FRM-CIM: Full-Digital Recursive MAC Computing in Memory System Based on MRAM for Neural Network Applications

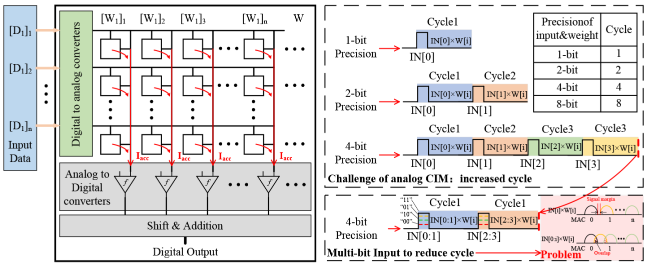
Jinkai Wang, Zekun Wang, Bojun Zhang, Zhengkun Gu, Youxiang Chen, Weisheng Zhao, Yue Zhang*
Fert Beijing Institute, MIIT Key Laboratory of Spintronics, School of Integrated Circuit Science and Engineering,
Beihang University, Beijing 100191, China
Email: yz@buaa.edu.cn

Abstract—Computing in memory (CIM) realizes energy-efficient neural network algorithms by implementing highly parallel multiply-and-accumulate (MAC) operation. However, the MAC delay of CIM will sharply increase with the improvement of computing precision, which restricts its development. In this work, we propose a full-digital recursive MAC (FRM) operation based on spin-transfer-torque magnetic random access memory (STT-MRAM) CIM system to enable fast and energy-efficient image recognition application. First, the fast FRM scheme is proposed by utilizing the recursive operations of read and addition in segmented bit-line array, which effectively reduces the delay of MAC operations to 3.5ns and 4ns for 8-bit and 16-bit input and weight precision, respectively. Second, we design an image recognition system using FRM-CIM architecture as the processing element (PE), where the adaptive pruning method for layers is proposed to improve the compatibility of it with the neural network. By performing image recognition for the MNIST and CIFAR-10 datasets, results show that the throughput and energy efficiency of the FRM-CIM system are 58.51TOPS/mm^2 and 11.3–56.72 TOPS/W under 8–16-bit precision, which are improved by 4.3 times and 2.6 times compared with the state-of-the-art works. Finally, the recognition accuracy can reach 96.65% and 82.7% on MNIST and CIFAR-10, respectively.

Keywords—Computing in memory (CIM), full-digital, recursive MAC, neural network, STT-MRAM.

I. Introduction

With the rapid development of artificial intelligence (AI), the data processed by neural network algorithms has increased exponentially. Von Neumann architecture is difficult to meet the performance requirements of AI because its structure of separated memory and computing results in a lot of energy and time spent on data transmission [1]. Computing in memory (CIM) eliminates above problem by integrating memory and computing functions. Meanwhile, CIM can perform highly parallel data processing, e.g., multiply-and-accumulate (MAC), by utilizing the characteristics of memory. Therefore, CIM is a promising approach to implement energy-efficient neural network processing with high performance.



Precisionof input&weight	Cycle
1-bit	1
2-bit	2
4-bit	4
8-bit	8

Fig. 1. (a) Analog CIM architecture. (b) Challenge of analog CIM.

As one of the cores of AI, neural network algorithm contains a large number of MAC operations. Utilizing the crossbar array structure of memory, the analog CIM architectures are proposed to efficiently implement MAC operations, as shown in Fig.1(a). The input data is converted from digital signals to analog signals of pulse or voltage through the digital-to-analog converters (DAC), which are loaded onto word-line (WL) to activate bit-cell stored the weight data. Each activated bit-cell will generate a logic current which represents the product property (i.e., I=VG) from the input data on WL and the datum stored in the bit-cell. These logic currents generated by the bit-cells on a column converge on the bit-line (BL) to implement the accumulation operation. An analog-to-digital converter (ADC) converts the total current into a binary number on each column to realize single-bit MAC operations. Then single-bit MAC results on each column are weighted using shift and addition circuit to obtain the final result of the multi-bit MAC. Analog CIM architecture realizes high energy efficiency by exploiting the high parallelism of the columns (i.e., single-bit MAC operations can be performed on each column.). Nevertheless, as the computing precision increases, the number of cycles required to obtain the final MAC result also increases, resulting in higher delay that will seriously reduce computing energy efficiency, as shown in Fig.1(b). To solve this problem, [2] proposes the multi-bit input method to reduce computing cycles. However, multi-bit input method will cause the decrease in signal margin between adjacent MAC results, which greatly affects the recognition accuracy of the ADC. Although charge-domain [3] and time-domain [4] reading schemes effectively improve the signal margin, they will increase the delay of a single cycle. Therefore, reducing the delay of MAC operation with high energy efficiency and reliability has become the focus of CIM technology. Digital CIM architecture has high reliability by implementing Boolean logic since the number of activated bit-cells in it is much smaller than analog CIM. At present, a variety of digital CIM schemes based on various memories have been proposed. Among them, spintronic memories are widely considered as a high potential candidate for digital CIM system designs, thanks to their intrinsic non-volatility, low power consumption and high speed of read/write operations [5], [6]. However, the MAC operations of them are executed by scheduling Boolean logic, which increases the complexity of computing and reduces the energy efficiency.

