

# Implementation of Error Trapping Technique In Cyclic Codes Using Lab VIEW

<sup>[1]</sup>Aneetta Jose, <sup>[2]</sup>Hena Prince, <sup>[3]</sup>Jismy Tom, <sup>[4]</sup>Malavika S, <sup>[5]</sup>Indu Reena Varughese  
 Electronics and Communication Dept. Amal Jyothi College of Engineering Kanjirappally, Kottayam, India

**Abstract:** — In modern communication system, multimedia contents in digital form are used to transmit the information. Probability of error is higher during this transmission which will affect the extraction of data. This paper considers the implementation of cyclic codes encoder and decoder for multimedia contents in the form of sound data using National Instruments LabView software. Cyclic codes can be defined by two parameters, which are code size  $n$  and information bit size  $k$ . LabView is user-friendly software which has many features for designing and prototyping. This project is based on channel coding implementation using cyclic codes in LabView. 16-bit sound data are used as test subjects for cyclic code encoding, decoding and error correction. The design can correct two-bit error in last  $n-k$  positions of the codeword. The design can correct short twobit error in last  $n-k$  position of the codeword.

**Keywords:** LabView, Cyclic codes, encoder, decoder, error trapping, error correction.

## I. INTRODUCTION

The newly developing technology is giving a great change to our world. Nowadays in modern communication system for the ease of transmission, information is sent in digital form. During transmission process there is always probability of error in the communication channel, which may disrupt and change the corresponding data. Thus the extraction of information becomes difficult. So to recover noise contained data, error correction is implemented in modern communication systems by adding redundant bits as parity bits into information bits in the data. By adding these redundant bits, received data which contains error can be recovered to the original data using specific algorithm. Modern communication systems use different error correction methods to remove noise in the received data. Cyclic codes, BCH (Bose, Chaudhuri, and Hocquenghem) codes, RS (Reed Solomon) codes and Convolution codes are being widely used in today's communication systems. Cyclic code is the predecessor of modern error correction method. It can be represented as  $(n, k)$ , where  $n$  is the code size and  $k$  is the number of information bits. Cyclic code is used to correct two-bit error contained in the bit sequence of the data. This paper is a preliminary research of error correction in LabView software. A design in LabView can be easily reprogrammed. The software also comes up with reprogrammable Graphical User Interface (GUI), which is an additional feature to provide information about the operation and result.

## II. BASIC THEORY

A. Galois Field Galois field is defined as  $F=GF(q)$ , where  $q=p^r$  with  $p$  prime. The elements  $0, 1, 2, \dots, p-1$  of  $F$  form a subfield  $F_0$  which is isomorphic to the integer's mod  $p$ , it is known as the prime subfield of  $F$ . For  $F$  to be a field the following conditions must be satisfied:

1. Associative law: For every  $a, b, c$  in  $F$   $a + (b + c) = (a + b) + c$  (1)  $a * (b * c) = (a * b) * c$
2. Commutative Law: For every  $a, b$  in  $F$   $a + b = b + a$  (2)  $a * b = b * a$
3. Distributive Law: For every  $a, b, c$  in  $F$   $a * (b + c) = (a * b) + (a * c)$  (3)
4. Identity: There exists an identity element of 0 for addition and 1 for multiplication which satisfies  $a + 0 = a$  (4)  $a * 1 = a$
5. Inverse: For any  $a$ , there is an element  $-a$  such that  $a + (-a) = 0$  (5)  $a * (a^{-1}) = 1$

B. Cyclic Codes A linear code is said to be cyclic if the  $i$ th cyclic rotation is also a codeword of the same code. In the case of groups, a group is said to be cyclic if it can be generated by successive powers of a given element. A cyclic code can be represented as polynomials defined over a Galois field  $GF(2^n)$ . A codeword  $c = (c_0, c_1, \dots, c_{n-1})$  is

in the form:  $c(x) = c_0 + c_1x + \dots + c_{n-1}x^{n-1}$   $c_i \in GF(2)$  (6)

Polynomials over  $GF(2)$  can be added/subtracted, multiplied/divided in the usual way using binary field arithmetic. Definition 1 A polynomial  $f(X)$  over  $GF(2)$  of degree  $m$  is said to be irreducible if it is not divisible by any polynomial over  $GF(2)$  of degree less than  $m$  but greater than zero. Definition 2 An irreducible polynomial  $f(X)$  of degree  $m$  is said to be primitive if the smallest positive integer  $n$  for which  $f(X)$  divides  $X^n + 1$  is  $n = 2^m - 1$ . The cyclic codes obey some characteristics. The properties of the cyclic codes are the following:

