proakis, ”Digital Communications”, Prentice-Hall, 4th edition, 2005.
- [5] S. Lin, and D.J. Costello Jr. “Error Control Coding Fundamentals and Applications”, Prentice-Hall, New Jersey, 1983.
- [6] R. Merha, G. Saini, and S. Singh, “FPGA Based High Speed BCH Encode for Wireless communication Applications”, International Conference on Communication System and Network Technologies, 2011.
- [7] B. Sklar, “Digital Communications Fundamentals and Applications”, Prentice Hall, 2nd edition, 2001.
- [8] I. Kuon, R. Tessier and J. Rose. “ FPGA Architecture: Survey and Challenges”, 2008.
- [9] J. P. Deschamps, G. J. A. Bioul and G. D. Sutter, “Synthesis of Arithmetic Circuits FPGA, ASIC and Embaded Systems”, John Wiley & Sons, 2006.
- [10] A.K. Maini, “ Digital Electronics Principles, Devices and Applications ”, John Wiley & Sons, 2007.
- [11] R. Woods, J. McAllister, G. Lightbody and Y. Yi, “FPGA-based Implementation of Signal Processing Systems”, John Wiley & Sons, 2008.
- [12] S. J. Mohammed, H. F. Abdulsada, “Design and Implementation of 2 BCH Error Correcting Codes using FPGA”, Journal of Telecommunicatios, Volume 19, ISSUE 2, APRIL 2013.
- [13] S. J. Mohammed, H. F. Abdulsada, “FPGA Implementation of 3 bits BCH Error Correcting Codes”, International Journal of Computer Applications, Volume 71– No.7, May 2013.