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. Although many decoding structures have been developed, 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

As Reed-Solomon (RS) codes can correct many errors and have an excellent ability to correct burst errors, the codes are being widely used in many applications including data communication such as ADSL and cable modem, optical data storages such as CD and DVD, and mobile communication such as W-CDMA.

Decoding schemes of RS codes are established decades ago [1-2], and many types of RS decoders have been proposed since then [3-8]. They are categorized by the employed algorithms, which can be classified into two: the Berlekamp-Massey (BM) algorithm [1-6] and the Euclid algorithm [2,7-8]. The BM algorithm has an advantage of less complexity and the Euclid algorithm is good at its regularity.

The implementations of RS decoders have been based on slow structures as the applications have not required high speed. However, the modern applications like DVD decoders require faster and faster decoding speed. One of the previous structures, the serial structure, can deal with this speed, but it requires long latency. Another structure, the parallel structure, has shorter latency, but cannot operate fast enough for the future applications.

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.

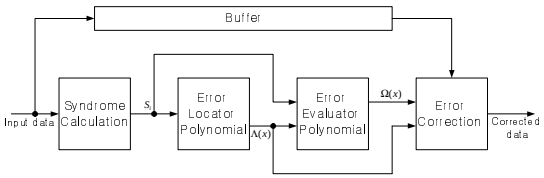

Fig. 1. Block diagram of an RS decoder.

This paper is organized as follows. Section 2 explains the RS decoding scheme using the BM algorithm, and the previous structures are introduced in Section 3. The dual-line structure is proposed in Section 4. Section 5 compares the proposed structure with the previous ones, and concluding remarks are made in Section 6.

2. REED-SOLOMON DECODER

RS codes are denoted by three parameters, N, K, and t, where N is the length of a block, K is the length of the information symbols, and $N-K=2t$ is the number of the parity symbols. With the (N,K) RS code, we can correct up to t errors. Fig. 1 shows the structure of an RS decoder.

From the input data, the first block calculates syndromes defined as follows:

$$S_i = \sum_{j=0}^{N-1} r_j \left(\alpha^i\right)^j \quad , i = 1,2,\cdots,2t \tag{1}$$

, where r_j is the j-th input data and α is the primitive element of Galois field. If no input data is affected by errors, all syndromes must be zero. Otherwise, the syndromes have all information on errors. From the syndromes, the second block of Fig. 1 generates an error-locator-polynomial that contains the information of error positions by the BM algorithm. In the original BM algorithm, a multiplicative-inversion circuit that needs much area and long delay is required. To cope with this, the inversion-less BM algorithm has been published as follows [3-6]:

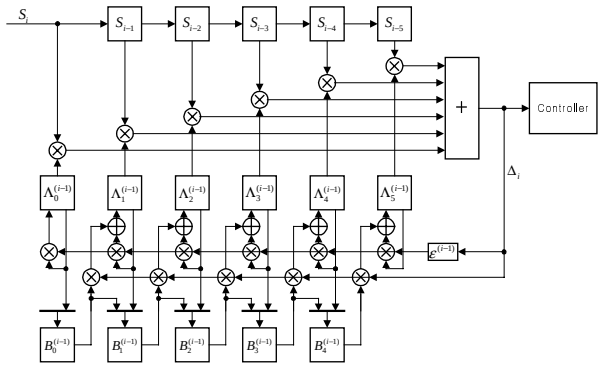

Fig. 2. Parallel RS decoding structure.

$$\Delta_i = \sum_{j=0}^{i-1} \Lambda_j^{(i-1)} S_{i-j} \tag{2}$$

$$L_i = \delta(i - L_{i-1}) + (1-\delta) L_{i-1} \tag{3}$$

$$\begin{bmatrix} \Lambda^{(i)}(x) \\ B^{(i)}(x) \end{bmatrix} = \begin{bmatrix} \varepsilon^{(i-1)} & -\Delta_i x \\ \delta & (1-\delta)x \end{bmatrix} \begin{bmatrix} \Lambda^{(i-1)}(x) \\ B^{(i-1)}(x) \end{bmatrix} \tag{4}$$

$$\varepsilon^{(i)} = \delta \cdot \Delta_i + (1-\delta) \cdot \varepsilon^{(i-1)}, \text{ for } i = 1, 2, \cdots, 2t. \tag{5}$$

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.

