Heterogeneous Multi-Functional Look-Up-Table-based Processing-in-Memory Architecture for Deep Learning Acceleration

Sathwika Bavikadi*, Purab Ranjan Sutradhar†, Amlan Ganguly† and Sai Manoj Pudukotai Dinakarrao*
* Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA, USA
† Department of Computer Engineering, Rochester Institute of Technology, Rochester, NY, USA.
*{sbavikad, spudukot}@gmu.edu, †{ps9525,axgeec}@rit.edu

2023 24th International Symposium on Quality Electronic Design (ISQED) | 979-8-3503-3475-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISQED57927.2023.10129338

***Abstract*—Emerging applications including deep neural networks (DNNs) and convolutional neural networks (CNNs) employ massive amounts of data to perform computations and data analysis. Such applications often lead to resource constraints and impose large overheads in data movement between memory and compute units. Several architectures such as Processing-in-Memory (PIM) are introduced to alleviate the bandwidth bottlenecks and inefficiency of traditional computing architectures. However, the existing PIM architectures represent a trade-off between power, performance, area, energy efficiency, and programmability. To better achieve the energy-efficiency and flexibility criteria simultaneously in hardware accelerators, we introduce a multi-functional look-up-table (LUT)-based reconfigurable PIM architecture in this work. The proposed architecture is a many-core architecture, each core comprises processing elements (PEs), a stand-alone processor with programmable functional units built using high-speed reconfigurable LUTs. The proposed LUTs can perform various operations, including convolutional, pooling, and activation that are required for CNN acceleration. Additionally, the proposed LUTs are capable of providing multiple outputs relating to different functionalities simultaneously without the need to design different LUTs for different functionalities. This leads to optimized area and power overheads. Furthermore, we also design special-function LUTs, which can provide simultaneous outputs for multiplication and accumulation as well as special activation functions such as hyperbolics and sigmoids. We have evaluated various CNNs such as LeNet, AlexNet, and ResNet-18,34,50. Our experimental results have demonstrated that when AlexNet is implemented on the proposed architecture shows a maximum of 200× higher energy efficiency and 1.5× higher throughput than a DRAM-based LUT-based PIM architecture.**

I. Introduction

The rapid advancements in hardware fabrication and integration, along with the software applications, lead to the development of various fields, including computer vision, image processing, artificial intelligence (AI) and natural language processing. These emerging applications led to an eventual increase in the demand for performance and efficiency, along with the data to be observed and analyzed. Machine learning (ML) and deep learning (DL) are introduced as a panacea to process and analyze such vast amounts of data [1]–[3].

To meet hardware efficiency and other performance requirements, several architectural innovations have been proposed in recent years. Custom-designed accelerators such as application-specific integrated circuits (ASICs) [4] though are energy efficient and optimized, they have extremely low flexibility. On the other hand, Field-programmable gate array (FPGA) [5], [6] accelerators address the programmability challenges but are hindered by low energy efficiency, complexity, and volatility challenges. For executing DL/ML applications, central processing units (CPUs) are less energy-efficient than ASICs. Thus, conventional von Neumann architecture-based computing systems, including general-purpose processors (GPPs), central processing units (CPUs), and graphics processing units (GPUs) [7] have extremely low energy efficiency and latency [2], [8]. This excessive cost of computing efficiency [2], [8] is associated with the expensive memory access and data movement caused by the physical separation between the processing unit and the memory unit inside a conventional von-Neumann architecture.

Computing architectures such as 'non-von Neumann' architectural paradigms [9] including processing-in-memory (PIM) a.k.a In-Memory Computing (IMC), near-data processing (NDP), are introduced to alleviate data transfer bottleneck [10]. IMC architectures [11] perform the computations on the memory chip itself and exhibit higher energy efficiency compared to other paradigms due to its intra-memory communication and computations. Numerous PIM designs are implemented on a wide range of emerging memory technologies such as traditional volatile static random access memory (SRAM) [12] and dynamic random access memory (DRAM) [11], [13]–[16], as well as non-volatile memory technologies like Resistive RAM (ReRAM) [17], and Magnetic RAM (STT/SOT-MRAM) [18]. However, DRAM is the most widely used memory technology for manufacturing external memory devices due to its higher memory density, lower power consumption, and lower cost of production compared to other memory technologies [19].

To overcome the limited processing speed of IMC, look-up-table (LUT)-based PIMs have emerged as a panacea [6], [20]. Numerous works have been introduced in recent years that use memory LUTs for performing arithmetic and logical computations [14], [16]. Although several designs propose implementing PIM architectures utilizing the LUTs, the existing architectures are confined to specific applications and operations i.e., lacks the flexibility to be adapted to other applications and programmability. The LUTs and PIM systems are designed to support only one type of functionality and only one type of application, either a compute-intensive or memory-

Authorized licensed use limited to: George Mason University. Downloaded on September 01,2023 at 00:08:55 UTC from IEEE Xplore. Restrictions apply.

intensive application with limited performance when executing other types of applications [20]. Therefore a flexible hardware platform that supports a variety of CNN/DNN operations is required.

To address these challenges and offer a larger degree of functional flexibility and programmability, *we introduce a DRAM-based multi-functional look-up-table-based reconfigurable PIM architecture that supports existing and emerging applications with low overheads and high programmability*. This proposed architecture consists of multiple clusters embedded with many heterogeneous reconfigurable LUT cores. Each cluster comprises three types of LUT cores: ALU LUT core, special ALU (S-ALU) LUT core, and special-function (SF) LUT core.

