



LOW-POWER PARALLEL CHIEN SEARCH ARCHITECTURE USING A TWO-STEP APPROACH

DEVASANI SAI TEJA¹, LANKA MADHU VENKATA MAHESH
VARMA², VONGURI AKASH³, DR. A. PRADEEP KUMAR⁴

^{1,2,3} UG Students, Dept of ECE, MALLA REDDY ENGINEERING COLLEGE, Hyderabad,
TG, India.

⁴ Professor, Dept of ECE, MALLA REDDY ENGINEERING COLLEGE, Hyderabad, TG,
India.

ABSTRACT

Error correction codes (ECCs) play a crucial role in modern communication systems, ensuring data integrity and reliability. The Chien search algorithm is widely used for evaluating polynomial roots in error-correcting codes such as Bose-Chaudhuri-Hocquenghem (BCH) and Reed-Solomon (RS) codes. However, conventional Chien search architectures are often power-intensive, limiting their efficiency in resource-constrained environments. This project presents a low-power parallel Chien search architecture utilizing a two-step approach to optimize power consumption and computational efficiency. The proposed method reduces switching activity by segmenting computations into two stages: an initial precomputation phase and a selective evaluation phase. By leveraging parallel processing, the design achieves significant power savings while maintaining high-speed error detection and correction performance. Experimental results demonstrate that the proposed architecture outperforms traditional Chien search implementations in terms of power efficiency and speed, making it suitable for high-performance error correction applications in communication and storage systems.

I. INTRODUCTION

Error detection and correction are essential components of modern digital communication and storage systems. BCH and Reed-Solomon codes are widely used in various applications, including wireless communication, optical storage, and data

transmission, to ensure reliable data recovery in the presence of noise and interference. The Chien search algorithm is a key step in these error correction processes, as it efficiently locates the error positions by evaluating the error locator polynomial over all possible codeword positions. However,

traditional Chien search architectures suffer from high power consumption, primarily due to redundant computations and excessive switching activity in hardware implementations. To address these challenges, this project proposes a low-power parallel Chien search architecture based on a two-step approach. The first step involves an initial precomputation phase, where partial evaluations are performed to eliminate unnecessary computations. In the second step, a selective evaluation phase dynamically refines the search, minimizing redundant operations and reducing overall power consumption. By leveraging parallel processing techniques, the proposed architecture enhances speed while



significantly lowering power dissipation. The rest of this study focuses on the design methodology, implementation details, and performance analysis of the proposed low-power Chien search architecture. Simulation results and comparative studies with conventional architectures demonstrate the advantages of the proposed approach in terms of power efficiency, processing speed, and hardware utilization. The findings of this research contribute to the development of more energy-efficient error correction mechanisms, which are critical for next-generation communication and storage systems.

II. LITERATURE SURVEY

A large number of errors, because they have the ability to correct the error correction described in this paper basically memory devices, the memory is used for the applications important to the majority logic decodable codes. However, the performance of the memory they need to have a big influence the decoding time. Technical standards, in memory devices become larger and more powerful error correction codes are needed. Euclidean geometry to overcome the problems in this paper as they use more modern codes. These codes can correct a number of errors, but usually requires complex decoders. Serially with the majority logic decoding circuitry can be implemented in a very simple, but it requires long periods of decoding. Memory, this is an important parameter for the access to the system can increase the time. Increase the size of the code, so the majority of the increase in the time of decoding the decoding logic (implemented serially), N iterations required.

EG-LDPC code, the code word is used in this method (Euclidean geometry -Low

Density Parity Check), there is a majority One Step Logic Decodable code. It uses the algorithm to check the code. There is nothing to check the algorithm, but code word is associated with a numerical value to be transmitted. The receiver then receives the code word at the end of the numerical values associated with the error identification is a comparison of some numerical value. There. The method is easy to implement using existing hardware. This method is more time for decoding. As well as the power consumption and the need for the region are high. One step is to identify the shortcomings of the majority of serially MLDD serial technique uses Logic decoder In this article, the author of the noisy channel (Additive White Gaussian Noise) in the presence of PSK and FSK modulation techniques Reed Solomon code (RSC) of the bit error rate (BER) on performance analysis explains. In this paper, 32- FSK (frequency shift keying), PSK (phase shift keying) modulation coded communication system is used for the simulation. In addition to the use of Monte Carlo simulation and calculation of the rate of BER MATLAB / SIMULINK is done using the tool. The results are shown using BERTOOL. In order to compare the performance of the block length is fixed, we have taken a different code rates.

