International Journal of Advanced Information Science and Technology (IJAIST) *ISSN: 2319:268*
Vol.2, No.4, April 2013 *DOI:10.15693/ijaist/2013.v2i4.177-181*

Reed-solomon decoder based on nielson's algorithm for Low-latency interpolation

R.Manojprabhakaran[1], M.Darani kumar[2]
[1]Department of ECE, Karpagam University, Coimbatore
[2]Asst. Professor/ ECE, Karpagam University, Coimbatore

***Abstract* - In wireless, satellite, and space communication systems, reducing error is critical. Algebraic soft decision decoding (ASD) of RS codes can obtain significant coding gain over the hard-decision decoding (HDD). Compared with other ASD algorithms, the low-complexity Chase (LCC) decoding algorithm needs less computation complexity with similar or higher coding gain. Besides employing complicated interpolation algorithm, the LCC decoding can also be implemented based on the Nielson algorithm. The Nielson algorithm works with a different scheduling, takes care of the limited growth of the polynomials, and shares the common interpolation points, .RS decoder can speed up by 57% and the area will be reduced to 62% compared with the original design for η = 3.**

Keywords* - *Algebraic soft decision decoding (ASD), hard-decision decoding (HDD), Low-complexity Chase (LCC).

I.INTRODUCTION

Reed–Solomon (RS) codes are widely employed as the error control code in numerous digital communication and storage systems. Berlekamp developed the first practical decoding procedure for RS codes in 1968 [2]. In recent years, the error control capability was improved by Koetter and Vardy [3], by incorporating the reliability information from the channel into the algebraic soft-decision (ASD) decoding process. Among all ASD algorithms, the low-complexity Chase (LCC) decoding needs to interpolate 2^η test vectors with maximum multiplicity one. Hence, the LCC decoding has less computation complexity and similar or higher coding gain compared with other ASD algorithms.

The Nielson's algorithm [5], [6] can be employed to implement the interpolation. Various techniques have been proposed to reduce the complexity of the interpolation based on this algorithm. The interpolation complexity can be further reduced by optimizing the discrepancy coefficient computation and candidate polynomial updating involved in each iteration of the Nielson's algorithm.

Usually, the interpolation is the common method in the LCC decoding algorithm to get the error locator and the evaluator polynomial. By applying the re-encoding and the coordinate transformation technique, the number of points needs to be interpolated can be reduced to n - k for an (n, k) RS code [4]–[6]. Much work has been done directly to simplify the complicated interpolation procedure [7], [8]. The decoding (HDD), the inversion-less Berlekamp–Massey (iBM)[11] algorithm. In this HDD-based LCC decoding algorithm, the test vectors are selected for correction during the decoding on occurrence of the HDD failure. Generally, this method leads to saved area and shorter latency for the LCC decoder. However, syndromes of all test vectors are needed in the LCC decoding and there are 2^η test vectors. Since η could be more than three to get higher coding gain, it requires a great number of hardware to compute and store these syndromes. Moreover, extra clock periods are necessary to finish the syndrome computation of each test vector before the key equation solver (KES) starts.

As the result, the decoding latency and the area are reduced to 64% and 62% for η = 3, respectively, which leads to 2.5 times speed over area ratio than the decoder in [9]. Further analysis shows the area of the proposed decoder is 69% of the decoder based on RCMI for η = 5, with 13% speed up.

II.RELATED WORK

Reed-Solomon codes are block based error correcting codes with a wide range of applications in digital communications and storage. It is vulnerable to the random errors but strong to burst errors. Hence, it has good performance in fading channel which have more burst errors. In coding theory Reed–Solomon (RS) codes are cyclic error correcting codes invented by Irving S.Reed and Gustave Solomon. They described a systematic way of building codes that could detect and correct multiple random symbol errors. By adding t check symbols to the data, an RS code can detect any combination of up to t erroneous symbols, and correct up to [t/2] symbols. As an erasure code, it can correct up to t known erasures, or it can detect and correct combinations of errors and erasures. Reed-Solomon codes are used to correct errors in many systems including:

- Storage devices (including tape, Compact Disk, DVD, barcodes, etc)
- Wireless or mobile communications (including cellular telephones, microwave links, etc)
- Satellite communications
- Digital television / DVB

The Reed -Solomon encoder takes a block of digital data and adds extra "redundant" bits. Errors occur during transmission or storage for a number of reasons (for example noise or interference, scratches on a CD, etc). The Reed -Solomon decoder processes each block and attempts to correct errors and recover the original data. The number and type of errors that can be corrected depends on the characteristics of the Reed -Solomon code.

