Hardware Implementation of BCH Error-Correcting Codes on a FPGA

Laurenţiu Mihai Ionescu
University of Piteşti

Constantin Anton
University of Piteşti

Ion Tutănescu
University of Piteşti

Alin Mazăre
University of Piteşti

Gheorghe Şerban
University of Piteşti

Abstract

Our paper presents the prototyping of a BCH (Bose, Chaudhuri, and Hocquenghem) encoder and decoder using a Field Programmable Gate Array (FPGA) reconfigurable chip. The solutions implemented on FPGA lead to a high calculation rate using parallelization. We implemented the BCH code in a 3s400FG456 FPGA. In this implementation we used 15 bit-size word code and the results show that the circuits work quite well.

1. Introduction

Using error correcting control is very important in modern communication system. BCH codes (Bose, Chaudhuri, and Hocquenghem) are being widely used in communication networks, computer networks, satellite communication, magnetic and optic storage systems. This paper presents the prototyping of a BCH encoder and decoder using FPGA.

BCH codes operate over finite fields or Galois fields. BCH codes can be defined by two parameters that are: length of code words, *n*, and the number of errors to be corrected, *t*.

The BCH codes are a class of cycle codes whose generator polynomial is product of distinct minimal polynomials corresponding to $\alpha, \alpha^2, \ldots, \alpha^{2t}$, where $\alpha \in GF(2^m)$ is a root of the primitive polynomial *g(x)*[1]. An irreducible polynomial *g(x)* of degree m is said to be primitive if only if it divides polynomial form of degree *n*, $x^n + 1$ for no *n* less than $2^m - 1$. In fact, every binary primitive polynomial *g(x)* of degree *m* is a factor of $x^{2^m - 1} + 1$.

Let $m_i(x)$ be the minimal polynomial of α^i.

Let

$$p(x) = p_0 + p_1 x + p_2 x^2 + \ldots + p_{n-1} x^{n-1} \qquad (1)$$

be a code polynomial with coefficients from GF(2). If *p(x)* has $\alpha, \alpha^2, \ldots, \alpha^{2t}$ as its roots, *p(x)* is divisible by the minimal polynomials $m_1(x), m_2(x), \ldots, m_{2t-1}(x)$. The generator polynomial *g(x)* of the *t*-error-correcting BCH of length code words $n = 2^m - 1$ and rate codes *k/n* is the lowest degree polynomial over GF(2) [2]. Thus, the generator polynomial of the code must be the least common multiple (LCM) of these minimal polynomials:

$$g(x) = LCM\{m_1(x), m_2(x), \ldots, m_{2t}(x)\}. \qquad (2)$$

In general, for any positive integer $m \geq 3$ and $t < n/2$, there is 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 $d_0 = 2t + 1 \leq d_{\min}$. The designed distance of the code is $d_0 = 2t + 1$.

The minimum distance $d_{\min}$ could be larger than d_0. Algorithm for designing BCH codes is:

1. Choose a primitive polynomial of degree m, and construct GF(2^m).
2. Find the minimal polynomial $m_i(x)$ of α^i for $i = 1, 2, \ldots, 2t$.
3. Obtain *g(x)*.
4. Determine *k* from *n* - *k*, which is the degree of *g(x)*.
5. Find the minimum distance $d_{\min} \geq 2t + 1$.

Suppose that a code word *t(x)* is transmitted and that because of the channel error *e(x)*, and to receive the word code is:

$$r(x) = t(x) + e(x), \qquad (3)$$

where *e(x)* is the error pattern.

If the correction power of code is *t*, then no more than *t* coefficients of *e(x)* are nonzero. Suppose that

l, $1 \le l \le t$, errors actually occur and they happen in unknown locations $j_1, j_2, \ldots, j_l$, that is

$$e(x) = \sum_{\lambda=1}^{l} x^{j_\lambda} \quad 0 \le j_\lambda \le n-1. \tag{4}$$

Since $\alpha, \alpha^2, \ldots, \alpha^{2t}$ are roots of each transmission word codes, $t(\alpha^i) = 0$, for $1 \le i \le 2t$. Then,

$r(\alpha^i) = e(\alpha^i)$, $i = 1, 2, \ldots, 2t$.

