International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE) Vol 2, Issue 3, Mar 2015

High Efficiency Low Complexity Chase Architecture for Reed-Solomon Decoder of RS(255,K)

[1]Remalli Dinesh, [2]Sandeep Bansal,[3]Damandeep Singh, [4]Shaik Shabeena
[1] [2] [3] [4] Department of Electronics and Communication Engineering
Lovely Professional University, Phagwara, Punjab
[1]r.dinesh.in@ieee.org, [2]sandeep.15732@lpu.co.in, [3]Damandeep.11005738@lpu.in, [4]shabeena19912@gmail.com

Abstract: **In contrast to all existing ASD soft-decision decoding concepts for RS codes, only low-complexity chase algorithm can attain improved tradeoff in performance-complexity with significant coding gain on HDD with polynomial complexity. LCC decoding scheme utilized 2^{η} test vectors having less computational complexity in addition to enhanced coding gain. Instead of short RS codes, LCC decoding is required to interpolate bulk of test vectors which results in long latency. Therefore, interpolations as well as polynomial selection are hefty part of LCC decoder with long RS code and significant value of η. Besides, innovative designs being developed to alter the interpolation and polynomial complexity for the efficient recovery of the codeword for the given test vectors in LCC decoding. So, toward the codeword recovery for an RS (n, k) code erasure decoding, Neilson algorithm, coordinate transformation etc. being applied on the RS code segment. Also Chine search method can be applied over the interpolation output and it can be realized through constant multipliers which save the cost factor. For the selected 8 test vectors applied on RS (255, k) efficiency can be lead up to very high with power reduction for GF (2^{10}).**

***Index Terms*— CMOS, LCC, Low power, Interpolation, Polynomial evaluation, Reed-Solomon, Soft-decision.**

INTRODUCTION

The Reed-Solomon codes are extensively used in digital communication and storage media devices for the purpose of error corrections. It also provides a proficient capability for correction of the burst errors [4]. Algebraic Soft-decision decoding algorithms (ASD) was anticipated to absorb the channel information to a decoding process which was based on interpolation while achieving considerable coding gain by means of polynomial complexity [1]. The decoding process which was practically applied for RS codes was first coined by Berlekamp in 1968. In past years, Koetter and Vardy enhanced the error control capability by assimilate the reliability information from the channel to the algebraic soft decision decoding (ASD) process [2]. Later Guruswami and Sudan also incorporate reliability information to ASD process resulted in generous coding gain more than hard-decision decoding (HDD) can be accomplished with a polynomial complexity compared to the codeword length [3].

This low-complexity chase (LCC) algorithm decoding process conduct testing on 2^{η} test vectors having interpolation points with the maximum multiplicity is one while in the η least consistent code positions the vectors are dissimilar [5]. The key steps in the ASD algorithms are considered to be the interpolation and factorization whereas there are ways to reduce complexities in these steps through re-encoding and coordinate transformation that radically reduce the area, timing and power complexity of the interpolation process [3] in addition the number of points in each test vector to be interpolated also reduce from RS (n, k) to RS (n-k, k) in each RS code [1].

A low-complexity scheme for polynomial selection was designed for re-encoded low-complexity chase (LCC) decoder. By transferring a single message symbol to the decoding process, it tests for the interpolation output whether it is zero or not through polynomial selection. Even though encoder should be improved and in this process single message symbol is lost and this polynomial selection scheme results in huge complexity reduction [6]. However, most of the previously existing work on Algebraic soft decision (ASD) decoder architecture design was for RS codes whose length is 255 bytes or shorter and when the code becomes longer, the error-correction and hardware complexity of decoder needs to be re-examine [7].

