Improved Spectral Efficiency through Iterative Concatenated Convolutional Reed-Solomon Software Decoding

Daniel Iancu[*], Hua Ye[*], John Glossner[†,*] Michael J. Schulte[††], Suman Mamidi[††], Jarmo Takala[**]

[*]Sandbridge Technologies, Inc.
1 North Lexington Ave.
White Plains, NY 10601 USA
{diancu, huaye, jglossner}@SandbridgeTech.com
[**]Tampere University of Technology, FI
jarmo.takala@tut.fi

[†]Delft University of Technology
EEMCS
Delft, The Netherlands
[††] University of Wisconsin-Madison
{mamidi, schulte}@cae.wisc.edu

Abstract

IIn this paper we describe an iterative algorithm for Concatenated Convolutional Reed-Solomon decoders that improve the spectral efficiency of the communication system by increasing the error correction capabilities and as a consequence lowering the retransmission rate. In our method, the decoding process starts assuming the received code word has at most t *errors, where* 2t+1 *is the Reed-Solomon code's minimum distance. If, after the decoding process, all the syndromes are zero the decoding is successful; otherwise there were more than* t *errors encountered. At this point, the decoder assumes* s *erasure positions based on the erasure information coming from the convolutional decoder. If the error locator polynomial has degree equal to* r = (2t - s)/2, *then most likely the error positions are in the current Galois-Field and a second decoding algorithm is performed. Otherwise,* s=s+2 *erasures are assumed and again the degree of the error locator polynomial is checked. This continues until the maximum number of erasures,* 2t, *is reached. The Reed-Solomon decoder is executed entirely in software on the Sandbridge Processor, which features special instructions for Single Instruction Multiple Data (SIMD) Galois Field (GF) multiplication and other SIMD operations [1]. Multiple decoder algorithms with different degrees of complexity are stored in external memory, such that for a particular RS data packet the one with less computational complexity can be employed, depending on the error/erasure information. By using our method, the packet retransmission rate is decreased, resulting in improved spectral efficiency. The improved spectral efficiency is reflected by a total link budget improvement of up to one dB.*

1. Introduction

The Reed-Solomon error correction capability t is determined by the code's minimum distance $2t+1$. Algorithms capable of correcting more than t errors are described in [2],[3],[4]. Correcting more than t errors implies additional computational complexity, reflected in additional silicon area and increased power consumption.

For large alphabet size Reed-Solomon decoders, the use of maximum-likelihood (ML) or near-to-ML decoding algorithms is ruled out because of their computational complexity. Simplified VLSI implementations of the Koetter-Vardy algorithm are reported in the literature [5]. In [6], Lammarca describes a soft input RS algorithm based on an algebraic bounded distance decoder. The decoder iterates through several decoding attempts, with different numbers of erasures and compares the Euclidean distance between the candidate codeword and the received data with a threshold defined by the acceptance criterion. The iteration starts by assuming $2t$ erasures and each iteration decreases the expected number of erasures by two.

To some extent, our approach is similar to the one presented in [5]. Channel State Information (CSI) from the convolutional decoder is also used to declare the erasures iteratively, such that we maximize the error correction capabilities of the RS decoder. Most of the erroneous data packets contain less than t errors per packet. In these cases, only the less computationally demanding Peterson-Gorenstein-Zierler RS decoding algorithm (see [5] for details), which suits our processor architecture, is used. If there are more than t errors in a data packet, a more complex decoding algorithm, a combination of the Messey and Forney decoding schemes, is employed. In our method we start with two erasures, based on the observation that erasure reliability information is inversely proportional to the number of erasures. Our decision to increase the number of erasures and start a new iteration is based on the degree of the error locator polynomial, as described in the next section. Based on Monte-Carlo analysis, for the end-to-end DVB-T/H system, our method provides a total link budget improvement of up to one dB.

This paper is organized as follows: Section II describes the overall algorithm. Section III presents our concatenated convolutional RS decoder for erasure declaration. Section IV describes our algorithm for error and erasure decoding. Numerical simulation results are given in Section V, followed by conclusions in Section VI.