Given an error-locator-polynomial and syndromes, an error-evaluator-polynomial is computed as follows:

$$\Omega(x) = [1 + S(x)]\Lambda(x) \mod x^{2t+1} \tag{6}$$

, where $\Omega(x)$ is the error-evaluator-polynomial and $S(x)$ is the polynomial whose coefficients are syndromes [1,2]. To implement this equation, a circuit is required for the polynomial multiplication. However, the error-evaluator-polynomial is usually computed by reusing the error-locator-polynomial block, as the circuit that calculates the summation of (2) is similar to the circuit for the polynomial multiplication.

3. PREVIOUS STRUCTURES

3.1. Parallel structure

Fig. 2 shows a circuit in which the BM algorithm is directly used. The figure shows a parallel decoder when $t = 5$. The upper part calculates the summation in (2), while the lower part implementing the matrix equation (4) computes $\Lambda^{(i)}(x)$ and $B^{(i)}(x)$. In this structure, one iteration of the BM algorithm can be completed in one clock cycle, thus the BM algorithm can be completed in $2t$ cycles.

3.2. Serial structure

A serial structure is shown in Fig. 3, where the BM algorithm can be implemented with a few multipliers and shift registers. The upper part calculates the summation in (2), and the lower part implements (4) as the parallel structure. Each part takes $(t+1)$ cycles to complete its calculation. The lower part can start after the upper part completes its computation of the same

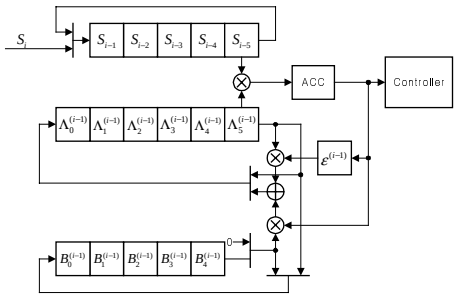

Fig. 3. Serial RS decoding structure.

iteration, and the upper part can start after the lower part completes the computation of the previous iteration. Therefore, each iteration takes $(2t+2)$ cycles. To reduce the number of cycles for each iteration, a serial structure that makes the two parts operate simultaneously was proposed in [6]. If we use this structure, each iteration can be reduced to $(t+1)$ cycles.

3.3. Comparison

In a system that uses RS codes, the complexity of the RS decoder is not a problem because the decoder is not a large component in the entire system. While the area of the RS decoder is not critical, the RS decoder can be a bottleneck in terms of clock frequency and latency. If the RS decoder operates slowly, a low clock frequency must be used for the entire system. The latency of the RS decoder becomes a part of the latency of the entire system.

While the critical path of the parallel structure consists of two multipliers and several adders, that of the serial structure has only one multiplier and one adder. Therefore, the serial structure can operate with a faster clock. The serial structure, however, takes about $(t+1)$ times as long latency as the parallel structure. The parallel structure has a merit in latency, and the serial structure has a merit in speed.

4. DUAL-LINE STRUCTURE

A new RS decoder structure will be proposed in this section. It operates as fast as the serial structure and has as short latency as the parallel structure. 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 $\Lambda^{(k-1)}(x) = \Lambda^{(i)}(x)$. Although $\Lambda^{(i)}(x)$ is changing in each iteration, the relationship can be maintained by properly updating the values of registers. Using this structure, the summation in (2) can be eliminated and thus the critical path can be reduced.

4.1. Principle

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

$$\Delta_k^{(i-1)} = \sum_{j=0}^{k-1} \Lambda_j^{(i-1)} S_{k-j}. \tag{9}$$

$\Delta_k^{(i-1)}$ s are the intermediate values that becomes Δ_k s required in

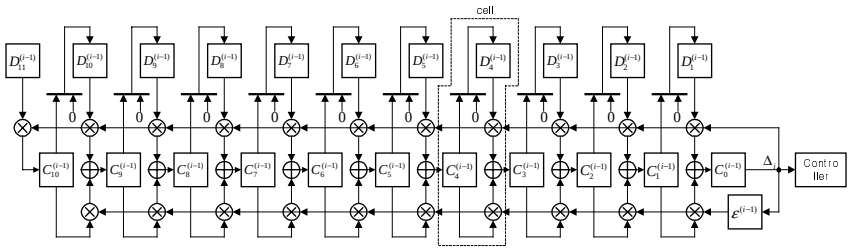

Fig. 4. Dual-line RS decoding structure.