1. For a  $(n, k)$  cyclic code, there exists a generator polynomial of degree  $(n-k)$ , which is the only code vector polynomial of minimum degree  $(n-k)$ .
2. The generator polynomial  $g(x)$  of a  $(n, k)$  cyclic code is a factor of  $x^n + 1$  i.e.,  $x^n + 1 = g(x) \cdot h(x)$  where,  $h(x)$  is parity check polynomial of degree  $k$ .
3. If  $g(x)$  is a polynomial of degree  $(n-k)$  and is a factor of  $x^n + 1$ , then it generates the  $(n, k)$  cyclic code.
4. The code-vector polynomial  $V(x) = D(x) \cdot g(x)$  Where,  $D(x)$  = message vector polynomial of degree  $k$ . This method generates non-systematic cyclic codes.
5. To generate systematic cyclic code, the remainder polynomial  $R(x)$  is obtained from division of  $x^{n-k} D(x)$  by  $g(x)$ . The coefficients of  $R(x)$  are placed in the beginning of the code vector followed by coefficients of  $D(x)$  to get the code vector.

Cyclic codes have a fair amount of algebraic structures that makes it possible to design codes with useful errorcorrecting properties. Also it is easy to implement encoding and syndrome calculating circuits which uses simple shift register with feedback connections and some basic gates.

### Special classes of cyclic codes

#### Bose-Chaudhuri-Hocquenghem (BCH)

For any positive integer  $m \geq 3$  and  $t < 2^{m-1} - 1$ , there exists a binary BCH code CBCH  $(n, k)$  with the following properties:

Block length:  $n = 2^m - 1$   
 Number of message bits:  $k \geq n - mt$

Minimum distance:  $d_{min} \geq 2t + 1$   
 Error correction capability:  $t$  errors in a code vector

#### Reed-Solomon (RS)

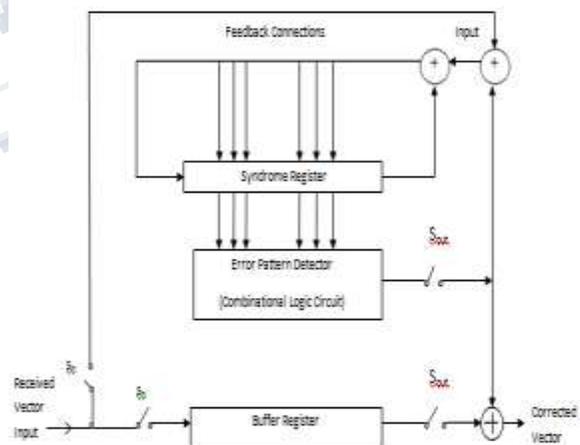
Reed-Solomon codes are also called non binary since coefficients of polynomials are elements in  $GF(2^m)$ . These coefficients can be represented as powers of a primitive element  $\alpha$ .

Block length:  $n = 2^m - 1$   
 Parity-check size:  $n - k = 2t$   
 Minimum distance:  $d_{min} \geq 2t + 1$   
 Error-correction capability:  $t$  errors in a code vector

#### C. Error Trapping Technique

Cyclic codes are extremely well suited for error detection. It can be implemented by simply adding on additional flip-flop to the syndrome calculator. If the syndrome is non-zero, flip-flop sets an indication of error.

In error correction, the decoder has to determine a correctable error pattern  $e(x)$  from the syndrome  $S(x)$  and add  $e(x)$  to  $r(x)$  to determine the transmitted code vector  $c(x)$ .



At the end of the decoding operation errors will have been corrected if they correspond to an error pattern built in to the detector and the syndrome register will contain all zeros. If the syndrome register does not contain all zeros, then an uncorrectable error pattern has been detected.

### III. DESIGN

The encoding and decoding of cyclic codes are implemented using LabView software. In this, Mathscript module is used for simplification process. The sound data used for testing purpose is 16 bit, so that a 31 bit codeword is produced at the receiver side.

The system design is divided into several parts, which are represented in Fig.2 and Fig.3.