A. Historical background

On January 2, 1959, Irving Reed and Gus Solomon submitted a paper to the Journal of the Society for Industrial and Applied Mathematics. In June of 1960 the paper was published: five pages under the rather unpretentious title "Polynomial Codes over Certain Finite Fields". This paper described a new class of error-correcting codes that are now called Reed-Solomon codes. In the decades since their discovery, Reed-Solomon codes have enjoyed countless applications, from living rooms all over the planet to spacecraft that are now well beyond the orbit of Pluto. Reed-Solomon codes have been an integral part of the telecommunications revolution in the last half of the twentieth century.

B. Encoding of RS codes

Reed Solomon codes are a subset of BCH codes and are linear block codes. A Reed -Solomon code is specified as RS (n, k) with s-bit symbols. This means that the encoder takes k data symbols of s bits each and adds parity symbols to make an n symbol codeword. There are n-k parity symbols of s bits each. A Reed Solomon decoder can correct up to t symbols that contain errors in a codeword, where 2t = n-k.

Given a symbol size s, the maximum codeword length (n) for a Reed-Solomon code is $n = 2s - 1$. For example, the maximum length of a code with 8 bit symbols (s=8) is 255 bytes. Reed Solomon codes may be shortened by (conceptually) making a number of data symbols zero at the encoder, not transmitting them, and then reinserting them at the decoder. The amount of processing "power" required to encode and decode Reed Solomon codes is related to the number of parity symbols per codeword. A large value of t means that a large number of errors can be corrected but requires more computational power than a small value of t [6].

C. Decoding of RS codes

Reed-Solomon algebraic decoding procedures can correct errors and erasures. An erasure occurs when the position of an erred symbol is known. A decoder can correct up to t errors or up to 2t erasures. Erasure information can often be supplied by the demodulator in a digital communication system, i.e. the demodulator "flags" received symbols that are likely to contain errors [7].

When a codeword is decoded, there are three possible outcomes:

1. If 2s + r < 2t (s errors, r erasures) then the original transmitted code word will always be recovered,

OTHERWISE

2. The decoder will detect that it cannot recover the original code word and indicate this fact.

OR

3. The decoder will mis-decode and recover an incorrect code word without any indication.

The probability of each of the three possibilities depends on the particular Reed -Solomon code and on the number and distribution of errors.

D. Coding gain

The advantage of using Reed Solomon codes is that the probability of an error remaining in the decoded data is (usually) much lower than the probability of an error if Reed Solomon is not used. This is often described as coding gain.

III.ASD ALGORITHM

A. Finite (Galois) field arithmetic

Reed-Solomon codes are based on a specialist area of mathematics known as Galois fields or finite fields. A finite field has the property that arithmetic operations (+, -, x, / etc.) on field elements always have a result in the field. A Reed - Solomon encoder or decoder needs to carry out these arithmetic operations. These operations require special hardware or software functions to implement [8,9].

B. Generator polynomial

A Reed-Solomon codeword is generated using a special polynomial. All valid code words are exactly divisible by the generator polynomial. The general form of the generator polynomial is:

$$g(x) = (x-ai)(x-ai1)\ldots.- i (a+2t) x \quad (3.1)$$

and the codeword is constructed using:

$$c(x) = g(x).i(x) \quad (3.2)$$

where g(x) is the generator polynomial, i(x) is the information block, c(x) is a valid codeword and a is referred to as a primitive element of the field. Example: Generator for RS(255,249)

$$g(x) = (x-a0)(x-a1)(x-a2)(x-a3)(x-a4)(x-a5) \quad (3.3)$$

$$g(x) = x6+g5x5+g4x4 + g3x3 + g2x2 + g1x1 + g0 \quad (3.4)$$

C. Encoder architecture

The 2t parity symbols in a systematic Reed -Solomon codeword are given by:

$$p(x) = i(x). xn-k \bmod g(x) \quad (3.5)$$

An architecture f or a systematic RS (255,249) encoder each of the 6 registers holds a symbol (8 bits).

The arithmetic operators carry out finite field addition or multiplication on a complete symbol.

D. Decoder architecture

The received codeword r(x) is the original (transmitted) codeword c(x) plus errors:

$$r(x) = c(x) + e(x) \quad (3.6)$$

A Reed-Solomon decoder attempts to identify the position and magnitude of up to t errors (or 2t erasures) and to correct the errors or erasures. Decoding is done by adopting the following steps:

(1) Multiplicity assignment:

First selects the η most unreliable code positions. Each of these code positions is assigned two interpolation points: (αj, βj) and (αj, βj).

