IG-CRM: Area/Energy-Efficient IGZO-Based Circuits and Architecture Design for Reconfigurable CIM/CAM Applications

Zeyu Guo[1,2], Jinshan Yue[1*], Shengzhe Yan[1,2], Zhuoyu Dai[1,2], Xiangqu Fu[1,2], Zhaori Cong[1,2], Zening Niu[1], Ke Hu[1,2], Lihua Xu[1,2], Jiawei Wang[1], Lingfei Wang[1,2], Guanhua Yang[1], Di Geng[1,2], Ling Li[1,2*]

[1]Institute of Microelectronics of the Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
*Corresponding authors: {yuejinshan, lingli}@ime.ac.cn

ABSTRACT

Artificial intelligence is evolving with various algorithms such as deep neural network (DNN), Transformer, recommendation system (RecSys) and graph convolutional network (GCN). Correspondingly, multiply-accumulate (MAC) and content search are two main operations, which can be efficiently executed on the emerging computing-in-memory (CIM) and content-addressable-memory (CAM) paradigms. Recently, the emerging Indium-Gallium-Zine-Oxide (IGZO) transistor becomes a promising candidate for both CIM/CAM circuits, featuring ultra-low leakage with >300s data retention time and high-density BEOL fabrication. This paper proposes IG-CRM, the first IGZO-based circuits and architecture design for Reconfigurable CIM/CAM applications. The main contributions include: 1) at cell level, propose IGZO-based 3T0C/4T0C cell design that enables both CIM and CAM functionalities while matching IGZO/CMOS voltage; 2) at circuit level, utilize the BEOL IGZO transistor to reduce digital adder tree area in CIM circuits; 3) at architecture level, propose a reconfigurable CIM/CAM architecture with four macro structures based on 3T0C/4T0C cells. The proposed IG-CRM architecture shows high area/energy efficiency on various applications including DNN, Transformer, RecSys and GCN. Experiment results show that IG-CRM achieves 8.09× area saving compared with the SRAM-based non-reconfigurable CIM/CAM baseline, and $1.53\times10^3\times$/51.9× speedup and $1.63\times10^4\times$/$7.62\times10^3\times$ energy efficiency improvement compared with CPU and GPU on average.

KEYWORDS

Reconfigurable, IGZO, Computing-in-memory, Content-addressable memory

1 INTRODUCTION

Artificial intelligence (AI) is evolving in a vast of scenarios with various algorithms such as deep neural networks (DNN)[1], Transformer[2], recommendation system (RecSys)[3], graph convolutional network (GCN)[4]. The multiply-accumulate (MAC) and content search are two dominant operations, which can be efficiently executed on the emerging computing-in-memory (CIM) and content-addressable-memory (CAM) paradigms. CIM architecture features high energy efficiency and throughput thanks to the integrated memory and computation units[5]. Meanwhile, CAM[6] is efficient for vector search operations, which can perform massive parallel searches in a single clock cycle.

Though various devices have been explored for CIM/CAM functions, they still suffer some non-ideal features, such as the low density of SRAM[7], resistance variations and limited endurance of RRAM[8]. Similar problems exist in the CAM functions.

Recently, the Indium-Gallium-Zine-Oxide (IGZO) transistor, previously used as a thin-film transistor (TFT), is gaining considerable attention. Thanks to its ultra-low leakage current, the 2-transistor-0-capacitor (2T0C) structure is a promising solution for future high-density storage and CIM/CAM functions with >300s retention time and $4F^2$ density (F: feature size)[9, 10, 11]. Further, IGZO transistor is back-end-of-line (BEOL) compatible, thus can be 3D-stack over CMOS to save area. However, the high operating voltage and low working frequency are still challenging for CIM/CAM functions.

Previous works have explored the CIM/CAM architectures for DNN or graph applications. For DNN applications, bare CIM architectures are implemented without CAM functions[12, 13, 14]. For graph applications, the existing architectures[15, 16, 17, 18, 19] mainly consist of separate CIM and CAM modules, which limits the flexibility for different types of AI applications. The DNN/Transformer algorithms usually feature heavy CIM operations without CAM operations, while the RecSys/GCN algorithms require heavy CAM but limited CIM operations. *To the best of our knowledge, no previous architecture can efficiently support all these types of algorithms.*

Motivated by the previous challenges, this work proposes IG-CRM, the first IGZO-based circuits and architecture design for Reconfigurable CIM/CAM applications. Area/energy-efficient IGZO-based CIM/CAM cells and hybrid adder trees are designed with a reconfigurable architecture for flexible CIM/CAM workloads. The main contributions are as follows:

- Propose IGZO-based 3T0C/4T0C cell circuits for reconfigurable CIM/CAM functions. By 3D-stack CMOS and IGZO transistors, the 3T0C/4T0C cells achieve a small footprint with CMOS-compatible voltage and high working frequency.
- Propose an area-efficient IGZO-CMOS hybrid adder tree by replacing the NMOS transistors with 3D-stack IGZO. It can save up to 50% area with 12.6%-37.9% additional power.
- Propose a reconfigurable CIM/CAM architecture with four types of CIM/CAM macros, enabling high utilization for varied AI workloads. Experiment results show $1.53\times10^3\times$/51.9× average speedup and $1.63\times10^4\times$/$7.62\times10^3\times$ average energy efficiency compared with the CPU and GPU platforms on DNN, Transformer, RecSys, and GCN workloads.

