SCIENCE CHINA
Information Sciences

• LETTER •

February 2023, Vol. 66 129403:1–129403:2
https://doi.org/10.1007/s11432-021-3381-3

Co-optimization strategy between array operation and weight mapping for flash computing arrays to achieve high computing efficiency and accuracy

Guihai YU, Peng HUANG*, Runze HAN, Lixia HAN, Xiaoyan LIU & Jinfeng KANG*

Institute of Microelectronics, Peking University, Beijing 100871*, China*

Received 28 June 2021/Revised 2 September 2021/Accepted 10 December 2021/Published online 14 November 2022

Citation Yu G H, Huang P, Han R Z, et al. Co-optimization strategy between array operation and weight mapping for flash computing arrays to achieve high computing efficiency and accuracy. Sci China Inf Sci, 2023, 66(2): 129403, https://doi.org/10.1007/s11432-021-3381-3

Dear editor,

Flash memory is a promising candidate to realize in-memory computing due to its mature fabrication technology [1]. Various efforts to implement neural networks using flash computing arrays have been reported [2, 3]. To realize higher computing efficiency, it is an effective method to activate more word lines (WL) at the same time for flash computing arrays [4]. When increasing the activated WL, more flash cells can involve in calculation simultaneously so the computing efficiency can be improved. However, the current on bit lines ($I_{\rm BL}$) will also increase when activating more WLs. The excessive current on bit line (BL) leads to two vital adverse effects, which restrict improving computing efficiency for flash computing arrays. One is that the $I_{\rm BL}$ will exceed the maximum current compliance of the metal interconnect line for BL (denoted as $I_{\rm Bm}$), which will degrade the reliability of BL. The other is the computing accuracy will decrease more seriously due to the increased influence of the line resistances.

Here, we propose a co-optimization strategy of array operation and weight mapping for flash computing arrays to avoid the excessive $I_{\rm BL}$, which can achieve high computing efficiency and accuracy. As shown in Figure 1(a), the source lines (SL) are applied with the positive voltage (SL+ in Figure 1(a)) as well as the negative voltage (SL− in Figure 1(a)). Figure 1(a) also shows the basic weight mapping for the proposed operation method. The positive and negative weights in one output channel are directly mapped to the same BL. The positive weights are mapped on the SL+s while the negative weights are mapped on the SL−s. The $I_{\rm BL}$ is the difference between the flash cell currents ($I_{\rm cell}$) for the positive and negative weights. The $I_{\rm BL}$s using the proposed array operation method can be much smaller, than the conventional method. Moreover, to correct the correlation between the weights and inputs, one input element should be applied to two conjugated WLs, which connects to the cells on the adjacent SL+ and SL−.

It can further avoid the excessive $I_{\rm BL}$ to optimize the weight mapping for the proposed array operation. The diagram of the proposed mapping optimization with example weights and $I_{\rm Bm}$ is shown in Figure 1(b). The first step is to find the minimum number of computing periods for finishing the calculation of the whole array ($N_{\rm CP}$), under the condition that $I_{\rm BL} \leqslant I_{\rm Bm}$. For the proposed array operation method, the potential largest $I_{\rm BL}$ ($I_{\rm BP}$) is defined by the non-zero weights involved in the calculation for each computing period. As an example, for the weights in Figure 1(b), if they are all calculated in one period, the $I_{\rm BP}$ will be $8 \times I_{\rm cell}$ when all the input elements for positive weights on the right BL (column) are "1" while all for negative weights are "0". If the $I_{\rm BP} \leqslant I_{\rm Bm}$, no $I_{\rm BL}$ will exceed $I_{\rm Bm}$ in this computing period. Therefore, the $N_{\rm CP}$ is determined as $N_{\rm CP} = \lceil I_{\rm BP0}/I_{\rm Bm} \rceil$, where $\lceil \cdot \rceil$ represents rounding up, $I_{\rm BP0}$ denotes the $I_{\rm BP}$ of the basic mapping when all the weights were calculated in one period.

After finding the $N_{\rm CP}$, the second step is to decide which weights should be calculated in each computing period. In the flash computing array, the weights mapped along the same WL are shared with the same input. To keep the correct correlation between the weights and inputs, the weights mapped along one WL are still mapped along one WL after the proposed mapping optimization. We denote the weights mapped along the same WL as a weight row (WR). In the proposed mapping optimization, the mapping rearrangement of weights is to rearrange the positions of WRs. At first, we should find out the WRs contributing to the $I_{\rm BP0}$ and divide them into the $N_{\rm CP}$ periods, which is to ensure that the $I_{\rm BP} \leqslant I_{\rm Bm}$ in any period. It follows two principles to decide which WRs can be divided in one specific computing period. The first principle is that the $I_{\rm BP}$ of this period cannot exceed $I_{\rm Bm}$ after dividing a new WR. The second principle is that the average $I_{\rm BL}$ should be the smallest one after dividing a new WR in this period, which helps to reduce the $I_{\rm BL}$s. Then, after the WRs contributing to $I_{\rm BP0}$ have been all divided, find the WRs contributing to the $I_{\rm BP0}$ of the rest weights and divide them into the $N_{\rm CP}$ periods

* Corresponding author (email: phwang@pku.edu.cn, kangjf@pku.edu.cn)

