Extended Abstracts of the 2018 International Conference on Solid State Devices and Materials, Tokyo, 2018, pp109-110

MTJ-Based Nonvolatile Logic Gate for Binarized Convolutional Neural Networks and Its Impact

Masanori Natsui, Tomoki Chiba, and Takahiro Hanyu
[1] Research Institute of Electrical Communication, Tohoku University
2-1-1 Katahira, Aoba-ku, Sendai 980-8577, JAPAN, Phone: +81-22-217-5552 E-mail: natsui@riec.tohoku.ac.jp

Abstract: An MTJ-based nonvolatile logic-in-memory (NV-LIM) architecture for binarized neural networks (BNNs) is proposed. NV-LIM-based BNN has a capability of reducing both computational cost and data transfer cost related to the inference function of deep neural networks. Through an experimental evaluation of a basic component of BNN hardware designed with NV-LIM architecture, we demonstrate its impact on the power, delay and area overhead reduction.

1. Introduction

Binarization [1] is attracting attention to adopt brain-inspired computing for applications with limited hardware cost, such as IoT sensor nodes and mobile devices. This is to realize highly efficient hardware implementation by replacing the operation conventionally performed using fixed- or floating representation of several bits in convolutional neural networks (CNNs) to binary operation. Various papers have reported that sufficient performance can be obtained even if the operations are binarized.

We have shown the effectiveness of MTJ-based nonvolatile logic-in-memory (NV-LIM) architecture, which merges MTJ devices [2] as nonvolatile memories (NVMs) with a logic circuit by exploiting their three-dimensional stackability and CMOS compatibility [3,4]. In this paper, we describe the application of NV-LIM gates for the implementation of compact, low power, and high performance binarized neural networks (BNNs). Through an experimental evaluation of an NV-LIM gate designed using a 40-nm CMOS process technology, we consider the impact of NV-LIM architecture on the hardware implementation of BNNs.

2. Binarized Neural Networks

Figure 1 shows one of the well-known CNN structure called LeNet [5], as well as the difference between convolution operations in a conventional neural network and binarized one. The structure of CNN can be roughly divided into convolution layers and full-connection layers, where multiply-and-accumulate (MAC) operations occupy a large part of the calculation contents in any of the layers. In BNN, these operations can be replaced by exclusive NOR (XNOR) operations and bitcount operations, which is the basic principle of reduction of computational cost by binarization.

Figure 2 shows a comparison of the hardware structure of the MAC operation in several architectures. Multipliers and adders can be replaced by smaller logic/arithmetic circuits by binarization, which greatly reduces the hardware cost. In addition, by holding input values corresponding to the weight coefficients of a neural network in NVMs connected to the inputs of logic gates, it is possible to reduce the power and delay associated with the memory access. Furthermore, by fusing a logic gate (XNOR gate) and an NVM in NV-LIM style, it is possible to improve its performance further as well as reduce area and power overhead.

Table 1 shows how the operation counts in LeNet change depending on the network structure. Here we assume that the input values and the weight coefficients are represented by 8-bit fixed representation in a conventional configuration. It is necessary to perform an extremely large number of 8x8-bit multiplications and 8-bit additions in the conventional configuration, whereas most of them are replaced with 1-bit 2-input XNOR operations and bitcount operations by binarization. In addition, by applying NV-LIM technology, the data transfer from memory can also be eliminated. This evaluation suggests that the XNOR operation is one of the most important operations in BNN, and its implementation with highly efficient hardware is expected to greatly contribute to the improvement of the overall performance of BNN hardware. From this viewpoint, we introduce NV-LIM XNOR gate as a candidate for this purpose in the next section.

3. MTJ-based NV-LIM XNOR Gate and Its Evaluation

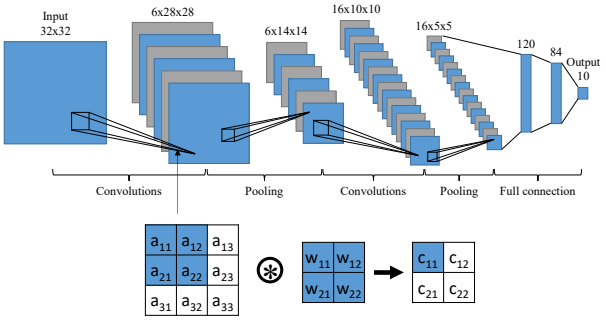
Figure 3 shows a circuit diagram of the NV-LIM XNOR gate, consisting of PCSA [6], XNOR logic tree, and MTJ devices as a 1-bit memory. Two MTJ devices take a complementary state and hold the logical value 0 or 1 in a nonvolatile manner. By performing dynamic operation taking two phases of pre-charge and evaluate, high-speed logic operation and low power consumption can be simultaneously realized.

