An SRAM-Based Hybrid Computation-in-Memory Macro Using Current-Reused Differential CCO

Injun Choi, *Graduate Student Member, IEEE*, Edward Jongyoon Choi, *Graduate Student Member, IEEE*, Donghyeon Yi, *Student Member, IEEE*, Yoontae Jung, *Student Member, IEEE*, Hoyong Seong, *Student Member, IEEE*, Hyuntak Jeon, *Member, IEEE*, Soon-Jae Kweon, *Member, IEEE*, Ik-Joon Chang, *Member, IEEE*, Sohmyung Ha, *Senior Member, IEEE*, and Minkyu Je, *Senior Member, IEEE*

***Abstract*— This work presents a 4 kb 8T-SRAM computation-in-memory (CIM) macro based on hybrid computation using digital in-memory-array computing (DIMAC) and phase-domain near-memory-array computing (PNMAC). By employing multiple local dual-column arrays (LDCAs), bit-wise multiplications are computed digitally in memory with high energy efficiency and throughput. The PNMAC performs the summation and accumulation in parallel with a high dynamic range by using a proposed steering-DAC-based differential current-controlled-oscillator (DCCO). After the phase-domain accumulation is completed, only a one-time digital conversion needs to be performed using a phase quantizer with negligible phase-to-digital conversion overhead. Moreover, by effectively reusing the steered current to accumulate the multiplication results fed from the DIMAC, the power consumption of the PNMAC can be greatly reduced. The macro fabricated in a 65 nm process achieves 22.4TOPS/W peak energy efficiency and 19.03 μW power consumption with a 59.8% zero-skipping rate, which is 96.05× lower than state of the art.**

***Index Terms*— Convolutional neural network (CNN), SRAM, computation in memory (CIM), digital in-memory-array computing (DIMAC), phase-domain near-memory-array computing (PNMAC), differential current-controlled-oscillator (DCCO).**

I. Introduction

RECENTLY, the demand for energy-efficient convolutional neural network (CNN) engines has been increasing as neural networks (NNs) are becoming widely used in edge artificial intelligence (AI) applications [1]. The CNN engines are typically required to perform an enormous number of multiply-and-accumulate (MAC) operations, which constitute a significant portion of the inference operations of the CNN. When performing the MAC operations, the CNN engines consume massive computational resources and high power. The energy efficiency is also relatively low on traditional CNN engines [2], [3] based on the Von Neumann architecture due to data movement between computing elements and memory, known as the memory wall problem [4]. Because of the limited computing resource and power budget, it is difficult to use traditional CNN engines in edge devices.

To overcome the memory wall problem, computation in memory (CIM) has been drawing huge attention as a promising solution by minimizing data movement between the memory system and computation units. The CIM architecture processes the computations *in* the memory system without moving data. For most NN-based AI applications, this memory-centric architecture saves energy consumption by reducing the data movement energy which is dominant in the traditional Von Neumann architecture. Moreover, CIM architectures can achieve high energy efficiency by enabling parallel data processing and performing multiple computing operations within memory modules in a single cycle [5].

CIM can be implemented using static random access memory (SRAM) or nonvolatile memory. Since nonvolatile-memory-based CIM (NVM-CIM) can keep the data even when powered off, the stored weight data does not require power to be maintained and does not need to be reloaded when the system is turned on. However, the cost of the write operation is much higher in nonvolatile memory, so the NVM-CIM is one approach to overcome the memory wall problem in the applications without the need for frequent data updating. By contrast, SRAM-based CIM (SRAM-CIM) can keep the data only during the time it is powered up, but it provides faster write speeds and lower write energy than the NVM-CIM. Furthermore, the SRAM-CIM shows good compatibility with state-of-the-art CMOS logic technology, so it is easily scaled down to reduce latency and increase energy efficiency. Due to these features, the SRAM-CIMs are considered one of the most promising options in most edge AI applications.

Manuscript received December 19, 2021; revised February 24, 2022; accepted March 14, 2022. Date of publication April 26, 2022; date of current version June 13, 2022. This work was supported in part by the Institute of Information and Communications Technology (IITP) funded by the Ministry of Science and Information and Communications Technology (MSIT), Korea, under Grant 2021-0-01764, and in part by the Integrated Circuit Design Education Center (IDEC), Korea. This article was recommended by Guest Editor T. Kim *(Corresponding authors: Ik-Joon Chang; Sohmyung Ha; Minkyu Je.)*

Injun Choi, Edward Jongyoon Choi, Donghyeon Yi, Yoontae Jung, Hoyong Seong, and Minkyu Je are with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 34141, South Korea (e-mail: injunchoi@kaist.ac.kr; mkje@kaist.ac.kr).

Hyuntak Jeon is with the Agency for Defense Development (ADD), Daejeon 34060, South Korea.

Soon-Jae Kweon is with the Division of Engineering, New York University Abu Dhabi, Abu Dhabi 129188, United Arab Emirates.

Ik-Joon Chang is with the Department of Electronics, Kyung Hee University, Yongin-si 17104, South Korea (e-mail: ichang@khu.ac.kr).

Sohmyung Ha is with the Division of Engineering, New York University Abu Dhabi, Abu Dhabi 129188, United Arab Emirates, and also with the Tandon School of Engineering, New York University, New York, NY 10003 USA (e-mail: sohmyung@nyu.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JETCAS.2022.3170595.

Digital Object Identifier 10.1109/JETCAS.2022.3170595

Many SRAM-CIMs perform MAC operations in the analog domain to achieve high energy efficiency and throughput. These benefits come from accessing multiple memory cells simultaneously and employing analog computational techniques, which can reuse the energy used to read the data for computing. However, the analog SRAM-CIMs have limitations due to the accuracy degradation by a low signal-to-noise ratio (SNR) of the analog computation results and the overhead in area and energy of the analog-to-digital converter (ADC) circuits. Also, the ADC often operates at a high sampling rate because the narrow dynamic range of the ADC limits the number of input elements of MAC operations. Hence, robust in-memory-array computing (IMAC) techniques and lightweight ADC design methods are primary considerations and challenges.

In this paper, we present an SRAM-CIM based on hybrid computation using digital IMAC (DIMAC) and phase-domain near-memory-array computing (PNMAC). To improve both robustness and accuracy, bit-wise multiplications are computed using DIMAC with high energy efficiency and throughput. By using PNMAC, we implement a lightweight ADC to minimize power overhead and obtain a wide dynamic range. Thanks to the wide dynamic range of the proposed PNMAC, it converts analog information to digital data only once at the end of MAC operation with ultra-low power consumption. In addition, the current-controlled oscillator (CCO), the main core of PNMAC, consists of an 11-stage ring oscillator, which occupies a small area.

The remainder of this article is organized as follows. Section II describes the background and motivation. Section III presents the proposed 8T-SRAM-based CIM macro based on hybrid computation. Section IV discusses the measurement results of the proposed design, and finally, the conclusions are drawn in Section V.

II. Background

Major computational techniques and state-of-the-art SRAM-based computing structures are reviewed in this section. First, we describe various computing techniques for MAC operation. Then, we present three main types of SRAM-based computing structures, and the design challenges are discussed for further development.

A. Computing Techniques for MAC Operation

Various SRAM-based computing techniques have been proposed for energy-efficient MAC operations by enabling parallel data processing within the memory macro [6]–[15]. In general, analog-domain computing is one of the key design choices to improve energy efficiency significantly. In [6], the current-domain computing is first proposed for an adaptive boosting machine learning classifier using a 6T SRAM cell array. This design computes 4b-input and 1b-weight MAC values with high energy efficiency. In [7], the current-domain computing based on twin 8T SRAM cell arrays is proposed to prevent write disturbances and improve computing accuracy. Multi-bit inputs are modulated to multi-level word-line (WL) voltages, which are used to generate weighted memory cell currents. [8] achieves a high computational SNR by using charge-domain computation based on metal-oxide-metal (MOM) capacitors. In this case, its high energy efficiency and throughput are achieved by charge accumulation, and the robustness is ensured by relying on the capacitance values that have much smaller variations than the transistor parameters [9]. These two analog computing techniques in the current and charge domains can perform energy-efficient MAC operations, but its MAC results should be digitized by ADCs, which dominate the area and power costs of CIM.

To reduce the overhead of ADCs, time- and phase-domain computing techniques have been proposed in [16]–[18]. In [16], a phase-domain MAC circuit is implemented using a gated-ring oscillator (GRO), which serves as its accumulation core. The partial MAC values are continuously accumulated in the GRO, and a readout logic samples the phase once at the end of MAC operation. In [17], a time-domain MAC computing is proposed to implement a CNN engine without any capacitors or ADCs. The design in [18] employs CCO-based ADCs, which are amenable to trading off the precision with the latency. However, they are implemented using a single-stage oscillator, which cannot produce multiple-phase information. These time- and phase-domain approaches produce high-precision digital outputs while minimizing the area and power consumption in general. However, they suffer from low throughput because the inputs need to be serially fed to the MAC operation circuit. To address this issue, the MAC rate is increased to a high frequency in [16], adversely increasing the power consumption of the entire system. In [18], a wide-range and high-frequency generator, which consumes hundreds of μA current and needs an additional feed-forward compensation circuit, is proposed to achieve high throughput.

