Implementation and Analysis of an Error Detection and Correction System on FPGA

Constantin Anton, Laurenţiu Mihai Ionescu, Ion Tutănescu, Alin Mazăre, Gheorghe Şerban
University of Piteşti, Romania

Abstract

This paper presents a solution to design and implement a hardware error detection and correction circuit using associative memories. This type of memory allows search of a binary value stored, having input data a partial (or modified) amount of this value. This property can be used in communication, for detection and correction of errors. In our analysis, the obtained experimental results were compared with performances of other hardware systems.

1. Introduction

Usage of error correcting control is very important in the modern communication system. BCH codes (Bose, Chaudhuri, and Hocquenghem) are 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 associative memory.

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 the 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 $x^{2^m - 1} + 1$ [2].

For our application we use generator polynomial:

$$g(x) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$$

which can correct 3 erroneous bits and detect 6 errors.

2. Hardware circuits for errors detection and correction

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 that of 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 circuits (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. On the other hand, they allow the parallel structures implementation, with smaller response times than a system with microprocessor.

FPGA is organized as a 2D array of configurable logic blocks (CLB). They can be interconnected via global bus, which is realized between configurable blocks, or by local bus, which is realized within a CLB. In turn, each CLB circuit has a total of 4 slices.

A slice contains a logic functions generator (can implement any logic function with 4 inputs and one output), an arithmetic logic, flip-flops and multiplexers to connect with other neighboring slices. Because this architecture, FPGA circuits are especially useful in applications that rely on network logic cells.

Associative memory is a type of memory that can be addressable by the content. Instead to know an address, to access a location is sufficient to know the content part of the location. Using partial content is a search until it finds the memory location it contains. Thus, it is partially associated content (data entry) with full content value (found at that location).

There are many works that have used FPGA circuits to detect and correct errors. This is because these devices are affordable and can be purchased at low prices. Also, development tools for these circuits are available.

So they were made in FPGA implementation of algorithms for calculating the checksum (CRC) and automatically attach it to the packet that is transmitted on the communication channel [5], implementation of BCH error correction codes [6], implementation of SR-ARQ hybrid algorithms [7], algorithms for checking parity checksum type [8] and the class of quasi-cyclic LDPC codes [9]. In all these examples, we have machines that perform calculations implemented in FPGA.

Associative memories were originally present especially in recognition forms or patterns of sounds. The problem in implementing these kinds of memories is related to the degree of complexity for one location. In general, associative memories are related to artificial neural networks (Hopfield network). They are used in recognizing images from partial images. There are many works where are proposed implementations of Hopfield networks as associative memories in FPGA.

Usage of hardware implemented associative memories for error detection in communications is rare, because this solution has a higher degree of complexity in implementations than other solutions. Associative memories are used with cellular automata [10].

Our solution has the following characteristics:

- firstly, we use our own version of implemented hardware associative memory which has a simplified structure composed of combinational networks; thus, it is comparable, in terms of the complexity of other solutions that rely on automatic calculation.

- secondly, simplification of our structure allows lower response times.

In this way we can use the advantages of associative memory in identifying and correcting errors

The system proposed in this paper is based on the use of reconfigurable FPGA circuits for hardware implementation of error detection and correction algorithms and is presented in Figure 1.

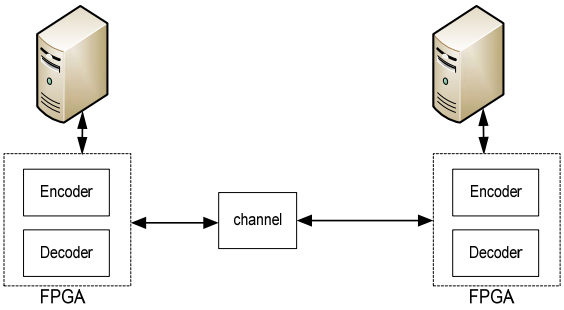


Figure 1. Communication system with hardware error detection and correction function.

