ISSN 2229-5518

Efficient Reed Solomon Decoder Based on pRiBM Algorithm for Effective Digital Signal Reception

R. Shanmuga Priya[1], A. Beno[2]
[1]PG Scholar, VLSI Design, [2]Assitant Professor, ECE Department,
[1&2]Dr. Sivanthi Aditanar College of Engineering, Tiruchendur.

Abstract— In this paper, Reed Solomon (RS) decoder adopting the pipelined reduced inversion free Berlekamp Massey (pRiBM) algorithm is proposed. This Reed Solomon decoder is designed by equipping the Syndrome Computation unit, Key Equation Solver (KES), Chien search and Forney algorithm unit. The RS decoder requires long latency for decoding large number of test vectors. Therefore, the Reed Solomon decoder using pipelined reduced inversion free Berlekamp-Massey algorithm is used to reduce latency. The Reed Solomon decoder is simulated with the Verilog HDL language using Xilinx 9.1 tool. The total delay obtained is 99.464 ns.

Index Terms— RS decoder, pRiBM, Syndrome Computation, KES, Chien search and Forney algorithm, Galois Field.

1 Introduction

In the present Digital Communication systems, it is highly possible that the data or message may get corrupted during transmission and reception through a noisy channel medium. The environmental interference and the physical defects in the medium are the main causes for the data or message corruption in the communication medium, which leads to the injection of random bits into the original message and corrupt the original message. To have a reliable communication through noisy medium error correcting codes (ECC) [1], [2], [3], [4] has to be used.

The error correction is based on mathematical formulas, which are used by error correcting codes (ECC). Error correction takes place by adding parity bits to the original message bits during transmission of the data. Because of the addition of parity bits to message bits, the size of the original message bits becomes longer. Now this longer message bits is called "Codeword". This codeword is received by the receiver at the destination, and could be decoded to retrieve the original message bits [5], [6].

Reed Solomon Codes are used for both encoding and decoding purposes. These codes are defined as RS (n, k) with m bit symbols. The RS codes detect and correct the errors in the symbols. Reed Solomon codes follow the Galois Field (GF) mathematic properties for encoding and decoding techniques [7], [8], [9].

In this paper, a RS decoder using pRiBM algorithm is proposed with the aim of reducing the delay and improving the speed for the RS (255,223) decoder [10], [11], [12]. The RS (255,223) code uses a primitive polynomial which is,

$$p(x) = 1+\alpha^2+ \alpha3+\alpha^4+ \alpha^8 \text{ over } GF(2^8) \quad (1)$$

The RS decoding is done in three levels. The very first step in the decoding process is to calculate the syndrome values which are used to check, whether the errors are occurred or not. The second step includes error location which tells us where the error is present. This is done by inversion-free Berlekamp Massey algorithm and the third one is the error evaluation which corrects the error. For error correction Chien search and Forney algorithm are used. The decoder has the capability of correcting up to t errors where n=k+2t [13], [14], [15].

2 Reed Solomon Decoder

The decoder has three main functions: 1. Syndrome computation 2. Key equation solver 3. Chien search and Forney algorithm.

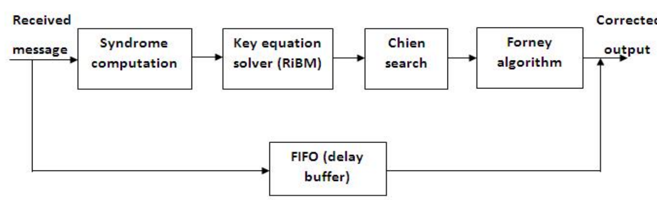

Fig .1 Reed Solomon Decoder

2.1 Syndrome Computation

The architecture for Syndrome Computation is shown in fig. 2. The input to this module is the corrupted codeword, r(x). The equations for the codeword, received bits and the error bits are given in the below equations.

