A 26.9 K 314.5 Mb/s Soft (32400,32208) BCH Decoder Chip for DVB-S2 System

Yi-Min Lin, Chih-Lung Chen, Hsie-Chia Chang, and Chen-Yi Lee, *Member, IEEE*

***Abstract*—This paper provides a soft Bose–Chaudhuri–Hochquenghem (BCH) decoder chip with soft information from the LDPC decoder for the DVB-S2 system. In contrast with the hard BCH decoder, the proposed soft BCH decoder that deals with least reliable bits can provide much lower complexity with similar error-correcting performance. Moreover, the error locator evaluator is proposed to evaluate error locations without the Chien search for higher throughput, and the Björck–Pereyra error magnitude solver (BP-EMS) is presented to improve decoding efficiency and hardware complexity. The chip measurement results reveal that our proposed soft (32400, 32208) BCH decoder for DVB-S2 system can achieve 314.5 Mb/s with a gate-count of 26.9 K in standard 90 nm 1P9M CMOS technology. Extended for fully supporting 21 modes in the DVB-S2 system, our approach can achieve 300 MHz operation frequency with a gate-count of 32.4 K.**

***Index Terms*—Bose–Chaudhuri–Hochquenghem (BCH) codes, digital video broadcasting, DVB-S2, error-correction coding.**

I. Introduction

THE Bose–Chaudhuri–Hocquenghen (BCH) codes are cyclic codes which operate under Galois Field [1]. With similar code rate and codeword length, additional 0.6 dB coding gain is observed for BCH codes as compared with error-only decoding RS codes [2]. Hence, BCH codes with different block lengths, ranging from short (<100 b) to extremely long (>60 Kb), are very popular in storage and communication systems, such as DMB-T [3], DVB-T2 [4], and DVB-S2 [5] broadcasting systems.

The DVB-S2 system shown in Fig. 1 is the second-generation standard of digital video broadcasting via satellites, which was developed by the European Telecommunications Standard Institute (ETSI). It provides higher order modulations and a powerful FEC system based on serial concatenation of BCH codes and low-density parity check (LDPC) codes, leading to 30% channel capacity gain over the existing DVB-S standard [6]. For the high-speed and long-distance data transmission, the BCH codes with long block length are specified to suppress the error floor due to iterative LDPC decoding. The BCH codes have 21 different code rates, where there are $GF(2^{16})$ for DVB-S2 *normal frame* and $GF(2^{14})$ for DVB-S2 *short frame*. Three kinds of correcting ability, $t = 8$, 10, and 12, are defined for these code rates. The detailed specifications of the 21 kinds of BCH codes are listed in Table I. The long block length, which has 58 320 b at most, demands considerably complex arithmetic over a large finite field, resulting in higher circuit complexity. Moreover, the storage requirement becomes costly due to the large codeword that should be buffered for error correction, where a memory bank is about 100 K gate-count for a BCH codeword in the DVB-S2 system.

Over 50 decoding iterations in the LDPC decoder result in a long decoding latency and a short period of data output time. The BCH decoder for the DVB-S2 system is required to achieve 250 Mb/s throughput at most that accommodates the LDPC decoder output. Conventional BCH decoding contains *syndrome calculator*, *key equation solver*, and *Chien search* [1]. For long-block-length BCH decoders, the decoding latency is dominated by the syndrome calculator and Chien search. Pipeline architecture can improve the throughput, but more memory banks are required to store each stage codeword. Parallelism Chien search is another approach to enhance the throughput [2], [7], but it causes more hardware complexity. Therefore, a decoder with a single-stage pipeline and serial architecture forms a low-complexity design.

To achieve the throughput requirement with low complexity, the soft information from the LDPC decoder can be employed by BCH decoders in the DVB-S2 system. Soft information can help the decoder to choose the least reliable bits, and then the decoder can find certain errors from those bits. This paper introduces a soft BCH decoding algorithm that focuses on the least reliable bits instead of the entire codeword for lower complexity, and the possible error locations can be limited. Based on this concept, an error locator evaluator is proposed to eliminate the Chien search for higher throughput. Furthermore, the decoder can maintain a similar performance because the soft information from the LDPC decoder provides sufficient reliability. As a result, this paper can provide a low-complexity and high-throughput BCH decoder for the DVB-S2 system.

This paper is organized as follows. Section II gives the background of existing BCH decoding methods with hard decision and soft decision. The proposed algorithms and architectures are presented in Sections III and IV, respectively. Based on the proposed method, Section V demonstrates the simulation and implementation results of hard and soft BCH decoders for the DVB-S2 system. Finally, we conclude this paper in Section VI.

II. Background

An (N, K, t) BCH code has a block length of N bits and information length of K bits. While operating under $GF(2^m)$, it

Manuscript received February 08, 2010; revised April 23, 2010; accepted June 01, 2010. Date of current version October 22, 2010. This paper was approved by Guest Editor Mototsugu Hamada. This work was supported by the National Science Council and Ministry of Economic Affairs of Taiwan under Grant NSC 97-2221-E-009-166 and Grant 98-EC-17-A-01-S1-124.

The authors are with the Department of Electronics Engineering and Institute of Electronics, National Chiao Tung University, Hsinchu 30050, Taiwan (e-mail: ymlin@si2lab.org).

Digital Object Identifier 10.1109/JSSC.2010.2065630

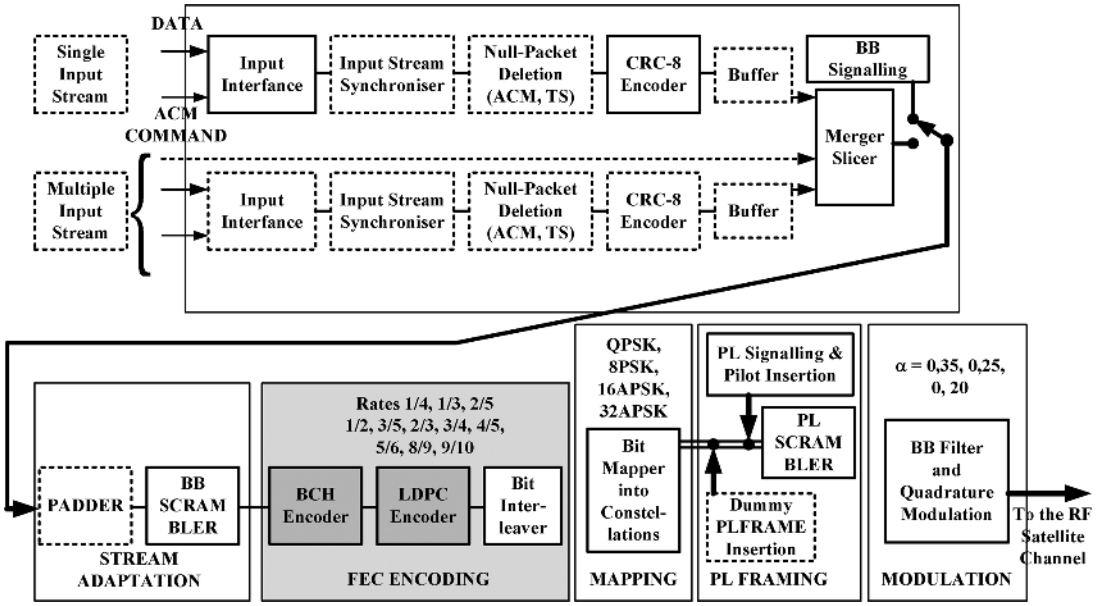

Fig. 1. Block diagram of the DVB-S2 system.