The decoding of received BCH word codes requires successive computational processed performed over GF(2^m) to be executed. These processes are the syndrome computations, error-locator polynomial determination.

The first step in decoding a t-error-correction BCH codes is to compute the $2t$ syndrome components $s_1, s_2, \ldots, s_{2t}$. These syndrome components may be obtained by substituting the field elements $\alpha, \alpha^2, \ldots, \alpha^{2t}$ into the received polynomial $r(x)$. Thus, the i^{th} component of the syndrome is

$$S_i = r(\alpha^i) = e(\alpha^i), \quad 1 \le i \le 2t, \tag{5}$$

from which we see that the syndrome S depends only on the error polynomial $e(x)$.

The error-locator polynomial $\sigma(x)$ can be:

$$\sigma(x) = \sigma_0 + \sigma_1 x + \ldots + \sigma_l x^l \text{ if } l \le t, \tag{6}$$

which is equivalent with:

$$\sigma(x) = (1 + \beta_1 x)(1 + \beta_2 x)\ldots(1 + \beta_l x) \tag{7}$$

Let's write $\beta_\lambda = \alpha^{j_l}$ for simplicity. Its coefficients and the error-locator number are related by the following set of equations:

$$\begin{aligned} &\sigma_0 = 1 \\ &\sigma_1 = \beta_1 + \beta_2 + \ldots + \beta_l \\ &\sigma_2 = \beta_1\beta_2 + \beta_2\beta_3 + \ldots + \beta_{l-1}\beta_l \\ &\ldots\ldots\ldots \\ &\sigma_l = \beta_1\beta_2\ldots\beta_l \end{aligned} \tag{8}$$

2. Field Programmable Gates Array (FPGA)

Field-Programmable Gate Arrays (FPGAs) have become one of the key digital circuit implementation media over the last decade [3]. One bit patterns will produce operational circuits and can be used in many areas like the communication systems.

Our hardware scheme is based on polynomial generator for errors detection and correction.

FPGA circuits represent a compromise between circuits with microprocessor and ASIC (Application Specific Integrated Circuits) [4].

First, they present flexibility in programming, called here reconfiguration, which is a feature for microprocessors. Even if FPGA cannot be programmable while operation, they can be configured anytime is needed, having a structure based on RAM programmable machines, as we see in Figure 1. On the other hand, they allow parallel structures implementation, with response time less than a system with microprocessor.

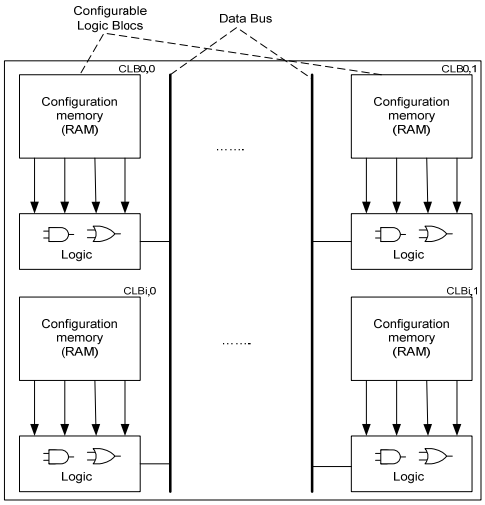


Figure 1. FPGA internal structure - block diagram

These features are used in our implementation which justifies the use of FPGA. Thus, we desire to implement an error correction circuit with enough flexibility to can be used, with small changes, to operate with different values of the communication system parameters such as the message bits number or the word ode size.

Therefore, we design modularly the entire hardware implementation, as we will describe in the following sections. The change of a parameter, such as message size, goes to addition of new blocks with same internal structure, addition which is possible by FPGA reconfigurable feature.

Also, we intend to build a communication system with real time response. Thus, when the message is transmitted from the computer to communication channel, we need a quick computation of control sum.