In this paper section II direct towards a variety of

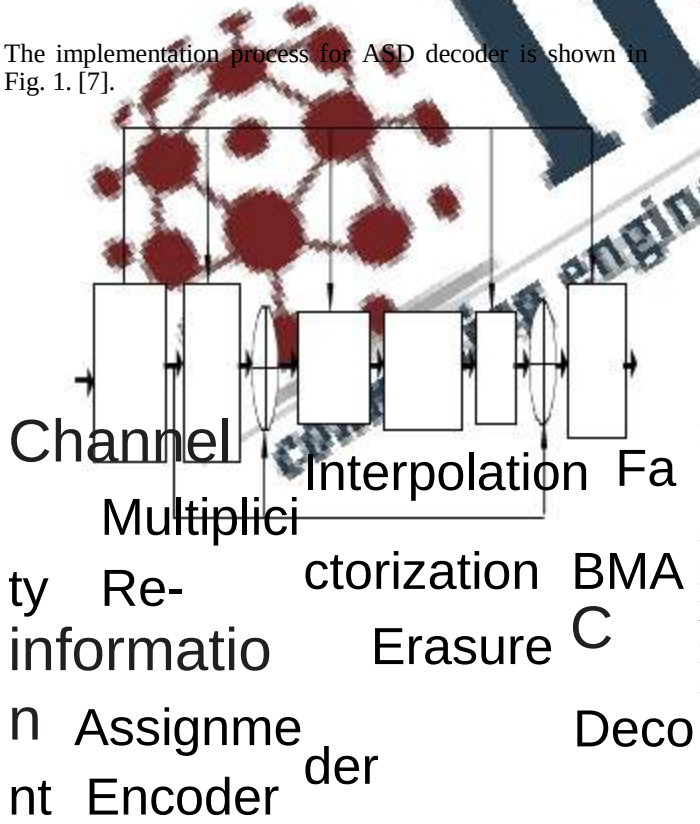
procedures for LCC decoding of RS codes. Section III explains about interpolation and polynomial complexity reducing algorithms. Section IV review in LCC decoding architecture. Section V encapsulates over hardware complexity issues and analyses them briefly. Finally we conclude our review on LCC decoding procedure in section VI.

PRACTICES IN LCC DECODING

ASD Decoding Algorithm

It takes into account an RS (n, k) code which would create over GF (2^q). It is executed in three steps namely multiplicity assignment, interpolation and factorization. The function of multiplicity assignment was to decide the interpolation points and interpolation locate a bivariate polynomial $Q(x, y)$ having minimum (1, k -1) weighted degree crossing each interpolation point. After that factorization calculates all factors for $Q(x, y)$ which is represented in the form of $y - f(x)$.

The implementation process for ASD decoder is shown in Fig. 1. [7].

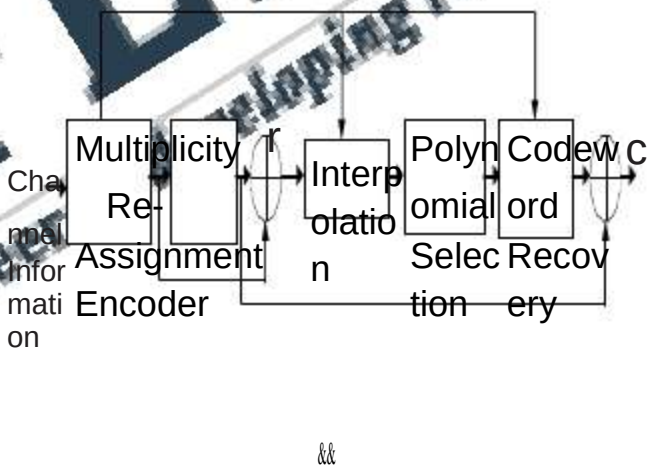


ASD Decoder Block

In Fig.1, the re-encoder is designed to locate the codeword ϕ which has the similar symbols as in received word r in most consistent k code position, which constituent a set R. Coordinate transformation also applies to the interpolation points. Accordingly, the decoding can be applied on $\phi + r$ and those points which have code locations in R was consider for interpolation process [7]. The outcome of factorization process used as syndromes in Berlekamp – Massey algorithm (BMA) for retrieving of errors in code position in R. Then, a new erasure decoding is

Re-Encoded LCC Decoding Technique

The LCC algorithm of ASD has multiplicity assignment one of its feature where (α_j, β_j) and (α_j, β_j') are the points which are allocate to each of η least consistent code position. In this field element is represented as α_j which was encoded in evaluation map encoding and the hard-decision symbol is shown by β_j whereas β_j' was second jth most expected symbol for code position. The polynomial $Q(x, y)$ which was found through interpolation can be solved through *kötter's* algorithm[5],which initiate with $Q^{(0)}(x, y) = 1$ and $Q^{(j)}(x, y) = y$ for LCC decoding having high rate codes. The Block diagram of LCC decoder is shown in Fig. 2 [5].