We designed this circuit by using a design flow developed by our research group with a hybrid process of 40-nm CMOS and MTJ devices. The resistance values of the MTJ devices were set to R_P = 8 [kΩ] and R_{AP} = 16 [kΩ]. Figure 4 shows the simulated waveform of this circuit. We can confirm that both XNOR operations with the stored value in the MTJ device and the input value, and a write operation to the MTJ devices are performed correctly.

Table 2 shows the comparison of the power, delay, and area of the three types of configurations: a multiplier used in the conventional CNN, a configuration in which an XNOR gate and a memory are separately arranged, and an NV-LIM configuration. By replacing the multiplier with an XNOR gate by binarization, it is possible to reduce power, delay, and area greatly, and it can also be confirmed that the NV-LIM configuration further improves the performance. This result suggests that high speed and low power neural network hardware can be compactly implemented by adopting NV-LIM XNOR as a basic component of BNN.

4. Conclusion

In this paper, we examined the effectiveness of NV-LIM gate for BNN through a performance evaluation of one of the main components, XNOR gate. As a prospect, we will consider the effectiveness of NV-LIM in other components such as bitcount, and evaluate the overall impact to BNN.



Conventional:

$c_{11} = a_{11} \times w_{11} + a_{12} \times w_{12} + a_{21} \times w_{21} + a_{22} \times w_{22}$

a_{ij}, w_{ij}: Fixed-point integers

$\times$: Multiplication with fixed-point integers

$+$: Addition with fixed-point integers

Binarized:

c_{11} = bitcount ($a_{11}^* \odot w_{11}^*, a_{12}^* \odot w_{12}^*, a_{21}^* \odot w_{21}^*, a_{22}^* \odot w_{22}^*$) $\times$ K

a_{ij}^*= sign (a_{ij}), w_{ij}^*= sign (w_{ij})

K: Coefficient

$\odot$: Exclusive-NOR with two binaries

Fig. 1 Typical structure of CNN and the difference between convolution operations.

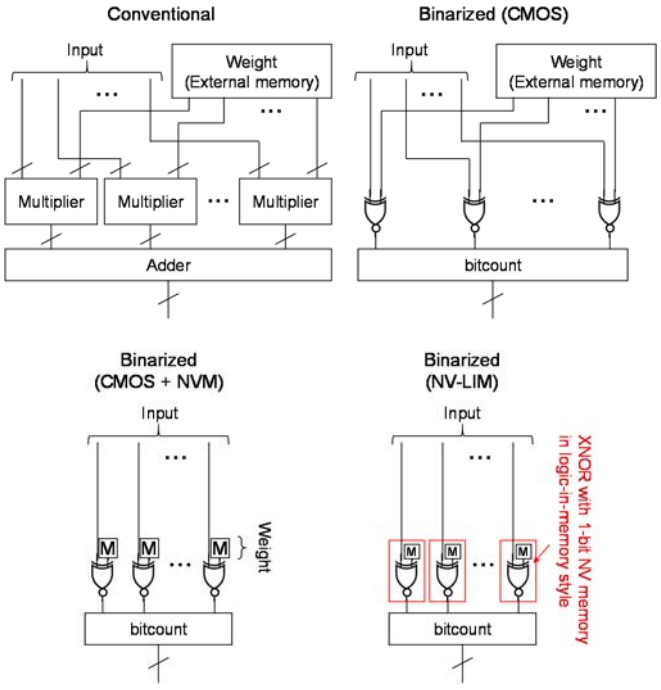


Fig. 2 Comparison of the hardware structure of the MAC operation in several architectures.

