

Unified reed solomon decoder for Correcting burst and random errors

VINOTH RAJ.R
M.E VLSI Design,
Srinivasan engineering college,
Perambalur.
vinoth.valentino.raj@gmail.com,

SARANYA REVATHI.G
Assistant Professor,
Srinivasan engineering college,
Perambalur.
sarshan8@gmail.com

Abstract - Uncertainties Are The Major Feature Of Burst And Random Errors. This Fact Makes Error Correction become a Difficult Task. Similar With Other Forward Correction Codes (FEC), When Using RS Codes As Channel Coding, The Errors Occurred In Transmission Procedure Are Typically Divided Into Random Errors And Burst Errors And Also Erasures. Currently, For Decoding RS Codes With Random-Error Correction And Burst Errors With Previously Employed Techniques Is Quite Difficult. Although Erasure Is One Type Of Error Which Is Not Corrected By Means Of Those Techniques Simultaneously With Burst And Random Errors. My Proposed Scheme With Unified Architecture For Reed Solomon Decoder Combined With Burst And Random Errors Is Possible And To Yield Maximum Number Of Bit Error Correcting Capabilities In Lesser Cycle Times, When Compared With Previously Employed Schemes And Also To Correct Erasure, Is Firstly Presented For Multi-Mode Decoding Requirements.

Keywords— *burst errors, random errors, erasures, RiBC algorithm, UHD decoder, key equation solver, galois field.*

I. Introduction

Reed-Solomon (RS) codes have been widely employed for error correction in modern digital communication and data storage systems. Similar with other forward correction codes (FEC), when using RS codes as channel coding, the errors occurred in transmission procedure are typically divided into random errors and burst errors. Currently, for decoding RS codes with random-error correction, numerous literatures have given extensive studies on theoretical algorithms as well as hardware implementations [1]–[3], [8]. However, for specific RS burst-error decoder design, although some dedicated algorithms had been reported [4], [5], the VLSI implementations for these burst-error correcting algorithms are still under-investigated, which are limited by their cubic computation complexity $O(4nt^2)$, where n and t length of CODE WORD and traditional correction capability respectively. Recently, a novel low-complexity RS burst-error correcting algorithm that only

requires $O(2nt)$ computation was proposed by Wu [6]. It can be proved that the algorithm is capable of correcting a long burst of errors together with possible random errors. In this brief, developed from the above new algorithm, a high-speed reformulated inversion less burst-error correcting (RiBC) algorithm is proposed, and a unified hybrid decoding (UHD) architecture that supports three decoding modes is presented for the first time. It will be shown that, compared with traditional RS decoder, the proposed UHD architecture can achieve significantly better burst-error correcting capability. The structure of this paper is organized as follows. Section II describes the proposed RiBC algorithm. The architecture and latency of new UHD decoder are presented in Section III. Section IV provides the key equation solver and section V gives the proposed working model. Final conclusion is drawn in Section VI.

ii. RiBC algorithm :

The RiBC algorithm is a kind of list decoding algorithm. Eight polynomials are updated simultaneously in each iteration. The proposed RiBC algorithm is targeted for correcting burst error plus some random errors. If the channel condition guarantees that only single long burst of errors occurs, Wu [6] presented a low-complexity single long burst of errors correcting (sLBC) algorithm for that case. RiBC algorithm is very effective for correcting combination of burst errors and random errors (mode-1), while sLBC and rEEC algorithms are well-suited for single burst (mode-2) and random errors and erasures (mode-3) correction. By observing the three algorithms, it can be founded that they share many common or similar computation steps. high-speed RiBC algorithm for RS code burst-error correcting.