Codeword equation

$$c(x) = c_0 + c_1x + c_2x^2 + ... + c_{n-1}x^{n-1} \quad (2)$$

Received bits equation

$$r(x) = r_0 + r_{1x} + r_{2x}{}^2 + ... + r_{n-1x}{}^{n-1} \quad (3)$$

Error bits equation

$$e(x) = e_0 + e_1x + e_2x^2 + ... + e_{n-1}x^{n-1} \quad (4)$$

Thus, the final transmitted data polynomial equation is given as

$$r(x) = c(x) + e(x) \quad (5)$$

Here the input received symbols are divided by the generator polynomial. The result should be zero. The parity is placed in the codeword to ensure that code is exactly divisible by the generator polynomial. If there is a remainder, then

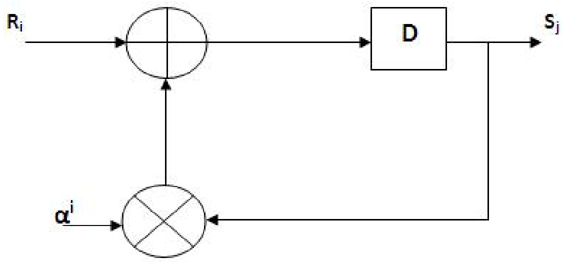

Fig .2 Syndrome Computation Architecture

there are errors. The remainder is called the syndrome.

The syndromes S_j can be defined as

$$S_j = \sum_{i=0}^{n-1} r_i \alpha^{i \cdot j} \quad \text{for } (1 \le j \le 2t) \quad (6)$$

2.2 Key Equation Solver

Once the syndromes are found, the next step is to determine the coefficients for the error location polynomial. For this, pipelined Reduced inversion-free Berlekamp Massey algorithm (pRiBM) is used. The architecture for this is shown in fig. 3. This stage involves the solving of the *2t* syndrome polynomials, formed in the previous stage. These polynomials have v unknowns, where v is the number of unknown errors prior to decoding. If the unknown locations are, i_1, i_2, ……, i_v the error polynomial can be expressed as

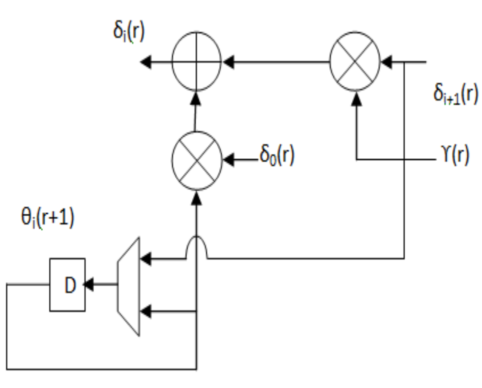

Fig .3 pRiBM Architecture

$$E(x) = Y_1 x^{i1} + Y_2 x^{i2} + \ldots\ldots + Y_v x^{iv} \quad (7)$$

where Y_1 is the magnitude of the 1st error at location i_1.

2.3 Chien Search and Forney algorithm

After the Key Equation Solver (KES) block, the error locator polynomial σ(x) and the error value polynomial ω(x) are fed into the Chien search block and Forney block, respectively. The architecture for Chien search is shown in figure 4 and the architecture for Forney algorithm is shown in fig. 5.

The job of the Chien Search block is to calculate the roots of the error-locator polynomial of degree t in GF (2^m), where t is the maximum number of errors that can be corrected in the RS code. When the result of an evaluation of error location polynomial equals 0, it indicates an error in the $(n-i)^{th}$ symbol in the codeword.