Unlike the existing works [14], [16], [21]–[23], the proposed LUT cores are heterogeneous multi-functional special LUT cores i.e., each of these cores are capable of performing distinct operations from each other and can provide multiple outputs corresponding to multiple functionalities in a multiplexed manner, thereby called multi-functional LUTs. This approach not only provides a reduced number of LUTs but also increases the utilization efficiency and functional support offered by LUTs. The ALU-LUT cores are specifically programmed to implement the MAC operations in the PIM. The special ALU (S-ALU) LUTs can provide multiple outputs relating to different functionalities simultaneously without the need to design different LUTs for different functionalities. For instance, S-ALU LUT cores can be programmed to do both multiplication and addition on the same given input in a single clock cycle. Thus providing the output of both operations without the need of programming two cores separately to do multiplication and addition operations. This leads to optimized area and power overheads. Finally, the special-function (SF) LUTs are designed to implement special-function operations such as hyperbolics and sigmoid, ReLU operations. In order to provide inherent computing support for MAC operations, activation operations such as sigmoid, hyperbolic, and ReLU, nine LUT core design exploration in a cluster is adapted.

To summarize, the novel contributions of this work are:

- We propose a novel heterogeneous multi-functional look-up-table-based reconfigurable processing-in-memory architecture to address the energy efficiency and flexibility criteria for computing architectures.
- Presenting a flexible architecture by introducing reconfigurable LUTs capable of performing multi-functional operations required to process different layers of a neural network for CNN acceleration.
- We propose special heterogeneous multi-function LUTs capable of producing multiple outputs for multiplication, accumulation, sigmoid, and hyperbolic, ReLU operations.
- We proposed three different kinds of LUT cores with different functionality: ALU LUT core, S-ALU LUT core, and SF-ALU LUT core, which are specially designed to tackle multi-functional operations required for various CNN acceleration.
- We evaluate the proposed architecture on various CNN architectures including LeNet, AlexNet, ResNet-18, -34, -50, and show that it outperforms the state-of-the-art techniques in terms of throughput, energy efficiency, and accuracy.

II. Background and Related Works

Deep Neural Network algorithms are dominated by a large number of simplistic, data-parallel computations, such as convolutions and matrix multiplications. These operations can be executed with a very high level of operational parallelism in the hardware. Non-von Neumann architectures such as Processing-in-Memory also known as In-Memory Computing devices are being widely investigated for DNN/ CNN applications in recent times. PIM architectures are able to perform massively parallel simple computations at surprisingly low latency and high energy efficiency. PIM architectures are memory-centric architectures [11], [13], which are entirely implemented on a memory chip. PIM devices have been demonstrated to offer better parallel performance than most CPUs and ASIC devices [7], as well as, better energy efficiency than GPUs [16]. This virtually eliminates the data bandwidth bottleneck of off-chip communications, otherwise suffered by state-of-the-art processing devices [2].

Recently, numerous works have been proposed on in-memory computing hardware accelerators using conventional CMOS and emerging memory devices. To overcome the large latency overheads due to the frequent data transfer between memory-logic units, IMC is seen as an efficient alternative for executing data-intensive ML applications. Despite efficiency in terms of energy consumption, in-memory computations including addition and multiplication operations are orders of magnitude slower compared to the traditional CMOS-based hardware accelerators. In addition to the large area and power overheads of the DRAM-based in-memory accelerators, they require significant modifications to the memory-bank architectures such as activation of multiple rows, high precision timers, and novel sense amplifiers to enable efficient IMC.

With the emergence of non-volatile memory (NVM), NVM-based in-memory computing techniques are introduced and adopted in academia and industry. The NVMs achieve higher integration densities i.e., low area, offer better scalability, and lower power consumption compared to the standard DRAM technology. This makes NVMs an ideal candidate for the design of hardware accelerators in this work. There exist multiple emerging NVMs which can potentially replace their CMOS counterparts such as ReRAM [17], Phase-Change Memory (PCM), Spin-Transfer Torque (STT)-MRAM [18], and Spin-Orbit Torque (SOT)-MRAM [24] technologies. Numerous IMC hardware accelerators that support ML applications are introduced in the literature [20]. However, due to the low voltage operation, asymmetric read/write current of emerging NVMs cause noise margin issues and are highly vulnerable to reliability concerns, and are not a viable option for CNN acceleration.

A majority of the IMC works [11], [13]–[15] focus on performing faster computations and do not consider the reconfigurability and networking concerns of the accelerators. However, the functionality of these architectures is almost

Authorized licensed use limited to: George Mason University. Downloaded on September 01,2023 at 00:08:55 UTC from IEEE Xplore. Restrictions apply.

