WISER: Deep Neural Network Weight-bit Inversion for State Error Reduction in MLC NAND Flash

Jaehun Jang
Department of Semiconductor and Display Engineering
Sungkyunkwan University, Suwon, Korea
Memory Division, Samsung Electronics, Hwaseong, Korea
hun0317@g.skku.edu

Jong Hwan Ko
Department of Electronic and Electrical Engineering
Sungkyunkwan University
Suwon, Korea
jhko@skku.edu

***Abstract*—When Flash memory is used to store the deep neural network (DNN) weights, inference accuracy can degrade due to the Flash memory state errors. To protect the weights from the state errors, the existing methods relied on ECC(Error Correction Code) or parity, which can incur power/storage overhead. In this study, we propose a weight bit inversion method that minimizes the accuracy loss due to the Flash memory state errors without using the ECC or parity. The method first applies WISE(Weight-bit Inversion for State Elimination) that removes the most error-prone state from MLC NAND, thereby improving both the error robustness and the MSB page read speed. If the initial accuracy loss due to weight inversion of WISE is unacceptable, we apply WISER(Weight-bit Inversion for State Error Reduction) that reduces weight mapping to the error-prone state with minimum weight value changes. The simulation results show that after 16K program-erase cycles in NAND Flash, WISER reduces CIFAR-100 accuracy loss by 2.92X for VGG-16 compared to the existing methods.**

***Index Terms*—NAND flash memory, deep neural network, image classification, reliability**

I. Introduction

DNN (Deep neural network) inference on low-power edge devices is becoming more prevalent due to emerging Internet-of-Things (IoT) applications. To enable efficient on-device inference, one should carefully consider required resources on the edge platform. One of the critical resource requirements is the storage requirement, as the entire weight parameters of the DNN should be stored on device. Recently, Flash memory is emerging as the solution to the DNN weight storage in edge devices because of its high bit density and low cost [1], [2].

In spite of the efficiency of Flash memories, the most critical drawback is its limited reliability. The error rate of NAND Flash memories grows with the number of programming and erase cycles, the number of read operations, and the retention age [4]. When NAND Flash is used for DNN weight storage, the increased error rate due to aging will degrade the DNN inference accuracy. To protect the weights from Flash memory errors, a conventional error correction technique such as ECC (Error Correction Code) can be applied. However, ECC requires additional storage space called parity, and incurs energy consumption overhead that can increase with the interface speed.

In order to preserve DNN inference accuracy with lower ECC energy, recent studies have proposed various solutions. Mizushina et al. [5] reduced the ECC energy by optimizing the maximum iterations of LDPC (Low Density Parity Check) decoder layer-by-layer. Another study [6] prevented accuracy loss without ECC by storing the important weight bits in locations with lower BER(Bit Error Rate), and by moving the weights from the error-prone state to another state using parity. However, these methods still depend on ECC or parity that requires additional energy/storage overhead.

In this study, we propose a novel method that maintains the accuracy of DNN inference without using ECC or parity under NAND Flash aging conditions(Fig. 1). First, we invert LSB bits of a weight element to remove one Flash state that can cause MSB bit errors of the weight, thereby minimizing accuracy and improving the MSB page read speed (WISE: Weight-bit Inversion for State Elimination). If the initial inference loss due to this weight bit inversion is not acceptable, we limit the weight bit inversion within the minimum weight magnitude change while reducing the mapping to the error-prone states (WISER: Weight-bit Inversion for State Error Reduction). The simulation results with VGG-16 for CIFAR-100 dataset show that the proposed method makes the DNN weights more robust to the errors under Flash memory aging conditions.

