Early-Stopped Technique for BCH Decoding Algorithm Under Tolerant Fault Probability

Shih-Shuan Wang[1], Hong-fu Chou[2], Xinchao Zhong[3], and Sean Longyu Ma[3]

[1] Transilvania University of Brasov, Romania
[2] Interdisciplinary Centre for Security, Reliability, and Trust, University of Luxembourg
[3] School of Computer Science, The University of Auckland, New Zealand
lma792@aucklanduni.ac.nz

Abstract. In this paper, a technique for the Berlekamp-Massey(BM) algorithm is provided to reduce the latency of decoding and save decoding power by early termination or early-stopped checking. We investigate the consecutive zero discrepancies during the decoding iteration and decide to early stop the decoding process. This technique is subject to decoding failure in exchange for the decoding latency. We analyze our proposed technique by considering the weight distribution of BCH code and estimating the bounds of undetected error probability as the event of erroneous stop checking. The proposed method is effective in numerical results and the probability of decoding failure is lower than 10^{-119} for decoding 16383 code length of BCH codes. Furthermore, the complexity compared the conventional early termination method with the proposed approach for decoding the long BCH code. The proposed approach reduces the complexity of the conventional approach by up to 80%. As a result, the FPGA testing on a USB device validates the reliability of the proposed method.

Keywords: BCH code, BCH decoding, Berlekamp-Massey algorithm, low latency design, early stop, early termination.

1 Introduction

Flash memory [1] performs as the main non-volatile storage device, and the flash interface unit is applied for system-on-chip (SoC) products. The market size of NAND flash memories is still growing and is projected to see a compound annual growth rate of 6.39% [2]. Flash memory provides a low-power solution for storage systems [3] and, small size and the light form factor are the essential properties for this type of storage. In the SoC applications, all of the boot information is generally stored in flash memory. The flash memory includes a number of partitions for the boot loader code and the flash file system is created in the flash memory [4]. The DMA interacts with the error control coding (ECC) block [5], which provides two main purposes. The first is to generate the ECC bytes and program in the spare area, and the second is to correct the data in the data buffer.

Consequently, the ECC engine is a critical issue regarding system performance. The chip area is dominated by the ECC decoder, comprising a high percentage of the flash controller [6].

The Bose-Chaudhuri-Hocquenghem (BCH) code has become the ultimate solution for the ECC engine in recent years. In coding theory, the BCH codes form a class of cyclic error-correcting codes that are constructed using finite fields. The decoding algorithm is based on a feasible implementation where the Berlekamp-Massey (BM) algorithm [8] has been widely selected in typical examples. The complexity of the decoding is competitive with respect to the BM properties of the linear feedback shift register. However, system latency suffers from larger t error correction capability which requires $2t$ iterations of conventional BM decoding and common applications require high error-correcting capability. The long decoding time has become a bottleneck in the system performance while using BM decoding. The error distribution for flash memory shows that few errors at the beginning of its usage and the low number of errors dominate the majority of the probability that will occur within a code block. In order to overcome this degradation, early termination of BM decoding is necessary to improve the system performance for high-speed applications. In [9], the authors adopt a restricted Gaussian elimination on the Hankel structured augmented syndrome matrix to reinterpret an early-stopped version of the Berlekamp-Massey algorithm. This approach has proven the minimal iterations $t + e$ of the Berlekamp-Massey algorithm where e is the number of error bits. Following the thread of [10], the author presents a feasible approach for early termination but the investigation of malfunction probability was present in [11].

In this paper, the probability of decoding failure is considered in exchange for early-stopped BM decoding feasibility. The proposed technique terminates conventional BM decoding after less than $t + e$ iterations so as to reduce redundant latency. However, the proposed technique is subject to the decoding failure problem. The probability that a detection error will occur must be evaluated to ensure the reliability of the proposed approach. Consequently, we propose an early-stopped technique for BM decoding by observing certain conditions while performing decoding iterations. In Section II, we present the early-stopped checking procedure of BM decoding by observing consecutive zero discrepancies. Since zero discrepancies provide the information of detectable decoding, it is an interesting problem to estimate the undetectable decoding after consecutive zero discrepancies. We provide an estimation of the erroneous early-stopped checking by means of the probability of undetected error probability in [7]. After combining the early-stopped checking criterion in [10], we propose our approach. In Section III, the complexity analysis is presented to compare with the conventional early-stopped BM approach. In Section IV, the numerical results are presented to evaluate the feasibility of a practical application. Conclusions are presented in Section V.