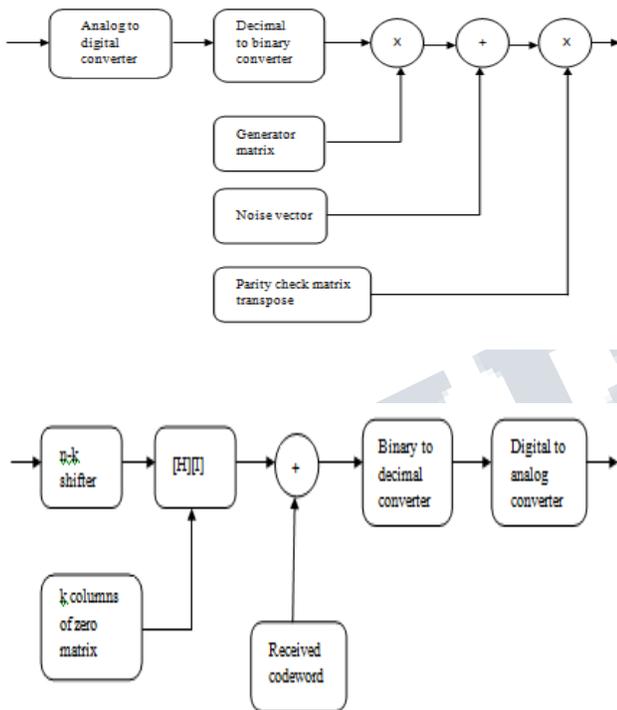


Figure 3. System block diagram (part 2)

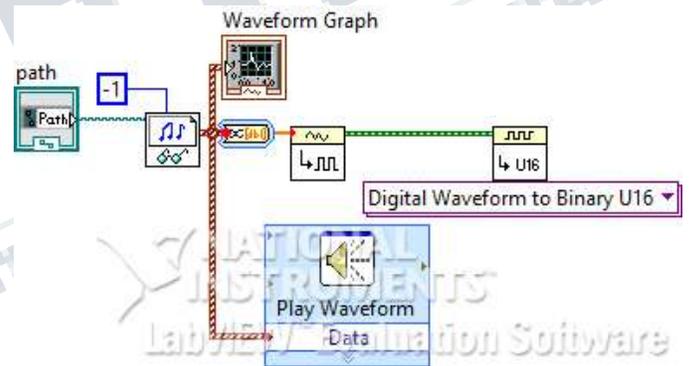
**A. Explanation**

In the initial stage, sound data in .wav file to be encoded is read and converted into analog form. Using analog to digital waveform function in Labview, analog waveform produced is then converted into digital waveform. This digital waveform is then converted into 16 bit unsigned data ,which are represented in decimal numbers(0-65535).This unsigned binary bit are then converted into 16 bit binary sequences. For the production of 31 bit codeword, each 16-bit sequence is multiplied by generator matrix, which consists of 15x16 parity matrix and 16x16 identity matrix. This generated codeword is then added with a 31-bit error code which produces received codeword r.

For the detection and correction of the error codeword, this codeword is multiplied by a parity check matrix which produces syndrome. An error trapping technique is applied using an n-k shifter to estimate error codeword. The resulting vector represented in matrix is combined with k elements zero matrix producing estimated error. Then the received codeword is added with the estimated error to convert the codeword back to the original codeword. The resulting codeword is converted to the original data by taking last 16 columns of the resulting codeword in matrix form. This 16-bit binary data is then converted into decimal form, which is then converted into digital waveform followed by conversion in to the analog waveform. Thus the original sound waveform is recovered.

**IV. RESULTS**

The Fig.4 shows the schematic of analog to digital converter. This depicts the sound data which is read in analog form is converted into digital waveform. A function for sound data playback (play waveform) is also used.



The Fig.5 shows the decimal to binary conversion. In this,16 division operations are performed to convert the decimal number produced to 16-bit binary data

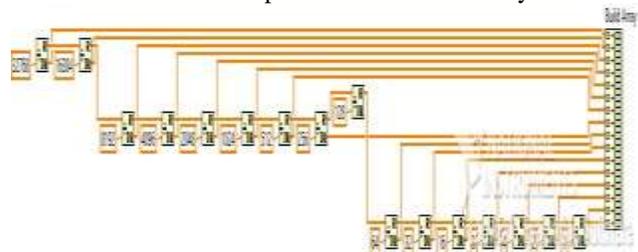


Figure 5. Decimal to binary converter

Schematic for generator matrix production is shown in Fig.6 Generator matrix is formed by multiplying each row of  $I_k$  by  $x^{n-k}$  and dividing the results by  $g(x)$ , and the results are parity bits  $P_{k \times n-k}$  which is combined with  $I_k$ .

