SMALL-AREA PARALLEL SYNDROME CALCULATION FOR STRONG BCH DECODING

Youngjoo Lee, Hoyoung Yoo and In-Cheol Park

KAIST

ABSTRACT

This paper presents a new optimization method to reduce the hardware complexity of syndrome calculation in strong BCH decoding. All the operations required in the parallel syndrome calculation are reformulated as a single matrix computation to enlarge the search area for common sub-expressions. The computational complexity of syndrome calculation is significantly reduced by finding and sharing common terms in the single matrix computation. Implementation results show that the proposed architecture saves 55% of area overheads compared to the conventional structure.

Index Terms— BCH code, Decoder, ECC, Syndrome.

1. INTRODUCTION

Error-correction codes are widely used in communication systems to recover multiple errors caused by poor environment. The Bose-Chaudhuri-Hochquenghem (BCH) code is one of algebraic codes, and it has been commonly applied to real-time systems because of its powerful error correction capability [1]. Moreover, recent applications such as advanced solid-state storage systems [2] and high-speed optical fiber communications [3] require high decoding throughput as well as large error-correction capability. Thus, massive-parallel BCH decoding is desired to satisfy such a high-throughput requirement, but the hardware overhead resulting from the high parallel factor is increased significantly. Therefore, a strong BCH decoder demands a structure that is efficient enough to lower the hardware complexity.

A BCH codeword is usually decoded by passing through three stages: i) syndrome calculation (SC) for the received polynomial $R(z)$; ii) generation of an error-locator polynomial $\Lambda(z)$; and iii) finding error positions by applying the Chien search (CS). In general, the SC and CS blocks are complicated if they are implemented with a large parallel factor, and the parallel SC stage takes almost 30% of the whole decoder area [4]. While many advanced optimization schemes have been developed for the Chien search [5]-[7], only a few works have been reported to reduce the complexity of a massive-parallel SC block.

In the conventional optimization, the constant GF multiplier in a parallel SC block is formulated as a matrix multiplication and then common sub-expressions (CSEs) are shared to reduce redundant nodes. The iterative matching algorithm (IMA) [8] is commonly used to find CSEs from a matrix. A more advanced scheme was reported in [9], where even-indexed syndromes are generated by applying power operations to odd-indexed syndromes. As this approach needs to compute only the odd-indexed syndromes directly, it eliminates a half of registers required to store intermediate data. Moreover, the search space for CSEs is enlarged by grouping power operations associated with the same input. As there are more-than-one matrices and each matrix is processed separately in finding CSEs, however, the approach cannot find CSEs that may be resident between different matrices.

To enlarge the search space for CSEs, the proposed method transforms all the operations of parallel SC into a single matrix multiplication. As a result, the proposed method maximizes CSEs and reduces the complexity of parallel SC significantly.

2. PREVIOUS SYNDROME CALCULATIONS

The BCH code is characterized as (n, k, t), where n is the code length, k is the data length, and t is the error correction capability. The n-bit codeword (r_0, r_1,…, r_{n-1}) can be interpreted as a received polynomial, $R(z) = r_0 + r_1 z^1 + \ldots + r_{n-1} z^{n-1}$. In SC, $2t$ syndromes are computed using the following equation:

$$s_i = R(\alpha^i) = \sum_{j=0}^{n-1} r_j \alpha^{ij} = r_0 + r_1 \alpha^i + \cdots + r_{n-1} \alpha^{i(n-1)} \quad (1)$$

The conventional SC unit computing the i-th syndrome is shown in Fig. 1, where p and i stand for the parallel factor and the current iteration, respectively [1]. The p-parallel SC block, which can process p input bits in a cycle, takes $\lceil n/p \rceil$ cycles and consists of $2t$ such units to generate $2t$ syndromes simultaneously. In order to find CSEs, a constant multiplication over GF(2^m) is represented as a matrix multiplication [5]. For example, $y=x\alpha^i$ can be expressed as follows:

This work was supported in part by IT R&D program of MKE /KEIT No.KI10035202 and by the IC Design Education Center (IDEC).

$$y = x\alpha^i = (x_0\alpha^0 + x_1\alpha^1 + \cdots + x_{m-1}\alpha^{m-1})\alpha^i$$
$$= x_0\alpha^i + x_1\alpha^{i+1} + \cdots + x_{m-1}\alpha^{i+m-1} \quad (2)$$