2. Iterative Decoding

1) The decoding process has the following steps:
2) The convolutional decoder outputs the hard decision bit and its reliability probability. The bits are packed into bytes and the total reliability probability for the bytes is derived, as described in Section III.
3) The erasures are chosen from the less reliable bytes in the RS packet.
4) The RS decoding starts in error-only decoding mode, with no erasures. For this first step of the decoding, a less computationally intensive method for the Reed-Solomon decoder is employed. If at the end of the error-only decoding process all the syndromes are zero, then the decoding is declared successful.
5) If after the first decoding process not all the syndromes are zero, most likely there are more than *t* errors. The error/erasure combinations are described by *2r-s=2t* and illustrated in TABLE 1 for t = 8. The decoder starts to search for each error/erasure combination shown in the table and corrects for both errors and erasures.
6) The decoder stops either if all the syndromes of the corrected codeword are zero or if the maximum number of erasures shown in the TABLE 1 is achieved.

In the flowchart (Figure 1) the Reed-Solomon decoding process starts at step 4).

TABLE 1
TABLE 2 ERROR/ERASURE COMBINATION

Errors (r)	Erasures (s)
8	0
7	2
6	4
5	6
4	8
3	10
2	12
1	14
0	16

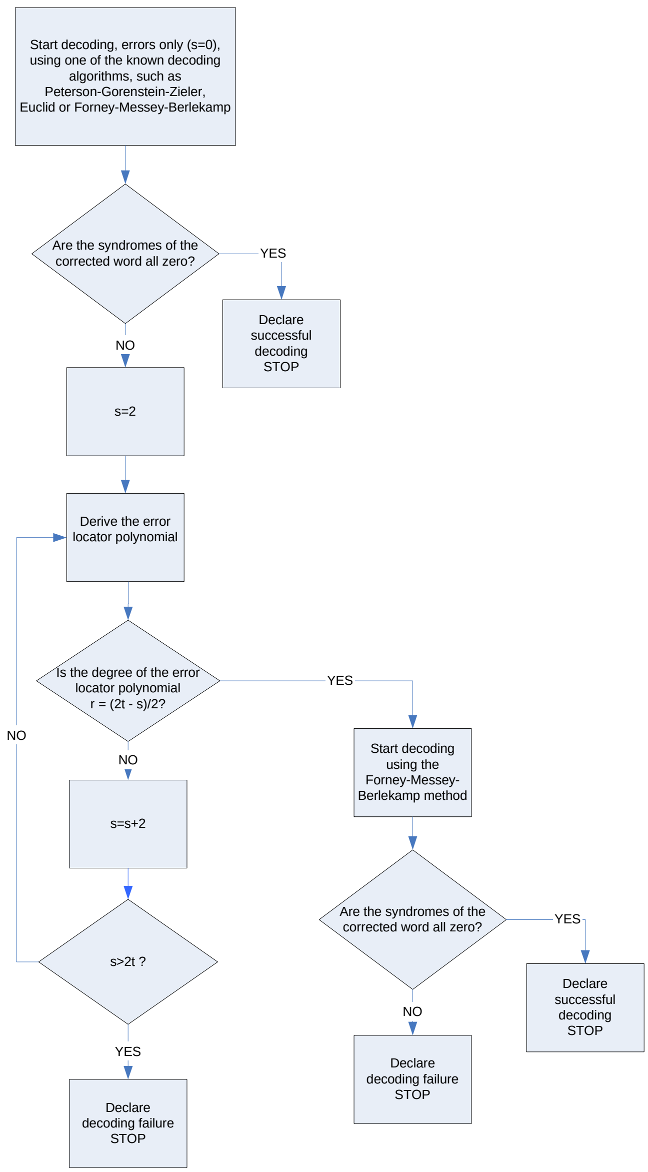

Figure 1 Reed-Solomon decoding flowchart.

3. Erasure Declaration

The Reed Solomon decoder is able to correct only the error/erasure combinations described in TABLE 1. In order for the RS decoder to be able to correct more than *t* errors, it has to be provided with the erasure information. The erasure information is provided by the convolutional decoder as a hard bit accompanied by the bit reliability probability.