The Forney algorithm block works in parallel with the Chien search block to calculate the magnitude of the error symbol at each location. A polynomial is formed by combining the error locations (as powers of x) with the error magnitudes (as powers of α). The codeword is corrected by combining this

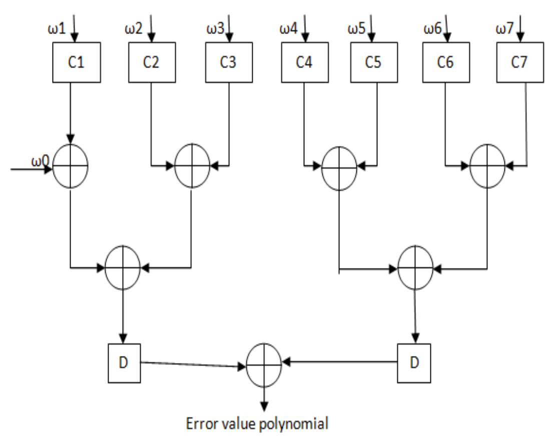

Fig .5 Forney Block

polynomial with the original codeword polynomial. The result is the corrected codeword.

3 Galois Field (GF) Arithmetic

Reed Solomon decoders are often based on Finite field also known as Galois field. An RS code with 8 bit symbols will use a GF (2^8) consisting of 256 symbols.

3.1 Galois Field Addition

Addition and subtraction of elements of GF (2^8) are simple XOR operations of the two operands. Each of the elements in the GF is first represented as a corresponding polynomial. The addition or subtraction operation is then represented by the XOR operation of the coefficient of corresponding polynomials.

3.2 Galois Field Multiplier

The critical arithmetic unit in the RiBM architecture is the *GF* multiplier. So to improve the clock frequency, efficient *GF* multiplier should be designed. Since the Chien search and Forney algorithm also uses the Galois field arithmetic, Galois field multipler is used. The basic difference between Galois field multiplication and general binary multiplication comes from the fact that the Galois field is finite. Hence the result of any operation should lie in the same field. In binary multiplication of two m bit data the result may go beyond m bits. But

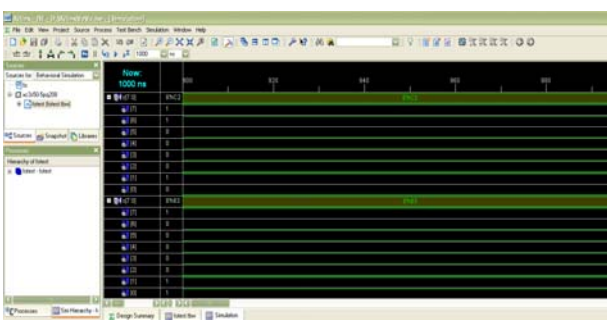

Fig .7 Simulation result for Forney algorithm

in case of Galois field as the field is defined as the set of m bit symbols, the result of any operation should be of m bit.

4 Results

The simulation result obtained for Forney algorithm is shown in fig. 7.

5 Conclsion

In this project, Berlekamp Massey algorithm is used as key equation solver in the Reed Solomon Decoder. The Reed Solomon Decoder used here is RS (255,223). The decoder is designed and simulated using Verilog code. The delay obtained for syndrome computation without pipelining is 18.536 ns and the delay is reduced to 15.73 ns through pipelining. The delay obtained for RiBM algorithm is 57.644 ns and the delay obtained for pRiBM algorithm is 49.705 ns. The delay obtained for Chien Search algorithm and Forney algorithm is 18.740 ns and 15.289 ns. Therefore the total delay obtained without pipelining is 110.209 ns and with pipelining is 99.464 ns. Thus the Reed Solomon Decoder using pRiBM algorithm is more efficient than Reed Solomon Decoder using RiBM algorithm.

References

[1] Arun Raghupathy, Member, IEEE, and K. J. R. Liu, Senior Member, "Algorithm-Based Low-Power/High-Speed Reed–Solomon Decoder Design", IEEE Transaction on circuits and systems - II: analog and digital signal processing, vol. 47, no. 11, November 2000.

[2] Cardarilli G C, Pontarelli S, et.al, "Concurrent Error Detection in Reed–Solomon Encoders and Decoders" IEEE transactions on very large scale integration (VLSI) systems vol. 15, no. 7, July 2007.