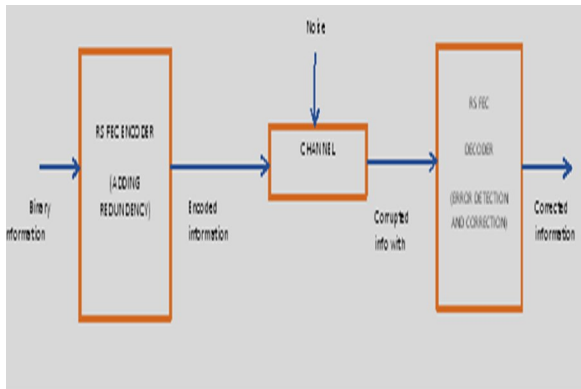


Fig 2.1 (overall block diagram of RS codes)

Information through a practical communication system may be corrupted by noise in the channel during transaction. Because of increasing demand for development of reliable telecommunication and wireless systems, it is important for communication systems to have adequate means for the detection and correction of errors in the information received

over communication channels and thus made error control coding an important and necessary step in communication system design. In recent years, original work of Shannon, Hamming & Golay on error control coding has turned into an important branch of communication system engineering which provides mathematical structure of the codes and large variety of very efficient coding algorithms and techniques. Reed-Solomon coding is a very efficient and popular Forward Error Correction technique discovered by Reed and Solomon in 1960 [1]. Reed-Solomon (RS) codes are among the most widely used block error-correcting codes in digital communication and storage systems [2] and are very effective in correcting random symbol errors and random burst errors. Therefore they are applied in many systems such as storage devices, mobile communications, digital television/DVB, high-speed modems etc. RS codes are adopted by various standards like DVB T, DVB S, DVB DSNG, DVB C, IEEE802.16, OTN G.709. The rest of the paper is organized as follows. Section II briefly review forward error correction codes. Details of the proposed RS Decoder for IEEE 802.16 network are described in Section III. The FPGA implementation and its results are presented in Section IV. Section V provides conclusion.

iii. UHD decoder:

Based on this interesting similarity, a unified hybrid decoding(UHD) architecture that is capable of correcting these three different types of errors pattern Three types of lines illustrate data flows for different work modes: solid line for

mode-1, dashed line for mode-2 and dotted line for mode-3. Different blocks are used to process different steps. The RiBM decoder, the UHD decoder can achieve significantly enhanced burst-error correcting capability with multiple work modes. In the channel environments that likely generate long burst of errors ($f > 8$), such as high-density storage systems, the traditional RiBM decoder fails to decode the CODE WORDs for its limited error correcting capability, while UHD decoder can be still effective (mode-1 and mode-2). For random error-and-erasure correction (mode-3), the proposed UHD design has lower throughput than RiBM.

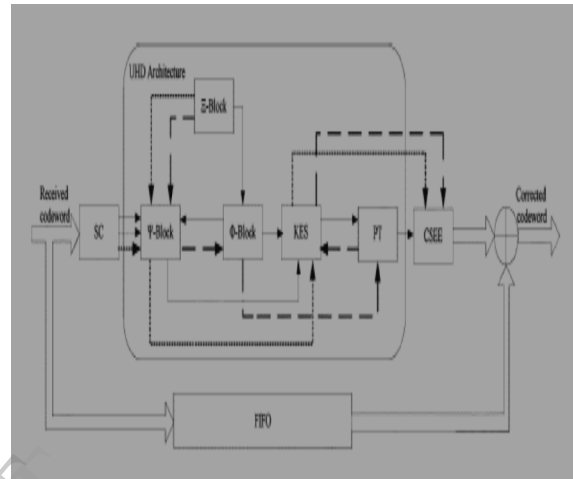


fig 3.1(Overall architecture of proposed UHD decoder)

In coding theory, **Reed-Solomon (RS) codes** are non-binary^[1] 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, or correct up to $\lfloor t/2 \rfloor$ symbols. As an erasure code, it can correct up to t known erasures, or it can detect and correct combinations of errors and erasures. Furthermore, RS codes are suitable as multiple-burst bit-error correcting codes, since a sequence of $b + 1$ consecutive bit errors can affect at most two symbols of size b .^[2] The choice of t is up to the designer of the code, and may be selected within wide limits.