DAC '24, June 23–27, 2024, San Francisco, CA, USA

ACM ISBN 979-8-4007-0601-1/24/06.
https://doi.org/10.1145/3649329.3656226

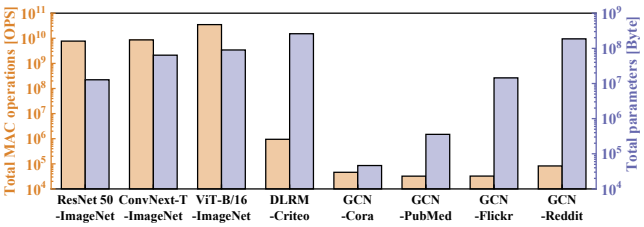


Figure 1: Varied operations/parameters of AI workloads.

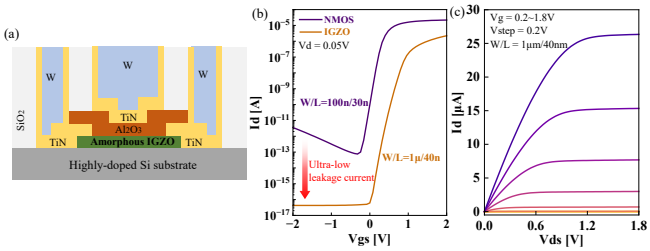


Figure 2: (a) IGZO cross-section schematic. (b) Transfer curve and leakage current compared with NMOS. (c) IGZO I-V curve.

2 BACKGROUND AND MOTIVATION

2.1 Various types of AI workloads

AI algorithms are evolving in a vast of application scenarios, in which two data structures are mainly adopted: Euclidean data and non-Euclidean data. Euclidean data can be represented by vectors/tensors, such as images, videos, and audio, which are usually processed with heavy MAC operations in DNN or Transformer algorithms. Meanwhile, non-Euclidean data are not represented by vectors/tensors. Graph is a typical type of non-Euclidean data, such as the social networks, the molecular structure of proteins, etc. Due to the random and scale-free nature of connectivity, graph data are usually processed with heavy content search operations and light MAC operations in RecSys and GCN algorithms. Corresponding to the heavy MAC and search operations, CIM and CAM are two promising paradigms. However, the varied CIM/CAM workloads make it challenging to efficiently support both CIM/CAM functions.

Fig. 1 presents several AI applications with different operations and parameters, which also reflects the CIM/CAM workload ratio. Firstly, DNN/Transformer and RecSys/GCN applications demand different CIM/CAM workload ratios. DNNs[1, 20] and Transformers[2] require heavy MAC operations together with heavy weight parameter storage. Instead, RecSys applications[3] and GCNs[4] feature 1-3 orders of magnitude higher parameters than MAC operations, mainly due to the rather sparse aggregation/embedding operations of graph data. Secondly, different scales of RecSys/GCN applications differ in the MAC and parameter sizes. The DLRM[3] model on Criteo dataset shows 20.6× MAC operations and >5600× parameters than GCN on Cora[21], which puts more pressure on the hardware resource allocation. The varied CIM/CAM workloads and the different scales of models motivate a reconfigurable CIM/CAM architecture with flexible workload mapping.

2.2 Existing CIM/CAM circuits & architectures

CIM and CAM circuits can be implemented on various devices, including SRAM[6, 7], RRAM[8], PCM, FeFET, etc. Both analog and digital CIM circuits are explored for SRAM CIM, where digital CIM is preferred for high accuracy. However, the low-density SRAM usually requires 6T or more transistors for CIM, and typically 9T/10T for CAM, which limits the area efficiency. The 1T1R RRAM is a high-density CIM solution that gathers current on bitline (BL) as MAC results. Two 1T1R cells can build a CAM cell to match '0' and '1'. However, the non-idealities of RRAM devices, such as resistance variation, are still accuracy concerns. Meanwhile, the limited write endurance (10^5-10^7[22]) prevents RRAM from deployment of frequently-changed applications.

At architecture level, previous works usually adopt separate CIM/CAM modules. Previous DNN architectures[12, 13, 14] are well explored but only implement CIM modules without CAM functions. In GaaS-X[16] for graph applications, the CAM module performs parallel search and generates a hit vector, which is utilized to activate the CIM operations. In iMARS[19] for RecSys, the configurable memory arrays (for CAM) store embedding tables and perform search operations, while the crossbar arrays (for CIM) implement MAC operations. Due to the separate and fixed CIM/CAM circuits, existing architectures cannot efficiently support varied CIM/CAM workloads in one chip.

2.3 Basics of IGZO transistors

IGZO transistor features ultra-low off-state current compared with CMOS. Fig. 2(a) shows the IGZO cross-sectional schematic. The gate-oxide layer acts as a natural capacitor to store data for a long time. IGZO transistor is BEOL-compatible, thus enabling 3D-stack over CMOS with $4F^2$ small footprint[10]. As a reference, SRAM usually takes 120-150F^2 while RRAM takes 60F^2 area[22]. Fig. 2(b) and (c) show the I-V curve of IGZO, with 4-5 orders of magnitude smaller leakage current than that of NMOS. The recent 2T0C IGZO structure has demonstrated >300s long retention time[9], which is sufficient for most AI applications.

IGZO shows high density, high endurance (similar to CMOS) and long retention time, which make it a promising device for CIM/CAM operations. However, there are still several challenges. Firstly, IGZO can only implement NMOS device, which means that it must be integrated with CMOS-based PMOS transistors. Secondly, IGZO requires a different working voltage compared with the CMOS transistors (e.g. 0.9V @28nm). Meanwhile, the low on-state current makes it difficult to realize high-frequency circuits. Therefore, dedicated hybrid IGZO/CMOS circuits design is required.