TABLE I
DVB-S2 SPECIFICATION (N: CODEWORD LENGTH, K: MESSAGE LENGTH)

	Normal Frame N_LDPC : 64800			Short Frame N_LDPC : 16200		
LDPC Code Rate	N_BCH	K_BCH	t	N_BCH	K_BCH	t
1/4	16200	16008	12	3240	3072	12
1/3	21600	21408	12	5400	5232	12
2/5	25920	25728	12	6480	6312	12
1/2	32400	32208	12	7200	7032	12
3/5	38880	38688	12	9720	9552	12
2/3	43200	43040	10	10800	10632	12
3/4	48600	48408	12	11880	11712	12
4/5	51840	51648	12	12600	12432	12
5/6	54000	53840	10	13320	13152	12
8/9	57600	57472	8	14400	14232	12
9/10	58320	58192	8	NA	NA	NA

has the error-correcting capability t, where $N-K \leq m \times t$. As shown in Fig. 2, the conventional BCH decoding contains three major steps. The received polynomial $R(x)$ is loaded into the FIFO and fed into the syndrome calculator to generate syndrome polynomial $S(x) = S_1 + S_2x^1 + \cdots + S_{2t}x^{2t-1}$, which is expressed as

$$S_j = R(\alpha^j) = \sum_{i=1}^{v} (\alpha^j)^{e_i} = \sum_{i=1}^{v} (\beta_{e_i})^j, \qquad \text{for j} = 1 \sim 2\text{t} \tag{1}$$

where α is the primitive element over $GF(2^m)$ and v is the number of actual errors. Notice that e_i is the ith actual error location and $\beta_{e_i} = \alpha^{e_i}$ indicates the corresponding error locator.

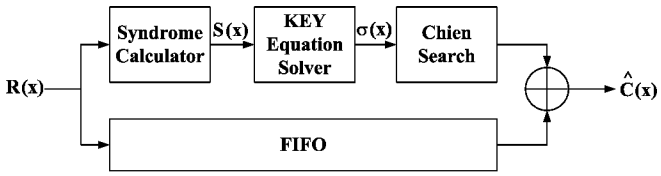

Fig. 2. Binary BCH decoding process.

The key equation solver is used to carry out the error location polynomial $\sigma(x)$, which is defined as

$$\begin{aligned}\sigma(x) &= (1 + x\beta_{e_1})(1 + x\beta_{e_2}) \cdots (1 + x\beta_{e_v}) \\ &= 1 + \sigma_1 x^1 + \sigma_2 x^2 + \sigma_3 x^3 \cdots + \sigma_v x^v. \end{aligned} \tag{2}$$

The key equation describing the relation between $S(x)$ and $\sigma(x)$ is derived as

$$\Omega(x) = S(x) \times \sigma(x) \bmod x^{2t} \tag{3}$$

where $\Omega(x)$ is the error evaluator polynomial. The most popular methods for solving the key equation are Berlekamp–Massey [8] and modified Euclidean algorithms [9]. After the key equation solver, Chien search is applied to find the roots of $\sigma(x)$. If an error is occurred at the e_ith position, α^{-e_i} will be a root of $\sigma(x)$. Finally, the estimated codeword polynomial $\hat{C}(x)$ is obtained by outputting $R(x)$ from the FIFO and inversing those values at error locations.

For higher error-correcting performance with the same code rate, the soft decoding algorithms of error control codes are the most popular methods and have recently prompted many research interests in BCH decoding. Forney developed a generalized-minimum-distance (GMD) method [10] by generating a list of candidate codewords with algebraic algorithms. Based on the block diagram of GMD decoding shown in Fig. 3, the GMD decoder uses a sorter to choose the least reliable bits and sends them into a candidate sequence generator to create several test

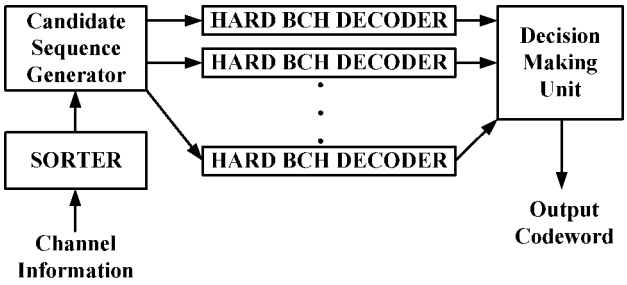

Fig. 3. GMD decoding block diagram.

sequences for hard BCH decoders. Each hard BCH decoder produces a candidate codeword to form a candidate codeword list, and a decision making unit chooses the most likely one from this list as the output codeword. Since several hard BCH decoders are utilized, a GMD BCH decoder is many times the hardware complexity of a hard BCH decoder. With the similar concept of candidate list, other algorithms such as Chase [11] and SEW [12] are also widely used in many applications. In the list decoding methods, the soft information is only used to generate test sequences. However, the soft information can be applied in the probability propagation, and several algorithms have been developed for advanced error correcting performance. A suboptimum maximum *a posteriori* (MAP) algorithm [13] with a Hamming SISO decoder for BCH decoding was proposed by Therattil and Thangaraj in 2005 [14]. Moreover, adaptive belief propagation, often used in LDPC decoding, was applied in soft BCH decoding in 2008 [15] by Baldi and Chiaraluce. Notice that the hardware complexity and the storage requirement become costlier due to the complex mathematical calculations.

In general, the complexity of a soft BCH decoder is much higher than a hard BCH decoder [10]–[15]. Nevertheless, soft BCH decoders with lower complexity can be revealed by focusing on the least reliable bits instead of the entire codeword. In 1997, a soft BCH decoding using error magnitudes to deal with the least reliable bits was developed [16]. This soft decoder collects unreliable bits for decoding. Due to the limited possible error locations, it can provide much lower complexity than other soft decoders, even lower than the traditional hard decoder.

The soft decoder corrects the errors only when all of the actual error locations are collected in the limited possible error locations; hence, the error-correcting performance highly depends on the reliability of the input signals. Fig. 4(a) shows that there is approximately 0.25 dB performance loss at BER $= 10^{-5}$ in the additive white Gaussian noise (AWGN) channel as compared with the hard (255,239) BCH decoder, indicating that soft information from the AWGN channel is not sufficiently reliable. However, the soft BCH decoder based on [16] can provide 0.25 dB performance gain with soft information from a 16-state Bahl–Cocke–Jelinek–Raviv (BCJR) decoder [17], as shown in Fig. 4(b). This is because the BCJR decoder provides more reliable soft information than the AWGN channel does. Accordingly, the soft BCH decoders with existing algorithms under the AWGN channel provide either better error-correcting performance or lower hardware complexity than a traditional hard BCH decoder. However, it is possible to provide a soft BCH decoder which has both correcting performance and hardware

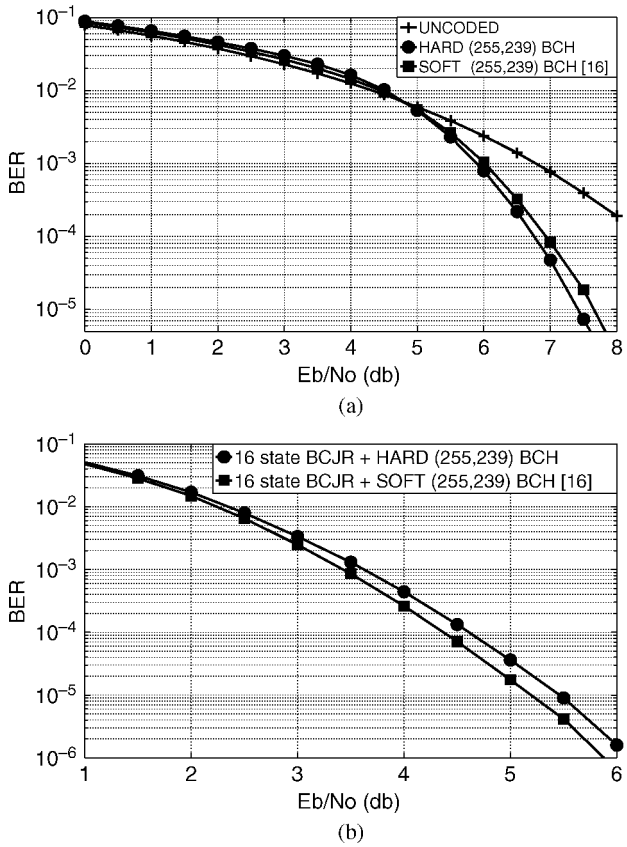

Fig. 4. Simulation results for (255,239) BCH under different soft information. (a) Information from the AWGN channel. (b) Information from the 16-state BCJR.

