ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

LDPC DECODING OF NAND FLASH MEMORY BY SOFT DECISION ERROR CORRECTION METHOD

S.Hasma Shruthi*, M.Indu, A.Nandhini
*PG Student, Department Of ECE, SNS College Of Technology, Coimbatore, Tamilnadu,India.

ABSTRACT

The reliability of data stored in an high-density Flash memory devices tends to drop off rapidly because of the reduced cell size and multilevel cell technology. The Soft-decision error correction algorithms make use of multiple-precision sensing for reading memory which can solve this problem. They require a very complex hardware for high-throughput decoding method. In this method, we present a rate-0.97 (68254, 65536) summarized Euclidean geometry low-density parity-check code and its VLSI operation for increased throughput NAND Flash memory system. The aim employ the normalized a posterior probability (APP)-based algorithm, serial schedule, and unrestricted update, which will lead to simple functional unit, halve decoding iterations, and compact power consumption. The pipelined equivalent structural design is adopted for high-throughput decode, and memory-reduction techniques is employed to minimize the total chip size. The proposed decoder method is implemented in 0.13-μm CMOS tools, and the chip size and power consumption of the decoder are compared with those that of a BCH (Bose–Chaudhuri–Hocquenghem) decoding circuit performance compared with the error-correcting presentation and throughput.

KEYWORDS: Euclidean geometry low-density parity-check (EG-LDPC), low power, LDPC codes, NAND Flash memory, soft decision error correction.

INTRODUCTION

NAND Flash memory is widely used in many mobile devices, such as cellular phones, digital cameras, and smart pads, because of high capacity, fast access speed, and low power consumption. In particular, solid-state drives (SSDs) for notebook computers have become popular as high density NAND Flash memory devices are available. NAND Flash memory stores the information by changing the threshold voltage of floating gate transistors. Most of today's high-density NAND Flash memory devices store multiple (usually two) bits in a cell, and this multilevel cell (MLC) technology significantly increases the bit-error rate (BER).Moreover, the feature size of NAND Flash memory shrinks, the number of electrons in the floating gate of a transistor also decreases, and as a result, the memory is very prone to charge loss caused by long data retention. The cell-to-cell interference (CCI) also increasingly deteriorates the reliability of information stored at the floating gates. It is also well known that SSD applications usually demand high program and-erase cycles, which greatly affects the reliability of NAND Flash memory. NAND Flash memory devices have a spare region at each page, where parity bits for error correction can be stored. Hard-decision decoding algorithms, such as Hamming and BCH codes, have been widely used for NAND Flash memory error correction. However, as the process technology scales down continuously, more advanced error-correcting codes are needed to keep NAND Flash memory reliable. It is especially important to employ error-correcting algorithms that show good performance with a limited parity data ratio. Soft-decision error correcting methods that sense the threshold voltage of a memory cell in multiple-precision can increase the error-correcting performance because the reliability of stored information can also be utilized.

The remainder of this paper is organized as follows. Section II introduces the proposed method. In Section III, Eg-Ldpc Codes and Decoding Algorithms is discussed. In section IV Error Performance over Nand Flash Memory Channel was presented. In section V the Baseline Sequential Architecture was discussed. In section VI the Pipelined-Parallel Architecture was presented. In section VII the Simulation and Result was shown. In section VIII Conclusion was given.

PROPOSED METHOD

NAND Flash memory is used for storing the information by changing the threshold voltage of floating gate transistors. The cell-to-cell interference (CCI) also gradually more deteriorates the reliability of information store at the floating gates It is also well known that SSD applications usually demand high program and erase cycles, which greatly affects the reliability of NAND Flash memory NAND Flash memory devices have a spare region at each page, where parity bits for error correction can be stored. Hard-decision decoding algorithms, such as Hamming and Bose Chaudhuri Hocquenghem(BCH) codes, have been widely used for NAND Flash memory error correction However, as the process technology scales down continuously, more advanced error-correcting codes are needed to keep NAND Flash memory reliable.

EG-LDPC CODES AND DECODING ALGORITHMS

EG-LDPC codes are a class of finite-geometry (FG) ones that are either cyclic or QC, and they show fast convergence speed and lower error-floors than other classes of LDPC codes .The FG-LDPC codes contain redundant parity checks that give additional improvement in error-correcting performance. Because of these characteristics, we consider FG-LDPC codes to be good candidates for error correction of NAND Flash memory, which demands very low error-floor performance as well as low access latency.

In this paper, a rate-0.96 (69615, 66897) EG-LDPC code is developed considering the page size (8 k B) and the spare data ratio (maximum 5%) of the current generation of NAND Flash memory devices .The parity-check matrix consists of 17 sub matrices, and each sub matrix is a cyclically right shifted 4095×4095 matrix with the row and column weights of 16.