3 PROPOSED IGZO-BASED CIM/CAM CELLS AND ADDER TREE CIRCUITS

3.1 IGZO-based 3T0C/4T0C CIM/CAM cells

This subsection introduces the proposed IGZO-based CIM/CAM cell circuits. Traditional 2T0C IGZO cell is shown in Fig. 3(a). The write transistor $T1$ writes '0'/'1' to the hidden capacitor in the storage node (SN), while the read transistor $T2$ performs non-destructive read operation. The low-leakage IGZO can retain SN voltage (V_{SN}) for 311s following the IGZO SPICE model[1][11]. The retention time is defined as a 0.1V voltage drop point that retains a correct read result. With long retention time and non-destructive reading, the 2T0C IGZO is almost non-volatile with negligible refresh power.

However, the high working voltage and low on-state current of IGZO incur voltage compatibility and low frequency concerns. Therefore, one NMOS transistor is adopted to replace the original IGZO read transistor ($T2$), as shown in Fig. 3(b). Now It is capable of high-frequency execution. An optional PMOS ($T3$) is added to support CAM function so that $T2$ and $T3$ can match '0' and '1',

[1]https://github.com/guozeyu0827/IGZO_model.git

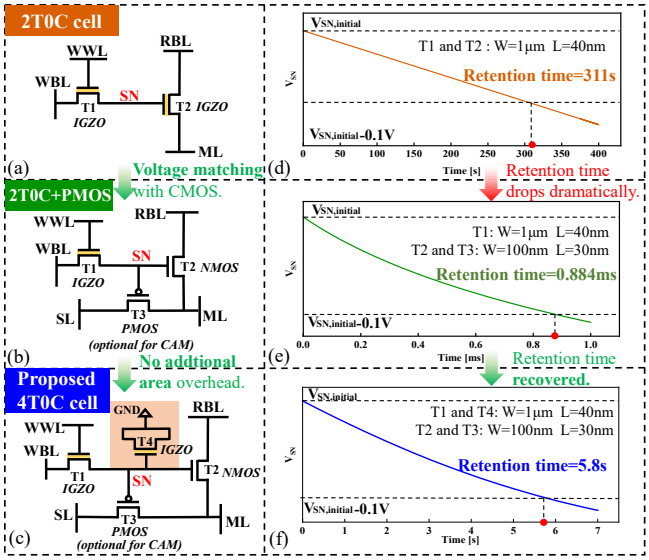


Figure 3: (a) Previous 2T0C IGZO cell. (b) 2T0C + PMOS. (c) 4T0C for CIM/CAM. (d)-(f) Corresponding retention time.

respectively. Only the write wordline (WWL) requires a 1.8V gate voltage, which is isolated from the other voltage domains.

Further, the 2T0C+PMOS structure suffers dramatically decreased retention time (0.884ms) due to the reduced hidden capacitor (replace $T2$ from IGZO to NMOS). To solve this issue, an additional IGZO ($T4$) connected to GND is added (in Fig. 3(c)), which serves as an additional hidden capacitor. The retention time can be recovered to 5.8s, which is enough for most real-time (<33ms/inference) applications. The retention power is still negligible. Note that the additional IGZO ($T4$) incurs no further area overhead since it can be 3D-stacked over the PMOS/NMOS transistors.

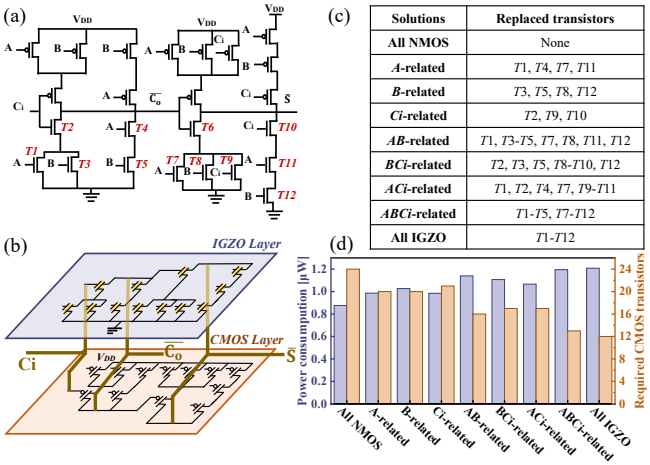
So far, the proposed 4T0C IGZO cell for CIM/CAM is finalized. For bare CIM function, $T3$ can be removed to form a 3T0C cell. The detailed CIM/CAM voltage configurations and waveforms are presented in Section 4.

3.2 Hybrid IGZO/CMOS 3D-stack adder tree

Besides the CIM/CAM cells, the CIM circuits, whether analog or digital, take up a considerable part of area. This paper adopts the digital adder tree as the CIM circuit for two reasons. Firstly, digital CIM shows comparable area and better accuracy compared with analog CIM based on DAC/ADCs. Secondly, it is more convenient to merge IGZO inside digital CMOS circuits to save area.

Since the IGZO transistor can be 3D-stacked over the CMOS transistors thanks to its BEOL fabrication process, we propose to replace the NMOS transistors with 3D-stack IGZO to save the adder tree area. This paper explores the IGZO replacement in a 1-bit full adder, which is the main component of a digital adder tree. Note that to realize the tightly integrated IGZO/CMOS, the IGZO transistor has to be scaled similar to [23] so that the working voltage of IGZO can be compatible with CMOS (e.g. 0.9V).