In Reed-Solomon coding, source symbols are viewed as coefficients of a polynomial $p(x)$ over a finite field. The original idea was to create n code symbols from k source symbols by over sampling $p(x)$ at $n > k$ distinct points, transmit the sampled points, and use interpolation techniques at the receiver to recover the original message. That is not how RS codes are used today. Instead, RS codes are viewed as cyclic BCH codes, where encoding symbols are derived from the coefficients of a polynomial constructed by multiplying $p(x)$ with a cyclic generator polynomial. This gives rise to efficient decoding algorithms

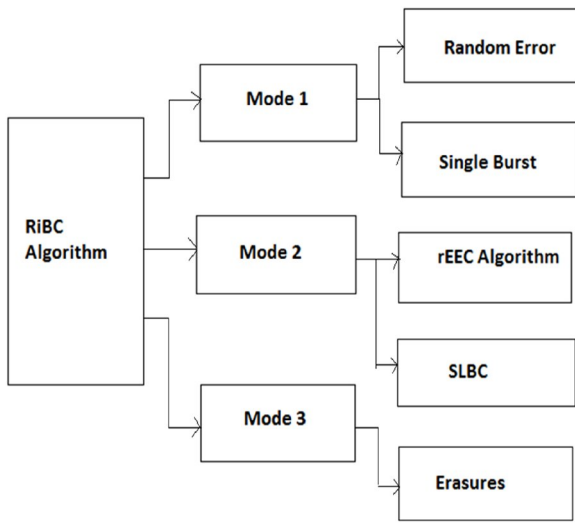


fig 3.2 (RiBC model)

iv. key equation solver:

This section describes the decoding algorithm, the overall architecture, and details of the proposed architecture. Three main algorithm, the modified Euclidean algorithm, the Chien search and Forney's algorithms are described and their hardware architectures are presented.

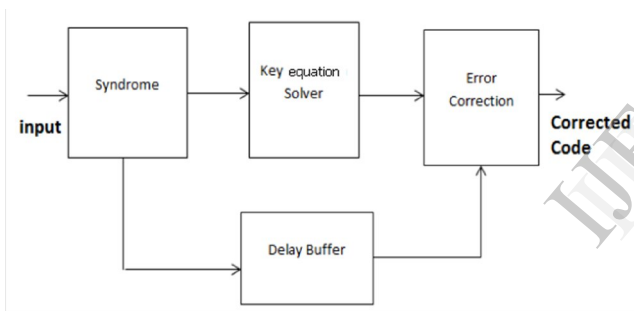


Fig 4.1(KES BLOCK)

The overall architecture of the proposed RS decoder is shown in Figure 1. The RS decoder consists of the syndrome calculation block, the key equation solver block, Chien search and Forney's algorithm blocks and FIFO memory. The syndrome calculation (SC) block evaluates the syndrome polynomial based on the symbols received as input(DIN). The Key Equation Solver (KES) block performs the concurrent computation of the error locator polynomial $\sigma(x)$ and the error value polynomial $\omega(x)$, based on a Modified Euclidean Algorithm implementing the Berlekamp-Massey algorithm. Chien search and Forney's algorithm(CSF) on the polynomials computed by the KES block, the CSF block implements the Chien search (time-domain error location) and the Forney algorithm to evaluate the quotient to be added to the received symbol in order to correct the error. The FIFO memory is used to delay the input symbols according to them to the latency.

The length of the FIFO is determined by the latency introduced by the SC, KES, and CSF blocks sequence. The decoder corrects received error symbols by adding the delayed FIFO outputs to error values. When an error is recognized, its output (DOUT) corresponds to the correct symbol.