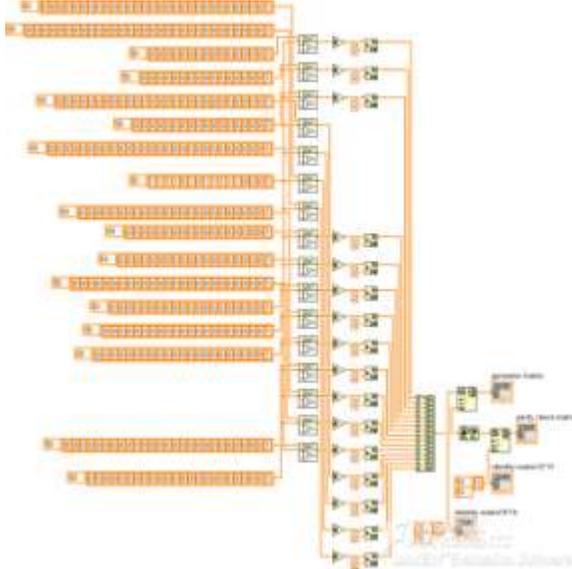


Figure 6. Generator matrix production schematic

Schematic for codeword generation is shown in the Fig.7. This depicts the generation of 31-bit codeword. Mathscript module is used for the simplification of design.

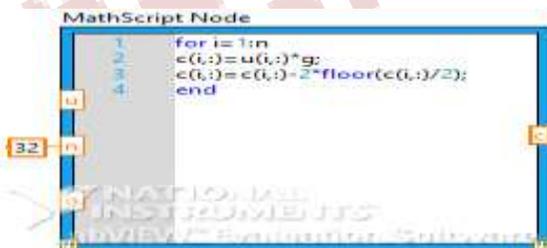


Figure 7. Codeword generation

Schematic used to add two-bit error to the generated codeword is shown in Fig.8 below.

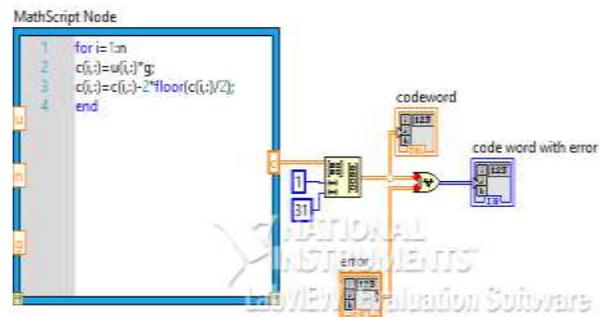


Figure 8. Error generator schematic

The Fig.9 shows the design of parity check matrix  $H$  and  $H^T$ . The parity check matrix is produced by combining  $P^T$  and  $I_{n-k}$ .

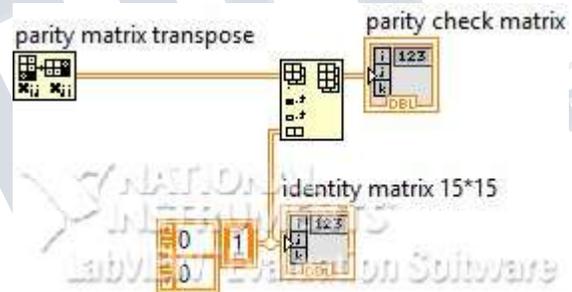


Figure9. Parity check matrix generation

Error estimation and correction schematic is shown in fig.10. The received codeword  $r$  is multiplied with  $H^T$  to produce syndrome. Each  $n-k$  shifted syndrome is combined with  $k \times k$  zero matrix, to produce estimated error in the corresponding codeword and the received codeword then added with the estimated error to recover the original codeword.

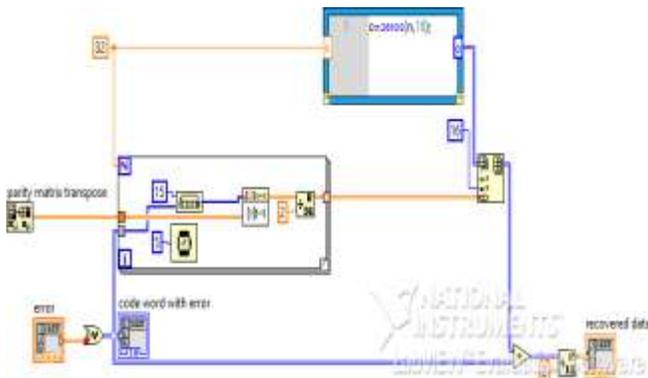


Figure10. Error estimation and correction

The schematic of Binary to Decimal converter is shown in Fig. 11 below. It consists of 16 multipliers, data extraction function, and a modulo-2 adder. Information contained in each codeword is extracted by extracting the last  $k$ -columns of the corresponding codeword. The resulting 16-bit binary data is then converted to decimal representation of the 16-bit sound data