2 Early stopped approach based on the view of discrepancy for the BM algorithm

In coding theory, BCH codes [12] [13] are constructed using polynomials over a finite field (also called the Galois field and is denoted as GF(q)). One of the key features of BCH codes is that, during code design, there is precise control over the number of symbol errors that are correctable by the code. In particular, it is possible to design binary BCH codes that can correct multiple-bit errors in discrete distribution under a correction capability of t bits. Another advantage of BCH codes is the ease with which they can be decoded, namely, via an algebraic method known as syndrome decoding. This simplifies the design of the decoder for these codes, using small low-power electronic hardware.

BCH codes are used in applications such as satellite communications, compact disc players, DVDs, disk drives, solid-state drives, etc.

There are many algorithms for decoding BCH codes. The most common follow this general outline:

1. Calculate the syndromes for the received vector
2. Determine the number of errors v and the error locator polynomial $N(x)$ from the syndromes
3. Calculate the roots of the error location polynomial to determine the error locations X_i
4. Calculate the error values at those error locations
5. Correct the errors

The decoding algorithm may determine that the received vector contains too many errors and cannot be corrected. For example, if the number of errors is greater than the correction capability, then the correction would fail. In a truncated (not primitive) code, an error location may be out of range. If the received vector has more errors than the code can correct, the decoder may unknowingly produce an apparently valid message that is not the one that was sent.

In order to determine any possible solutions to shorten the BM decoding process, based on the result in [10] and [9], we classify the solutions in two conditions as follows.

Condition 1**:**
For the u-th iteration of the BM algorithm, the discrepancy at iteration u is presented as d_u, and any discrepancies in the next t-l_u-1 steps of the iteration are zero.

Condition 2**:**
If the number of errors in the received polynomials is v, only $t+v$ steps of the iteration are needed in order to determine the error-location polynomials.

2.1 Heuristics for consecutive zero discrepancies

Following the thread of *Condition* 2, the probability of the erroneous event based on the view of the discrepancy is investigated as follows. The discrepancies in

certain iterations equal to zero, as shown in *Condition* 1 represent the detection capability reach in a certain level of l_u iterations, i.e. $l_u = v$, where v is the number of error bits hypothesized by our proposed approach.

Heuristic 1: Let a BCH code ζ have minimum Hamming distance $d \geq 2t+1$ and consider that $\zeta^{v+\kappa} \subset \zeta$ denotes a BCH code subset with minimum Hamming distance $d_s \geq v + \kappa$ and κ is the number of consecutive zero discrepancies for the v-th iteration of BM algorithm. The next κ steps actually occurred with $v + \kappa \leq 2t$. *Rationale*: The Hamming distance for the received codeword r and the transmitted codeword c is presented as $d(r, c) = i$, $i < t$, where $c \in \zeta^{v+\kappa}$.

Heuristic 2: The error pattern ξ defects the codeword c, it can also be presented as $r = c + \xi$ and $d(r, c) = d(\xi, c)$. *Rationale*: Assume $e = v$ and e denotes the exact number of error bits caused by the channel without the decoding fault. Otherwise, a malfunction occurs when the location of the error pattern is beyond the detection capability at $l_u = v+\kappa$ iteration which indicates the case of $v + \kappa < e$.