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

$$\Delta_k^{(i)} = \varepsilon^{(i-1)} \cdot \Delta_k^{(i-1)} + \sum_{j=0}^{k-1} \Delta_i \cdot B_{j-1}^{(i-1)} \cdot S_{k-j}. \quad (10)$$

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

$$\begin{aligned} \Delta_k^{(i)} &= \varepsilon^{(i-1)} \cdot \Delta_k^{(i-1)} + \sum_{j=0}^{k-1} \Delta_i \cdot B_{j-1}^{(i-1)} \cdot S_{k-j} \\ &= \varepsilon^{(i-1)} \cdot \Delta_k^{(i-1)} + \Delta_i \sum_{j=0}^{(k-i+l)-1} \Lambda_j^{(l-1)} \cdot S_{(k-i+l)-j} \\ &= \varepsilon^{(i-1)} \cdot \Delta_k^{(i-1)} + \Delta_i \Delta_{k'}^{(l-1)} \end{aligned} \quad (11)$$

, where $k' = k - i + l$. These equations lead to the fact that $\Delta_k^{(i)}$ can be obtained from $\Delta_k^{(i-1)}$ and $\Delta_k^{(l-1)}$. If this concept is used, the summation in (2) does not have to be calculated in each iteration.

4.2. Detailed dual-line structure

The equations in the previous sub-section are implemented with two series of registers, C and D, and some arithmetic units that connect them. Fig. 4 is an example for $t = 5$. C registers have $\Delta_k^{(i)}$ s and D registers have the correction terms. The values of C registers are defined as $C_k^{(i-1)} = \Delta_{k+i}^{(i-1)}$. Since $\Delta_k^{(i)}$ s are updated in each iteration, $C_k^{(i)}$ s are also updated in each iteration to maintain that relationship as follows.

If $\Lambda_j^{(i)} = \varepsilon^{(i-1)} \cdot \Lambda_j^{(i-1)}$, $C_k^{(i)} = \Delta_{k+i+1}^{(i)} = \varepsilon^{(i)} \cdot \Delta_{(k+1)+i}^{(i-1)} = \varepsilon^{(i)} \cdot C_{k+1}^{(i-1)}$. 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 (11).

$$\begin{aligned} C_k^{(i)} = \Delta_{k+i+1}^{(i)} &= \varepsilon^{(i-1)} \cdot \Delta_{k+i+1}^{(i-1)} + \Delta_i \Delta_{k+i+1-i+l}^{(l-1)} \\ &= \varepsilon^{(i-1)} \cdot \Delta_{(k+1)+i}^{(i-1)} + \Delta_i \Delta_{(k+1)+l}^{(l-1)} \\ &= \varepsilon^{(i-1)} \cdot C_{k+1}^{(i-1)} + \Delta_i C_{k+1}^{(l-1)} \end{aligned} \quad (12)$$

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

$$D_k^{(i)} = \begin{cases} C_k^{(i-1)} & , when\ \Delta_i \neq 0 \wedge 2L_{i-1} \leq i-1 \\ D_k^{(i-1)} & , otherwise, \end{cases} \quad (13)$$

then

$$C_k^{(i)} = \begin{cases} \varepsilon^{(i-1)} \cdot C_{k+1}^{(i-1)} & , if\ \Delta_i = 0 \\ \varepsilon^{(i-1)} \cdot C_{k+1}^{(i-1)} + \Delta_i D_{k+1}^{(i-1)} & , otherwise. \end{cases} \quad (14)$$

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

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 registers can be used to calculate $\Lambda_j^{(i)}$ by defining them as follows:

$$C_k^{(i-1)} = \Lambda_{k-2t+i-1}^{(i-1)}, \ for\ k \geq 2t-i+1. \quad (15)$$

Because $\Lambda_j^{(i)} = 0$ for $j > i$,

$$C_k^{(i-1)} = 0, \ for\ k > 2t. \quad (16)$$

Therefore, $\Delta_k^{(i)}$ s and $\Lambda_j^{(i)}$ s can be calculated with C registers whose number is $(2t+1)$.

