IEEE Asian Solid-State Circuits Conference
November 16-18, 2009 / Taipei, Taiwan

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

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

Department of Electronics Engineering National Chiao Tung University
Hsinchu, Taiwan 300, R.O.C.
E-mail: ymlin@si2lab.org

***Abstract*—This paper provides a soft BCH decoder using error magnitudes to deal with least reliable bits. With soft information from the previous decoder defined in digital video broadcasting (DVB), the proposed soft BCH decoder provides much lower complexity and latency than the traditional hard BCH decoder while still maintaining performance. The proposed error locator evaluator architecture evaluates error locations without Chien search, leading to high throughput. Börck-Pereyra error magnitudes solvers (BP-EMS) is presented to improve decoding efficiency and hardware complexity. The experimental result reveals that our proposed soft (32400, 32208) BCH decoder defined in DVB-S2 system can save 50.0% gate-count and achieve 314.5Mbps in standard CMOS 90nm technology.**

***Index Terms*—Error correction coding, Bose-Chaudhuri-Hochquenghem (BCH) codes, Digital Video Broadcasting.**

I. Introduction

The Bose-Chaudhuri-Hocquenghen (BCH) [1] codes are popular in storage and communication systems recently. From DMB-T [2] and DVB-S2 [3] applications shown in Fig. 1, the BCH codes with long block length are specified to suppress the error floor due to iterative LDPC decoding. Since BCH codes perform as outer codes in those communication systems, the soft information from the inner decoder can be employed to further improve the error-correcting performance.

Soft decision decoding of BCH codes has aroused many research interests. Forney developed the generalized-minimum-distance (GMD) [4], which uses algebraic algorithms to generate a list of candidate codewords and chooses a most likely codeword from the list. Other algorithms with the same concept of candidate list, such as Chase [5] and SEW [6], are also widely used in many applications. This paper illustrates a soft BCH decoding method using error magnitudes [7] to deal with the least reliable bits. For example, Fig. 2 shows the results of a concatenated code with 16-state BCJR [8] and (255,239) BCH over $GF(2^8)$. Based on the soft information from previous decoder, the performance gain of the BCH decoder is about 0.73 db at BER = 10^{-6} when 2t+1 candidate bits within a codeword are chosen to correct errors.

Conventional BCH decoding contains *syndromes calculation*, *key equation solver*, and *Chien search* [1]. In general, the complexity of a soft BCH decoder, such as [9], is much higher than a hard BCH decoder for decoding an entire codeword. Nevertheless, soft BCH decoders with lower complexity can

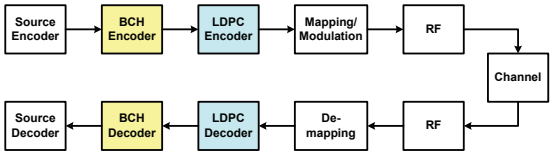

Fig. 1. Block diagram of DMB-T and DVB-S2 systems

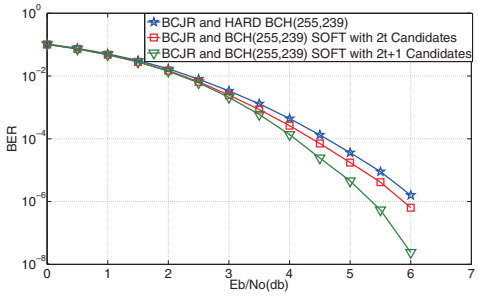

Fig. 2. Simulation results for (255,239) BCH concatenating with a 16-state BCJR under BPSK modulation and AWGN channel.

be revealed by focusing on the least reliable bits instead of the whole codeword. For long block length BCH decoders, the decoding latency is dominated by the syndromes calculation and Chien search. Unlike conventional algorithms using parallelism Chien search [10] [11], we proposed a error locations aimed architecture to eliminate Chien search for high throughput in this paper.

This paper is organized as follows. Section II describes the soft BCH decoding method using error magnitudes. The proposed algorithms and architectures are presented in section III. Based on the proposed method, Section IV demonstrates the simulation results and the implementation of the soft BCH decoder. Section V concludes the paper.

