SCIENCE CHINA
Information Sciences

• LETTER •

June 2021, Vol. 64 169401:1–169401:2
https://doi.org/10.1007/s11432-020-2942-2

Flash memory based computing-in-memory system to solve partial differential equations

Yang FENG, Fei WANG, Xuepeng ZHAN, Yuan LI & Jiezhi CHEN*

School of Information Science and Engineering, Shandong University, Qingdao 266237, China

Received 13 March 2020/Revised 9 May 2020/Accepted 4 June 2020/Published online 23 September 2020

Citation Feng Y, Wang F, Zhan X P, et al. Flash memory based computing-in-memory system to solve partial differential equations. Sci China Inf Sci, 2021, 64(6): 169401, https://doi.org/10.1007/s11432-020-2942-2

Dear editor,

Many emerging non-volatile memories (NVM), such as resistive random access memory (RRAM) [1], phase-change memory (PCM) [2], and ferroelectric RAM (FeRAM) [3], together with the conventional flash memory [4, 5], have demonstrated their good capabilities in many artificial neural networks.

Similar to artificial neural network processing, a hardware matrix for vector-matrix multiply-and-accumulate (MAC) operation is necessary for partial differential equation (PDE) calculations. Recently, one-step solvers without iterations were developed by Sun et al. [6] for linear systems. However, PDEs in most cases are solved iteratively. The study from Zidan et al. [7] proposed an algorithm of memristor-based CIM system as PDE solvers. Owing to the low precision of memristor devices, the matrix array unit is small (3×3 matrix) and the precision extension technique has to be integrated. To solve this issue, flash memory represents a great choice because it is a non-volatile memory technology with ultra-high density, low cost, robust reliabilities, and better control of cell variations [8].

In this study, we design a flash-memory-based CIM hardware system to perform hard tasks requiring high precision and accurate solutions. On the basis of Jacobi iteration algorithm, we construct the iteration matrix to solve elliptic PDEs. The saturation region of memory cells is used for vector-matrix MAC operations to suppress device-to-device variations, and the computation convergence of different memory arrays is comparably studied in detail.

The core processing unit is constructed with flash memory arrays of 65 nm NOR flash memory technology, as shown in Figure 1(a). To achieve an accurate solution for numerical computations, the saturation region of flash memory cell is used in this study. The characteristic of the saturation region is described as $I = \beta \cdot (V_g - V_{th})^2$, were V_{th} is an adjustable parameter by changing the program states of memory cells, which is considered as one of the multipliers. By accumulating the current through the array cells adjusted by V_{th}, the result of integrator is the dot product of the pulse time and the current of array cells with a common source line, i.e., $Q = I \cdot T$. Based on Kirchhoff's current law, matrix addition can be achieved by measuring the integral of currents. The elements of the input vector and the coefficient matrix are represented by the pulse time and the threshold voltage, respectively.

To evaluate the performance of the flash-memory-based CIM system, we solve two typical elliptic PDEs: Laplace's equation ($\nabla^2 u = 0$) and Poisson's equation ($\nabla^2 u = f(x, y)$). When the solution of the unknown is divided into 12×12 2D distributed grids, the coefficient matrix becomes a 144×144 matrix. The large matrix magnifies the hardware challenges and reduces the computing efficiency, and the coefficient matrix M (Figure S1) is a typical sparse matrix with only a few nonzero elements, which is easy to extract useful values. To address this issue, we slice the big matrix into small slices, and two kinds of matrix partition schemes are studied comparably. Matrix partition method not only reduces the flash memory array, but also improves the calculation efficiency and reduces the power consumption.

The first matrix partition scheme is to slice the big matrix into 36×6 matrix slices with 36 inputs and 6 outputs for each matrix (see Appendix B for more details), and only intercepted active slices are considered (the slices containing non-zero elements). In addition, considering that the Jacobi iteration matrix only has nonzero values in the four diagonals, we can extract the four diagonals and convert them to four columns (144×1 matrix) containing nonzero values (see Appendix C for more details). We first solve a equation as follows:

$$\begin{cases} \nabla^2 u = 0, & \\ u(x, 1) = x, & 1 \leqslant x \leqslant 9, \\ u(x, 9) = 10 - x, & 1 \leqslant x \leqslant 9, \\ u(1, y) = y, & 1 \leqslant y \leqslant 9, \\ u(9, y) = 10 - y, & 1 \leqslant y \leqslant 9. \end{cases}$$

Next, we further solve a Poisson's equation. The exact solution of the equation is $x \cdot y \cdot \mathrm{e}^{-(x^2+y^2)}$. The equation can

* Corresponding author (email: chen.jiezhi@sdu.edu.cn)