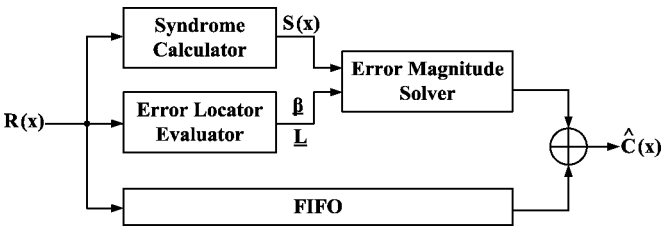

Fig. 5. Soft BCH decoding block diagram.

complexity advantages as long as sufficiently reliable soft information is provided.

III. Proposed Soft BCH Decoding Algorithm

Soft information can help the decoder to choose the least reliable bits which can be decoded by the soft decoder instead of the entire codeword for lower hardware complexity. Based on the concept of [16], Fig. 5 shows the proposed soft BCH decoder that includes three major steps: *syndrome calculator, error locator evaluator*, and *error magnitude solver* [18].

From the received polynomial $R(x)$, the syndrome polynomial $S(x) = S_1 + S_2x^1 + \cdots + S_{2t}x^{2t-1}$ is defined in (1). With the soft inputs, the decoder chooses $2t$ least reliable inputs and evaluates their corresponding error locators to form the error locator set $\underline{\mathcal{B}} = [\beta_{l_1}, \beta_{l_2}, \ldots, \beta_{l_{2t}}]^T$. Also, the error location set $\underline{L} = [l_1, l_2, \ldots, l_{2t}]^T$ can be calculated with $\underline{\mathcal{B}}$ because β_{l_i} is the error locator of the l_ith location and $\beta_{l_i} = \alpha^{l_i}$.

The relation between $\underline{\mathcal{B}}$ and the syndrome vector, $\underline{S} = [S_1, S_2, \ldots, S_{2t}]^T$, can be formulated as

$$\begin{bmatrix} \beta_{l_1} & \beta_{l_2} & \cdots & \beta_{l_{2t}} \\ \beta_{l_1}^2 & \beta_{l_2}^2 & \cdots & \beta_{l_{2t}}^2 \\ \vdots & \vdots & \cdots & \vdots \\ \beta_{l_1}^{2t} & \beta_{l_2}^{2t} & \cdots & \beta_{l_{2t}}^{2t} \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_{2t} \end{bmatrix} = \begin{bmatrix} \mathrm{S}_1 \\ \mathrm{S}_2 \\ \vdots \\ \mathrm{S}_{2t} \end{bmatrix} \tag{4}$$

where γ_i is the error magnitude in the l_ith location.

Notice that, in BCH codes, the valid error magnitude set $\underline{\Gamma} = [\gamma_1, \gamma_2, \ldots, \gamma_{2t}]^T$ must be a binary vector. If the l_ith location is the exact error location, γ_i is equal to 1; otherwise, γ_i is equal to 0. The $2t \times 2t$ matrix in (4) is defined as the *error locator matrix* $\mathbf{B}$. The estimated codeword polynomial $\hat{C}(x)$ can be obtained by XORing γ_i with R_{l_i}. In this decoding algorithm, $2t$ locations are collected in $\underline{L}$ such that at most $2t$ errors can be corrected.

To represent the difference between $\underline{S}$ and the multiplication result of $\mathbf{B}$ and $\underline{\Gamma}$, a discrepancy vector $\underline{\Delta} = [\delta_1, \delta_2, \ldots, \delta_{2t}]^T$ is defined as

$$\underline{\Delta} = \mathbf{B} \times \underline{\Gamma} + \underline{S}. \tag{5}$$

Notice that both the operations in (4) and (5) are under $GF(2^m)$. It is evident that, if all of the errors are occurred in the location set $\underline{L}$, the valid $\underline{\Gamma}$ can be determined to make $\underline{\Delta}$ be a zero vector; otherwise, this decoding approach calculates $\underline{\Gamma}$ as a nonbinary vector and fails to correct errors. For example, if there are three errors in the third, seventh, and ninth locations for a (255,239) BCH decoder which can correct two errors, $\underline{S}$ is expressed as

$$\underline{S} = \begin{bmatrix} \beta_3 + \beta_7 + \beta_9 \\ \beta_3^2 + \beta_7^2 + \beta_9^2 \\ \beta_3^3 + \beta_7^3 + \beta_9^3 \\ \beta_3^4 + \beta_7^4 + \beta_9^4 \end{bmatrix}.$$

In hard BCH decoding, these three errors are unable to be corrected. However, in the case that the decoder chooses the least reliable bits and forms the $\underline{\mathcal{B}}$ as $[\beta_3, \beta_4, \beta_7, \beta_9]$, $\underline{\Delta}$ becomes

$$\underline{\Delta} = \begin{bmatrix} \beta_3 & \beta_4 & \beta_7 & \beta_9 \\ \beta_3^2 & \beta_4^2 & \beta_7^2 & \beta_9^2 \\ \beta_3^3 & \beta_4^3 & \beta_7^3 & \beta_9^3 \\ \beta_3^4 & \beta_4^4 & \beta_7^4 & \beta_9^4 \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \end{bmatrix} + \begin{bmatrix} \beta_3 + \beta_7 + \beta_9 \\ \beta_3^2 + \beta_7^2 + \beta_9^2 \\ \beta_3^3 + \beta_7^3 + \beta_9^3 \\ \beta_3^4 + \beta_7^4 + \beta_9^4 \end{bmatrix}.$$

After the *Gauss Elimination* method, $\underline{\Gamma}$ is calculated as [1, 0, 1, 1] to make $\underline{\Delta}$ be a zero vector. According to $\underline{\mathcal{B}}$ and $\underline{\Gamma}$, the errors at the third, seventh, and ninth locations can be corrected.

In the proposed decoder, the syndrome calculator is used to calculate $\underline{S}$. The error locator evaluator classifies the soft input to choose $2t$ least reliability and creates $\underline{\mathcal{B}}$ and $\underline{L}$. The error magnitudes solver (EMS) is used to solve (4) to get $\underline{\Gamma}$. The Gauss Elimination method is the most intuitive way to solve (4), but the complexity is $O(n^3)$. Therefore, the EMS dominates the hardware complexity in the proposed decoder. In this paper, two alternative algorithms for improving decoding complexity are revealed. The heuristic error magnitudes solver (H-EMS) uses the characteristic that the valid error magnitude in BCH codes is either 0 or 1, and the Björck–Pereyra EMS (BP-EMS) employs the quick Vandermonde matrix solution. These two algorithms provide different architectures based on different error-correcting ability conditions. H-EMS is more suitable for smaller t applications while BP-EMS is adequate for larger t applications. The detailed algorithms are discussed in following subsections, and the hardware architectures are presented in Section IV.

TABLE II
PROPOSED H-EMS

Input : $\underline{\mathcal{B}}$, $\underline{S_{odd}}$ *and* $\underline{\Gamma} = 0$
1) *Construct the* $\mathbf{B_{odd}}$ *with* $\underline{\mathcal{B}}$
2) $\underline{\Delta_{odd}} = \mathbf{B_{odd}} \times \underline{\Gamma} + \underline{S_{odd}}$
3) *if* $\underline{\Delta_{odd}}$ *is a zero vector*
Successful Decoding !!!
else
if $\underline{\Gamma} == 2^{2t} - 1$
Failed Decoding !!!
else
$\underline{\Gamma} = \underline{\Gamma} + 1$
Go to 2)
Output : $\underline{\Gamma}$, *where* γ_i *is the error magnitude at* l_i − th *location*