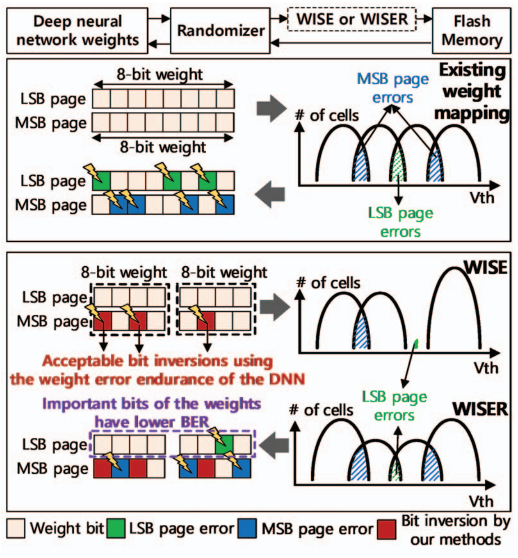


Fig. 1: Concept of weight-bit inversion schemes

II. Related Work

A system with Flash memory as a DNN weight storage may apply ECC(Error Correction Code) to protect data from errors. However, ECC needs additional space called parity, and consumes 21.8mW @ 400MB/s energy [7]. To address this issue, Mizushina et al. [5] reduced the ECC energy by optimizing the number of LDPC iterations layer-by-layer using each layer's acceptable BER. To reduce the important weight-bit errors, Deguchi et al. proposed the VADM(Value Aware Data Mapping) technique [6]. In MLC Flash memory, a single word line stores an LSB page and an MSB page. While the LSB page read requires only one SLC(Single Level Cell) read, the MSB page read requires two SLC reads to distinguish erase state and P3 state. Therefore, MSB page has a higher BER than LSB's. Using this BER difference, VADM stores LSBs of the weight in MSB page where errors occur more, and vice versa for the rest of the weights.

To avoid being programmed in error-prone states, the study [6] also proposed CPER(Critical Page Error Reduction), which uses additional storage space to encode data. For example, in MLC NAND Flash, there are four states (erase state, program state-1(P1), program state-2(P2) and program state-3(P3)) distinguished by each cell's V_{th}(threshold voltage). As P3 is the state with the most frequent errors cased by time retention, CPER selectively inverts all the weight bits to minimize mapping to P3, and writes flags in the parity whether inverted or not. The major drawback of this method is the use of parity bits. The storage overhead due to parity is from 6.25% to 25%, and this overhead increases as we want more robustness. Also, if there are errors in the parity, CPER cannot decode the inverted weight which has all errors compared to the original weight.

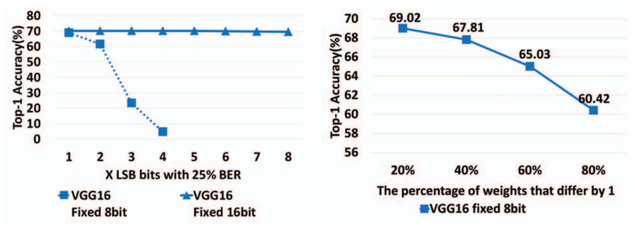
In this work, we propose a technique that enhances DNN inference robustness to the Flash errors without using ECC or parity. The proposed method simply inverts non-critical weight bits to change the weight mapping to the less error-prone Flash memory states.

III. Proposed Method

A. Motivation

In Flash memory, the data to be written is randomized with the equal number of randomly distributed zeroes and ones to minimize the data-dependent error behaviors [4]. Then, two bits of the randomized data are stored in one cell depending on the state to be programmed. In case of MLC NAND Flash, there are four possible states. The V_{th} of each cell can be changed by P/E cycling, read disturb, or retention age. Due to these reasons, some cells are not distinguished by the four states, thereby causing wrong state bits being read during the read operations. For example, P1 and P2 are error-prone states of the LSB page. Erase state under read disturb condition and P3 under retention condition are error-prone states of MSB page. To avoid the important weight-bits programmed to error-prone state, we can change the data mapping to one of the other states.

When changing the mapping between the original weights to the states, we can utilize the dependency of error robustness according to the bit error location in the weights. A recent study shows that 5% BER on two MSB bits leads to 20% accuracy loss, while the same BER on 2 LSB bits result in less than only 1% accuracy loss, with a fully connected network for MNIST [8].



(a) By LSB errors (b) By minimum weight change

Fig. 2: DNN accuracy loss with VGG-16

We also performed similar experiments with VGG-16 for the CIFAR-100 image classification task. Fig. 2(a) shows the inference accuracy when errors are added with 25% probability on the half of the weight bits in the least significant side (4 LSBs for FP(Fixed Point) 8bit, 8 LSBs for FP 16bit). The results indicate that there is a significant accuracy loss with the 8-bit fixed point weights, while the accuracy is almost preserved with the 16-bit fixed point weights. Fig. 2(b) is another test that adds errors causing the minimum value change of each weight element. For example, the minimum value is one when the weight is represented as the integer type. The result shows that the baseline accuracy is maintained within 1% drop when 35% of the weights changed by the minimum value for VGG-16, even with the 8-bit fixed point representation.