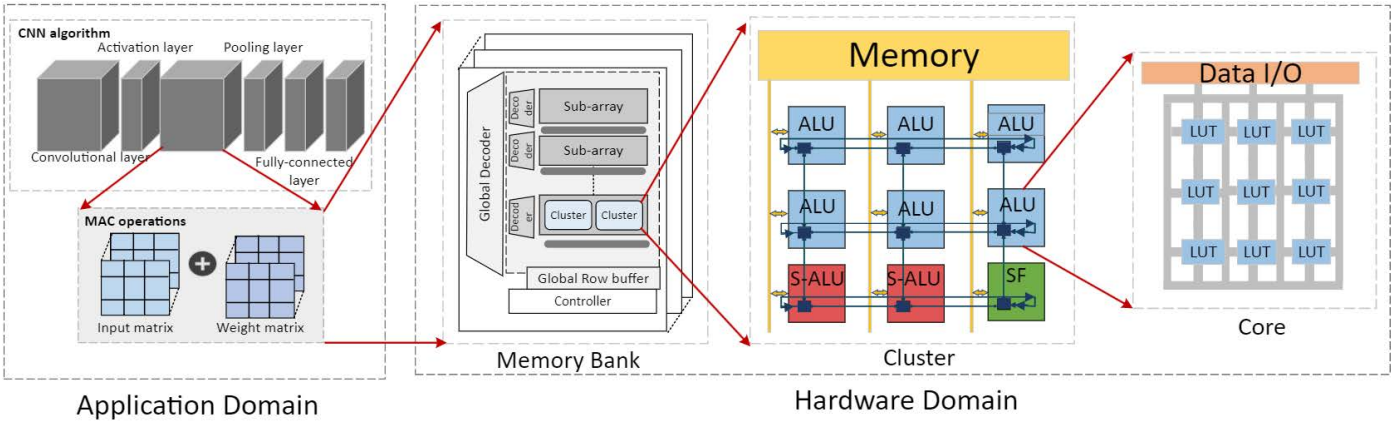

Fig. 1. Hierarchical Architecture showing the cluster arrangement and multi-functional heterogeneous core organization inside the cluster.

exclusively limited by their application, reconfigurability, overheads, latency, and inference of CNN/DNNs. To overcome the aforementioned challenges, the proposed work introduces a multi-functional LUT-based reconfigurable PIM architecture to achieve high-speed reconfiguration for accelerating various ML algorithms.

III. Proposed Multi-functionality LUT-based Heterogeneous DL Accelerator Architecture

Figure 1 shows the hardware architecture of the proposed heterogeneous multi-functional LUT-based reconfigurable DL accelerator. The reconfigurable LUTs are capable of supporting different precision data, and fewer LUTs are required for reduced precision operations. Using lower precision LUT for computational operations leverages improved latency and energy efficiency without compromising the accuracy of CNN algorithms. This architecture is composed of multiple clusters, each cluster comprises nine reconfigurable heterogeneous cores which facilitate multi-functional programmable operations on a pair of 4-bit or a single 8-bit input data. We chose this precision as most computer vision applications perform reliably at this precision with minimal accuracy loss compared to higher precision [25]. Nine of the reconfigurable heterogeneous cores consist of special multi-functional LUTs (ALU-LUT, S-ALU-LUT, SF-LUT), that are grouped together and interconnected by a router to form a single cluster. Each cluster can be programmed to perform a wide range of operations such as multiply and accumulate, substitution, comparison, bit-wise logic operations, hyperbolics, sigmoid, and ReLU activation operations. Therefore, an array of these clusters can be utilized to implement different layers of CNNs and DNNs such as Convolutional Layers, Fully-connected Layers, Activation, and Pooling layers for various CNN inference applications.

A. LUT Core Architecture

The primary goal of the proposed nine LUT core design explorations in a cluster is to facilitate intrinsic computational support to perform MAC operations, sigmoid, hyperbolic, and ReLU operations. The LUT-based design approach for our PIM core provides functional flexibility to configure the core's to

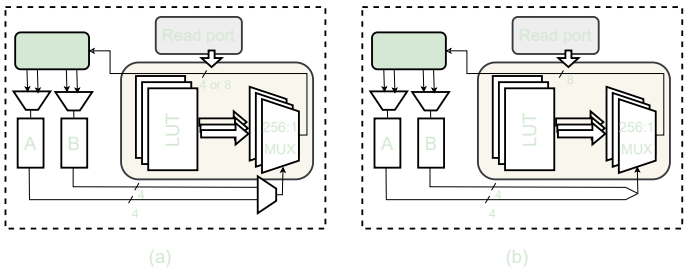

Fig. 2. Microarchitectures of Heterogenous LUT-based PIM cores (a) Microarchitecture of ALU-LUT core and SF-LUT core (b) Microarchitecture of S-ALU-LUT core

do any arbitrary operation. The LUTs are implemented using 8-bit 256-to-1 multiplexers. For example, in order to perform an activation operation with an 8-bit operand, the 8-bit MUX in the PIM core is used to perform a look-up operation and provide 8-bit output. Each LUT core can either support a single 8-bit operand or a pair of 4-bit operands in order to perform operations. Consequently, our proposed heterogeneous multi-functional LUT cores can perform any kind of in-memory computations utilized to implement different layers of a neural network for ML acceleration. Functionalities of the proposed cluster and core design are discussed in Figure 1, referring to corresponding color codes.

ALU-LUT core (Core 1 to 6): The blue squares in Figure 1 represent the multi-functional LUT-based PIM cores, that are programmed to perform 4-bit AND or XOR operations on a pair of 4-bit data input and provide 4-bit output. A multiplexer is used to select the functionality required for the different operations of the CNN algorithm, to either perform XOR or AND operation on the inputs as shown in Figure 2 (a). Based on the multiplexer input, the multi-functional core performs either AND or XOR operation on the input data. The cluster is accommodated with 6 of the ALU-LUT cores.