A. H-EMS

In BCH codes, the valid error magnitude in $\underline{\Gamma}$ is a binary value. The problem can be formulated into checking all combinations of γ_i over $GF(2)$ instead of calculating exact error magnitudes. Thus, a $2t$-b counter is used to execute a heuristic search for all binary combinations. Notice that $S_1^2 = S_2, S_2^2 = S_4, \ldots, S_t^2 = S_{2t}$ in BCH codes, the computation of the even part syndromes $(S_2, S_4, \ldots, S_{2t})$ can be eliminated. The odd syndrome vector $\underline{S_{odd}} = [S_1, S_3, \ldots, S_{2t-1}]^T$ and the error locator matrix with half rows $\mathbf{B_{odd}}$ are applied to simplify (4) as

$$\begin{bmatrix} \beta_{l_1} & \beta_{l_2} & \cdots & \beta_{l_{2t}} \\ \beta_{l_1}^3 & \beta_{l_2}^3 & \cdots & \beta_{l_{2t}}^3 \\ \vdots & \vdots & \cdots & \vdots \\ \beta_{l_1}^{2t-1} & \beta_{l_2}^{2t-1} & \cdots & \beta_{l_{2t}}^{2t-1} \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_{2t-1} \\ \gamma_{2t} \end{bmatrix} = \begin{bmatrix} \mathrm{S}_1 \\ \mathrm{S}_3 \\ \vdots \\ \mathrm{S}_{2t-1} \end{bmatrix}. \tag{6}$$

The discrepancy vector for (6) is modified as

$$\begin{aligned} \underline{\Delta_{\mathrm{odd}}} &= [\delta_1, \delta_3, \ldots, \delta_{2t-1}]^T \\ &= \mathbf{B_{odd}} \times \underline{\Gamma} + \underline{S_{\mathrm{odd}}}. \end{aligned} \tag{7}$$

Notice that only t rows in the $\mathbf{B_{odd}}$ and $\underline{S_{\mathrm{odd}}}$ means only half computation in $\underline{\Delta_{\mathrm{odd}}}$ calculation as compared with $\underline{\Delta}$ calculation. The complexity of $\underline{\Delta_{\mathrm{odd}}}$ calculation is significantly reduced. The detail of the low-complexity implementation of H-EMS is illustrated in Table II. The heuristic search for all binary combinations is processed by iteratively counting $\underline{\Gamma}$ value, and the values of $\underline{\Delta_{odd}}$ are updated and verified whether or not $\underline{\Delta_{odd}}$ becomes a zero vector at each iteration. The decoding procedure successfully completes if certain $\underline{\Gamma}$ makes $\underline{\Delta_{\mathrm{odd}}}$ become a zero vector; otherwise, this decoding procedure fails to correct the errors.

TABLE III
PROPOSED BP-EMS

Input : $\underline{\mathcal{B}}$ *and* $\underline{S}$

1) for $(k = 1; k < 2t, k{+}{+})$
   for $(i = 2t; i > k, i\text{-}\text{-})$
   $S_i = S_i - \beta_{l_k} S_{i-1}$
2) for $(k = 2t - 1; k > 0, k\text{-}\text{-})$
   for $(i = k + 1; i \leq 2t, i{+}{+})$
   $S_i = S_i/(\beta_{l_i} - \beta_{l_{i-k}})$
   for $(i = k; i < 2t, i{+}{+})$
   $S_i = S_i - S_{i+1}$
3) for $(k = 1; k \leq 2t, k{+}{+})$
   $S_i = S_k/\beta_{l_k}$
4) *if* $\underline{S}$ *is a binary sequence*
   Successful Decoding !!!
   else
   Failed Decoding !!!

Output : $\underline{S}$, *where* S_i *is the error magnitude at* $l_i -$ th *location*

B. BP-EMS

The matrix with the terms of a geometric progression in each row or column is called a Vandermonde matrix, of which the order n takes the form

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ X_1 & X_2 & \cdots & X_n \\ X_1^2 & X_2^2 & \cdots & X_n^2 \\ \vdots & \vdots & \cdots & \vdots \\ X_1^{n-1} & X_2^{n-1} & \cdots & X_n^{n-1} \end{bmatrix}.$$

To solve (4), the inverse matrix of $\mathbf{B}$ is calculated, and it is very complicated to calculate the inverse matrix of an arbitrary matrix. However, the computation of (4) can be simplified because $\mathbf{B}$ is a Vandermonde matrix. The quick Vandermonde matrix computation is applied with the Björck–Pereyra algorithm [19], [20] to evaluate the error magnitude efficiently. Instead of using the whole $\mathbf{B}$ to compute (4), the geometric relation in each column makes the proposed BP-EMS only need $\underline{\mathcal{B}}$ for the complexity reduction. Table III shows the detail of the low-complexity implementation of BP-EMS.

In the proposed BP-EMS algorithm, the variable S_i, which initially represents the ith syndrome value, is updated iteratively. Each calculation of the syndrome represents a row operation in (4). After all computations, S_i indicates the l_ith error magnitude. Although the valid error magnitude in BCH codes is either 0 or 1, a nonbinary S_i may be generated because not all errors are in the $\underline{L}$. Therefore, a binary sequence check of $\underline{S}$ is applied in the final step to determine the decoding successful or not. The proposed BP-EMS is composed of division, multiplication, and addition operations. The regular operations make BP-EMS suitable for low complexity hardware implementation.

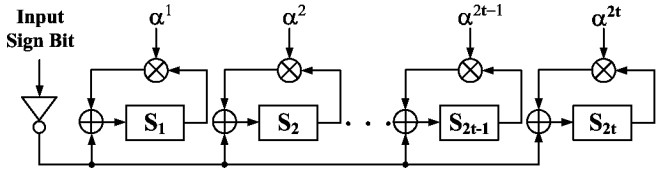

Fig. 6. Syndrome calculator architecture.

IV. VLSI ARCHITECTURE FOR THE PROPOSED SOFT BCH DECODERS

As mentioned in Section III, the proposed soft decoders include three major steps. The syndrome calculator and the error locator evaluator architectures are designed to deal with serial input. Following are the efficient architectures of H-EMS and BP-EMS. The architecture comparison between the hard BCH decoder and the proposed soft BCH decoders is discussed at the end of this section as well.

A. Syndrome Calculator

The syndrome calculator generates the syndrome polynomial $S(x)$ from the received polynomial $R(x)$. Fig. 6 shows totally $2t$ syndrome cells composed of constant multipliers, adders, and registers. Receiving data from R_{N-1} to R_0, the ith syndrome cell can be formulated as

$$\begin{aligned} S_i &= R(\alpha^i) \\ &= \left(\left((R_{N-1}\alpha^i + R_{N-2})\alpha^i + R_{N-3}\right)\alpha^i + \cdots\right)\alpha^i + R_0. \end{aligned}$$

In total, N cycles are required to receive and calculate all data for an (N, K, t) BCH code. At each cycle, the partial result is stored in the register as partial syndrome. The output of the register is multiplied with α^i and then accumulated with the received data. Once all data are received, the accumulated value is the ith syndrome. Notice that the input of the soft decoder is its reliability value, and R_{N-j} is the inverse of the jth input sign bit.

B. Error Locator Evaluator

The proposed error locator evaluator shown in Fig. 7 classifies the soft inputs for choosing $2t$ least reliable inputs according to the candidate reliabilities, $R_{l_1}, R_{l_2}, \ldots, R_{l_{2t}}$. Their corresponding error locators β_{l_i} and error locations l_i are also calculated and stored in registers while R_{l_i} is chosen. The proposed error locator evaluator architecture includes three major parts: the *reliability part*, the *error locator part*, and the *error location part*. The reliability part stores $2t$ candidate reliabilities $R_{l_1} \sim R_{l_{2t}}$ from the first stage to the $2t$th stage. The stored values will be compared with soft inputs to generate the select signals SEL_i to control the multiplexers. In the ith stage, if the input is smaller than $R_{l_{i-1}}$, the ith stage value is updated with the $(i-1)$th stage value. If the input is greater than $R_{l_{i-1}}$ and smaller than R_{l_i}, the ith stage value is updated with the input value. Otherwise, the ith stage value holds its current value.