After getting familiarized with the classification and characteristics of forward error correction codes, error detection and correction codes described in this section are compared to some. Hamming code - a Hamming encoder bits of parity bits are inserted into the message. The parity bits and parity bits of data to determine the different combinations to impose a fixed parity. The combinations are fixed parity check decoder. The parity bits are set

according to the decoder. The combination of the binary equivalent of the error location is determined. OK, that particular bit of data that is being flipped. Hamming code in the same error code. If any of the attempted correction of double errors can be detected.

III. PROPOSED TWO-STEP CS ARCHITECTURE

As indicated in (4), the p-parallel CS examines p error positions simultaneously, each of which generates a $1 \times m$ binary matrix denoting a Galois field (GF) element by computing

$$Y(\alpha^{wp+i}) = \sum_{j=1}^t \text{FFM}_{ij} = \sum_{j=1}^t \Omega_j A_{ij} = [\Omega_1 \ \Omega_2 \ \dots \ \Omega_t] \begin{bmatrix} A_{i1} \\ A_{i2} \\ \vdots \\ A_{it} \end{bmatrix} \quad (5)$$

where i ranges from 1 to p. The CS determines the presence of an error when $Y(\alpha^{wp+i})$ is 1, which implies that α^{wp+i} is a root of the error locator polynomial. In the GF of dimension m, the multiplicative identity element, α^0 or α^{2^m-1} , is defined as 1, i.e., $0(m-1:1)1(0)$, more precisely. The main idea comes from the fact that the absence of errors is guaranteed if some bits of $Y(\alpha^{wp+i})$ are not equal to those of $0(m-1:1)1(0)$. In the case of GF(24), for example, no presence of errors is guaranteed if $Y(\alpha^{wp+i})(3:2) = 0$. Similar to [9], a two-step approach is employed for early detection. In other words, the possibility of error presence is found by searching only the 1 MSBs rather than the entire m bits. Using this property, (5) can be decomposed into two matrix multiplications as (6)

$$\begin{aligned} Y(\alpha^{wp+i}) &= [\Omega_1 \ \Omega_2 \ \dots \ \Omega_t] \\ &\times \left(\begin{bmatrix} A_{i1,(m-1:m-l)} & 0_{(m-l-1:0)} \\ A_{i2,(m-1:m-l)} & 0_{(m-l-1:0)} \\ \vdots & \vdots \\ A_{it,(m-1:m-l)} & 0_{(m-l-1:0)} \end{bmatrix} \right. \\ &+ \left. \begin{bmatrix} 0_{(m-1:m-l)} & A_{i1,(m-l-1:0)} \\ 0_{(m-1:m-l)} & A_{i2,(m-l-1:0)} \\ \vdots & \vdots \\ 0_{(m-1:m-l)} & A_{it,(m-l-1:0)} \end{bmatrix} \right) \\ &= \text{concat} \left\{ \begin{bmatrix} A_{i1,(m-1:m-l)} \\ A_{i2,(m-1:m-l)} \\ \vdots \\ A_{it,(m-1:m-l)} \end{bmatrix}, \begin{bmatrix} A_{i1,(m-l-1:0)} \\ A_{i2,(m-l-1:0)} \\ \vdots \\ A_{it,(m-l-1:0)} \end{bmatrix} \right\} \quad (6) \end{aligned}$$