Algorithm 1: Serial Schedule of Normalized APP-based Algorithm with Conditional Node Update.

1:Initialize k=0
2: Initialize all $L_{MN}^{(-1)}=0$
3:Initialize all $Z_n=I_n=\log(p(x_n=0/y_n)/(p(x_n=1/y_n)=-2y_n/\alpha^2$
4:for m=1 to M do
5:for every n € N(m) do
6:end for
7: for every n € N(m) do
8: $Z_n=\{z_n$if $|z_n|=2^{q-1}-1, z_n+ L_{MN}^{(k-1)}- L_{MN}^{(k)}$ otherwise
9:end for
10:end for
11:Decide a hard decision vector $\dot{w}= \{\dot{w}_1,\dot{w}_2 \ldots \dot{w}_n\}$based on
$\dot{W}n=\{0$ if $z_n^{(k)}>0.1$ otherwise
12:if $Hw^T=0$ for maximum iteration number is related
Then
13:output the hard decision $\dot{w}$
14:else
15:k=k+1
16:Go to line 4;
17:end if

ERROR PERFORMANCE OVER NAND FLASH MEMORY CHANNEL

The error-correcting performance of the proposed algorithm is also measured using an MLC NAND Flash memory simulation model that includes random telegraph noise, incremental step pulse programming, CCI, and non uniform quantization. In particular, in order to support soft-decision LDPC decoding, we evaluate the effects of memory output precision using the LLR computation method that assumes a mixture of Gaussian distributions 2-bit MLC NAND Flash memory and the employed voltage sensing scheme, where the left-most peak corresponds to the erased state (symbol 11), and the remaining ones are three different programmed states (symbol 01, 00, and 10, respectively). In conventional NAND Flash memory with hard decision data output, only three sensing reference voltages (SRVs) are needed to resolve four different symbols, which corresponds to four-level output quantization. Soft decision error correction needs an increased number of SRVs, especially in the overlapping regions, R1, R2, and R3.

BASELINE SEQUENTIAL ARCHITECTURE

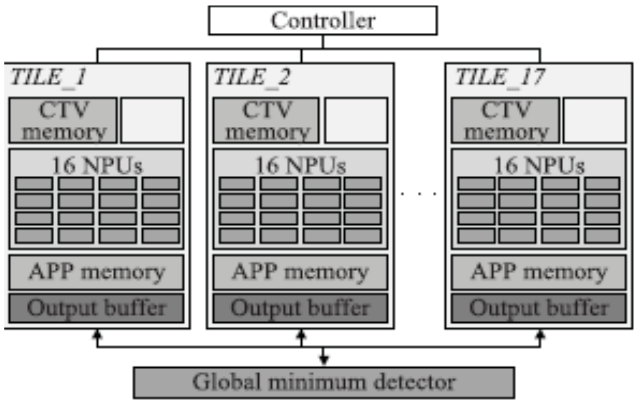

Fig.5.1 (a)Baseline architecture of the proposed LDPC decoder, Overall architecture.

Fig.5.1 (a) shows the overall baseline architecture that contains 17 tiles, a global minimum detector, and control logic. Each tile, shown in Fig5.2 (b), conducts the operations assigned to each sub matrix of the parity-check matrix.

The exclusive OR gate tree that computes the overall sign is omitted for clarity. Each tile contains 16 NPUs, an APP memory block, a CTV memory block, an output buffer, and a local minimum detector. The APP memory consists of 4095×q-bit shift registers for storing a posterior LLRs. The CTV memory, which consists of a 4 k×16×q-bit dual-port SRAM block, keeps 4095×16 CTV messages. We use 7 bits for the word-length, q, as explained in Section II. The capacity of the CTV memory is 7.44 M bits, which results from 4kchecks×272 CTVs/check×7 bits/CTV.

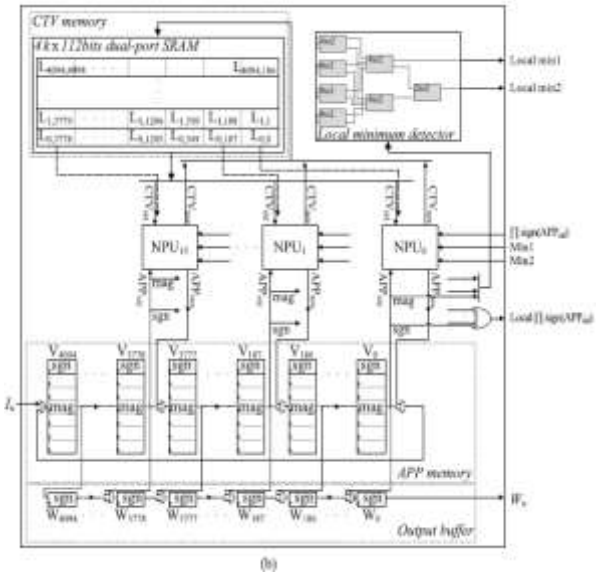

Fig.5.2 (b) Structure of a tile.