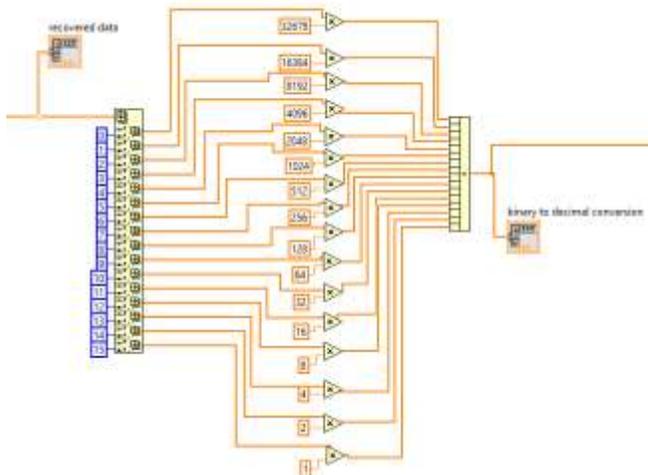


Figure11. Binary to Decimal converter

The schematic of Digital to Analog converter is shown in fig.12 below. The recovered sound data in decimal form is converted back to analog waveform, which is reconstructed back to original .wav sound data.

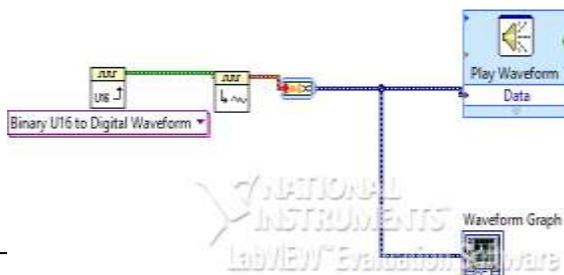


Figure12. Digital to analog converter

The fig.13 below shows the experimental result of this work. The schematic shows the original codeword, message vector, received codeword and the recovered output data.

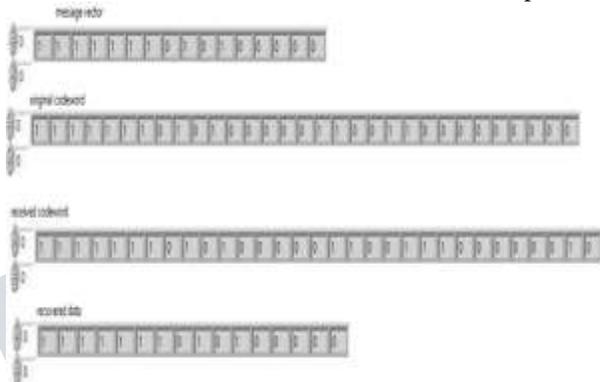


Figure13. Experimental result

### CONCLUSION

In this work, we have used LabView software for the implementation of error correction in a 31-bit codeword. It consists of 15-bit parity and 16-bit information data. The two-bit error contained in the 16-bit sound data is corrected using cyclic code. Error trapping technique is used for the estimation and correction of error in the data. The result shows that the designed system can correct the error contained in the data, as expected.

### ACKNOWLEDGMENT

First and foremost, the authors would like to thank God Almighty for His blessings for this work. The authors would also like to sincerely thank Assistant Professor Ms. Indu Reena Varghese, Assistant Professor Mr. Mathew George and Mr. D. Jobish for their valuable comments.

### REFERENCES

- [1] K.Giridhar, "Information Theory and Coding", Pooja Publications,2006.
- [2] Simon Haykin, "Communication Systems", John Wiley & Sons Inc.,2001.
- [3] Shu Lin and Daniel J. Costello Jr., "Error Control Coding: Fundamentals and Applications", Prentice-Hall, 1983.
- [4] Udayanto Dwi Atmojo, Sri Suning Kusumawardani, "Implementation of Error Trapping Technique in (31,16) Cyclic Codes for Two-bit Error Correction in 16-bit Sound Data Using Labview Software", International Conference on DFMA, 2010.
- [5] Jorge Castineria Moreira and Patrick Guy Farrel, "Essentials of Error- Control Coding", John Wiley& Sons Ltd., 2006.