A block diagram of the concatenated convolutional RS decoder is presented in Figure 2. The inner decoder is a SOVA (Soft Output Viterbi Algorithm) convolutional decoder [7]. The inner decoder processes the input signal stream **x** and its corresponding SNR estimates. The output of the SOVA is the decoded information bit (hard decision) stream **u** and its associated log-likelihood ratio reliability (soft decision, or **L** value) sequence **L(u)**. The **L(u)** and **u** streams go through a bit to byte packaging block to generate the packed byte streams **v** and **L(v)** for outer processing. The outer deinterleaver outputs RS codewords **w** and their associated erasure information **E(w)** for the erasure RS decoder.

A. *Inner Decoder Description*

The following notation will be used for the SOVA decoder: code rate *1/N*, constraint length *K*, decoder memory *m=K-1*, and number of states 2^m. The accumulated path metric specific to each state S_k, $S = 0,....2^m - 1$ is described by the following equation:

$$\Gamma(S_{k-1}, S_k) = \Gamma(S_{k-1}) + \sum_{n=0}^{N-1} \frac{E_s}{N_o} (y_{kn} - x_{kn}^{(i)})^2 \qquad (1)$$

where $i = 1,2$, k is the time instant, y_{kn} is the n^{th} output bit from the encoder if the state transitions from $k-1$ to k, $x^{(i)}_{kn}$ is the n^{th} soft input bit on the i^{th} path at time k, and $\frac{E_s}{N_o}$ is the corresponding SNR estimate for input bit x^i_{kn}. The survivor state is chosen as:

$$\Gamma(S_k) = \min[\Gamma(S_{k-1}, S_k)] \qquad (2)$$

and it is stored with its associated survivor pre-state.

In the above classical Viterbi algorithm, for two paths ending at the same node, only the information regarding the survivor path is saved. The information regarding the other path (the competing path) is discarded. To estimate the bit reliability, the algorithm looks at the path metric difference:

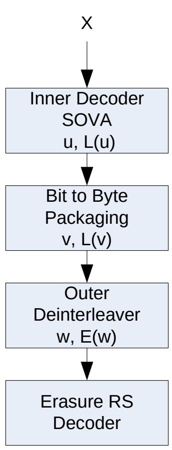

Figure 2 Block diagram for the concatenated convolutional/erasure RS decoder. The Erasure RS Decoder is described in Figure 1.

$$\Delta_{S_k}^k = \max[\Gamma(S_{k-1}, S_k)] - \min[\Gamma(S_{k-1}, S_k)] \qquad (3)$$

Here is how it works; at time instant k for state S_k, $S = 0,....2^m - 1$, the reliability of selecting the right survivor path at time k for state S_k can be approximated as the path metric difference $\Delta_{S_k}^k$. Each time when a survivor path is selected, the reliability values along the survivor path for the associated node need to be updated if the survivor and competing paths yield different bit decisions on those nodes. Figure 3 shows such an example. The survivor and competing paths yield different bit decisions for nodes $k-2$ and $k-3$. The reliability values along the survivor path on these two nodes need to be updated as:

$$L_{S_k}^p = \min[L_{S_k}^p, \Delta_{S_k}^k] \qquad (4)$$
$$p = (k-2) \quad and \quad (k-3)$$

When making decoding decisions along the final ML survivor path, the corresponding reliability values for the decoded bit will be output as the "soft" decision for further processing.

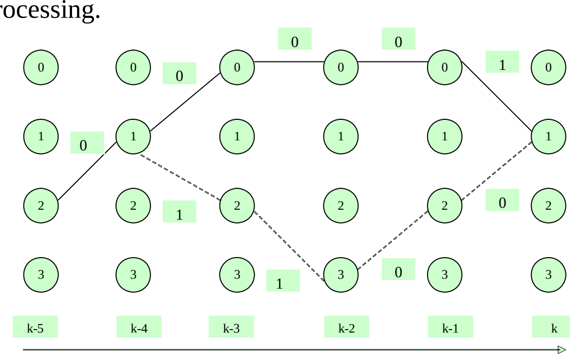

Figure 3 Illustration of SOVA survivor, competing paths, and reliability updates.