To solve the above issues, combining the advantages of digital and analog CIM architecture, we propose a full-digital recursive MAC (FRM) CIM system by using segmented bit-line (BL) array structure in spin-transfer-torque magnetic random access memory (STT-MRAM), which greatly improves the delay and energy efficiency of MAC operation to increase the efficiency of performing image recognition. First, the FRM method is proposed by scheduling the read and addition operations, where a voltage-following read cell

(VFRC) and segmented bit-line array are designed in order to recognize the datum of bit-cell with high parallelism and speed. Results show that the delay of MAC operation reduces to 3.5ns for 8-bit precision, which dropped by 1.7 times compared with [7]. Meanwhile, the MAC delay for16-bit and 32-bit precision are 4ns and 5.2ns respectively, which is the fastest among the reported works. Then, based on the FRM-CIM architecture, we construct an image recognition system. To further improve the compatibility of the FRM-CIM architecture with the neural network algorithms, we design the adaptive pruning method for each hidden layer. Finally, we evaluate the FRM-CIM system performance by performing image recognition for the MNIST and CIFAR-10 dataset. Results show that its throughput and energy efficiency are improved by 4.3 times and 2.6 times compared with state-of-the-art works. Meanwhile, the recognition accuracy can reach 96.65% and 82.7% on MNIST and CIFAR-10 dataset, respectively.

The rest of this article is organized as follows. Section II introduces the principle and structure of the proposed FRM-CIM architecture. Section III describes the designed image recognition system based on FRM-CIM architecture. Section IV presents the performance of the image recognition system. Conclusions are presented in Section V.

II. FRM-CIM Architecture

A. *Principle of FRM scheme*

Due to the recognition accuracy limitation of ADC circuit, the rang of MAC values distinguished on a column is limited. Therefore, the number of input data bits in a cycle is restricted, currently up to 3 bits, which results in an exponential increase in the number of execution cycles as the computing precision of the MAC operation increases, as shown in Fig. 2(a). For example, 8 cycles are usually required to complete the MAC operation of 8-bit input and weight. In each cycle, the signal-bit MAC operation is first implemented in each column of the memory array to achieve local MAC (MAC_{Ln}) result. Then the MAC_{Ln} result of each column will be weighted according to the weight and they are summed to obtain the global MAC (MAC_{Gi}) result. Finally, 8 MAC_{Gi} results obtained after 8 cycles will be weighted again according to the input and they are summed to achieve the final MAC (MAC_{Fin}) result. Although the number of cycles can be reduced by inputting multiple bits in one cycle, such as the cycle can be reduced to 4 cycles when 2 bits are input in one cycle, the rang of MAC values distinguished on a column is expanded resulting in the signal margin decline. Meanwhile, the shift operation for weighted values and addition are performed in each cycle of MAC operation caused by non-full precision input, thereby increasing delay and energy.

The proposed FRM scheme effectively solves the above problems through the full precision input, as shown in Fig. 2(b). In a cycle of FRM scheme, a full precision data is input into the CIM architecture to implement the multiplication operation with each bit of a weight data. They are added to the results obtained in the next cycle. For example, in first cycle-1, the first data IN_0 is input and multiplied by each bit of weight data W_0 on each column of memory array, i.e., IN_0×W_0[0], IN_0×W_0[1], …, IN_0×W_0[n], which achieves n MAC ($1MAC_{Ln}$) result. Then, in second cycle-2, the second data IN_1 is input and multiplied by each bit of weight data W_1, which achieves n MAC ($2MAC_{Ln}$) result. At this time, the $2MAC_{Ln}$ result on each column are added to the $1MAC_{Ln}$ result from the cycle-1. In this way, the multiplication and addition operations are then performed in subsequent cycles, which is similar to recursive operations. In final cycle-n, the $iMAC_{Ln}$ result will be weighted according to the input and they are summed to achieve the final MAC (MAC_{Fin}) result. Obviously, the shift and addition due to the weighted operation are eliminated, which effectively reduces the complexity of MAC operations and improves computational efficiency in CIM architecture. Meanwhile, the number of cycles executed is only related to the number of accumulations and has nothing to do with the precision of the input data, which can greatly reduce the execution cycle of high precision MAC operations.

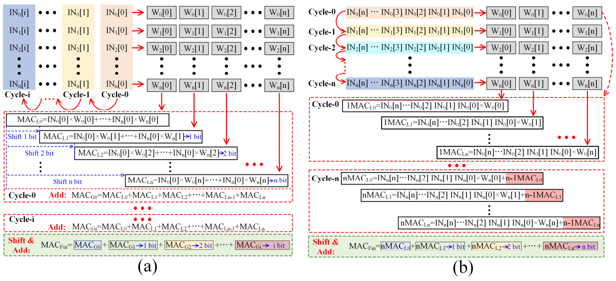


Fig. 2. (a) Principle of the MAC operation in analog CIM. (b) Principle of the proposed FRM scheme.