(2) Interpolation:

Find a minimal bivariate polynomial P(X,Y) that passes through all the points with the prescribed multiplicities.

(3) Find an error locator polynomial

This can be done using the Berlekamp -Massey algorithm or Euclid's algorithm. Euclid's al used in practice because it is easier to implement however, the Berlekamp-Massey algorithm tends to lead to more efficient hardware and software implementations.

(4) Find the roots of this polynomial(Factorization):

This is done using the Chien search algorithm. Find all the factors of P(X,Y) of the form g-f(x) with deg f(x)<k.

(5) Finding the Symbol Error Values

Again, this involves solving simultaneous equations with t unknowns. A widely used fast algorithm is the Forney algorithm.

IV.LCC decoding algorithm based on Nielson's algorithm

An (n, k) RS code constructed over the finite field GF (2q), n = 2q−1, having the maximum separable distance d = 2t +1 = n−k. For the ASD decoding, there are generally three steps, multiplicity assignment, interpolation, and factorization. The hard-decision codeword should be,

$$rHD(x) = r0\_HD + r1\_HD \cdot x \cdot \cdot \cdot + r(n-1)\_HD \cdot xn-1 \quad (4.1)$$

and the second-best decision codeword should be ,

$$r2HD(x) = r0\_2HD + r1\_2HD \cdot x \cdot \cdot \cdot + r(n-1)\_2HD \cdot xn-1 \quad (4.2)$$

The multiplicity assignment first compares the reliability of each symbol in one codeword. The reliability can be expressed as the ratio of probability between the hard-decision symbol ri_HD and the second-best decision symbol ri_2HD for position i, $0 \le i \le n-1$, Pro(ri |ri_HD)/Pro(ri |ri_2HD).

Unreliable set $\overline{R}$ is formed in multiplicity assignment by selection of the n − k unreliable symbol positions according to the ratio of probability, while the set of rest k reliable symbol positions is denoted as R. Multiplicity given as,

$$m(x) = m0+m1 \cdot x \cdot \cdot \cdot +mn-1 \cdot xn-1 \quad (4.3)$$

Points with position i , $i \in \overline{R}$ are given multiplicity of one, mi = 1, while points with position i , $i \in R$ are assigned mi = 0. The set of the most unreliable η symbol positions in $\overline{R}$ is denoted as Z, and the symbols in these positions are assigned both ri_HD and ri_2HD, $i \in Z$, while in other n − η positions, only one hard-decision symbol ri_HD are given, $i \in R + \overline{R} - Z$. In this way, multiplicity assignment can generate 2η test vectors by selecting between ri_HD and ri_2HD, $i \in Z$. In this paper, we focus on the interpolation step. Popular interpolation algorithms include the Nielson's algorithm and the Lee-O'Sullivan (LO) algorithm [13]. However, the LO algorithm cannot take in more points once the interpolation started. In this paper, we use the Nielson's algorithm in our design. The pseudo codes of this algorithm are listed in Algorithm 1.

Algorithm 1. Nielson's interpolation for m=1,L=1
Input: Interpolation points (α.β)
Initialization: $g_0(x, y)=1$, $g_1(x, y)=y$, $deg_0=1$,$deg_1= -1$
Interpolation:
For i=0 to length (α) – 1
A1: $\delta_0= g_0(\alpha i, \beta i)$,$\delta_1= g_1(\alpha i,\beta i)$
if($min(deg_0,deg_1)=deg_0$) and $\delta_0 \neq 0$)
A2:$g_1(x, y)= g_1(x, y).\delta_0+ g_0(x, y).\delta_1$
A3: $g_0(x, y)= g_0(x, y)(x+ \alpha i)$
$deg_0=deg_0+1$
elsif($min(deg_0 .deg_1)=deg_1$)and $\delta_1 \neq 0$)
A2: $g_0(x, y)= g_1(x, y).\delta_0+ g_0(x, y).\delta_1$
A3: $g_1(x, y)=g_1(x, y)(x+ \alpha i)$
$deg_1=deg_1+1$
Output: $g_0(x, y)$, $g_1(x, y)$

The re-encoding and coordinate transformation [7], [8] can convert the *Y* coordinate of the *k* most reliable interpolation points to zero. As a result, the interpolation over these points can be presolved by simple univariate interpolation. In addition, the point-serial scheme [9], [10] and a hybrid finite field element representation [11] can be employed to increase the speed of the discrepancy coefficient computation. The decoder based on Nielsons algorithm is given in fig.1.

