www.ijiarec.com

ISSN:2348-2079

Volume-6 Issue-1

International Journal of Intellectual Advancements and Research in Engineering Computations

A survey of solid-state drives using BCH decoding architecture

V.Marimuthu[1],D.Ramya[2],R.K.Uma Maheswari[3],S.Arunodhaya[4]
M.E VLSI Design
Department of Electronics and Communication Engineering
Nandha Engineering College(Autonomous),Erode.
[1]E-mail: marimuthuvelusamy1997@gmail.com
[2]E-mail: ramyadeva95@gmail.com
[3]E-mail: umahap@gmail.com
[4]E-mail: arunodhaya333@gmail.com

***Abstract*—This paper presents a high-throughput and low-complexity BCH decoder for NAND flash memory applications, which is developed to achieve a high data rate demanded in the recent serial interface standards. To reduce the decoding latency, a data sequence read from a flash memory channel is re-encoded by using the encoder that is idle at that time. In addition, several optimizing methods are proposed to relax the hardware complexity of a massive-parallel BCH decoder and increase the operating frequency. In a 130-nm CMOS process, a (8640, 8192, 32) BCH decoder designed as a prototype provides a decoding throughput of 6.4 Gb/s while occupying an area of 0.85 mm^2**

Index Terms—BCH code, circuit optimization, digital integrated circuits (ICs), flash memory, VLSI.

I. INTRODUCTION

A solid-state drive (SSD), or solid-state disk is a solid-state storage device that uses integrated circuit assemblies as memory to store data persistently. SSD technology primarily uses electronic interfaces compatible with traditional block input/output (I/O) hard disk drives (HDDs), which permit simple replacements in common applications. New I/O interfaces like SATA Express and M.2 have been designed to address specific requirements of the SSD technology. SSDs have no moving mechanical components. This distinguishes them from traditional electromechanical drives such as hard disk drives(HDDs) or floppy disks, which contain spinning disks and movable read/write heads. Compared with electromechanical drives, SSDs are typically more resistant to physical shock, run silently, have quicker access time and lower latency. However, while the price of SSDs has continued to decline over time (24 cents per gigabyte as of 2017), consumer-grade SSDs are (as of 2017) still roughly four times more expensive per unit of storage than consumer-grade HDDs. As of 2017, most SSDs use 3D TLC NAND-based flash memory, which is a type of non-volatile memory that retains data when power is lost. For applications requiring fast access but not necessarily data persistence after power loss, SSDs may be constructed from random-access memory (RAM). Such devices may employ batteries as integrated power sources to retain data for a certain amount of time after external power is lost. However, all SSDs still store data in electrical charges, which slowly leak over time if left without power. This causes worn out drives (that passed their endurance rating) to start losing data typically after one (if stored at 30°C) to two (at 25°C) years in storage; for new drives it takes longer. Therefore, SSDs are not suited for archival purposes. Hybrid drives or solid-state hybrid drives (SSHDs) combine the features of SSDs and HDDs in the same unit, containing a large hard disk drive and an SSD cache to improve performance of frequently accessed data. In coding theory, the BCH codes or Bose–Chaudhuri–Hocquenghem codes form a class of cyclic error-correcting codes that are constructed using polynomials over a finite field (also called Galois field). BCH codes were invented in 1959 by French mathematician Alexis Hocquenghem, and independently in 1960 by Raj Bose and D. K. Ray-Chaudhuri. The name Bose–Chaudhuri–Hocquenghem (and the acronym BCH) arises from the initials of the inventors' surnames (mistakenly, in the case of Ray-Chaudhuri). One of the key features of BCH codes is that during code design, there is a precise control over the number of symbol errors correctable by the code. In particular, it is possible to design binary BCH codes that can correct multiple bit errors. 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 and two-dimensional bar codes.

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

1. Calculate the syndromes s_j for the received vector
2. Determine the number of errors t and the error locator polynomial $\Lambda(x)$ from the syndromes
3. Calculate the roots of the error location polynomial to find the error locations X_i
4. Calculate the error values Y_i at those error locations
5. Correct the errors

During some of these steps, the decoding algorithm may determine that the received vector has too many errors and cannot be corrected. For example, if an appropriate value oft is not found, 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.

II.RELATED WORK

There have been a lot of work done on BCH Decoding Architecture for Solid-State Drives New implementation techniques and different methods are introduced in the Decoding Architecture.