B. *FRM-CIM architecture*

Based on the principle of FRM scheme, we designed the FRM-CIM architecture. Fig.3 shows the proposed structure of FRM-CIM scheme based on segmented BL technique, where a BL is divided into 16 segments and each segment has 8 bit-cells. To achieve connection in the memory model and disconnection in the computing model, BL and source-line (SL) of adjacent segments are connected through NMOS transistors controlled by the ENCIM signal. Moreover, each segment of BL is set the proposed voltage-following read cell (VFRC) to simultaneously read a cell performed computing in each segment. Meanwhile, the outputs of all VFRCs on a BL are connected to computing line (CL) through NMOS transistors controlled by the DEi signal. The register-accumulate circuit mainly consists of the rising edge-triggered D flip-flops (DFFs) and full adders (FAs). The proposed all-digital MAC operation includes two phases: (1) read the computing bit-cell, and (2) accumulate the input data.

Before the beginning of FRM scheme, the ENCIM is high voltage to turn on the NMOS transistors in the model switch (MS), which makes all segments on a BL connected.

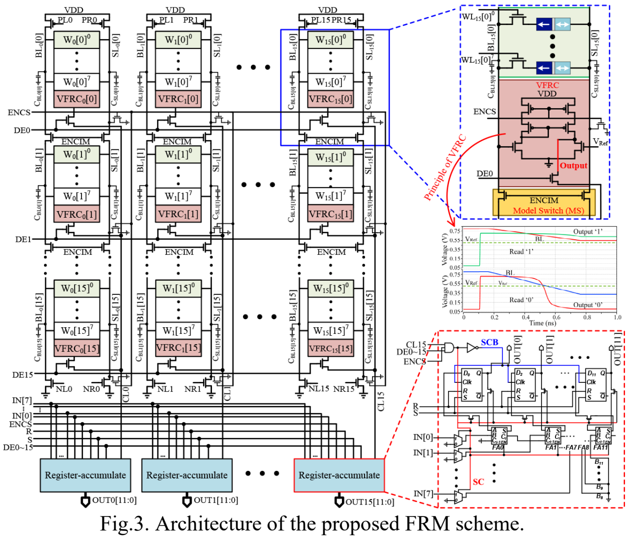


Fig.3. Architecture of the proposed FRM scheme.

Meanwhile, the BL is pre-charged to VDD by the PMOS transistor controlled by PLi signal. Then, ENCIM and EMCS are rise to VDD, causing each segment to enter the computing model independently. Note that the segmented BL technique also enables its parasitic capacitance to be equally divided, which means the parasitic capacitance of each segment ($C_{BLi[i]}$) is small. According to the characteristics of the proposed VFRC structure, this will greatly reduce the delay of the VFRC read operation. Besides, the read operation of the computing bit-cell in each segment on BL is carried out at the same time to decrease the access time of all-digital MAC operations. After that, the VFRCi[0] output of the first segment (i.e., 0 segment) on a BL is transferred to CLi by activated DE0 signal. For example, the VFRCi[0] output of '0' in Fig.4 (a) is transferred to CLi in Fig.4 (d). Note that the activation of DE0 requires a period of time for the register-accumulate circuit to perform addition and register operations, as shown in Fig.4 (b). In register-accumulate circuit, an AND logic is performed between CLi, DE0 and EMCS. If the CLi is '0' during the time that DEi is activated (i.e., the datum of weight is '0'), then SC signal is also '0', as shown in Fig.4 (e), which makes the FAs off and the DFFs in its original states, i.e, register-accumulate circuit does not work. That achieves weight-sparsity-aware energy-saving. On the contrary, if CLi is '1, for example, the VFRCi[1] output of '1' in Fig.4 (b) is transferred to CLi in Fig.4 (d) when DE1 is activated, then SC signal is '1' to enable the register-accumulate circuit work. At this time, the FA chain performs the addition operation for the input data (i.e., IN1[7:0]) and the data registered in DFFs. Because the SCB signal is the inverse of SC, SCB generates a rising edge when SC is a falling edge, as shown in Fig.4 (f), which is used as the Clk signal of rising edge-triggered DFF. On the other word, the DFFs will record the results generated by the FA chain when the DE0 activation ends. Therefore, the activation time of DEi is longer than the delay of the FA chain to ensure the accuracy of the computing. According to the above principle, the output of VFRC in each segment on a BL is transmitted to CLi sequentially to implement addition and register operations for input data in register-accumulate circuit, thereby realizing 16 accumulations without weight for 8-bit input data after activating DE0 to DE15. The main timing sequences of the proposed scheme are shown in Fig.4 (g). Finally, the 16 accumulations with weight for 8-bit input data can be achieved by the adder tree, where the OUT7 [11:0] is connected to the adder tree as the highest bit weight in the 8-bit weight MAC operation.

