Multiple Error Correction Controlling

Engineering

KEYWORDS : Error correcting codes, BCH codes, BCH encoder, FPGA

M. Venkataramanamma Master of Technology (ECE) & m.venkataramanamma c/o m.sreenivas reddy, Bhimavaram village, Marripadu mandal, nellore distirct, Andhra Pradesh.

U. Kalpana Reddy Associate Professor and Head of the Department, Progressive Engineering College

ABSTRACT

Using a Field Programmable Gate Array (FPGA) reconfigurable chip the design and implementation of (31,k) binary BCH (Bose, Chaudhuri, and Hocquenghem) encoder is presented in this paper. Amongst the most important cyclic block codes BCH code is one. 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, 21, 2), (31, 16, 3) and (31, 11, 5) have implemented on FPGA. The results show that the systems work quite well.

INTRODUCTION

The digital information and communication systems main objective is to provide trustworthy transmission of information over noisy channels. 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. 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.

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.

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 (2m). Let α be a primitive element in GF (2m).The generator polynomial g(x) of the t error-correcting BCH code of length 2m -1 is the lowest-degree polynomial over GF(2) which has $\alpha,\alpha 2,\alpha 3,...,\alpha 2t$ as its roots[i.e., g(αi)=0 for $1 \le i \ge 2t$)]. Let ϕi(x) be the minimal polynomial of αi. Then g(x) must be the least common multiple of ϕ1(x), ϕ2(x), ... , ϕ2t (x), that is,

$$g(x)= LCM\{ of\ \phi1(x), \phi2(x), ... , \phi2t (x)\} \qquad (1)$$

For any positive integers m ($m \ge 3$) and t (t <2m-1), there exists a binary BCH code with parameters of code words length n=2m-1, number of parity check bits $n\text{-}k \le mt$, and minimum distance (dmin ≥2t+1).

BACKGROUND AND RELATED WORK

There two methods which are used in this proposed system. They are explained below.

Polynomial BCH Codes

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

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:

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

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

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

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

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

It follows from the definition of a t-error-correcting BCH code of length $n=2^m-1$ that-each code polynomial has $\alpha,\alpha^2,\alpha^3,...,\alpha^{2t}$ as roots, c(α^i), for ($1 \le i2 \le t$).

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:

- **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 kth 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) .

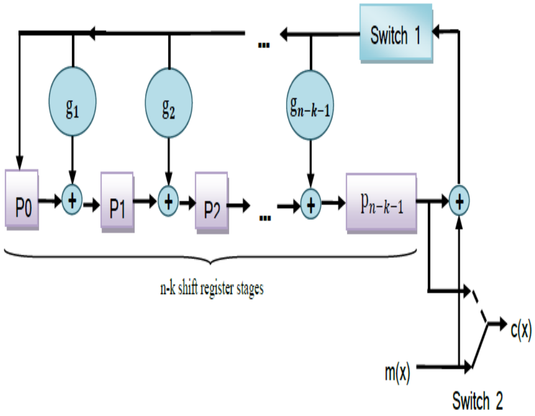

Fig 1: Encoding with an (n-k) stage shift register.

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.

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.

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.

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.

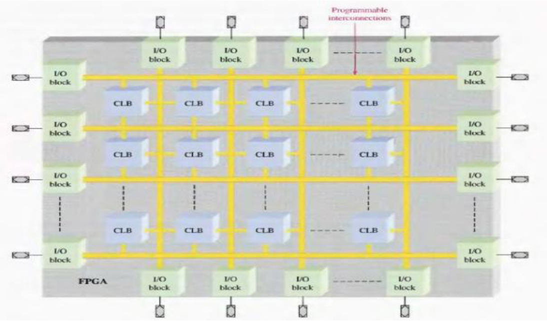

Fig 2: FPGA architecture

MODEL ASSUMPTION

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, 21, 2), (31, 16, 3), and (31, 11, 5).

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+x3+x5+x6+x8+x9+x10

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.(3).

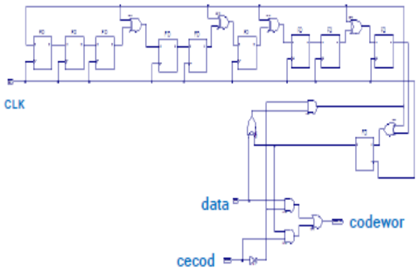

Fig 3: Encoding logic circuit of (31, 21) BCH code implemented on FPGA.

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.

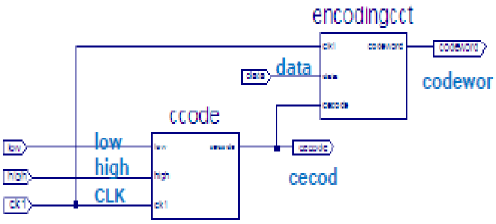

Fig 4: Schematic of (31, 21) BCH Encoder

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+x3+x5+x6+x8+x9+x10

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.(5).

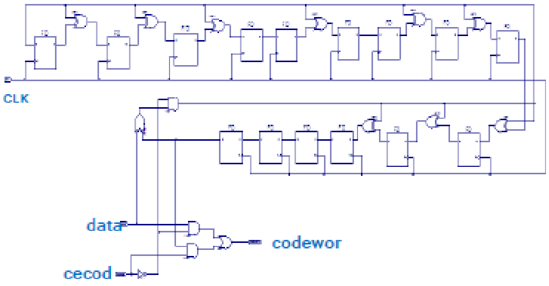

Fig 5: Encoding logic circuit of (31,16) BCH code implemented on FPGA.

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.(6).

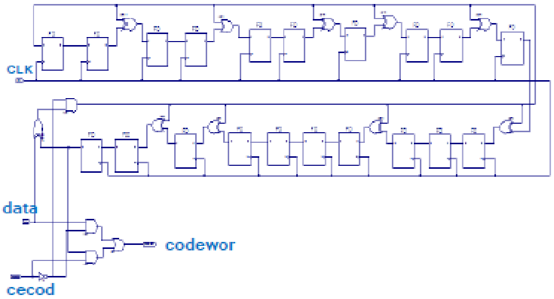

Fig 6: Encoding logic circuit of (31,11) BCH code implemented on FPGA

CONCLUSIONS

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, 21, 2), (31, 16, 3), (31, 11, 5).The three 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.

REFERENCE

[1] Neubauer, J. Freudenberger and V. Kuhn. (2007), "Coding Theory Algorithms, Architectures and Applications" John Wiley & Sons. | [2] T. K. Moon. (2005)," Error Correction Coding John Wiley & Sons, 2005. | [3] A. S. Das, S. Das, and J. Bhaumik, (2013), "Design of RS (255,251) Encoder and Decoder in FPGA." international journal of soft computing and engineering, January, Volume- 2, Issue-6. | [4] J. G. Proakis. (2005), "Digital Communications", Prentice-Hall, 4th edition. | [5] S. Lin, and D.J. Costello Jr . (1983), "Error Control Coding Fundamentals and Applications." Prentice-Hall, New Jersey. | |