II. Soft Decision BCH Decoding

As shown in Fig. 3, the soft BCH decoding using error magnitudes [7] includes three major steps: *syndromes calculation, error locators evaluator*, and *error magnitudes solver*. From the received polynomial R(x), the syndromes polynomial

978-1-4244-4434-2/09/$25.00 ©2009 IEEE

$S(x) = S_1 + S_2x^1 + \cdots + S_{2t}x^{2t-1}$ are expressed as

$$S_j = R(\alpha^j) = \sum_{i=1}^{v}(\alpha^j)^{l_i} = \sum_{i=1}^{v}(\beta_{l_i})^j \quad (1)$$

for j = $1, 2, \cdots, 2$t, where α is the primitive element over $GF(2^m)$. Notice that l_i is the i-th actual error location and $\beta_{l_i} = \alpha^{l_i}$ indicates the corresponding error locator.

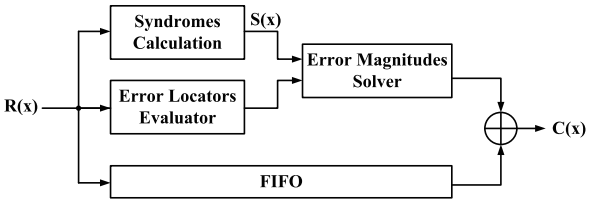

Fig. 3. Soft Decision BCH Decoding Block Diagram

With soft inputs, error locators evaluator can choose 2t least reliable inputs and evaluates their corresponding error locator values to form a β-vector, $[\beta_{c_1}, \beta_{c_2}, \ldots, \beta_{c_{2t}}]$. Also, the error location set, $\{L_{c_1}, L_{c_2}, \ldots, L_{c_{2t}}\}$, can be calculated with β-vector because the β value of the L_{c_i}-th location is $\beta_{c_i} = \alpha^{L_{c_i}}$. The relation between β_{c_i} and the syndromes can be formulated as

$$\begin{bmatrix} \beta_{c_1} & \beta_{c_2} & \cdots & \beta_{c_{2t}} \\ \beta_{c_1}^2 & \beta_{c_2}^2 & \cdots & \beta_{c_{2t}}^2 \\ \vdots & \vdots & \cdots & \vdots \\ \beta_{c_1}^{2t} & \beta_{c_2}^{2t} & \cdots & \beta_{c_{2t}}^{2t} \end{bmatrix} \begin{bmatrix} \gamma_{c_1} \\ \gamma_{c_2} \\ \vdots \\ \gamma_{c_{2t}} \end{bmatrix} = \begin{bmatrix} \mathrm{S}_1 \\ \mathrm{S}_2 \\ \vdots \\ \mathrm{S}_{2t} \end{bmatrix} \quad (2)$$

where γ_{c_i} is the error magnitude corresponding to β_{c_i} for $i = 1, 2, \ldots, 2t$. The left 2t $\times$ 2t matrix in (2) is defined as β-*matrix*.

From (1) and (2), it is evident that if all the errors are in the error location set, the exact γ_{c_i} value can be determinated; otherwise, this decoding approach fails to correct errors. The error magnitudes solver shown in Fig. 3 is used to solve (2) to get γ_{c_i}. For those γ_{c_i} equal to 1, the corresponding L_{c_i} are the exact error locations. The codeword polynomial C(x) can be obtained by inversing the L_{c_i}-th values in the received polynomial R(x).

III. Proposed Algorithm and Architecture

A. Error Locators Evaluator

As shown in Fig. 4, error locators evaluator architecture includes the *reliability part*, the *error locator part* and the *error location part*. The upper part is the reliability part which stores the reliabilities of 2t least reliable candidates $R_{c_1}, R_{c_2}, \ldots, R_{c_{2t}}$. The medium part is the error locator part to construct the β-vector. Because the β value of the i-th location is α^i, the β value of (i+1)-th locations is α times the β values of i-th location. The β value can be computed by multiplying α^{-1} with register REG if the input is serial in from the highest degree coefficient of R(x). Thus, the error locator part can use a constant multiplier to calculate the error locator of each input. Notice that register REG initially contains the β value of the first input. The bottom part is the error location part. The decoding method focuses on the least reliable bits instead of the whole codeword, so the error location part uses a counter to compute the error location L_{c_i} corresponding to each R_{c_i} for serial input. Hence, the Chien search procedure is no longer required and a lot of redundant decoding latencies can be eliminated.

Error locators evaluator classifies the soft inputs to choose 2t least reliable inputs as the candidate reliabilities $R_{c_1}, R_{c_2}, \ldots, R_{c_{2t}}$. Their corresponding error locators β_{c_i} and error locations L_{c_i} are also calculated and stored in registers. Error locators evaluator compares the soft inputs with R_{c_i}, and then generates the select signals SEL_i to control the multiplexers. In the i-th stage, if the input is smaller than $R_{c_{i-1}}$, the i-th stage is updated with (i-1)-th stage value. If the input is greater than $R_{c_{i-1}}$ and smaller than R_{c_i}, the i-th stage is updated with the input value. Otherwise, the i-th stage holds its current value.