Some SRAM-based computing schemes [14], [15] are implemented in the digital domain for high accuracy rather than in the analog domain, where the SNR and accuracy are fundamentally limited. [14] proposes a zero-skipping convolution SRAM to perform energy-efficient in-memory operations and a charge reuse scheme to further reduce the energy consumption of in-memory operations. These all-digital approaches can implement SRAM-based CNN engines without any accuracy loss in MAC operations. However, all-digital full-precision computing is hard to achieve as high energy efficiency as its analog counterpart operating at low-to-medium computing precision.

In summary, these computing techniques have trade-offs among computation energy, dynamic range, read accuracy, and area efficiency. The current- and charge-domain computing approaches achieve high energy efficiency and throughput at the cost of data conversion. The phase-domain computing can perform ultra-low-power computing with a wide dynamic range by exploiting the recursive nature of the GRO. It also achieves a better area efficiency but results in slower throughput due to its sequential operation. Digital domain computing is robust to noise and process variation but has limitations in energy efficiency.

In this paper, we employ phase- and digital-domain computing to minimize the power overhead of ADCs and improve the robustness of IMAC. The issues of two employed computing

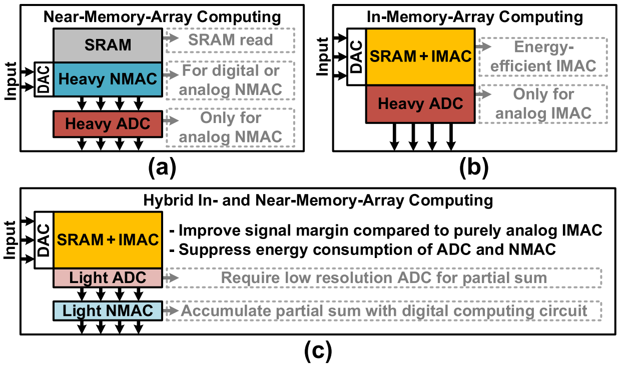

Near-Memory-Array Computing
SRAM
SRAM read
Input
DAC
Heavy NMAC
For digital or analog NMAC
Heavy ADC
Only for analog NMAC
(a)
In-Memory-Array Computing
Input
DAC
SRAM + IMAC
Energy-efficient IMAC
Heavy ADC
Only for analog IMAC
(b)
Hybrid In- and Near-Memory-Array Computing
Input
DAC
SRAM + IMAC
- Improve signal margin compared to purely analog IMAC
- Suppress energy consumption of ADC and NMAC
Light ADC
Require low resolution ADC for partial sum
Light NMAC
Accumulate partial sum with digital computing circuit
(c)

Fig. 1. State-of-the-art SRAM-based computing structures: (a) near-memory-array computing, (b) in-memory-array computing, and (c) hybrid in- and near-memory-array computing.

schemes are addressed by taking the following approaches: 1) the low-throughput problem of phase-domain computing is solved by the proposed PNMAC that performs the summation and accumulation in parallel, and 2) the energy efficiency limitation of digital-domain computing is mitigated by the bit-serial digital computing that reduces the complexity and energy consumption of the in-memory multiplication.

B. SRAM-Based Computing Structures

Fig. 1(a) shows a general near-memory-array computing (NMAC) structure using an SRAM as a typical memory and performing the computation much easier near the memory. Here either analog or digital computing can be employed for energy-efficient operation. This approach can use a conventional SRAM directly and be implemented by connecting the SRAM macro with computing blocks through a customized interface. In the case of [11], it employs multi-row READ access in the memory macro and a capacitive-charge-sharing scheme to compute MAC results. Its NMAC proves highly robust to process-voltage-temperature (PVT) variations, including spatial transistor threshold voltage variations and bit-line (BL) voltage dependence of the discharge path current. Moreover, it does not require DACs thanks to the multi-row READ scheme but needs heavy ADCs to convert the analog MAC results to digital.

An overall IMAC structure is illustrated in Fig. 1(b). IMACs read multiple data in parallel and simultaneously perform operations by applying the multi-bit inputs to the SRAM macro. IMACs often employ analog computing techniques to achieve high energy efficiency and throughput. For example, in [12], an 8T1C CIM macro referred to as C3SRAM is proposed using capacitive-coupling computing (C3) performed in the memory array. It performs fully parallel vector-matrix multiplication, resulting in high energy efficiency and throughput. However, it needs flash ADCs, which consume much power and have relatively low resolution. These IMAC structures suffer from limited output precision due to the limited analog signal margin for multi-bit MAC operations.

As shown in Fig. 1(c), hybrids of in- and near-memory-array computing structures are proposed to overcome this signal margin problem [13]. In analog IMAC structures, all MAC operations are performed in the analog domain, so the signal margin is reduced as the required output precision increases. In the hybrid structures, partial MAC operations are performed in the analog domain, and the remaining MAC operations are performed in the digital domain. This approach allows the ADC to digitize the partial MAC with a sufficient signal margin that ensures robust ADC output accuracy [13].

Using hybrid structures is a practical approach to implementing the CIM macro with high energy efficiency, flexibility, and programmability. However, the ADCs still consume a large portion of the total power. Furthermore, many more analog-to-digital conversions are required than IMAC structures because the conversion is required for each partial MAC operation. To solve this challenging problem, we propose a CIM macro based on hybrid computation that continuously accumulates the partial MAC results in the phase domain, implementing ultra-low-power and wide-dynamic-range analog-to-digital conversion.

III. Proposed SRAM-Based Hybrid Computation Using DIMAC and PNMAC

This section describes the proposed SRAM-based hybrid CIM macro using DIMAC and PNMAC. The PNMAC allows the partial MAC values to be accumulated in the phase domain with ultra-low power consumption. By employing the proposed local dual-column array (LDCA), the DIMAC performs an accurate in-memory AND operation and enables the CIM macro to conduct channel-wise zero-skipping with high energy efficiency and throughput. This section also presents the proposed parallel phase-domain bit-wise accumulation, which increases the energy efficiency and throughput of PNMAC.

A. Overall Architecture of Hybrid CIM Macro

Fig. 2(a) shows the overall architecture of the proposed hybrid CIM, which consists of two main parts: 1) DIMAC performing energy-efficient in-memory bit-wise multiplications and generating pulse-width-modulated (PWM) signals to control the other main part, PNMAC; 2) PNMAC accumulating the partial MAC results continuously and converting the accumulation result to the digital data only once at the end of MAC operation with ultra-low power consumption.

The high-level block diagram of the DIMAC is shown in Fig. 2(a). Each two-column pair of the 64×64 SRAM array are divided into 4 LDCAs, each of which performs two functions for the DIMAC. First, the bit-wise multiplication performed by the AND operation, W[m] × X[n], is computed through the 8T SRAM cell and local precharger (LPC). The global BL (GBL) shared by the 4 LDCAs reads out the multiplication result from the local BLs (LBLs) through a tri-state buffer, which is enabled by EN[k]. Second, the LDCA generates 4b multiplication data, W[m] × X[3:0], represented as a PWM pulse in the time domain. The 4b kernel weight, S, W[2:0], represented in the sign-and-magnitude (S&M) format, comes from a weight buffer. The weight data set, W_{32r+i}, is loaded to the weight buffer, BUF_i, by the weight load unit, where $r = 0, \ldots, 31$. The first weight data set, $\mathrm{W}_0[m]$, $\mathrm{W}_{32}[m]$, …, $\mathrm{W}_{992}[m]$, is serially fed to the first LDCA group from the first weight