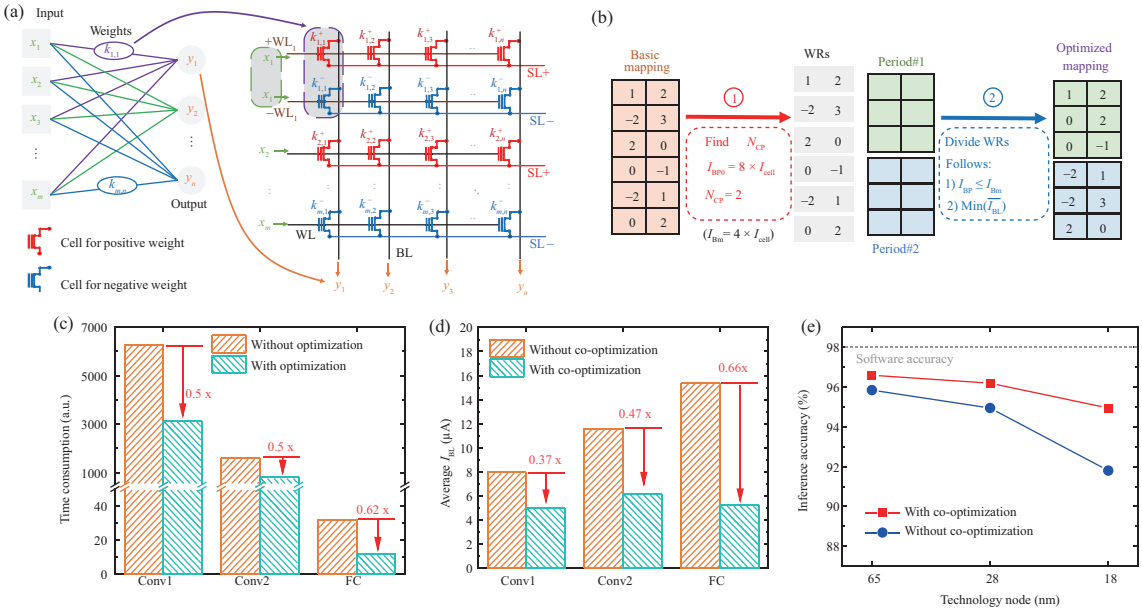

Figure 1 (Color online) (a) The proposed operation method of flash computing arrays and its basic weight mapping; (b) schematic of proposed weight mapping optimization with example weights and limit of I_{Bm}; (c) the computing time of each layer for LeNet3; (d) the reduction of the average I_{BL}; (e) the inference accuracy of LeNet3 with line resistances of different technology nodes.

using the two principles. Finally, repeat the dividing steps until all WRs are allocated. In addition, due to dividing WRs, the weights mapped on the same WL still correspond to the same input after optimization. The rearranged order of the input elements is fixed throughout the whole computing process. Therefore, to correct the relationship between the input and weights, we can write the fixed rearranged order into the inherent input control instructions, which do not need any complex input circuits.

We adapt LeNet3 to evaluate the performance of the co-optimization. This example CNN (convolutional neural network) has two convolution layers (Conv1, Conv2) and one fully-connected (FC) layer for MNIST recognition. The inference tasks are performed using the HSPICE tool. The I_{cell} is set as 2 μA according to [5] and the I_{Bm} is set as 80 μA according to ITRS [6]. As shown in Figure 1(c), under the same I_{Bm}, the total computing time using proposed co-optimization is reduced to about 0.5x of the one without co-optimization. Moreover, in order to reasonably evaluate the effect of the proposed method in reducing I_{BL} after improving the computing efficiency, the N_{CP}s without the co-optimization for comparison are set to be as same as the method with co-optimization, assuming that its reliability of BLs would not be damaged. As shown in Figure 1(d), the average I_{BL}s in the computing periods for the three layers of the example CNN are reduced by 37%, 47%, and 66% respectively using the proposed method. According to ITRS [6], the interconnect resistance of the 65, 28, and 18 nm flash memory technology node is about 0.9, 10.4, and 21.5 Ω, respectively. As shown in Figure 1(e), the inference accuracy losses induced by these interconnect resistances are all reduced using co-optimization, which is maximally reduced to 62% of the ones without the co-optimization.

Conclusion. In this study, we propose a co-optimization method of array operation and weight mapping for flash computing arrays to improve computing efficiency and accuracy. The proposed co-optimization can minimize the required computing periods to improve computing efficiency and can maximally decrease the I_{BL}s to alleviate the impact of interconnect resistance. The results show the computing speed is improved by 2x and the accuracy loss induced by the interconnect resistance is maximally reduced to 62%, compared to the conventional method.

Acknowledgements This work was supported by National Key Research and Development (Grant Nos. 2018YFE0100800, 2019YFB2205100), National Nature Science Foundation of China (Grant No. 61841404), and the 111 project (Grant No. B18001).

References

1 Yu S M. Neuro-inspired computing with emerging nonvolatile memorys. Proc IEEE, 2018, 106: 260–285
2 Lin Y Y, Lee F M, Lee M H, et al. A novel voltage-accumulation vector-matrix multiplication architecture using resistor-shunted floating gate flash memory device for low-power and high-density neural network applications. In: Proceedings of IEEE International Electron Devices Meeting (IEDM), 2018. 39–42
3 Malavena G, Spinelli A S, Compagnoni C M. Implementing spike-timing-dependent plasticity and unsupervised learning in a mainstream NOR flash memory array. In: Proceedings of IEEE International Electron Devices Meeting (IEDM), 2018. 35–38
4 Han R Z, Xiang Y C, Huang P, et al. Flash memory array for efficient implementation of deep neural networks. Adv Intell Syst, 2021, 3: 2000161
5 Xiang Y C, Huang P, Yang H Z, et al. Storage reliability of multi-bit flash oriented to deep neural network. In: Proceedings of IEEE International Electron Devices Meeting (IEDM), 2019. 919–922
6 International Technology Roadmap for Semiconductors (ITRS). 2015. http://www.semiconductors.org/resources/2015-international-technology-roadmap-for-semiconductors-itrs/