Secondly, to receiver, when the data package (word code) is received is needed a fast detection of errors. All those are possible with the implementation of structures which operate in parallel mode and with small propagation times, close to ASIC's.

3. Description of errors detection and correction system

The system proposed in this paper is based on the use of reconfigurable FPGA circuits for hardware implementation of error detection and correction algorithms. Error detection and correction is based on the algorithm presented in section 1. To the transmitter it is computed the remainder of division between message polynomial multiply with power of 2 for check sum polynomial size and the generator polynomial. This remainder represent check sum which is added to the message resulting the word code. This is the data package which is transmitted to communication channel.

To receiver, the entire word code is received and is divided to generator polynomial. If the remainder of division is 0 then we don't have error in the message. Else, there are errors in the message and by a decoder we indentify wrong bits in the message and correct them.

The block diagram of the system is presented in Figure 2.

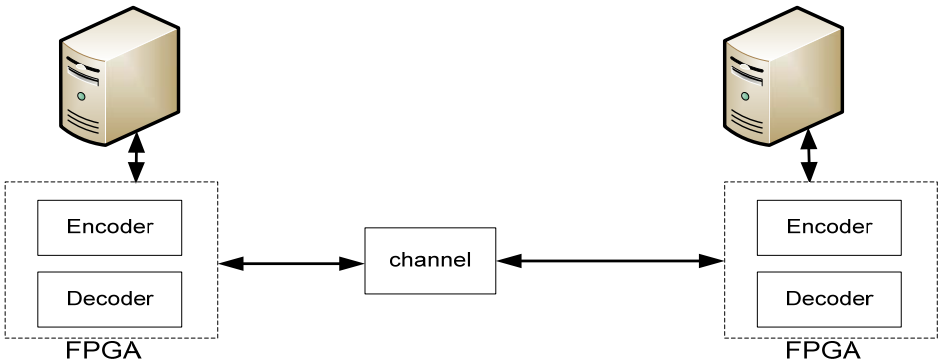


Figure 2. Communication system with FPGA implemented detection and correction errors algorithms

The communication is made between two computers. The algorithm of error detection and correction run to a FPGA circuit connected to the computer instead to run as a software application in computer. Thus, it is obtained a very small response time, compare with response time of computer and the computer can configure the FPGA when is desired the change of parameters of communication system. On both FPGA circuits connected to the computers, either transmitter computer or receiver computer, we have implemented a coder and a decoder to allow a bidirectional communication. While a computer transmits the other receive.

3.1. Transmitter module

The FPGA implemented transmitter system contains first a remainder (control sum) computing circuit, the control sum that will be attached to the message. The remainder computing circuit is based on hardware implementation of the polynomial division algorithm, with block diagram presented in the Figure 3 a).

As we show in [5], serial computing module allows starting of the division algorithm even when first bit is received from computer! Next, each received bit means a step in the division algorithm Thus, when the entire message is received from the computer, we have already a partial result from the division. Why we proceed in this way? Because all the communication performed at this time are serial communications (USB, Ethernet or wireless). When we receive serial data bits, we have already done some computations, in same time with the reception, to decrease whole computation time of the check sum. We name this computation (which is performed in the same time with serial reception of the data) serial computation and the hardware module which implements it - serial computation module.

Note that the operation which is made to encoder is illustrated in the expression bellow, for binary representation of polynomials:

$$S = \frac{M \times 2^{\deg S}}{G}$$

where: M is message polynomial, G is generator polynomial, $degS$ is order of check sum polynomial, and S is check sum polynomial.

To the end of message M reception, the division M/G was performed. Next, the result of the serial computation must be multiplied with 2 at degS power and divided to G. All these operations must be executed as quickly possible to obtain check sum S. This will be added to M which is already transmitted to communication channel.

To execute very fast this computation we built modules which operate in parallel, to get a small

response time. The parallel computing module consists of more cells which execute each a task from the division algorithm.