Acknowledgments

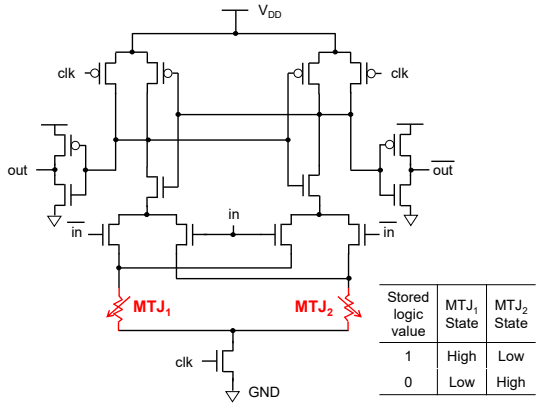
The authors thank Y. Takako of Focal Agency for excellent technical assistance. Part of this research was supported by the JSPS IMPACT Program, Brainware LSI Project by MEXT, JST OPERA, and JSPS KAKENHI Grant Number 16KT0187.

References

[1] M. Coubariaux, et al., Advanec in Neural Information Processing Systems, 3123 (2015).
[2] S. Ikeda, et al., Nat. Mater. 9, 721 (2010).
[3] M. Natsui, et al., 2013 IEEE ISSCC, 194 (2013).
[4] M. Natsui, et al., IEEE JSSC 50, 476 (2015).
[5] Y. LeCun, et al., Proc. IEEE, 86(11), 2278 (1998).
[6] W. Zhao, IEEE Trans. Mag. 45(10), 3784 (2009).

Table. 1 Operations counts on each network structure.

Convolution	CNN	BNN	BNN+NV-LIM
8x8-bit mult.	328365	18948 (-94.2%)	**18948**
8-bit add	320520	4800 (-98.5%)	**4800**
8-bit data trans.	326640	0 (-100.0%)	**0**
1-bit data trans.	0	326640	**0 (-100.0%)**
XNOR	0	326640	**326640**
bitcount (25bit)	0	12624	**12624**
Full-connection	**CNN**	**BNN**	**BNN+NV-LIM**
8x8-bit mult.	10920	94 (-99.1%)	**94**
8-bit add	10826	0 (-100.0%)	**0**
8-bit data trans.	10920	0 (-100.0%)	**0**
1-bit data trans.	0	10920	**0 (-100.0%)**
XNOR	0	10920	**10920**
bitcount (84 bit)	0	84	**84**
bitcount (120 bit)	0	10	**10**



Stored logic value	MTJ_1 State	MTJ_2 State
1	High	Low
0	Low	High

Fig. 3 MTJ-based NV-LIM XNOR gate.

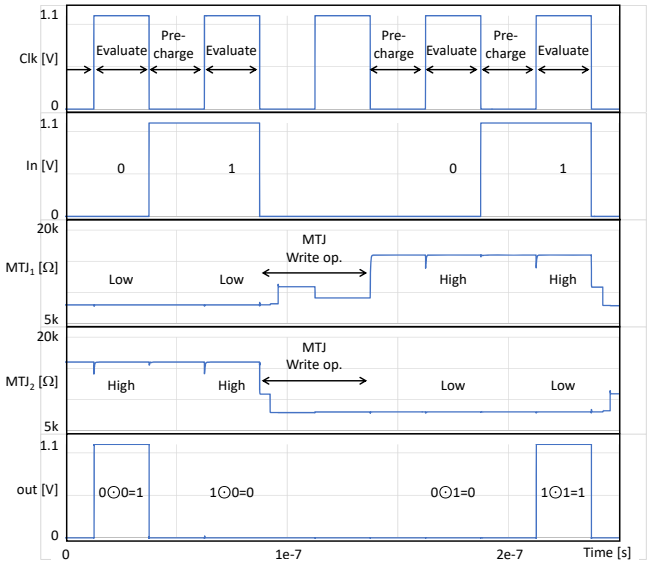


Fig. 4 Simulated waveform.

Table. 2 Performance comparison.

	Power [W]	Delay [s]	PDP [W·s]	Area [a.u.]††
8x8 Multiplier†	1.57×10^{-5}	4.29×10^{-10}	6.74×10^{-14}	8259.7
CMOS XNOR + NVM	2.88×10^{-7}	1.70×10^{-10}	4.88×10^{-17}	127.5
NV-LIM	$\mathbf{2.15 \times 10^{-7}}$	$\mathbf{1.69 \times 10^{-10}}$	$\mathbf{3.64 \times 10^{-17}}$	**86.5**

† Power/delay cost related to the data transfer is not included.
†† The size of a transistor with minimum width/length is set to 1.