The error locator part constructs $\underline{\mathcal{B}}$. The input data is received from R_{N-1} to R_0 serially, and the error locator of the l_ith location is α^{l_i}. Accordingly, the error locator can be computed by

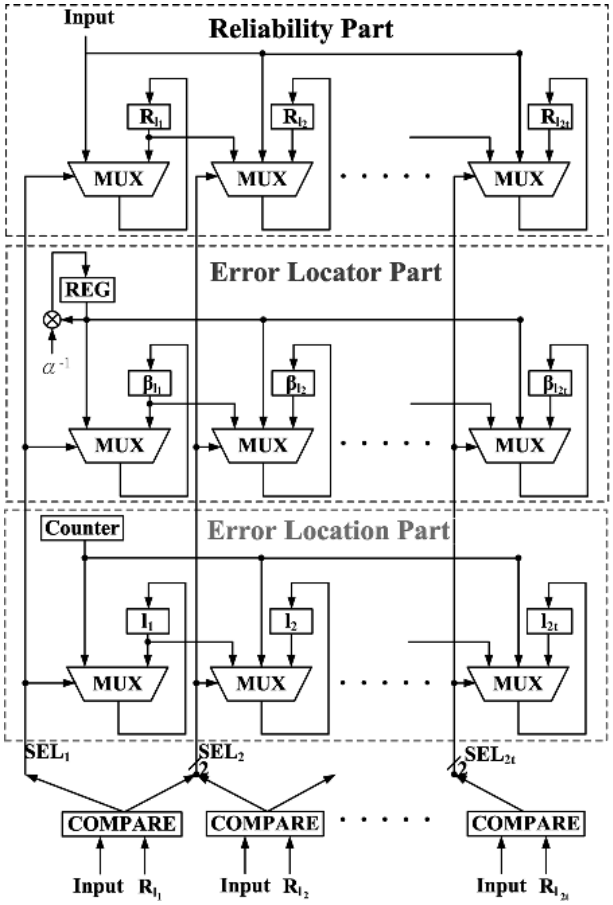

Fig. 7. Error locator evaluator architecture for serial input.

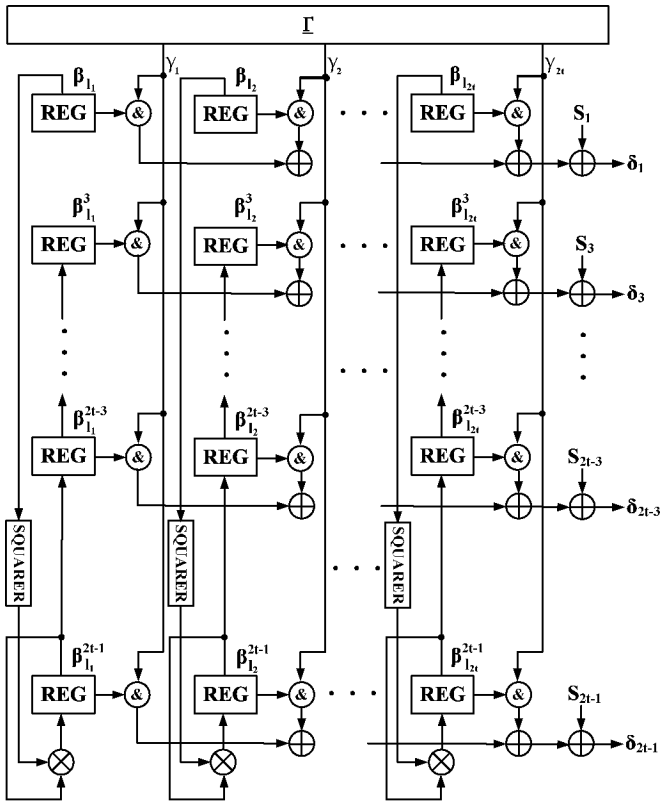

Fig. 8. H-EMS architecture.

multiplying α^{-1} with register REG, which stores the error locator of previous location. The error locator part uses a constant multiplier to calculate the error locator of each input. Notice that α^{N-1} is the initial value of register REG. The error location part uses a counter to compute the corresponding error location l_i of each R_{l_i} to form the error location set $\underline{L}$. The least reliable bits, instead of the entire codeword, are dealt with the proposed decoding method. After the EMS solves the error magnitude set $\underline{\Gamma}$, the actual error locations are obtained from $\underline{L}$ according to the corresponding error magnitudes. Hence, the Chien search procedure is no longer required and many redundant decoding latencies can be eliminated.

C. H-EMS

Based on Table II, Fig. 8 illustrates the proposed H-EMS architecture to evaluate $\underline{\Delta_{odd}} = \mathbf{B_{odd}} \times \underline{\Gamma} + \underline{S_{odd}}$ while given $\underline{S_{odd}}$ and $\underline{\mathcal{B}}$. There are $2t^2$ registers used to store all entries in $\mathbf{B_{odd}}$ matrix. The register values of the ith column forms a geometric progression with the common ratio $\beta_{l_i}^2$. In the ith column, the initial values of the top registers is set as β_{l_i} so that the output of the squarer will always be $\beta_{l_i}^2$. Also, iteratively multiplied by $\beta_{l_i}^2$, the bottom register with initial values β_{l_i} as well generates $\beta_{l_i}^{2j+1}$ for $\mathrm{j} = 0 \sim \mathrm{t} - 1$. Each register, except for the top register, iteratively updates upper register with the generated value. Hence, $\mathbf{B_{odd}}$ matrix is calculated with, in total, only $2t$ multipliers and squarers in $t - 1$ cycles and is stored in the registers. The registers will hold their values in matrix multiplication procedure.

Matrix multiplication is evaluated in the following 2^{2t} cycles. By counting $\underline{\Gamma}$ value, a heuristic search for all binary combinations can be completed. The $\underline{\Gamma}$ block in Fig. 8 is represented as a counter to generate a new $\underline{\Gamma}$ iteratively. At each iteration, each $\beta_{l_i}^j$ value will be calculated with γ_i, and the solver verifies whether $\underline{\Delta_{\mathrm{odd}}}$ becomes a zero vector or not. A successful decoding is completed when $\underline{\Gamma}$ satisfies the verification.

D. BP-EMS

The proposed BP-EMS has division, multiplication, and addition operations from Table III. To minimize the hardware complexity with only one divider, one multiplier and several adders simultaneously, the operations in Table III are coped with sequentially and take $4t^2 - 2t$, $2t^2 - t$, $2t$ cycles for steps 1)–3), respectively. Step 4), which is the binary sequence check of $\underline{S}$, is executed as soon as each S_i value has been fully updated. Notice that the multiplier can be shared if the divider can be decomposed into an inversion unit and a multiplier. Thus, as shown in Fig. 9, BP-EMS only contains one multiplier, one inversion unit, three adders, and a control logic. The control logic determines the computation order of S_i and β_{l_i}, and the computation results will be used to update each S_i value. The inversion unit in the proposed architecture is carried out in composite field because the finite field inversion over $GF(2^m)$ is very complex with lookup-table (LUT) implementation for large m.

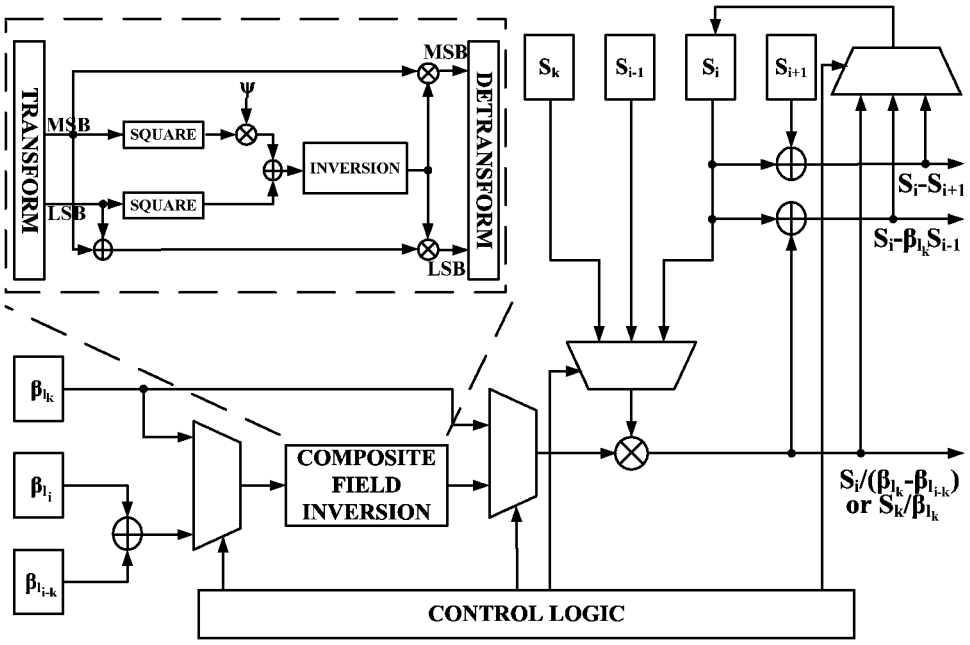

Fig. 9. BP-EMS architecture.