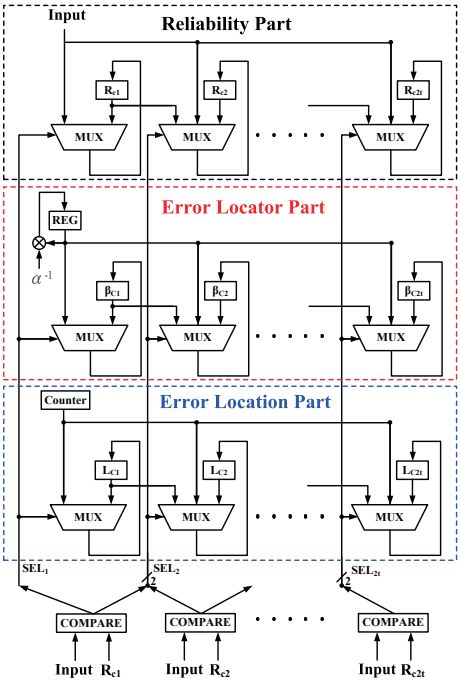

Fig. 4. Error Locators Evaluator Architecture for Serial Input

B. Börck-Pereyra Error Magnitudes Solver (BP-EMS)

To obtain the valid γ_{c_i} value in (2), the *Gauss Elimination* method is the most intuitive way but the complexity is $O(n^3)$. For small t, like 1 or 2, we can check all combinations of γ_{ci} over $GF(2)$ instead of calculating real error magnitudes because the valid error magnitude in (2) is either 0 or 1 in BCH codes. For larg t, since the β_{c_i} matrix is a Vandermonde matrix, Börck-Pereyra algorithm [12] [13] shown in TABLE I can calculate the error magnitudes efficiently for large matrix. In Börck-Pereyra algorithm, the variable S_i which initially contains the i-th syndrome value is updated iteratively. Instead

TABLE I
BÖRCK-PEREYRA ALGORITHM

$Input$: $syndromes\ vector\ \ [S_1, S_2, \ldots, S_{2t}]$
$\beta_{c_i}\ vector\ [\beta_{c_1}, \beta_{c_2}, \ldots, \beta_{c_{2t}}]$
for $(k = 1; k < 2t, k{+}{+})$
for $(i = 2t; i > k, i\text{-}\ \text{-})$
$S_i = S_i - \beta_{c_k} S_{i-1}$
for $(k = 2t - 1; k > 0, k\text{-}\ \text{-})$
for $(i = k + 1; i \leq 2t, i{+}{+})$
$S_i = S_i / (\beta_{c_i} - \beta_{c_{i-k}})$
for $(i = k; i < 2t, i{+}{+})$
$S_i = S_i - S_{i+1}$
for $(k = 1; k \leq 2t, k{+}{+})$
$S_i = S_k / \beta_{c_k}$
$Output : S_i\ is\ error\ magnitude$

of using β-matrix to compute (2), Börck-Pereyra algorithm uses β-vector to reduce the implementation complexity. After all computations, S_i indicates the i-th error magnitude.

From TABLE I, Börck-Pereyra algorithm has division, multiplication and addition operations. Notice that the multiplier can be shared if the divider can be decomposed into an inversion and a multiplier. Thus, as shown in Fig. 5, BP-EMS only contains 1 multiplier, 1 inversion, 3 adders and a control logic. The control logic determines the computation order of the syndromes and β_{c_i}, and the computation results will be used to update each S_i value. The inversion in the proposed architecture is carried out in composite field because the finite field inversion over $GF(2^m)$ is costly and infeasible with table-lookup implementation for large m.

Composite field [14] 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 (3) with the polynomial $x^2 + x + \psi$ [14], where b and c are over $GF(2^8)$.