2. Encoder and decoder: design and implementation

We designed the encoder and decoder using dedicated Boolean circuits. Thus, the encoder will be attached as a physical device to any system which transmits data to a communication channel, while the decoder will be attached to the system which receives the data. In this chapter we described the operation of the two systems and our design method proposed for them.

2.1. Encoder

The operation of encoder is illustrated in Figure 2.

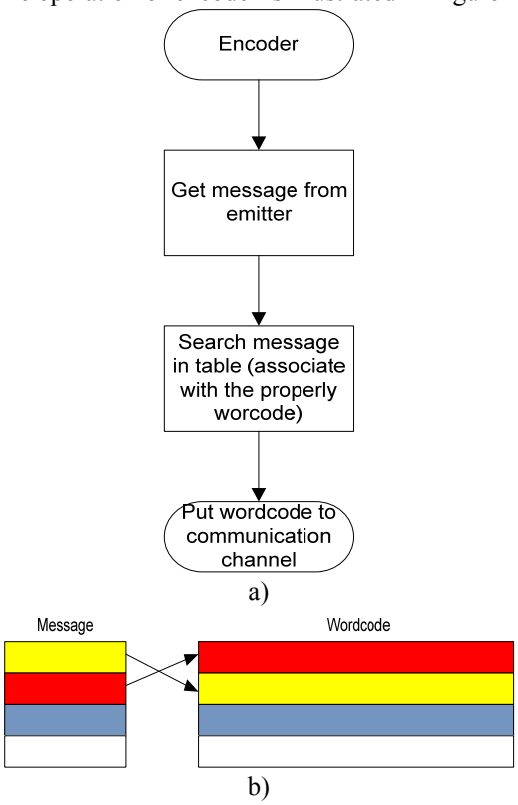


Figure 2. Encoder: a) operation flowchart; b) association between message and BCH wordcode to the encoder.

First, the message is received from the emitter system. This can be received serial (Ethernet or USB) or parallel if the emitter uses a data protocol. The experiments were performed by using 5-bits size message words (thus can be encoded letters from Latin alphabets).

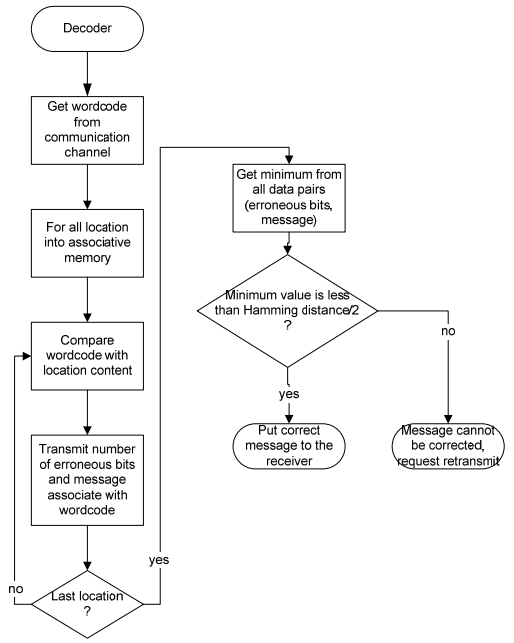
Each message has associated a wordcode. All wordcode have Hamming distance between them equal with 7 and are transmitted to the communication channel. They allow correcting up to 3 erroneous bits.

The association table to the encoder was generated ”off line” by using a computer application and implemented as a simple binary decoder (see Figure 2b). Thus, this design method aims to use a reduced number of hardware resources.