V.PROPOSED WORKING MODEL:

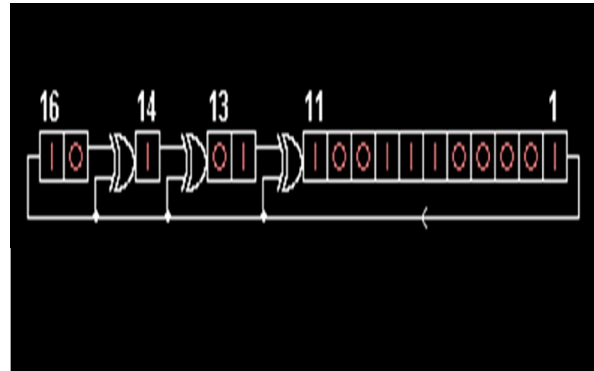


Fig 5.1 (16-bit Galois LFSR)

GALOIS LFSR:

LFSR is a shift register whose input bit is a linear function of its previous state. The most commonly used linear function of single bits is XOR. Thus LFSR is most often a shift register whose input bit is driven by the XOR of some bits of the overall shift register value. The feedback tap numbers in white correspond to a primitive polynomial. Bit position that affects the next state are called the taps. From the above diagram the taps are (16,14,13,11). The right most of the LFSR is called the output bit. The taps are XOR'ed sequentially with the output bit and then feedback in to the left most bit. The sequence of bits in the right most position is called the output stream. The bits in the LFSR which influences the input bits are called taps. The maximum length LFSR produces a M sequences unless it contains all zeros.

The feedback polynomials is

$$x^{16} + x^{14} + x^{13} + x^{11} + 1$$

GALOIS FIELD:

It is a field that contains a finite number of elements. The finite fields are classified by size there is exactly one field up to isomorphism of size p^k for each prime 'p' and the positive integer 'k'. each finite field 'q' is the specified field of the polynomial $x^q - x$.

VI. Conclusion

The complexity of RS (255,223) decoder proposed in this paper is about 130,000 gates, the total latency is 560 cycles, and the throughput is 180Mbps under 20MHz. By using the proposed architecture, we reduced its area, and improved its speed. Its performance is better than other similar RS (255,223) decoder. This RS decoder works correctly, which can correct up to 16 errors, reaching the predetermined goal.

[8] X. Zhang and J. Zhu, "High-throughput interpolation architecture for algebraic soft-decision Reed-Solomon decoding," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 3, pp. 581–591, Mar. 2010.

REFERENCES

[1] D. V. Sarwate and N. R. Shanbhag, "High-speed architectures for Reed-Solomon decoders," *IEEE Trans. Very Large Scale Integr.(VLSI) Syst.*, vol. 9, no. 5, pp. 641–655, Oct. 2001.

[2] T. Zhang and K. K. Parhi, "On the high-speed VLSI implementation of errors-and-erasures correcting Reed-Solomon decoders," in *Proc. ACM Great Lake Symp. VLSI (GLVLSI)*, 2002, pp. 89–93.

[3] Z. Wang and J. Ma, "High-speed interpolation architecture for soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 9, pp. 937–950, Sep. 2006.

[4] E. Dawson and A. Khodkar, "Burst error-correcting algorithm for Reed-Solomon codes," *Electron. Lett.*, vol. 31, pp. 848–849, 1995.

[5] L. Yin, J. Lu, K. B. Letaief, and Y. Wu, "Burst-error-correcting algorithm for Reed-Solomon codes," *Electron. Lett.*, vol. 37, no. 11, pp. 695–697, May 2001.

[6] Y. Wu, "Novel burst error correcting algorithms for Reed-Solomon codes," in *Proc. IEEE Allerton Conf. Commun., Control, Comput.*, 2009, pp. 1047–1052.

[7] S. Shamshiri and K.-T. Cheng, "Error-locality-aware linear coding to correct multi-bit upsets in SRAMs," in *Proc. IEEE Int. Test Conf.*, 2010, pp. 1–10.