2.2 Numerical Analysis of fault probability for the proposed early stopped technique

Based on the above heuristics, the error event of observing consecutive zero discrepancies during decoding iterations is investigated as follows. A non-zero discrepancy occurs after performing $v+\kappa$ BM decoding iterations and the codeword $c \in \{\zeta^{v+\kappa} - \zeta\}$ which results in the proposed technique failing to provide a correct BM decoding. Hence, the probability of malfunction is given as follows.

$$\begin{aligned} P_{mf} = p[v + \kappa < e] &= \sum_{i=0}^{t} P[d(r, c) = i | c \in \zeta^{i+\kappa} - \zeta] \\ &= \sum_{i=0}^{t} P[d(\xi, c) = i | c \in \zeta^{i+\kappa}] - P[d(\xi, c) = i | c \in \zeta] \end{aligned} \quad (1)$$

According to [7], the bounds of the probability P_{ud} that an undetected error will occur can be bound by the assumption of a long codeword length n and m is equal to the message length,

$$P_{ud} = \sum_{i=0}^{t} P[d(\xi, c) = i | c \in \zeta] \cong 2^{-mt} \sum_{s=0}^{t} \binom{n}{s} \sum_{h=t+1}^{n} \binom{n}{h} \varepsilon^h (1-\varepsilon)^{n-h} \quad (2)$$

The undetected error probability of the difference between upper and lower bounds is limited to 1%. We further extend the bounds of the probability of

an error pattern given by [14] and [7]. The conditional probability of a BCH code $\zeta^{d'}$ that has minimum Hamming distance d' is interpreted as follows.

$$P[d(\xi, c) = i | c \in \zeta^{d'}] \cong \sum_{h=(d'+1)/2}^{n} \binom{n}{h} \varepsilon^h (1-\varepsilon)^{n-h} \qquad (3)$$

Substituting (3) into (1), the probability of malfunction can be estimated as

$$\begin{aligned} P_{mf} \cong 2^{-mt} [& \sum_{s=0}^{t} \binom{n}{s} \sum_{h=(s+\kappa+1)/2}^{n} \binom{n}{h} \varepsilon^h (1-\varepsilon)^{n-h} \\ & - \sum_{s=0}^{t} \binom{n}{s} \sum_{h=t+1}^{n} \binom{n}{h} \varepsilon^h (1-\varepsilon)^{n-h}] \end{aligned} \qquad (4)$$

Furthermore, (4) can be simplified further by bounds of the type considered in [7] and define $\lambda_1 = (s+\kappa+1)/(2n)$ and $\lambda_2 = (t+1)/n$.

$$P_{mf} \cong 2^{-mt} \sum_{s=0}^{t} \binom{n}{s} [2^{-nE(\lambda_1, \varepsilon)} - 2^{-nE(\lambda_2, \varepsilon)}] \qquad (5)$$

where $E(\lambda, \varepsilon)$ is the relative entropy between the binary probability distribution λ and ε.

$$\begin{aligned} & E(\lambda, \varepsilon) = H(\varepsilon) + (\lambda - \varepsilon) H(\varepsilon) - H(\lambda) \qquad (6) \\ & = \lambda log_2(\lambda/\varepsilon) + (1-\lambda) log_2((1-\lambda)/(1-\varepsilon)) \end{aligned}$$

Based on the above observing d_j discrepancies during BM iteration, we illustrate the proposed early-stopped checking method, which is described below. The proposed method is denoted as the early-stopped(ES) version, and we provide three different versions. For BM decoding of the j-th iteration, we observe the following discrepancy based on the proposed method. We denote that δ_{max} represents the maximum error location degree of the BM algorithm.

Algorithm 1 ES version 1

Beginning from $j = 4$ as j-th iteration of the BM algorithm, verify the following steps:
1. Check Case A: $t + \delta_{max}/2 = j$
2. Check Case B: d_j, d_{j-1}, d_{j-2} and d_{j-3} are all zero.
3. If Case A and Case B are satisfied, terminate the BM decoding. Otherwise, proceed to the next BM iteration and return to Step 1.

Algorithm 2 ES version 2

Beginning from $j = 6$ as j-th iteration of the BM algorithm, verify the following steps:
1. Check Case A: $t + \delta_{max}/2 = j$
2. Check Case B: d_j, d_{j-1}, d_{j-2}, d_{j-3}, d_{j-4}, d_{j-5} are all zero.
3. If Case A and Case B are satisfied, terminate the BM decoding. Otherwise, proceed to the next BM iteration and return to Step 1.