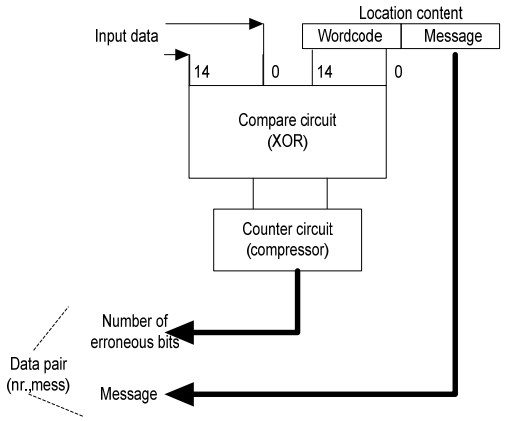
2.2. Decoder

The decoder has a more complex structure. This is because it contains the errors’ detection and correction functions. The decoder can match received words with expected wordcodes but, in some cases, mistakenly received words could not be found in any of those which are stored.

The error correction actually identifies the correct wordcode. This involves the finding the closest word in memory. Once identified it will determine what the message is received. The tasks which are performed at the receiver are presented in the following chart:



a)



b)

Figure 3. Decoder: a) operation flowchart; b) the structure of location in associative memory.

First, the wordcode is taken from the communication channel. In our experiments wordcode has a size of 15 bits.

Then, this wordcode is compared with all stored words. It is a binary comparison.

Finally, we know which erroneous bits are. Each memory cell counts the erroneous bits and we can determine minimal value from all location. The location which contains the wordcode with minimum erroneous bits is associated with the correct message.

The associative memory has a more complex structure that the encoder. Thus, each location consists in a binary comparation circuit, as is illustrated in Figure 3.b, and a bits counter (compressor). This count 1 bit from the comparator output. There is also a register in which are stored the wordcode and the message.

The message with erroneous bits will be transmitted as a 9 bits size data pair (5 bits message and 4 bits number of erroneous bits). All these data pairs, at each location separately, will be compared to determine the minimum.

For this operation we use a combinational network which gets the minimal value of number of erroneous bits, as it is illustrated in Figure 4.

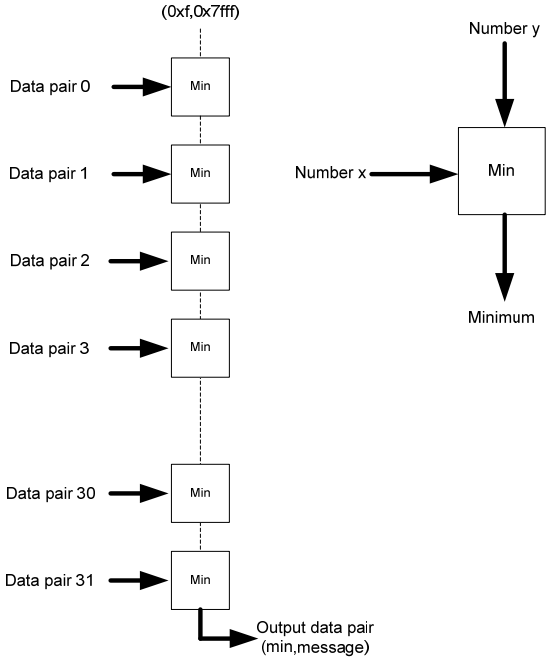


Figure 4. Minimum computation circuit from decoder.

The data pairs from the associative memory are applied to left side (x axis). To y axis we find the data pairs with minimal value of erroneous bits which travel to bottom side of circuit.

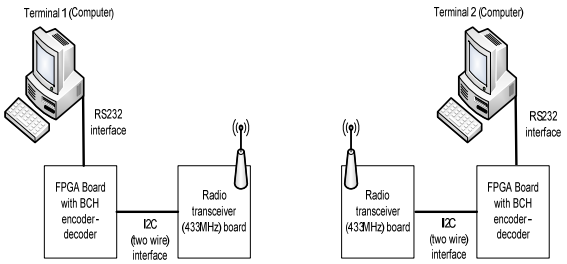
The minimum cell (Min), which consist entire circuit, compare number of erroneous bits from two data pairs and select only data pair with the minimum.