B. *Byte Packaging for SOVA Information*

The SOVA output bit stream *u* and its associated reliability information stream *L(u)* are run through a bit to byte packaging process to prepare for outer RS decoding. Due to the byte processing nature of RS decoding, the associated output reliability information for the packaged byte $v = [u_k, u_{k-1}, ...u_{k-7}]$ is:

$$L(v) = \min[L(u_{k-i})], \quad i = 0,1,......k-7.$$

C. *Outer Deinterleaving and Erasure Declaring*

The decoded byte stream v and its associated reliability information stream $L(v)$ run through the outer deinterleaving block to generate the RS decoder input codeword w, and its corresponding reliability information block $L(w)$. For an RS(k,n) code, we find the *n* smallest locations $L(w_i), i = 0,1,......n-1$ as possible candidates for erasures. Since the erasure RS decoder requires more MIPs than the error-only RS decoder, an iterative decoding process is proposed, where the first normal pass of RS decoding is an error-only RS decoder. In most cases, when the number of errors is eight or less, as in the example of RS(204,188,t=8), the decoding is successful. When the number of errors is greater than eight, the first pass error-only decoding fails, and the RS decoder enters an iterative decoding process, in which it runs RS decoders with 2, 4, 6, and 8 erasures. When using over eight erasures, base upon our simulations, the erasure positions can not be predicted accurately enough to see any improvement in the decoder performance. The iterative decoding process stops either when the RS decoding is successful or when the next RS package is ready for processing.

4. Erasure Decoding

The error and erasure decoding algorithm described in this section is applicable to any communication protocol utilizing RS GF(2^m) fields. It applies for any GF(2^m) field for either full length or shortened codes. The total number of errors and erasures the algorithm can correct for is described in TABLE 1. For most, if not all communication protocols, the link budget assumed in the standard is such that if the SNR at the receiver is in the bounds specified by the conformance testing, most of the time the RS packets have no errors at all. For example, based on our simulations for the DVB-T/H protocols, in the Quasi Error Free (QEF) transmission mode, on average only 0.02% of the packets have errors and the average number of erroneous symbols per packet is less than four. As the SNR further decreases, the number of packets with errors grows exponentially.

For an average of 99.98% of the time, there are no errors in the packets or, the number of errors is less than eight. In our simulations, for less than eight errors, we used the Peterson-Gorenstein-Zierler [8] error-only correction algorithm. As shown in [1], the Peterson-Gorenstein-Zierler decoder suits well our parallel DSP architecture.

For low SNR, when more than eight erroneous symbols per packet are encountered, the error and erasure algorithm we developed, which is described below, becomes useful. The following notations are used throughout this section:

x=(x_0, x_1, … , x_{n-1}) is a valid codeword,

r=(r_0, r_1,…, r_{n-1}) is the received codeword,

$g(D) = \prod_{i=0}^{2t-1}(D-\alpha^{i+j})$ is the generator polynomial,

where: α^j is the j^{th} unity root, and *2t+1* is the code's minimum distance.

m=(m_1, m_2, … , m_t) is the error position vector,

e=(e_{m1}, e_{m2}, … , e_{mt}) is the error magnitude vecror,

S_i , $0 \le i \le \delta - 2$ are the syndromes,

$B(D) = \prod_{i=0}^{2t-1}(1-L_pD) = 1 + B_1D + ... + B_tD^t$ is the connection polynomial with error locators $L_p = \alpha^{m_p}$, *p=1,…, t*.

Any solution of the system equation:

$$S_i = \sum_{p=1}^{t} B_p S_{i-p} \quad (1)$$

leads to the error locators. The error locations and magnitudes are determined using Peterson-Gorenstein-Zierler decoding, described in [8].

After decoding, the syndromes are computed again and tested against zero. If all the syndromes are zero, the decoding process is successful and the next packet is passed to the decoder. If there are nonzero syndromes, the error and erasure decoding is enabled. The error and erasure locators are $L_p = \alpha^{m_p}$, *p=1,…, r* and $Z_p = \alpha^{m_p}$, *p=1,…, s*, respectively, with *r* and *s* described in TABLE 1.