The above observations indicate that the Flash error rate is dependent on the programmed state, and the DNN inference robustness is dependent on the error location. It can be also inferred that the DNN inference accuracy has some level of robustness to the small amount changes of the weight values. Using these characteristics, this study proposes inverting non-critical bits of the weight to map the important weight bits to less error-prone states, thus maintaining the accuracy under NAND Flash aging conditions without using ECC or parity.

B. WISE(Weight-bit Inversion for State Elimination)

WISE reduces the LSB page BER by inverting the weight bit so that all the weights originally mapped to the P2 state are mapped to the P3 state. As described in Fig. 3, the method first finds all the bit combinations of the P2 state, which become the candidates for moving to the P3 state. Then, for those bit combinations, the bits in the MSB page (or the LSB part of the weight) are inverted to map them to the P3 state. As weight-bit inversions occur only in the half LSBs of the weights, the accuracy loss is negligible with the 16-bit fixed point representation, as shown in Fig. 2(a). All LSB page bits moving to P3 states have a lower BER because the voltage margin between P1 and P3 in Fig. 4(a) gets significantly wider than normal V_{th} distribution.

Fig. 4 (a) shows the change of V_{th} distribution after applying WISE. Originally, the weight data is uniformly mapped to all

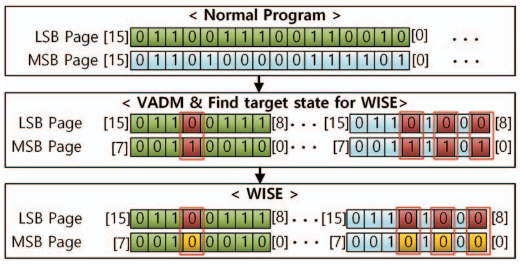
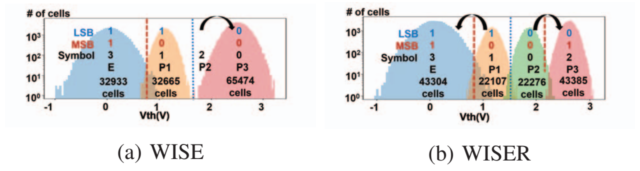


Fig. 3: WISE algorithm



(a) WISE (b) WISER

Fig. 4: V$_{th}$ distribution when we apply WISE or WISER

four states because of the randomizer. To distinguish between 0 and 1 out of a 2-bit symbol, the LSB page needs one SLC read with a read level colored with blue, and the MSB page needs two SLC reads. After applying WISE, only three states exist and the number of cells in the P3 state is doubled because all the weights originally mapped to P2 state are re-mapped to P3 state. If we re-map the symbol definitions for P2 and P3 in the reverse order, we can distinguish 0 and 1 for the MSB page by only one read level, thereby improving MSB page read speed.

C. WISER(Weight-bit Inversion for State Error Reduction)

WISE simply removes one error-prone state, and does not consider the actual value change due to the weight bit inversion. However, as mentioned in Section III.A, the DNN inference accuracy is almost preserved when we limit the weight value change as minimum. Therefore, for the DNN weights vulnerable to the bit inversion caused by WISE, we propose applying WISER that reduces weight mapping to two error-prone states of the LSB page while minimizing the magnitude change of the weight value. To minimize the magnitude change, we limit the maximum absolute difference due to weight-bit inversion to one.

Fig. 5 shows an example of WISER. The first step is finding candidate bits for weight re-mapping (P1 and P2 states) after randomization. These states are a set of zero data in the MSB page by the state symbol definition in Fig. 4(b). The second step is finding minimum weight change by inversions of candidate bits. Only 0_{th} bit inversion(12→13) and inversions in 0_{th} to 2_{nd} bits (12→11) make the minimum weight changes by one. WISER selects the second case because more error-prone states are moved to the other states by more inversions. Finally, three cells move to erase state or P3 state, therefore three MSBs of the weight has a lower BER.

Fig. 4(b) shows the V$_{th}$ distribution of the VGG-16's 32KB weights after applying WISER. It shows that 8.14% of the total number of cells are moved from P1 state to erase state, and