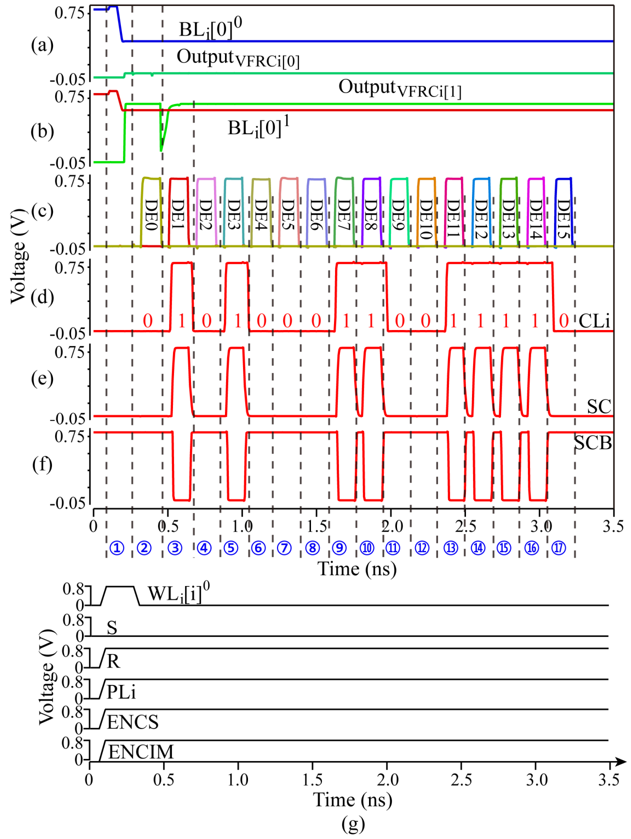


Fig.4. Transient simulation results of FRM scheme. (a) Read 0 of VFRCi[0]. (b) Read 0 of VFRCi[1]. (c) Waveform of DEi. (d) Waveform of CLi. (e) Waveform of SC. (f) Waveform of SCB. (g) Timing sequences of the proposed FRM scheme. ① Read each segment of BL in parallel. ②~⑰ Accumulation of input data.

III. Image Recognition System Based on FRM-CIM Architecture

A. *Overall of image recognition system*

Based on FRM-CIM architecture, we further design an image recognition system to preform deep neural network (DNN). To efficiently provide input feature data, the buffer is assigned to the FRM-CIM architecture, thus constituting the process element (PE) in image recognition system. Note that the memory array of FRM-CIM architecture is divided into two parts through the NMOS transistors that are divided the WL into two segments, where each part has 8 columns, to realize the MAC operations of 8-bit and 16-bit precision in an architecture. Moreover, the 16 register-accumulate circuits are also divided into two parts according to the column connecting them. Besides, the proposed image recognition system also consists of the global data buffer (GDB), accumulator (ACC), data width pruner (DWP) and the finite state machine (FSM), as shown in Fig. 5. GDB stores the input feature from off-chip memory or input feature betweens layers and it allocates input feature data to the PE. 8 rows×32 columns PEs form a PE block. Moreover, the ACC circuit is integrated into each PE block to implement accumulation operations for the PE results. Then, the results of ACC module are transmitted to the DWP circuit, which will detect the maximum value of output feature and prune it to appropriate size. The principle of DWP is introduced in section B. Finally, the operations of the above circuits are controlled by FSM module. Besides, a predictor module is integrated into the image recognition system based on the FRM-CIM architecture to directly demonstrate the output feature.

Fig.6 shows the mapping scheme that the image data are input into the proposed image recognition system. Firstly, the

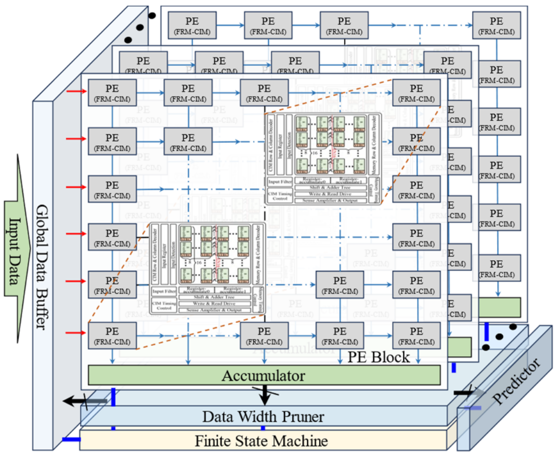


Fig.5. Image recognition system based on FRM-CIM architecture.

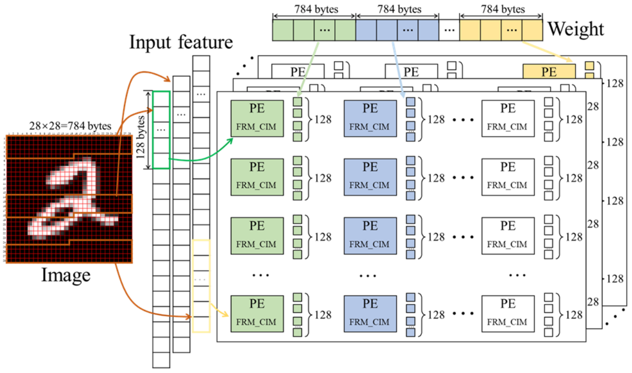


Fig.6. Mapping scheme of image recognition system based on FRM-CIM architecture.

weight matrix is expended into one long vector. The number of weights bytes that is the same as the number of input feature bytes is stored in a column of PEs. To improve the parallelism of computing, a column of PEs is set a channel and the input feature data is transmitted to it to implement DNN algorithm. Note that each PE can process 128 bytes of the input feature data. Therefore, a PE block can store no more than weights of 32 channels and the remaining weight bytes are stored in the next PE block.