where $\text{concat}\{a, b\}$ stands for the concatenation of two binary matrices a and b. The former and the latter matrix multiplications are responsible for the 1 MSBs and the $m - 1$ LSBs of $Y(\alpha^{wp+i})$, respectively. Except the FFMs in the pth row, which is in fact used to update the registers, the two-step approach can be applied to the other FFMs in the p-parallel CS shown in Fig. 2.1. The two-step approach, in general, induces the longer critical path since one computation is decomposed into two small computations in series. To resolve the problem, the long critical path can be broken by inserting delay elements, which makes the two computations operate in a pipelined manner. Thus, the partial FFM for the LSBs is activated at the next clock cycle only when the partial FFM for the MSBs results in zero. Since the intermediate values in the registers are updated every cycle, the straightforward pipelining method is to latch all the intermediate values into separate registers to provide them to the partial FFM for the LSBs at the next cycle. However, this method demands a large amount of hardware resources. To prevent the increase in hardware complexity, in [9], the update of intermediate values was postponed when the former condition is satisfied. Hence, additional clock cycles are inevitable as one cycle is additionally taken whenever one of the $p - 1$ former computations is successful. Unlike the straightforward methods presented in [9], we save only activation signals, which are to enable the inputs to be

fed to the latter partial FFMs. For this, (2) is modified to (7), which processes the same computation as (2) with different intermediate values. Note that the multiplication by the multiplicative identity α^{2m-1} is to make the exponent a positive integer. Using (6) and (7), the two-step approach is finally reformulated as (8). As a result, the proposed two-step approach can detect the case of no errors early without degrading the critical path delay or the performance. Fig.3.1.1 illustrates the low-power CS architecture based on the proposed two-step approach. According to (8), the m -bit FFMs in the conventional CS are replaced with the pipelined two partial FFMs except for those in the p th row. Given the intermediate values from the registers, the first partial FFM processes the 1 MSBs and activates the second partial FFM responsible for the remaining $m-1$ LSBs at the next clock cycle only when the output of the former is 0. Otherwise, we can reduce the dynamic switching power by disabling the latter partial FFMs. Since each intermediate register can hold one of all possible GF elements, the latter partial FFM is activated once every 2^l clock cycles on the average. Furthermore, it is worth noting that the sum of the hardware complexity for the former and the latter partial FFMs is almost the same as the conventional FFM. Therefore, additionally required in the proposed architecture are the p 1-bit registers and the $(p-1)tm$ -bit buffers. It is important to decide how many 1 bits are appropriate for the former partial FFMs. The more bits are examined in the former, the latter partial FFMs will be accessed less resulting in a large power reduction, but the former partial FFM will suffer from the increased power dissipation, and vice versa. To find an optimal bit width of the first

processing, the ratio of the power saving achieved by the two-step approach is simply modeled as

$$P(p, l, m) = \frac{1}{p} \times \frac{m}{m} + \frac{p-1}{p} \times \frac{l}{m} + \frac{p-1}{p} \times \frac{m-l}{m} \times \frac{1}{2^l}. \quad (9)$$

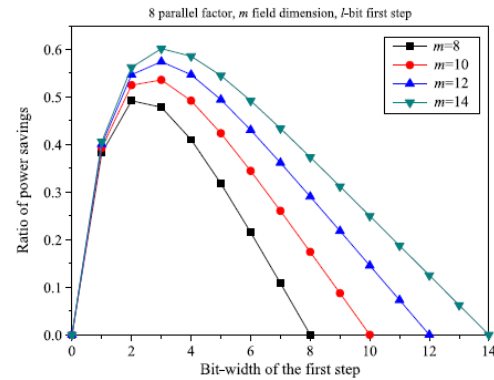


Fig.1. Estimated power savings versus the bit width of the first step for various field dimensions.

The simplified model is based on two assumptions: 1) The power dissipation mainly comes from FFMs; and 2) the power consumed in an FFM is proportional to the bit width and the number of access. Although the model is highly simplified, it is quite proper to estimate the overall tendency, as the power dominance of FFMs has been observed in a similar application [13]. As a matter of fact, the simplified model is considerably accurate as described in Section IV. The first term stands for the FFMs in the p th row, and the second and third terms denote the first and second partial FFMs for the other FFMs. Based on (9), Fig.3.2 describes the ratio of power saving, and the maximum indicates the optimal power saving of a configuration. For example, in the case of $m = 14$, 60% power saving is expected compared with the conventional architecture when the first step processes three MSBs. It is recommended in a practical realization to find a more accurate optimal bit width by investigating

several candidate bit widths near the bit width resulting from the model.