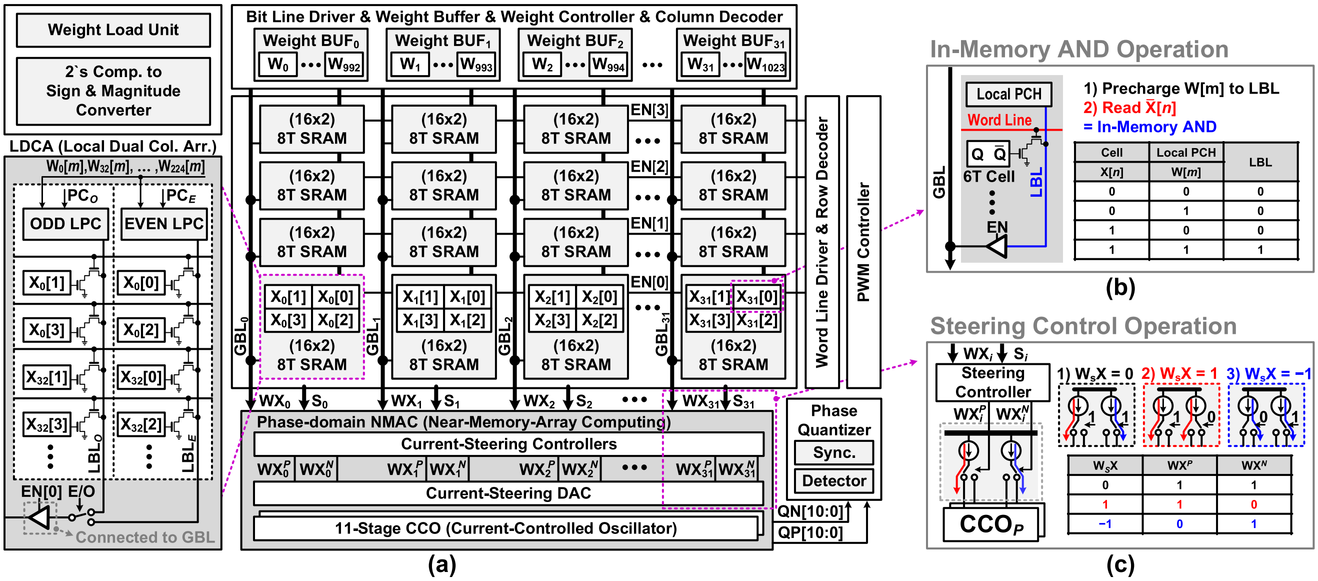


Fig. 2. (a) High-level block diagram of the proposed hybrid CIM macro with digital in-memory-array computing (DIMAC) and phase-domain near-memory-array computing (PNMAC), (b) in-memory AND operation in DIMAC, and (c) steering control operation in PNMAC.

buffer, BUF_0. We adopt zero-skipping convolution SRAM, which is composed of the 8T-SRAM cells and hierarchical BL structure [14]. It allows performing bit-wise multiplication energy-efficiently by reducing the parasitic capacitances of the BL and reusing the charge in the GBLs. As shown in Fig. 2(b), the in-memory AND operation is performed in the following two steps: 1) precharge W[m] to the LBL and 2) read $\bar{\mathrm{X}}[n]$ from the SRAM cell. Then, the AND value is turned to the LBL connected to the global buffer. Our proposed LDCA not only modulates the multiplication results into the PWM signal easily but also employs the overlapped precharging technique to achieve higher throughput compared to the conventional precharging scheme.

The PNMAC consists of current-steering controllers (CSCs), differential current source pairs (DCSPs), and DCCO. The DCCO includes a positive-side (P-side) CCO and a negative-side (N-side) CCO, which are respectively denoted as CCO_P and CCO_N. Each CCO is an 11-stage ring oscillator with a current-steering DAC included in the DCSPs. The DCSPs and DCCO accumulate the phase information controlled by the CSCs. Each GBL is connected to a dedicated CSC to generate the positive-current control signal, WX^P, and negative-current control signal, WX^N, as shown in Fig. 2(c). The DCSP converts the signed multiplication values, WX^P and WX^N, in the form of PWM voltage signals to a differential PWM current signal. Then the output current of the DCSP is translated to the phase by the DCCO. If a certain $\mathrm{W}_S\mathrm{X}$ ($= \mathrm{S} \times \mathrm{W}[m] \times \mathrm{X}[3:0]$) is applied, the DCSP provides one of the following three: 1) a zero differential current, 2) a PWM current steered to CCO_P, or 3) a PWM current steered to CCO_N. The frequencies of the two CCOs become different according to the steered current, and the frequency difference accumulates as the phase difference of the CCO pair. The PNMAC continuously accumulates the partial MAC results during the MAC operation, and the final MAC result is sampled by the phase quantizer. Unlike [18], which needs conversion from frequency domain to a digital domain for each MAC operation, we accumulate partial MAC values in a phase domain without any conversion to the digital domain until we sample the final MAC result. In addition, we achieve higher energy efficiency than [16], [17] by using the DCSPs that can receive 32 inputs in parallel.

B. Phase-Domain Near-Memory-Array Computing

One main advantage of the PNMAC is that the accumulation is performed in parallel with a wide dynamic range without any analog-to-digital conversions. It is possible because the DCCO has a wide linear input range, and the differential current can be parallelly controlled by the multiple DCSPs. Furthermore, the wide dynamic range is achieved at very low power as the oscillator core uses only several-μA-order DC current from a 0.8V supply voltage. 0.8V supply voltage is sufficient to drive the ring oscillator because there is only a minimal voltage drop caused by the current source and steering switch. In addition, the CCO-based ADC is very attractive in low-voltage conditions, as its range and resolution are not limited by the voltage rail [19].

As shown in Fig. 3, the PNMAC consists of the CSCs, DCSPs, and DCCO. The CSCs control the DCSPs in parallel based on the 4b multiplication values from the GBLs of the DIMAC and the sign bits of weights from the weight controller. The DCSPs generate P-side current, I_P, and N-side current, I_N. I_P and I_N are expressed as follows:

$$\mathrm{I}_P = 32\mathrm{I}_u + \mathrm{I}_u \sum_{i=0}^{31} (1 - 2\mathrm{S}_i)\mathrm{WX}_i, \tag{1}$$

and

$$\mathrm{I}_N = 32\mathrm{I}_u + \mathrm{I}_u \sum_{i=0}^{31} (2\mathrm{S}_i - 1)\mathrm{WX}_i, \tag{2}$$

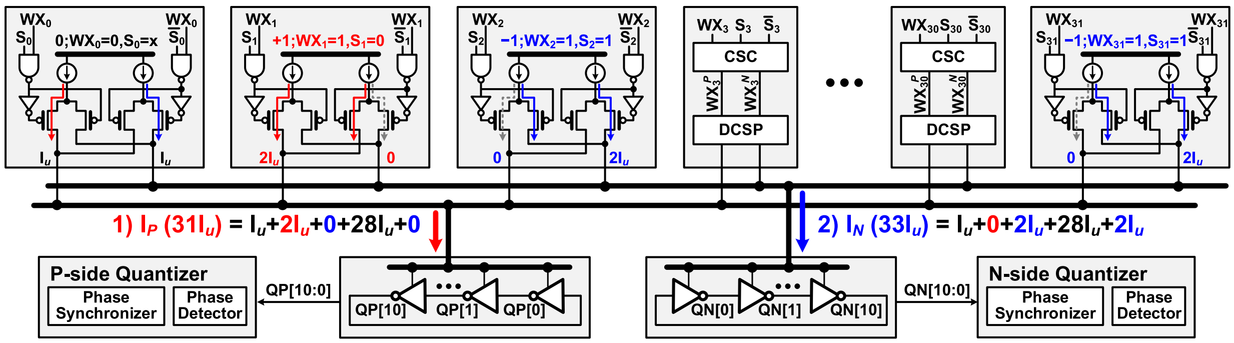


Fig. 3. 32 parallel phase-domain bit-wise accumulation for phase-domain near-memory-array computing.

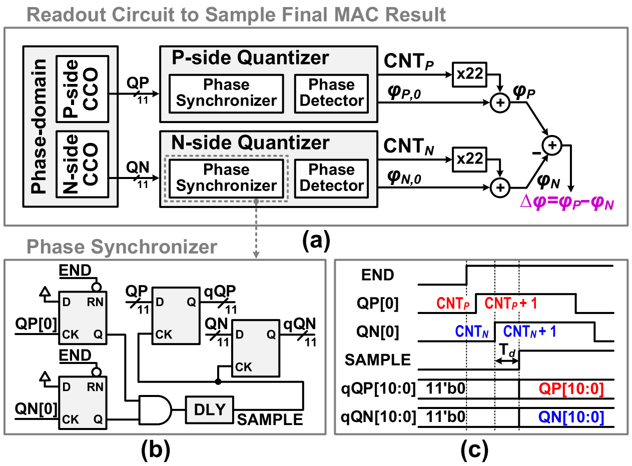


Fig. 4. (a) Readout circuit to sample the MAC results, (b) phase synchronizer, and (c) timing diagram of the readout operation.

where I_u is the unit current of DCSP, S_i is the sign bit of the weight, and WX_i is the multiplication result from the GBL of the DIMAC. We select 32 DCSPs as a group by considering the trade-off among linearity, signal margin, and throughput. The current-steering DAC can achieve higher linearity and reduce data-dependent switching noise because the group of 32 DCSPs allows most of the partial MACs to be located near zero. The local computing scheme with grouped DCSPs is more advantageous in obtaining a large signal margin and lower power consumption but at the cost of low throughput.