S-ALU LUT core (Core 7 and 8): The second kind of core used in the cluster, represented in red squares in Figure 1, are the special LUT-based PIM cores. The cluster contains two of these cores, that are programmed such that the output

Authorized licensed use limited to: George Mason University. Downloaded on September 01,2023 at 00:08:55 UTC from IEEE Xplore. Restrictions apply.

consists of two entirely different operations (XOR and AN on the same pair of inputs. Despite the fact that the S-AL LUT core supports the same operations (XOR and AND) the ALU-LUT core, its functionality is entirely different. T core is used in a special scenario when we need both X(and AND operations for the same input data, mainly used the accumulation process. This core is programmed to prod 8-bit output data for a pair of 4-bit inputs, the upper half of core output represents the 4 bits XOR operation of the in data while the lower half represents the 4-bit AND operat of the same input data as shown in Figure 2 (b). Thus, with the need to create separate LUT cores for various purposes, t unique S-ALU-LUT core may deliver several outputs pertain to different functionality concurrently.

SF LUT core (Core 9): The third kind of heterogene core used in the proposed architecture is represented in green square in Figure 1. The cluster contains only one these special multi-functional LUT-based PIM cores, wh is programmed to perform 8-bit special-function activat operations such as sigmoid, hyperbolic, and ReLU using 8- LUT cores. Similar to the ALU-LUT core a multiplexer is u to select the different activation operations to be implemented in SF-LUT as shown in Figure 2 (a). This core is programmed to produce 8-bit output on 8-bit input. Based on the multiplexer, the multi-functional core performs either sigmoid, hyperbolic, or ReLU activation operation on the input.

Each of these cores is capable of performing distinct operations from each other and can provide multiple outputs corresponding to multiple functionalities in a multiplexed manner, thereby called heterogeneous multi-functional LUTs. This provides functional flexibility for the PIM to support various operations required for CNN acceleration. The ALU-LUT and S-ALU-LUT cores are specifically programmed to implement the MAC operations in the PIM. Whereas the SF-LUT core is designed to implement special-function activation operations such as hyperbolics and sigmoid, ReLU operations. Therefore, with the proposed nine LUT core design explorations in the cluster, the proposed PIM can support the computational support required for CNN acceleration.

B. Cluster Architecture

As shown in Figure 1, the cluster formed by nine LUT cores is placed inside the memory banks in order to allow the quickest access to the memory data and to perform the in-memory operation with significantly lower latency. Nine cores in the proposed PIM cluster constitute six ALU-LUT cores that support either AND or XOR operations and two cores S-ALU-LUT cores that can perform both AND and XOR operations for the given input data. Whereas the SF-LUT core is programmed to support activation operations such as hyperbolic, sigmoid and ReLU operations.

Nine of these heterogenous muti-functional LUT cores inside the cluster are programmed in a specific way, interconnected by a routing mechanism in order to perform complex operations such as MAC operations, sigmoid, hyperbolic, and ReLU operations required for CNN acceleration. These operations can be performed in a multi-staged pipeline by organizing a series

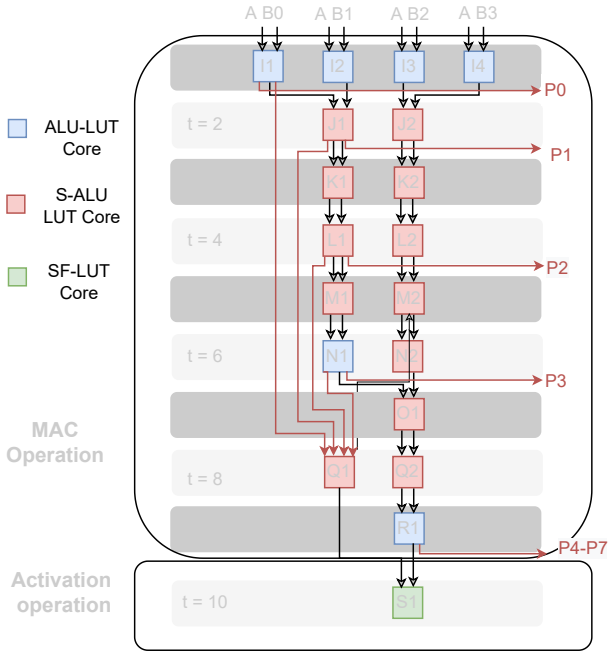

Fig. 3. Overview of the dataflow for MAC operation in Muti-functional Heterogeneous PIM architecture

of micro-operations across the nine LUT cores with the help of a routing mechanism.

The distribution of the operands during every single stage in the operational stage is performed with the help of the router. The router enables parallel communication by connecting every component of the cluster including the cores and the read/write ports. The router is used to connect all the cores, in order to access any core data at any point of time during the execution. The router plays a vital role during the implementation. The memory read/write buffer of the cluster is used to read the data input from the memory and write outputs back into the memory, in order to perform the required operations for CNN acceleration.

The data communication among clusters inside the memory chip is achieved through the routing mechanism. This makes the proposed architecture easily distribute a particular task among multiple clusters. At the same time, different clusters inside the memory bank execute parallel and independent tasks in a single instruction multiple data (SIMD) fashion.