III. RESULTS

The low-power CS architecture based on the proposed twostep approach is compared with the conventional architecture for various configurations of field dimension, parallel factor, and error-correction capability. All the CS blocks are synthesized in a 130-nm CMOS technology at the operating frequency of 200 MHz, and the equally probable error model [7], [8] is adopted for the power consumption simulations. More precisely, when v errors occur in the BCH (n, k, t) codes, the average bit distance between two adjacent errors becomes n/v since the model assumes that each bit in a received code word is corrupted with the same error probability. Fig. 3.3 shows how the field dimension and the bit width of the first step affect the power-saving ratio. For fair comparison, all the BCH codes are designed with a code rate of 0.93 and the parallel factor is set to 8. As shown in Fig. 3.3, the improvement resulting from the proposed architecture becomes more significant as the field dimension increases, and a small number of bits are sufficient in maximizing the power saving. For instance, the proposed two-step architecture for the BCH $(8752, 8192, 40)$ code over GF(214) achieves 49.3% power saving when the first partial FFM processes four MSBs early. Moreover, Figs. 5 and 6 illustrate how the parallel factor and

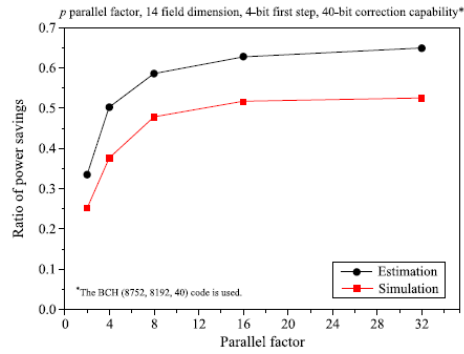


Fig. 2. Power-saving ratio versus parallel factor.

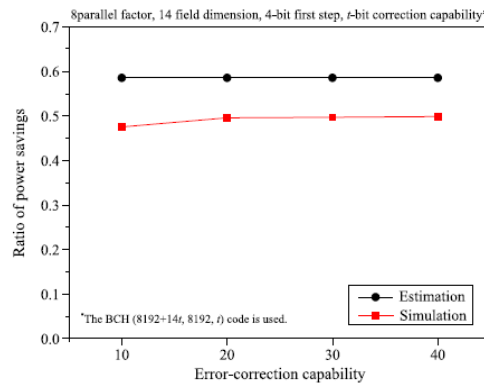


Fig.3. Power-saving ratio versus error-correction capability.

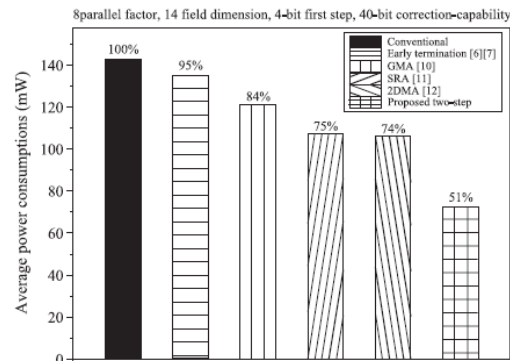


Fig.4. Average power consumption of eight-parallel CS blocks.

the error-correction capability affect the power saving. As the parallel factor increases, the power-saving ratio converges to 50% approximately and becomes saturated when the parallel factor is 8. It is also shown that the power-saving ratio is almost independent of the error-correction capability. In Figs. 3.2-3.4, we can see that the simplified power model (9) can be used as a good estimate, although

there is an offset between the estimated and the simulated power savings. Due to the additional buffers addressed in Section III, the proposed architecture increases 10% hardware complexity approximately. The overall power consumption for the BCH (8752, 8192, 40) code is shown in Fig. 7, which is also achieved with the parallel factor of 8. The early termination [6], [7] is not effective in power saving when the number of errors is not small. Note that the POR [8] is not suitable for the parallel CS since complex computations caused by the polynomial update should be processed in parallel. The power consumption values are also measured for area-efficient architectures [10]–[12] that share common substructures. Although the previous area-efficient architectures are developed to reduce hardware complexity, they are also effective in saving power consumption. However, the power saving is limited to around 25%. On the other hand, the proposed two-step architecture saves almost half of power consumption by activating the second step only when the first step is successful. As the early termination [6], [7] and the area-efficient architectures [10]–[12] are independent of the proposed two-step method, they can be combined into the proposed method to further reduce the power consumption.