Fig. 3 describes how 32 signed multiplication results are all summed up in current by the DCSPs. If W_SX_1 is 1, W_SX_2 is -1, W_SX_{31} is -1, and the others are all 0, the resulting current difference, $I_P - I_N$, becomes $-2I_u$. The phases translated from the currents are accumulated in the DCCO without any analog to digital conversions. Then, the final accumulated phases, QP[10:0] and QN[10:0], are sampled by the phase quantizers (Fig. 4(a)) at the end of MAC operation. As illustrated in Fig. 3, the signal margin of PNMAC is improved by doubling the frequency difference by reusing the steering current in the CCO on the opposite side.

The MAC-result-readout circuit consists of two phase quantizers, three adders, and two constant multipliers as shown in Fig. 4(a). The counter outputs and the residual phases of the DCCO are sampled by the readout circuit as the MSBs and LSBs, respectively. The final sampled digital output is generated by the counter and phase detector. In the P-side phase quantizer, the counter counts the positive edges of QP[0] to sample the MSBs of the phase, CNT_P, while the LSBs of the phase, $\varphi_{P,0}$, are sampled by latching all the inverter outputs of the CCO_P. Thus, the phases of the CCO_P and CCO_N are expressed as follows:

$$\varphi_P = 22 \times \mathrm{CNT}_P + \varphi_{P,0}, \tag{3}$$

and

$$\varphi_N = 22 \times \mathrm{CNT}_N + \varphi_{N,0}, \tag{4}$$

where φ_P and φ_N are the phases of the CCO_P and CCO_N, respectively. Then, the phase difference, $\Delta\varphi$, is calculated as

$$\Delta\varphi = \varphi_P - \varphi_N. \tag{5}$$

The readout circuit generates $\Delta\varphi$ represented by 10b digital data. $\Delta\varphi$ is proportional to the final MAC result, so the final computational result is given by

$$\sum_{i=0}^{num-1} (1 - 2S_i)WX_i = \alpha_{CCO} \times \Delta\varphi, \tag{6}$$

where α_{CCO} is a scale factor, and *num* is the number of elements in the MAC operation.

Since the DCCO outputs, QP and QN are asynchronous with the main clock signal, metastable states, which can seriously degrade the accuracy, may happen. To resolve the metastability issue, we implement a phase synchronizer composed of two D flip-flops for rising-edge detection, an AND gate, a delay cell, and two sets of D flip-flops latching the DCCO phases. As shown in Fig. 4(b), the rising-edge detection synchronizers are enabled by the END signal triggered at the end of MAC operation to detect the rising edges of QP[0] and QN[0]. The SAMPLE signal, which is synchronized to QP[0] and QN[0], is switched from low to high after a delay time, T_d, passes. Then, all the data are sampled at the rising edge of SAMPLE, as shown in Fig. 4(c).

C. Digital In-Memory-Array Computing

Recent analog IMAC works, which perform computation inside the memory array, show significantly high energy efficiency and throughput. However, these advantages come at

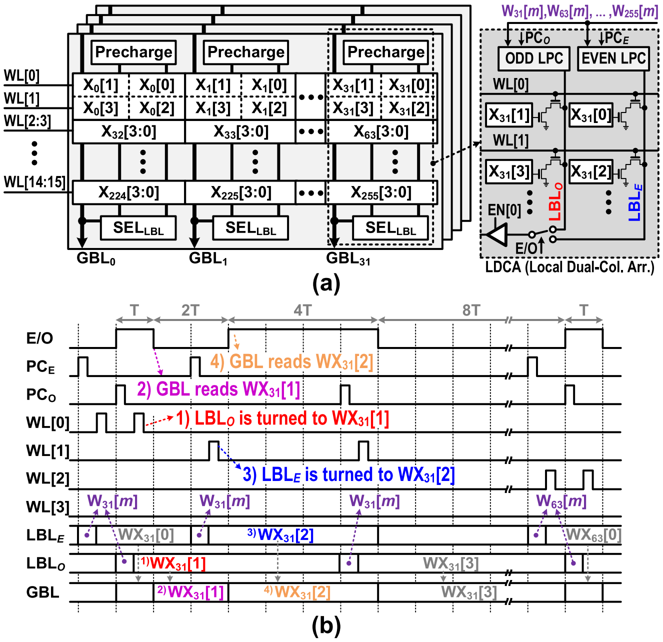

Precharge
Precharge
Precharge
WL[0]
WL[1]
WL[2:3]
WL[14:15]
X0[1] X0[0] X1[1] X1[0] X31[1] X31[0]
X0[3] X0[2] X1[3] X1[2] X31[3] X31[2]
X32[3:0] X33[3:0] X63[3:0]
X224[3:0] X225[3:0] X255[3:0]
SELLBL
GBL0 GBL1 GBL31
W31[m],W63[m], ... ,W255[m]
PCO PCE
ODD LPC EVEN LPC
WL[0]
WL[1]
X31[1] X31[0] X31[3] X31[2]
EN[0]
LBLO LBLE
E/O
LDCA (Local Dual-Col. Arr.)
(a)
T 2T 4T 8T T
E/O
PCE
PCO
WL[0]
WL[1]
WL[2]
WL[3]
LBLE
LBLO
GBL
4) GBL reads WX31[2]
2) GBL reads WX31[1]
1) LBLO is turned to WX31[1]
3) LBLE is turned to WX31[2]
W31[m] W31[m] W31[m] W63[m]
WX31[0] 3)WX31[2] WX63[0]
1)WX31[1] WX31[3]
2)WX31[1] 4)WX31[2] WX31[3]
(b)

Fig. 5. (a) Local dual-column array (LDCA) circuit and (b) timing diagram of the PWM pulse generation.

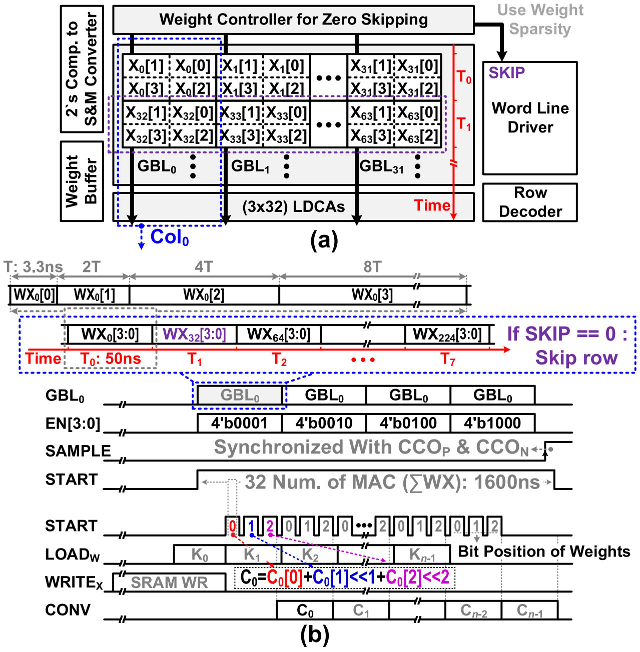

2's Comp. to S&M Converter
Weight Buffer
Weight Controller for Zero Skipping
Use Weight Sparsity
SKIP
Word Line Driver
Row Decoder
X0[1] X0[0] X1[1] X1[0] X31[1] X31[0]
X0[3] X0[2] X1[3] X1[2] X31[3] X31[2]
X32[1] X32[0] X33[1] X33[0] X63[1] X63[0]
X32[3] X32[2] X33[3] X33[2] X63[3] X63[2]
T0 T1
GBL0 GBL1 GBL31
(3x32) LDCAs
Time
Col0
(a)
T: 3.3ns 2T 4T 8T
WX0[0] WX0[1] WX0[2] WX0[3]
WX0[3:0] WX32[3:0] WX64[3:0] WX224[3:0]
If SKIP == 0 : Skip row
Time T0: 50ns T1 T2 T7
GBL0
EN[3:0]
4'b0001 4'b0010 4'b0100 4'b1000
SAMPLE
Synchronized With CCOP & CCON
START
32 Num. of MAC (∑WX): 1600ns
START
0 1 2 0 1 2 0 2 0 1 2 0 1 2
LOADW
K0 K1 K2 Kn-1
Bit Position of Weights
WRITEX
SRAM WR
C0=C0[0]+C0[1]<<1+C0[2]<<2
CONV
C0 C1 Cn-2 Cn-1
(b)

Fig. 6. (a) Zero-skipping operation with the weights in the S&M number system and (b) overall timing diagram of the MAC operation with the multi-bit weights.

the cost of the accuracy degradation from analog computing nonidealities. Its intrinsic analog-computing nature makes in-memory architectures vulnerable to PVT variations. Hence, we employ the accurate and robust DIMAC with the PNMAC using a well-designed current source to mitigate the nonlinearity and inconsistency of analog computing.

Fig. 5(a) shows the data-mapping diagram of the DIMAC and detailed circuit of the LDCA that generates PWM signals with the multiplication values, $W[m] \times X[3:0]$. The 4b data, X[3:0], are stored in two columns by arranging them in even-bit and odd-bit positions. One of the even-bit and odd-bit columns is selected by the E/O signal, and EN[3:0] enables one of 4 LDCAs. $W[m]$ is precharged to the LBL by the LPC, and $X[n]$ is read from the cell through the WL. Then, the bit-wise multiplication result remains in the LBL, and the GBL reads that result through the tri-state buffer.