By employing the re-encoding and the coordinate transformation, the number of points need to be interpolated is reduced to n−k and the values of the

interpolation points are transformed to βi_HD and βi_2HD from ri_HD and ri_2HD, i ∈ Z and to βi_HD from ri_HD, i ∈ $\overline{R}$ − Z. After get the interpolation polynomials of 2^η test vectors, the polynomial selection algorithm is required to select the correct interpolation polynomial for the successful decoding.

By comparing the degree of the error location polynomial and its roots number, the correct interpolation result can be selected. If the degree equals the root number, this polynomial will be regarded as the right interpolation result. After the polynomial selection, the Chien search and Forney algorithm (CSFA) could be applied to correct the error symbols in the codeword. Finally, an erasure decoding is used to recover the whole codeword.

Each position in Z will be also assigned two symbols, ri_HD and ri_2HD, i ∈ Z. The positions in R, which are reliable will only have the hard decision symbol, ri_HD, i ∈ R. Thus, 2^η test vectors can also be generated. Instead of interpolating the 2^η test vectors and choosing interpolation results, the syndromes of all test vectors are calculated in the Nielson decoding algorithm based on the same decoding performance.

V.HARDWARE AND LATENCY ANALYSES

A. Hardware Analyses

The interpolation module, the Factorization, and the BMA (IFB) are in one stage, so the proposed decoder consists of three pipelining stages, Interpolation, Factorization and CSFA,The advantage of adopting Chien search again in CSFA is to eliminate roots storage in the polynomial selection module. To estimate the hardware requirement of the whole decoder, a serial approximation is assumed. In GF(2^8), by employing the composite field arithmetic, a regular multiplier needs 64 XOR gates and 48 AND gates while an inverter consists of 121 XOR gates and 36 AND gates.

The hardware requirement of the decoder is reduced to 19426/31414 = 62%. The reduction in area comes from the three parts, Interpolation, the Factorization, and the BMA. The saving in polynomial is 8638 − 2928 − 844 = 4866 XOR gates and the KES saves 11696 − 8640 = 3056 XOR gates. Since only one polynomial selection module is required in the proposed decoder.

However, the contribution of the Nielson algorithm is more than 4866 XOR gates, because without this algorithm, the RiBM would require additional $2t(2^{\eta}-1)$ bytes registers to store the syndrome results of $2^{\eta}-1$ test vectors for pipelining, which equals 2688 XOR gates for $\eta = 3$ and 5760 XOR gates for $\eta = 4$. For $\eta = 4$, the proposed decoder needs 19594 XOR gates. The hardware requirement of the decoder is reduced to 19594/54278 = 36%.

Table 1 : Throughput analysis

	Registers	LUTs	Slice	Max clock Frequency	Throughput
Proposed LCC decoder	2247	5621	1534	149.5 MHz	1109.0 Mb/s
LCC decoder in [12]	5399	5114	2527	150.5 MHz	710.7 Mb/s

B. Latency Analysis

The decoding latency is decided by the longest stage latency. The interpolation module takes 255 clock periods, then the codeword and syndromes are passed to the IFB module. The output of decoder is given in Fig. 2.

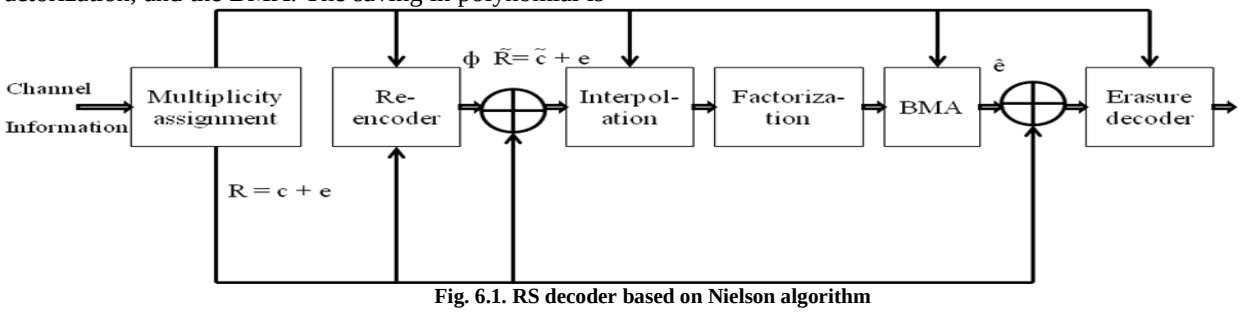

Fig. 6.1. RS decoder based on Nielson algorithm

For the IFB, the Interpolation 2*t* clock periods to update the next test vector. What should be noted is that there are two additional clock periods at the beginning of update to select the point to be updated and get the value in S_{diff} , only for the first test vector. It also takes the 2*t* clock periods to carry out the error locator polynomial and the error evaluator polynomial of one test vector. The throughput and the area will be given in table.1.