Algorithm 3 ES version 3

Beginning from $j = \kappa$ as j-th iteration of the BM algorithm, verify the following steps:
1. Check the Case A: d_j, d_{j-1}, ..., $d_{j-\kappa+1}$ are all zero.
2. If Case A is satisfied, terminate the BM decoding. Otherwise, proceed to the next BM iteration and verify Step 1.
κ is set to 4, 5 or 6 before simulation.

ES version 1 in Algorithm 1 for checking 4 consecutive zero discrepancies and ES version 2 in Algorithm 2 for checking 6 zeros are presented to summarize a combination of early-stopping approaches considering [10] and our technique. However, ES version 3 in Algorithm 3 is the main core of our proposed approach to reveal the best complexity reduction.

3 Complexity analysis

The early stopped technique saves processing time and lowers power consumption. In this section, the analysis of multiplicative complexity is presented. Thanks to the author in [9] the upper bound of complexity analysis can be applied to evaluate the proposed technique by comparing it with the conventional BM algorithm and its related early-stopped technique. Since our proposed technique stops the conventional BM algorithm by certain conditions, the complexity of decoding can be computed by considering stopping the conventional BM algorithm at $e+\kappa$ iterations. Following the thread in [9], the multiplicative complexity C_{ES3} of the proposed ES version 3 is upper bound by $2e(e + \kappa) - 1$ which require at most $e + \kappa$ steps to check the discrepancies d_j. We summarize the comparison in Table I to show the merit of our proposed technique. e denotes the exact number of error bits caused by the channel. To compare with the proposed technique, the conventional BM algorithm, and conventional early-stopped technique enjoy low complexity when decoding the short codeword BCH code with a small t. However, the complexity of our proposed technique is not related to the parameter t and is only dominated by $e^2 + e\kappa$ which is quite beneficial for decoding long BCH code with larger correcting bits t. The complexity analysis results contribute to applications such as NAND flash and future satellite communication. A 16384 code length BCH code with large $t = 72$ is considered. For an example of $t = 72$, $e = 2$ and $\kappa = 6$, $1 - C_{ES3}/C_{ESBM}$ denote as the complexity reduction ratio of the proposed technique is equal to 79%. We present the complexity reduction ratio in Fig. 1 and the proposed technique can reach up to 80% improvement over the early-stopped approach in [9]. The complexity reduction comes from taking

the risk of decoding failure. Hence, we investigate the probability of decoding failure for the proposed technique in the following section.

Table 1: COMPARISON OF UPPER BOUNDS OF FINITE-FIELD MULTIPLICATIVE COMPLEXITY

C_{ESBM} [9]	C_{HV}	C_{BM}	C_{ES3}
$te + e^2 - 1$	$2te + \frac{1}{2}(e^2 - e)$	$2et - 1$	$2e(e + \kappa) - 1$

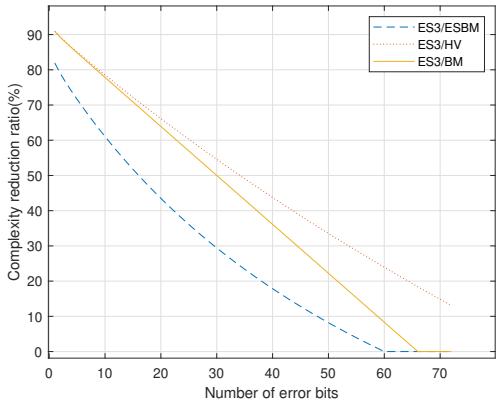

Fig. 1: The complexity reduction of the proposed technique with t=72.

4 Numerical results

The proposed early stopped technique has the capability to reduce the decoding latency. For example, the case of t error correcting which is equal to 72 leads to a huge cost of the area to implement the BCH decoder and the decoding latency of BM decoding degrades the system performance of the DMA accessing the flash memory. The authors in [7] obtained bounds on the probability of undetected errors in binary primitive BCH codes by applying the result to the code and showed that the bounds are quantified by the deviation factor of the true weight distribution from the binomial-like weight distribution. This approach presents a promising prediction for us to investigate that a long primitive BCH code can be robust to applying an early-stopped technique for a NAND flash system.