B. *Principle of data width pruner*

In order to adapt to the different DNN algorithms and improve the compatibility of the FRM-CIM architecture with them, the proposed image recognition system will be repeatedly invoked to execute different layers of DNN. Therefore, the number of the input feature data bits need to be matched with the output data. However, the computation of each hidden layer results in an increase in the number of output data bits due to the data overflow. It is necessary to prune the output data to match the number of input features data bits.

However, the conventional pruning scheme that intercepts the fixed number of bits from the output data ignores the value range of the output data itself, which will cause the data to be clipped to zero or the data overflows. To solve this issue, we propose an adaptive pruning method that can determine the pruning position based on the distribution of the output data, as shown in Algorithm 1. In the proposed adaptive pruning method, a parameter reserve_bit is set and represents the number of signed bits reserved in pruning, ranging from 1 to 7. DWP circuit will first detect the index of the highest valid bit (HVB) from the output data generated by each channel (i.e., a column of PE). Note that the highest valid bit is the first bit opposite to the signed bit. The maximum index among them is selected and added to the reserve_bit to obtain the prune_bit. Then DWP will intercept the 8-bit data after the bit of prune_bit as the output data of it, which is stored in GDB as the input data of next hidden layer.

Algorithm 1: Prune for layers

```
Input: N: the number of input feature, W: the width of input feature data,
In_data: input feature data.
Output: Out_data: output feature data
Parameter: reserve_bit: bits reserved for signed bit
for i ← 1,2,3,…,N do
    for j ← W-1,W-2,W-3,…,1 then
        if In_data[i][j] ≠ In_data[i][W] then
            HVB ← j
            break
    if i = 1 then
        MAX_HVB ← HVB
    else then
        if HVB > MAX_HVB then
            MAX_HVB ← HVB
end
Prune_bit ← MAX_HVB + reserve_bit
for i ← 1,2,3,…,N do
    Out_data[i] ← In_data[i][Prune_bit:Prune_bit-8]
end
return Out_data
```

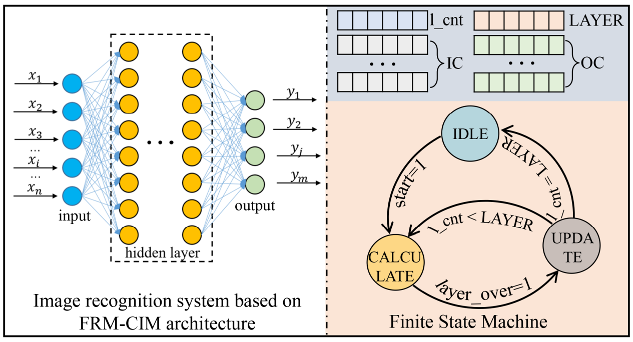


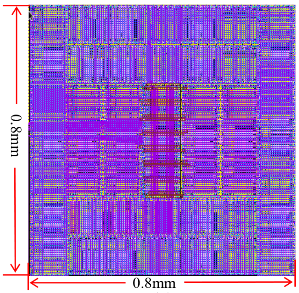
Fig.7. Working flow of the proposed image recognition system.

C. *Working flow of image recognition system*

Finite state machines (FSM) of image recognition system with a set of status signals is assigned to two-fixed point bits to represent IDLE state ("00"), COMPUTE state ("01"), and UPDATE state ("10"). IDLE state means the system is idle and is waiting for order. COMPUTE state means the system is computing a layer of DNN. UPDATE state means the system is updating parameters which are needed for computing a layer of DNN. The system receives signal "start" with FSM changing to CALCULATE state, and loads input feature. Input feature flows into PE array and becomes output feature, which is stored into GDB after pruning. When computing finish, the signal layer_over is set to 1 and FSM changes to UPDATE state. Inside of FSM, there a several register groups preserving parameters of current DNN layer, as shown in Fig. 7. l_cnt register group records the number of layers that has been computed. LAYER register group records the number of current DNN layers. IC register group records the input channel size of each layer. OC register group records the output channel size of each layer. These register groups are updated during UPDATE state. After updating, if the value stored in l_cnt register group is still smaller than that in LAYER, FSM turns to COMPUTE state. If the value stored in l_cnt register group equals to LAYER, output feature flows into predictor and the system outputs the predict label.

IV. Performance Evaluation and Analysis

Digital-analog hybrid simulations are implemented by using 14 nm CMOS process technology and the process design kit of MTJ. Fig. 8(a) shows the layout of the image recognition system based on FRM-CIM architecture, which can achieve the performance results closer to the chip. Fig. 8(b)



Parameter	Value
MTJ area	40 nm x 40 nm x π/4
Oxide barrier height of MTJ	0.85 nm
free layer height of MTJ	1.3 nm
Temperature	300 K
Nominal R_{MTJ} at R_L (R_H) of MTJ	3.98 KΩ (8.75 KΩ)
Critical switching current R_H - R_L of MTJ	28 μA
Critical switching current R_L - R_H of MTJ	72 μA
TMR	120%

(a) (b)

Fig. 8 (a) Layout of the proposed image recognition system based on FRM-CIM architecture. (b) Key parameters of the MTJ device.