The LDCA employs an overlapped precharging technique that we propose to increase the throughput of the DIMAC. The detailed timing diagram of the PWM pulse generation using the overlapped precharging is shown in Fig. 5(b). If the bit position of $X[n]$ is an odd number, the LBL_O is turned to $WX[n]$ while the GBL is connected to the LBL_E. Then, E/O is flipped to read the LBL_O. It increases the throughput by hiding the precharging time in the PWM pulse. For the case of the 4b PWM modulation scheme, the data throughput is improved by 26.6% compared to the conventional precharging technique.

The zero-skipping operation and overall timing diagram of the MAC operation are shown in Fig. 6. The weight controller reads only when there is a non-zero value in the fetched weights and forwards each bit to the LPC. If the fetched weights do not have any non-zero values, the SKIPn signal is switched from high to low to disable the WL driver and weight driver. Also, GBLs are turned to zero by the weight controller. The weight is selected as an external source to use weight sparsity [14].

Fig. 6(b) shows the overall timing diagram of the MAC operation with multi-bit weights in the first LDCA group sharing GBL_0. The first MAC operation, $C_0[0]$, starts after all weights of K_0 are stored in the weight buffer. 32 MAC computations are sequentially performed in the first LDCA group and take 480 cycles, which is 1600ns at 300MHz operation frequency. Since the 32 LDCA groups sharing GBL_0 to GBL_{31} operate in parallel, up to 1024 MAC computations are performed for $C_0[0]$ in our CNN engine. After all these MAC operations are finished, the MAC result, $C_0[0]$ $(= \sum W[0]X[3:0])$, is sampled once by the readout circuit. Then, $C_0[1]$ and $C_0[2]$ are calculated in the same way as $C_0[0]$. Finally, the remaining MAC operations are performed in the digital domain to calculate the final MAC result as in the following equation:

$$C_n = C_n[0] + (C_n[1] \ll 1) + (C_n[2] \ll 2). \qquad (7)$$

In this work, a high skip rate is achieved by using the S&M number system, which can further increase the bit-level sparsity of weights. Fig. 7 shows the weight-concentrated region for 2's complement and S&M number systems. In general, most of the weights in neural networks are concentrated near zero. As shown in Fig. 7, the positive weights have high bit-level sparsity in both number systems, while the negative weights have high bit-level sparsity only in the S&M system. Since bin(−1) is mapped to 1111, the 2's complement system has low bit-level sparsity for negative weights. To increase the bit-level zero-skipping rate, we employ the S&M, which provides higher bit-level sparsity than the 2's complement.

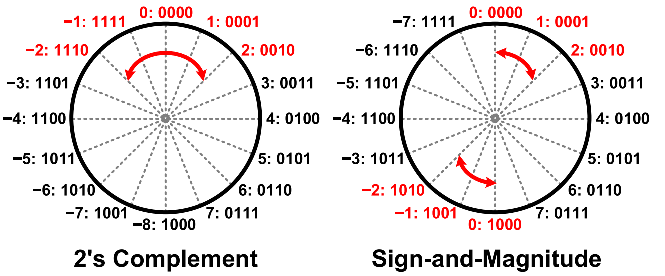


Fig. 7. Weight-concentrated region in two different number systems: 2's complement and sign-and-magnitude systems.

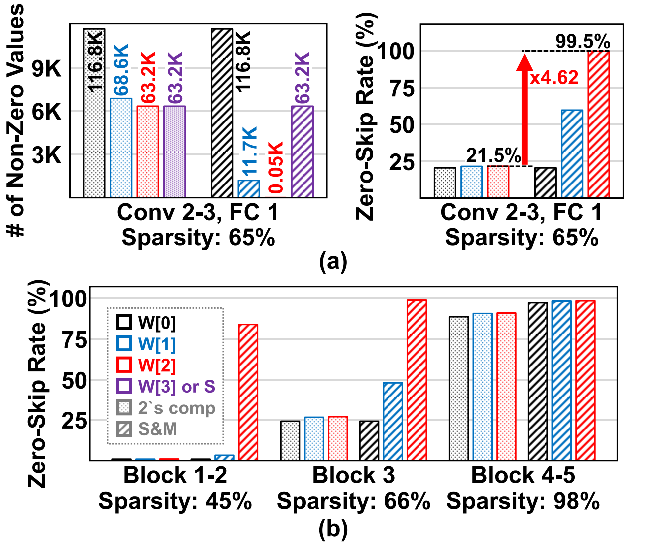


Fig. 8. (a) Number of non-zero values in the weights and zero-skipping rate in our simple-CNN and (b) zero-skipping rate in the VGG-16.

IV. Performance and Experimental Results

This section presents the measured results of the prototype IC, simulation analysis, and performance of the CNN accuracy. The proposed 4kb hybrid 8T-SRAM-based CIM is fabricated in a 65nm CMOS technology and packaged in a 40-pin quad-flat no-leads (QFN). To verify the proposed architecture and circuit operation, the weight sparsity and the classification accuracy with analog nonidealities are simulated, and the measured classification results and performances are presented in this section.

A. Bit-Level Sparsity of Weights

Zero-skipping has been widely used recently because high energy efficiency can be achieved utilizing data sparsity. This work also employs a zero-skipping scheme to skip unnecessary operations and achieve a higher zero-skipping rate by using the S&M number system. As shown in Fig. 8(a), the S&M has fewer non-zero values in W[1] and W[2]. In the S&M, the overall average zero-skipping rate is 59.8%, and W[2] has a zero-skipping rate of 99.5%. By representing the weight data in S&M, the skipping rate of W[2] and the average skipping rate increase by 4.62× and 2.83× compared to the 2's complement representation, respectively. Fig. 8(b) shows the zero-skipping rate in the VGG-16. As shown in this analysis result, S&M can obtain a very large zero-skip rate gain in a layer with low sparsity.

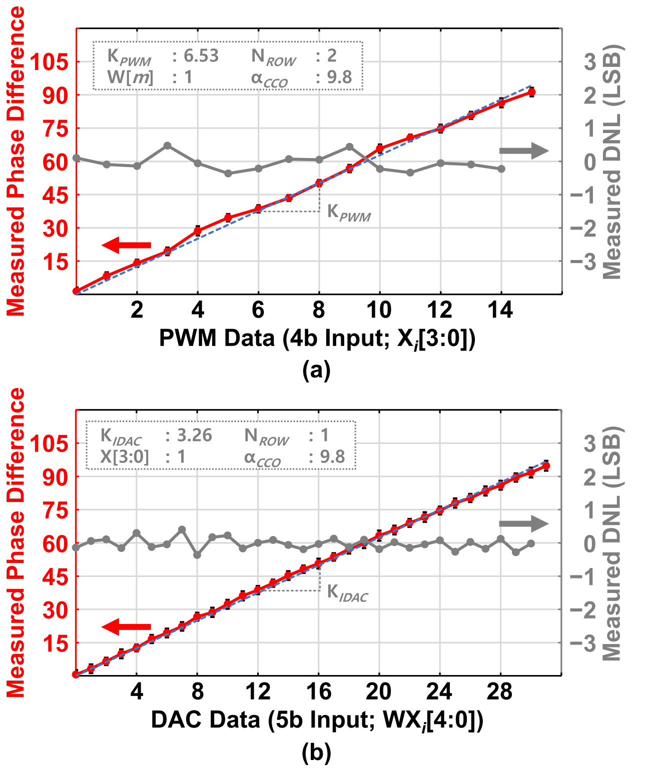


Fig. 9. Measured transfer functions and DNLs of the (a) PWM and (b) IDAC.

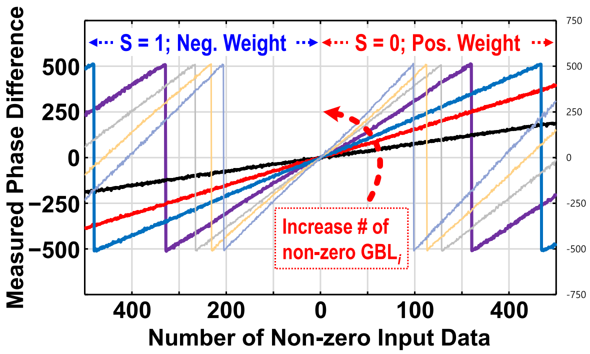


Fig. 10. Measured computation results generated by the combination of the PWM and IDAC transfer functions.

