A High-Speed and Low-Latency Reed-Solomon Decoder Based on a Dual-Line Structure

Hyeong-Ju Kang and In-Cheol Park

Dept. of EE, KAIST, 373-1, Guseong-dong, Yuseong-gu, Daejeon, Korea

{dk,icpark}@ics.kaist.ac.kr

Abstract

This paper presents a new decoding structure of Reed-Solomon (RS) codes that are widely used for channel coding. The serial structures have long latency and the parallel structures are not fast enough to deal with the demands of high-speed decoding. To achieve both short latency and fast operation, the summation of the products of syndromes is eliminated and the difference used to calculate the error locator polynomial is incrementally updated. The proposed structure called a dual-line structure can operate as fast as the serial structure and has as short latency as the parallel structure. In addition, the dual-line structure is regular and easy to implement. Experimental results confirm these advantages at the cost of a small hardware increase.

1. Introduction

Decoding schemes of RS codes are established decades ago [1-2], and many types of RS decoders have been proposed since then [3-7]. They are categorized by the employed algorithms, which can be classified into two: the Berlekamp-Massey (BM) algorithm [1-5] and the Euclid algorithm [2,6-7]. The BM algorithm has an advantage of less complexity and the Euclid algorithm is good at its regularity.

In this paper, we propose a new RS decoding structure that can provide fast operation and short latency. The reason why the previous structures cannot have fast operation and short latency together is that a difference is calculated from the saved syndromes and the saved intermediate error-locator polynomial and is used to produce a new intermediate error-locator polynomial in each iteration. In the proposed structure, the differences are saved and used directly to produce intermediate error-locator polynomials. Since the step of difference calculation is eliminated, the critical path is significantly reduced.

This paper is organized as follows. The Berlekamp-Massey (BM) algorithm is introduced in Section 2, and the dual-line structure is proposed in Section 3. Section 4 compares the proposed structure with the previous ones, and concluding remarks are made in Section 5.

2. Berlekamp-Massey (BM) algorithm

The inversion-less BM algorithm has been published as follows when t is the error correction capability [3-5]:

$$\Delta_i = \sum_{j=0}^{i-1} \Lambda_j^{(i-1)} S_{i-j} \tag{1}$$

$$L_i = \delta(i - L_{i-1}) + (1-\delta) L_{i-1} \tag{2}$$

$$\begin{bmatrix} \Lambda^{(i)}(x) \\ B^{(i)}(x) \end{bmatrix} = \begin{bmatrix} \varepsilon^{(i-1)} & -\Delta_i x \\ \delta & (1-\delta)x \end{bmatrix} \begin{bmatrix} \Lambda^{(i-1)}(x) \\ B^{(i-1)}(x) \end{bmatrix} \tag{3}$$

$$\varepsilon^{(i)} = \delta \cdot \Delta_i + (1-\delta) \cdot \varepsilon^{(i-1)}, \text{ for } i = 1, 2, \cdots, 2t. \tag{4}$$

The initial conditions are $\Lambda^{(0)}(x) = 1$, $B^{(0)}(x) = 1$, $L_0 = 0$, and $\varepsilon^{(0)} = 1$. If $\Delta_i \neq 0$ and $2L_{i-1} \leq i-1$, $\delta = 1$, otherwise $\delta = 0$. $\Lambda^{(2t)}(x) = \Lambda(x)$ is the error-locator-polynomial.

The BM algorithm can be implemented in two structures. The first one is the parallel structure where one iteration is processed in one cycle, and the other one is the serial structure where one iteration is processed in 2(t+1) cycles.

3. Dual-Line Structure

A new RS decoder structure will be proposed in this section. In the previous structures, the syndromes and $\Lambda^{(i)}(x)$ are saved in registers and Δ_i is calculated from them. In the dual-line structure, however, the registers have an intermediate $\Delta_k^{(i)}$ s that become Δ_k s when $\Lambda^{(k-1)}(x) = \Lambda^{(i)}(x)$. Using this structure, the summation in (1) can be eliminated and thus the critical path can be reduced.

Let us define $\Delta_k^{(i-1)}$ as follows:

$$\Delta_k^{(i-1)} = \sum_{j=0}^{k-1} \Lambda_j^{(i-1)} S_{k-j}. \tag{5}$$

$\Delta_k^{(i-1)}$ s are the intermediate values that becomes Δ_k s required in the algorithm when $i = k$. While Δ_k s are obtained from $\Lambda_j^{(i)}$ s and S_{k-j} s in the previous structure, $\Delta_k^{(i)}$ s are calculated from $\Delta_k^{(i-1)}$ s with the following equations.