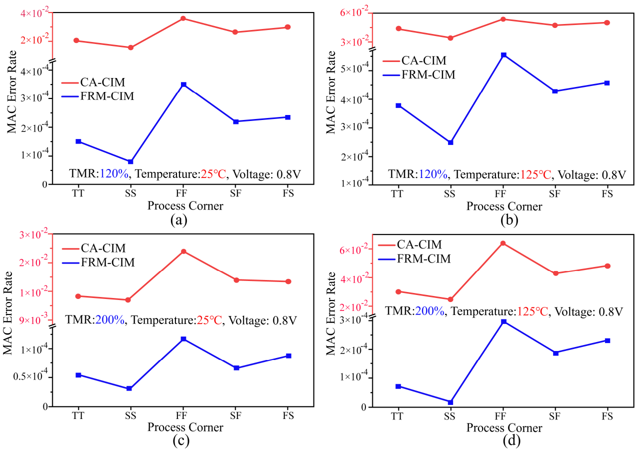


Fig.9. MER of the proposed FRM-CIM architecture under different process corner. (a) TMR of 120%, 25°C, 0.8V. (b) TMR of 120%, 125°C, 0.8V. (c) TMR of 200%, 25°C, 0.8V. (d) TMR of 200%, 125°C, 0.8V.

summarizes the key parameters of the MTJ device, which are dependent on physical models and experimental measurements [8].

As reliability is crucial for implementing computing, we first analyze the reliability of the proposed FRM-CIM architecture. In FRM-CIM architecture, the register-accumulate structure is composed of logic gates, which has high reliability. Therefore, the main factor that affects the reliability of the MAC operation comes from the VFRC circuit. Moreover, every MTJ is slightly different in its physical properties due to imperfections in the fabrication processes, as are every CMOS. We carry out the Monte Carlo simulations of 10^5 samples for the proposed FRM-CIM architecture to evaluate its reliability based on the parameter of MAC error rate (MER). Note that the process deviation of the MTJ resistance follows a Gaussian distribution with 5% variability [9]. Results show that the proposed FRM-CIM architecture reduces the error rate of MAC operation by two orders of magnitude compared with the conventional analog CIM (CA-CIM) architecture, as shown in Fig.9. Besides, increasing the TMR can increase the difference between high resistance state and low resistance state of MTJ, which improves the read operation accuracy of the proposed VFRC circuit, thereby further reducing the MER value. Obviously, the MAC operation accuracy of the FRM-CIM architecture is comparable to that of the digital circuit.

Fig.10 shows the delay and energy of the FRM-CIM architecture when performing 8-bit and 16-bit MAC operation under different supply voltage, where the MAC operation is divided into four parts: row and column decoding operation, read operation of VFRC, register-accumulate operation as well as shift and addition (Shift & Add) operation. Obviously, the delay and energy of register-accumulate operation are the largest, which is caused by the following two reasons: 1) the register-accumulate circuit contains a large number of DFFs and FAs to implement MAC operations. 2) The DFFs needed to be powered on all the time during the computing to store the results. Therefore, an effective method to reduce energy consumption in FRM-CIM architecture is to use the low-power DFF proposed by various literatures [10]. Besides, it can also be observed that the delay and energy of decoding operation in 8-bit MAC operation are almost the same as those in 16-bit, which benefits from the full precision data input to enable the same number of decoding operations to be performed when the number of accumulations is the same. Meanwhile, Fig.9 demonstrates that the difference in MAC operation delay under different input and weight precisions mainly comes from the Shift & Add operations, i.e., the delay of performing the weighted addition for the single-bit MAC results generated by each column. Therefore, the proposed FRM-CIM architecture eliminates the correlation between delay and precision of MAC operation in the analog CIM.

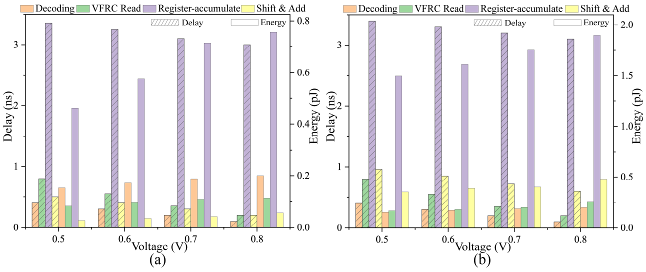


Fig.10. Delay and energy of FRM-CIM architecture when performing MAC operation under different supply voltage. (a) Input and weight are both 8-bit precision. (b) Input and weight are both 16-bit precision.