B. Measured Transfer Function

Figs. 9 and 10 present the measured results of the proposed CIM at 300MHz operating frequency. The measured PWM transfer function and DAC current (IDAC) transfer function are shown in Fig. 9(a) and (b), respectively. The slope of the PWM transfer function, K_{PWM}, is determined as

$$K_{PWM} = (W[m] \times 32N_{ROW})/\alpha_{CCO}, \tag{8}$$

and the slope of the IDAC transfer function, K_{IDAC}, is given by

$$K_{IDAC} = (X[3:0] \times 32N_{ROW})/\alpha_{CCO}, \tag{9}$$

where N_{ROW} is the number of activated rows in the CIM. To analyze the PWM characteristics as in Fig. 9(a), two rows of the SRAM array store 4b data, X_i[3:0], and the other rows store zeros. All weights are set to one, and the phase

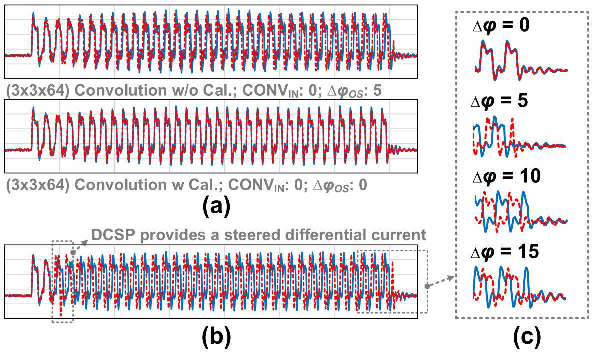


Fig. 11. Measured waveforms of the DCCO outputs (a) with and without calibration, (b) with a differential current, and (c) at the end of operation according to various input data.

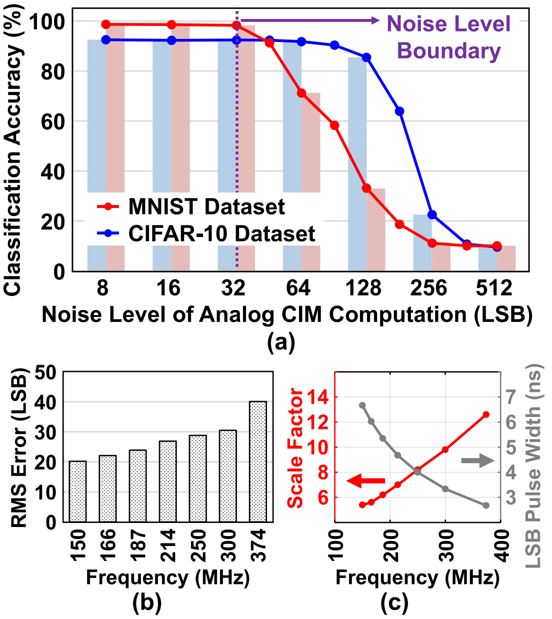


Fig. 12. (a) Simulated classification accuracy vs. the noise level of the analog CIM computation for MNIST and CFAR-10 datasets, (b) measured RMS error vs. the operation frequency, and (c) measured scale factor and LSB pulse width vs. the operation frequency.

difference of the DCCO is measured with changing X_i[3:0]. For characterization of the IDAC shown in Fig. 9(b), one row of the SRAM array stores one, and the other rows store zeros. The phase difference of the DCCO is measured with changing the number of weights having a value of one. An initial one-time calibration is needed to find α_{CCO} (= the scale factor of the DCCO), which is easily achieved through a 2-point calibration. As shown in Fig. 9, good linearity is observed in both the PWM and IDAC transfer functions with the DNL less than 1 LSB. The measured computation results generated by the combination of the PWM and IDAC functions also have good linearity, as shown in Fig. 10. The slope in Fig. 10 is determined by the number of non-zero LDCA groups which have dedicated GBLs.

The actual waveforms of the DCCO outputs measured by the oscilloscope are presented in Fig. 11. The offset is initially calibrated once to prevent accuracy degradation, as shown in Fig. 11(a). Fig. 11(b) presents that the phase difference of the CCO pair occurs due to the frequency difference caused by the steered differential current of the DCSP. The phase difference is sampled at the end of MAC operation, and the phase difference according to the various input data is shown in Fig. 11(c).

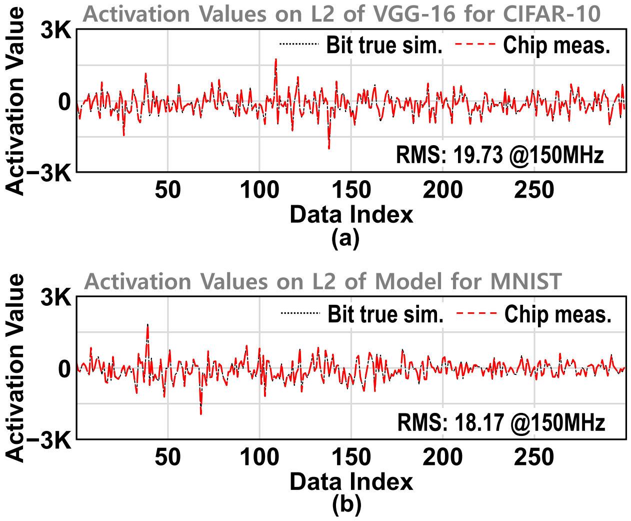


Fig. 13. Measured and simulated activation values of the CNN: (a) values on the VGG-16 network for CIFAR-10 and (b) values on the simple-CNN network for MNIST.

Measured MNIST Classification

Dataset (# of test set)	MNIST (10,000)
Accuracy of Chip (Sim.)	98.09% (98.36%)
Bit Resolution	X: 4b, W: 4b, A: 10b
Energy per Classification	831.9nJ [1]

1) L2, L3, and 512 FC layer computation energy at 0.75V (IMAC) + 0.8V (NMAC). Off-chip memory access is not included

Network Topology

64C3-MP2-64C3-MP2-64C3-MP2-512FC-10FC [2]

2) *n*C*k* – *k*×*k* kernel with n filters, *m*FC – *m* neuron FC layer, and MP*p* – *p*×*p* pooling size

Fig. 14. Measured MNIST classification results and network topology.

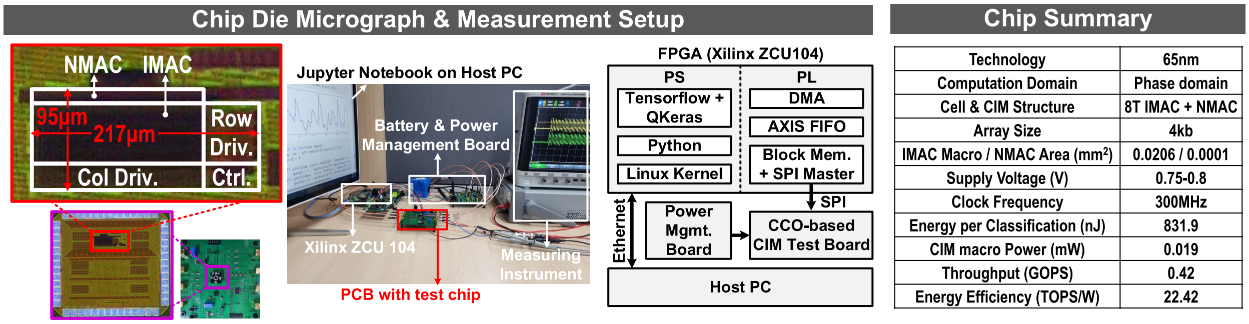
C. Noise Analysis and Measured Results on CNNs

To demonstrate the proposed CIM macro's functionality for real CNN applications, the simulated and measured classification results and the measured error performances are presented in Fig. 12 to 14. The noise-level-tolerance result in Fig. 12(a) shows a degradation in classification accuracy when the noise level of analog computation increases. The noise level here is defined as the RMS error of the analog computation, which is caused by various noise contributors: quantization noise, thermal noise, and others. Based on these simulation results, the boundary of allowable noise level is found and used for the design. The computation noise analysis of the proposed hybrid CIM indicates the main trade-off between the operation frequency and resolution, as shown in Fig. 12(b). As the operation frequency increases, the scale factor increases, and the PWM pulse width becomes shorter, thus increasing the quantization noise. Fig. 12(c) shows the scale factor, α_{CCO}, as a function of the operation frequency. The maximum frequency is chosen at 300MHz to limit the RMS error within the boundary. The scale factor is also susceptible to process variations. From corner simulations, it is found that the CCO has a frequency variation of about 13%. However, such variations can be compensated with our calibration scheme.

