Institución Universitaria
SALAZAR Y HERRERA

QUID Nº29, pp. 66-70, julio-diciembre de 2017, ISSN: 1692-343X, Medellín-Colombia

REED SOLOMON ENCODER SIMULATION IN ACCORDANCE WITH G709

(Recibido el 7-08-2017. Aprobado el 27-12-2017)

Hosein Salehi Azad
M.S Student of Telecommunication Engineering,
Faculty of Electrical & Computer Engineering,
Islamic Azad University, North Tehran Branch,
Tehran, Iran

Amir Abbas Heydari
M.S Student of Telecommunication Engineering,
Faculty of Electrical & Computer Engineering,
Islamic Azad University, North Tehran Branch,
Tehran, Iran

Abstract. In transmission of information from one place to other place such as computer network or sending information between different components of a computer system, it is possible that information be changed in the way because of presence of electromagnetic waves and thousands of other reasons. One of the ways to deal with the noises in communication channels and storage systems is to use code control errors. Reed Solomon code is one of the most famous kind of these codes. So the aim of the present study is to simulate Reed Solomon encoder according to G 709. This paper presents an implementation for the encoder and the decoder of optical communication systems, according to the ITU-T G.709 standard. It presents an approach that multiplexes the traditional decoder blocks. The implementation promotes an expressive area reduction in an FPGA. It also presents the circuit implementation in a Virtex 5 FPGA, using software Xilinx ISE 10.1 tools.

Keywords: simulation, Reed Solomon, G 709, Decoder.

Citar, estilo APA: Salehi, H. & Abbas, A. (2017). Reed solomon encoder simulation in accordance with G709. *Revista QUID (29)*, 66-70.

1. INTRODUCTION

Nowadays, storage and digital communications present in all aspects of our lives: satellite data transmission, computer file transmission, radio communications etc.

This transmission of information is done by a channel that is prone to error could cause an error occurs that the received information be different from what is sent. Ones are difficulty able to comprehensive and recognize the occurred errors in storage and transmission of data and information and if error control technique is not used, it will be impossible to store and certain transmission of information. Error in storage and data transformation systems can be resulted from various sources such as random noise, interference, channel fading, or physical impairment. These error channels should be reduced in order to proper and acceptable level of confidence in the storage and transmission of data. Should be used ways to against with possible errors. The main aim to transfer data is to send information by a channel without any error. In the resent years, different technique has been suggested to transfer certainly and surel) Lin and Costellot, 2004).

Encoding theory including error recognition and correction has been studied by researchers for more than forty years and has become more important by the development of new technologies in the storage and transmission systems. Encoding theory is started by shanon's classic article in 1948 (Hocquenghem, 1959).

After that, a lot of research is devoted to find effective planes and schemes to encode digital information that by use of it to be able to transfer the information from a transferring interference channel. Nowadays, error correction codes are applied vastly in application like to numeric recording and to store information in storage discs and return images from the depth of space. The encoding article is very interesting arithmetically because it is broadly based on pure mathematics and specially shows the beauty and power of algebra. The subject of a correction error code is to encode information by adding an extra particular amount to massage that if an error occurs, it is possible to retrieve the original massage again. They initially add a redundancy to sending massage that this process is called encoding which is performed in transmitter and then created error based on redundancy is corrected in the process named decoding which is created in the receiver.

The process output is massage and its redundancy that redundancy is called as the balance. Encoding theory uses different codes to encode data in order to transmission and storage under a communication channel or storage media and then decoding them.

The behavior of these codes varies in different communication and storage channels. Depending on how redundancy is added, there are two categories of codes. One of the codes is named block codes in which encoding and decoding data is created block to block. In contrast another family of codes is convolution codes which perform on continuous flow of data and encoding and decoding operation in this category depend not only on the current data but also on the previous data. In one system, it is possible that operation of some codes be better than other codes.