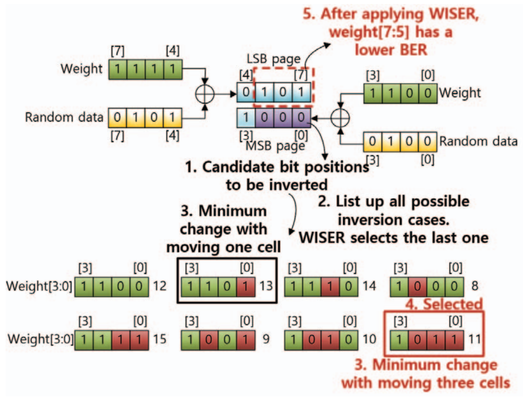

5. After applying WISER, weight[7:5] has a lower BER
Weight
Random data
LSB page
MSB page
1. Candidate bit positions to be inverted
2. List up all possible inversion cases. WISER selects the last one
3. Minimum change with moving one cell
Weight[3:0]
4. Selected
3. Minimum change with moving three cells

Fig. 5: WISER algorithm

8.03% moved from P2 state to P3 state, resulting in a total of 16.17% cells are moved to other states. When experimenting with all the weights of VGG-16, 52% of the weights are changed by WISER.

D. Algorithm Flow

The proposed algorithm flow is based on the criteria of the acceptable weight error rate. First, VADM is applied to store the half MSBs of the weights in the LSB pages. Then, when errors with 25% BER in half LSBs are acceptable, we apply WISE that inverts all the MSB page's bits of P2 state. If the accuracy loss due to inversions is unacceptable, we apply WISER that selectively inverts MSB page's bits with minimum weight value changes.

IV. Experimental Results

A. Simulation Framework

1) MLC NAND Flash Error Modeling: Various distributions can be used to model the V$_{th}$ distribution of Flash memory, but the Gaussian distribution is most similar to the real NAND Flash's characteristics [9]. Therefore, the Gaussian mixture distribution is used to model MLC NAND, and it is implemented by referring to the average and standard deviation for each state in [10]. The difference between the modeling results and real NAND data is within 5% BER.

2) DNN Model Configurations: VGG-16 for CIFAR-100 is trained with 32-bit floating-point weights and the SGD(Stochastic Gradient Descent) optimizer. We set the learning rate, batch size, and max epoch as 0.005, 128, and 250, respectively. As WISE and WISER change the trained weights via bit inversion, the initial accuracy loss is inevitable. WISE with the 16-bit fixed point weights decreases the accuracy by 0.77%. For the 8-bit fixed point weights, WISE results in a significant accuracy loss, so we apply WISER for 8-bit precision.

B. Analysis of BER by Bit Position

We conducted performance comparison in terms of the BER of each bit position. Fig. 6 shows the BER of the 8-bit weights after 16K P/E cycling. When VADM is applied, four MSB bits

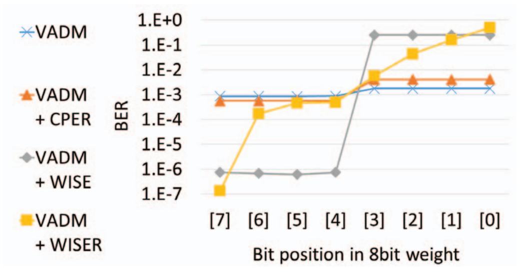
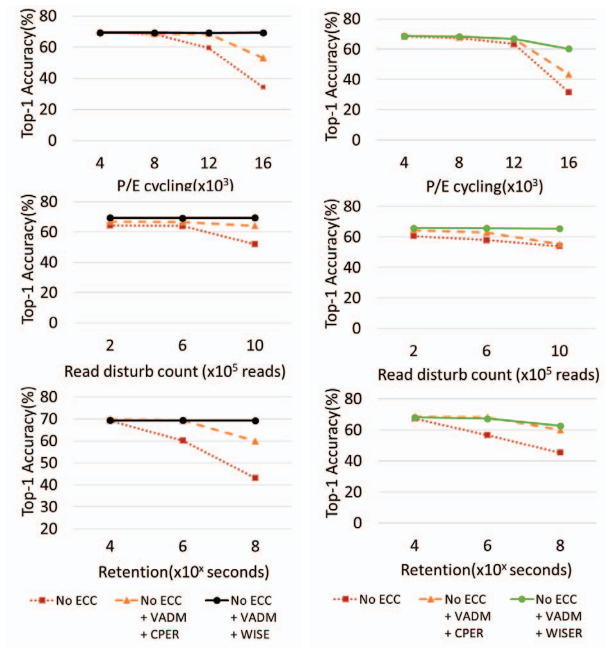