3. Experimental results

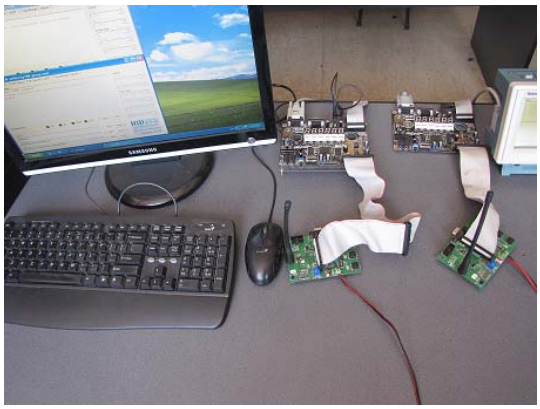
We realized the error correction-detection module testing for radio communication. The test system is illustrated in Figure 5.

It consists of two personal computers. Each computer is connected to an error detection and correction module implemented in FPGA. These

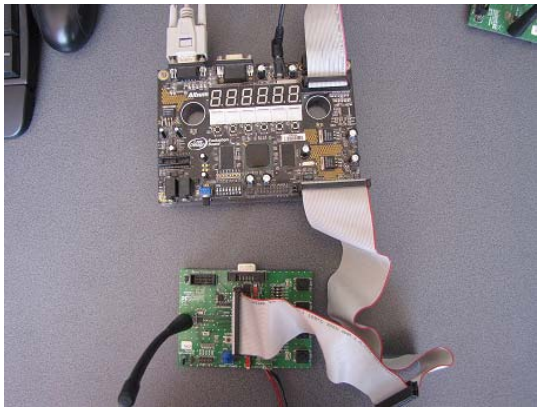
modules were realized on a FPGA Xilinx Spartan 3 circuit XC3S, which is integrated in Altium Live Design board.



a)



b)



c)

Figure 5. The test system: a) block diagram, b, c) Images with the system.

Communication between computer and system is performed via RS232 serial interface. FPGA system (encoder and decoder) is connected to a radio transceiver. The radio transceiver is manufactured by Nordic VLSI. It is a NRF9E5 circuit possessing a 433MHz frequency radio band transceiver (free band) with a long range, up to 110dB. We chose to test the radio communications because they are subject to more noise than conventional wired communications.

In the Figure 5b you can see two communication modules. At the top of the figure are the FPGA systems, on which are implemented encoders and decoders. On the bottom of the Figure 5b there are presented the transceiver systems. The system that is attached to a terminal is represented in the figure 5c.

The communication between computer and FPGA system is realized on an asynchronous serial RS232 interface. Communication between FPGA system and radio transceiver is achieved through a I2C interface (a synchronous serial communication).

The integrated system ensures the errors' detection and correction, which the goal of this project. It also allows the implementation - on the same circuit - of other modules, such as an adapter that can interface two different types of communications.

The adapter is not the subject of our paper, so we performed experiments related to errors' correction and detection module performance.

Further on, we make an analysis of the module in terms of surface area occupied on the chip, implementation and response time. Then, we make a comparative analysis of its performance against other types of modules with errors' correction and detection hardware.

It should be noted that there are also realized software solutions to detect and correct errors, but they assume the existence of a system (computer, etc.) that runs a program which performs the calculations. In terms of response time and used resources, the software solutions cannot be compared to hardware solutions, which are cheaper and faster.

Being a hardware module, we are interested in the occupied area, in order to see which would be the cost of implementation and, of course, the response time that directly influences the communication speed. The circuits used to implement the system were FPGA type circuits. They are currently available in the market and, in terms of price, are comparable with microcontrollers. The application development solutions for FPGA circuits are affordable and cheap, to an extent even greater than for ASIC applications development.

FPGA devices are organized in reconfigurable logic blocks, each configurable block consisting of 4 slices. So, when we analyze the number of used slices, we actually show the used circuit resources. Each slice contains logic functions generators (referred Look Up Table - LUT) and connection logic.