Fig. 4(a) shows a typical 24T full adder structure with the NMOS transistors marked as $T1$-$T12$. Fig. 4(b) shows the hybrid IGZO/CMOS full adder circuit by simply replacing all NMOS transistors ($T1$-$T12$) with BEOL 3D-stack IGZO. This would directly reduce the number of transistors by 50% but lead to 37.9% additional power due to the longer transistor switch time (lower on-state current) of IGZO.



Solutions	Replaced transistors
All NMOS	None
A-related	*T*1, *T*4, *T*7, *T*11
B-related	*T*3, *T*5, *T*8, *T*12
Ci-related	*T*2, *T*9, *T*10
AB-related	*T*1, *T*3-*T*5, *T*7, *T*8, *T*11, *T*12
BCi-related	*T*2, *T*3, *T*5, *T*8-*T*10, *T*12
ACi-related	*T*1, *T*2, *T*4, *T*7, *T*9-*T*11
ABCi-related	*T*1-*T*5, *T*7-*T*12
All IGZO	*T*1-*T*12

Figure 4: (a) Typical CMOS full adder. (b) IGZO-CMOS hybrid full adder. (c) Different replace solutions and (d) power/area.

Therefore, different types of IGZO replacement solutions are explored, as shown in Fig. 4(c). For example, the '*A*-related' solution replaces $T1$, $T4$, $T7$ and $T11$ with IGZO, whose gate ports are connected to the input signal 'A' of a full adder. While the '*Ci*-related' solution replaces transistors $T2$, $T9$ and $T10$ that are connected to 'Ci'. All the listed IGZO replacement solutions can work at ⩾1GHz frequency in front-end simulation in Cadence Spectre. Fig. 4(d) lists the corresponding power consumption and replaced CMOS transistors. The number of transistors can roughly reflect the area reduction. Among the listed solutions, 12.5%-50% area can be saved with 12.6%-37.9% additional power by replacing NMOS with IGZO. Flexible power/area trade-offs can be made among these replacement solutions. In the experiment part of this paper, the 'All IGZO' solution is selected for evaluation to achieve better area efficiency.

4 PROPOSED IGZO-BASED CIM/CAM MACROS

The multifarious AI applications lead to utilization concerns under fixed CIM/CAM modules. To address this issue, this section proposes four types of CIM/CAM macros based on the 4T0C/3T0C cells, supporting reconfigurable, bare CIM and bare CAM functions.

4.1 Proposed IGZO CIM/CAM macros

The proposed 4T0C/3T0C cells are redrawn in Fig. 5(a), and then inserted into the four macros in Fig. 5(c)-(f). The 4T0C cross-sectional schematic is shown in Fig. 5(b), in which the IGZO transistors are 3D-stack over NMOS/PMOS. Therefore, the 4T0C cell takes the area of NMOS+PMOS transistors. For 3T0C cell, the two IGZO transistors can be stacked in two BEOL layers so that it only needs the area of one NMOS transistor. Figure 5(c)-(f) show four CIM/CAM macros as illustrated below.

Reconfigurable (reconf.) CIM/CAM macro. The $reconf.$ 4T0C macro consists of peripheral drivers, CIM input modules and multiple sub-macros. Each sub-macro contains 4T0C cell arrays, current sense amplifiers(CSA), local SRAM cells and CIM circuits (AND gates, an adder tree). In the sub-macro, each column of 4T0C cells shares WBL and ML, while each row shares WWL, SL and RBL. For CIM operation, each column is connected to a 6T SRAM cell via ML. Weight data stored in one row are readout to the SRAM cells through CSA, which then perform CIM operations with input activation data in the CIM circuits. In this process, the PMOS transistors $T3$ is idle. For CAM operations, the embedding table or graph data are stored in the 4T0C cell array. The search vector is applied to

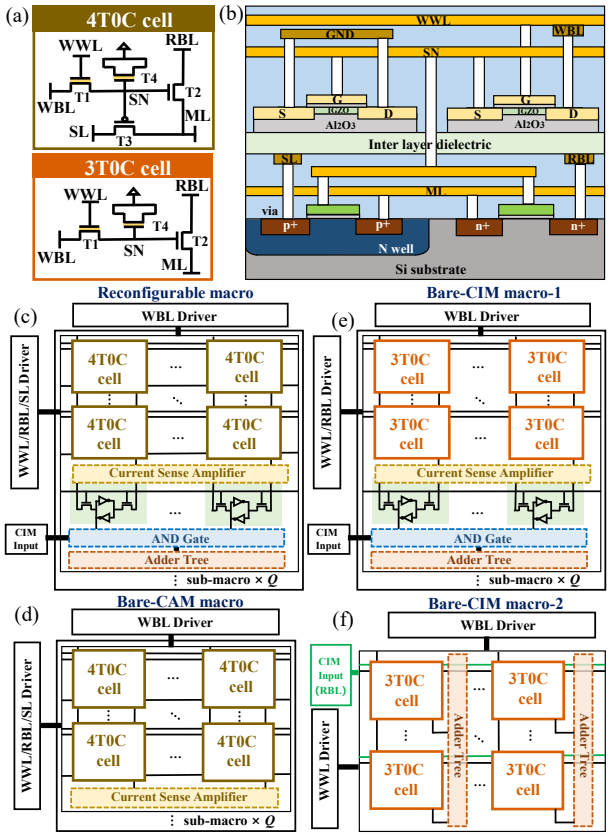


Figure 5: (a) The 4T0C/3T0C cells. (b) 4T0C cross-sectional schematic. (c)-(f) Four types of *reconf*./CIM/CAM macros.

multiple RBLs and SLs to match '0' and '1', respectively. The detailed voltage configurations and waveforms are shown in Section 4.2. The search results are readout through CSAs, which then generate the result address by a priority encoder. All the Q sub-macros work in parallel under both CIM and CAM configurations.