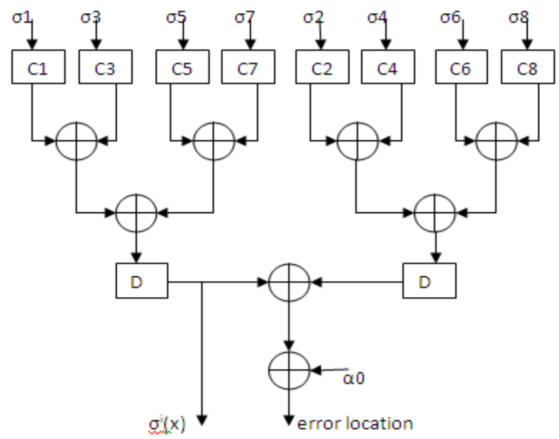

Fig .4 Chien Search Block

[3] Garcia-Herrero F, Valls J, et.al, "High speed RS(255, 239) decoder based on LCC decoding," Circuits System Signal Process., vol. 30, no. 6, pp. 1643–1669. June 2011.

[4] Jiangli Zhu, Student Member, IEEE, and Xinmiao Zhang, Member, IEEE, "Efficient VLSI Architecture for Soft-Decision Decoding of Reed–Solomon Codes", IEEE Transaction on circuits and systems - I: regular papers, vol. 55, no. 10, November 2008.

[5] Jih-Chiang Yeo, Huai-Yi Hsu, et.al, "A scalable Reed-Solomon decoding processor based on unified finite-field processing element design" Graduate Institute of Electronic Engineering and Department of Electrical Engineering, National Taiwan University, 2004.

[6] Jun Ma, Student Member, IEEE, Alexander Vardy, Fellow, IEEE, and Zhongfeng Wang, Senior Member, "Low-Latency Factorization Architecture for Algebraic Soft-Decision Decoding of Reed–Solomon Codes", IEEE transactions on very large scale integration (VLSI) systems, vol. 15, no. 11, November 2007.

[7] Kihoon Lee, Han-Gil Kang, et.al, "A High-Speed Low-Complexity Concatenated BCH Decoder Architecture for 100 Gb/s Optical Communications",. Springer Science + Business Media, August 2010.

[8] Koetter R and Vardy A, "Algebraic soft-decision decoding of Reed–Solomon codes," IEEE Transaction on Information. Theory, vol. 49, no. 11, pp. 2809–2825, November 2003.

[9] Lui W, Rho J, and Sung W, "Low-power high-throughput BCH error

correction VLSI design for multi-level cell NAND flash memories," in Proc. IEEE Workshop Signal Processing System, October 2006, pp. 303–308.

[10] Sarwate D V and Shanbhag N R, "High-speed architecture for Reed–Solomon decoders," IEEE Transaction on Very Large Scale Integration (VLSI) System, vol. 9, no. 5, pp. 641–655, Oct. 2001.

[11] Toshihiko Morita, Member, IEEE, Mitsuhiko Ohta, and Takao Sugawara, "Efficiency of Short LDPC Codes Combined with Long Reed Solomon Codes for Magnetic Recording Channels" IEEE transactions on magnetics, vol. 40, no. 4, July 2004.

[12] Wang Z and Ma J, "High-speed interpolation architecture for soft decision decoding of Reed–Solomon codes," IEEE Transaction Very Large Scale Integration (VLSI) System, vol. 14, no. 9, pp. 937–950, September 2006.

[13] Whitaker S, Canaris J, and Cameron K, "Reed Solomon VLSI codec for advanced television," IEEE Transaction on Circuits System Video Technology, vol. 1, pp. 230–236, June 1991.

[14] Wilhelm W, "A New Scalable VLSI Architecture for Reed-Solomon Decoders," IEEE Journal of Solid-State Circuits, vol. 34, no. 3, March 1999.

[15] Zhang R X and Zhu J, "Algebraic soft-decision decoder architectures for long Reed Solomon codes," IEEE Transaction on Circuits System II, vol. 57, no. 10, pp. 787–792, October 2010.