It can be formulated as

$$\mathbf{y} = \begin{bmatrix} x_0 & x_1 & \cdots & x_{m-1} \end{bmatrix} \times \begin{bmatrix} \alpha_0^i & \alpha_1^i & \cdots & \alpha_{m-1}^i \\ \alpha_0^{i+1} & \alpha_1^{i+1} & \cdots & \alpha_{m-1}^{i+1} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_0^{i+m-1} & \alpha_1^{i+m-1} & \cdots & \alpha_{m-1}^{i+m-1} \end{bmatrix}$$
$$= \mathbf{x} \times \mathbf{A}_i \quad (3)$$

where **y** and **x** represent 1×m binary matrices each denoting an element in GF(2^m), and $\mathbf{A}_i$ is a m×m binary matrix for the constant GF multiplication. A bit-level optimization called IMA [8] is iteratively applied to find CSEs. A more efficient method was presented in [9], where even-indexed syndromes are derived from odd-indexed syndromes according to the following equation:

$$s_{2^r \times i} = R(\alpha^{2^r \times i}) = R^{2^r}(\alpha^i) = s_i^{2^r} \quad (4)$$

Note that every even-indexed syndrome can be obtained by applying a power operation to a certain odd-indexed syndrome. For example, Fig. 2(a) illustrates what power operations are needed to compute even-indexed syndromes for t=4. Only 4 odd-indexed syndromes are calculated from the received codeword, and then the remaining even-indexed syndromes are obtained by applying power operations. Therefore, the approach eliminates a half of registers that are otherwise needed to store temporary values for even-indexed syndromes. Moreover, the search space for CSEs can be enlarged, as multiple power operations that share the same input can be grouped together. The power operation, $y=x^w$, over GF(2^m) can be expressed as

$$y = x^w = (x_0\alpha^0 + x_1\alpha^1 + \cdots + x_{m-1}\alpha^{m-1})^w$$
$$= x_0\alpha^0 + x_1\alpha^w + \cdots + x_{m-1}\alpha^{(m-1)w} \quad (5)$$

where w is restricted to 2^r for an even-indexed syndrome. It can be reformulated as

$$\mathbf{y} = \begin{bmatrix} x_0 & x_1 & \cdots & x_{m-1} \end{bmatrix} \times \begin{bmatrix} \alpha_0^0 & \alpha_1^0 & \cdots & \alpha_{m-1}^0 \\ \alpha_0^w & \alpha_1^w & \cdots & \alpha_{m-1}^w \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_0^{(m-1)w} & \alpha_1^{(m-1)w} & \cdots & \alpha_{m-1}^{(m-1)w} \end{bmatrix}$$
$$= \mathbf{x} \times \mathbf{B}_w \quad (6)$$

where $\mathbf{B}_w$ is a m×m binary matrix representing the w-th power operation. From (6), r even-indexed syndromes,

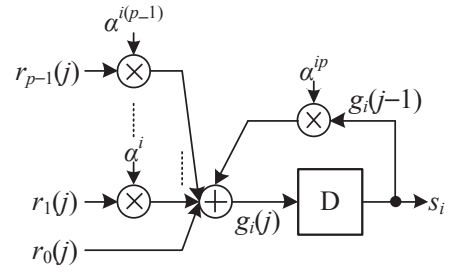

Fig. 1. A conventional p-parallel SC unit.

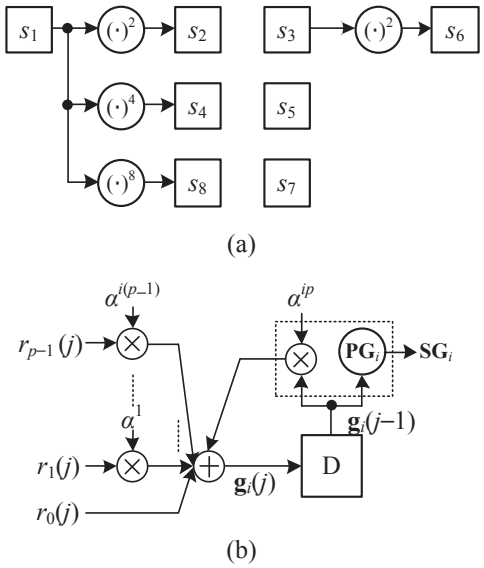

Fig. 2. A parallel SC based on power operations [9] (a) Calculation of even-indexed syndromes for t=4 and (b) the SC unit having an enlarged search space for common sub-expressions.