4T0C bare-CAM macro. Compared with the *reconf*. macro, the bare-CAM macro eliminates SRAM cells, AND gates and adder trees for CIM functions. It performs parallel CAM operations similar to the *reconf*. macro. If CAM function is not needed, such as in DNN/Transformer tasks, the bare-CAM macro can serve as a high-density storage for intermediate data to replace conventional SRAM.

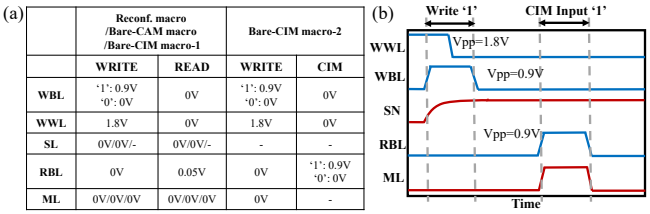
3T0C bare-CIM macro-1. The bare-CIM macro-1 replaces 4T0C cells with 3T0C cells. The local SRAM cells are adopted to store the temporal weight data. Since each weight data would perform CIM operations with different input activations in many cycles, it helps to reduce frequent IGZO read operations to save power.

3T0C bare-CIM macro-2. The difference between bare-CIM macro-1 and macro-2 is whether to share storage cells for one adder tree. Bare-CIM macro-2 connects all the 3T0C cells to adder trees so that CSA is not needed. The input activation can be directly applied to RBL and the AND operation is naturally executed by $T2$. Correspondingly, the adder trees are vertically distributed.

4.2 CIM/CAM configurations and waveforms

This subsection illustrates the detailed voltage configurations and waveforms in CIM/CAM modes of the above four macros.

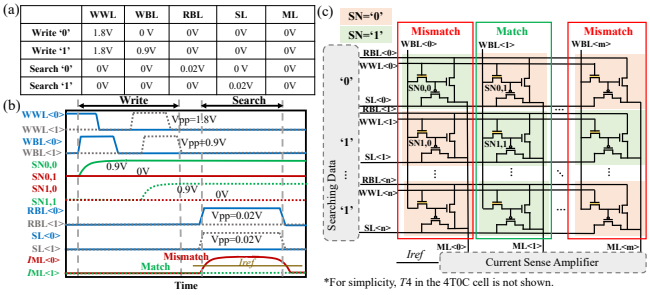
Write/read operation. As shown in Fig. 6(a), the write operations for all four macros are the same. WWL is set to 1.8V so



(a)

	Reconf. macro /Bare-CAM macro /Bare-CIM macro-1		Bare-CIM macro-2	
	WRITE	READ	WRITE	CIM
WBL	'1': 0.9V '0': 0V	0V	'1': 0.9V '0': 0V	0V
WWL	1.8V	0V	1.8V	0V
SL	0V/0V/-	0V/0V/-	-	-
RBL	0V	0.05V	0V	'1': 0.9V '0': 0V
ML	0V/0V/0V	0V/0V/0V	0V	-

Figure 6: Voltage configuration and waveform in CIM mode.



(a)

	WWL	WBL	RBL	SL	ML
Write '0'	1.8V	0 V	0V	0V	0V
Write '1'	1.8V	0.9V	0V	0V	0V
Search '0'	0V	0V	0.02V	0 V	0V
Search '1'	0V	0V	0V	0.02V	0V

Figure 7: Voltage configuration and waveform in CAM mode.

that transistor $T1$ is activated. Then a 0V/0.9V voltage is written from WBL to storage node SN representing '0'/'1', respectively. Only the WWL voltage requires 1.8V, which is isolated from the other CMOS circuits, where CMOS-compatible $\leqslant$0.9V voltage is applied. For read operations in the *reconf*., bare-CAM and bare-CIM macro-1, a 0.05V voltage is applied on the target RBL, while RBLs of other rows and MLs are connected to GND. Then the current ($\sim 10^{-10}$A/$\sim 10^{-5}$A) on ML is sensed by CSA to determine '0'/'1'.

CIM operation. The bare-CIM macro-1 performs CIM with a pre-read operation that stores weights into SRAM cells. For bare-CIM macro-2, read operation is eliminated since it is only used for CIM. A 0V/0.9V voltage for input activation '0'/'1' is applied on RBL, which can directly get the AND result through transistor $T2$. The AND result is sent to the adder tree for CIM operations. The corresponding waveform is presented in Fig. 6(b).

CAM operation. As shown in Fig. 7, for CAM operations, the write ports (WWL, WBL) and MLs are set to 0V, while the search vectors are applied onto RBLs and SLs with 0V/0.02V, respectively. For '0' search, RBL (SL) is set to 0.02V (0V). If the stored value is '1' (mismatch), transistor $T2$ is turned on with a current on ML. Otherwise, there is no current on ML (match). Similarly, for '1' search, RBL (SL) is set to 0V (0.02V). It generates a current on ML if a mismatch happens (with stored value '0' and transistor $T3$ turned on). If the whole vector is matched, there is no current on ML. The ML current is sensed by CSA to readout the CAM result, and then get the match address by a priority encoder. Note that for simplicity, transistor $T4$ in the 4T0C cell is not shown in Fig. 7.