The decoding procedure is described in Figure 1 and a brief description follows. The interested reader can find a more detailed description in [8]. First the original syndromes are calculated and modified using Forney's method [3]:

$$T_i = \sum_{p=0}^{s} \lambda_p S_{i+s-p}$$

where *i=0,…,δ-s-2* and T_i are the modified syndromes, which contain the error and erasure information. Next, the error locations are calculated using Massey's method [8]. At this point, if the degree of the error locator polynomial is different from the assumed value *r*, the algorithm is reiterated for a different number of errors and erasures as illustrated in the flow chart.

5. Numerical Simulation

The iterative decoding algorithm described in this paper has been tested in the end-to-end DVB-T simulated system, specified by ETSI EN 744 V1.4.1 (2001-01). TABLE 3 illustrates the simulation results for 16-QAM mode with different propagation channels. The simulations were performed in fixed point C using the Sandbridge simulation tools [9].

TABLE 3 Results for the DVB-T 16-QAM test modes.

		SNR Improvement (dB)	SNR Improvement (dB)
Modulation	Code Rate	Ricean Fading Channel	Rayleigh Fading Channel

16-QAM	1/2	0.56	0.82
16-QAM	2/3	0.12	0.15
16-QAM	3/4	0.0	0.73
16-QAM	5/6	0.0	0.38
16-QAM	7/8	0.23	1.02

6. Conclusions

In this paper a new iterative Concatenated Convolutional RS decoding algorithm has been proposed for long RS codewords. The proposed algorithm has demonstrated up to one dB improvement on some of the DVB-T simulation test cases specified by the ETSI EN 744 V1.4.1 (2001-01) standard. However in some cases, code rates 3/4 and 5/6, we did not see any significant improvement. We expect that combining our method with the General Minimum Distance decoding algorithm [4] will further improve the error rate without adding significant computational expense. Since the algorithm is used only in those cases where the number of errors is more than *t* per data packet and in all other cases the error-only decoder is employed, there is no significant computational expense added to the processor.

References

[1] Suman Mamidi, Daniel Iancu, Andrei Iancu, Michael J. Schulte, and John Glossner "Instruction set extensions for Reed-Solomon encoding and decoding", *ASAP*, Samos, pp. 231-237, July 2005.

[2] E. Berlecamp, "Bounded distance+1 soft decision Reed-Solomon decoding", *IEEE Trans. Inform. Theory.* vol. 42, pp. 704-720, May 1996.

[3] G. D. Forney, "Generalized Minimum Distance Decoding", *IEEE Trans. Inform Theory*, vol. IT-12. pp. 125-131, Apr. 1996.

[4] D. J. Taipale and M.B.Pursley, "An improvement to generalized minimum distance decoding", *IEEE Trans. Inform. Theory,* vol. IT-37, pp. 167-172, Jan. 1991.

[5] W. J. Gross, F. R. Kschischang, R. Koetter, and P. G. Gulak, "Towards a VLSI architecture for interpolation-based soft-decoding Reed-Solomon decoders" *Journal of VLSI Sign. Proc.*, vol. 39, pp. 93-111, 2005.

[6] M.Lamarca, J.Sala-Alvarez, and A.Martinez, "Iterative decoding algorithm for RS-convolutional concatenated codes", *Proc. of 3rd Intl. Symposium on Turbo Codes and Related Topics*, pp.543-546, Brest (France), Sept. 2003

[7] J. Hagenauer and P. Hoeher "A Viterbi algorithm with soft decision outputs and its applications", *Proc. Of GLOBECOM*, pp. 1680-1686, Nov. 1989

[8] S. B. Wicker, Error Control Systems for Digital Communication and Storage, Prentice Hall Inc.,1995.

[9] J. Glossner, D. Iancu, J. Lu, E. Hokenek, and M. Moudgill, "A Software Defined Communications Baseband Design", IEEE Communications Magazine, vol. 41, no. 1, pp. 120-128, January 2003.