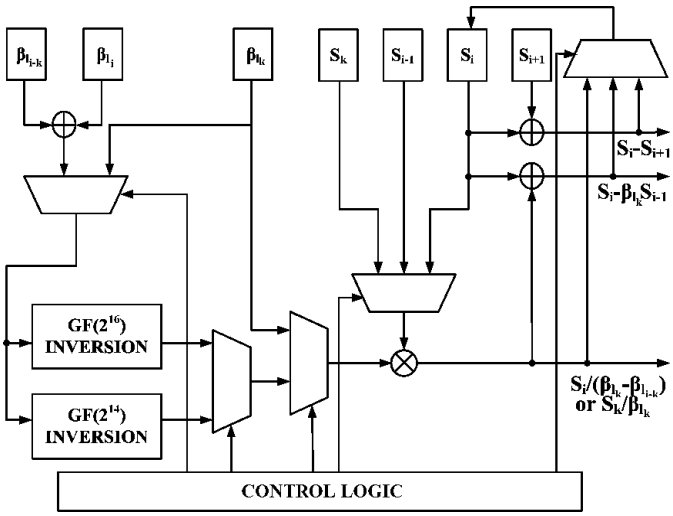

Fig. 10. Multimode BP-EMS architecture for DVB-S2 system.

TABLE IV
COMPARISON TABLE FOR AN (N, K, t) BCH CODE

	Hard BCH (iBM)	Soft BCH * (H-EMS)	Soft BCH ** (BP-EMS)
Register	$5t+2$	$2t^2+5t$	$8t$
Multiplier	$3t+3$	$2t$	1
Constant Multiplier	$3t$	$t+1$	$2t+1$
Squarer	0	$2t$	0
Inversion Unit	0	0	1
Latency	$2n+2t$	$n+2^{2t}+t-1$	$n+6t^2-t$

* In the special case: $t=1$, the number of multipliers and squarers is 0. If t is very small, like 1 or 2, we can check all combinations of γ_{ci} over $GF(2)$ at one cycle.

** Registers can be inserted into composite field inversion to reduce the critical path with the doubled latency in the EMS step.

The composite field [21] is viewed as an extension field of $GF(2^k)$ while given $m = kr$. The finite field $GF(2^m)$ can be constructed by coefficients from the subfield $GF(2^k)$. Operating in subfield leads to lower implementation complexity and better computation efficiency. For example, every element in $GF(2^{16})$ can be represented by $bx+c$, and inversion of $bx+c$ can be derived as

$$\frac{1}{bx+c} = (b^2\psi + bc + c^2)^{-1}(bx+b+c). \tag{8}$$

with the polynomial $x^2+x+\psi$ [21], where b and c are over $GF(2^8)$.

The composite field inversion units over $GF(2^{14})$ and $GF(2^{16})$ are only 1.1 K and 2.1 K gate-count in CMOS 90 nm technology, respectively, while the inversion units using the LUT method are about 41.5 K and 186 K gate-count, respectively.

Fig. 10 illustrates a multimode BP-EMS architecture to support 21 modes in the DVB-S2 system, including normal and short frames. Except for an additional inversion unit for $GF(2^{14})$, most components can be shared with a single mode design, leading to only a few control logics requirement. Two composite field inversion units are provided for supporting $GF(2^{14})$ and $GF(2^{16})$ operations. Also, the partial product and modular operation in the multiplier design over $GF(2^{14})$ and $GF(2^{16})$ are shared such that one reconfigurable multiplier can be used in both normal and short frames.

TABLE V
COMPARISON TABLE UNDER DIFFERENT CORRECT ABILITIES

	Hard BCH (iBM) t = 1	Soft BCH (H-EMS) t = 1	Soft BCH (BP-EMS) t = 1	Hard BCH (iBM) t = 2	Soft BCH (H-EMS) t = 2	Soft BCH (BP-EMS) t = 2	Hard BCH (iBM) t = 12	Soft BCH (H-EMS) t = 12	Soft BCH (BP-EMS) t = 12
Register	7	7	8	12	18	16	62	348	96
Multiplier	6	0	1	9	4	1	39	24	1
Constant Multiplier	3	2	3	6	3	5	36	13	25
Squarer	0	0	0	0	4	0	0	24	0
Inversion Unit	0	0	1	0	0	1	0	0	1
Normalized Complexity * (number of register)	76	13	52	120	73	66	560	663	206
Latency	2n+2	n+4	n+5	2n+4	n+17	n+22	2n+24	$n+2^{24}+11$	n+852

* According to the synthesis results in CMOS 90 nm technology, the complexity ratio over $GF(2^{16})$ among register, multiplier, constant multiplier, squarer, and inversion units is 1: 10: 3: 1.5: 25

E. Architecture Comparison

The architectures of a hard BCH decoder and two proposed soft BCH decoders are compared in Table IV. The two proposed soft BCH decoders are designed with H-EMS and BP-EMS respectively while hard BCH decoder is designed with inversionless Berlekamp-Massey (iBM) algorithm [22]. As compared with the soft BCH decoder with BP-EMS, only half syndromes are required for soft BCH decoder with H-EMS. In H-EMS, $2t$ multipliers and $2t$ squarers are used to construct the $\mathbf{B_{odd}}$. Notice that, if the error-correcting capability is equal to 1, the number of multipliers and squarers is 0 because only S_1 will be computed. The first row registers in the H-EMS can be shared with registers of the error locator part in error locator evaluator so that totally $2t^2-2t$ registers are used in this part. In BP-EMS, only one multiplier and one inversion unit are employed to evaluate $\underline{\Gamma}$. Both the soft decoders take n clock cycles for syndrome calculator and error locator evaluator simultaneously, and H-EMS and BP-EMS take $2^{2t}+t-1$ and $6t^2-t$ clock cycles, respectively.

Table V illustrates the number of each component at $t=1, 2$, and 12. Notice that the synthesis results in CMOS 90 nm technology shows that the complexity ratio over $GF(2^{16})$ among 16-b register, multiplier, constant multiplier, squarer, and inversion unit is 1: 10: 3: 1.5: 25. With this normalized ratio, Fig. 11 shows the normalized complexity analysis among hard and soft decoders. In large finite field operations, a multiplier is much more complicated than a register. Due to fewer multipliers, the proposed soft BCH decoders with more registers have much lower hardware complexity as compared with the hard BCH decoder with iBM algorithm under different error correcting capabilities t. Because of the nonlinear increment of the number of registers, the complexity of the soft BCH decoder with H-EMS is less than that of a hard BCH decoder only if t is smaller than 8, as shown in Fig. 11. H-EMS is more suitable to be applied in small t applications, like the (762,752) BCH decoder defined in the DMB-T system. In addition, the soft BCH decoder with BP-EMS can always provide lower complexity

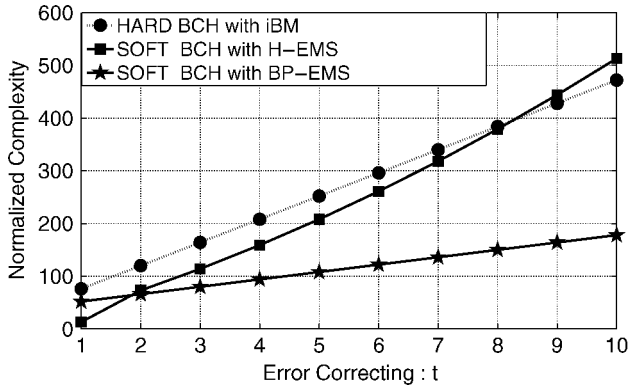

Fig. 11. Normalized hardware complexity analysis of BCH decoders over $GF(2^{16})$.

than a hard BCH decoder. In this paper, we applied BP-EMS in the proposed design according to the high error-correcting capability requirement. The proposed soft decoders, searching error locations at error locator evaluator procedure, lead to a lot of latency saving. Consequentially, the proposed soft decoders provide both higher throughput and much lower hardware complexity.