2.1 “On-chip error correcting techniques for new-generation flash memories” ,[1]

In new-generation Flash memories, issues such as disturbs and data retention become more and more critical as a consequence of reduced cell size and decreased oxide thickness. Furthermore, the progressive increase in the cell count within a single die tends to decrease device reliability. In particular, reliability issues turn out to be more critical in multilevel (ML) Flash memories, due to the reduced spacing between adjacent programmed levels. It is therefore deemed that the use of on-chip error correction codes (ECCs) will gain widespread acceptance in large-capacity Flash memories. ECCs for Flash memories must have very fast and compact encoding/decoding circuitry so as to have a minimum impact on memory access time. The area penalty due to check cells must also be minimized. Moreover, specific codes must be developed for ML storage. This paper presents error control coding techniques and schemes for new-generation Flash memories, focusing on ML devices. The basic concepts of error control coding are reviewed, and the on-chip ECC design procedure is analyzed. Dedicated codes such as polyvalent ECCs, able to correct data stored in ML memories working at a variable number of bits per cell, and bit-layer organized ECCs are described. This paper has addressed these issues, describing error control coding techniques and schemes for new-generation Flash memories. The attention has been focused on the case of multilevel storage, which is considered the most critical from reliability standpoint.

2.2 “Low-Cost, Low-Power and High-Throughput BCH Decoder for NAND Flash Memory” ,[2]

The BCH codes are widely used as Error Correcting Code (ECC) schemes for NAND Flash Memories. There have been strong demands to implement NAND Flash controller having low cost, low power and high throughput. We focus on BCH implementation since it has the largest portion in the controller. In this paper, we configure the 3-stage pipelined BCH decoder: syndrome computation, Berlekamp-Massey algorithm and Chien search. We implement BCH decoder supporting 800MB/s using the pipelined structure and the early termination method. we implemented the BCH decoder supporting 800MB/s with 3-stage pipelined structure for the syndrome computation, the Berlekamp-Massey algorithm and the Chien search. This efficient structure enables to lower the size down to 230K gate counts. In addition, implemented BCH decoder has low power by using the early termination method.

2.3 "A 26.9 K 314.5 Mb/s Soft (32400,32208) BCH Decoder Chip for DVB-S2 System" ,[3]

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. 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.

2.4 "An Adaptive-Rate Error Correction Scheme for NAND Flash Memory" ,[4]

ECC has been widely used to enhance flash memory endurance and reliability. In this work, we propose an adaptiverate ECC scheme with BCH codes that is implemented on the flash memory controller. With this scheme, flash memory can trade storage space for higher error correction capability to keep We have discussed the hardware implementations of the BCH codec. To tackle the increasingly serious problem of reliability and endurance of flash memory, we have proposed an adaptive-rate error correction scheme that allows changing the correction capability. We have also designed an example of 4-mode adaptive-rate BCH codec, with four different correction capabilities. The synthesis result shows that our design will not influence the flash memory access because the throughput of our design has reached 119MB/s by using a TSMC 0.13μmtechnology, while most flash memories only provide 40MB/s data I/O. Therefore, with this scheme, flash memory can trade storage space for higher error correction capability to keep it usable even when there is a high noise level. it usable even when there is a high noise level.

2.5 "Low-Power High-Throughput BCH Error Correction VLSI Design for Multi-Level Cell NAND Flash Memories" ,[5]

The reliability is a critical issue for new generation multi-level cell (MLC) flash memories, there is growing call for fast and compact error correction code (ECC) circuit with minimum impact on memory access time and chip area. This paper presents a high-throughput and low-power ECC scheme for MLC NAND flash memories. The BCH encoder and decoder architecture features byte-wise processing and a low complexity key equation solver using a simplified Berlekamp-Massey algorithm. Resource sharing and power reduction techniques are also applied. Synthesized using 0.25-jm CMOS technology in a supply voltage of 2.5V, the proposed BCH (4148,

4096) encoder/decoder achieves byte-wise processing, and it needs an estimated cell area of 0.2mm2, and an average power of 3.18mW with 50MB/s throughput. Parallel BCH encoder and decoder are also presented with an architecture-level solution to the excessive complexity. Power reduction schemes are also applied.