IV. Operations Supported by the Proposed Heterogeneous Multi-functional Architecture

The main benefit of the proposed architecture is that its LUTs can be programmed to implement virtually any type of computation. This equips it with the functional flexibility required for implementing different operations required by various DL applications such as linear algebraic operations, activation, and pooling operations. Among the nine heterogeneous LUT cores, 8 (6 ALU-LUT cores, 2 S-ALU-LUT cores) of them are used for performing MAC operations, and the remaining 1 (SF-LUT core) is designed to implement activation operations using the memory look-up approach. These operations are carried

Authorized licensed use limited to: George Mason University. Downloaded on September 01,2023 at 00:08:55 UTC from IEEE Xplore. Restrictions apply.

out within the cluster by executing a multi-stage pipeline of the nine heterogeneous LUT cores, coupled together with a routing mechanism. Since each core is capable of performing operations on a pair of 4-bit, or a single 8-bit operand. The MAC operations are performed on pair of 4-bit data in parallel to obtain the output of 8-bit inputs using the ALU-LUT and S-ALU LUT cores. Later the output of the MAC operation is passed to the SF LUT cores to perform activation functions such as sigmoid, hyperbolic, and ReLU operations.

In order to perform the Multiplication and Accumulation operation on two 4-bit data operands, initially both the input data, A and B are split into sections A_3, A_2, A_1, A_0, and B_3, B_2, B_1, B_0 respectively. The 4-bit multiplication is performed similarly to decimal multiplication. As demonstrated in Figure 3, a special routing mechanism is used to perform the MAC operation in a multi-stage pipeline. Figure 3 also illustrates how each process in the dataflow has been assigned a special tag consisting of a letter and a number for ease of implementation and testing. Numbers 0, 1, 2, and 3 denote various parallel operations carried in each clock cycle, whereas letters I, J, K, L, M, N, O, Q, R, and S denote the clock steps of LUT operations, P0-P7 represent the MAC operation output. During the runtime, P0-P7 of the MAC operation is accumulated using the S-ALU-LUT core and passed to the SF-LUT core to do the activation operation.

The MAC operation inside the cluster is implemented in a combinational circuit manner by utilizing the LUT cores such that the multiplication is implemented using a series of AND logic operations performed by the ALU-LUT cores and accumulation process by the S-ALU LUT cores as shown in Figure 3. Utilizing the multi-functional S-ALU LUT instead of ALU-LUT for the accumulation process improves the area, power, and latency overheads of the proposed architecture. To further improve core utilization, overlapping of two consecutive accumulations in parallel for executing the MAC operation is enabled.

For the 4-bit input A and B, partial products are obtained by multiplying each bit of input B with the entire 4-bit of input A operand. The first partial product is obtained by multiplying B_0 with A_3, A_2, A_1, A_0, and the second partial product is formed by multiplying B_1 with A_3, A_2, A_1, A_0 likewise for third and forth partial products. So these partial products can be implemented with AND operator using ALU-LUT core as shown in Figure 3. The ALU-LUT core takes two 4-bit input operands and performs logical AND operations using the LUTs to provide 4-bit output. All these operations can be performed in a single clock cycle during the execution. These partial products are then added by using 4-bit S-ALU LUT cores to parallelize the addition process. The first partial product is added to the second partial product, then this result is added to the next partial product with carry-out and it goes on till the final partial product. Finally, it produces an 8-bit output which indicates the MAC value of the two 4-bit input operands. A combined multiplication and addition process can be executed in a 9-clock cycle pipeline as shown in Figure 3.

The output of the MAC operation is passed to the multi-functional SF-LUT core to implement activation functions. A multiplexer is used to select the different activation operations to be implemented in the SF-LUT. Based on the input from the multiplexer, the multi-functional core performs either sigmoid, hyperbolic, or ReLU activation operations on the input data. This operation can be performed in a single clock cycle during the execution. The router is used to enable the chain of operations required for MAC and activation operations inside the cluster. The key advantage of the proposed architecture is that it enables a special routing scheme, and parallelization process in order to efficiently utilize the cores inside the cluster. Moreover, it can be said that the LUTs in the proposed architecture are capable of reprogramming at run-time to perform complex computational operations to implement CNN at ultra-low latency.

V. Evaluation

A. Design Verification

We verified the architecture using ASIC *via* Verilog HDL implementation. We evaluate the performance using different metrics (*such as* operational latency, power consumption and active area) from HDL synthesis on Synopsys Design Compiler using 28 nm standard cell library from TSMC and are presented in Table I. Within a cluster, a single 8-bit MAC requires computations inside PIM cores as well as communication across cores, which adds to the delay. Whereas, the cluster's power consumption is equal to the sum of each core's as well as the core-to-core communication. The power and delay for intra and inter-subarray data transfers are obtained from [15] and [26]. These metrics are used in the system-level performance evaluation.

TABLE I
CHARACTERISTICS OF MULTI-FUNCTIONAL HETEROGENEOUS HARDWARE ACCELERATOR AND ITS COMPONENTS IN 28 NM TECHNOLOGY NODE

Component	Delay (ns)	Power (mW)	Active Area(μm^2)
ALU-LUT Core	0.10	0.00177	8010
S-ALU-LUT Core	0.26	0.00497	13210
SF-LUT Core	0.7	0.01853	141304
Multi-functional Heterogeneous Cluster	1.62	0.05539	199764
LUT Core [16]	0.8	2.7	4196.64
LUT Cluster (MAC Operation) [16]	6.4	8.2-11	37769.81
Intra-Subarray Communication [26]*	63.0	0.028 μJ/comm	N/A
Inter-Subarray Communication [15] for subarrays 1/7/15 hops away*	148.5/ 196.5/ 260.5	0.09/ 0.12/ 0.17 μJ/comm	N/A

**Represented in 28nm technology node*

Firstly from Table I, it is observed that due to the different operational support provided by heterogeneous cores, they have different delay, area, and power metrics. Since the SF-LUTs process 8-bit data on 8-bit memory LUTs, which is different from the ALU and S-ALU cores, the SF-LUT has the highest delay, area, and power consumption. The ALU-LUT core is designed to process a pair of 4-bit data on 4-bit memory LUTs and has the least delay, area, and power consumption. However, compared to the LUT core [16], the proposed cores have

Authorized licensed use limited to: George Mason University. Downloaded on September 01,2023 at 00:08:55 UTC from IEEE Xplore. Restrictions apply.

relatively less delay and power consumption, but the active area is 2× greater. However, the proposed heterogeneous PIM core can provide multiple functionalities simultaneously, whereas multiple traditional LUT cores are required to provide multiple functionalities. This indicates the increased area overheads can be well justified and minimal for systems that perform complex operations.

