Implementation of Reed Solomon Encoding Algorithm

P.Sunitha[1], G.V.Ujwala[2]

[1 2] Associate Professor, Pragati Engineering College,ECE

Abstract- In this paper, VHDL implementation of the Reed Solomon (RS) encoding algorithm is described briefly. The efficient RS encoder design is based on the algorithm widely used in communication system Taking RS encoder in DVB system for example, we introduce the structure of RS encoder. And an improved algorithm on Galois Field multiplier is proposed which saves the numbers of the XOR gates. The proposed architecture implements various programmable primitive polynomials. A lot of VLSI implementations have been described in literature.

Keywords- Reed Solomon, Galois Field, DVB, ECC, SNR

1 Introduction

An important function of any modern digital communications system is error control coding (ECC). Such coding is the field of communications that deals with techniques for detecting and correcting errors in a signal. Though used in a variety of systems, ECC is especially useful in wireless communications systems. Such systems typically operate with a low signal-to-noise ratio (SNR) and suffer from distortion because of a multipath channel. Typical communications systems use several codes that are suited to correcting different types of errors. RS codes are the most powerful in the family of linear block codes and are arguably the most widely used type of error control codes [6]. RS code is a type of Forward Error Correction (FEC) code and it is a non-binary, linear and cyclic block error correcting code. The basic idea of FEC codes is to systematically add redundancy at the end of the messages so as to enable the correct retrieval of messages despite errors in the received sequences [3].

RS codes which was discovered by Irving S. Reed and Gustave Solomon in Lincoln laboratory of MIT in 1960, is a kind of multi-BCH (Bose-Chaudhuri-Hocquenghem) code with strong error correction capability, which is currently one of the most effective and widely used for error control codes [4]. RS codes have been extensively applied in deep space communication systems and storage systems due to their excellent capability of correcting both random and burst errors. RS Codes have been widely used in a variety of error correcting systems found just about anywhere including: Storage devices, Wireless communications, Digital TV, Satellite communications, Broadband Technologies, etc [1].The symbols in RS coding are elements of a finite field or Galois Field (GF). A GF is a set that consists of finite number of elements. Galois field arithmetic is used for encoding and decoding of RS codes. Galois field multipliers are used for encoding the information block. The encoder attaches parity symbols to the data using a predetermined algorithm before transmission.

II. Reed Solmon Encoding

- *Operation*

The Reed-Solomon encoder reads in k data symbols, computes the n-k parity symbols, and appends the parity symbols to the k data symbols for a total of n symbols. The encoder is essentially a 2t tap shift register where each register is m bits wide. The multiplier coefficients are the coefficients of the RS generator polynomial. RS encoder design should effectively perform

the following two operations, namely division and shifting. Both operations can be easily implemented using Linear-Feedback Shift Registers (LFSR) [8].

Reed-Solomon codes are *nonbinary cyclic* codes with symbols made up of *m*-bit sequences, where *m* is any positive integer having a value greater than 2.R-S (*n*, *k*) codes on *m*-bit symbols exist for all *n* and *k*
For which

$$0 < k < n < 2m + 2$$

where *k* is the number of data symbols being encoded, and *n* is the total number of code symbols in the encoded block. For the most conventional R-S (*n*, *k*) code, (*n*, *k*) = (2*m* - 1, 2*m* - 1 - 2*t*)

where *t* is the symbol-error correcting capability of the code, and *n* - *k* = 2*t* is the number of parity symbols. An extended R-S code can be made up with *n* = 2*m* or*n* = 2*m* + 1, but not any further.Reed-Solomon codes achieve the *largest possible* code minimum distance for any linear code with the same encoder input and output block

$$dmin = n - k + 1$$

The code is capable of correcting any combination of *t* or fewer errors, where *t* can be expressed as

$$t = [d-1]/2$$

$$= [n-k]/2$$

Reed-Solomon Error Probability

The Reed-Solomon (R-S) codes are particularly useful for *burst-error correction*;that is, they are effective for channels that have memory. Also, they can be usedefficiently on channels where the set of input symbols is large. An interestingfeature of the R-S code is that as many as two information symbols can be added toan R-S code of length *n* without reducing its minimum distance.

This extended R-Scode has length *n* + 2 and the same number of parity check symbols as the originalcode. The R-S decoded symbol-error probability, *PE*, in terms of the channelsymbol-error probabilityk lengths. For nonbinary codes, the distance between two codewords is defined (analogous to Hamming distance) as the number of symbols in which the sequences differ.