&&

Fig 3. Block Diagram of Re-Encoded LCC Decoder

Now, developing a *Grobner* basis by allowing iteratively updating of two polynomials so that it can pass an additional point at a time. Then the polynomial found to be lowest weighted degree was considered to be the least weighted degree among all the present polynomial. Hence, lowest weighted degree polynomial from the most recent iteration was the required interpolation output. Then making functional for recovering of complete codeword [7].

–

factorization determines all the factors related to $Q(x, y)$ in the form such as $y - f(x)$ with constituent degree of $(f(x)) < k$ and each $f(x)$ in the list are equivalent to a message polynomial [5].

Transformed LCC Decoding with Re-Encoding

In order to make simpler the interpolation in algebraic soft decision (ASD), the re-encoding and coordinate transformation algorithm can be applied on it. Let indicate the arrangement of majority of k reliable code position by R in r. The erasure decoding applied in re-encoding to obtain the codewordϕ . Now, the decoding is carried on [8]:

$$r = r + \phi \quad (1)$$

Let us assume the error vector e is being adjoin with the

codeword and it can be represented as [8]:

$$r = c + e \quad (2)$$

Also, we can obtain another codeword using the similar error vector which can be represented as [8]:

$$\overline{r} = c + \phi + e = \overline{c} + e \quad (3)$$

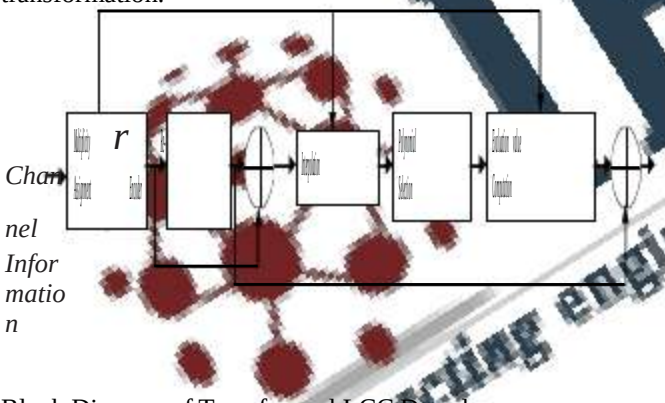
In addition for $i \in R$,

$$\overline{r_i} = r_i + \phi_i = 0 \quad (4)$$

So the interpolation process can be applied on various code positions which were available in R and it can be applied on rest of the code position which was accessible only in $n-k$ code position. Therefore, the polynomial which was pre-computed and factor of these polynomial can be computed as [8]:

$$v(x) = \Pi_{i \in R}(x + \alpha_i) \quad (5)$$

The length of polynomial can be further decrease up to k by taken out the factor using the process of coordinate transformation.



Block Diagram of Transformed LCC Decoder

One approach can be applied using $q_1(x)$ and $q_0(x)$ such that errors can be locate in code position of R while using Chien search and Forney's algorithm [9]. After the errors in R are rectified so to avoid the complexity originate from factorization, the erasure decoding procedure can be applied to retrieve the $n-k$ symbols in R. Instead to gain the message polynomial $f(x)$ of c proceed with the multiplication of $v(x)$ back to the interpolation output.

Complexity Reduction Using Diverse Algorithms

Polynomial Selection and Interpolation

Code $Q(0, f_o)$ tested for polynomial selection which was developed from [1]:

$$Q(0, f_0) / \Pi_{j \in R} \alpha_j$$

$$\hat{Q}(0, f_0) = q_1(0)\, \hat{f}_0 + q_0(0) / \Pi_{j \in R} \alpha_j$$

(6) $\hat{f}_0 = \overline{f_0} \Pi_{j \in R} \alpha_j$

Where,

(7)

Although $\hat{Q}(0, \hat{f}_0)$ can be obtain by follow the polynomial updating while interpolation without having any knowledge of values of $q_1(0)$ and $q_0(0)$ as required in the computation. This important concept helps greatly in the new interpolations scheme as described below in the following steps:

Step1: Assign the values

$$Q^t_{(0)}(m, n) = y^t \quad (8)$$

$$Q^t_{(0)}(0, \hat{f}_0) = (0)\hat{f} \quad \text{For } l = 0,1$$

$$Q^t_{(0)}(\alpha_j, \beta_j) = (\beta_j)^t \quad (10)$$

$$Q^t_{(0)}(\alpha_j, \beta'_j) = (\beta'_j)^t \text{ For } l = 0,1,\ j \in I_\eta \quad (11)$$