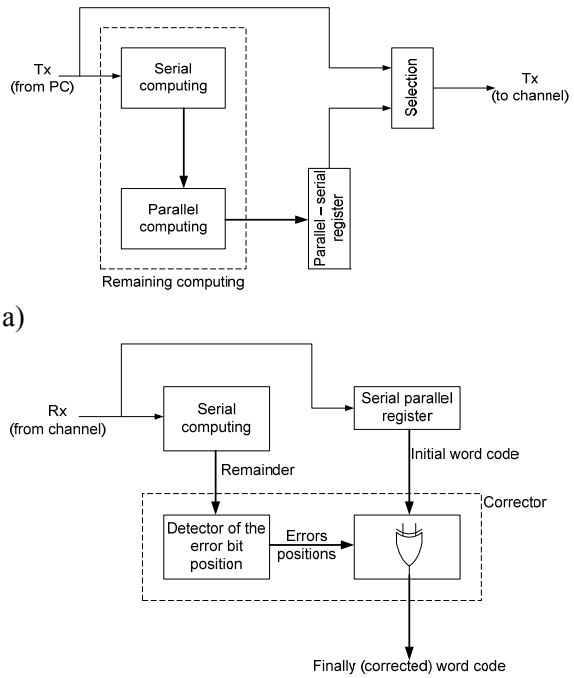


Figure 3. Encoder (a) and decoder (b) circuit

The data flow for the encoder is presented in Figure 4 a. In the top of this figure the raw string received from the computer (only message string *M*) is presented. In bottom side, final string (which is transmitted to communication channel) is presented.

The two strings are represented in relation with time. With yellow are illustrated the bits which will be involved in serial computation. When a message bit is received from computer it will be transmitted immediately to communication channel, via a selection circuit (see Figure 3).

After entire message is transmitted, follow multiplication with 2^{degS} and the final steps in division. Both operations are performed using parallel computation module. First, to partial result is appending a number of 0 equal with check sum polynomial order. This is equivalent to polynomial multiplication of the message polynomial with the remainder degree. Then the final division is performed using parallel cells.

To attach the check sum to the message and sent it to communication channel we use a parallel - serial register which take parallel output from parallel computing module and convert it in serial string which is selected to be transmitted by selection circuit (Figure 3). Here a small delay appears (with red color in Figure 4) which is given by the response time of the parallel module. With green are represented bits involved in or obtained from parallel computation.

3.2. Receiver module

To receiver module, the system contains the detection and correction parts. The hardware block diagram of this system is presented in the Figure 3 b.

The serial computing module is the same as to the encoder. This executes a division between word code polynomial (message and check sum) and generator.

The division is performed using a serial algorithm, as to the encoder, while the data packet is received from communication channel. After the division results a remainder. Depending of its value, the wrong bits (their positions) are identified in data packet.

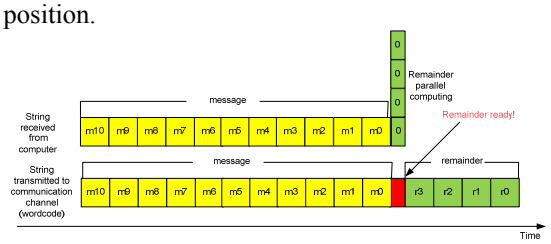
The erroneous bits detector is practically a decoder, with a bit string output with size of data packet size. It takes the remainder and set high 1 logic output bit which correspond with wrong bit positions. If the remainder is 0 then all bits are correct so the output string from detector is 0. Else, detector convert remainder value in a bit string with 1 to the positions where are wrong bits.

The string with wrong bits positions is transmitted to the errors corrector module. It consists in a network of Boolean XOR circuits which can invert from original message bits which correspond to 1 position from detector output.

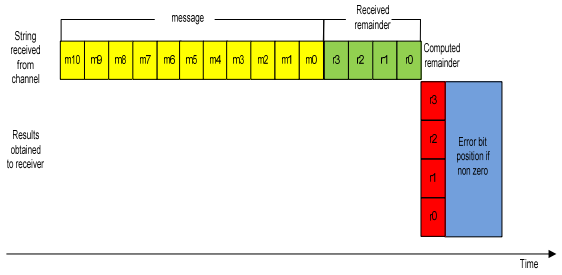
In Figure 4 b, the data flow to the reception is illustrated. With yellow are represented message bits and with green check sum bits. Both the message and check sum (word code) are serial processed.