One of subdiviations of block codes is rotary codes that the most important kind of these codes is Reed Solomon code. These codes are the error preventive correction code kind and nowadays, it is used in spatial and wireless communications, storage disks like mobile phones and digital TVs. Using error control in modern communicational systems is important.

1.1. Reed Solomon Code

Reed Solomon (RS) codes are the subset of BCH codes as well as linear block codes. A particular RS code is specified as RS (n, k) with s-bit symbols (Fang etal, 2009). This means that the encoder takes k s-bit data symbols each time and encodes them into a codeword of n s-bit symbols.

There are $n - k$ parity symbols of s bits each. A RS decoder can correct up to t symbols that contain error in a code-word, where $2t = n - k$. Figure 1 shows a typical RS codeword.

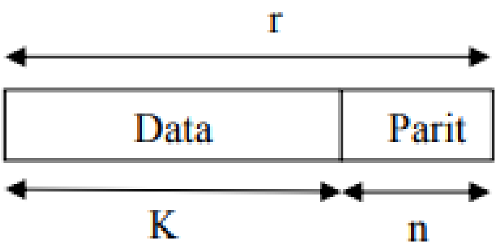


Figure 1. Typical RS

RS codes are particularly suitable to correct burst errors, where a series of bits in the codeword are received in errors.

Reed Solomon coding is a well-known technique for FEC; it has been used for such applications as the Compact Disk. Data is collected into a specific size and is provided with a distinctive checksum of a specific size. This checksum allows not only errors to be detected but also a definite number of errors to be corrected.

The data is systematically coded. It is first sent the original data, followed by the parity symbols, which are calculated according to the polynomial generator. It is used a "2t step" polynomial generator, given by Equation 1 (Morelos-Zaragoza, 2006; Moreira and Farrell, 2006; Wai and Yang, 2006).

$$g(x) = \prod_{i=b}^{b+2t-1}(x + \alpha^{i}) \qquad (1)$$

where *b* is a random number. It must be carefully chosen to avoid increasing the system complexity. The G.709 standard takes *b* as 0.

1.2. Decoder

The decoder is comprised of 4 blocks: Syndrome Calculator, Key Equation Solver, Error Locator and Error Evaluator

The Syndrome Calculator is the first block. It monitors the arriving data to evaluate if data correction processing is necessary. It verifies if an error occurred during the transmission by obtaining all the polynomial roots throughout the data. If the number of errors is smaller than or equal to *t*, then the syndrome is delivered to the next block, the Key Equation Solver. If the number of errors is larger than *t*, an error signal is generated to indicate that no correction is possible. Finally, if no errors are found, the data is delivered without any further processing, besides removing the parity symbols (Le-Ngoct et.al, 1990).

2. IMPLEMENTATION

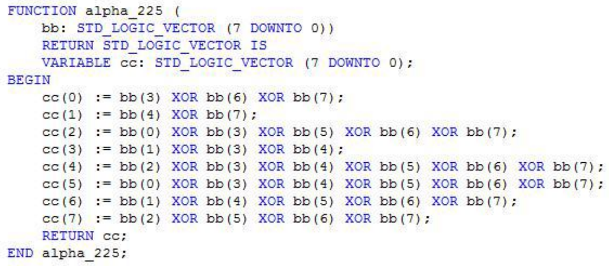
The Reed-Solomon (255,239) was implemented using VHDL hardware description language, which offers high abstraction level during the implementation (Smith et.al, 1993). The VHDL description was simulated and implemented in a XC5VFX70T Virtex 5 FPGA.

2.1. *Encoder*

As previously described, the polynomial generator can be obtained by Equation 1, where *i* is zero, according to the G.709 standard. The obtained polynomial generator is given by Equation 2.

$$g(x) = X^{16} + 59X^{15} + 13X^{14} + 10X^{13} + 104X^{13} + 189X^{12} + 68X^{12} + 209X^{10} + 30^{9} + 8X^{8} + 163X^{7} + 56X^{6} + 41X^{5} + 229X^{5} + 98X^{3} + 50X^{2} + 36X + 39 \qquad (2)$$

An example of VHDL multiplication is shown in Figure 2. The multiplication is implemented as a function since it will be also used in other blocks.