Step2: Interpolation of points in addition with the code position in I_η and updating of previous values to derive new values for:

$$Q^t_{n-k-\eta}(0, \hat{f}_0) \quad (12)$$

$$Q^t_{n-k-\eta}(\alpha_j, \beta_j) \quad (13)$$

$$Q^{t'}_{n-k-\eta}(\alpha_j, \beta\ ) \quad (14)$$

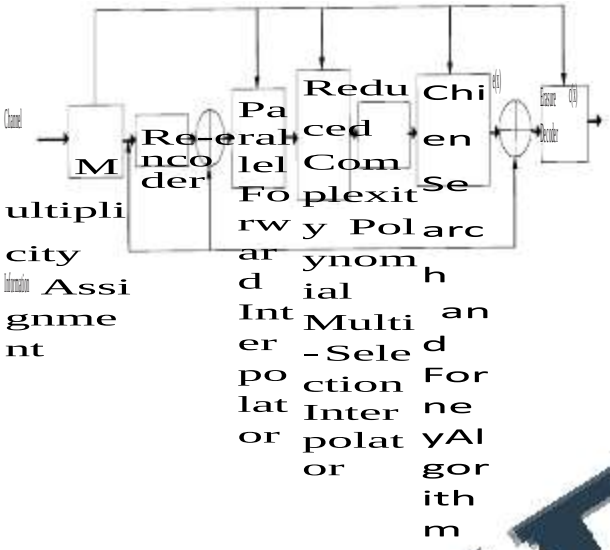
Step3: Now the objective is to update the evaluation values including $Q_{n-k}(0, \hat{f}_o) = 0$ and obtain $Q_{n-k}(0, \hat{f}_o)$ for every single vector. Also, choose the starting two test vector

Step4: At last for every single remaining η test vector perform interpolation over it.

The above proposed interpolation technique need two interpolators and a single Evaluation value updating (EVU) unit. In step 2 it needs a single interpolator and in step 3 it require both interpolator 1 and 2 in parallel. Besides the EVU is utilize to renew the evaluation values present at step 2 and step3.

Interpolator Architecture Based on Pipelining

As in LCC decoding the two vectors from start among 2_η test vectors are interpolate. The pipelining architecture of interpolator can sort out this problem by help in separate the interpolation in two different stages. The architecture for this pipelining is shown below in Fig. 4



Block Diagram for LCC through Pipelining

In this forward interpolator is attached with parallel forward interpolator (PFI) and in addition to it one Reduced complexity multi interpolator (RCMI) is utilize to complete the remaining unified forward-backward algorithm for interpolation. The PFI is designed to work by using $n-k-\eta-\sigma$ interpolation vectors of the forward interpolator where σ is defined as the total interpolation points which need to be interpolated in the RCMI. To minimize the critical latency in the forward interpolator, the PFI renew two coefficients and polynomial evaluation block code as in [10], intended for the τ^{th} test vector and the also determine the double coefficients in a single clock. After that they obtained parameters as in result will be transfer to further stage which is an RCMI and it makes to complete the interpolation process [2].

Cohesive Syndrome Calculation

LCC decoding depend on the CSC, for this all the 2η test vector in the polynomial need to be work out on their syndromes. There are various ways to find out the syndrome for diverse values of „n" and „k" in the RS(n, k)

$$S_{k-1} = \sum_{i\_HD} r_{i\_HD} \cdot \alpha^{ik}$$

Where α is the primitive element of $GF(2^{th})$ [10]. symbol which was in position i where $i \in Z$ set of integers, considered to be $r_{\_HD}$ or $r_{\_2HD}$ and this preference is denoted as $\xi_{i\_\tau} \cdot \xi_{i\_\tau}$ is HD or 2HD refer to distinctive point selection as in the test vector.

For the initial test vector which choose all $r_{\_HD}$, $i \in z$.