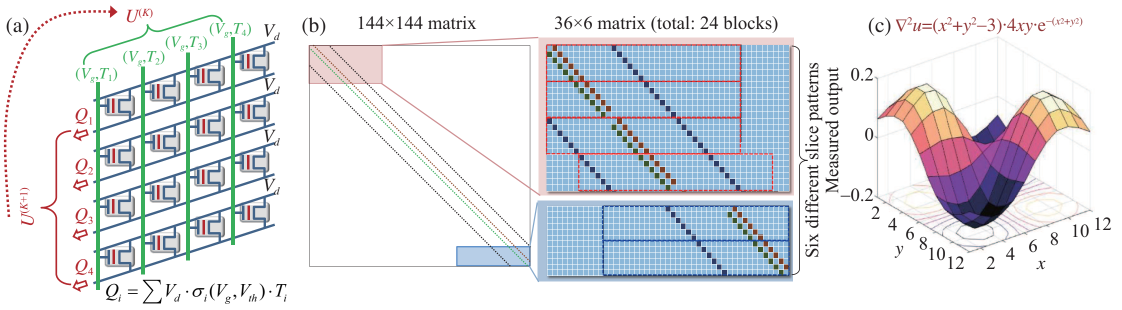
(a)
$U^{(K)}$
(V_g,T_1)
(V_g,T_2)
(V_g,T_3)
(V_g,T_4)
V_d
Q_1
Q_2
Q_3
Q_4
$U^{(K+1)}$
$Q_i=\sum V_d\cdot\sigma_i(V_g,V_{th})\cdot T_i$
(b)
144×144 matrix
36×6 matrix (total: 24 blocks)
Six different slice patterns
(c) $\nabla^2 u=(x^2+y^2-3)\cdot 4xy\cdot e^{-(x^2+y^2)}$
Measured output
0.2
0
−0.2
y
x

Figure 1 (Color online) (a) Schematic diagram of flash memory cell arrays. The values of the elements are mapped as pulse time and threshold voltage, and the vector-matrix multiplication operations of the slice are performed by supplying the input vector as voltage pulses to the rows and reading out the charge outputs at each string. (b) The way we divide the blocks leads to totally six different slice patterns by enlarging the selected area of 144×144 coefficient matrix generated by the FDM method. (c) Measured output obtained from 144×1 matrix after 50 iterations.

be described as

$$\begin{cases} \nabla^2 u = \left(x^2+y^2-3\right)\cdot 4xy\cdot \mathrm{e}^{-\left(x^2+y^2\right)}, \\ u\left(x,-0.2\right) = x\cdot(-0.2)\cdot \mathrm{e}^{-\left(x^2+0.04\right)}, & -0.2\leqslant x\leqslant 0.2, \\ u\left(x,+0.2\right) = x\cdot(+0.2)\cdot \mathrm{e}^{-\left(x^2+0.04\right)}, & -0.2\leqslant x\leqslant 0.2, \\ u\left(-0.2,y\right) = y\cdot(-0.2)\cdot \mathrm{e}^{-\left(y^2+0.04\right)}, & -0.2\leqslant y\leqslant 0.2, \\ u\left(+0.2,y\right) = y\cdot(+0.2)\cdot \mathrm{e}^{-\left(y^2+0.04\right)}, & -0.2\leqslant y\leqslant 0.2. \end{cases}$$

We used the aforementioned two matrix partition methods to solve equations and compared the results with those of the case of no matrix division. The results obtained from the 144×1 matrix are shown in Figure 1(c), and the others are summarized in Appendix D.

Acknowledgements This work was supported by China Key Research and Development Program (Grant No. 2016YFA0201802) and National Natural Science Foundation of China (Grant Nos. 91964105, 61874068).

Supporting information Appendixes A–D. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Ma W, Zidan M A, Lu W D. Neuromorphic computing with memristive devices. Sci China Inf Sci, 2018, 61: 060422
2 Raoux S, Wełnic W, Ielmini D. Phase change materials and their application to nonvolatile memories. Chem Rev, 2010, 110: 240–267
3 Mikolajick T, Dehm C, Hartner W, et al. FeRAM technology for high density applications. Microelectron Reliab, 2001, 41: 947–950
4 Guo X, Bayat F M, Bavandpour M, et al. Fast, energy-efficient, robust, and reproducible mixed-signal neuromorphic classifier based on embedded NOR flash memory technology. In: Proceedings of 2017 IEEE International Electron Devices Meeting (IEDM), 2017. 1–4
5 Bavandpour M, Mahmoodi M R, Strukov D B. Energy-efficient time-domain vector-by-matrix multiplier for neurocomputing and beyond. IEEE Trans Circ Syst II, 2019, 66: 1512–1516
6 Sun Z, Pedretti G, Ambrosi E, et al. Solving matrix equations in one step with cross-point resistive arrays. Proc Natl Acad Sci USA, 2019, 116: 4123–4128
7 Zidan M A, Jeong Y J, Lee J, et al. A general memristor-based partial differential equation solver. Nat Electron, 2018, 1: 411–420
8 Siau C, Kim K H, Lee S, et al. 13.5 A 512 Gb 3-bit/cell 3D flash memory on 128-wordline-layer with 132 MB/s write performance featuring circuit-under-array technology. In: Proceedings of 2019 IEEE International Solid-State Circuits Conference, 2019. 218–220