which are driven from s_1 by taking 2^1 to 2^r power operations, can be expressed in a matrix multiplication as follows:

$$\mathbf{SG}_1 = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \mathbf{s}_4 & \cdots & \mathbf{s}_{2^r} \end{bmatrix}$$
$$= \mathbf{s}_1 \times \begin{bmatrix} \mathbf{I} & \mathbf{B}_2 & \mathbf{B}_4 & \ldots & \mathbf{B}_{2^r} \end{bmatrix} = \mathbf{s}_1 \times \mathbf{PG}_1 \quad (7)$$

where $\mathbf{s}_i$ is an 1×m matrix corresponding to an m-bit syndrome, and $\mathbf{SG}_1$ is the first syndrome group that can be generated from s_1. Note that $\mathbf{PG}_1$ is an m×m(r+1) binary matrix that generates all the syndromes relevant to s_1. The i-th SC unit that calculates $\mathbf{SG}_i$ is drawn in Fig. 2(b), where i is an odd number less than 2t. Note that each unit has a separate search space denoted by a dashed box in Fig. 2(b).

3. THE PROPOSED ARCHITECTURE

The computational complexity of p-parallel SC is reduced in [9] by eliminating registers and CSEs in the enlarged matrices. As this approach has multiple search domains each

of which is searched independently, it may be possible to find more CSEs if all the search domains be merged together. In the proposed SC structure, all the operations of p-parallel SC, including the power operations needed to compute even-indexed syndromes, are collected into a single matrix multiplication so as to find as many CSEs as possible. To obtain a single matrix multiplication performing p-parallel SC, the i-th SC unit of the conventional architecture shown in Fig. 1 is formulated as

$$\begin{aligned}\mathbf{g}_i(j) &= \begin{bmatrix} r_0(j) & r_1(j) & \cdots & r_{p-1}(j) & \mathbf{g}_{i0}(j-1) & \mathbf{g}_{i1}(j-1) & \cdots & \mathbf{g}_{i,m-1}(j-1)\end{bmatrix} \\ &\times \begin{bmatrix} \alpha_0^0 & \alpha_1^0 & \cdots & \alpha_{m-1}^0 \\ \alpha_0^i & \alpha_1^i & \cdots & \alpha_{m-1}^i \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_0^{(p-1)i} & \alpha_1^{(p-1)i} & \cdots & \alpha_{m-1}^{(p-1)i} \\ \alpha_0^{ip} & \alpha_1^{ip} & \cdots & \alpha_{m-1}^{ip} \\ \alpha_0^{ip+1} & \alpha_1^{ip+1} & \cdots & \alpha_{m-1}^{ip+1} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_0^{ip+m-1} & \alpha_1^{ip+m-1} & \cdots & \alpha_{m-1}^{ip+m-1}\end{bmatrix} \\ &= \begin{bmatrix}\mathbf{R}(j) & \mathbf{g}(j-1)\end{bmatrix} \times \begin{bmatrix}\mathbf{C}_i \\ \mathbf{A}_i\end{bmatrix}\end{aligned} \tag{8}$$

where $\mathbf{g}_i(j)$ is an $1\times m$ binary matrix representing the m-bit intermediate value of the i-th syndrome and $\mathbf{R}(j)$ is an $1\times p$ binary matrix representing p bits of the received codeword to be fed at the j-th iteration. The $\mathbf{C}_i$ matrix is a $p\times m$ binary matrix relevant to the input bits. In (8), we can see that all the equations in a single SC unit are collected into a matrix. If power operations are used to reduce the number of registers, t $\mathbf{g}_i(j)$ values can be represented in a single matrix multiplication, where i is a positive odd number less than $2t$. The enlarged matrix multiplication that computes t intermediate values for the next iteration can be expressed as

$$\begin{aligned}\mathbf{G}(j) &= \begin{bmatrix}\mathbf{g}_1(j) & \mathbf{g}_3(j) & \cdots & \mathbf{g}_{2t-1}(j)\end{bmatrix} \\ &= \begin{bmatrix}\mathbf{R}(j) & \mathbf{g}_1(j-1) & \mathbf{g}_3(j-1) & \cdots & \mathbf{g}_{2t-1}(j-1)\end{bmatrix} \\ &\times \begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_3 & \cdots & \mathbf{C}_{2t-1} \\ \mathbf{A}_{1p} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{3p} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_{2t-1,p}\end{bmatrix} = \begin{bmatrix}\mathbf{R}(j) & \mathbf{G}(j-1)\end{bmatrix} \times \begin{bmatrix}\mathbf{X_R} \\ \mathbf{X_G}\end{bmatrix}\end{aligned} \tag{9}$$