We summarize the above equations for k = 0, 1, …, 2t as follows:

$$C_k^{(i)} = \begin{cases} \varepsilon^{(i-1)} \cdot C_{k+1}^{(i-1)} & , when\ \Delta_i = 0 \\ \varepsilon^{(i-1)} \cdot C_{k+1}^{(i-1)} + \Delta_i D_{k+1}^{(i-1)} & , when\ \Delta_i \neq 0 \end{cases} \quad (17)$$

$$D_k^{(i)} = \begin{cases} 0 & , when\ k = 2t-i \\ C_k^{(i-1)}, & when\ k \neq 2t-i \wedge (\Delta_i \neq 0 \wedge 2L_{i-1} \leq i-1) \\ D_k^{(i-1)}, & otherwise. \end{cases} \quad (18)$$

Initial values are

$$C_k^{(0)} = \begin{cases} S_{k+1} & , for\ 0 \leq k \leq 2t-1 \\ 1 & , for\ k = 2t \end{cases} \quad (19)$$

$$D_k^{(0)} = \begin{cases} S_k & , for\ 1 \leq k \leq 2t-1 \\ 0 & , for\ k = 2t \\ 1 & , for\ k = 2t+1. \end{cases} \quad (20)$$

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.

Since the calculation of the error-evaluator-polynomial is similar to that of error-locator-polynomial, this structure can be reused to produce the error-evaluator-polynomial. When the dual-line structure completes $2t$ iterations, $C_k^{(2t)}$ registers are re-initialized properly and processed again. After $(t+1)$ cycles, the error-evaluator-polynomial is in C registers.

TABLE Ⅰ
COMPARISON OF STRUCTURES

		Parallel	Serial	Serial [6]	Dual-line (proposed)
Gate-count	Non-memory	10,580	5,446	6,338	13,618
	Memory	42,830	59,962	51,396	42,830
	Total	53,410	65,408	57,734	56,448
Latency cycles		274	408	343	274
Cycle time		4.71ns	2.99ns	2.70ns	2.97ns

Comparing to the parallel structure, the proposed structure need $(t-1)$ additional multipliers and t additional registers. Tthe critical path of the dual-line structure, however, consists of only one multiplier and one adder, which 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. 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. 6, 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. EXPERIMENTAL RESULTS

The dual-line structure is compared to the previous structures in terms of gate-count, 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. Each decoder was described in Verilog HDL and synthesized with UMC 0.25㎛ FS90A_B standard cell library. It was assumed that the error-evaluator-polynomial is generated by re-using the error-locator-polynomial block. A single-port SRAM memory is used for buffers. As memory read and write must occur simultaneously, the SRAM memory is divided into several banks, each of which is 8-bit wide and has 32 entries. The multiplier in [9] is used because it needs the least gates among the Galois filed multipliers.

Table Ⅰ shows the synthesis results. The parallel structure has the smallest area and the dual-line structure follows it. However, the decoders except the serial structure of the second column have almost the same area. The area differences between them are up to 4,500 gates, which is less than 10% of the decoder areas. This does not have a significant effect on the entire system. While the parallel structure and the dual-line structure have short latency, the serial structures have long latency. Due to the long latency, the serial structure has a large memory for buffers. In addition, this long latency lengthens the latency of the entire system and can degrade the performance of the communication systems. In terms of delay, the serial structures and the dual-line structure have shorter delay. This means a higher operating frequency can be used in those structures. To use a high operating frequency in the parallel structure, pipeline stages should be inserted, which increases the area and latency.

The dual-line structure can operate as fast as the serial structures. Moreover, it does not have the long latency that the serial structures have. Though the parallel structure also has the property of the short latency, the dual-line structure is better than the parallel structure in speed. In other words, the dual-line structure is superior in the view of the delay and the latency.

6. CONCLUSIONS

In this paper, we have proposed an RS decoder structure based on a new concept. In the previous structures, the difference that is used to produce an intermediate error-locator polynomial of the present iteration is calculated by summing the products of the syndromes and the intermediate error-locator polynomial of the former iteration. 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 MICROS center at KAIST, Korea.

7. 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] H. O. Burton, "Inversionless Decoding of Binary BCH Codes," *IEEE Trans. Inform. Theory*, vol. IT-17, pp. 464-466, Jul. 1971.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] L. Song and K. K. Parhi, "Efficient Finite Field Serial/Parallel Multiplication," in *Proc. Int. Conf. Application Specific Syst., Architectures and Processors*, 1996, pp. 72-82.