Fig. 6: Comparison of bit error rate after 16K P/E cycling



(a) WISE with 16bit fixed point (b) WISER with 8bit fixed point

Fig. 7: Comparison of accuracy loss

are stored in the LSB page and the rest four LSB bits are stored in the MSB page. Therefore, the BER of four LSB bits and four MSB bits follows the BER of the MSB page and LSB page, respectively. When CPER is applied, the BERs of four MSB bits and four LSB bits are polarized, i.e., the BER of four LSB bits increases by 2.32x compared to VADM, while the BER of four MSB bits is reduced to 0.65x. When WISE is applied, the BER of four LSB bits is increased by 142x compared to VADM, but BER of four MSB bits is reduced to about 1/1000. It shows that WISE makes the BER difference extremely polarized than CPER without using the parity. When applying WISER, the BERs for each bit within the same page are changed, and the BER for sign bit of the weight is greatly reduced. By inverting 50% of 0_{th} bits, the BER of the sign bit is reduced to 1/4000 compared to CPER and even 1/5 compared to WISE, which has the lowest BER for the sign bit.

C. Inference Accuracy Comparison with NAND Flash Aging

1) 16-bit Fixed Point: Fig. 7 shows the accuracy drop of DNNs with the 16-bit fixed-point weights under the Flash memory aging conditions. We use 25 percent parity for CPER as mentioned in [11]. When using the normal storage method without ECC, it has the most accuracy loss. When VADM and CPER are applied on VGG-16, the inference accuracy is droped by 17.09% for 16K P/E cycling, 5.96% for 10E+5 read disturb, and 10.15% for 1E+8 seconds retention. On the other hand, when using VADM and WISE together, the accuracy is maintained under all the aging conditions. There is only up to 0.79% accuracy loss for VGG-16.

2) 8-bit Fixed Point: Fig. 7 shows the accuracy drop of DNNs with the 8-bit fixed-point weights under the aging conditions. With VGG-16, WISER reduces the accuracy loss by 2.92x at 16K P/E cycling, 3.77x at 10E+5 read disturb, and 1.42x at 1E+8 seconds retention.

V. Conclusion

For systems storing DNN weights in Flash memory, we proposed two novel methods that minimize the accuracy loss under Flash aging conditions without using ECC or parity. The first method, WISE, selectively inverts weight-bits to avoid the weight mapping to the error-prone states, while reducing the number of read operations. For the weights vulnerable to the bit inversions by WISE, we proposed WISER that inverts weight bits while minimizing the value change. The results showed that the accuracy loss under Flash aging is significantly reduced compared to the existing methods.

VI. Acknowledgement

This work was partly supported by NRF (2019R1F1A1048115) and IITP (IITP-2020-0-01821) grant funded by the Korea government.

References

[1] S. Liang et al., "Ins-dla: An In-SSD Deep Learning Accelerator for Near-Data Processing," 29th FPL 2019.
[2] M. Hasan et al., "Reliability of Nand Flash Memory as a Weight Storage Device of Artificial Neural Network," IEEE Transactions on Device and Materials Reliability, 2020.
[3] M. Donato et al., "On-chip Deep Neural Network Storage with Multi-Level eNVM," DAC 2018.
[4] Y. Cai et al., "Errors in Flash Memory-Based Solid-State Drives: Analysis, Mitigation, and Recovery," arXiv preprint arXiv:1711.11427, 2017.
[5] K. Mizushina et al., "Layer-by-Layer Adaptively Optimized ECC of NAND Flash-based SSD Storing Convolutional Neural Network Weight for Scene Recognition," ISCAS 2018.
[6] Y. Deguchi et al., "3-D NAND Flash Value-Aware SSD: Error-Tolerant SSD without ECCs for Image Recognition," JSSC 2019.
[7] W. Lin et al., "A Low Power and Ultra High Reliability LDPC Error Correction Engine with Digital Signal Processing for Embedded NAND Flash Controller in 40nm CMOS," 2014 Symposium on VLSI Circuits Digest of Technical Papers.
[8] M. Hasan and B. Ray, "Tolerance of Deep Neural Network against the Bit Error Rate of NAND Flash Memory," IRPS 2019.
[9] Y. Cai et al., "Threshold Voltage Distribution in MLC NAND Flash Memory: Characterization, Analysis, and Modeling," DATE 2013.
[10] Y. Luo et al., "Improving 3D NAND Flash Memory Lifetime by Tolerating Early Retention Loss and Process Variation," POMACS 2018.
[11] Y. Deguchi et al., "12× Bit Error Acceptable, 300× Extended Data-Retention Time, Value-Aware SSD with Vertical 3D-TLC NAND Flash Memories for Image Recognition," CICC 2017.