where, $\mathbf{G}(j)$ is a $1\times mt$ binary matrix containing t temporary values of the j-th iteration. $\mathbf{X_R}$ is a $p\times mt$ binary matrix for the input codeword and $\mathbf{X_G}$ is an $mt\times mt$ binary matrix denoting t constant multiplications. Therefore, all the $\mathbf{g}_i(j)$ values can be obtained by multiplying the input vector [$\mathbf{R}(j)$ $\mathbf{G}(j-1)$] and the constant matrix including $\mathbf{X_R}$ and $\mathbf{X_G}$. The t odd-indexed syndromes can be obtained directly from $\mathbf{G}(j-1)$, and the remaining t even-indexed syndromes can be calculated based on power operations which can be formulated as in (6). The computation of $2t$ syndromes from $\mathbf{G}(j-1)$ can be expressed as

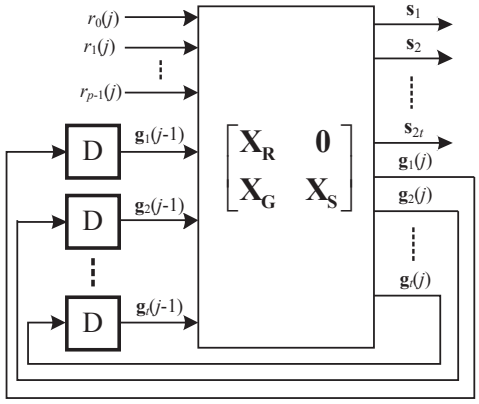

Fig. 3. The proposed p-parallel SC block.

$$\begin{aligned}\mathbf{S}(j) &= \begin{bmatrix}\mathbf{s}_1(j) & \mathbf{s}_2(j) & \cdots & \mathbf{s}_{2t}(j)\end{bmatrix} \\ &= \begin{bmatrix}\mathbf{g}_1(j-1) & \mathbf{g}_3(j-1) & \cdots & \mathbf{g}_{2t-1}(j-1)\end{bmatrix} \\ &\times \begin{bmatrix} \mathbf{I} & \mathbf{B}_2 & \mathbf{0} & \mathbf{B}_4 & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{B}_2 & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots\end{bmatrix} = \mathbf{G}(j-1) \times \mathbf{X_S}\end{aligned} \tag{10}$$

where $\mathbf{S}(j)$ is a $1\times 2mt$ binary matrix denoting $2t$ syndrome outputs at the j-th iteration. Note that $\mathbf{S}(j)$ becomes the desired syndromes at the last iteration. $\mathbf{X_S}$, which is an $mt\times 2mt$ binary matrix, consists of $m\times m$ identity matrices and $m\times m$ binary matrices for power operations. As a result, all the computations needed in p-parallel SC are transformed into a single matrix multiplication as follows;

$$\begin{bmatrix}\mathbf{S}(j) & \mathbf{G}(j)\end{bmatrix} = \begin{bmatrix}\mathbf{R}(j) & \mathbf{G}(j-1)\end{bmatrix} \times \begin{bmatrix}\mathbf{X_R} & \mathbf{0} \\ \mathbf{X_G} & \mathbf{X_S}\end{bmatrix} \tag{11}$$

As the single matrix multiplication in (11) covers all the partial products related to the computations of t intermediate values and $2t$ syndromes, we can find more CSEs from (11) than the previous works do. Therefore, a low-complexity SC block can be achieved by employing the proposed reformulated equation. Fig. 3 illustrates the proposed hardware structure of p-parallel SC, which is based on (11). Compared to the previous architectures associated with multiple matrix multiplications, the proposed architecture has only a single matrix multiplication.

4. IMPLEMENTATION RESULTS

For diverse code parameters, three SC blocks are designed based on the conventional architecture, the previous architecture that computes even-indexed syndromes using power operations [9], and the proposed architecture. They are all synthesized in a 0.13μm CMOS process, and tuned to operate at 375MHz for fair comparisons. The proposed

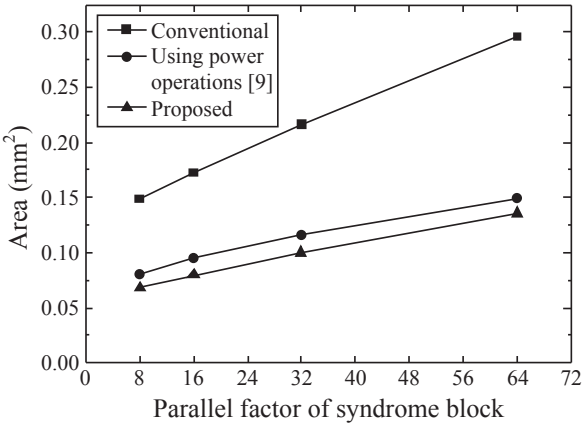

Fig. 4. The area of SC versus the parallel factor for the (8752, 8192, 40) shortened BCH code.