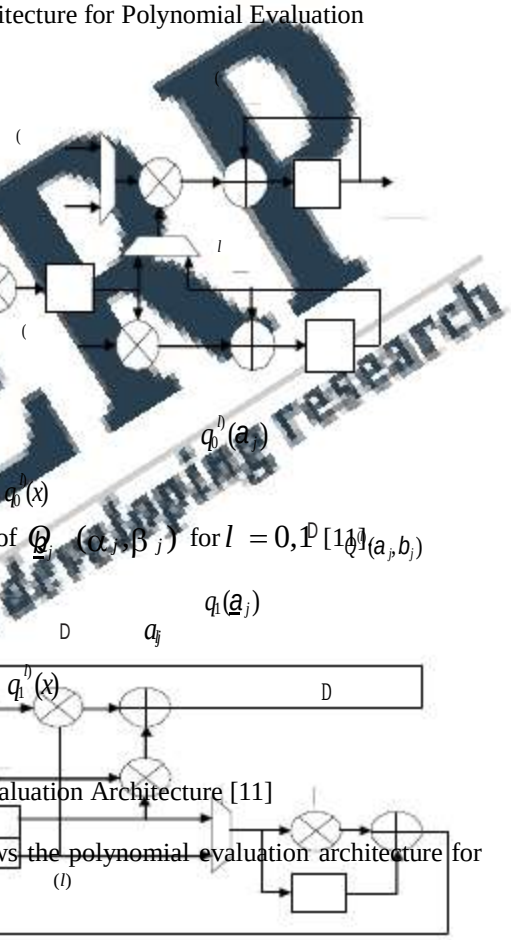
$$\sum_{i=0}^{n-1} \qquad \text{For k=1 to 2t} \qquad (15)$$

Since one different symbol occur in between $(\tau-1)th$ and τth while the location of this symbol is represented as ε and for the left over test vectors:

the primary test vector with the addition of syndrome difference.

LCC DECODING ARCHITECTURE FOR RS(255,K)

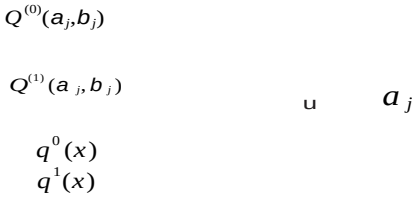
Modified Architecture for Polynomial Evaluation



Polynomial Evaluation Architecture [11]

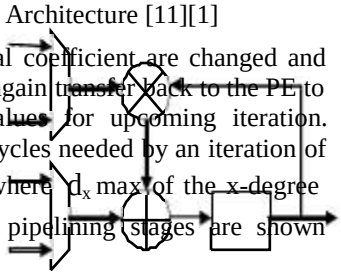
Fig. 5 shows the polynomial evaluation architecture for the evaluation of $Q_{a_j}(\alpha_j, \beta_j)$ for $l = 0,1$ [11]

Architecture for Polynomial Update



Polynomial Update Architecture [11][1]

As soon as the polynomial coefficient are changed and renewed by the PU, they are again transfer back to the PE to determine the evaluation values for upcoming iteration. Hence, the amount of clock cycles needed by an iteration of interpolation is $d_x + 1 + \xi$ where d_x max of the x-degree polynomial [11] is and the pipelining stages are shown by ξ.



Enhanced Architecture for Erasure Computation

Also,

$$S_{k\_\tau} = S_k$$

$$1/(a_i^{n-k}(a_i \prod_{i \neq j, l \in R}(a_i - a_l))$$

$$_{\_(\tau-1)} + (r$$

$$_{\_\xi\varepsilon\_\tau} - r$$

$$_{\_\xi\varepsilon\_(\tau-1)}) \cdot \alpha$$

$$\varepsilon^{k} \quad (16)$$

$$\lambda(\varepsilon,\xi) = (r_{\_\xi\varepsilon\_} \quad _{\_(\tau-1)}) \cdot \alpha \varepsilon^{k} \qquad (17)$$

So, the term $\lambda(\varepsilon,\xi)$ denotes the syndrome difference and it can be utilized in update depend on the outcome of

Erasure Magnitude Computation [3]

Fig. 7 depicts that the end result of erasure decoder ϕ_i can be gained with this architecture and it is apply to recover the complete codeword [3].

HARDWARE COMPLEXITY TO EXISTING TECHNOLOGY

Since various architectures for the LCC have been depicted, therefore there is not an exact solution to a question that which LCC decoding technique was more proficient with the alteration of various decoding parameters. The diverse η and code rates simultaneously resolve the error correction capability. In the LCC decoder combination of higher rate RS code with smaller η may have the related performance as in larger η and smaller rate code. Also, the lesser t will minimize the hardware complexity whereas the greater η raise the amount of test vectors in a vividly manner.