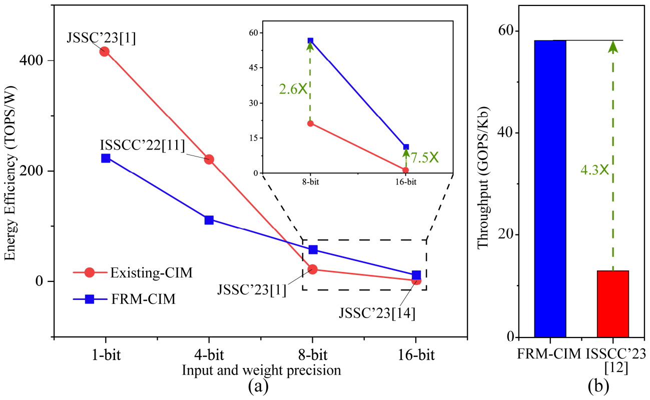
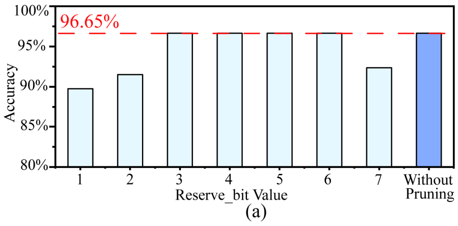


Fig.11. Performance of the FRM-CIM architecture performing MAC operation. (a) Energy efficiency under different input and weight precision. (b) Normalized to throughput per Kb

Fig.11 shows the energy efficiency and throughput of the FRM-CIM architecture performing MAC operation. For 1-bit and 4-bit input and weight precision, the energy efficiency of the FRM-CIM architecture is lower than the existing CIM architectures. The reason for this problem is that the FRM-CIM architecture needs to perform a large number of addition and register operations by using logic gates (i.e., register-accumulate circuit). However, with the improvement of input and weight precision, the number of cycles in analog CIM architecture increases exponentially, resulting in its energy being much greater than that of the register-accumulate circuit. Therefore, it can be observed in Fig.11 (a) that the energy efficiency FRM-CIM architecture at 8-bit and 16-bit is increased by 2.6 times and 7.5 times, respectively, compared with the state-of-the-art works. Besides, compared with the [12], the throughput of FRM-CIM architecture is improved by 4.3 times due to its smaller delay, as shown in Fig.11 (b).

In the designed image recognition system based on the FRM-MAC architecture, an adaptive pruning method is



	MNIST	CIFAR -10
Accuracy	96.65%	82.7%
Execution Time	12μs	42μs
Power	38mW	112mW

(b)

Fig.12. Performance of the image recognition system based on the FRM-MAC architecture (a) Accuracy for MNIST dataset. (b) Accuracy, time and power for executing DNN to identify MNIST and CIFAR-10 datasets.

TABLE II COMPARISON WITH PREVIOUS WORKS

		This work		ISSCC'23[13]	JSSC'23[1]	ISSCC'23[7]	ISSCC'23[12]	JSSC'23[14]		ISSCC'23[15]
CMOS Technology		**14nm**		22nm	22nm	28nm	4nm	28nm		22nm
Memory Type		**STT-MRAM**		STT-MRAM	ReRAM	SRAM	SRAM	SRAM		SRAM
Capacity		**2Mb**		8Mb	8Mb	4Kb	54Kb	16Kb		13Kb
Supply Voltage (V)		**0.5~0.8**		0.8	0.8	0.6~0.9	0.32~1	0.7/0.8		0.72~0.82
CIM mode		**Full-Digital**		Analog	Time-Domain	Analog	Digital	Digital		Analog
Bit Precision (bit)	Input	**8**	**16**	8	8	8	8	8	16	8
	Weight	**8**	**16**	8	8	8	8	8	16	8
	Output	**24**	**36**	26	19	20	24	NA	NA	24
Access Time (ns)[1]		**3.5**	**4**	10.56	14.4	6	6	180	340	NA
Throughput (TOPS/mm^2)		**58.51**	**51.2**	0.05	0.387	1.125	13.7	0.27	0.068	1.4
Energy Efficiency (TOPS/W)		**56.72**	**11.3**	46.4[2]	21.6	33.44	41.3	4.55	1.5	21.38

[1]Delay of the MAC operation; [2]50% input sparsity.

adapted to crop the output data of the hidden layer to a low precision data determined by the parameter reserve_bit value, which leads to the decrease in image recognition accuracy. If the reserve_bit value is small, most bits of data are reserved, which causes the data overflow during subsequent computations. Meanwhile, if the reserve_bit value is big, only little bits of data are reserved, which causes data information loss. Fig.12 provides a detailed analysis of the reserve_bit value on image recognition accuracy for MNIST dataset. It can be observed that the accuracy loss is large when the reserve_bit value is 1 and 7. When it is 3, 4, 5 and 6, there is no accuracy loss compared with the without pruning. Table II compares the proposed FRM-CIM architecture with state-of-the-art CIM architectures published in the recent years. Compared with [13] and [1] that also build based on the non-volatile memory, the access time of performing MAC operation in the FRM-CIM architecture has been greatly decreased. Besides, the energy efficiency of 16-bit input and weight MAC operation in FRM-CIM architecture is up to 11.3 TOPS/W, which is a great advantage compared with SRAM-based CIM [14] known for its high energy efficiency.