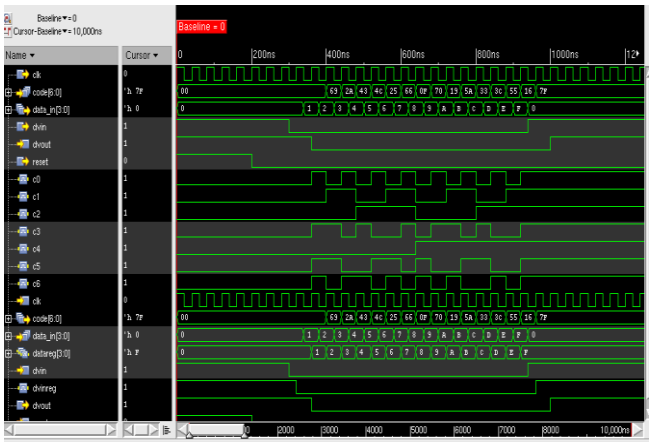

Fig.2 Output waveform of Nielson algorithm based RS decoder

The modified polynomial selection needs *n/p* **+** 1 clock periods to finish the root search after the KES finishes on one test vector. It will add *n/p* **+** 1 clock periods to the stage latency after the KES and the update all finish in the IFB, if the polynomial selection is put into the second pipelining stage. However, it saves the memory of one codeword as the stage number is reduced from 4 to 3.

VI. CONCLUSION

Nielson algorithm was proposed in this brief and the improved RS decoder, gives better performance on throughput and area. The proposed decoder is significantly more efficient than prior designs. Next, we will focus on further improving the throughput of the RS decoder by reducing test vectors. Less RAM is required for pipelining.

VII. REFERENCES

[1] Sanjeev Kumar., Ragini Gupta .,Bit Error Rate Analysis of Reed-Solomon Code for Efficient Communication System, IJCA (0975 – 8887), Vol: 30– No.12, September 2011

[2] Wei Zhang, Hao Wang, and Boyang Pan., Reduced-Complexity LCC Reed–Solomon Decoder Based on Unified Syndrome Computation,Jan 2012

[3] Daniel J., Costello, JR., Error Control Coding, Fundamentals and Applications, Prentice Hall, Inc. Englewood Cliffs, New Jersey, 1983

[4] S. Lin and D.J. Costello, Jr. Error Control Coding: Fundamentals and Applications, Englewood Cliffs, NJ: Prentice Hall, 1983

[5] J.Proakis, Digital communications , NY: McGraw Hill, 2001

[6] I.S. Reed and G. Solomon, "Po Certain Finite Fields," SIAM Mathematics, vol. 8, 1960, pp. 300304.

[7] J.McEliece, L.Swanson, probability for Reed Solomon codes" Inf .Theory, Vol.IT-32, pp.701-703

[8] R. E. Blahut, "Transform Techniques for Error Control Codes," IBM Journal of Research and Development, Volume 23, pp. 299-315, 1979.

[9] S. B. Wicker, Error Control Systems for Digital Communication and Storage, Englewood Cliffs, N.J.: Prentice-Hall, 1994.

[10] J. Zhu, X. Zhang, and Z. Wang, "Combined interpolation architecture for soft-decision decoding of Reed–Solomon codes," in Proc. IEEE Conf. Comput. Design, Lake Tahoe, CA, Oct. 2008, pp. 526–531.

[11] X. Zhang and J. Zhu, "Algebraic soft-decision decoder architectures for long Reed–Solomon codes," IEEE Trans. Circuits Syst. II, vol. 57, no. 10, pp. 787–792, Oct. 2010.

[12] F. Garcia-Herrero, J. Valls, and P. K. Meher, "High speed RS(255, 239) decoder based on LCC decoding," Circuits Syst. Signal Process., vol. 30, no. 6, pp. 1643–1669. Jun. 2011.

Authors

Author's Name: R.Manojprabhakaran
Author's profile: Doing M.E(VLSI Design) in Karpagam University. Completed B.E(ECE) at Kongu Engg College, Erode.

Author name: M.Darani kumar M.E.,
Author's profile: Working as asst professor in Karpagam University, Coimbatore