An analysis of occupied LUT in circuit indicates the logic resources used for circuit operation. These resources, together with the interconnection buses, represent the total area allocated in the circuit.

The response time has a direct influence on the speed of communication. Obviously, a package cannot be sent unless the communication channel

coding has been performed, also a package cannot be analyzed for detection and correction of errors in the reception if necessary calculations were not performed. Once the calculations (the encoding and decoding) are realized, the packet can be transmitted or received. After the calculations, the only limitation is the capacity of communication channel. During the process of coding, a 5 bits word is transformed into a 15-bit wordcode that the decoder must turn back in 5 bits.

However, the process of encoding and decoding is parallel. Therefore, the length of wordcode which comes to be transmitted on the communication channel affects only the communication speed on communication channel and not the encoding and decoding speed.

3.1. System chip-area and response time analysis

Since it's a hardware system used to detect and correct errors in communication, items that are considered for performance analysis of this system are given by response time, area occupied on silicon surface and communication speed. We experienced a system which performing BCH encoding 5-bits message words with 15-bit wordcode. This coding allows correction up to 3 erroneous bits and detection up to 6 erroneous bits (Hamming distance between worcode is 7).

Implementation was performed on a FPGA Xilinx Spartan 3 XC3S50, a very low cost circuit (3-5$ chip). A complete communication system integrates both encoder and decoder on the same chip. In the following we will analyze separately.

The encoder consists of a binary decoder. Its structure is very simple, in terms of hardware, occupying only 1% of all chip resources. Circuit simplicity implies a very small response time of 10 ns for each 5-bits size message which means a communication speed of 500 Mbps.

Decoder, more complex than the encoder, still has a simple structure consisting only of combinational circuits arranged in an array. A comparison operation between the input data into the associative memory (received wordcode from the communication channel) and the stored wordcode is performed in parallel for all locations.

Table 1. Results from encoder and decoder implementation - synthesis report.

Circuit	Area on silicon (XC3S50)	Response time[a]
Encoder	Slices 1%(9 of 768), LUT 1%(16 of 1536)	10.617 ns
Decoder	Slices 45%(355 of 768) LUT 40% (631 of 1536)	35.593 ns

a. for a 5 bits word

Therefore, having a silicon area occupied in FPGA chip of 45%, the response time for a 5 bits word is 35.593 ns (7.1186 ns per bit), so the communication speed is approximately 140 Mbps (see Table 1).

Because of BCH coding, there is a reduction in speed of communication with 1/3 (5-bits message is encoded with 15-bits wordcode). For example, if communication speed is 1 Gbps, the real speed is 333 Mbps.

However, in our approach, this reduction is not presented because of parallel computation of entire message word. Thus, we have a real 140 Mbps communication given by encoder and decoder operation.

Our system can be integrated in Ethernet communications, as additional services added to physical layer (see Figure 6).

Interposition of this circuit in Ethernet communication system, will automatically corrected 3 error bit locally. Thus, it provides error correction services for protocols in the highest layer.

The TCP services will be significantly relieved of task that, in other circumstances, would have their back. To increase the communication speed, we can use a "secured" UDP communication, because we have already, from physical layer, error correction services.

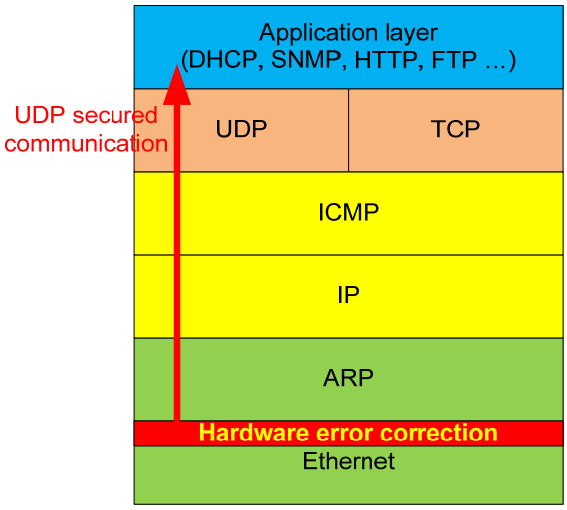


Figure 6. Hardware correction layer services by using our BCH encoder-decoder.