At the initialization phase, the channel LLRs obtained by reading the Flash memory with multiple-bit precision are transferred to the APP memory, while the CTV memory is initialized to zero. The initialization phase takes 4095 clock cycles, which is determined by the number of rows in the parity-check matrix, M.

After the initialization, the local minimum detectors find the two smallest magnitudes among 16 a posterior LLRs in each tile, whereas the global minimum detector selects the two smallest values among the output of the local detectors and provides the result to all the tiles. Then, each NPU updates the posterior LLR. Finally, the newly updated a posterior LLRs and CTV messages are written back to the APP and CTV memories, respectively.

As a check node and its neigh boring variable nodes are updated at each clock cycle and M is 4095, it takes 4095 clock cycles to complete one iteration. At the end of each iteration, the sign bits of the current a posterior LLRs are stored in the output buffer.

PIPELINED-PARALLEL ARCHITECTURE

In the serial schedule, a check node is updated first, and then the variable nodes connected to the renewed check node are changed using the newly modified CTV messages, which results in a long critical path.

As the critical path delay of the baseline architecture is 16.3 ns, the maximum clock frequency is limited to around 59 MHz unless pipelining technique is employed. In this case, the minimum decoding throughput is 106 Mb/s. To reduce the critical path delay, four pipeline registers are inserted in the NPU and the minimum detectors as shown in Fig.3.9. Which compares the cell area, the critical path delay, and the minimum throughput of the four decoders:

1) The baseline;
2) The pipelined;
3) The pipelined-parallel;
4) The proposed decoders (i.e., the pipelined-parallel architecture with the three memory reduction techniques).

The pipelined architecture reduces the critical path delay from 16.3to 4.4 ns with only 0.8% area overhead.

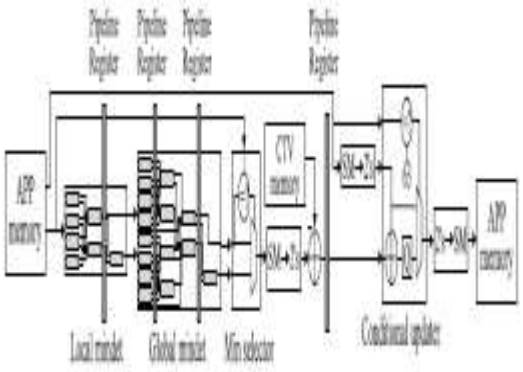

Fig.6.1 Critical path splitting through pipelining

SIMULATION AND RESULT

The proposed pipelined-parallel LDPC decoder for NAND Flash memory is synthesized, placed, and routed in 0.13-μm CMOS technology using Xilinx tools. The performance was discussed by using D flip flop,which is 1-bit storage device. The simulation shows the 1D with error and without error and 2D with error and without error and it is performed by using error correction method for providing high throughput and low power consumption.