5 OVERALL IG-CRM ARCHITECTURE

Based on the 4T0C/3T0C cells, hybrid adder tree and the four CIM/CAM macros, the reconfigurable IG-CRM architecture is presented in Fig. 8, enabling flexible CIM/CAM workloads.

IG-CRM contains three types of tiles: N CAM tiles, N CIM tiles and $N{\times}M$ *reconf*. tiles, each of which is composed of $P{\times}S$ corresponding macros. The input buffer and intermediate result buffer transfer data among these tiles, while the single-instruction-multiple-data (SIMD) and pooling, activation units perform additional light operations.

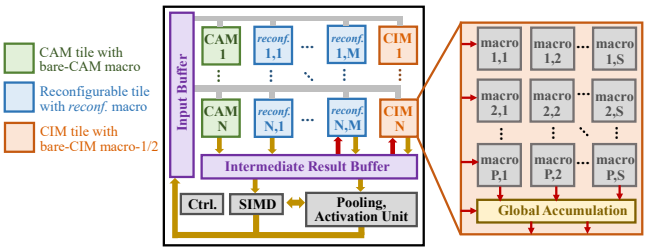


Figure 8: The proposed reconfigurable IG-CRM architecture.

For a given AI application, the search/MAC workloads are offline analyzed and then different ratios of *reconf*. tiles are assigned as CAM/CIM modes for optimal resource utilization. Each tile can be individually configured for CIM/CAM. The CAM tiles perform search operations such as embedding or vertex search. Light addition operations might be required for the gathered features from CAM, which are performed on the SIMD module. Then the intermediate results are sent to CIM tiles for dense MAC computation. The CAM and CIM tiles work in pipeline. For DNN and Transformer applications without CAM operations, all the *reconf*. tiles are set as CIM tiles. The CAM and *reconf*. tiles can also serve as intermediate data storage (such as activation) to replace conventional SRAM, while the idle tiles are power-off.

For CIM operations, the input parallelism and output parallelism can be flexibly adjusted on the 2D interconnections among tiles and macros. For example, for a DNN layer with small input channels, the input parallelism is small, so different rows of macros are mapped with computations of different output channels. The activation height/width can also be divided into multiple blocks for parallel computation. The flexible input/output parallelism helps to increase the resource utilization and reduce the intermediate data access.

6 EVALUATIONS

6.1 Experiment Setup

We evaluate IG-CRM architecture on several AI applications, including DNN (ResNet50[1], ConvNext-T[20] on ImageNet), Transformer (ViT-B/16[2] on ImageNet), RecSys (DLRM[3] on Criteo) and GCNs[4] (with datasets shown in Table 1). All the workloads assign 8bit/8bit computation except for 4bit weight in ResNet50.

Table 2 summarizes the configurations of IG-CRM components. Depending on the selected model sizes, $[N, M, P, S]$ in Fig. 8 is set to $[4, 7, 8, 8]$, which means 4/28/4 CIM/*reconf*./CAM tiles, each with 8×8 macros. Bare-CIM macro-2 is selected for CIM tile. The size of each bare-CAM/*reconf*. macro is 4096×128 (i.e. each 32 rows form a sub-macro), while the bare-CIM macro size is 128×128. We utilize Cadence Spectre to simulate the write/read/CIM/CAM power of 4T0C/3T0C arrays. The CMOS circuits are based on a 28nm CMOS process, while the IGZO model is from [11]. The area of 4T0C/3T0C cells is from the layout of a PMOS+NMOS pair and a single NMOS, respectively, where the IGZO transistors are 3D-stack over CMOS. The total IG-CRM architecture takes $393mm^2$

Table 1: Datasets for GCNs

Datasets	Nodes	Edges	Classes	Features
Cora	2,708	5,429	7	1,433
PubMed	19,717	44,338	3	500
Flickr	89,250	899,756	7	500
Reddit	232,965	11,606,919	41	602

Table 2: Parameters of the IG-CRM architecthure

Component	Config.	Area (mm^2)	Power (W)[1] Write[2]	Read	CIM	CAM
Bare-CAM tile (number: 4×1)						
CAM cell array	4096×128 8×8 macros	13.1	0.0202	0.0117	/	0.822
Peripheral	8×8 macros	5.45	0.0286	0.567	/	0.567
Sum		18.6	0.0488	0.579	/	1.39
Reconfigurable tile (number: 4×7)						
Reconf. cell array	4096×128 8×8 macros	91.6	0.142	0.0819	/	5.75
Peripheral	8×8 macros	38.1	0.200	3.97	/	3.97
IGZO/CMOS adder tree	128×1bit×128 8×8 macros	212	/	/	16.9	/
Sum		342	0.342	4.05	16.9	9.72
Bare-CIM tile (number: 4×1)						
CIM cell array	128×128 8×8 macros	0.205	0.0202	0.0117	/	/
Peripheral	8×8 macros	1.37	0.0286	0.567	/	/
IGZO/CMOS adder tree	128×1bit×128 8×8 macros	30.3	/	/	2.42	/
Sum		31.9	0.0488	0.579	2.42	/
Others (input/output buffer, SIMD, Ctrl., etc.)		0.460	0.0289			
Total Sum		393 mm^2	20.7W (*Recf*. tiles for CIM) 13.5W (*Recf*. tiles for CAM)			

[1] Peak power. Assume all macros are working.
[2] 20MHz frequency for IGZO write. 400MHz for Read/CIM/CAM operations.