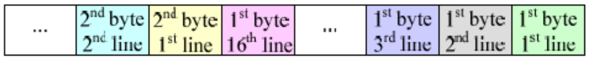


```
FUNCTION alpha_225 (
    bb: STD_LOGIC_VECTOR (7 DOWNTO 0))
    RETURN STD_LOGIC_VECTOR IS
    VARIABLE cc: STD_LOGIC_VECTOR (7 DOWNTO 0);
BEGIN
    cc(0) := bb(3) XOR bb(6) XOR bb(7);
    cc(1) := bb(4) XOR bb(7);
    cc(2) := bb(0) XOR bb(3) XOR bb(5) XOR bb(6) XOR bb(7);
    cc(3) := bb(1) XOR bb(3) XOR bb(4);
    cc(4) := bb(2) XOR bb(3) XOR bb(4) XOR bb(5) XOR bb(6) XOR bb(7);
    cc(5) := bb(0) XOR bb(3) XOR bb(4) XOR bb(5) XOR bb(6) XOR bb(7);
    cc(6) := bb(1) XOR bb(4) XOR bb(5) XOR bb(6) XOR bb(7);
    cc(7) := bb(2) XOR bb(5) XOR bb(6) XOR bb(7);
    RETURN cc;
END alpha_225;
```

Figure 2 Multiplication implemented in VHDL

The example shows the multiplication of a Galois field element (represented by bb) by the constant element α225. The AND logic does not appear in the listing due to the circuit simplification. An AND 0 operation results 0 and therefore is eliminated. An AND 1 operation is represented by the element itself. Therefore, the number of logic gates used to implement an operation depends on the operands. The same operation occurs both at the transmitter and the receiver.

The encoding of line in the OTN frame requires 16 Reed Solomon encoders. Each line contains 3824 symbols that are supplied by the “interleaver” in a multiplexed fashion in 16 sets of 239 symbols each (ITU-T G.709/Y.1331, 2001). The symbols are transmitted in an interleaved fashion, as indicated in Figure 3.



...	2nd byte 2nd line	2nd byte 1st line	1st byte 16th line	...	1st byte 3rd line	1st byte 2nd line	1st byte 1st line

Figure 3 OTN transmission structure

Table 1. Encoding logic utilization

Blocks	Amount Used
Combinational ALUT	2,592
Dedicated Logic Registers	2,320
Block Memory	0

2.2. Decoder

The decoder receives 4,080 symbols sequentially, corresponding to an OTN frame line. The 4,080 symbols, corresponding to 16 channels, are delivered one byte at a time, as previously shown in Figure 3.

Since the system runs at 166 MHz and two symbols arrive at each 8 clock cycles, the FIFO takes 12 µs to load one frame line. The implemented decoder takes a maximum of 10 µs to process the error correction. The timing diagram in Figure 4 represents that situation.

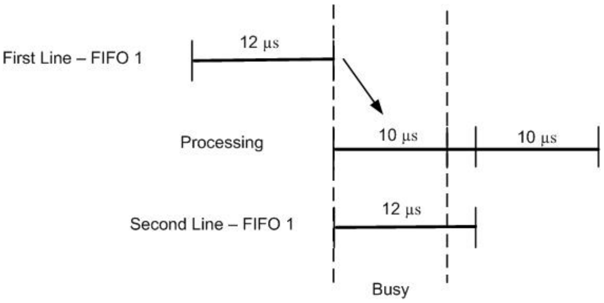


Figure 4. Decoding timing diagram.

From the analysis of the timing diagram of Figure 4, it can be observed that a single set of 16 decoders would not be enough to implement the system. The decoder can start its operation only after the FIFO is completely loaded. Therefore, while it processes one line, the other line is being transmitted and consequently must be stored. Another set of 16 decoders is necessary for the second line, requiring a large FPGA area.