To apply correction is necessary to save the entire message. This is made in a shift/parallel register with the dimension of the message (see Figure 3 b).

Each received bit is saved here when is received. Next, for correction, information from the register reaches into a network of XOR gates with the binary string provided by the memory with error bit position.



a)



b)

Figure 4. Data flow to the encoder (a) and decoder (b)

The experiments described in this paper refer to identify and correcting one single bit error, but they can be extended by storing others remainder/bit error pair and increasing the remainder degree.

4. Experimental results

Experimental system is presented in Figure 2 as block diagram. Both the encoder and decoder are integrated inside an FPGA chip Xilinx Spartan 3 XC3S400.

This is a very cheap family of FPGA that can be used in commercial applications. In Figure 5 there are illustrated pictures of the experimental system.

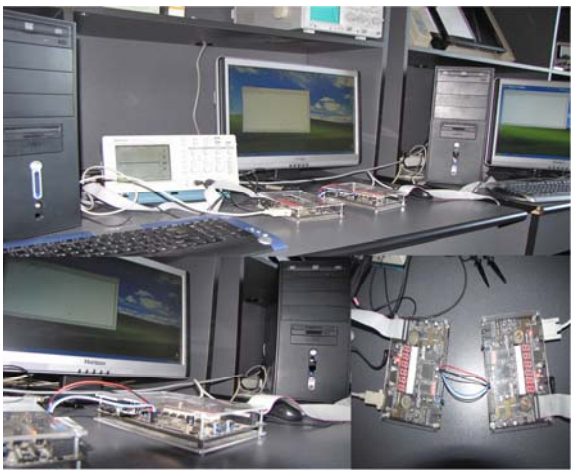


Figure 5. Pictures of the experimental system

In the top side of the figure we can see the computers used as transmitter and receiver and the FPGA boards where are implemented the encoder and decoder. In bottom side of the figure are illustrated the computer and attached board and two boards while communicating using RS 232 interface with computers and a 2 wire interface as communication channel.

The FPGA circuit is integrated inside a development board (Altium Live Design). The board already contains a RS232 converter to communicate with PC and more general purpose Input/Output extension pins. A functional schematics of the experimental module is presented in Figure 6.

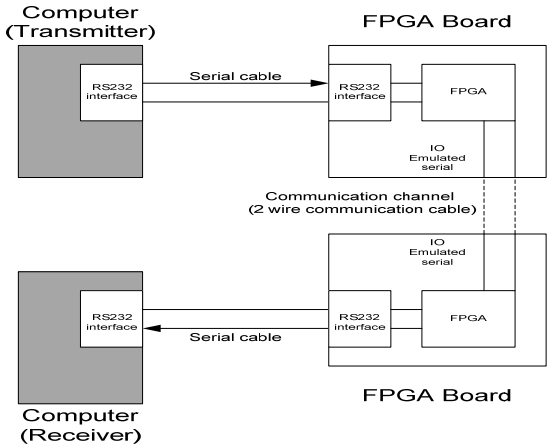


Figure 6. Experimental system operation diagram

From the transmitter computer is sent a message. This arrives, via RS232 interface, to the FPGA board. In FPGA is the encoder which computes the check sum and send it, attached to the message, to communication channel. The communication between encoder and decoder (FPGA boards) is made using a two wire serial interface, emulated on two I/O pins.

To the receiver is taken the word code and then is computed the error inside message, as we presented in the previous sections. The corrected message is transmitted to the receiver computer, via RS232.

The serial interfaces are used only to test encoder and decoder systems inside a complete communication system. They can be replaced with any serial communication environment. To the evaluation of communication speed, are take into account only response times from encoder and decoder, without the propagation time through the serial interface.

The evaluation of the system presented in this paper is performed for a message size of 11 bits and a generator of 4 degree:

$$g(x) = x^4 + x + 1 \qquad (10)$$

The remainder is 3 degree and is represented as a 4-bits size binary string. Therefore, the size of a word code is 15 bits.