Nine of these cores are grouped together as discussed in Section III-A, forming a single cluster. From the system-level perspective, the PIM requires 256 of these PIM clusters in order to perform computational operations for 8-bit data precision. In order to facilitate that, we consider infusing one PIM bank with 256 PIM clusters per DRAM chip in the entire rank of the DRAM chips for a DIMM (dual in-line memory module)

For the cluster characteristics when implementing the 8-bit MAC operation and activation operation on the proposed architecture, the delay is observed to be 1.62 ns, whereas for the LUT core [16] to perform just the MAC operation the delay is 6.4 ns. Which is almost 4× faster implementation of MAC operation on multi-functional cores compared to the LUT core [16]. Therefore, it is observed that the multi-functional architecture is highly suitable for ultra-low latency, low-power applications such as real-time IoT devices, and edge devices. Even though it is observed that the proposed architecture has more area than IMC LUT-based design [16], it is still observed to achieve a lower area in the case of edge devices.

B. Performance Evaluation

In this subsection, we perform a comparative performance analysis of the proposed architecture in terms of throughput and energy efficiency on LeNet, AlexNet, ResNet-18, -34, and -50 CNN algorithms for a batch size of 64. Energy efficiency is defined as the number of frames processed in the processor per unit of energy (Joules). Figure 4 presents comparisons of the throughput (in Frames per second) and energy efficiency (in Frames per Joule) of inference on all these CNNs deployed on the proposed multi-functional heterogeneous architecture.

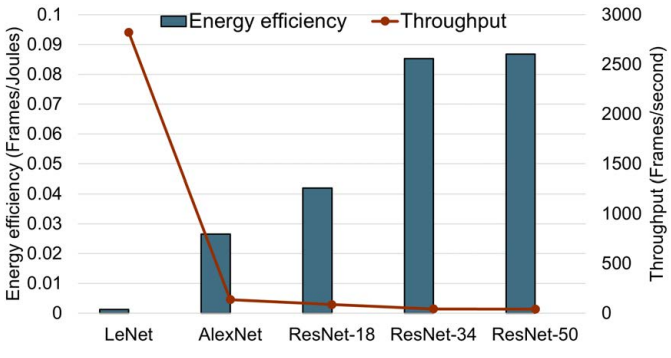

Fig. 4. Comparison of Energy efficiency (Frames/Joules) and Throughput (Frames/second) for LeNet, AlexNet, ResNet18, ResNet34, ResNet50 on the proposed multi-functional heterogeneous architecture

Firstly, Figure 4 shows the energy efficiency of the CNN algorithms is proportional to the depth of the network. As the number of layers increases, more MAC, activation operations are needed to be performed which implies more parallelization to perform these operations. Therefore, for a higher number of layers in the CNN algorithm, the energy efficiency achieved is high. It is observed that LeNet, AlexNet, and ResNet 18 achieved the inference energy efficiency of 0.0011 Frames/Joule, 0.024 Frames/Joule, and 0.038 Frames/Joule respectively.

Figure 4 also shows that the proposed architecture achieves better performance for CNN algorithms with a comparatively lower computational workload such as LeNet. However, for AlexNet with 8 layers, the proposed architecture achieves an inference throughput of 150.3 Frames/s and 50 layered ResNet algorithm achieves an inference throughput of 45.9 Frames/s. Therefore it can be said that the proposed architecture can achieve impressive performance while implementing MAC, activation operations, for the convolutional layers in the CNN/DNNs to process very efficiently. For instance, ResNet-50, the largest network implemented on the proposed architecture consists of 50 layers with thirty-eight billion computations that can be processed within 10 ms on the proposed architecture.

C. Inference Accuracy

We evaluate on our proposed architecture for various state-of-the-art deep neural networks such as LeNet [27], AlexNet [1], ResNet -18,-34,-50 [28]. These deep learning algorithms are implemented on the proposed hardware accelerator using MNIST [29] (28×28×1 dimensions), CIFAR-10 [30] dataset (32 x 32 x 3 dimensions). Figure 5 shows the Top 5 accuracy comparison plots for 16-bit floating-point (FP) and 8-bit fixed-point data precision for both datasets. It is observed that the accuracies obtained on the evaluated networks are very similar for 16-bit and 8-bit precision data (inputs and weights). The Top-1 accuracy obtained for the MNIST dataset when implemented on AlexNet is 98.89% and 99.43% for 16-bit and 8-bit precision respectively. On the other hand, the Top-1 accuracy obtained for the CIFAR-10 dataset when implemented on AlexNet is 83.5% and 82% for 16-bit and 8-bit precision respectively. It is also observed that the CNN accuracies for the CIFAR-10 dataset are noticeably lower when compared to the MNIST dataset, also shown in Figure 5. The performance degradation