The proposed solution in this work is the duplication of the Syndrome and the FIFO blocks, while maintaining a single set of the other blocks, as indicated in Figure 5.

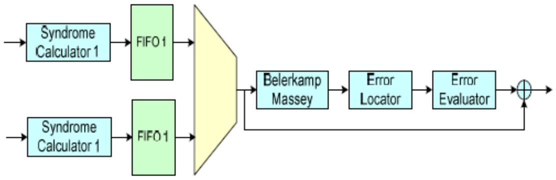


Figure 5. Modified RS decoder

Initially, each line passes through a Syndrome Calculator and them it is loaded into a FIFO memory. If necessary, each line is processed by the Belerkamp-Massey, Error Locator and Error Evaluator blocks. While the line is being processed, the other memory is loading another incoming line.

The proposed decoding architecture saves 16 Berlekamp- Massey blocks, 16 Error Locator and 16 Error Evaluator blocks. These blocks require large FPGA area, as shown by the summary presented in Table 2.

Table 2. Decoding comparison

Blocks	Total Available	32 RS	16 RS
LTUs	44,800	18,383	12,270
Registers	44,800	35,144	23,991
Block Memory	148	16	16
Response Time	-	5 ns	5.1 ns

As can be observed, the proposed decoding architecture saves FPGA area. The number of registers is reduced by 14% and the number of look-up tables is reduced by 25%, while delay is increased by just 0.1 ns (increase of 2%) due to the inclusion of a multiplexer. The delay is acceptable, considering that a clock cycle at 166MHz requires 6 ns. Therefore, the OTN frame decoding requires just 16 RS decoders instead of 32.

The 4080 symbols are decoded by 16 RS (255,239) decoders that produce 3,824 symbols, which correspond to an OTN frame line without the parity symbols.

3. CONCLUSION

Transmission and storage of information by a channel that is prone to error could cause error occur that received information be different with the sending information. Because of destructive effect and the present of more noises in a channel, error controlling strategy should be used. In this article Reed-Solomon error correction code is introduced.

This article presented a new Reed-Solomon decoding structure in accordance with G 709 that can save a lot of FPGA area. The saved area represents a cost reduction for the system since a smaller and cheaper FPGA can be used.

REFERENCES

A.Hocquenghem, (1959). “Codes correcteurs d’erreurs,” Chiffres, 2: 147- 56.

Fang Xu, Ali Khalighi, (2009). Patrice Causs´e and Salah Bourennane”Channel coding and time-diversity for optical wireless links” Vol. 17, No. 2 / OPTICS EXPRESS 875/19 January.

ITU-T G.709/Y.1331, Interfaces for Optical Transport Network (OTN), (2001).

J. C. Moreira and P. G. Farrell, (2006). Essentials of Error-Control Coding, p. 2, 166, 3, 166, John Wiley & Sons Ltd, Chichester.

K. C. C. Wai and S. J. Yang, (2006). Field Programmable Gate Array Implementation of Reed-Solomon Code, RS (255,239), New York.

Lin, S., and Costello, D.J.: (2004). “Error control coding: fundamentals andapplications” (Prentice-Hall Inc.).

R. H. Morelos-Zaragoza, (2006). The Art of Error Correcting Coding, p. 1, 74, John Wiley & Sons Ltd, Chichester.

S. Smith, M. Benaissa and D. Taylor, (1993). In VHDL (Very High Speed Integrated Circuits Hardware Description Language) - Applications and CAE Advances/1993, High Level Synthesis of an (N,K) Reed-Solomon Encoder Using VHDL, London.

T. Le-Ngoct, M. T. Vot, B. Mallettt and V. K. Bhargava, in Military Communications Conference, (1990). MILCOM '90, Conference Record, A New Era/1990, p. 121-125, A gate-array-based programmable Reed-Solomon codec:structure-implementation-applications, Monterey, CA.