TABLE I
COMPARISON WITH STATE-OF-THE-ART WORKS

	ISSCC '20 [13]	JSSC '19 [16]	ISSCC '19 [17]	ISSCC '20 [10]	ISSCC '21 [15]	This work
Technology	28nm	28nm	40nm	7nm	22nm	65nm
Domain	Analog	Phase	Time	Analog	Digital	Phase
Cell & CIM Structure	6T IMAC + NMAC	Only MAC engine	Only MAC engine	8T IMAC	6T Cell + Digital	8T IMAC + NMAC
Array Size	64kb	-	-	4kb	64kb	4kb
Macro Area (mm^2)	0.3230	[2]0.0012	[2]0.124	0.0032	0.202	0.0206
Bit Precision (Input / Weight / Output)	4~8b / 4~8b / 12~20b	8b / 8b / 10b	8b / 1b / 8b	4b / 4b / 4b	1~8b / 4~16b / 16~25b	4b / 1.5~4b / 10b
Supply Voltage (V)	0.7~0.9	0.7	0.375~1.1	0.8	0.72	0.75~0.8
Dataset	CIFAR-10	MNIST	MNIST	MNIST	-	MNIST / CIFAR-10
Measured Accuracy (Software Baseline) (%)	91.5 (91.7)	98.1 (98.2)	[7]98.42 (98.92)	98.5 (99.63)	-	98.1 / [7]92.3 (98.36 / 92.48)
Total CIM / NMAC Power (mW)	[1]1.825 / -	- / [2]0.152	- / [2]0.030	[1]1.061 / -	[1]37.078 / -	[3]0.019 / [2]0.0033
MAC rate (MHz)	[4]119~244	780	0.19~3.12	[4]181	[4]55~100	20
Throughput (GOPS)	124.88 (4b/4b)	[5,6]8.512	[6]0.182	372.4	3300 (4b/4b)	0.42 (4b/4b)
Energy Efficiency (TOPS/W)	68.44 (4b/4b)	[2,5,6]56.0	[2,6]6.05	351	89 (4b/4b)	[3]22.4 / [2]128.6 (4b/4b)

[1] Calculated by dividing throughput (GOPS) by energy efficiency (TOPS/W)
[2] Metrics for MAC engine or NMAC
[3] Total CIM (DIMAC+PNMAC) power and energy with a 59.8% zero-skipping rate
[4] Calculated by dividing 1 by cycle time (ns)
[5] Calculated when the input activation rate is 3%
[6] Normalized to 4b input and 4b weight
[7] Simulation result with circuit non-ideal effects



Technology	65nm
Computation Domain	Phase domain
Cell & CIM Structure	8T IMAC + NMAC
Array Size	4kb
IMAC Macro / NMAC Area (mm^2)	0.0206 / 0.0001
Supply Voltage (V)	0.75-0.8
Clock Frequency	300MHz
Energy per Classification (nJ)	831.9
CIM macro Power (mW)	0.019
Throughput (GOPS)	0.42
Energy Efficiency (TOPS/W)	22.42

Fig. 15. Chip micrograph, chip measurement summary, and measurement setup for CNN applications with the Tensorflow libraries on Xilinx ZCU104.

Fig. 13 shows the chip-measured activation values on the L2 layer of the VGG-16 for CIFAR-10 and the L2 layer of the simple-CNN for MNIST. The chip-measured activation values almost match the outputs of the bit-true simulation, and their RMS errors are 19.73 and 18.17. The correlation coefficients, which can quantitatively show the similarity between the bit-true simulation values and chip-measured values, are 0.997 and 0.979 in the CIFAR-10 and MNIST, respectively. In the simple CNN topology, the proposed system achieves an inference accuracy of 98.09% with 831.9nJ of energy per classification for 10,000 MNIST test sets, as shown in Fig. 14. The measured inference accuracy of MNIST classification is reduced only by less than 0.3% compared to that of the bit-true simulation.

D. Measurement Setup and Chip Summary

Fig. 15 shows the die micrograph and measurement setup along with a top system block diagram. A chip summary is also presented. The prototype IC is implemented using a 65nm CMOS technology, and the IMAC and NMAC blocks occupy 0.0206 and 0.0001mm^2, respectively. The measurement setup consists of an FPGA, a power management board, and a PCB with the prototype IC. The Xilinx ZCU104 FPGA board is used to feed the trained weights and activation inputs to the prototype IC. The FPGA is connected to the prototype IC through the SPI interface and to the host PC through Ethernet. The simple-CNN runs in the processing system (PS) using the TensorFlow [20] and QKeras [21] software framework. The SPI master implemented in the FPGA has an FIFO and a block memory for real-time CNN inference with the CIM macro.

E. Comparison With Prior Works

Table I shows the performance comparison with state-of-the-art works. The measured peak energy efficiency of this work is 128.6TOPS/W for the NMAC and 22.4TOPS/W for the overall CIM macro at 300MHz operation frequency. The power consumption of the proposed NMAC is only 3.3μW at 0.8V, which is 46.06× and 9.09× lower than [16] and [17], respectively. Compared to the state-of-the-art hybrid CIMs, this work consumes 19.03μW with a 59.8% zero-skipping rate, which is 96.05× lower power than [13]. This work that can perform energy-efficient AI computing with ultra-low power consumption is suitable for resource-constrained edge devices.

V. Conclusion

This paper presents a CIM macro based on hybrid computation using DIMAC and PNMAC. The proposed hybrid CIM is suitable for resource-constrained edge devices by performing energy-efficient AI computing with ultra-low power consumption. The PNMAC with the proposed steering-DAC-based differential CCO consumes only 3.3μW while performing accumulation and data conversion. In addition, the signal margin of the PNMAC is improved by doubling the frequency difference by reusing the steering current. The overlapped precharging technique and the S&M bit-wise zero-skipping scheme are proposed to improve the throughput and energy efficiency. The proposed CIM macro has been implemented in a 65nm CMOS process and operates with an ultra-low power consumption of 19.03μW and high energy efficiency of 22.4TOPS/W at 300MHz operation frequency.

References

[1] S. M. Nabavinejad, M. Baharloo, K.-C. Chen, M. Palesi, T. Kogel, and M. Ebrahimi, "An overview of efficient interconnection networks for deep neural network accelerators," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 10, no. 3, pp. 268–282, Sep. 2020.

[2] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, Nov. 2017.

[3] Y.-H. Chen, T.-J. Yang, J. Emer, and V. Sze, "Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices," *IEEE J. Emerging Sel. Topics Circuits Syst.*, vol. 9, no. 2, pp. 292–308, Jun. 2019.

[4] W. A. Wulf and S. A. McKee, "Hitting the memory wall: Implications of the obvious," *ACM SIGARCH Comput. Archit. News*, vol. 23, no. 1, pp. 20–24, Mar. 1995.

[5] J.-M. Hung *et al.*, "Challenges and trends indeveloping nonvolatile memory-enabled computing chips for intelligent edge devices," *IEEE Trans. Electron Devices*, vol. 67, no. 4, pp. 1444–1453, Apr. 2020.

[6] J. Zhang, Z. Wang, and N. Verma, "In-memory computation of a machine-learning classifier in a standard 6T SRAM array," *IEEE J. Solid-State Circuits*, vol. 52, no. 4, pp. 915–924, Apr. 2017.

[7] X. Si *et al.*, "A twin-8T SRAM computation-in-memory unit-macro for multibit CNN-based AI edge processors," *IEEE J. Solid-State Circuits*, vol. 55, no. 1, pp. 189–202, Jan. 2020.

[8] H. Valavi, P. J. Ramadge, E. Nestler, and N. Verma, "A 64-tile 2.4-Mb in-memory-computing CNN accelerator employing charge-domain compute," *IEEE J. Solid-State Circuits*, vol. 54, no. 6, pp. 1789–1799, Jun. 2019.

[9] Z. Chen *et al.*, "CAP-RAM: A charge-domain in-memory computing 6T-SRAM for accurate and precision-programmable CNN inference," *IEEE J. Solid-State Circuits*, vol. 56, no. 6, pp. 1924–1935, Jun. 2021.

[10] M. E. Sinangil *et al.*, "A 7-nm compute-in-memory SRAM macro supporting multi-bit input, weight and output and achieving 351 TOPS/W and 372.4 GOPS," *IEEE J. Solid-State Circuits*, vol. 56, no. 1, pp. 188–198, Jan. 2021.

[11] S. K. Gonugondla, M. Kang, and N. R. Shanbhag, "A Variation-tolerant in-memory machine learning classifier via on-chip training," *IEEE J. Solid-State Circuits*, vol. 53, no. 11, pp. 3163–3173, Nov. 2018.

[12] Z. Jiang, S. Yin, J.-S. Seo, and M. Seok, "C3SRAM: An in-memory-computing SRAM macro based on robust capacitive coupling computing mechanism," *IEEE J. Solid-State Circuits*, vol. 55, no. 7, pp. 1888–1897, Jul. 2020.

[13] X. Si *et al.*, "A local computing cell and 6T SRAM-based computing-in-memory macro with 8-b MAC operation for edge AI chips," *IEEE J. Solid-State Circuits*, vol. 56, no. 9, pp. 2817–2831, Sep. 2021.