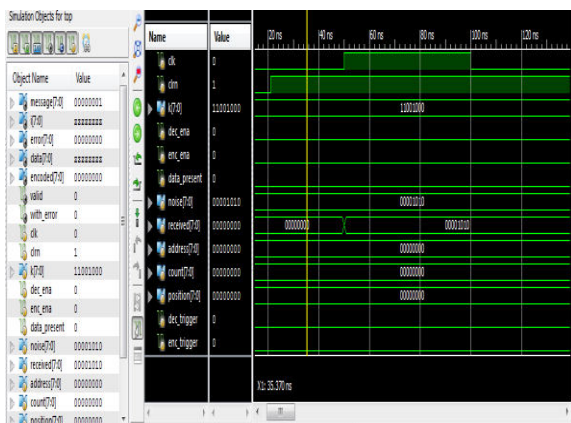


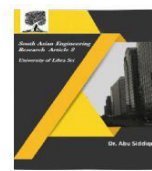
Fig.5. Output results.

IV. CONCLUSION

This brief has presented a new low-power architecture for parallel CS. The conventional CS is decomposed into two steps to achieve a significant power saving by reducing access to the second step. Under the equally probable error model, the low-power CS architecture is compared with the conventional architecture for various configurations of field dimension, parallel factor, and error-correction capability. Experimental results show that the proposed architecture reduces up to 50% power consumption compared with the conventional parallel CS. The power saving becomes more significant as the parallel factor or the field dimension increases. The proposed two-step CS is also applicable to other linear block codes such as the Reed–Solomon codes.

V. REFERENCES

- [1] S. Lin and D. J. Costello, Error Control Coding: Fundamentals and Applications, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2004.
- [2] S. B. Wicker, Error Control Systems for Digital Communication and Storage. Englewood Cliffs, NJ, USA: Prentice-Hall, 1994.
- [3] Y. Lin, C. Yang, C. Hsu, H. Chang, and C. Lee, “A MPCN-based parallel architecture in BCH decoders for NAND Flash memory devices,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 58, no. 10, pp. 682–686, Oct. 2011.
- [4] Y. Lee, H. Yoo, and I.-C. Park, “High-throughput and low-complexity BCH decoding architecture for solid-state drives,” IEEE Trans. Very Large Scale Integr. Syst., vol. 22, no. 5, pp. 1183–1187, May 2014.



- [5] X. Zhang and Z. Wang, "A low-complexity three-error-correcting BCH decoder for optical transport network," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 59, no. 10, pp. 663–667, Oct. 2012.
- [6] K. Lee, S. Lim, and J. Kim, "Low-cost, low-power and high-throughput BCH decoder for NAND flash memory," in *Proc. IEEE ISCAS*, May 2012, pp. 413–415.
- [7] Y. Wu, "Low power decoding of BCH codes," in *Proc. IEEE ISCAS*, May 2004, pp. II-369–II-372.
- [8] S. Wong, C. Chen, and Q. M. Wu, "Low power Chien search for BCH decoder using RT-level power management," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 19, no. 2, pp. 338–341, Feb. 2011.
- [9] H. Weingarten, E. Sterin, O. A. Kanter, and M. Katz, "Low Power Chien-Search Based BCH/RS Decoding System for Flash Memory, Mobile Communications Devices and Other Applications," U.S. Patent 2010 013 1831 A1, May 27, 2010.
- [10] Y. Chen and K. K. Parhi, "Small area parallel Chien search architectures for long BCH codes," *IEEE Trans. Very Large Scale Integr. Syst.*, 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. Circuits Syst. II, Exp. Briefs*, vol. 55, no. 5, pp. 427–431, May 2008.
- [12] Y. Lee, H. Yoo, and I.-C. Park, "Low-complexity parallel Chien search structure using two-dimensional optimization," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 8, pp. 522–526, Aug. 2011.
- [13] H. Choi, W. Liu, and W. Sung, "VLSI implementation of BCH error correction for multilevel cell NAND Flash memory," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 18, no. 5, pp. 843–847, May 2010.