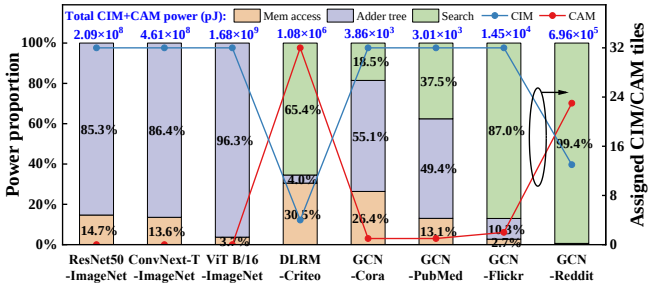


Figure 9: Power and tile proportion under varied workloads.

area. When the *reconf*. tiles are all applied for CIM/CAM, the peak power is 20.7W/13.5W, respectively.

6.2 Results and Analysis

CIM/CAM reconfiguration. IG-CRM can flexibly support different types of AI applications with reconfigurable CIM/CAM tiles. Fig. 9 shows the number of assigned CIM/CAM tiles and power breakdown under different DNN/Transformer/RecSys/GCN workloads. ResNet50, ConNext-T and ViT B/16 require no CAM operations, thus all *reconf*. tiles are assigned to CIM mode. The four CAM tiles are used as intermediate storage. The varied RecSys/GCN workloads show different CIM/CAM configurations, mainly depending on the embedding table or graph storage size. For the DLRM model with 0.12GB embedding table, all *reconf*. tiles are used as CAM tiles. For Reddit model, 19 *reconf*. tiles work in CAM mode while the others work in CIM mode. The sizes of Cora, PubMed, and Flickr are rather small, so the bare-CAM tiles are enough for CAM workloads. In Fig. 9, CIM adder tree, CAM search and memory access are three main parts of power consumption, depending on the CAM/CIM workload ratio and the CIM computation/access ratio.

Area saving. Fig. 10 illustrates the breakdown of area savings. The traditional SRAM-based non-reconfigurable architecture is

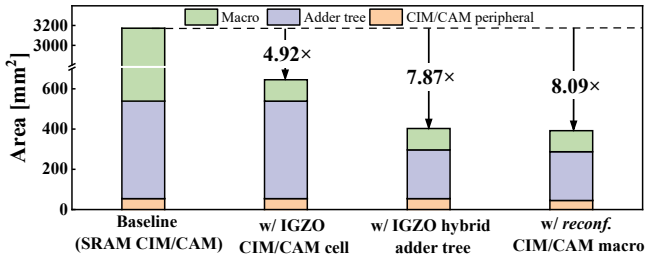


Figure 10: Breakdown of area savings.

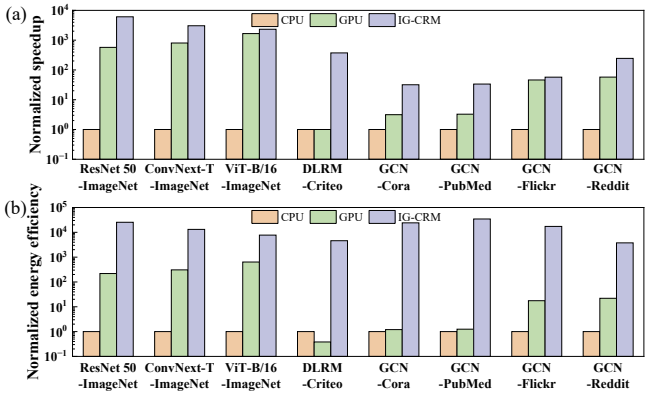


Figure 11: Normalized (a) speedup and (b) energy efficiency compared with CPU/GPU platforms.

adopted as the baseline, which has 32 CIM and 32 CAM tiles (i.e. the same CIM/CAM performance as IG-CRM). Thanks to the 3T0C/4T0C CIM/CAM cells, IGZO hybrid adder tree and *reconf*. CIM/CAM macros, IG-CRM shows 4.92×, 7.87×, and 8.09× area savings. The cell area reduction mainly comes from the replacement of SRAM CAM cells. The hybrid IGZO adder tree barely saves 50% of the adder tree area thanks to the BEOL-compatible characteristic. The *reconf*. architecture further saves the peripheral circuits area.

Speedup and energy efficiency. Fig. 11 shows the normalized speedup and energy efficiency compared with CPU and GPU platforms. The Intel i7-11700F CPU and NVIDIA RTX3060 GPU are selected as baselines. IG-CRM shows 3.8×-10.6× speedup over GPU on DNNs thanks to higher CIM parallelism. The speedup ratio is a bit lower on Transformer due to the insufficient utilization of small matrix operations. Due to the low utilization for graph search or embedding operations on GPU, IG-CRM shows higher speedup on several ResSys and GCNs with parallel CAM operations (1-2 orders of magnitude speedup than GPU on Criteo, Cora and PubMed datasets). Due to the low power consumption of IGZO-based CIM/CAM circuits, the energy efficiency improvement compared with CPU/GPU is higher than the speedup. Overall, IG-CRM achieves 31.8×-6.09×10^3× (average: 1.53×10^3×) speedup and 3.77×10^3×-3.45×10^4× (average: 1.63×10^4×) energy efficiency improvement over CPU, 1.24×-3.73×10^2× (average: 51.9×) speedup and 12.1×-2.76×10^4× (average: 7.62×10^3×) energy efficiency improvement over GPU.