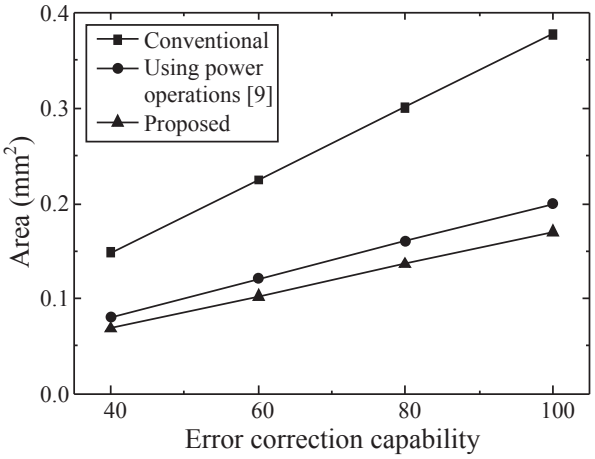

Fig. 5. The area of SC versus the error correction capability for the (*n*, 8192, *t*) shortened BCH code.

method always has less complexity compared to the previous works, as the single matrix multiplication guarantees finding more CSEs. For the (8752, 8192, 40) shortened BCH code, Fig. 4 compares how the area overheads of the three architectures are changing according to the parallel factor.

On the average, the proposed method reduces the hardware complexity by more than 15% and 55% compared to the SC method presented in [9] and the conventional method, respectively. Fig. 5 shows how the complexity is related to error correction capability with assuming a fixed message size of 8,192 bits. The proposed structure is always associated with the smallest area among the three structures. As a result, the proposed method that formulates all the SC operations into a single matrix multiplication is quite effective in searching for CSEs, reducing the hardware complexity of SC for a strong BCH decoder.

5. CONCLUSION

In this paper, an area-efficient SC architecture has been presented. The proposed architecture has only a single matrix multiplication, as all the operations in parallel syndrome calculation are formulated as a single matrix multiplication to enlarge search space for common sub-expressions (CSEs). Compared to the previous works having multiple matrices, the proposed method maximizes the number of CSEs and reduces the hardware complexity of a SC block by sharing such CSEs. Implementation results show that the proposed architecture reduces up to 55% and 15% of area requirements, compared to the conventional implementation and the previous power-based architecture [9], respectively. The proposed method becomes more attractive when designing a BCH decoder associated with the higher parallel factor and error-correction capability.

6. REFERENCES

[1] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2004.

[2] W. Liu, J. Rho, and W. Sung, "Low-power high-throughput BCH error correction VLSI design for multi-level cell NAND flash memories," in *Proc. IEEE Workshop SiPS*, 2006, pp. 303-308.

[3] *Forward Error Correction for Submarine Systems, ITU Telecommunication Standardization Sector*, ITU-T Recommendation G.975, 2000.

[4] W. Xueqiang *et al*., "A high-speed two-cell BCH decoder for error correcting in MLC nor flash memories," *IEEE Trans. Circuits Syst. II Exp. Briefs*, vol. 56, no. 11, pp. 865-869, Nov. 2009.

[5] Y. Chen and K. K. Parhi, "Small area parallel Chien search architectures for long BCH codes," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 5, pp. 545-549, May 2004.

[6] J. Cho and W. Sung, "Strength-reduced parallel Chien search architecture for strong BCH codes," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 55, no. 5, pp. 427-431, May 2008.

[7] Y. Lee, H. Yoo, and I.-C. Park, "Low-complexity parallel Chien search structure using two-dimensional optimization," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 8, pp. 522-526, Aug. 2011.

[8] M. Potkonjak, M. B. srivastava, and A. P. Chandrakasan, "Multiple constant multiplications: Efficient and versatile framework and algorithms for exploring common subexpression elimination," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 15, no. 2, pp. 151-165, Feb. 1996.

[9] H. Yoo, Y. Lee, and I.-C. Park, "Area-efficient syndrome calculation for strong BCH decoding," *IEE Electronics Letters*, vol. 47, no. 2, pp. 107-108, Jan. 2011.