If $\Lambda_j^{(i)} = \varepsilon^{(i-1)} \cdot \Lambda_j^{(i-1)}$, $\Delta_k^{(i)} = \varepsilon^{(i-1)} \cdot \Delta_k^{(i-1)}$. If $\Lambda_j^{(i)} = \varepsilon^{(i-1)} \cdot \Lambda_j^{(i-1)} + \Delta_i \cdot B_{j-1}^{(i-1)}$,

$$\Delta_k^{(i)} = \varepsilon^{(i-1)} \cdot \Delta_k^{(i-1)} + \sum_{j=0}^{k-1} \Delta_i \cdot B_{j-1}^{(i-1)} \cdot S_{k-j}. \tag{6}$$

If the last change of $B^{(i)}(x)$ occurs at the l-th cycle (in other words, $\Delta_l \neq 0$ and $2L_{l-1} \leq l-1$), $B_{j-1}^{(i-1)} = \Lambda_{j-i+l}^{(l-1)}$ and

$$\Delta_k^{(i)} = \varepsilon^{(i-1)} \cdot \Delta_k^{(i-1)} + \Delta_i \Delta_{k'}^{(l-1)} \tag{7}$$

, where $k' = k - i + l$. These equations lead to the fact that $\Delta_k^{(i)}$ can be obtained from $\Delta_k^{(i-1)}$ and $\Delta_k^{(l-1)}$.

The above equations are implemented with two series of registers, C and D. Fig. 1 is an example for $t = 5$. The values of C registers are defined as $C_k^{(i-1)} = \Delta_{k+i}^{(i-1)}$. If $\Lambda_j^{(i)} = \varepsilon^{(i-1)} \cdot \Lambda_j^{(i-1)}$, $C_k^{(i)} = \Delta_{k+i+1}^{(i)} = \varepsilon^{(i)} \cdot C_{k+1}^{(i-1)}$. If $\Lambda_j^{(i)} = \varepsilon^{(i-1)} \cdot \Lambda_j^{(i-1)} + \Delta_i \cdot B_{j-1}^{(i-1)}$, the following equation can be induced from (7).

$$C_k^{(i)} = \Delta_{k+i+1}^{(i)} = \varepsilon^{(i-1)} \cdot C_{k+1}^{(i-1)} + \Delta_i C_{k+1}^{(l-1)} \tag{8}$$

, where $\Delta_i = \Delta_i^{(i-1)} = C_0^{(i-1)}$. If we define $D_k^{(i-1)}$ as

$$D_k^{(i)} = \begin{cases} C_k^{(i-1)} & , \text{when } \Delta_i \neq 0 \wedge 2L_{i-1} \leq i-1 \\ D_k^{(i-1)} & , \text{otherwise,} \end{cases} \tag{9}$$

then

$$C_k^{(i)} = \begin{cases} \varepsilon^{(i-1)} \cdot C_{k+1}^{(i-1)} & , \text{if } \Delta_i = 0 \\ \varepsilon^{(i-1)} \cdot C_{k+1}^{(i-1)} + \Delta_i D_{k+1}^{(i-1)} & , \text{otherwise.} \end{cases} \tag{10}$$

$\Delta_k^{(i)}$ s can be obtained from C registers and D registers with the above equations.

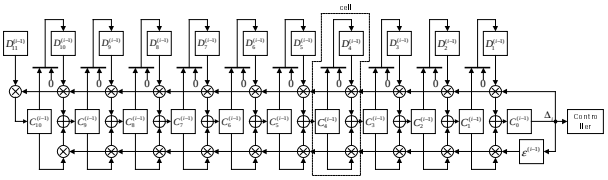

Fig. 1. Dual-line RS decoding structure.

A direct implementation of the above equations can lead to $6t$ registers, $2t$ for C registers, $2t$ for D registers, and $2t$ for $\Lambda_j^{(i)}$ calculation. However, a structure that requires only $4t$ registers can be obtained. Since $\Delta_k^{(i-1)}$s are saved in $2t-(i-1)$ registers, $C_k^{(i-1)}$s are zero for $k \geq 2t-(i-1)$. These zero-valued registers can be used to calculate $\Lambda_j^{(i)}$.