LCC DECODER FOR VARIOUS RATE AND η [2].

As it is seen from the TABLE I that the lower rate in LCC decoder are found to be more hardwa

RS Decoder (255,k)	Latency	Area	Efficiency	SNR
RS(255,239)	275	19426	1	9.80
RS(255,243)	256	25306	0.82	9.96
RS(255,239)	275	19594	1	9.71
RS(255,243)	401	25450		

if we increase η by 1, it will cause the test vectors to be doubled and in return which may cause to double the hardware to maintain them and trigger more latency. Conversely, the gain in t will costs restricted hardware.

CONCLUSION

A detailed and reviewed work on LCC implementation of RS decoder has been presented. Different forms of decoder designs; selection of test vectors and after the interpolation is applied only on the specific selected test vector. In addition proficient architecture of interpolation, polynomial selection and evaluation, erasure computation was developed. Although the reduction in various technology parameters was observed with an increase in η. Compared to past approaches, considerable area reduction and efficiency enhancement has been achieved without any loss of throughput. The latency of the RS decoder can be reduced with the help of pipelining and syndrome computation. Future work will concentrate on further advancement in technology and improvement in code recovery through various technology based parameters.

ACKNOWLEDGMENT

The authors are genuinely acknowledging the research scholars and associated members of Lovely Professional University for assistance in VLSI Design.

REFERENCES

[1] Xinmiao Zhang, Yingquan Wu, Jiangli Zhu, and Yu Zheng. "Novel Interpolation and Polynomial Selection for Low-Complexity Chase Soft-Decision Reed-Solomon Decoding" IEEE TRANSACTIONS on Very Large Scale Integration (VLSI) Systems; Vol. 20, No.7, pp:1318-1322, July 2012.

[2] Hao Wang, Wei Zhang, Jing Wang, and Zhe Jiang. "Hardware Complexities of Low-Complexity Chase Reed Solomon Decoders and Comparisons" IEEE; pp:216-219, 2012.

[3] W. Zhang, J. Wang, H. Wang, Y.Y.Liu, Z. Jiang, and S.Q. Wu. "Low-power high-efficiency architecture for low-complexity chase soft-decision Reed-solomon decoding" IET Communication; Vol. 6, No. 17, pp:3046-3052, August 2012.

[4] Ching-Lung Chi, and Tsung-Hsiu Chi. "A Fast Reed-Solomon Decoder Using Step-by-Step Algorithm". IEEE; pp:3326-3329, 2011.

[5] Xinmiao Zhang, and Yu Zheng. "Systematically Re-encoded Algebraic Soft-Decision Reed-Solomon Decoder" IEEE TRANSACTIONS on Circuits and Systems; Vol. 59, No.6, pp:376-380,June 2012.

[6] X. Zhang, F. Wu, and J. Zhu, "A novel polynomial for low-complexity Chase algebraic d-Solomon decoding" ISCAS; Rio de Janeiro, 2011.

[7] Xinmiao Zhang, and Jiangli Zhu. "Algebraic Soft-Decision Decoder Architectures for Long Reed-Solomon Codes". IEEE TRANSACTIONS on Circuits and Systems; Vol. 57, No.10,pp:787-792, Oct 2010.

[8] Xinmiao Zhang, and Yu Zheng. "Efficient Codeword Recovery Architecture for Low-Complexity Chase Reed-Solomon Decoding". National Science Foundation, IEEE.

[9] Dong-Sun Kim, and Jong-Chan Choi, and Duck-Jin Chung. "Implementation of High-Speed Reed-Solomon Decoder". IEEE; pp:808-812, 1999.

[10] Remalli Dinesh, and Sandeep Bansal. " Design and Formulative Analysis of VLSI Syndrome Generator for RS(128,Kx) and RS(64,Ky) ". International Journal of Computer Applications, IJCA; Vol. 92, No.8, pp:17-21, April 2014.

[11] Xinmiao Zhang, and Jiangli Zhu. "Interpolation based Hard-Decision Reed-Solomon Decoders". ISIC; pp: 175-178, 2009.