V. CONCLUSION

This work proposes an FRM-CIM architecture to perform fast and energy-efficient execution of the MAC operation by utilizing segmented BL array in STT-MRAM. First, the FRM operation based on recursive execution of addition and register operations is presented to realize full precision input of data, which efficiency eliminates the correlation between input precision and delay, thereby greatly reducing the delay of high precision MAC operation in the CIM architecture. Meanwhile, based on the FRM-CIM architecture, we further construct an image recognition system to perform image recognition for the MNIST and CIFAR-10 datasets by using DNN algorithm, where an adaptive pruning method for each hidden layer is proposed to improve the compatibility of the RM-CIM architecture with DNN. Finally, results in layout level show that the FRM-CIM architecture achieves 56.72 TOPS/W and 11.3 TOPS/W for 8-bit and 16-bit MAC operations, respectively, outperforming existing CIM architectures. In summary, this work has significance for further research on MRAM to realize high throughput and energy-efficient neural network platform.

ACKNOWLEDGEMENT

This work was supported in part by the National Natural Science Foundation of China under Grant 61971024, 62122008, and 51901008, and in part by the Project funded by China Postdoctoral Science Foundation 2022M720344.

REFERENCES

[1] J. M. Hung et al., "8-b Precision 8-Mb ReRAM Compute-in-Memory Macro Using Direct-Current-Free Time-Domain Readout Scheme for AI Edge Devices," *IEEE Journal of Solid-State Circuits*, vol. 58, no. 1, pp. 303-315, Jan. 2023.

[2] C. X. Xue et al., "A 22nm 4Mb 8b-Precision ReRAM Computing-in-Memory Macro with 11.91 to 195.7TOPS/W for Tiny AI Edge Devices," *IEEE International Solid-State Circuits Conference*, San Francisco, CA, USA, 2021.

[3] H. Wang et al., "A Charge Domain SRAM Compute-in-Memory Macro With C-2C Ladder-Based 8-Bit MAC Unit in 22-nm FinFET Process for Edge Inference," *IEEE Journal of Solid-State Circuits*, vol. 58, no. 4, pp. 1037-1050, April 2023.

[4] Y. Zhang et al., "Time-Domain Computing in Memory Using Spintronics for Energy-Efficient Convolutional Neural Network," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 3, pp. 1193-1205, March 2021.

[5] S. Jung et al. "A crossbar array of magnetoresistive memory devices for in-memory computing," *Nature*, vol. 601, pp. 211-216, Jan. 2022.

[6] J. Wang et al., "TAM: A Computing in Memory based on Tandem Array within STT-MRAM for Energy-Efficient Analog MAC Operation," *Design, Automation & Test in Europe Conference & Exhibition*, Antwerp, Belgium, 2023

[7] B. Wang et al., "A 28nm Horizontal-Weight-Shift and Vertical-feature-Shift-Based Separate-WL 6T-SRAM Computation-in-Memory Unit-Macro for Edge Depthwise Neural-Networks," *IEEE International Solid-State Circuits Conference*, San Francisco, CA, USA, 2023.

[8] Y. Zhang et al., "Compact modeling of perpendicular-anisotropy CoFeB/MgO magnetic tunnel junctions," *IEEE Transactions on Electron Devices*, vol. 59, no. 3, pp. 819–826, Mar. 2012.

[9] J. K. Wang et al., "A self-matching complementary-reference sensing scheme for high-speed and reliable toggle spin torque MRAM," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 12, pp. 4247–4258, Dec. 2020.

[10] G. Scotti et al., "Design of Low-Voltage High-Speed CML D-Latches in Nanometer CMOS Technologies," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 12, pp. 3509-3520, Dec. 2017.

[11] H. Fujiwara et al., "A 5-nm 254-TOPS/W 221-TOPS/mm2 Fully-Digital Computing-in-Memory Macro Supporting Wide-Range Dynamic-Voltage-Frequency Scaling and Simultaneous MAC and Write Operations," *IEEE International Solid-State Circuits Conference, San Francisco*, CA, USA, 2022.

[12] H. Mori et al., "A 4nm 6163-TOPS/W/b 4790-TOPS/mm^2/b SRAM Based Digital-Computing-in-Memory Macro Supporting Bit-Width Flexibility and Simultaneous MAC and Weight Update," *IEEE International Solid-State Circuits Conference, San Francisco*, CA, USA, 2023.

[13] Y. C. Chiu et al., "A 22nm 8Mb STT-MRAM Near-Memory-Computing Macro with 8b-Precision and 46.4-160.1TOPS/W for Edge-AI Devices," *IEEE International Solid-State Circuits Conference*, San Francisco, CA, USA, 2023.

[14] C. Y. Yao, T. Y. Wu, H. C. Liang, Y. K. Chen and T. T. Liu, "A Fully Bit-Flexible Computation in Memory Macro Using Multi-Functional Computing Bit Cell and Embedded Input Sparsity Sensing," IEEE Journal of Solid-State Circuits, vol. 58, no. 5, pp. 1487-1495, May 2023.

[15] P. Chen et al., "A 22nm Delta-Sigma Computing-In-Memory ($\Delta\sum$CIM) SRAM Macro with Near-Zero-Mean Outputs and LSB-First ADCs Achieving 21.38TOPS/W for 8b-MAC Edge AI Processing," I*EEE International Solid-State Circuits Conference*, San Francisco, CA, USA, 2023.