$$\frac{1}{bx+c} = (b^2\psi + bc + c^2)^{-1}(bx + b + c) \qquad (3)$$

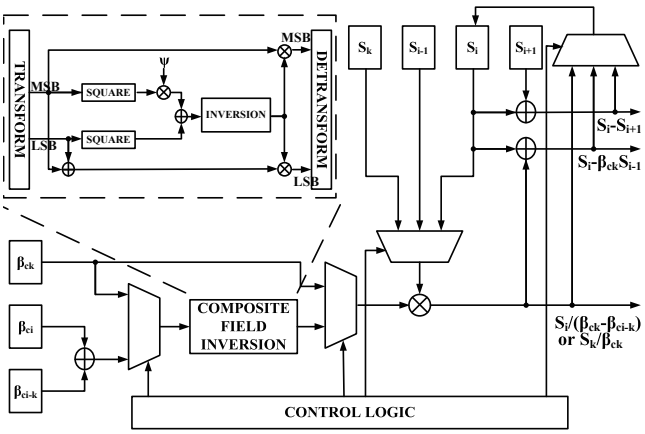

Fig. 5. Börck-Pereyra Error Magnitudes Solver Architecture

The composite field inversion over $GF(2^{16})$ is only 2.1K gate count in CMOS 90nm technology while the inversion using Look Up Table method is about 186K gate count.

C. Architecture Comparison

The architectures of a hard BCH decoder and the soft BCH decoder are compared in TABLE II. The soft BCH decoder is constructed with BP-EMS while the hard BCH decoder is constructed with inversionless Berlekamp-Massey (iBM) algorithm [15].

The soft BCH decoder has less multipliers but more registers than the hard BCH decoder. However, it can have lower hardware complexity since multipliers is much larger than registers. Also, searching error locations at error locators evaluator procedure leads to a lot of latency saving. For further improvement on latency, BP-EMS could insert registers in composite field inversion for improvement on operation frequency. Notice that latency in EMS step will be double but it is only few percentage of overall decoding procedure for long block length BCH decoders. Thus the throughputs of the soft BCH decoder can be approximated to doubled throughput.

TABLE II
COMPARISON TABLE FOR A (n, k, t) BCH CODE

	Hard BCH with iBM	Soft BCH with BP-EMS [1][2]
Register	$5t + 2$	$8t$
multiplier	$3t + 3$	1
constant multiplier	$2t + 1$	$2t + 1$
inversion	0	1
latency	$2n + 2t$	$n + 6t^2 - t$

[1] If t is very small, like 1 or 2, we can check all combinations of γ_{ci} over $GF(2)$ instead of calculating real error magnitudes.
[2] Registers can be inserted in composite field inversion to decrease the critical path with the doubled latency in EMS step.

IV. SIMULATION AND IMPLEMENTATION RESULTS IN DVB-S2 SYSTEMS

In DVB-S2 system, (32400, 32208) BCH over $GF(2^{16})$ is defined to be concatenated with LDPC codes. Fig. 6 shows the simulation results for DVB-S2 system with (64800, 32400) LDPC at 50 iterations.

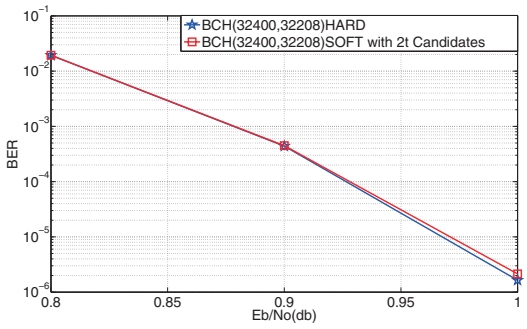

Fig. 6. Simulation results for (32400, 32208,12) BCH in DVB-S2 system under QPSK modulation and AWGN channel

The proposed soft BCH decoders have similar performance in DVB-S2 at BER = 10^{-5}. The BCH decoder is implemented

with hard decision and soft decision methods and demonstrated in TABLE III.

TABLE III
SUMMARY OF IMPLEMENTATION RESULTS

	Hard (32400, 32208) BCH t = 12	Soft (32400, 32208) BCH t = 12
Technology	90nm	90nm
Architecture	iBM + Chien Search	BP-EMS
Operation Frequency	200MHz (Post Layout)	333MHz (Measurement)
Core Area	$190497\mu m^2$	$102400\mu m^2$
Gate Count	54.0K	26.9K
Latency	64824	34104
Throughput	99.3Mbps	314.5Mbps

The hard BCH decoder uses iBM algorithm to solve key equation and needs Chien search to get error locations. By inserting registers in composite field inversion, the operation frequency of the soft BCH decoder with BP-EMS is enhanced from 166MHz to 333MHz with only 2.5% latency increment in overall decoding procedure. Computing error locations without Chien search, the soft BCH decoder has almost half latencies of the hard BCH decoder. Hence, the soft BCH decoder has much better throughputs than the hard BCH decoder. The measurement result reveals that the soft BCH decoder saves 50.0% gate-count and 47.4% clock cycle latency as compared with the hard BCH decoder. Fig. 7 is the chip microphoto of the proposed soft (32400, 32208) BCH decoder.