2.6 "A Fully Parallel BCH Codec with Double Error Correcting Capability for NOR Flash Applications" ,[6]

A double error correcting (DEC) BCH codec is designed for NOR flash memory systems to improve reliability. Due to the latency constraint less than 10 ns, the fully parallel architecture with huge hardware cost is utilized to process both the encoding and decoding scheme within one clock cycle. Notice that encoder and decoder will not be activated simultaneously in NOR flash applications, so we combine the encoder and syndrome calculator based on the property of minimal polynomials in order to efficiently arrange silicon area. Furthermore, a new error location polynomial is developed to reduce the number of constant finite filed multipliers (CFFMs) in Chien search. According to 90 nm CMOS technology, our propose DEC BCH codec can achieve 2.5 ns latency with 41,705μm 2 area. In this paper, a fully parallel DEC BCH codec for NOR flash memories is presented. Based on our proposed encoding method, the combined encoder and syndrome calculator architecture is introduced to reduce the hardware cost. Moreover, the number of CFFMs, which dominate the hardware complexity of fully parallel Chien search unit, is significantly decreased from 2n to n+m. After implemented in 90 nm CMOS technology, the proposed fully parallel (274,256;2) BCH codec can achieve 2.5 ns latency with 41,705μm 2 silicon area.

2.7 "Hydra: A Block-Mapped Parallel Flash Memory Solid-State Disk Architecture" ,[7]

Flash memory solid-state disks (SSDs) are replacing hard disk drives (HDDs) in mobile computing systems because of their lower power consumption, faster random access, and greater shock resistance. We describe Hydra, a high-performance flash memory SSD architecture that translates the parallelism inherent in multiple flash memory chips into improved performance, by means of both bus-level and chip-level interleaving. Hydra has a prioritized structure of memory controllers, consisting of a single high-priority foreground unit, to deal with read requests, and multiple background units, all capable of autonomous execution of sequences of highlevel flash memory operations. Hydra also employs an aggressive write buffering mechanism based on block mapping to ensure that multiple flash memory chips are used effectively, and also to expedite the processing of write requests. Performance evaluation of an FPGA implementation of the Hydra SSD architecture shows that its performance is more than 80 percent better than the best of the comparable HDDs and SSDs that we considered. We plan to compare the two types of SSD architecture using workloads that are sufficient to trigger the utilization-dependent effects of garbage collection in the case of page-level mapping. We hope that this performance characterization will allow us to design an efficient RAID architecture, in which both SSD architectures are combined to achieve an improved level of overall performance.

2.8 "Low-Complexity Parallel Chien Search Structure Using Two-Dimensional Optimization" ,[8]

To achieve a high-throughput decoder, massive parallel computations are normally applied to the Chien search, but the parallel realization increases the hardware complexity significantly. To reduce the hardware complexity of the parallel Chien search, this brief proposes a 2-D optimization method. In contrast to the previous 1-D optimizations, the proposed method maximizes the sharing of common sub expressions in both the row and column directions. All the partial products needed in the parallel structure are represented in a single matrix, and the finite-field adders are completely eliminated in effect. Simulation results show that the proposed 2-D optimization leads to a significant reduction of the hardware complexity. For the (8191, 7684, 39) BCH code, the count of XOR gates in the parallel Chien search is reduced by 92% and 22%, compared to the straightforward and strength reduced structures, respectively. The single matrix operation makes it possible to find more CSEs that can be shared in the implementation, lowering the hardware complexity of the Chien search block. Moreover, addition circuits required in the previous architectures are completely removed. Experimental results show that the proposed architecture reduces up to 92% and 22% of XOR gates, compared with the straightforward implementation and the strength-reduced architecture, respectively. The proposed optimization algorithm is also applicable to the RS decoding as the Chien search is an essential processing step of the RS decoding.

2.9 "Ultra Folded High-Speed Architectures for Reed Solomon Decoders" ,[9]

In this paper, a new high-speed VLSI architecture for decoding Reed-Solomon codes with the Berlekamp Massey algorithm is presented. The proposed scheme uses the fully folded systolic architecture in which a single array of processors, computes both the error locator and the error-evaluator polynomials. The proposed scheme utilizes the folding property of systolic array architectures and reduces the number of multipliers and adders drastically at the expense of some compromise in the speed. More interestingly, the proposed architecture requires approximately 60% fewer multipliers and a simpler control structure than the popular RiBM architecture. The reduction in the number of multipliers and adders in the proposed architecture leads to smaller silicon area and lower power consumption. It can be concluded that the proposed uiBM and UiBM architectures utilize the folding of systolic array processors, which saves huge amount of silicon area and reduces the power consumption compared to other existing RS decoder architectures. The uiBM uses approximately 60% of the area compared to other architectures at the expense of sacrificing some cycles for solving the key equation. The UiBM architecture is single processing element based architecture, which saves huge amount of silicon area and is very useful in applications where the number of error correcting locations is small or data rate is not very high. The critical path of the uiBM and the UiBM architectures is less than half that of conventional architectures such as the iBM architecture and is comparable to those of riBM and RiBM architectures.