For further improvement on latency, H-EMS could complete all the computations in one cycle with less hardware overhead for small t. In addition, BP-EMS could insert registers into composite field inversion for operation frequency improvement. Although the latency in EMS step will be doubled, it is only few percentage of overall decoding procedure for long block length BCH decoders, resulting in that throughputs of the soft BCH decoder can be nearly doubled.

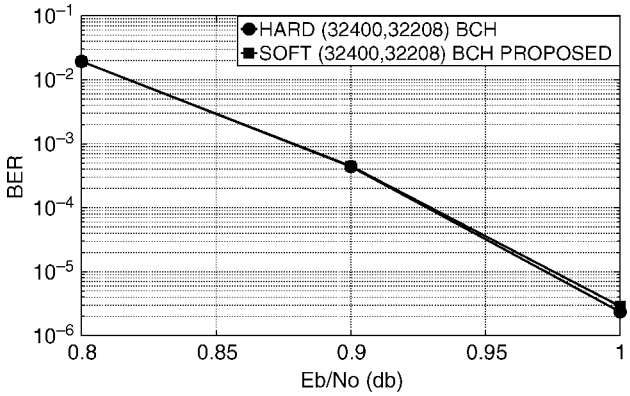

Fig. 12. Floating-point simulation results for (32400,32208) BCH in the DVB-S2 system.

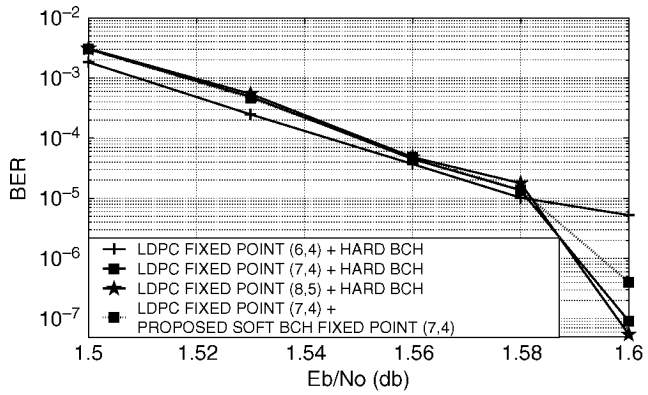

Fig. 13. Fixed-point simulation results for (32400,32208) BCH in the DVB-S2 system.

V. Simulation and Implementation Results in DVB-S2 Systems

In the DVB-S2 system, (32400,32208) BCH over $GF(2^{16})$ is defined to be concatenated with (64 800,32 400) LDPC code. Fig. 12 presents the BER performance at 50 LDPC decoding iterations under QPSK modulation and AWGN channel. The proposed soft BCH decoder has similar performance as the hard BCH decoder in DVB-S2 system at BER $= 10^{-5}$. Fig. 13 analyzes the required bit-width for the LDPC and BCH codes in the DVB-S2 system. The simulation parameters *fixed point (A,B)* represents A-bit input quantization with B-bit decimal fraction. The BER performance for the fixed-point LDPC code concatenated with the hard BCH code is demonstrated with the solid lines. The BER curves indicate that the fixed point (6,4) LDPC code has performance loss and the fixed point (7,4) LDPC code is sufficient to achieve similar performance to the fixed point (8,5) LDPC code. Since the more input quantization in LDPC code results in a linear increment for longer critical path delay and message storage, the fixed-point (7,4) LDPC code is adequate in our approach. In addition, the proposed soft BCH code with 7-b input quantization is also simulated in Fig. 13 with the dotted line while receiving the soft information from the fixed-point (7,4) LDPC code. It can achieve less than 0.01 dB performance loss in contrast to the hard BCH code at BER $= 10^{-6}$, indicating that 7-b input quantization is sufficient for our approach. Notice that the input quantization of the proposed soft BCH code only affects the size of the comparators and components of reliability part in error locator evaluator. Consequently, it has little influence on hardware complexity.

Fig. 14 shows the proposed soft (32400,32208) BCH decoder die photograph, which is implemented with the cell-based design flow and fabricated in 90 nm 1P9M CMOS process. The chip is verified by Agilent 93000 SOC test system, and the Shmoo plot shown in Fig. 15 indicates that our design can achieve 333 MHz operation frequency at 0.94 V supply. Table VI illustrates the chip summary as well as a hard BCH decoder for comparison. To minimize the storage requirement, all of the designs in Table VI are constructed with single-stage pipeline architecture. In the hard BCH decoder, the iBM algorithm is utilized to solve key equations as well as Chien search is applied to get error locations. By inserting registers into composite field inversion unit, the operation frequency of our proposed soft BCH decoder can be enhanced from 166 to 333 MHz with only 2.5% latency overhead. Moreover, our soft BCH decoder that computes error locations without Chien search has almost half latencies of the hard BCH decoder. The measurement results reveal that our proposed soft BCH decoder performs three times throughput with 50.0% gate-count saving as compared with the hard BCH decoder. The measured power consumption of the proposed soft BCH decoder is 8.43 mW with 1.0 V supply at 314.5 Mb/s, and the energy efficiency is 26.8 pJ/b while that of the hard BCH decoder is 63.6 pJ/b.

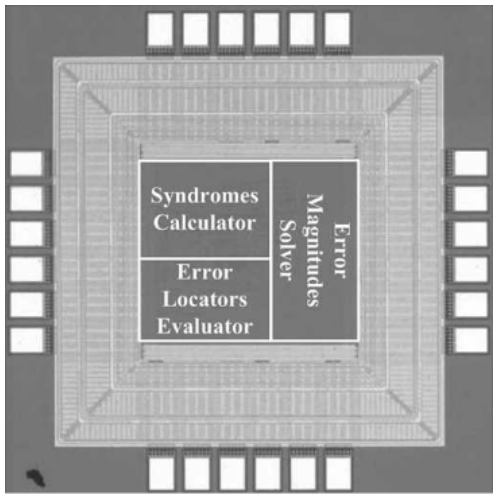

Fig. 14. Microphotograph of soft (32400,32208) BCH chip.

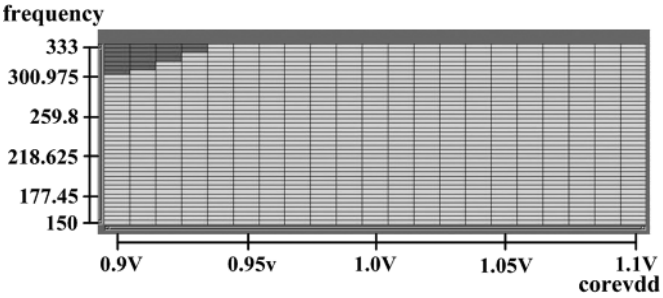

Fig. 15. Shmoo plot of soft (32400,32208) BCH chip.

TABLE VI
SUMMARY OF IMPLEMENTATION RESULTS

	Hard (32400,32208) BCH t = 12	Soft (32400,32208) BCH t = 12	Soft DVB-S2 BCH Normal Frame	Soft DVB-S2 BCH Normal + Short Frame
Technology	90 nm	90 nm	90 nm	90 nm
Architecture	iBM + Chien Search	BP-EMS w/o Chien Search	BP-EMS w/o Chien Search	BP-EMS w/o Chien Search
Pipeline Stage	1	1	1	1
Number of Mode	1	1	11	21
Operation Frequency	200 MHz (Post Layout)	333 MHz (Measurement)	300 MHz (Post Layout)	300 MHz (Post Layout)
Core Area	$190,497\mu m^2$	$102,400\mu m^2$	$105,625\mu m^2$	$119,025\mu m^2$
Gate Count	54.0 K	26.9 K	28.2 K	32.4 K
Normalized Gate Count	2.06	1	1.05	1.2
Maximum Throughput	99.3 Mb/s	314.5 Mb/s	295.5 Mb/s *	295.5 Mb/s *
Normalized Throughput	1	3.17	2.98	2.98
Maximum Latency	64,824	34,104	59,072 *	59,072 *
Power	6.32 mW @200 MHz	8.43 mW @333 MHz	9.45 mW @300 MHz	11.9 mW @300 MHz
Energy Efficiency	63.6 pJ/bit	26.8 pJ/bit	32 pJ/bit	40.2 pJ/bit

* The maximum latency and throughput are provided by (58320,58192) BCH code.