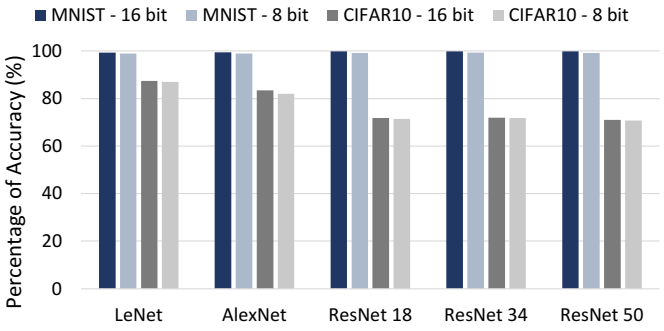

Fig. 5. Comparison of Top-5 accuracies of LeNet, AlexNet, ResNet-18, -34 and -50 on MNIST, CIFAR-10 dataset for 16-bit, 8-bit data precision

Authorized licensed use limited to: George Mason University. Downloaded on September 01,2023 at 00:08:55 UTC from IEEE Xplore. Restrictions apply.

is around 10%-15% for all the CNNs deployed. The accuracy of the CIFAR10 dataset, in general, is significantly lower than MNIST dataset due to the comparatively higher complexity of the dataset. Although higher accuracy with CIFAR-10 is reported in the literature, it is with higher data precision than those adopted in this paper [25].

D. Performance Comparison with State-of-the-Art Hardware Accelerators for CNN Implementation

Performance is evaluated by comparing the proposed architecture with state-of-the-art PIM accelerator architectures in terms of power consumption (Watt) and throughput (Frames/second), as shown in Figure 6.

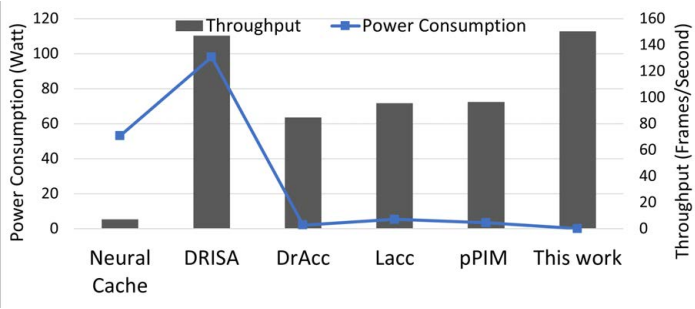

Fig. 6. Comparative performance analysis of proposed multi-functional heterogeneous architecture with respect to state-of-the-art PIM architectures in terms of throughput (Frames/second) and power consumption (Watt)

As a proof of concept, we evaluate and implement AlexNet [1] on the proposed architecture with the 8-bit width precision. The PIM architectures under comparison in this section include DRAM-based bulk bit-wise processing devices DRISA [11], and DrAcc [13], SRAM-implemented Neural Cache [12], LUT-based PIM implemented on the DRAM platforms such as LAcc [14], and pPIM architecture [16].

Among the PIMs studied here, Neural Cache [12] is the slowest due to its limited processing capabilities and comparatively slower bit-serial computing mechanism. On the other hand, a relatively higher throughput is observed for DRISA [11] due to its ability to parallelize operations across multiple banks. Whereas DrAcc [13] implements 8-bit ternary precision inferences through very minimal circuit modifications which allows it to obtain high performance similar to that of pPIM [16]. The benefits of adopting LUTs in order to utilize pre-calculated results instead of performing in-memory logic operations are convincingly demonstrated by LAcc [14], pPIM [16] which achieve impressive inference performances.

The proposed architecture, on the other hand, utilizes the multi-functional heterogeneous memory LUTs to perform the CNN algorithms and is observed to have relatively higher AlexNet throughput than LUT-based PIMs under comparison. It is also observed to have a much higher throughput when compared to other PIM architectures such as DRISA, Dracc, and Neural cache as shown in Figure 6. A similar trend is observed in for power consumption comparison, the proposed architecture is observed to have lower power consumption compared to the PIM architectures as shown in Figure 6. It is also observed that the proposed architecture outperforms LAcc and pPIM by almost 1.5× for AlexNet inference throughput. The proposed architecture is also observed to achieve a maximum of 200× higher energy efficiency than LAcc and pPIM implementation for AlexNet inference.

VI. Conclusion

In order to address the energy efficiency and flexibility requirements for computer architectures, we present a novel multi-functional heterogeneous look-up table-based reconfigurable PIM architecture in this work. The proposed architecture is aimed at CNN and DNN inference applications that support existing and emerging applications with low overheads and high programmability. The proposed hardware accelerator's heterogeneous reconfigurable LUTs enable multi-functional programming to carry out almost any arithmetic or logical operation. As a result, it can process Convolutional, Fully-connected, Activation, and Pooling Layers in a CNN/DNN algorithm. Performance is evaluated by comparing the proposed architecture with state-of-the-art PIM architectures. We have evaluated various CNNs such as LeNet, AlexNet, and ResNet-18,34,50 on the proposed architecture. Our experimental results have demonstrated that when AlexNet is implemented on the proposed architecture, it shows a maximum of 200× higher energy efficiency and 1.5× higher throughput than a DRAM-based LUT-based PIM architecture. Although the proposed architecture is primarily designed for CNN acceleration, its heterogeneous multi-functionality, reconfiguration, and ultra-low latency implementation make it suitable for a wider range of application domains such as real-time IoT, edge devices, mobile applications, automated robots, and automated computers.