2.10 "An Overview of Error Control Codes for Data Storage" ,[10]

Early work on error control codes (ECC) began with technologists, mostly mathematicians, utilizing algorithms for simple applications. As circuit technology became more powerful and practical, work on ECC progressed to implement more sophisticated codes. The result is a variety of robust ECC algorithms and implementations. Every commercially available data storage device incorporates ECC designed specifically for its particular media, transport, electronics and operating characteristics. As a result, today's data storage devices tolerate a broad range of harsh conditions. In this paper, we will discuss the various codes which have ken employed in data storage devices; including tape, disks and solid-state recorders. We will then describe DATATAPE's new approach to implementing Reed-Solomon (RS) algorithms into programmable devices to achieve flexibility in ECC capability. The algorithms involved will be a function of media and operational characteristics. The goal is to achieve high speed, low cost, configurable RS encoded decoder devices, easily adapted to fit specific applications.

III.METHOD AND ANALYSIS

In a 130-nm CMOS process, we have implemented a (8640, 8192, 32) BCH decoder as a part of the prototype SSD controller .The 32-parallel architecture is selected to supply a throughput of 6.4 Gb/s at 200 MHz. As a result of the proposed optimizations, the proposed decoder takes only 110-k gates and occupies 0.85 mm 2. In addition, the decoding latency is reduced to 2.56 µs due to the re-encoding technique. Table I compares various BCH decoders including the proposed work. In terms of throughput, complexity and energy efficiency, the proposed BCH decoder is superior to the previous works, even though it is designed for higher error correcting capability except. If the technologies are normalized, the proposed decoder achieves much better energy efficiency than . A micro photo of the prototype.

IV. CONCLUSION

The decoding latency was remarkably reduced by applying the re-encoding method. The proposed fan-out limited CSE sharing method allowed a high-speed realization while saving a significant amount of gates. In addition, implemented BCH decoder has low power by using the early termination method. As compared with the conventional hard BCH decoder, our BCH decoder with soft information from LDPC decoder provides similar system performance.

REFERENCES

[1]. S. Gregori, A. Cabrini, O. Khouri, and G. Torelli, "On-chip error correcting techniques for new-generation flash memories,"Proc. IEEE, vol. 91, no. 4, pp. 602–616, Apr. 2003.

[2]. 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. (VLSI) Syst., vol. 18, no. 5, pp. 843–847, May 2010.

[3]. Y.-M. Lin, C.-L. Chen, H.-C. Chang, C.-Y. Lee, "A 26.9K 314.5 Mb/s soft (32400, 32208) BCH decoder chip for DVB-S2 system," IEEE J. Solid-State Circuits, vol. 45, no. 11, pp. 2330–2340, Nov. 2010.

[4]. T.-H. Chen, Y.-Y. Hsiao, Y.-T. Hsing, C.-W. Wu, "An adaptive-rate error correction scheme for NAND flash memory," inProc. 27th IEEE VLSI Test Symp., May 2009, pp. 53–58.

[5]. W. Liu, J. Rho, and W. Sung, "Low-power, high-throughput BCH error correction VLSI design for multi-level cell NANDflash memories," in Proc. IEEE Workshop Signal Process. Syst., Jan. 2006, pp. 303–308.

[6]. C.-C. Chu, Y.-M. Lin, C.-H. Yang, H.-C. Chang, "A fully parallel BCH codec with double error correcting capability for NOR flash applications," in Proc. IEEE Int. Conf. Acoust. Speech, Signal Process., Mar. 2012, pp. 1605–1608.

[7]. Y. J. Seong, E. H. Nam, J. Yoon, H. Kim, J.-Y. Choi, S. Lee, Y. H. Bae, J. Lee, Y. Cho, S. L. Min, "Hydra: A block-mapped parallel flash memory solid-state disk architecture," IEEE Trans. Comput., vol. 57, no. 7, pp. 905–921, Jul. 2010.

[8]. 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.

[9]. D. V. Sarwate and N. R. Shanbhag, "High-speed architecture for reed solomon decoders," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 9, no. 5, pp. 641–655, Oct. 2001.

[10]. B. Benjauthrit, L. Coday, and M. Trcka, "An overview of error control codes for data storage," inProc. IEEE Int. Non-Volatile Memory Technology Conf., 1996, pp. 120–126.