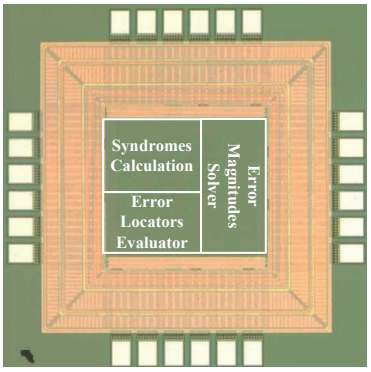

Fig. 7. Microphoto of Soft (32400, 32208) BCH Chip

V. CONCLUSION

This paper provides a 26.9K 314.5Mbps soft (32400, 32208) BCH decoder chip for DVB-S2 System. From the simulation, the proposed soft decoder performs much lower complexity with similar system performance as compared with the conventional hard decoder. In addition, Chien search step can be eliminated due to evaluate the error locations in the proposed error locators evaluator procedure. Thus, a lot of redundant decoding latencies can be eliminated and higher throughputs can be achieved without parallelism. The proposed soft BCH decoder can achieve 314.5Mbps while saving 50.0% gate-count as compared with the a 99.3Mbps traditional hard BCH decoder in CMOS 90nm technology.

ACKNOWLEDGEMENT

This work was supported by the NSC, Taiwan, R.O.C., under grant NSC 97-2221-E-009-166. The authors would like to thank Dr. Chien-Ching Lin and Dr. Yen-Chin Liao for the assistance. The authors also thank UMC for manufacturing the test chip and Chip Implementation Center for providing the CAD tools and measurement equipment.

REFERENCES

[1] C. R. Baugh and B. A. Wooley, *Theory and Practice of Error Control Codes*. Addison-Wesley, 1983.
[2] *Framing Structure, Channel Coding and Modulation for Digital Television Terrestrial Broadcasting System*, NSPRC Std. GB 20 600-2006, 2007.
[3] *Digital Video Bracasting (DVB) Second Generation System for Broadcasting, Interactive Services, News Gathering and Other Broadband Satellite Applications*, ETSI Std. En 302 307, 2005.
[4] G. D. Forney, "Generalized Minimum Distance Decoding," *IEEE Trans. Inform. Theory*, vol. 12, p. 125V131, Apr. 1966.
[5] D. Chase, "A Class of Algorithms for Decoding Block Codes with Channel Measurement Information," *IEEE Trans. Inform. Theory*, vol. IT-18, p. 170V182, Jan. 1972.
[6] M. Lalam, K. .Amis, D. Lerous, D. Feng, and J. Yuan, "An Improved Iterative Decoding Algorithm for Block Turbo Codes," *IEEE Int. Symp. on Info. Theory*, pp. 2403–2407, July 2006.
[7] W. J. ReidIII, L. L. Joiner, and J. J. Komo, "Soft Decision Decoding of BCH Codes Using Error Magnitudes," *IEEE Int. Symp. on Info. Theory*, p. 303, June 1997.
[8] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Trans. Inform. Theory,*, vol. 20, pp. 284–287, Mar. 1974.
[9] M. Baldi and F. Chiaraluce, "A Simple Scheme for Belief Propagation Decoding of BCH and RS Codes in Multimedia Transmissions," *International Journal of Digital Multimedia Broadcasting*, vol. 2008, 2008.
[10] Y. Chen and K. Parhi, "Small Area Parallel Chien Search Architectures for Long BCH Codes," *IEEE Trans. on VLSI*, vol. 12, no. 5, pp. 545–549, May 2004.
[11] J. Cho and W. Sung, "Strength-Reduced Parallel Chien Search Architecture for Strong BCH Codes," *IEEE Trans. on Circuits and Systems II*, vol. 55, no. 5, pp. 427–431, May 2008.
[12] A. Björck and V. Pereyra, "Solution of Vandermonde Systems of Equations," *Math. Computation*, vol. 24, pp. 893–903, Oct. 1970.
[13] J. Hong and M. Vetterli, "Simple Algorithms for BCH Decoding," *IEEE Trans. on Communication*, vol. 43, pp. 2324–2333, Aug. 1995.
[14] C. Parr, "Efficient VLSI Architectures for Bit-Parallel Computation in Galois Fields," Ph.D. dissertation, Inst. for Experimental Mathematics of Univ. of Essen Germany, 1994.
[15] 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, Sept. 1991.