Why R-S Codes Perform Well Against Burst Noise

Consider an (*n*, *k*) = (255, 247) R-S code, where each symbol is made up of *m* = 8 bits (such symbols are typically referred to as *bytes*). Since *n* - *k* = 8, Equation (4)indicates that this code can correct any four symbol errors in a block of 255. Imagine the presence of a noise burst, lasting for 25-bit durations and disturbing one block of data during transmissionData block disturbed by 25-bit noise burst.

In this example, notice that a burst of noise that lasts for a duration of 25 contiguous bits must disturb exactly four symbols. The R-S decoder for the (255, 247) code will correct *any* four-symbol errors without regard to the type of damage suffered by the symbol. In other words, when a decoder corrects a byte, it replaces the incorrect byte with the correct one, whether the error was caused by one bit being corrupted or all eight bits being corrupted. Thus if a symbol is wrong, it might as well be wrong in all of its bit positions.

This gives an R-S code a tremendous burst-noise advantage over binary codes, even allowing for the interleaving of binary codes. In this example, if the 25-bit noise disturbance had occurred in a random fashion rather than as a contiguous burst, it should be clear that many more than four symbols would be affected (as many as 25 symbols might be disturbed). Of course, that would be beyond the capability of the (255, 247) code.

R-S Performance as a Function of Size, Redundancy, and Code Rate

For a code to successfully combat the effects of noise, the noise duration has to represent a relatively small percentage of the codeword. To ensure that this happens most of the time, the received noise should be averaged over a long period of time,

reducing the effect of a freak streak of bad luck. Hence, error-correcting codes become more efficient (error performance improves) as the code block size increases, making R-S codes an attractive choice whenever long block lengths are desired [5]. This is seen by the family of curves , where the rate of the code is held at a constant 7/8, while its block size increases from n = 32 symbols (with m = 5 bits per symbol) to n = 256 symbols (with m = 8 bits per symbol). Thus, the block size increases from 160 bits to 2048 bits.

Finite Fields

In order to understand the encoding and decoding principles of nonbinary codes, such as Reed-Solomon (R-S) codes, it is necessary to venture into the area of finite fields known as *Galois Fields* (GF). For any prime number, p, there exists a finite field denoted GF(p) that contains p elements. It is possible to extend GF(p) to a field of pm elements, called an *extension field* of GF(p), and denoted by GF(P^m), where m is a nonzero positive integer. Note that GF(p^m) contains as a subset the elements of GF(p). Symbols from the extension field GF(2^m) are used in the construction of Reed-Solomon (R-S) codes. The binary field GF(2) is a subfield of the extension field GF(2^m), in much the same way as the real number field is a subfield of the complex number field. Besides the numbers 0 and 1, there are additional unique elements in the extension field that will be represented with a new symbol α. Each nonzero element in GF(2^m) can be represented by a power of α. An *infinite* set of elements, F, is formed by starting with the elements {0, 1, α}, and generating additional elements by progressively multiplying the last entry by α,

F = {0, 1, α, α2, …, α j, …} = {0, α0, α1, α2, …, α j, …}

- *Architecture*

The encoder is implemented by LFSR. The coefficients of are derived from generator polynomial. Fig.1 shows basic block diagram of RS Encode.

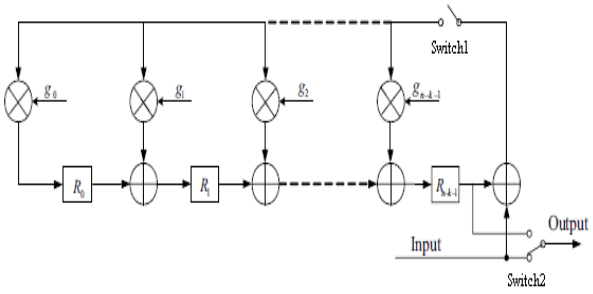

Fig.1. Block diagram of RS Encoder

The output codeword c(x) is systematically encoded and defined in as a function of the transmitted message m(x), the generator polynomial g(x) and the number of parity symbols 2t .

The 2t no. of parity symbols can be calculated using generator polynomial and Galois field multiplier.

The generator polynomial of degree 2t is given by

III. Work Done & Results

(i) RS (15, 11) encoder:

Symbol length m = 4 bits

Error correcting capability t = 2

Simulation result

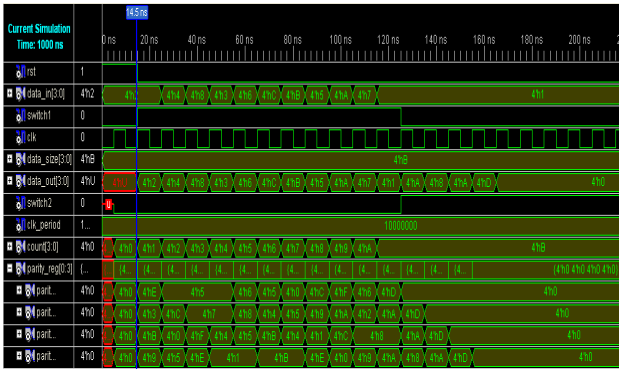

Fig.4. RS (15, 11) Encoder output

(ii) RS (255, 247) encoder:

Symbol length m = 8 bits

Error correcting capability t = 4

Simulation result

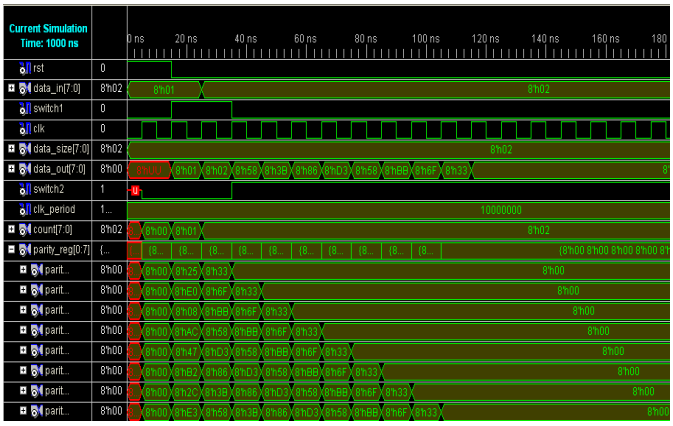

Fig.5. RS (255, 247) Encoder output.

(iii). RS (255, 239) encoder:

Symbol length m = 8 bits

Error correcting capability t = 8

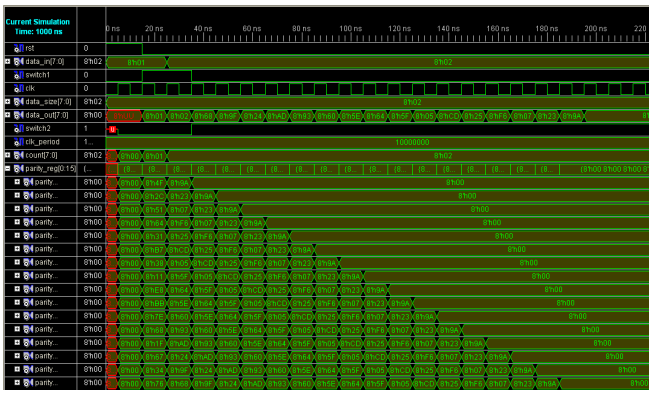

Fig.6. RS (255, 239) Encoder output

iv. Conclusion

This paper briefly describes the RS encoder of DVB systems. And the design method of Finite Galois Field multiplier in RS(15,11), RS(255,247) and RS(255,239) encoder circuit is analyzed in detail. Finally, an improved scheme is proposed. This method compared with the idea of local optimization in [1] saves more hardware resources.

Reference

[1] Xiaojun Wu; Xianghui Shen; Zhibin Zeng "An improved RS Encoding Algorithm" Consumer Electronics Communications and Networks (CECNet), 2012 2nd International Conference on Digital Object Identifier, Page(s): 1648 – 1652.

[2] Yung-Kuei Lu; Ming-Der Shieh "Efficient Architecture for Reed-Solomon Decoder" VLSI Design, Automation, and Test (VLSI-DAT), 2012 International Symposium on Digital Object Identifier, Page(s): 1 – 4.

[3] Http://www.ece.nus.edu.sg/stfpage/eleak/pdf/soia04.pdf.

[4] Zhigang Ren; Dongping Yao "An Improved High-speed RS Encoding Algorithm" Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications, 2009 3rd IEEE International Symposium on Digital Object Identifier, Page(s): 521 – 523.

[5] Barbosa, T.C.; Moreno, R.L.; Pereira, T.C.; Ferreira, L.H.C. "FPGA Implementation of a Reed-Solomon CODEC for OTN G.709 Standard with Reduced Decoder Area" Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on Digital Object Identifier, Page(s): 1 – 4.

[6] Seungpum Kang, Sunwook Lee, Changgun Kim, and Yong Jee"ASIC Implementation of Reed-Solomon Error Correction Circuits for Low Area Overhead on Memory System" Proceedings of ICEIC2008, June 24-27, 2008.

[7] Http://mobiledevdesign.com/hardware_news/radio_error control_coding.Aqib.

[8] Al Azad, Minhazul. Huq, Iqbalur. Rahman Rokon" Efficient Hardware Implementation of Reed Solomon Encoder and Decoder in FPGA using Verilog" International Conference on Advancements in Electronics and Power Engineering (ICAEPE'2011) Bangkok Dec. 2011.