7 CONCLUSION

This work presents IG-CRM, an area/energy-efficient IGZO-based reconfigurable CIM/CAM architecture, including IGZO 3T0C/4T0C CIM/CAM circuits, hybrid IGZO/CMOS 3D-stack adder tree, and reconfigurable CIM/CAM macros. IG-CRM can efficiently support multifarious AI applications with different CIM/CAM workloads in one chip, and significantly save area thanks to the BEOL-compatible characteristic of IGZO and reconfigurable modules. Experiment results show that IG-CRM achieves 1.53×10^3×/51.9× speedup and 1.63×10^4×/7.62×10^3× energy efficiency on average compared with CPU and GPU platforms.

ACKNOWLEDGMENTS

This work was supported in part by National Key R&D Program of China 2022YFB3606902; NSFC Grant 62204256, 61888102, 92264204, 62274178, 61720106013, 61904195, and 62004214; Beijing Nova Program Z211100002121125; China Postdoctoral Science Foundation BX20220330 and 2021M703444; and the Strategic Priority Research Program of Chinese Academy of Sciences Grant XDB44000000.

REFERENCES

[1] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.
[2] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).
[3] Maxim Naumov et al. "Deep learning recommendation model for personalization and recommendation systems". In: *arXiv preprint arXiv:1906.00091* (2019).
[4] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016).
[5] Chuan-Jia Jhang et al. "Challenges and Trends of SRAM-Based Computing-In-Memory for AI Edge Devices". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 68.5 (2021), pp. 1773–1786.
[6] Kostas Pagiamtzis and Ali Sheikholeslami. "Content-addressable memory (CAM) circuits and architectures: A tutorial and survey". In: *IEEE journal of solid-state circuits* 41.3 (2006), pp. 712–727.
[7] Yu-Der Chih et al. "An 89TOPS/W and 16.3TOPS/mm2 all-digital SRAM-based full-precision compute-in memory macro in 22nm for machine-learning edge applications". In: *IEEE International Solid-State Circuits Conference(ISSCC)*. 2021.
[8] Cheng-Xin Xue et al. "A 22nm 2Mb ReRAM compute-in-memory macro with 121-28TOPS/W for multibit MAC computing for tiny AI edge devices". In: *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE. 2020.
[9] A Belmonte et al. "Capacitor-less, long-retention (> 400s) DRAM cell paving the way towards low-power and high-density monolithic 3D DRAM". In: *2020 IEEE International Electron Devices Meeting (IEDM)*. IEEE. 2020, pp. 28–2.
[10] Xinlv Duan et al. "Novel vertical channel-all-around (CAA) In-Ga-Zn-O FET for 2T0C-DRAM with high density beyond 4F^2 by monolithic stacking". In: *IEEE Transactions on Electron Devices* 69.4 (2022), pp. 2196–2202.
[11] Jingrui Guo et al. "A new surface potential and physics based compact model for a-IGZO TFTs at multinanoscale for high retention and low-power DRAM application". In: *IEEE International Electron Devices Meeting (IEDM)*. IEEE. 2021.
[12] Ping Chi et al. "Prime: A novel processing-in-memory architecture for neural network computation in reram-based main memory". In: *ACM/IEEE International Symposium on Computer Architecture (ISCA)* (2016).
[13] Linghao Song et al. "Pipelayer: A pipelined reram-based accelerator for deep learning". In: *IEEE high performance computer architecture (HPCA)*. IEEE. 2017.
[14] Lixia Han et al. "A Convolution Neural Network Accelerator Design with Weight Mapping and Pipeline Optimization". In: *2023 60th ACM/IEEE Design Automation Conference (DAC)*. IEEE. 2023, pp. 1–6.
[15] Nagadastagiri Challapalle et al. "Crossbar based processing in memory accelerator architecture for graph convolutional networks". In: *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. IEEE. 2021.
[16] Nagadastagiri Challapalle et al. "GaaS-X: Graph analytics accelerator supporting sparse data representation using crossbar architectures". In: *2020 ACM/IEEE International Symposium on Computer Architecture (ISCA)*. IEEE. 2020.
[17] Linghao Song et al. "GraphR: Accelerating graph processing using ReRAM". In: *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE. 2018, pp. 531–543.
[18] Tao Yang et al. "PIMGCN: a reram-based PIM design for graph convolutional network acceleration". In: *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE. 2021, pp. 583–588.
[19] Mengyuan Li et al. "Imars: An in-memory-computing architecture for recommendation systems". In: *Proceedings of the 59th ACM/IEEE Design Automation Conference*. 2022, pp. 463–468.
[20] Zhuang Liu et al. "A convnet for the 2020s". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 11976–11986.
[21] Prithviraj Sen et al. "Collective classification in network data". In: *AI magazine* 29.3 (2008), pp. 93–93.
[22] Ali Keshavarzi et al. "Ferroelectronics for edge intelligence". In: *IEEE MICRO* 40.6 (2020), pp. 33–48.
[23] Menggan Liu et al. "Analog Monolayer MoS2 Transistor with Record-high Intrinsic Gain (> 100 dB) and Ultra-low Saturation Voltage (< 0.1 V) by Source Engineering". In: *2021 Symposium on VLSI Technology*. IEEE. 2021, pp. 1–2.