First, we consider a BCH code with a length that is equal to 31 in $GF(2^5)$, and that can correct $t = 3$, which has an outcome of 2^{31} codewords. During the decoding of the received codewords used to compute the discrepancy, we consider

Table 2: A FAILURE CASE OF EARLY STOPPED CHECKING

Discrepancy	> 0	0	0	d'
BM iteration	1	2	3	4

the following case in Table II. If we observe that the number of discrepancies is consecutively zero, we can compute the probability of a failure event occurring if d' is equal to non-zero. A conditional failure event can cause the proposed method to fail to decode a correct codeword which is subject to the observation of consecutively zero discrepancies. Consequently, it is interesting to investigate how should we set the parameter κ. The probability of erroneous early-stopped checking for the proposed ES version can be calculated using equation (5). In Fig 2, a BCH code with a length 1024 and $t = 17$ is presented to show that the highest probability of an erroneous event for proposed ES version 3 is 1.63752×10^{-12} for $\kappa = 1$, 1.7629×10^{-15} for $\kappa = 2$ and 1.77413×10^{-18} for $\kappa = 3$ respectively. As a result, we trade the failure probability with the early-stopped technique is not good enough while we use $\kappa = 1, 2, 3$. In particular, a threshold of κ is set as $\kappa \geqslant 4$ to obtain the result with the probability of an erroneous event as 1.7005×10^{-21} for $\kappa = 4$ and 1.85605×10^{-26} for $\kappa = 6$.

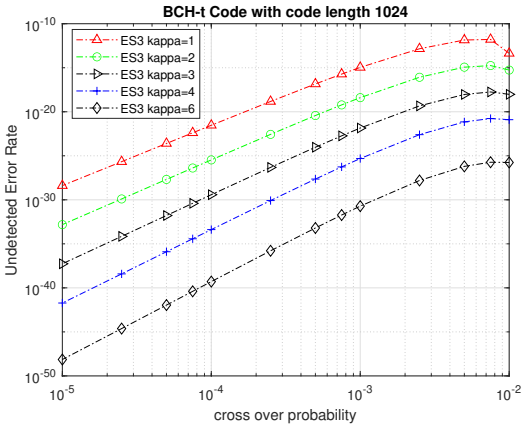

Fig. 2: The probability of undetected errors during early termination checking for ES version 3 using $GF(2^{10})$ BCH code t=17.

Furthermore, we show that the problem of decoding failure caused by early-stopped techniques can be neglected with the nature of long BCH codes. By using equation (5) as shown in Fig 3, a BCH code with a length 16384 and $t = 72$ is presented as an example to reveal the effectiveness of the proposed early-stopped checking method. For ES version 3 with $\kappa = 6$, the highest probability of undetected errors is calculated as 6.49437×10^{-119} over the cross-over

probability at 2.5×10^{-3}. It can be shown as an example that ES version 3

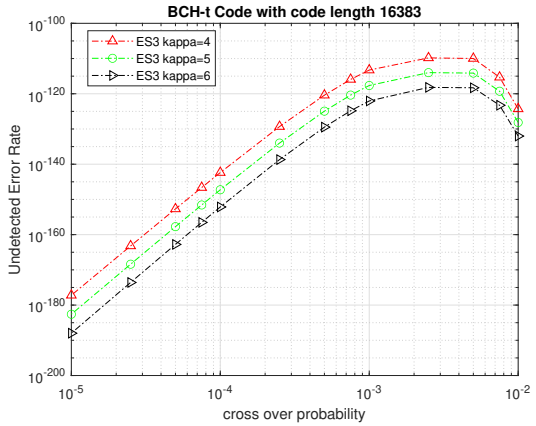

Fig. 3: The probability of undetected errors during early termination checking for ES version 3 using $GF(2^{14})$ BCH code t=72.

provides a reliable result for early termination checking by observing that the number of discrepancies is consecutively zeros. For practical applications, the proposed ES version 3 should be considered to prevent decoding failure over the firmware and decoder commuting period. As a matter of fact, the reliability of the early stopped method is the major concern for the flash controller rather than comparing the performance. If the detection failure occurred from the BCH decoder, the credibility of hard decoding would collapse. To address this issue, this paper focuses on the practical consideration of investigating the malfunction probability in this sense. To evaluate the credibility of the proposed method, we have given a complete test sample based on an FPGA board from the Altera family Statix II which operates at a clock rate of 110Mhz and uses BCH code length of 16384 that is suitable for a USB firmware testing. The system throughput is set to 480Mbps based on the USB 2.0 standard. The whole test sample quantity has a great amount of 5.9793×10^{35}. Each test sample contains the data package of 3 BCH code blocks and the code length is 16383 using $GF(2^{14})$ BCH code t=72. This result means that we never encountered any decoding failure during the time using a storage device based on the proposed design.