[14] J.-H. Kim, J. Lee, J. Lee, J. Heo, and J.-Y. Kim, "Z-PIM: A sparsity-aware processing-in-memory architecture with fully variable weight bit-precision for energy-efficient deep neural networks," *IEEE J. Solid-State Circuits*, vol. 56, no. 4, pp. 1093–1104, Apr. 2021.

[15] Y.-D. Chih *et al.*, "16.4 An 89 TOPS/W and 16.3 TOPS/mm^2 all-digital SRAM-based full-precision compute-in memory macro in 22 nm for machine-learning edge applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2021, pp. 252–254.

[16] Y. Toyama, K. Yoshioka, K. Ban, S. Maya, A. Sai, and K. Onizuka, "An 8 bit 12.4 TOPS/W phase-domain MAC circuit for energy-constrained deep learning accelerators," *IEEE J. Solid-State Circuits*, vol. 54, no. 10, pp. 2730–2742, Oct. 2019.

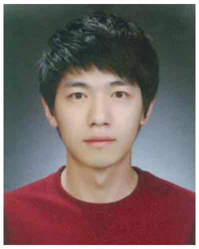
[17] A. Sayal, S. S. T. Nibhanupudi, S. Fathima, and J. P. Kulkarni, "A 12.08-TOPS/W all-digital time-domain CNN engine using bi-directional memory delay lines for energy efficient edge computing," *IEEE J. Solid-State Circuits*, vol. 55, no. 1, pp. 60–75, Jan. 2020.

[18] R. Khaddam-Aljameh *et al.*, "HERMES core—A 14 nm CMOS and PCM-based in-memory compute core using an array of 300ps/LSB linearized CCO-based ADCs and local digital processing," in *Proc. Symp. VLSI Circuits*, Jun. 2021, pp. 1–2.

[19] Y. G. Yoon, J. Kim, T. K. Jang, and S. Cho, "A time-based bandpass ADC using time-interleaved voltage-controlled oscillators," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 11, pp. 3571–3581, Dec. 2008.

[20] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*.

[21] C. N. Coelho *et al.*, "Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors," *Nature Mach. Intell.*, vol. 3, pp. 675–686, Jun. 2021.



Injun Choi (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in electronics engineering from Chungnam National University, Daejeon, South Korea, in 2014 and 2016, respectively. He is currently pursuing the Ph.D. degree with the Korea Advanced Institute of Science and Technology, Daejeon.

His research interests include circuit design, memory, and in-memory computation for machine learning.

Edward Jongyoon Choi (Graduate Student Member, IEEE) received the B.S. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign, Champaign, Illinois, USA, in 2019, and the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2021, where he is currently pursuing the Ph.D. degree in electrical engineering.

His research interest includes integrated circuit (IC) design for low-power memory computing circuits.

Donghyeon Yi (Student Member, IEEE) received the dual B.S. degree in electrical engineering and mechanical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2020.

His research interests include designing low-power neural network processing ICs, and applications on biomedical devices and wireless sensor nodes.

Yoontae Jung (Student Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2017 and 2019, respectively, where he is currently pursuing the Ph.D. degree in electrical engineering.

His current research interests include design of low-power sensor interface, biomedical integrated circuits, and data converters. He served as a Reviewer for the IEEE JOURNAL OF SOLID-STATE CIRCUITS.

Hoyong Seong (Student Member, IEEE) received the B.S. degree in electronics engineering from Kwangwoon University, Seoul, South Korea, in 2017, and the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2019, where he is currently pursuing the Ph.D. degree in electrical engineering.

His current research interest includes mixed-signal circuit design for sensor interface IC.

Hyuntak Jeon (Member, IEEE) received the B.S. degree in electronic and electrical engineering from Hongik University, Seoul, South Korea, in 2015, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2017 and 2021, respectively.

He is currently working as a Senior Researcher at the Agency for Defense Development (ADD), Daejeon. His research interests include mixed-signal circuit design for sensor interface IC and electrical system design for guided ballistic missile systems.

Soon-Jae Kweon (Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2010 and 2018, respectively.

He was a Post-Doctoral Researcher with the Information Engineering and Electronics Research Institute, KAIST, from 2018 to 2020. After finishing the first Post-Doctoral Researcher, he is currently with New York University Abu Dhabi, United Arab Emirates, as a Post-Doctoral Associate. His research interests include designing low-power sensor interface ICs, wireless communication ICs, data converters for miniature biomedical devices, and wireless sensor nodes.

Ik-Joon Chang (Member, IEEE) received the B.S. degree *(summa cum laude)* in electrical engineering from Seoul National University, Seoul, South Korea, and the M.S. and Ph.D. degrees from the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA, in 2005 and 2009, respectively.

After graduation, he was with the Samsung Electronics NAND Flash Design Team for two years. He is currently an Assistant Professor with Kyung Hee University, South Korea. He was awarded by the Samsung Scholarship Foundation in 2005.

Sohmyung Ha (Senior Member, IEEE) received the B.S. (*summa cum laude*) and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2004 and 2006, respectively, and the M.S. and Ph.D. degrees in biomedical engineering from the Department of Bioengineering, University of California San Diego, La Jolla, CA, USA, in 2015 and 2016, respectively.

From 2006 to 2010, he worked as an Analog and Mixed-Signal Circuit Designer at Samsung Electronics, Yongin, South Korea. He was a part of the engineering team responsible for several of the world's best-selling multimedia devices, smartphones, and TVs. After an extended career in industry, he returned to academia as a full bright scholar. Since 2016, he has been an Assistant Professor of electrical engineering and bioengineering at New York University Abu Dhabi, Abu Dhabi, UAE, and a Global Network Assistant Professor at the Department of Electrical and Computer Engineering and the Department of Biomedical Engineering, Tandon School of Engineering, New York University, New York, NY, USA. His research interests include advancing the engineering and applications of silicon integrated technology interfacing with biology in a variety of forms ranging from implantable biomedical devices to unobtrusive wearable sensors. He is a member of the Analog Signal Processing Technical Committee (ASP TC) and the Biomedical and Life Science Circuits and Systems Technical Committee (BioCAS TC), IEEE Circuits and Systems Society (CASS). He has been a member of the International Technical Program Committee (ITPC), International Solid-State Circuits Conference (ISSCC), since 2022. He was a recipient of the Best Ph.D. Thesis Award. He served as an Associate Editor for *Smart Health* (*Elsevier*) from 2016 to 2021. He currently serves as an Associate Editor for IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS and *Frontiers in Electronics*.

Minkyu Je (Senior Member, IEEE) received the M.S. and Ph.D. degrees in electrical engineering and computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 1998 and 2003, respectively.

In 2003, he joined Samsung Electronics, Giheung, South Korea, as a Senior Engineer. He worked on multi-mode multi-band RF transceiver SoCs for GSM/GPRS/EDGE/WCDMA standards. From 2006 to 2013, he was with the Institute of Microelectronics (IME), Agency for Science, Technology, and Research (A*STAR), Singapore. He worked as a Senior Research Engineer from 2006 to 2007, a member of Technical Staff from 2008 to 2011, a Senior Scientist in 2012, and the Deputy Director in 2013. From 2011 to 2013, he led the Integrated Circuits and Systems Laboratory, IME, as the Department Head. He was also the Program Director of the neurodevices program under A*STAR Science and Engineering Research Council (SERC) from 2011 to 2013 and an Adjunct Assistant Professor with the Department of Electrical and Computer Engineering, National University of Singapore (NUS), from 2010 to 2013. He was an Associate Professor with the Department of Information and Communication Engineering, Daegu Gyeongbuk Institute of Science and Technology (DGIST), South Korea, from 2014 to 2015. Since 2016, he has been an Associate Professor with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST). In IME, he led various projects developing low-power 3D accelerometer ASICs for high-end medical motion sensing applications, readout ASICs for nanowire biosensor arrays detecting DNA/RNA and protein biomarkers for point-of-care diagnostics, ultra-low-power sensor node SoCs for continuous real-time wireless health monitoring, wireless implantable sensor ASICs for medical devices, low-power radio SoCs, and MEMS interface/control SoCs for consumer electronics and industrial applications. He is an editor of one book and an author of six book chapters. He has more than 300 peer-reviewed international conference and journal publications in the areas of sensor interface IC, wireless IC, biomedical microsystem, 3D IC, device modeling, and nanoelectronics. He also has more than 50 patents issued or filed. His main research interests include advanced IC platform development, including smart sensor interface ICs and ultra-low-power wireless communication ICs, as well as microsystem integration leveraging the advanced IC platform for emerging applications, such as intelligent miniature biomedical devices, ubiquitous wireless sensor nodes, and future mobile devices.

Dr. Je has served on the Technical Program Committee and Organizing Committee for various international conferences, symposiums, and workshops, including IEEE International Solid-State Circuits Conference (ISSCC), IEEE Asian Solid-State Circuits Conference (A-SSCC), and IEEE Symposium on VLSI Circuits (SOVC). He is currently working as a Distinguished Lecturer of the IEEE Circuits and Systems Society.