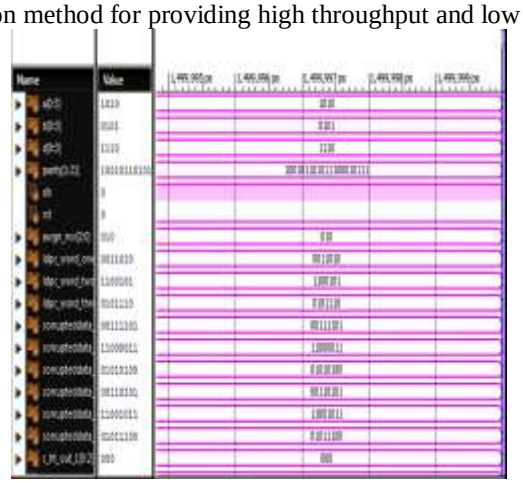

Fig.4.1 One dimensional with error

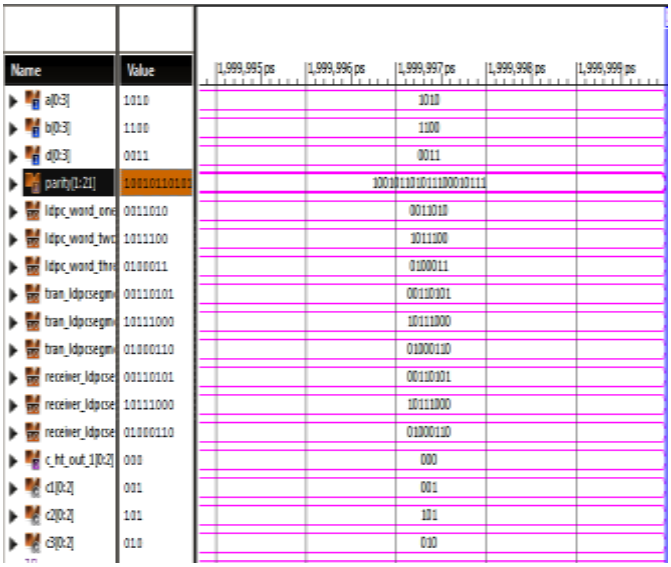

Fig.4.2 One dimensional without error

CONCLUSION

In this paper, we have developed a VLSI that implements a rate-0.96 (68254, 65536) EG-LDPC code for the application to soft-decision error correction of NAND Flash memory. The proposed decoder adopts five-stage pipelined eight-way parallel architecture for high throughput, and also employs a few chip area reduction techniques including word-length optimization, compression of check-to-variable messages, and approximation of the second minimum.

The serial decoding of the normalized APP-based algorithm with conditional node update is employed to lower the complexity. The developed LDPC decoder only demands 4% of the parity data ratio, while a hard-decision-based BCH decoding algorithm showing a comparable performance needs approximately 7.6% of the ratio. The developed VLSI achieves the maximum throughput of 8.13 GB/s with the chip area of 63.08 mm 2 in 0.13-μm CMOS process technology. When compared to the BCH decoding circuit, the LDPC code-based decoder demands approximately twice the chip size, but it needs only about half of the parity ratio, consumes less energy, and shows much higher throughput unless the signal-to-noise ratio of multiple precision memory output is near the un decodable region.

REFERENCES

[1] Blad. A, Gustafson. O, and Wanhammar. L (2005) "An early decision decoding algorithm for LDPC codes using dynamic thresholds," in Proc. Eur. Conf. Circuit Theory Design, Sep. pp.285– 288.
[2] Borkar. S (1999) "Design challenges of technology scaling," IEEE Micro, vol. 19, no. 4, pp.23–29.
[3] Cabrini. A, Gregori. S, Khouri. O, and Torelli. G (2003) "On-chip error correcting techniques for new-generation flash memories,"Proc. IEEE, vol. 91, no. 4, pp. 602–616.
[4] Caulfield A. M, Grupp L. M, Coburn. J, Swanson. S, Yaakobi. E, Siegel P.H, and Wolf J.K, (2009) "Characterizing flash memory: Anomalies, observations, and applications," in Proc. 42nd Ann .IEEE/ACM Int. Symp. Micro architecture, pp. 24–33.
[5] Chen. J, Dholakia. A, Eleftheriou. E, Fossorier M. P. C., and Hu X.-Y., (2005) "Reduced-complexity decoding of LDPC codes," IEEE Trans. Commun. vol. 53, no. 8, pp. 1288–1299.
[6] Chen. J and Fossorier M. P. C., (2001) "Decoding low-density parity check codes with normalized APP-based algorithm," in Proc. IEEE Global Telecommun. Conf., vol. 2. pp. 1026–1030.
[7] Chen T.-H., Hsiao Y.-Y., Hsing Y.-T., and Wu C.-W., (2009)"An adaptive-rate error correction scheme for NAND flash memory," in Proc. 27th IEEE VLSI Test Symp.,pp. 53–58.
[8] Cho. J, Kim. J, and Sung. W, (2010) "VLSI implementation of a high throughput soft-bit-flipping decoder for geometric LDPC codes," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 5, pp. 1083–1094.

[9] Compagnoni C.M, Ghidotti. M, Lacaita A.L, Spinelli A.S, and A. Visconti, (2009) "Random telegraph noise effect on the programmed threshold-voltage distribution of flash memories," IEEE Electron Device Lett., vol. 30, no. 9, pp. 984–986.
[10] Cui. Z, Wang. Z, and Liu. Y, (2009) "High-throughput layered LDPC decoding architecture," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 17, no. 4, pp. 582–587.
[11] Darabiha. A, Carusone A. C, and schischang F. K, (2008) "Power reduction techniques for LDPC decoders," IEEE J. Solid-State Circuits, vol. 43, no. 8, pp. 1835–1845.
[12] Darabiha. A, A. Carusone, and F. Kschischang, (2008) "Block-interlaced LDPC decoders with reduced interconnect complexity, "IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 55, no. 1, pp. 74–78.
[13] Dong. G, Xie. N, and Zhang. T, (2011) "On the use of soft-decision error correction codes in NAND flash memory," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 58, no. 2, pp. 429–439.
[14] Dong. G, Pan. Y, Xie. N, Varanasi. C, and Zhang. T, (2012) "Estimating information-theoretical NAND flash memory storage capacity and its implication to memory system design space exploration," IEEE Trans.Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 9, pp. 1705–1714.
[15] Dong. G, Li. S, and Zhang. T (2010) "Using data post compensation and pre distortion to tolerate cell-to-cell interference in MLC NAND flash memory,"IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 10, pp. 2718–2728.