3.1. Comparative analysis

Compared with other structures, based on hardware implementation of error correction circuit, the performances were improved in our system (see Table 2).

Thus, we have a parallel computing circuit which corrects 1 erroneous bit, with no detection, presented in [5]. Because of parallel computing structure, we have here a 149 Mbps communication speed.

Table 2. Comparison with other hardware implemented error correction circuits

Method	Correction (t) and detection (f) ability	Communication speed
Parallel computing, classic	T = 1 erroneous bit F = 0	149Mbps
Hardware BCH SR-ARQ	T = 3 erroneous bits F = 7 erroneous bits	87Mbps
Associative memory BCH	T = 3 erroneous bits F = 6 erroneous bits	140Mbps

Another circuit uses a hardware hybrid BCH SR-ARQ error correction and detection algorithms, presented in [6].

Having 3 erroneous bit correction and 7 errors detection it is a very high correction and detection capacity.

The solution presented in this paper is comparable in performance to hybrid BCH, but at a speed close to first (note that first can correct only 1 single error).

4. Conclusions

Our paper presents a modern method concerning the implementation of an error correcting code with associative memory. Associative memories allow a very easy design for error correcting codes and also obtain good performances in errors' detection and correction. Also, they provide a very high response speed because of parallel processing.

Our method, presented in this paper, consists in storing of a set of wordcodes and of the message associated with them.

The correction is achieved by comparing the received wordcode with all stored wordcodes and by selecting the nearest wordcode, in terms of Hamming distance.

Also, this method allows the detection of errors while the correction is performed, increasing the computing speed.

In future works, we will implement error correcting codes with a higher power correction and detection in FPGA.

Our method improves the capacity of communication channel in radio or Ethernet communications.

5. 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, pp. 299–305, 1998.

[5] C. Anton, L. Ionescu, I. Tutănescu, A. Mazăre, G. Şerban - "*FPGA-implemented CRC algorithm*", Applied Electronics, 2009. AE 2009, ISSN 1803-7232, pp. 25- 29, 2009.

[6] L. Ionescu, C. Anton, I. Tutănescu, A. Mazăre, G. Şerban, "*Hardware implementation of BCH Error Correcting Codes on FPGA*", International Journal of Intelligent Computing Research (IJICR), Volume 1, Issue 3, ISSN 2042-4655, Published by Infonomics Society, pp.148-153, September 2010.

[7] G. Şerban, C. Anton, L. Ionescu, I. Tutănescu, A. Mazăre, "*Implementation of a 64-bit hybrid SR-ARQ algorithm on FPGA*", Proceeding of International Conference on Applied Electronics, ISBN 978-80-7043-987-6, ISSN 1803-7232, IEEE Catalog Number CFP1169A-PRT, pp. 337-340, Pilsen, 7-8 September 2011.

[8] F. Verdier, D. Declercq, "*A Low-Cost Parallel Scalable and Irregular LDPC Decoding*", IEEE Transaction on Communcations, Vol.54, No.7, ISSN 0090-6778, pp.1215-1223, July 2006.

[9] C. Spagnol, W. Marnane, "*A class of quasy-cyclic LDPC codes over GF(2m)*", IEEE Transactions on Communications, Vol.57, No.9, ISSN 0090-6778, pp. 2524-2527, September 2009.

[10] P. Maji, N. Ganguly, P. Chaudhuri, "*Error correcting capability of cellular automata based associative memory*", IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems And Humans, Vol. 33, No. 4, ISSN 1083-4427, pp.466-480, July 2003.