We summarize the above equations as follows:

$$C_k^{(i)} = \begin{cases} \varepsilon^{(i-1)} \cdot C_{k+1}^{(i-1)} & , when\ \Delta_i = 0 \\ \varepsilon^{(i-1)} \cdot C_{k+1}^{(i-1)} + \Delta_i D_{k+1}^{(i-1)} & , when\ \Delta_i \neq 0 \end{cases} \quad (11)$$

$$D_k^{(i)} = \begin{cases} 0 & , when\ k = 2t-i \\ C_k^{(i-1)}, when\ k \neq 2t-i \wedge (\Delta_i \neq 0 \wedge 2L_{i-1} \leq i-1) & \\ D_k^{(i-1)}, otherwise. & \end{cases} \quad (12)$$

When i becomes $2t$, $\Lambda(x)$ can be obtained by setting $\Lambda_j = C_j$, for $0 \leq j \leq t$. This structure is named a dual-line structure because it consists of two series of registers, C registers and D registers.

4. Comparisons

The dual-line structure is compared to the previous structures in terms of the number of arithmetic units, latency, and delay. Four decoders that use the parallel structure, the serial structure, the improved serial structure, and the dual-line structure are implemented for RS(255,245) code that is used in cable modem and has the same error-correction capability as RS code used in DVD decoders.

Table Ⅰ shows the estimation results. In the critical path row, T_{MUL} is the delay time of the Galois field multiplier, T_{ADD} is that of the Galois field adder, and T_{ADD_TREE} is that of the adder tree used in the summation of the parallel structure. The serial structures have less arithmetic units. However, their buffer is very long, and it is estimated that the area differences between the above structures are small. The critical path of the dual-line structure is the same as that of the serial structure. As one iteration in the dual-line structure consists of one cycle, it can generate the error-locator-polynomial in $2t$ cycles, which is the same latency as that of the parallel structure. Therefore, the dual-line structure is superior in the view of the delay and the latency with a small hardware increase.

Another advantage of the proposed structure is its regularity. The parallel and the serial structures are irregular, leading to a difficulty in implementation. As shown in Fig. 1, the dual-line structure is a series of the cell that is enclosed by a dotted line in the figure. This means it is very regular and easy to implement.

5. Conclusions

In this paper, we have proposed an RS decoder structure based on a new concept. In the proposed structure, the dual-line structure, the intermediate differences are saved and calculated from the former intermediate differences. The summation that increases the delay in the parallel structure and the latency in the serial structure is eliminated in the proposed structure. Consequently, it can operate as fast as the serial structure and has as short latency as the parallel structure. These advantages were confirmed by comparing the proposed one with several other types of RS decoders in terms of area and latency. In addition, the proposed structure is very regular and easy to implement. It costs only small area-increase. The proposed structure can be used in applications that require high-speed and low-latency, such as high-end DVD decoders.

Acknowledgment

This work was supported (in part) by the Korea Science and Engineering Foundation through the Micro Information and Communication Remote Object-oriented System (MICROS) center at KAIST, Korea.

6. References

[1] E. Berlekamp, *Algebraic Coding Theory*. New York: McGraw-Hill, 1968.

[2] S. B. Wicker and V. K. Bhargava, *Reed-Solomon Codes and Their Application*. New York: IEEE Press, 1994.

[3] I. S. Reed, M. T. Shih, and T. K. Truong, "VLSI Design of Inverse-free Berlekamp-Massey Algorithm," *IEE Proc.-E*, vol. 138, pp. 295-298, Sep. 1991.

[4] J. H. Jeng and T. K. Truong, "On Decoding of Both Errors and Erasures of a Reed-Solomon Code Using an Inverse-Free Berlekamp-Massey Algorithm," *IEEE Trans. Commun.*, vol. 47, pp. 1488-1494, Oct. 1999.

[5] H. C. Chang, C. B. Shung, and C. Y. Lee, "A Reed-Solomon Product-Code (RS-PC) Decoder Chip for DVD Applications," *IEEE J. Solid-State Circuits*, vol. 36, no. 8, pp. 229-238 Feb. 2001.

[6] M. H. Lee, S. B. Choi, and J. S. Chang, "A High Speed Reed-Solomon Decoder," *IEEE Trans. Consumer Electron.*, vol. 41, pp. 1142-1149, Nov. 1995.

[7] S. Kwon and H. Shin, "An Area-Efficient VLSI Architecture of a Reed-Solomon Decoder-Encoder for Digital VCRs," *IEEE Trans. Consumer Electron.*, vol. 43, pp. 1019-1027, Nov. 1997.

TABLE Ⅰ

COMPARISON OF STRUCTURES

		Parallel	Serial	Serial [10]	Dual-line (proposed)
Arithmetic Units	Adders	10	2	2	10
	Multipliers	17	3	3	21
	Registers	17	18	18	23
Latency cycle (buffer length)		274	408	343	274
Critical path		$2 \cdot T_{MUL} + T_{ADD\_TREE} + T_{ADD}$	$T_{MUL} + T_{ADD}$	$T_{MUL} + T_{ADD}$	$T_{MUL} + T_{ADD}$