After the hardware synthesis of the transmitter system, the following report was obtained:

Device utilization summary:

Selected Device : 3s400fg456-4

Number of Slices:	**11**	**out of**	**3584**	**0%**
Number of Slice Flip Flops:	**16**	**out of**	**7168**	**0%**
Number of 4 input LUTs:	**14**	**out of**	**7168**	**0%**
Number of bonded IOBs:	**8**	**out of**	**264**	**3%**
Number of GCLKs:	**1**	**out of**	**8**	**12%**

Timing constraint: Default period analysis for Clock 'clk'
Clock period: 4.919ns (frequency: 203.293MHz)
Total number of paths / destination ports: 42 / 14

To the receiver system, it is obtained (after synthesis):

Device utilization summary:

Selected Device : 3s400fg456-4
Number of Slices: 23 out of 3584 0%
Number of Slice Flip Flops: 31 out of 7168 0%
Number of 4 input LUTs: 36 out of 7168 0%
Number of bonded IOBs: 18 out of 264 6%
Number of GCLKs: 1 out of 8 12%
Timing constraint: Default period analysis for Clock 'clk'
Clock period: 4.919ns (frequency: 203.293MHz)
Total number of paths / destination ports: 57 / 29

As can be seen from the two reports, the chip area occupied in a 3s400FG456 FPGA is very small. Therefore, this systems can be integrated on the same chip with others modules.

Besides the high degree of integration, another major advantage is the decrease of computing time over a software solution.

In the Table 1 there are compared the computation time of the hardware system and the estimated response time, when the detection and correction algorithm runs on a computer with a processor which work at 2,8 GHz clock frequency.

Table 1. A comparison between hardware and software processing at transmitter.

k	n	v_t	$T_{hardware}$	$T_{software}$
11	15	0,73	73,785 ns	80,3 ns

To receiver, the computation speed of the system compared with software implementation is higher, because to the hardware system a parallel module is used to correct the error.

Table 2. A comparison between hardware and software processing at receiver

k	n	v_t	$T_{hardware}$	$T_{software}$
11	15	0,73	73,785 ns	107,1 ns

5. Conclusion

The usage of error correcting control is very important in a modern communication system. The BCH codes are being widely used in communication networks, computer networks, satellite communication, magnetic and optical storage systems.

We have presented in this paper the prototyping of a BCH encoder and decoder using a Field Programmable Gate Array (FPGA) reconfigurable chip. We implemented the BCH code in a 3s400FG456 FPGA.

In this implementation we used 15 bits-size word code and the results show that the circuits work quite well. The FPGA implementation of BCH codes leads to a high calculation rate using parallelization. So, we reduce the computation time of control sum at transmission and detection and correction error to receiver.

Our solution can be used for data transmission in real time application. The difference between $T_{hardware}$ and $T_{software}$ increases with the size of word code. A future research direction is to extend our system in order to correct more error bits for larger word code.

6. References

[1] M.Y. Rhee - "*Error Correcting Coding Theory*", McGraw-Hill, Singapore, 1989.

[2] S. Lin, and D.J. Costello Jr. - "*Error Control Coding*", Prentice-Hall, New Jersey, 1983.

[3] J. Rose S.D. *Brown, R.J. Francis – "Field Programmable Gate Arrays*", Kluwer Academic Publishers, 1992.

[4] T. Fogarty, J. Miller, and P. Thompson - "*Evolving digital logic circuits on Xilinx 6000 family FPGAs*," in Soft Computing in Engineering Design and Manufacturing, P. Chawdhry, R. Roy, and R. Pant (eds.), Springer: Berlin, 1998, pp. 299–305.

[5] C. Anton, L. Ionescu, I. Tutănescu, A. Mazăre, G. Şerban - "*FPGA-implemented CRC Algorithm*", International Conference "Applied Electronics", 9-10 September 2009, p. 25-30, ISSN 1803-7232, ISBN 978-80-7043-781-0, IEEE Catalog Number CFP0969A-PRT, University of West Bohemia, Pilsen, Czech Republic, 2009.