In addition, the proposed architectures are favorable to multi-mode designs because most components can be shared and only a few control logics are required. While extended to process DVB-S2 normal frame which consists of 11 modes, the proposed design can achieve 300 MHz operation frequency with only 5% gate-count increment as compared to the proposed soft (32400,32208) BCH decoder. To support those 21 modes defined in DVB-S2 system, including normal and short frames, our approach can achieve 300 MHz operation frequency with gate-count of 32.4 K, which is 20% more than the original single mode design. The post-layout simulation results of these two multi-modes design are also illustrated in Table VI.

VI. CONCLUSION

This paper presents a 26.9 K 314.5 Mb/s soft (32400,32208) BCH decoder chip for DVB-S2 system. The proposed decoder architecture not only deals with the least reliable bits to reduce complexity but also utilizes the single-stage pipeline to minimize the memory bank usage. The proposed error locator evaluator eliminates Chien search to ensure sufficient throughput without parallelism. As compared with the conventional hard BCH decoder, our BCH decoder with soft information from LDPC decoder provides similar system performance. From the measurement results, the proposed soft BCH decoder can achieve 314.5 Mb/s with 50.0% gate-count reduction in contrast to a 99.3 Mb/s traditional hard BCH decoder in CMOS 90 nm technology. While extended to fully support 21 modes in DVB-S2 system, the proposed design can operate at 300 MHz frequency with 32.4 K gate-count.

ACKNOWLEDGMENT

The authors would like to thank Dr. C.-C. Chung for layout assistance and Dr. C.-C. Lin and Y.-C. Liao for many fruitful discussions. The authors would also like to thank UMC for fabrication of the test chip and Chip Implementation Center for providing the CAD tools and test equipment.

REFERENCES

[1] C. R. Baugh and B. A. Wooley, *Theory and Practice of Error Control Codes*. Reading, MA: Addison-Wesley, 1983.
[2] Y. Chen and 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.
[3] *Framing Structure, Channel Coding and Modulation for Digital Television Terrestrial Broadcasting System*, NSPRC Std. GB 20 600-2006, 2007.
[4] *Digital Video Broadcasting (DVB); Frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system (DVB-T2)*, ETSI Std. EN 302 755 V1.2.0b, 2008.
[5] *Digital Video Bracasting (DVB) Second Generation System for Broadcasting, Interactive Services, News Gathering and Other Broadband Satellite Applications*, ETSI Std. En 302 307, 2005.
[6] *Digital Video Broadcasting (DVB): Framing Structure, Channel Coding and Modulation for 11/12 GHz Satellite Services*, ETSI Std. EN 300 421 v1.1.2, 1997.
[7] 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.
[8] E. R. Berlekamp, *Algrbraic Coding Theory*. New York: McGraw-Hill, 1968.
[9] H. Shao, T. Truong, L. Deutsch, J. Yuen, and I. Reed, "A VLSI design of a pipeline reed-solomon decoder," *IEEE Trans. Comput.*, vol. C-34, no. 5, pp. 393–403, May 1985.
[10] G. D. Forney, "Generalized minimum distance decoding," *IEEE Trans. Inf. Theory*, vol. IT-12, no. , pp. 125–131, Apr. 1966.
[11] D. Chase, "A class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inf. Theory*, vol. IT-18, no. 1, pp. 170–182, Jan. 1972.
[12] M. Lalam, K. Amis, D. Lerous, D. Feng, and J. Yuan, "An improved iterative decoding algorithm for block turbo codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2006, pp. 2403–2407.
[13] C. Hartmann and L. Rudolph, "An optimum symbol-by-symbol decoding rule for linear codes," *IEEE Trans. Inf. Theory*, vol. IT-22, no. 5, pp. 514–517, Sep. 1976.
[14] F. Therattil and A. Thangaraj, "A low-complexity soft-decision decoder for extended BCH and RS-like codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Sep. 2005, pp. 1320–1324.

[15] M. Baldi and F. Chiaraluce, "A simple scheme for belief propagation decoding of BCH and RS codes in multimedia transmissions," *Int. J. Digit. Multimedia Broadcasting*, vol. 2008, 2008, Art. ID 957846.
[16] W. J. ReidIII, L. L. Joiner, and J. J. Komo, "Soft decision decoding of BCH codes using error magnitudes," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 1997, p. 303.
[17] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 2, pp. 284–287, Mar. 1974.
[18] Y.-M. Lin, C.-L. Chen, H.-C. Chang, and C.-Y. Lee, "A 26.9 K 314.5 Mbps Soft (32400, 32208) BCH decoder chip for DVB-S2 system," in *Proc. IEEE Asia Solid State Circuits Conf. (ASSCC)*, Nov. 2009, pp. 373–376.
[19] A. Björck and V. Pereyra, "Solution of vandermonde systems of equations," *Math. Comput.*, vol. 24, pp. 893–903, Oct. 1970.
[20] J. Hong and M. Vetterli, "Simple algorithms for BCH decoding," *IEEE Trans. Commun.*, vol. 43, no. 8, pp. 2324–2333, Aug. 1995.
[21] C. Parr, "Efficient VLSI architectures for bit-parallel computation in Galois fields," Ph.D. dissertation, Inst. for Experimental Math., Univ. of Essen, Essen, Germany, 1994.
[22] I. S. Reed, M. T. Shih, and T. K. Truong, "VLSI design of inverse-free berlekamp-massey algorithm," *Proc. Inst. Elect. Eng*, vol. 138, pp. 295–298, Sep. 1991.

Yi-Min Lin received the B.S. degree in electrical engineering from the National Tsing Hua University, Hsinchu, Taiwan, in 2005, He is currently pursuing the Ph.D degree in the Institute of Electronics, National Chiao Tung University, Hsinchu, Taiwan.

His research interests include coding theory, VLSI architectures, and integrated circuit design for communications and signal processing.

Chih-Lung Chen received the B.E. and M.S. degrees in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2004 and 2006, respectively, where he is currently working toward the Ph.D. degree in electronics engineering.

His general research interests include VLSI implementation of error control codes and wireless communication systems.

Hsie-Chia Chang received the B.S., M.S., and Ph.D. degrees in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1995, 1997, and 2002, respectively.

From 2002 to 2003, he was with OSP/DE1 in MediaTek Corporation, working in the area of decoding architectures for Combo single chip. In February 2003, he joined the faculty of the Electronics Engineering Department, National Chiao Tung University, Hisnchu, Taiwan, where he became an Associate Professor in August 2007. His research interests include algorithms and VLSI architectures in signal processing, especially for error control codes and crypto-systems. Recently, he has also committed himself to designing high code-rate ECC schemes for flash memory and multi-Gb/s chip implementations for wireless communications.

Chen-Yi Lee (M'01) received the B.S. degree from National Chiao Tung University, Hsinchu, Taiwan, in 1982, and the M.S. and Ph.D. degrees from Katholieke University Leuven (KUL), Leuven, Belgium, in 1986 and 1990, respectively, all in electrical engineering.

From 1986 to 1990, he was with IMEC/VSDM, working in the area of architecture synthesis for DSP. In February 1991, he joined the faculty of the Electronics Engineering Department, National Chiao Tung University, Hsinchu, Taiwan, where he is currently a Professor and Dean of the Research and Development Office. His research interests mainly include VLSI algorithms and architectures for high-throughput DSP applications. He is also active in various aspects of high-speed networking, system-on-chip design technology, very low power designs, and multimedia signal processing. In these areas, he has authored or coauthored more than 180 papers and holds decades of patents. He served as the Director of Chip Implementation Center (CIC), an organization for IC design promotion in Taiwan (2000/8 2003/12), and the microelectronics program coordinator of the Engineering Division under the National Science Council of Taiwan (2003/1 2005/12).

Dr. Lee was the former IEEE Circuits and Systems Society Taipei Chapter Chair.