5 Conclusion

We have provided a practical solution for early termination checking while decoding BCH code. The complexity analysis and numerical results are presented to show the merit of the proposed technique which is suitable for long and large error-correcting capability of BCH code with complexity reduction up to 80%

over conventional early-stopped approach in [9]. The decoding failure is successful in exchange for decoding latency since the numerical result illustrates that the probability of undetected errors is lower than 6.49437×10^{-119} for $GF(2^{14})$ BCH code t=72. The FPGA testing on a USB device using 16384 code length of BCH code has been implemented to justify the reliability of the early termination checking strategy and the number of testing samples is accumulated up to 5.9793×10^{35}. This approach is shown to provide a solution for a practical design.

References

1. Y. Nishi, *Advances in Non-volatile Memory and Storage Technology*, Electronic and Optical Materials: Woodhead Publishing, 2014.
2. M. Srinivasan and D. V. Sanvate, *NAND Flash Memory Market Trends, Share, Size, Growth, Forecast 2030*, *Straits Research*, 2021.
3. Ruan, Mingkang and Titcheu, Thierry and Zhai, Ennan and Li, Zhenhua and Liu, Yao and E, Jinlong and Cui, Yong and Xu, Hong, *On the Synchronization Bottleneck of OpenStack Swift-Like Cloud Storage Systems*, *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 9, pp. 2059-2074, 2018.
4. Ma, Longyu and Sham, Chiu-Wing and Sun, Jing and Valencia Tenorio, Raul, *A Real-Time Flexible Telecommunication Decoding Architecture Using FPGA Partial Reconfiguration*, *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 10, pp. 2149-2153, 2020.
5. Ma, Longyu and Sham, Chiu Wing, *Optimized Layer Architecture for Layered LDPC Code Decoder*, *2018 International Conference on Advanced Technologies for Communications (ATC)*, pp. 287-291, doi=10.1109/ATC.2018.8587568.
6. Ma, Longyu and Chou, Hong-Fu and Sham, Chiu-Wing, *A Novel Data Packing Technique for QC-LDPC Decoder Architecture applied to NAND flash controller*, *2019 IEEE 8th Global Conference on Consumer Electronics (GCCE)*, pp. 897-898, doi=10.1109/GCCE46687.2019.9015393.
7. M.-G. Kim and J. H. Lee, *Undetected error probabilities of binary primitive BCH codes for both error correction and detection*, *IEEE Transactions on Communications*, vol. 44, no. 5, pp. 575-580, May 1996.
8. E. R. Berlekamp, *Algebraic Coding Theory*, New York, NY: McGraw-Hill, 1968.
9. C.-C. L. Chih-Wei Liu, *A view of Gaussian elimination applied to early-stopped Berlekamp-Massey algorithm*, *IEEE Transactions on Communications*, vol. 55, no. 6, pp. 1131-1143, Jun. 2007.
10. C. L. CHEN, *High-speed decoding of BCH codes*, *IEEE Transactions on Information Theory*, vol. 27, no. 2, pp. 254-256, 1981.
11. D. V. Sanvate and R. D. Morrison, *Decoder malfunction in BCH decoders*, *IEEE Transactions on Information Theory*, vol. 36, no. 4, pp. 884-889, Jul. 1990.
12. S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications (2nd edition)*, NJ: Prentice Hall, 2004.
13. W. Peterson and E. Weldon, *Error-Correcting Codes*, Comabridge, MA: MIT Press, 1972.
14. M. Srinivasan and D. V. Sanvate, *Malfunction in the Peterson-Gorenstein-Zierler decoder*, *IEEE Transactions on Information Theory*, vol. 40, no. 5, pp. 1649-1653, 1994.