VII. Acknowledgements

This work was supported in part by the US National Science Foundation (NSF) Grant CNS-2228239. The views, opinions, and/or findings contained in this article are those of the author(s) and should not be interpreted as representing the official views or policies, either expressed or implied, of the US NSF.

References

[1] M. Z. Alom *et al.*, "The history began from alexnet: A comprehensive survey on deep learning approaches," *arXiv*, 2018.

[2] S.-L. Lu *et al.*, "Scaling the "memory wall": Designer track," in *2012 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2012, pp. 271–272.

[3] S. Rafatirad *et al.*, *Machine Learning for Computer Scientists and Data Analysts: From an Applied Perspective*. Springer Nature, 2022.

[4] Y. Chen *et al.*, "Dadiannao: A machine-learning supercomputer," in *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*, 2014, pp. 609–622.

[5] J. Fowers *et al.*, "A performance and energy comparison of FPGAs, GPUs, and multicores for sliding-window applications," in *ACM/SIGDA Int. Symp. on Field Programmable Gate Arrays*, 2012.

[6] S. Bavikadi *et al.*, "A survey on machine learning accelerators and evolutionary hardware platforms," *IEEE Design & Test*, vol. 39, no. 3, pp. 91–116, 2022.

[7] N. P. Jouppi *et al.*, "In-datacenter performance analysis of a tensor processing unit," *CoRR*, vol. abs/1704.04760, 2017. [Online]. Available: http://arxiv.org/abs/1704.04760

[8] O. Villa *et al.*, "Scaling the power wall: A path to exascale," in *SC '14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, Nov 2014, pp. 830–841.

Authorized licensed use limited to: George Mason University. Downloaded on September 01,2023 at 00:08:55 UTC from IEEE Xplore. Restrictions apply.

[9] A. Ganguly, R. Muralidhar, and V. Singh, "Towards energy efficient non-von neumann architectures for deep learning," in *Int. Symp. on Quality Electronic Design (ISQED)*, 2019.

[10] M. Gao, G. Ayers, and C. Kozyrakis, "Practical near-data processing for in-memory analytics frameworks," 10 2015, pp. 113–124.

[11] S. Li *et al.*, "Drisa: A dram-based reconfigurable in-situ accelerator," in *2017 50th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2017, pp. 288–301.

[12] C. Eckert *et al.*, "Neural cache: Bit-serial in-cache acceleration of deep neural networks," in *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, 2018, pp. 383–396.

[13] Q. Deng *et al.*, "Dracc: a dram based accelerator for accurate cnn inference," in *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*, 2018, pp. 1–6.

[14] Q. Deng *et al.*, "Lacc: Exploiting lookup table-based fast and accurate vector multiplication in dram-based cnn accelerator," *2019 56th ACM/IEEE Design Automation Conference (DAC)*, pp. 1–6, 2019.

[15] K. K. Chang *et al.*, "Low-cost inter-linked subarrays (lisa): Enabling fast inter-subarray data movement in dram," in *IEEE Int. Symp. on High Performance Computer Arch (HPCA)*, March 2016, pp. 568–580.

[16] P. R. Sutradhar *et al.*, "pPIM: A programmable processor-in-memory architecture with precision-scaling for deep learning," *IEEE Computer Architecture Letters*, vol. 19, no. 2, pp. 118–121, 2020.

[17] L. Song *et al.*, "Pipelayer: A pipelined reram-based accelerator for deep learning," in *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2017, pp. 541–552.

[18] S. Angizi *et al.*, "Mrima: An mram-based in-memory accelerator," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 5, pp. 1123–1136, 2020.

[19] T. P. Morgan, "Accelerating compute by cramming it into dram memory," Oct 2019. [Online]. Available: https://www.upmem.com/nextplatform-com-2019-10-03-accelerating-compute-by-cramming-it-into-dram/

[20] S. Bavikadi *et al.*, "A review of in-memory computing architectures for machine learning applications," ser. GLSVLSI '20, 2020.

[21] P. R. Sutradhar *et al.*, "Look-up-table based processing-in-memoryarchitecture with programmable precision-scalingfor deep learning applications," *IEEE TPDS*, 2021.

[22] S. Bavikadi *et al.*, "upim: Performance-aware online learning capable processing-in-memory," in *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2021, pp. 1–4.

[23] S. Bavikadi *et al.*, "Polar: Performance-aware on-device learning capable programmable processing-in-memory architecture for low-power ml applications," in *2022 25th Euromicro Conference on Digital System Design (DSD)*, 2022, pp. 889–898.

[24] G. Yuan *et al.*, "A sot-mram-based processing-in-memory engine for highly compressed dnn implementation," 2019. [Online]. Available: https://arxiv.org/abs/1912.05416

[25] K. Vasquez *et al.*, "Activation Density based Mixed-Precision Quantization for Energy Efficient Neural Networks," *arXiv e-prints*, p. arXiv:2101.04354, Jan. 2021.

[26] V. Seshadri *et al.*, "Rowclone: Fast and energy-efficient in-dram bulk data copy and initialization," in *2013 46th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Dec 2013, pp. 185–197.

[27] Y. Lecun *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[28] K. He *et al.*, "Deep residual learning for image recognition," *arXiv*, 2015.

[29] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, Nov 2012.

[30] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009.

Authorized licensed use limited to: George Mason University. Downloaded on September 01,2023 at 00:08:55 UTC from IEEE Xplore. Restrictions apply.